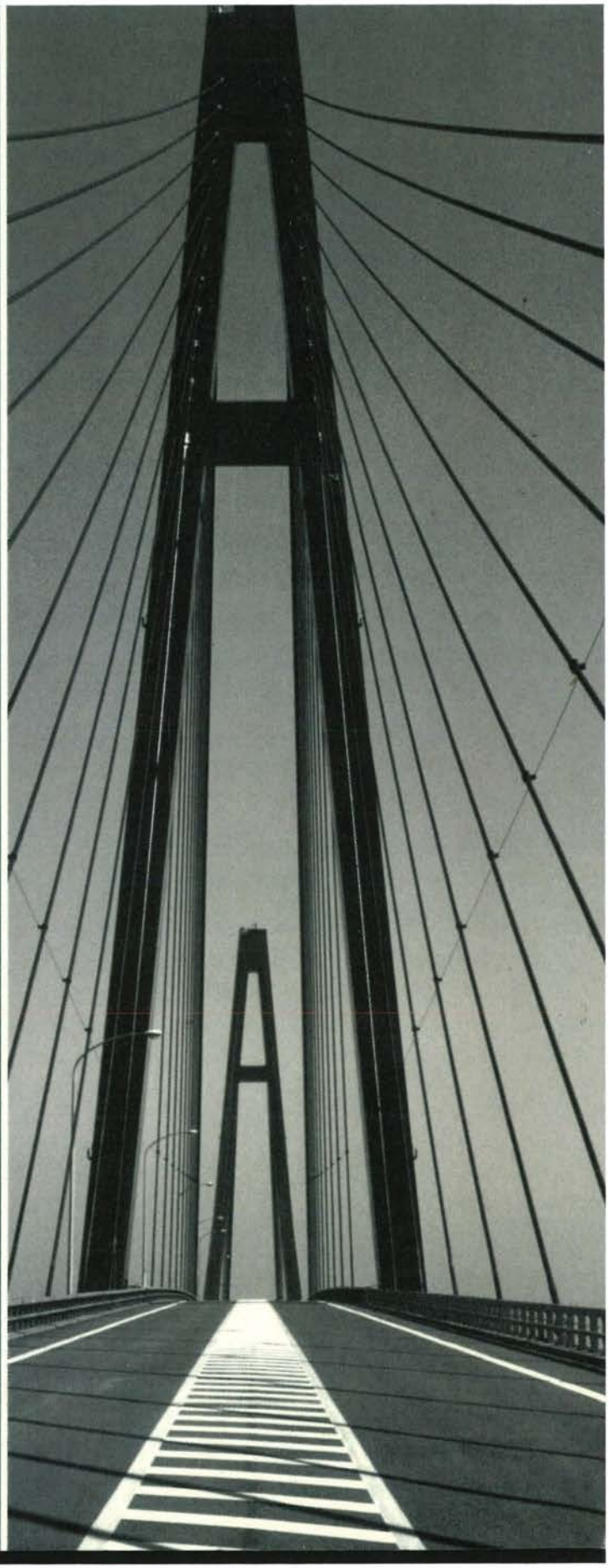
Juha Heiskala
John Terry, Ph.D.

OFDM Wireless LANs: A Theoretical and Practical Guide



SAMS

HONDA 1013

OFDM Wireless LANs: A Theoretical and Practical Guide

John Terry and Juha Heiskala

SAMS

201 West 103rd St., Indianapolis, Indiana, 46290 USA

OFDM Wireless LANs: A Theoretical and Practical Guide

Copyright © 2002 by Sams Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-32157-2

Library of Congress Catalog Card Number: 00-112047

Printed in the United States of America

First Printing: December 2001

04 03 02 01 4 3 2 1

Figures 2.4, 6.7, 6.8, 6.11, and 6.13–6.16 have material in part or full reproduced with permission from IEEE Std. 802.11a-1999, Copyright © 1999, by IEEE. The IEEE disclaims any responsibility or liability resulting from the placement and use in the described manner.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

ASSOCIATE PUBLISHER
Linda Engelman

ACQUISITIONS EDITOR
Dayna Isley

DEVELOPMENT EDITOR
Rebecca Greenberg

MANAGING EDITOR
Charlotte Clapp

PROJECT EDITOR
Carol Bowers

COPY EDITOR
Michael Henry

INDEXER
Ron Strauss

TECHNICAL EDITORS
Tony Reid, Douglas B. Williams

TEAM COORDINATOR
Pamalee Nelson

MEDIA DEVELOPER
Dan Scherf

INTERIOR DESIGNER
Anne Jones

COVER DESIGNER
Aren Howell

PAGE LAYOUT
Octal Publishing, Inc.

Contents at a Glance

Preface ix

1 Background and WLAN Overview 1

2 Synchronization 47

3 Modulation and Coding 87

4 Antenna Diversity 123

5 RF Distortion Analysis for OFDM WLANs 171

6 Medium Access Control (MAC) for IEEE 802.11 Networks 215

7 Medium Access Control (MAC) for HiperLAN/2 Networks 257

8 Rapid Prototyping of WLANs Using FPGA 277

Index 299

Contents

Preface ix

1 Background and WLAN Overview 1

Review of Stochastic Processes and Random Variables 2
Random Variables 3
Ensemble Averages 3
Review of Discrete-Time Signal Processing 6
Discrete-Time Signals 6
Discrete-Time Systems 7
Filtering Random Processes 9
Discrete Fourier Transform (DFT) 11
Components of a Digital Communication System 12
Source Formatting 13
Source Coding 18
Channel Coding 19
Modulation 25
Multiple Access Techniques 25
Channel Model 28
OFDM WLAN Overview 31
MAC for WLAN Standards 32
Physical Layer Specifications for WLAN Standards 36
Single Carrier Versus OFDM Comparison 38
Synchronization Errors 40
Effects of Frequency Errors 41
Bibliography 45

2 Synchronization 47

Timing Estimation 49
Packet Detection 49
Symbol Timing 57
Sample Clock Tracking 62
Frequency Synchronization 66
Post DFT Approach to Frequency Error Estimation 70
Comments on Frequency Error Estimation Algorithms 72
Alternative Techniques for Frequency Error Estimation 73
Carrier Phase Tracking 73
Channel Estimation 77
Frequency Domain Approach for Channel Estimation 77
Time Domain Approach for Channel Estimation 79
Analysis of the Time Domain and Frequency Domain Approaches for Channel Estimation 80
Enhancing the Channel Estimate 81

Clear Channel Assessment 81
Signal Quality 82
Bibliography 83

3 Modulation and Coding 87

Modulation 88
Coherent Modulations 90
Detection of Coherent Modulations 97
Non-Coherent Modulations 98
Linear and Nonlinear Modulation 101
Interleaving 102
Block Interleaver 103
Convolutional Interleaver 104
Interleaving in IEEE 802.11a 104
Channel Codes 106
Performance of IEEE 802.11a 114
Trellis Coded Modulation 118
Block Codes 119
Turbo Codes 119
Bibliography 120

4 Antenna Diversity 123

Background 124
Limits of Capacity in Fading Environments 125
Channel Model for Multiple-Input/Multiple-Output (MIMO) Systems 129
Introducing Diversity 130
Receive Diversity 131
Selection Diversity 131
Maximal Ratio Combining 133
Transmit Diversity 136
Transmit Diversity Design Criteria for Fading Channels 136
Delay Diversity 139
Trellis Space-Time Codes 140
Layered Space-Time Codes 146
Block Space-Time Codes 148
Multidimensional Space-Time Codes 151
Spherical Coding 151
SSTC in OFDM Framework 152
Water-Filling for Single Antenna Systems 154
Water-Filling for Multi-Antenna Systems 157
Adaptive Modulation 158
Bibliography 163

5 RF Distortion Analysis for OFDM WLAN 171

Components of the Radio Frequency Subsystem 172
Amplifier Classification and Distortion 173
Predistortion Techniques for Nonlinear Distortion Mitigation 178
A Decision-Aided Method for Clipping Amplitude Recovery 181
Bayesian Inference Method for Clipping Amplitude Recovery 184
Effects of Amplifier Nonlinearities Without Amplitude Clipping . 187
Adaptive Predistortion Techniques 190
Coding Techniques for Amplifier Nonlinear Distortion Mitigation 197
Partial Transmit Sequence Techniques 198
Modification of the PTS Algorithm 199
Selective Mapping 202
Phase Noise 205
IQ Imbalance 208
Bibliography 210

6 Medium Access Control (MAC) for IEEE 802.11 Networks 215

MAC Overview 216
MAC Services 217
MAC Frames 218
MAC Information Management 219
MAC System Architecture 219
Basic Service Sets 219
Extended Service Sets 221
MAC Frame Formats 222
IEEE 802.11 Frame Format 222
MAC Frame Subtypes 227
Frame Fragmentation 229
MAC Data Services 231
Distributed Coordination Function 231
Point Coordination Function 234
Frame Exchange with RTS/CTS 238
MAC Management Services 239
Power Management 239
Synchronization 243
Session Management 246
Privacy 250
MAC Management Information Base 251
Station Management Attributes 251
MAC Attributes 254
Bibliography 256

7 Medium Access Control (MAC) for HiperLAN/2 Networks 257

Network Architecture 259
DLC Functions 260
MAC Overview 261
Basic MAC Message Formats 262
Transport Channels 264
PDU Trains 266
Logical Channels 268
MAC Frame Structure 269
Order of the Transport Channels 271
Schedule Addressing 271
Building a MAC Frame 273
MAC Frame Processing 274
Access Point 274
Mobile Terminal 275
Bibliography 275

8 Rapid Prototyping for WLANs 277

Introduction to Rapid Prototype Design 278
An Example of a Rapid Prototype Design Flow 280
Good Digital Design Practices 284
Rapid Prototyping of a WLAN System 290
IEEE 802.11a Transmitter 292
IEEE 802.11a Receiver 293
IEEE 802.11a Baseband Demonstrator 295
Bibliography 298

Index 299

Preface

This book will address the subject of broadband communications using orthogonal frequency division multiplexing (OFDM). OFDM is a special case of multicarrier modulation (MCM), which is the principle of transmitting data by dividing the stream into several parallel bit streams and modulating each of these data stream onto individual carriers or subcarriers. Although the origin of MCM dates back to the 1950s and early 1960s with military high frequency (HF) radio links, R.W. Chang in the mid 60s first published a paper demonstrating the concept we today call OFDM. Chang demonstrated the principle of transmission of multiple messages simultaneously through a linear band-limited channel without *interchannel interference* (ICI) and *intersymbol interference* (ISI). The multichannel or OFDM system developed by Chang differed from traditional MCM in that the spectra of the subcarriers were allowed to overlap under the restriction they were all mutually orthogonal. This characteristic of OFDM systems required the abandonment of steep bandpass filters used in older MCM systems to separate the spectra of the individual subcarriers.

Weinsten and Ebert were the first to suggest using the *discrete Fourier Transform* (DFT) and *inverse discrete Fourier Transform* (IDFT) to perform baseband modulation and demodulation in 1971. Currently, OFDM systems utilize the Fast Fourier Transform (FFT) and Inverse FFT to perform modulating and demodulating of the information data. Saltzberg performed a performance analysis of OFDM, shortly after Chang published his paper, and concluded that the dominate impairment in OFDM is ICI. To combat ICI and ISI, Peled and Ruiz introduced the concept of a cyclic prefix (CP). Rather than using an empty guard space, a cyclic extension of the OFDM symbol is used instead. This effectively simulates a channel performing circular convolution as long as the CP is longer than the impulse response of the channel. The penalty of using a CP is loss of signal energy proportional to the length of the CP, yet the benefits of using a CP generally outweighs any loss of signal energy.

Presently, OFDM appears in several standards relating to wireless communications at high data rates such as terrestrial digital audio broadcasting (DAB) and digital video broadcasting (DVB-T) in Europe. Presumably, one of the reasons OFDM was chosen as the DAB standard is that it is possible to deploy single frequency subnetworks within its main networks. Hence, main and relay broadcast transmitters may use the same set of subcarriers. In areas with reception from multiple transmitters, receive diversity gains are experienced. Based on coded OFDM, DVB-T is the youngest and most sophisticated of the three core DVB systems. Combining channel coding with OFDM permits reliable transmission over dispersing channels. Furthermore, the inherent structure of OFDM allows for flexible transmission rates.

Finally, WLAN, the main subject of this book, is another application for OFDM technology. For instance, next generation wireless LAN standards such as IEEE 802.11a, High Performance Local Area Network type 2 (HiperLAN/2), and Mobile Multimedia Access Communication

(MMAC) system have accepted OFDM as their physical layer specifications. These WLAN systems also incorporate coding with OFDM to combat dispersing channels. It has been shown that coded OFDM modulation over modest dispersing channels can improve, rather than deteriorate, the reliability of the transmission. This interesting counterintuitive phenomenon can be attributed to the inherent frequency diversity provided by OFDM. Arguably, this characteristic is the most attractive feature of OFDM.

Interactive Learning

Clearly with the growing in interest in OFDM for high data rate wireless communications, in particular WLANs, there is a need in the technical community for a book that reviews the subject of OFDM WLANs. Typically, this is accomplished in a classroom setting. Unfortunately, many engineers and scientists today cannot afford the time required to attend classes at a university. What is needed is a tool to allow each reader to learn each of the concepts presented in the chapters at his/her own pace. We have provided that by means of an interactive simulation environment. Please visit our Web site at http://www.samspublishing.com and search for the OFDM book. More specifically, the site contains a complete OFDM WLAN physical layer simulation developed in MATLAB. We developed the simulation tool to illustrate the concepts discussed in Chapters 2–5.

To aid in the learning process, exercises are provided in each of these chapters. The exercises require the use of the OFDM system simulation tool and the simple programs you develop. Most of the examples given in this book are reproducible with the simulation program. The OFDM system simulation is executed through a graphical user interface (GUI) to facilitate system reconfigurability. The GUI is called from the MATLAB command window, which allows users to test quickly and easily many of the concepts in this book with a few clicks of the mouse. The novice and expert alike will thoroughly enjoy the endless combinations of test conditions available to them. With this learning tool, readers can further improve their understanding of the concepts presented in this book. In addition, readers interested in testing their algorithms over a WLAN environment will save months of software development time by using the simulation program located at our Web site.

Intended Audience

The primary audiences for this book are engineers and scientists without prior knowledge of OFDM. In the development of the text, we consider our primary audience to fall within two broad categories of readers: novice and advanced. For the novice, we envision someone with a background in engineering, mathematics, and some knowledge of communication theory. For that audience, this book provides the basics of OFDM theory with many examples and illustrations demonstrating concepts. An example novice reader might be a researcher in digital image processing, who is in interested in understanding what effects does an OFDM WLAN network

might have on the quality of the video. Another example of the novice reader could be a radio frequency (RF) engineer, who is interested in the additional requirements imposed by OFDM modulation on the RF subsystems in the access point (AP) and mobile terminal (MT).

An example of an advanced reader is an engineer or scientist familiar with basic OFDM concepts. For those individuals, this book is intended as a source for practical guidelines as well as introductory material of advanced research topics in OFDM.

The secondary audiences for this book include individuals, such as network system engineers, product engineers, or managers, for whom some of the mathematical development presented in this text is slightly beyond their scope of understanding. For those individuals, explanatory text is provided throughout this book to give an intuitive feel of many of the concepts discussed.

It is assumed that the all audiences have a background in calculus, physics, and random and stochastic processes. Thus, the majority of the text in this book is written at the undergraduate level, with the exception of the advanced research topics, which are written at the first-year graduate level. In addition, the reader will be provided in each chapter all the relevant mathematical foundations necessary to understand the OFDM principles discussed. As mentioned earlier, explanatory text is also given to provide a better understanding of these OFDM principles from the mathematical expressions.

A final point concerning the audience: to reap the fullest benefit of this book, it is advantageous to the reader to become proficient in the use of MATLAB.

We expect this book to attract a broad range of readers, as it is written to do so. Certainly, no book can be all things to everyone. However, no matter your interest level in OFDM WLANs, this book has some insight to offer.

Organization of this Book

This book is organized as follows. Chapter 1, "Background and WLAN Overview," is dedicated to background material as well as an overview of OFDM WLANs. The background material covers relevant concepts in digital signal and stochastic processing. It expected that readers will refer to this chapter as needed to understand the concepts in latter chapters. Chapters 2–5 focus on the physical layer specifications of OFDM WLAN. Chapter 2, "Synchronization," provides a detailed discussion of many of the popular synchronization algorithms used in OFDM networks. Specifically, timing synchronization algorithms, which include packet detection, symbol timing recovery, and sample clock tracking, are covered. Also covered are frequency, channel estimation, and clear channel assessment (CCA) algorithms. Chapter 3, "Modulation and Coding," provides a brief overview of modulation and coding techniques. In particular, the phase-shift keying (PSK) and quadrature amplitude modulations (QAM) found in OFDM WLAN standards are covered. With respect of channel coding, discussions on block and convolutional codes are provided. Performance evaluation of several operational modes of the IEEE 802.11a physical specification

are given. Chapter 3 can be thought of as the central theme or key technology area of current OFDM WLAN systems.

Chapter 4, "Antenna Diversity," is dedicated to the central theme or key technology area of future OFDM WLAN, antenna diversity. Several popular transmit and receive diversity schemes are discussed in their context to OFDM systems. Examples show that drastic improvement in error rate performance is achievable when these techniques are deployed. Chapter 5, "RF Distortion Analysis for OFDM WLANs," focuses on the system impairments of the OFDM system resulting from RF nonlinearities. Particularly attention is given to the peak-to-average power (PAPR) problem present in all OFDM systems. In this chapter, a survey of the more popular techniques to handle this problem is analyzed. In addition, other system impairments such as phase noise and in-phase and quadrature (IQ) imbalances are covered.

In Chapters 6 and 7, an introduction of the medium access control (MAC) layer is given. Chapter 6, "Medium Access Control (MAC) for IEEE 802.11 Networks," summarizes the IEEE 802.11a MAC, while Chapter 7, "Medium Access Control (MAC) for HiperLAN/2 Networks," summarizes the HiperLAN/2 MAC. Both chapter details of the interaction between the MAC layer and the physical layer.

Interestingly, a major criticism of OFDM has been the complexity issues associated with real-time implementation of the FFT and IFFT. However, steady improvements in semiconductor process technology has allowed for real-time prototyping of OFDM systems with Field Programmable Gate Array (FPGA) technology and cost effective solutions with Application Specific Integrated Circuit (ASIC) technology. Chapter 8, "Rapid Prototying of WLANs Using FPGA," is dedicated to the issues associated with real-time prototyping of an IEEE 802.11a radio using FPGA technology.

About the Authors

John Terry, Ph.D., is a senior research engineer at Nokia Research Center in Dallas, TX. He is currently managing the OFDM modulation and coding project in the High Speed Access (HSA) group. Dr. Terry has published several conference and journal papers, given numerous presentations on wireless communications, and generated six pending patents related to OFDM wireless LANs. He has 12 years of experience working in wireless communications, including tenures at NASA Glen Research Center and Texas Instruments, Inc. In addition, Dr. Terry is the vice-chair of the IEEE 802.11 Task Group g and serves as a technical reviewer for several conference and journal publications of the IEEE in wireless communications.

Juha Heiskala is a senior research engineer at Nokia Research Center in Dallas, TX. He is active in the IEEE 802.11 standards bodies and has been tasked with developing the 802.11a system simulation on several software platforms. He is the inventor/co-inventor of three pending patents in the area of OFDM LANs and co-designed with Dr. John Terry the modulation and coding scheme for achieving 100 Mbps speeds within currently allocated bandwidth specifications for OFDM wireless LANs.

About the Technical Editors

Douglas Williams received B.S.E.E., M.S., and Ph.D. degrees in electrical and computer engineering from Rice University, Houston, Texas, in 1984, 1987, and 1989, respectively. In 1989, he joined the faculty of the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, Georgia, where he is currently an Associate Professor. There he is also affiliated with the Center for Signal and Image Processing and teaches courses in signal processing and telecommunications. Dr. Williams has served as an Associate Editor of the IEEE Transactions on Signal Processing and is a member of the IEEE Signal Processing Society's Signal Processing Theory and Methods technical committee. He was co-editor of the *Digital Signal Processing Handbook* published in 1998 by CRC Press and IEEE Press. Dr. Williams is a member of the Tau Beta Pi, Eta Kappa Nu, and Phi Beta Kappa honor societies.

Anthony Reid received his B.S.E.E. (cum laude), from Rensselaer Polytechnic Institute, Troy, New York, M.S.E.E., from Stanford University, Palo Alto, California and Ph.D., from Southern Methodist University (S.M.U.), Dallas, Texas. Dr. Reid is currently a Principal Research Scientist at Nokia Research Center doing applied research in modulation, coding and equalization for 3-G (third generation) wireless systems. He also worked at Nortel Networks on Advanced Network Architectures for 3-G wireless networks. Before joining Nortel and Nokia, Tony was an Engineering Fellow and Branch Manager of Advanced Signal Processing in the Systems Technology Center in Raytheon Systems Corp. (RSC). This group developed broadband communications systems capability for both military and commercial applications. In the commercial arena capabilities were developed for high-speed SATCOM modems. In the past, this group developed image and signal processing algorithms for autonomous detection/tracking of military targets in infrared imagery and radar.

Dedication

To my beautiful and loving wife Sonya Terry, my son Amiel Terry, and the fond memory of my grandfather John D. Terry, all of whom made it possible for this work to be completed.

—John

To my parents.

—Juha

Acknowledgments

The authors of this book will like to thank the reviewers, Dr. Tony Reid from Nokia Research Center in Dallas and Prof. Douglas Williams from Georgia Institute of Technology, for their invaluable comments, which greatly improved the content of this book. Finally, the authors would like to express their deepest gratitude to Nokia, Inc. for providing the opportunity to work in this new and exciting area of wireless communications.

Tell Us What You Think!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an Associate Publisher for Sams Publishing, I welcome your comments. You can fax, e-mail, or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.

When you write, please be sure to include this book's title and author as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

Fax: 317-581-4770

Email: `feedback@samspublishing.com`

Mail: Linda Engelman
Associate Publisher
Sams Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA

Background and WLAN Overview

CHAPTER 1

IN THIS CHAPTER

- Review of Stochastic Processes and Random Variables 2
- Review of Discrete-Time Signal Processing 6
- Components of a Digital Communication System 12
- OFDM WLAN Overview 31
- Single Carrier Versus OFDM Comparison 38
- Bibliography 45

Before delving into the details of orthogonal frequency division multiplexing (OFDM), relevant background material must be presented first. The purpose of this chapter is to provide the necessary building blocks for the development of OFDM principles. Included in this chapter are reviews of stochastic and random process, discrete-time signals and systems, and the Discrete Fourier Transform (DFT). Tooled with the necessary mathematical foundation, we proceed with an overview of digital communication systems and OFDM communication systems. We conclude the chapter with summaries of the OFDM wireless LAN standards currently in existence and a high-level comparison of single carrier systems versus OFDM.

The main objective of a communication system is to convey information over a channel. The subject of digital communications involves the transmission of information in digital form from one location to another. The attractiveness of digital communications is the ease with which digital signals are recovered as compared to their analog counterparts. Analog signals are continuous-time waveforms and any amount of noise introduced into the signal bandwidth can not be removed by amplification or filtering. In contrast, digital signals are generated from a finite set of discrete values; even when noise is present with the signal, it is possible to reliably recover the information bit stream exactly. In the sections to follow, brief reviews of stochastic random processes and discrete-time signal processing are given to facilitate presentation of concepts introduced later.

Review of Stochastic Processes and Random Variables

The necessity for reviewing the subject of random processes in this text is that many digital communication signals [20, 21, 22, 25] can be characterized by a *random or stochastic process*. In general, a signal can be broadly classified as either *deterministic* or *random.* Deterministic signals or waveforms can be known precisely at instant of time, usually expressed as a mathematical function of time. In constrast, random signals or waveforms always possess a measure of uncertainty in their values at any instant in time since random variables are rules for assigning a real number for every outcome ξ of a probabilistic event. In other words, deterministic signals can be reproduced exactly with repeated measurements and random signals cannot.

A stochastic or random process is a rule of correspondence for assigning to every outcome ξ to a function $X(t,\xi)$, where t denotes time. In other words, a stochastic process is a family of time-functions that depends on the parameter ξ. When random variables are observed over very long periods, certain regularities in their behavior are exhibited. These behaviors are generally described in terms of probabilities and statistical averages such as the mean, variance, and correlation. Properties of the averages, such as the notion of stationarity and ergodicity, are briefly introduced in this section.

Random Variables

A random variable is a mapping between a discrete or continuous random event and a real number. The *distribution function,* $F_X(\alpha)$, of the random variable, X, is given by

$$F_X(\alpha) = \Pr(X \leq \alpha) \tag{1.1}$$

where $\Pr(X \leq \alpha)$ is the probability that the value taken on by the random variable X is less than or equal to a real number α. The distribution function $F_X(\alpha)$ has the following properties:

- $0 \leq F_X(\alpha) \leq 1$
- $F_X(\alpha) \leq F_X(\beta)$ if $\alpha \leq \beta$
- $F_X(-\infty) = 0$
- $F_X(+\infty) = 1$

Another useful statistical characterization of a random variable is the *probability density function* (pdf), $f_X(\alpha)$, defined as

$$f_X(\alpha) = \frac{\partial}{\partial \alpha} F_X(\alpha) \tag{1.2}$$

Based on properties of $F_X(\alpha)$ and noting the relationship in Equation 1.2, the following properties of the pdf easily deducted:

- $f_X(\alpha) \geq 0$
- $\int_{-\infty}^{\infty} f_X(\alpha)\partial\alpha = F_X(+\infty) - F_X(-\infty) = 1$

Thus, the pdf is always a nonnegative function with unit area.

Ensemble Averages

In practice, complete statistical characterization of a random variable is rarely available. In many applications, however, the average or *expected value* behavior of a random variable is sufficient. In latter chapters of this book, emphasis is placed on the expected value of a random variable or function of a random variable. The mean or expected value of a continuous random variable is defined as

$$m_X = E\{X\} = \int_{-\infty}^{\infty} \alpha f_X(\alpha)\partial\alpha \tag{1.3}$$

and a discrete random variable as

$$m_X = E\{X\} = \sum_k \alpha_k \Pr(X = \alpha_k) \tag{1.4}$$

where $E\{\cdot\}$ is called the expected value operator. A very important quantity in communication systems is the *mean squared value* of a random variable, X, which is defined as

$$E\left\{X^2\right\} = \int_{-\infty}^{\infty} \alpha^2 f_X(\alpha)\partial\alpha \tag{1.5}$$

for continuous random variables and

$$E\left\{X^2\right\} = \sum_k \alpha_k^2 \Pr(X = \alpha_k) \tag{1.6}$$

for discrete random variables. The mean squared value of a random variables is a measure of the average power of a random variable. The *variance* of X is the mean of the second central moment and defined as

$$\sigma_X^2 = E\left\{(X - m_X)^2\right\} = \int_{-\infty}^{\infty} (\alpha - m_X)^2 f_X(\alpha)\partial\alpha \tag{1.7}$$

Note a similar definition holds for the variance of discrete random variables by replacing the integral with a summation. The variance is a measure of the "random" spread of the random variable X. Another well-cited characteristic of a random X is its *standard deviation* σ_X, which is defined as the square root of its variance. One point worth noting is the relationship between the variance and mean square value of a random variable, i.e.,

$$\begin{aligned} \sigma_X^2 &= E\left\{X^2 - 2Xm_X + m_X^2\right\} \\ &= E\left\{X^2\right\} - 2E\{X\}m_X + m_X^2 \\ &= E\left\{X^2\right\} - m_X^2 \end{aligned} \tag{1.8}$$

In view of Equation 1.8, the variance is simply the difference between the mean square value and the square of the mean.

Two additional ensemble averages importantance in the study of random variables are the correlation and covariance. Both quantities are expressions of the interdependence of two or more random variables to each other. The correlation between complex random variables X and Y, r_{XY}, is defined as

$$r_{XY} = E\left\{XY^*\right\} \tag{1.9}$$

where * denotes the complex conjugate of the complex random variable. A closely related quantity to the correlation of between random variables is their covariance c_{XY}, which it is defined as

$$c_{XY} = E\{[X - m_X][Y - m_Y]^*\} = E\{XY^*\} - m_X m_Y^* \tag{1.10}$$

Clearly, if either X or Y has zero mean, the covariance is equal to the correlation. The random variables X and Y need not stem from separate probabilistic events; in fact, X and Y can be samples of the same event A observed at two different time instants t_1 and t_2. For this situation, the correlation r_{XY} and covariance c_{XY} become the autocorrelation $R_X(t_1, t_2)$ and autocovariance $C_X(t_1, t_2)$, respectively, which are defined as

$$\begin{aligned} R_X(t_1, t_2) &= E\{X(t_1)X^*(t_2)\} \\ C_X(t_1, t_2) &= E\{[X(t_1) - m_X(t_1)][X(t_2) - m_X(t_2)]^*\} \end{aligned} \tag{1.11}$$

Thus, the autocorrelation and autocovariance are measures of the degree to which two time samples of the same random process are related.

There are many examples of random variables that arise in which one random variable does not depend on the value of another. Such random variables are said to be statistically independent. A more precise expression of the meaning of statistical independence is given in the following definition.

Definition 1 *Two random variables X and Y are said to be statistically independent if the joint probability density function is equal to the product of the individual pdfs, i.e.,*

$$f_{XY}(\alpha, \beta) = f_X(\alpha) f_Y(\beta) \tag{1.12}$$

A weaker form of independence occurs when the correlation r_{XY} between two random variable is equal to the product of their means, i.e.,

$$r_{XY} = E\{XY^*\} = m_X m_Y^* \tag{1.13}$$

Two random variables that satisfy Equation 1.13 are said to be *uncorrelated.* Note that since

$$c_{XY} = r_{XY} - m_X m_Y^* \tag{1.14}$$

then two random variables X and Y will be uncorrelated if their covariance is zero. Note, statistically independent random variables are always uncorrelated. The converse, however, is usually not true in general.

Up to this point, most of the discussions have focused on random variables. In this section, we focus on random processess. Previously, we stated that a random process is a rule of correspondence for assigning to every outcome ξ of a probabilistic event to a function $X(t, \xi)$. A collection of $X(t, \xi)$ resulting from many outcomes defines an ensemble for $X(t, \xi)$. Another, more useful,

definition for a random process is an indexed sequence of random variables. A random processes is said to be *stationary* in the *strict-sense* if none of its statistics are affected by a shift in time origin. In other words, the statistics depend on the length of time it is observed and not when it is started. Furthermore, a random process is said to be *wide-sense stationary* (WSS) if its mean and variance do not vary with a shift in the time origin, i.e.,

$$m_X = E\{X(k)\} = a\ constant, \quad \forall k$$

and

$$R_X = (\tau + k, k) = R_X(\tau)$$

Strict-sense stationarity implies wide-sense stationary, but not vice versa. Most random processes in communication theory are assumed WSS. From a practical view, it is not necessary for a random process to be stationary for all time, but only for some observation interval of interest. Note that the autocorrelation function for a WSS process depends only on time difference τ. For zero mean WSS processes, $R_X(\tau)$ indicates the time over which samples of the random process X are correlated. The autocorrelation of WSS processes has the following properties:

- $R_X(\tau) = R_X(-\tau)$
- $R_X(\tau) \leq R_X(0)$ for all τ
- $R_X(0) = E\{X^2(t)\}$

Unfortunately, computing m_X and $R_X(\tau)$ by ensemble averaging requires knowledge of a collection realizations of the random process, which is not normally available. Therefore, time averages from a single realization are generally used. Random processes whose time averages equal their ensemble averages are known as *ergodic processes.*

Review of Discrete-Time Signal Processing

In this brief overview of discrete-time signal processing, emphasis is placed on the specification and characterization of discrete-time signals and discrete-time systems. The review of stochastic processes and random variables was useful to model most digital communication signals. A review of linear discrete-time signal processing, on the other hand, is needed to model the effects of the channel on digital communication signals. Digital-time signal processing is a vast and well-documented area of engineering. For interested readers, there are several excellent texts [7, 10, 18] that give a more rigorous treatment of discrete-time signal processing to supplement the material given in this section.

Discrete-Time Signals

A discrete-time signal is simply an indexed sequence of real or complex numbers. Hence, a random process is also a discrete-time signal. Many discrete-time signals arise from sampling a

continuous-time signal, such as video or speech, with an analog-to-digital (A/D) converter. We refer interested readers to "Discrete-Time Signals" for further details on A/D converters and sampled continuous-time signals. Other discrete-time signals are considered to occur naturally such as time of arrival of employees to work, the number of cars on a freeway at an instant of time, and population statistics. For most information-bearing signals of practical interest, three simple yet important discrete-time signals are used frequently to described them. These are the unit sample, unit step, and the complex exponential. The *unit sample,* denoted by $\delta(n)$, is defined as

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{1.15}$$

The unit sample may be used to represent an arbitrary signal as a sum of weighted sample as follows

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{1.16}$$

This decomposition is the discrete version of the *sifting property* for continuous-time signals. The *unit step*, denoted by $u(n)$, defined as

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \tag{1.17}$$

and is related to the unit sample by

$$u(n) = \sum_{k=-\infty}^{\infty} \delta(k) \tag{1.18}$$

Finally, the *complex exponential* is defined as

$$e^{jn\omega_0} = \cos(n\omega_0) + j\sin(n\omega_0) \tag{1.19}$$

where ω_0 is some real constant measured in radians. Later in the book, we will see that complex exponentials are extremely useful for analyzing linear systems and performing Fourier decompositions.

Discrete-Time Systems

A discrete-time system is a rule of correspondence that transforms an input signal into the output signal. The notation $T[\cdot]$ will be used to represent a general transformation. Our discussions shall be limited to a special class of discrete-time systems called *linear shift-invariant* (LSI) systems.

As a notation aside, discrete-time systems are usually classified in terms of properties they possess. The most common properties include linearity, shift-invariance, causality, and stability, which are described below.

Linearity and Shift-Invariance

Two of the most desirable properties of discrete-time system for ease of analysis and design are *linearity* and *shift-invariance*. A system is said to be *linear* if the response to the superposition of weighted input signals is the superposition of the corresponding individual outputs weighted in accordance to the input signals, i.e.,

$$y = T[\alpha x_1 + \beta x_2] = \alpha T[x_1] + \beta T[x_2] \tag{1.20}$$

A system is said to be shift-invariant if a shift in the input by n_0 results in a shift in the output by n_0. In other words, shift-invariance means that the properties of the system do not change with time.

Causality

A very important property for real-time applications is *causality*. A system is said to be *causal* if the response of the system at time n_0 does not depend of future input values, i.e.,

$$y(n_0) = T[x(n-\tau)], \qquad 0 \le \tau \le \infty \tag{1.21}$$

Thus, for a causal system, it is not possible for changes in the output to precede changes in the input.

Stability

In many applications, it is important for a system to have a response that is bounded in amplitude whenever the input is bounded. In other words, if the unit sample response of LSI system is absolutely summable, i.e.,

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \tag{1.22}$$

then for any bounded input $|x(n)| \le A < \infty$ the output is bounded $|y(n)| \le B < \infty$. This system is said to be *stable* in the Bounded-Input Bounded-Output (BIBO) sense. There are many other definitions for stability for a system, which can found in [7, 18]; however, BIBO is one of the most frequently used.

Thus far, we have discussed only the properties of LSI systems without providing an example of it. Consider an LSI system whose q coefficients are contained in the vector h. The response of the system to an input sequence $x(n)$ is given by the following relationship

$$y(n) = \sum_{k=0}^{q} x(k)h(n-k) \tag{1.23}$$

is referred to as a *Finite length Impulse Response* (FIR) system. Notice that the output depends only on the input values in the system. It is possible, however, for the output of the system to depend on past outputs of the system as well as the current inputs, i.e.,

$$y(n) = \sum_{k=0}^{q} x(k)h(n-k) - \sum_{k=1}^{p} y(k)g(n-k) \tag{1.24}$$

This type of system is referred to as an *Infinite length Impulse Response* (IIR) system.

Filtering Random Processes

Earlier we have mentioned that all digital communication signals can be viewed as random processes. Thus, it is important to qualify the effects filtering has on the statistics of a random process. Of particular importance are LSI filters because of their frequent use in signal representation, detection, and estimation. In this section, we examine the effects of LSI filtering on the mean and autocorrelation of an input random process.

Let $x(n)$ be a WSS random process with mean m_x, and autocorrelation $R_x(n)$. If $x(n)$ is filtered by a stable LSI filter having a unit sample response $h(n)$, the output $y(n)$ is a random process that is related to input random process $x(n)$ via the convolution sum

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{1.25}$$

The mean of $y(n)$, m_y, is found by taking the expected value of Equation 1.25 as follows,

$$E\{y(n)\} = E\left\{\sum_{k=-\infty}^{\infty} h(k)x(n-k)\right\} = \sum_{k=-\infty}^{\infty} h(k)E\{x(n-k)\}$$

$$m_y = E\{y(n)] = m_x \sum_{k=-\infty}^{\infty} h(k) = m_x H\left(e^{j0}\right)$$

where $H(e^{j0})$ is the zero-frequency response, or non-time varying response for the filter. Hence, the mean of $y(n)$ is a constant equal to the mean $x(n)$ scaled by the sample average of the unit sample response.

The autocorrelation of $y(n)$, is derived and understood best by first proceeding with the cross-correlation r_{yx} between $x(n)$ and $y(n)$, which is given by

$$r_{yx}(k) = E\{y(n+k)x^*(n)\} = E\left\{\sum_{k=-\infty}^{\infty} h(l)x(n+k-l)x^*(n)\right\}$$
$$= \sum_{k=-\infty}^{\infty} h(l)E\{x(n+k-l)x^*(n)\} \quad (1.26)$$
$$= \sum_{k=-\infty}^{\infty} h(l)r_x(k-l) = r_x(k) * h(k)$$

Recall, under the assumption of WSS, the cross-correlation r_{xy} depends only on the difference between sampling instants. It is interesting to note that the cross-correlation r_{xy} as defined in Equation 1.26 is just the convolution of the input autocorrelation $R_x(n)$ with the channel impulse response $h(n)$. We will use this result, shortly, to relate the output autocorrelation $R_y(n)$ to input autocorrelation $R_x(n)$.

The output autocorrelation is defined as

$$R_y(n) = E\{y(n+k)y^*(k)\} = E\left\{y(n+k)\sum_{l=-\infty}^{\infty} x^*(l)h^*(k-l)\right\}$$
$$= \sum_{l=-\infty}^{\infty} h^*(k-l)E\{y(n+k)x^*(l)\} \quad (1.27)$$
$$= \sum_{l=-\infty}^{\infty} h^*(k-l)r_{yx}(n+k-l)$$

Inspection of Equation 1.27 reveals that the autocorrelation of $y(n)$, is really a convolution sum. Changing the index of summation by setting $m = l - k$, we obtain

$$R_y(n) = \sum_{l=-\infty}^{\infty} h^*(-m)r_{yx}(n-m) = r_{yx}(n) * h^*(-n) \quad (1.28)$$

Combining Equations 1.26 and 1.28 we have

$$R_y(n) = R_x(n) * h(n) * h^*(-n) \quad (1.29)$$

Equation 1.29 is a key result exploited often in the reception of wireless communication signals. Figure 1.1 illustrates the concepts expressed in Equations 1.26 and 1.28.

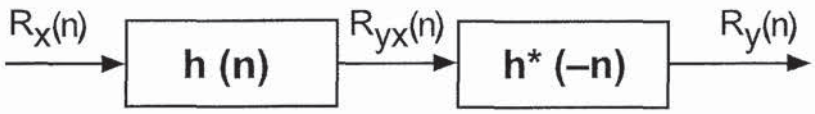


FIGURE 1.1
Input-output autocorrelation for filtered random processes.

Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) is, arguably, the most widely used design and analysis tool in electrical engineering. For many situations, frequency-domain analysis of discrete-time signals and systems provide insights into their characteristics that are not easily ascertainable in the time-domain. The DFT is a discrete version of the discrete-time Fourier transforms (DTFT); that is, the DTFT is a function of a continuous frequency variable, whereas the DFT is a function of a discrete frequency variable. The DFT is useful because it is more amenable to digital implementations. The *N-point* DFT of a finite length sequence $x(n)$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \tag{1.30}$$

Clearly, it can be seen from Equation 1.30 that the DFT is a sample version of the DTFT, i.e.,

$$X(k) = X(\omega)\Big|_{\omega=2\pi k/N} \tag{1.31}$$

where

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-jn\omega} \tag{1.32}$$

The DFT has an inverse transformation called the *inverse DFT* (IDFT). The IDFT provides a means of recovering the finite length sequence $x(n)$ through the following relationship,

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \tag{1.33}$$

Let's now consider the DFTs of the discrete-time signals given in "Discrete-Time Signals," since they form the basic building blocks to generate more complex signals. Using the definition in Equation 1.15, the DFT of the unit sample $\delta(n)$ becomes

$$X(k) = \sum_{n=0}^{N-1} \delta(n)e^{-j2\pi kn/N} = 1 \tag{1.34}$$

Using the definition of the unit step given in Equation 1.17, its DFT is given by

$$X(k) = \sum_{n=0}^{N-1} e^{-j2\pi kn/N} = N\delta(k) \tag{1.35}$$

The result in Equation 1.35 follows directly since, with the exception of the $n = 0$ term, each of the complex exponentials sums to zero over the sample period of length N. Finally, the DFT of the complex exponential is given by

$$X(k) = \sum_{n=0}^{N-1} e^{jn\omega_0} e^{-j2\pi kn/N} = N\delta(k - \frac{N\omega_0}{2\pi}) \tag{1.36}$$

Besides the DFTs given above, some useful properties of the discrete Fourier transforms that facilitate communication system analysis are summarized in Table 1.1. where $\circledast$ denotes circular convolution, $(\cdot)_N$ denotes modulo-N operation, and * denotes the complex conjugate as previously defined. This concludes the review of the mathematical background material.

TABLE 1.1 Some Useful Properties of the DFT

Discrete-Time Signal		*DFT*
$x(n)$	$\leftrightarrow$	$X(k)$
$x(n-n_0)$	$\leftrightarrow$	$X(k)e^{-j2\pi kn_0/N}$
$\alpha x_1(n) + \beta x_2(n)$	$\leftrightarrow$	$\alpha X_1(k) + \beta X_2(k)$
$x^*(n)$	$\leftrightarrow$	$X((-k))_N$
$y(n) \circledast x(n)$	$\leftrightarrow$	$Y(k)X(k)$
$x^*((-n))_N$	$\leftrightarrow$	$X^*(k)$
$e^{-j2\pi mn/N}x(n)$	$\leftrightarrow$	$X((k-m))_N$

Components of a Digital Communication System

In this section, the basic elements of a digital communication system are reviewed. The fundamental principle that governs digital communications is the "divide and conquer" strategy. More specifically, the information source is divided into its smallest intrinsic content, referred to as a *bit*. Then each bit of information is transmitted reliably across the channel. In general, the information source may be either analog or digital. Analog sources are considered first since they require an additional processing step before transmission. Examples of analog sources used in our everyday lives are radios, cameras, and camcorders. Each of these devices is capable of generating analog signals, such as voice and music in the case of radios and video images in the case

of camcorders. Unfortunately, an analog signal can not be transmitted directly by means of digital communications; it must be first converted into a suitable format. With reference to the top chain of Figure 1.2, each basic element and its corresponding receiver function will be reviewed in the order they appear left to right.

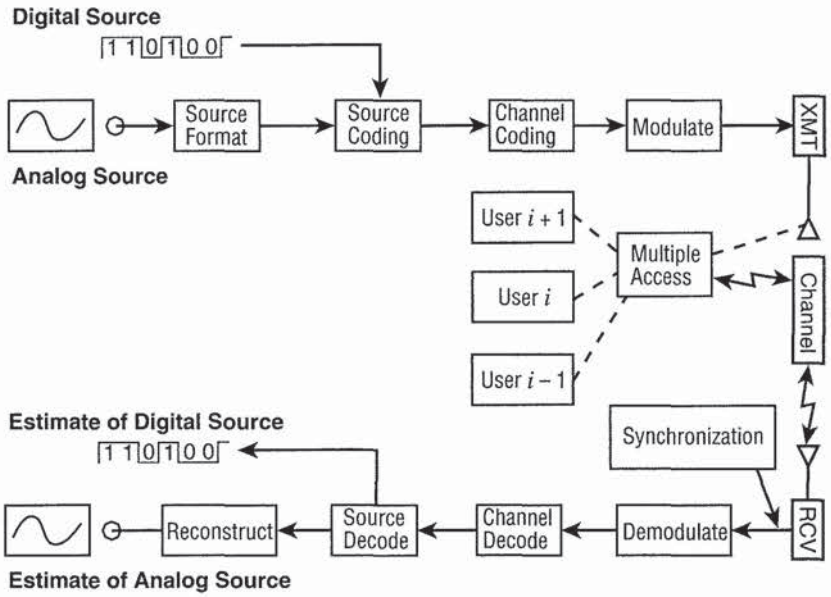


FIGURE 1.2
Basic elements of a digital communication system.

Source Formatting

Source formatting is the process by which an analog signal or continuous-time signal is converted digital signal or discrete-time signal. The device used to achieve this conversion is referred to as *analog-to-digital (A/D)* converter. The basic elements of an A/D converter shown in Figure 1.3 consists of a sampler, quantizer, and encoder. The first component, the sampler, extracts sample values of the input signal at the sampling times. The output of the sampler is a discrete-time signal but with a continuous-valued amplitude. These signals are often referred to as *sampled data signals*; refer to Figure 1.4 for an illustration. Digital signals, by definition, are not permitted to have continuous-valued amplitudes; thus, the second component, the quantizer, is needed to quantize the continuous range of sample values into a finite number of sample values. Finally, the encoder maps each quantized sample value onto a digital word.

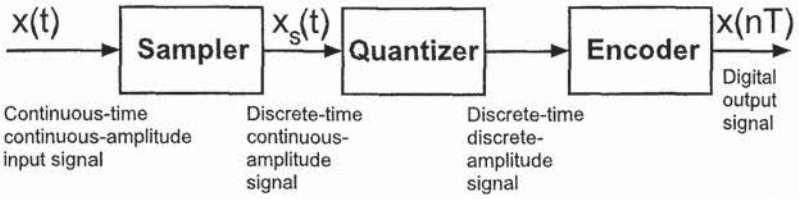


FIGURE 1.3
Basic elements of A/D converter.

Sampling and Reconstruction

To model the sampling operation of a continuous-time signal, we make use of a variant of the unit sample function $\delta(n)$ defined in "Discrete-Time Signals." An ideal sampler is a mathematical abstraction but is useful for analysis and given by

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{1.37}$$

where T_s is the sampling interval. Now, the sampled data signal $x_s(t)$ can be expressed as

$$x_s(t) = x(t)p(t) \tag{1.38}$$

where $x(t)$ is the continuous-time continuous-amplitude input signal as seen in Figure 1.4. Obviously, the more samples of the $x(t)$ we have, the easier it will be to reconstruct the signal. Each sample will be eventually transmitted over the channel. In order to save bandwidth, we would like to send the bare minimum number of samples needed to reconstruct the signal. The sampling rate that produces this is called the *Nyquist rate.* Nyquist sampling theorem states that samples taken at a uniform rate $2f_h$ where f_h is the highest frequency component of a band-limited signal $x(t)$, is sufficient to completely recover the signal. The Nyquist Sampling Theorem is conceptually illustrated in Figure 1.5. Notice first that sampling introduces periodic spectral copies of the original spectrum center at the origin. Second, the copies are sufficiently spaced apart such that they do not overlap. This simple principle is the essence of the Nyquist Sampling Theorem. If the spectral copies of the spectrum were permitted to overlap, aliasing of the spectral copies would occur. It would then be impossible to recover the original spectrum by passing it through a low-pass filter whose ideal frequency response is represented by the dashed box.

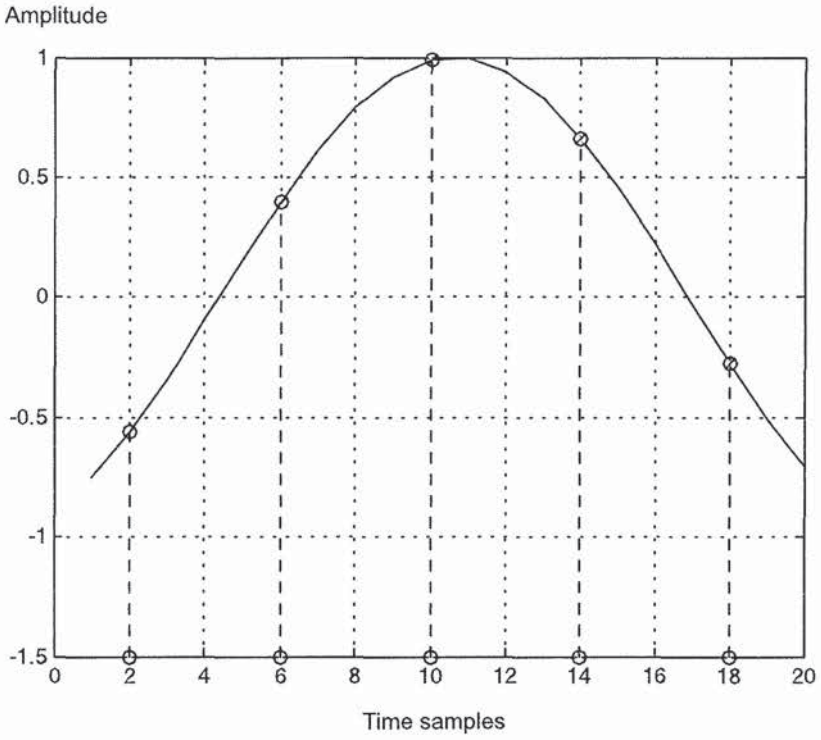


FIGURE 1.4
An example of a discrete-time continuous amplitude signal $X_s(t)$.

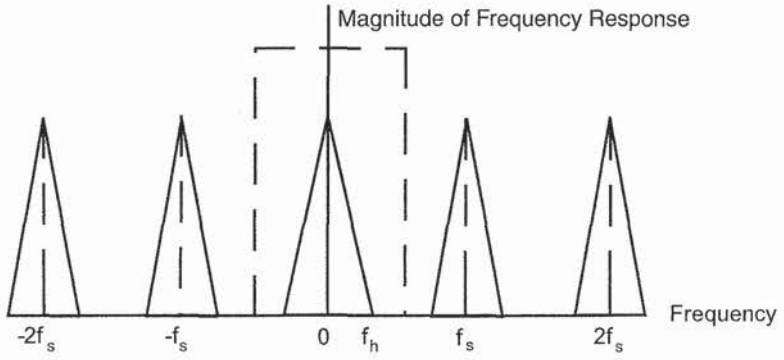


FIGURE 1.5
Spectrum of a sampled waveform.

Quantization and Encoding

After the signal has been sampled, the amplitude values must be quantized in a discrete set of values. The process of quantization entails rounding off the sample value to the nearest of a finite set of permissible values. Encoding is accomplished by assigning a digital word of length k, which represents the binary equivalent for the numerical value of the quantization level, as depicted in Figure 1.6. For example, at sampling time T, the amplitude of the signal lies within quantization level 12, which has a binary representation of 1101. Furthermore, the number of quantization levels q and the digital word length k are related by

$$q = 2^k$$

Another key observation that should be taken from Figure 1.6 is that the quantization process introduces an error source usually referred to as *quantization noise.* When decoding, it is usually assumed the amplitude falls at the center of the quantization level. Hence, the maximum error which can occur is $\pm\frac{1}{2}\Delta L$, where

$$\Delta L = \frac{A}{2^k} \tag{1.39}$$

and A is the difference between the maximum and minimum values of the continuous-time signal. If we assume a large number of quantization levels, error function is nearly linear within the quantization level, i.e.,

$$e(t) = \frac{\Delta L}{2\tau} t \tag{1.40}$$

where 2τ is the time the continuous-time signal remaining within the quantization level. The mean-square error P_n is thereby given by

$$P_n = \frac{1}{2\tau}\int_{-\tau}^{\tau} e^2(\lambda)\partial\lambda = \frac{\Delta L^2}{4\tau^3}\int_0^{\tau} \lambda^2 \partial\lambda \tag{1.41}$$

$$P_n = \frac{\Delta L^2}{12} \tag{1.42}$$

The quantity of most interest, however, is the *signal-to-noise ratio* (SNR) at the output of the A/D converter, which is defined as

$$SNR = \frac{P_s}{P_n} \tag{1.43}$$

where P_s and P_n are the power in the signal and noise, respectively. The signal power for an input sinusoidal signal is defined as

$$P_s = \frac{(A/2)^2}{2} = \Delta L^2\left(2^{2k-3}\right) \tag{1.44}$$

which leads to a SNR of

$$SNR = \frac{12\Delta L^2\left(2^{2k-1}\right)}{\Delta L^2} = \frac{3}{2}\left(2^{2k}\right) \tag{1.45}$$

or, in decibels,

$$SNR(dB) = 10\log_{10}(SNR) = 1.76 + 6.02k$$

Thus, the SNR at the output of the A/D converter increases by approximately 6 dB for each bit added to the word length, assuming the output signal is equal to the input signal to the A/D converter.

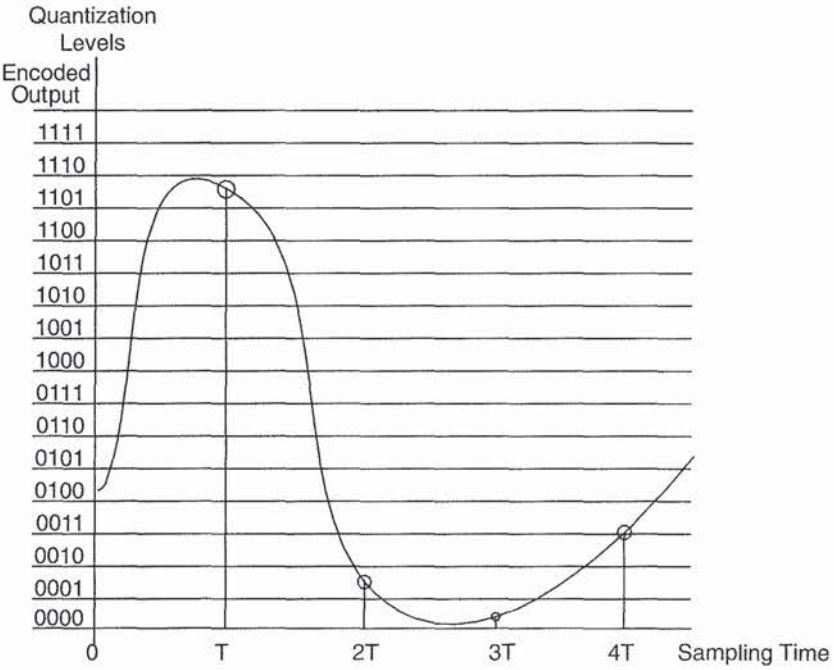


FIGURE 1.6
Quantization and digital encoding.

Source Coding

Source coding entails the efficient representation of information sources. For both discrete and continuous sources, the correlations between samples are exploited to produce an efficient representation of the information. The aim of source coding is either to improve the SNR for a given bit rate or to reduce the bit rate for a given SNR. An obvious benefit of the latter is a reduction in the necessary system resource of bandwidth and/or energy per bit to transmit the information source.

A discussion on source coding requires the definition of a quantity, which measures the average *self-information* in each discrete alphabet, termed *source entropy* [14]. The self-information $I(x_j)$ for discrete symbol or alphabet x_j is defined as

$$I(x_j) = -\log_2(p_j) \tag{1.46}$$

where p_j is the probability of occurrence for x_j. The source entropy $H(x_j)$ is found by taking the statistical expectation of the self-information, i.e.,

$$H(x_j) = E\{I(x_j)\} = -\sum_{j=1}^{M} p_j \log_2(p_j) \tag{1.47}$$

where M is the cardinality of the discrete alphabet set and the units of measure for $H(x_j)$ are bits/symbol. The source entropy can also be thought of as the average amount of uncertainty contained in the alphabet x_j. Therefore, the source entropy is the average amount of information that must be communicated across the channel per symbol. It can be easily shown that $H(x_j)$ is bounded by the following expression

$$0 \le H(x_j) \le \log_2 M \tag{1.48}$$

In other words, the source entropy is bounded below by zero if there is no uncertainty, and above by $\log_2 M$ if there is maximum uncertainty. As an example, consider a binary source x_j that generates independent symbols 0 and 1 with respective probabilities of p_0 and p_1. The source entropy is given by

$$H(x_j) = -[p_1 \log_2(p_1) + p_0 \log_2(p_0)] \tag{1.49}$$

Table 1.2 lists the source entropy as the probabilities of p_0 and p_1 are varied between 0 and 1.

TABLE 1.2 Source Entropy

p_0	p_1	$H(x_j)$
0.5	0.5	1.00
0.1	0.9	0.47
0.2	0.8	0.72

Table 1.2 illustrates that as the relative probabilities vary, the number of bits/symbol needed to transmit the information vary as well. One might wonder what information source would have such probabilistic distribution. English text, for instance, is known to have unequal probability for its alphabet; in other words, some letters appear in English text more frequently than others. Another key consideration when determining the source entropy is whether the source is *memoryless*. A discrete source is said to be memoryless if the sequence of symbols is statistically independent. The readers should refer to "Review of Stochastic Processes and Random Variables" for a definition of statistical independence. In contrast, if a discrete source is said to have memory, the sequence of symbols is not independent. Again, consider English text as an example. Given that the letter "q" has been transmitted, the next letter will probably be a u. Transmission of the letter "u" as the next symbol resolves very little uncertainty from a communication perspective. We can thus conclude that the entropy of an M-tuple from a source with memory is always less than the entropy of a source with the same alphabet and symbol probability without memory, i.e.,

$$H_M(x_j)_{with\ memory} < H_M(x_j)_{without\ memory} \tag{1.50}$$

Another way to interpret the relationship in Equation 1.50 is that the average entropy per symbol of an M-tuple from a source with memory decreases as the length M increases. Hence it is more efficient to encode symbols from a source with memory three at a time than it is to encode them two at a time or one at a time. Encoder complexity, memory constraints, and delay considerations require that practical source encoding be performed on finite-length sequences. Interested readers are encouraged to examine References [8, 13, 15, 25] dealing with source coding.

Channel Coding

Channel coding refers to the class of signal transformations designed to improve communications performance by enabling the transmitted signal to better withstand the effects of various channel impairments, such as noise, fading, and jamming. The goal of channel coding is to improve the bit error rate (BER) performance of power-limited and/or bandwidth limited channels by adding structured redundancy to the transmitted data. The two major categories of channel coding are block coding and convolutional coding. In the following sections, we describe the fundamental principles governing each of these two types of codes.

Linear Block Codes

Linear block codes are a class of parity check codes that can be characterized by an integer pair (n, k). As such, the parity bits or symbols are formed by a linear sum of the information bits. In general, the encoder transforms any block of k message symbols into a longer block of n symbols called a *codeword*. A special class of linear block codes called systematic codes append the parity symbols to the end of the k symbol message to form the coded sequence. These codes are of particular interest since they result in encoders of reduced complexity.

For binary inputs, the k-bit messages, referred to as *k-tuples,* form 2^k distinct message sequences. The n-bit blocks, referred to as n-tuples, form 2^k distinct codewords or message sequences out of 2^n possible n-tuples in the finite dimensional vector space.

Hence, one can view linear block codes as a finite dimensional vector subspace defined over the extended Galois fields $GF(2^p)$. The transformation from a k-dimensional to an n-dimensional vector space is performed according to a linear mapping specified in the generator matrix $\mathbf{G}$. In other words, an (n, k) linear block code $\mathbf{C}$ is formed by partitioning the information sequence into message blocks of length k. Each message block $\mathbf{u}_i$ is encoded or mapped to a unique codeword or vector $\mathbf{v}_i$ in accordance with $\mathbf{G}$

$$\mathbf{v}_i = \mathbf{u}_i\mathbf{G} \tag{1.51}$$

Another useful matrix, often defined when describing linear block codes, is the parity-check matrix $\mathbf{H}$. The parity check matrix has the property that it generates code words that lie in the null space of $\mathbf{G}$. Mathematically, we describe this relationship as

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}_{n-k} \tag{1.52}$$

where $\mathbf{0}_{n-k}$ is the zero vector of length $n-k$. As we will see in the following paragraphs, $\mathbf{H}$ plays an integral part in the detection process for linear block codes.

Consider a code vector $\mathbf{v}_i$ transmitted over a noisy channel. Denote the received vector from the channel as $\mathbf{r}_i$. The equation relating $\mathbf{r}_i$ to $\mathbf{v}_i$ is

$$\mathbf{r}_i = \mathbf{v}_i + \mathbf{e} \tag{1.53}$$

where $\mathbf{e}$ is an error pattern or vector from the noisy channel. At the receiver, the decoder tries to determine $\mathbf{v}_i$ given $\mathbf{r}_i$. This decision is accomplished using a two-step process: one, compute the *syndrome*; and two, add the *coset leader* corresponding to the syndrome to the received vector. We have just introduced two new terms that need defining. The syndrome $\mathbf{s}$ is defined as the projection of the received vector onto the subspace generated by $\mathbf{H}$

$$\mathbf{s}_i = \mathbf{r}_i\mathbf{H}^T \tag{1.54}$$

Using Equations 1.51–1.53, we see that Equation 1.54 simplifies to

$$\mathbf{s}_i = \mathbf{e}\mathbf{H}^T \tag{1.55}$$

The foregoing development is evidence that the syndrome test, whether performed on a corrupted codeword or on the error pattern that caused it, yields the same syndrome. An important property of linear block codes, fundamental in the decoding process, is the one-to-one mapping between correctable error patterns and the associated syndrome. This equivalence leads us to our discussion on coset leaders.

The coset leaders consist of all the correctable error patterns. A correctable error pattern is one whose *Hamming weight* is less than or equal to $\lfloor (d_{\min} - 1)/2 \rfloor$, where $\lfloor x \rfloor$ means the largest integer not to exceed x, and $d_{\min}$ is defined as the minimum *Hamming distance* between any two code words for an (n, k) linear block code. For binary vectors, the Hamming weight is defined as the number of non-zero elements in a vector, and the Hamming distance between two code vectors is defined as the number of elements in which they differ. It is interesting to note that $d_{\min}$ for a linear block code is equal to the non-zero codeword with minimum Hamming weight. These concepts are easily generalized to the extended Galois field.

Note that there are exactly 2^{n-k} coset leaders for the binary case. Now, let's form a matrix as follows: first, list all the coset leaders in the first column; second, list all the possible code words in the top row; and, last, form the (i, j) element of the matrix from the sum of ith element of the first column with the jth element of the top row. The resulting matrix represents all possible received vectors and is often referred to as the *standard array.*

Now, we will relate how all the above fits into the decoding process. As stated earlier, the task of the decoder is to estimate the transmitted code vector, $\mathbf{v}_i$ from the received vector $\mathbf{r}_i$. The maximum likelihood solution $\hat{\mathbf{v}}_{ML}$ found by maximizing the conditional probability density function for the received vector for all possible code vectors $\mathbf{v}_j$; namely,

$$\hat{\mathbf{v}}_{ML} = \arg \max_{\mathbf{v}_j} \Pr\left(\mathbf{r}_i \middle| \mathbf{v}_j\right) \tag{1.56}$$

The optimization criterion for Equation 1.56 over a binary symmetric channel (BSC) is to decide in favor of the code word that is the minimum Hamming distance from the received vector

$$d(\mathbf{r}_i, \mathbf{v}_i) = \min d\left(\mathbf{r}_i, \mathbf{v}_j\right) \quad \forall \mathbf{v}_j \tag{1.57}$$

A systematic procedure for the decoding process proceeds as follows:

1. Calculate the syndrome of $\mathbf{r}$ using $\mathbf{s} = \mathbf{r}\mathbf{H}^T$.
2. Locate the coset leader whose syndrome equals $\mathbf{r}\mathbf{H}^T$ from the standard array table.
3. This error pattern is added to the received vector to compute the estimate of $\mathbf{v}$ or read directly from the corresponding column in the standard array table.

Convolutional Codes

Another major category of channel coding is convolutional coding. An important characteristic of convolutional codes, different from block codes, is that the encoder has memory. That is, the n-tuple generated by the convolutional encoder is a function of not only the input k-tuple but also the previous $K-1$ input k-tuples. The integer K is a parameter known as the *constraint length*, which represents the number of memory elements in the encoder. A shorthand convention typically employed in the description of a convolutional encoder is (n, k, K) for the integers defined above.

To understand the fundamental principles governing the convolutional codes, we direct our attention to the (2,1,2) convolutional encoder depicted in Figure 1.7. This encoder, at any given sampling time instant, accepts k input bits, makes a transition from its current state to one of the 2^K possible successor states, and outputs n bits.

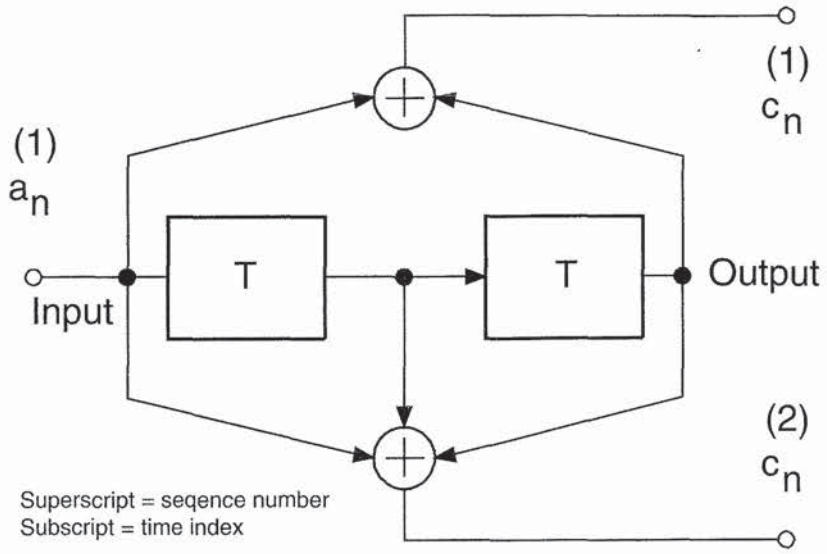


FIGURE 1.7
Four state convolutional encode.

The two noteworthy characteristics of a convolutional encoder are the tap connections and the contents of the memory elements. For the tap connections of the encoder, the generator polynomial representation is commonly used. Yet, for the state information as well as the output codewords, a state diagram or trellis diagram is typically employed.

With the generator polynomial representation, the taps for each output of the encoder are specified by a polynomial $g_i(D)$ where the coefficients of $g_i(D)$ are taken from $GF(2^p)$. Note, p equal to one corresponds to the binary field. For this case, a 1 coefficient denotes a connection and a 0

coefficient denotes no connection. The argument of the polynomial, D, denotes a unit time delay. In other words, we represent two unit time delays as D^2. The coded bits are formed by computing the sum over $GF(2^p)$ dictated by the tap coefficients. The two components of the sum are the current state of memory elements and the k input bits. To clarify these concepts, again consider the encoder shown in Figure 1.7. The generator polynomials for the outputs of this encoder are

$$g_1(D) = 1 + D^2 \tag{1.58}$$

$$g_2(D) = 1 + D + D^2 \tag{1.59}$$

We now briefly describe a method for viewing the state information as well as the output bits: the trellis diagram. We noted earlier that the convolutional encoder is a finite state machine. Thus, a natural way to describe its behavior is to view its state transition at each time instant. The state of the encoder is considered to be the contents of its memory elements. For a convolutional code with K memory elements, there are 2^K states. Each state is assigned a number from 0 to $2^K - 1$, which is obtained from the binary representation of its memory elements. The trellis diagram shows the state transitions and associated outputs for all possible inputs as seen in Figure 1.8. For binary inputs, solid and dashed lines are used, respectively, to represent a 0 or 1 input into the encoder. We also notice in the figure that there are two branches entering and leaving each state at every time instant. In general, there are 2^k branches emanating from each state and 2^k branches merging to each state. Each new sequence of k input bits causes a transition from an initial state to 2^k states at the next node in the trellis or time instant. Note, the encoder output is always uniquely determined by the initial state and the current input. Additionally, the trellis diagram shows the time evolution of the the states. Later, we will see that the trellis diagram is also very useful for evaluating the performance of these codes.

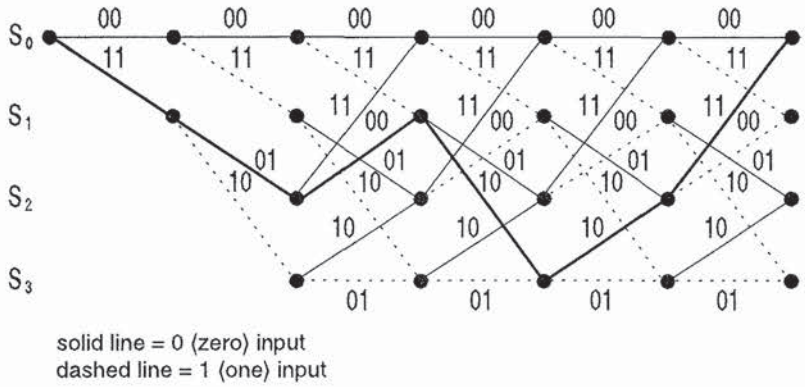


FIGURE 1.8
Trellis diagram for the four state convolutional encoder.

Note that after three stages into the trellis diagram, each node at subsequent stages has two branches entering and leaving it. Furthermore, it can be shown that the branches originate from a common node three stages back into the trellis. In general, for any constraint length (n, k, K) convolutional code, every K stages into the trellis diagram mark the point where 2^k branches merge and diverge at each node. Furthermore, the merging branches in a node can be traced back K stages into the past to the same originating node. This observation led to the development of a systematic approach for the optimum decoding of convolutional codes.

Decoder

As was the case for linear block codes, the decoder task is to estimate the transmitted code word from a received vector. It was shown that the optimum decoder strategy for linear block codes is to select the codeword with the minimum distance metric, the Hamming distance. The same is true for convolutional codes as well. Recall that an encoder for a convolutional code has memory. Thus, it is a reasonable approach to use all codewords associated with a particular symbol in the decision determination for it. Hence, for convolutional codes a sequence of codewords, or paths through the trellis, are compared to determine the transmitted codeword. In 1967, Viterbi [25] developed an algorithm that performs the maximum likelihood decoding with reduced computational load by taking advantage of the special structure of the code trellis.

The basis of *Viterbi decoding* is the following observation. If any two paths in the trellis merge to a single state, one of them can always be eliminated in the search for the optimum path. A summary of the Viterbi decoding algorithm proceeds as follows:

- Measure the similarity or distance between the received signals at each sampling instant t_i and all the paths entering each state or node at time t_i.
- The Viterbi algorithm eliminates from consideration the paths from the trellis whose distance metrics are not the minimum for a particular node. The distance metric can be either the Hamming distance or Euclidean distance. In other words, when two paths enter the same state, the one having the best distance metric is chosen. This path is called the *surviving path*. The selection of the surviving paths is performed for all the states.
- The decoder continues in this way, advancing deeper in the trellis and making decisions by elimination of least likely paths. In the process, the cumulative distance metric for each surviving path is recorded and used later to determine the maximum likelihood path.

The early rejection of the unlikely paths reduces the decoding complexity. In 1969, Omura [25] demonstrated that the Viterbi algorithm is, in fact, the maximum likelihood estimator. For a discussion of more advance of coding techniques, the reader is referred to Chapter 3, "Modulation and Coding," of this book.

Modulation

Modulation is the process by which information signals, analog or digital, are transformed into waveforms suitable for transmission across channel. Hence, digital modulation is the process by digital information are transform into digital waveforms. For baseband modulation, the waveforms are pulses; but for band-pass modulation, the information signals are transformed into radio frequency (RF) carriers, which have embedded the digital information. Since RF carriers are sinusoids, the three salient features are its amplitude, phase, and frequency. Therefore, digital band-pass modulation can be defined as the process whereby the amplitude, phase, or frequency of an RF carrier, or any combination of them, is varied in accordance with the digital information to be transmitted. The general form of a complex RF carrier is given by

$$s(t) = A(t)\exp(\omega_c t + \theta(t)) \tag{1.60}$$

where ω_c is the *radian frequency* of the carrier and $\theta(t)$ is the time-varying phase. The radian frequency of the RF carrier is related to its frequency in Hertz by

$$\omega_c = 2\pi f_c \tag{1.61}$$

At the receiver, the transmitted information embedded in the RF carrier must be recovered. When the receiver uses knowledge of the phase of the carrier to detect the signal, the process is called *coherent detection;* otherwise, the process is known as *non-coherent detection.* The advantage of non-coherent detection over coherent detection is reduced complexity at the price of increased probability of symbol error (P_E).

Whether the receiver uses coherent or non-coherent detection, it must decide which of the possible digital waveforms most closely resembles the received signal, taking into account the effects of the channel. A more rigorous treatment of common digital modulation formats and demodulation techniques is given in Chapter 3 of this book.

Multiple Access Techniques

Multiple access refers to the remote sharing of a fixed communication resource (CR), such as a wireless channel, by a group of users. For wireless communications, the CR can be thought of as a hyperplane in frequency and time. The goal of multiple access is to allow users to share the CR without creating unmanageable interferences with each other. In this section, we review the three most basic multiple access techniques for wireless communications: frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA). Other more sophisticated techniques are combinations or variants of these three.

FDMA

In FDMA systems, the frequency-time plane is partitioned into non-overlapping frequency bands. The top graph in Figure 1.10 illustrates the FDMA concept. Each RF carrier in the FDMA system is assigned a specific frequency band within the allocated bandwidth specified by the

Federal Communications Commission (FCC). The spectral regions between adjacent channels are called guard bands, which help reduce the interferences between channels. As a result of the non-linearities of the power amplifier (PA) at the transmitter, signals experience spectral spreading—broadening of the bandwidth of the signal. The situation worsen when multiple RF carriers are present simultaneously in the PA, as in the case with OFDM, and adjacent channel interference may result. Another source of frequency domain distortions is co-channel interference.

Co-channel interference results when frequency reuse is employed in wireless communication systems as depicted in Figure 1.9. A frequency band is said to be reused when it is shared by two or more downlinks* of the base station. The co-channel reuse ratio (CRR) is defined as the ratio the distance d between cells using the same frequency to the cell radius r. The capacity of the cellular network can be increased by splitting existing cells into smaller cells but maintain the same d/r ratio. CRR is chosen to provide adequate protection against co-channel interference from neighboring sites. However, Jakes [12] shows that the interference protection degenerates in the presence of Rayleigh fading.

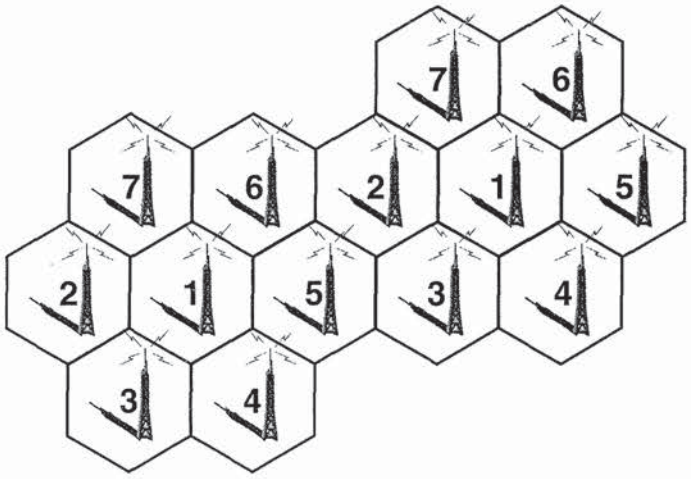


FIGURE 1.9
A macrocell layout using reuse factor of 7.

TDMA

In TDMA, sharing of the CR is accomplished by dividing the frequency-time plane into non-overlapping time slots which are transmitted in periodic bursts to the satellite. During the period of transmission, the entire allocated bandwidth is available to an user as illustrated in the middle graph of Figure 1.10. Time is segmented into intervals called frames. Each frame is further

*The downlink transmission is the communication link originating from the base station and terminating at the mobile.

partitioned into user assignable time slots. An integer number of time slots constitute a burst time or burst. Guard times are allotted between bursts to prevent overlapping of bursts. Each burst is comprised of a preamble and the message portion.

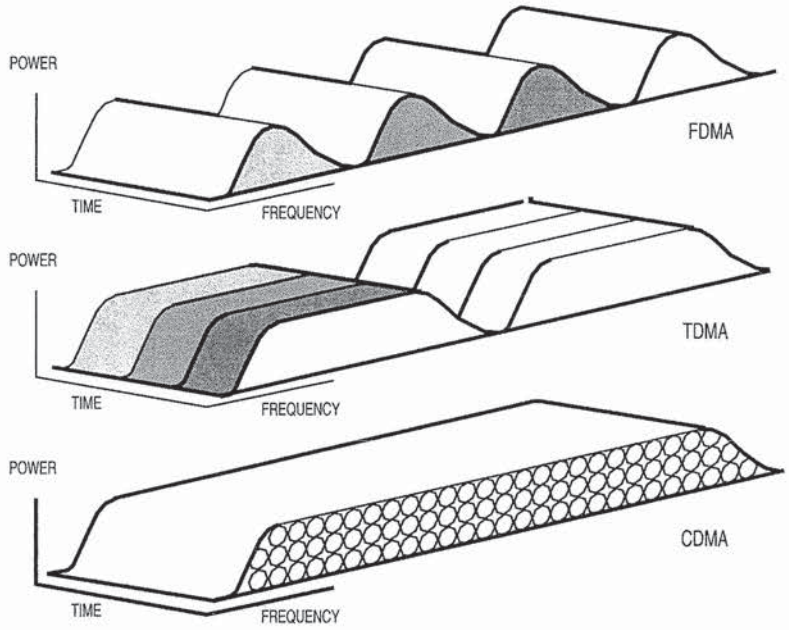


FIGURE 1.10
Communication resource hyperplane.

The preamble is the initial portion of a burst used for carrier and clock recovery and station-identification and other housekeeping tasks. The message portion of a burst contains the coded information sequence. In some systems, a training sequence is inserted in the middle of the coded information sequence. The advantage of this scheme is that it can aid the receiver in mitigating the effects of the channel and interferers. The disadvantage is that it lowers frame efficiency; that is, the ratio of the bit available for messages to the total frame length.

A point worth noting is that both FDMA and TDMA system performances degrade in the presence of the multipath fading. More specifically, due to the high transmission rates of the TDMA systems, the time dispersive channel (a consequence of delay spread phenomenon) causes inter-symbol interference (ISI). This is a serious problem in TDMA systems thus requiring adaptive techniques to maintain system performance.

CDMA

Unlike FDMA and TDMA who share only a portion of the frequency-time, CDMA systems share the entire frequency-time plane. Sharing of the CR is accomplished by assigning each user a unique quasi-orthogonal code as illustrated by the bottom graph of Figure 1.10.

Generally, CDMA is regarded as an application of direct sequence spread-spectrum techniques; a technique whereby the information signal is modulated by a high-speed (wideband) spreading signal and the resulting baseband signal is modulated onto an RF carrier. For digital communications, pseudonoise or pseudorandom (PN) signals are used as the spreading signals. Pseudonoise or pseudorandom signals are deterministic signals whose statistical properties are similar to sampled white noise. At the receiver in a CDMA system, the information signal is recovered by correlating the received spread signal with a synchronized replica of the spreading signal.

The ratio of the code chip rate and the data rate is called the processing gain GP. The processing gain represents the amount of interference protection provided by the code. Sklar [22] demonstrates that CDMA systems provide immunity to frequency selective fading and interferers by virtue of correlation property of the codes. Unfortunately, each additional user increases the overall noise level thus degrading the quality for all the users. In the next section, we describe the various impairments caused by the channel on the digital waveform.

Channel Model

In mobile wireless communications, the information signals are subjected to distortions caused by reflections and diffractions generated by the signals interacting with obstacles and terrain conditions as depicted in Figure 1.11. The distortions experienced by the communication signals include delay spread, attenuation in signal strength, and frequency broadening. Bello [1] shows that the unpredictable nature of the time variations in the channel may be described by narrowband random processes. For a large number of signal reflections impinging at the receiver, the central limit theorem can be invoked to model the distortions as complex-valued Gaussian random processes. The envelope of the received signals is comprised of two components: rapid-varying fluctuations superimposed onto slow-varying ones. When the mean envelope suffers a drastic reduction in signal strength resulting from "destructive" combining of the phase terms from the individual paths, the signal is said to be experiencing a *fade* in signal strength.

Multipath

In this section, we will develop a model to predict the effects of *multipath* on the transmitted communication signal. Multipath is a term used to describe the reception of multiple transmission paths to the receiver. As mentioned above, the channel can be accurately described by a random process; hence, the state of the channel will be characterized by its channel correlation function. The baseband transmit signal can be accurately modeled as narrowband process related to information bearing signal $x(t)$ by

$$s(t) = x(t)e^{-j2\pi f_c t} \tag{1.62}$$

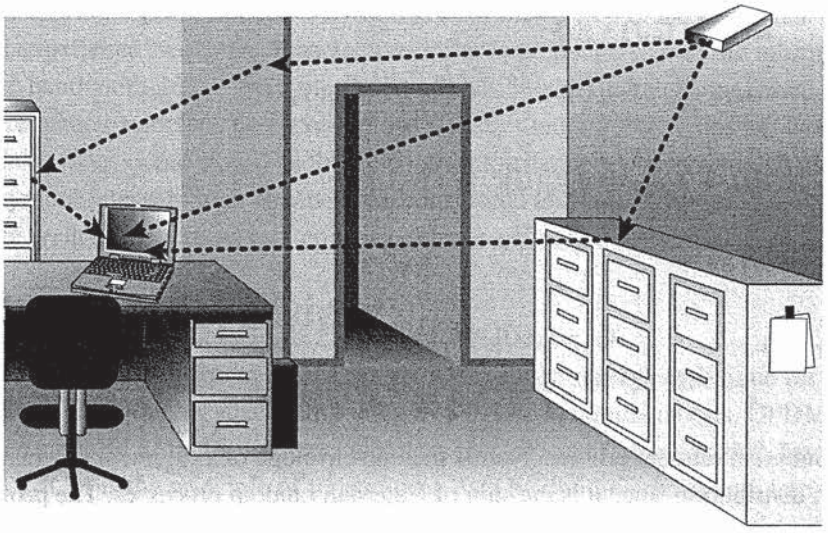


FIGURE 1.11
Multipath scattering in mobile communications.

Under the assumptions of Gaussian scatters and multiple propagation paths to the receiver, the channel is characterized by time-varying propagation delays, attenuation factors, and Doppler shifts. Proakis [21] shows that the time-variant impulse response is given by

$$c(\tau_n, t) = \sum_n \alpha_n(\tau_n(t)) e^{-j2\pi f_{D_n} \tau_n(t)} \delta[t - \tau_n(t)] \tag{1.63}$$

where

- $c(\tau_n, t)$ is the response of the channel at time t due to an impulse applied at time $t - \tau_n(t)$
- $\alpha_n(t)$ is the attenuation factor for the signal received on the nth path
- $\tau_n(t)$ is the propagation delay for the nth path
- f_{D_n} is the Doppler shift for the signal received on the nth path

The Doppler shift from relative motion between the vehicle and the receiver can be expressed as

$$f_{D_n} = \frac{v \cos(\phi_n)}{\lambda} \tag{1.64}$$

where v is the relative velocity of the vehicle, λ is the wavelength of the carrier, and ϕ_n is a random phase angle uniformly distributed from 0 to 2π.

Therefore, the output of the channel is simply the convolution of Equations 1.62 and 1.63, i.e.,

$$z(t) = c(\tau_n, t) * s(t) \tag{1.65a}$$

$$z(t) = \sum_n \alpha_n(\tau_n(t)) e^{-j2\pi(f_c + f_{D_n})\tau_n(t)} x[t - \tau_n(t)] e^{-j2\pi f_c t} \tag{1.65b}$$

$$z(t) = \sum_n \beta_n x[t - \tau_n(t)] e^{-j2\pi f_c t} \tag{1.65c}$$

where

$$x[t - \tau_n(t)] = \delta[t - \tau_n(t)] * x(t) \tag{1.66}$$

$$e^{-j2\pi(f_c + f_{D_n})\tau_n(t)} e^{-j2\pi f_c t} = \delta[t - \tau_n(t)] * e^{-j2\pi f_c t} e^{-j2\pi f_{D_n}\tau_n(t)} \tag{1.67}$$

and β_n is a Gaussian random process. Notice that the envelope of $c(\tau_n, t)$, at any instant t, exhibits a Rayleigh distribution since it is the sum of Gaussian random processes. The probability density function for a Rayleigh fading channel is given by

$$f_Z(Z) = \frac{Z}{\sigma^2} e^{-\frac{Z^2}{2\sigma^2}} \tag{1.68}$$

A channel with this distribution is typically termed a Rayleigh fading channel. In the event that there are fixed scatterers and a line of sight (LOS) path to the receiver, the envelope of $c(\tau_n, t)$ has a Rice distribution whose density is given by

$$f_Z(Z) = \frac{Z}{\sigma^2} I_0\left(\frac{Z\eta}{\sigma^2}\right) e^{-\frac{Z^2+\eta^2}{2\sigma^2}} \tag{1.69}$$

where I_0 is the modified Bessel function of the zeroth kind and η is the mean due to the fixed scatters or LOS path. In this case, the channel is said to be a Rician fading channel.

Proakis [21] shows that the autocorrelation function for $c(\tau, t)$,

$$\Lambda_c(\tau, \Delta t) = E\{c(\tau, t)c^*(\tau, t + \Delta t)\}$$

can be measured in practice by transmitting very narrow pulses and cross correlating the received signal with a conjugate delayed version of itself. Also, the average power output of the channel is found by setting $\Delta t = 0$; i.e., $\Lambda_c(\tau, 0) \equiv \Lambda_c(\tau)$. This quantity is called the *multipath intensity profile* or the *delay power spectrum* of the channel. The range of values of τ over which $\Lambda_c(\tau)$ is essentially nonzero is called the *multipath spread* of the channel, denoted by T_m. The reciprocal of the multipath spread is a measure of the *coherence bandwidth* of the channel, i.e.,

$$B_m \approx \frac{1}{T_m} \tag{1.70}$$

Information bearing signals whose bandwidth is small compared to the coherence bandwidth of the channel experience frequency nonselective or flat fading. However, if the information bearing signals have bandwidth greater than the coherence bandwidth of the channel, then the channel is said to be frequency-selective. Channels whose statistics remain fairly constant over several symbol intervals are considered slowly fading in contrast to channels whose statistics changes rapidly during an symbol interval. Such channels are considered fast fading. In general, indoor wireless channels are well-characterized by frequency-selective slowly fading channels.

OFDM WLAN Overview

Orthogonal frequency division multiplexing (OFDM) is a promising technique for achieving high data rate and combating multipath fading in wireless communications. OFDM can be thought of as a hybrid of multi-carrier modulation (MCM) and frequency shift keying (FSK) modulation. MCM is the principle of transmitting data by dividing the stream into several parallel bit streams and modulating each of these data streams onto individual carriers or subcarriers; FSK modulation is a technique whereby data is transmitted on one carrier from a set of orthogonal carriers in each symbol duration. Orthogonality amongst the carriers is achieved by separating the carriers by an integer multiples of the inverse of symbol duration of the parallel bit streams. With OFDM, all the orthogonal carriers are transmitted simultaneously. In other words, the entire allocated channel is occupied through the aggregated sum of the narrow orthogonal subbands. By transmitting several symbols in parallel, the symbol duration is increased proportionately, which reduces the effects of ISI caused by the dispersive Rayleigh-fading environment. Here we briefly focus on describing some of the fundamental principles of FSK modulation as they pertain to OFDM. The input sequence determines which of the carriers is transmitted during the signaling interval, that is,

$$\mathbf{s}_i(t) = A\exp(2\pi f_i t)\prod(t/T) \tag{1.71}$$

where

$$f_i = f_c + \frac{i}{T}, \qquad i = 0,1,\ldots,N-1 \tag{1.72}$$

$$\prod(t/T) = \begin{cases} 1, & for \quad -T/2 \le t \le T/2 \\ 0, & otherwise \end{cases} \tag{1.73}$$

N is the total number of subband carriers, and T is the symbol duration for the information sequence. In order that the carriers do not interfere with each other during detection, the spectral peak of each carrier must coincide with the zero crossing of all the other carriers as depicted in Figure 1.12. Thus, the difference between the center lobe and the first zero crossing represents the minimum required spacing and is equal to $1/T$. An OFDM signal is constructed by assigning

parallel bit streams to these subband carriers, normalizing the signal energy, and extending the bit duration, i.e.,

$$\widetilde{\mathbf{s}(n)} = \frac{A}{N}\sum_{i=0}^{N-1}\mathbf{x}_i(n)\exp(2\pi f_i n), \quad for \quad 0 \le n \le N;\ 0 \le i \le N \tag{1.74}$$

where $\mathbf{x}_i(n)$ is the nth bit of the ith data stream. Recall from "Discrete Fourier Transform (DFT)," Equation 1.74 is just the IDFT of $\mathbf{x}_i(n)$ scaled by A. The output sequence $\widetilde{\mathbf{s}(n)}$ is transmitted one symbol at a time across the channel. Prior to transmission, a cyclic prefix (CP) is prepended to the front of the sequence to yield $\mathbf{s}(n)$. A cyclic prefix is a copy of the last part of the OFDM symbol. This makes a portion of the transmitted signal periodic with period N, i.e.,

$$s(n-m) = s(N+n-m), \quad for \quad n-m \le p \tag{1.75}$$

where p is length of the CP. Hence, the received signal using vector notation is given by

$$\mathbf{r}(n) = \mathbf{s}(n) \circledast \mathbf{h}(n) + \mathbf{v}(n) \tag{1.76}$$

where $\circledast$ denotes linear convolution, $\mathbf{h}$ is the channel impulse response vector, and $\mathbf{v}$ is the additive noise vector. Now, if length of CP is longer than the delay spread of the channel, the linear convolution in Equation 1.76 becomes a circular one. Note the DFT transform pair for the convolution theory given Table 1.1 is based on circular convolution. Hence, the demodulated received signal is

$$\mathbf{R}(k) = \mathbf{S}(k)\mathbf{H}(k) + \mathbf{N}(k) \tag{1.77}$$

where $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{H}$ are the respective DFTs of $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{h}$. Henceforth, for notation simplicity, when Equation 1.76 is satisfied, the channel shall be referred to as *circulant.*

MAC for WLAN Standards

Currently, there are three approved world WLAN standards that utilize OFDM for their physical layer specifications, which are listed in Table 1.3, where HiperLAN/2 stands for High Performance Local Area Network type 2 and MMAC stands for Mobile Multimedia Access Communications. Each standard offers data rates ranging from 6 Mbps to 54 Mbps in the 5 GHz band. The major difference between the standards is the medium access control (MAC) used by each. IEEE 802.11a uses a distributed MAC based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA); HiperLAN/2 uses a centralized and scheduled MAC based on wireless asynchronous transfer mode (ATM); and MMAC supports both of these MACs. In the remaining portion of this section, brief summaries of these MAC protocols are given. Although we would like to provide a thorough treatment of each of these MACs, we focus our attention on the most widely used MAC, the IEEE 802.11 MAC. Further treatment of this protocol is given in Chapter 6, "Medium Access Control (MAC) for IEEE 802.11 Networks," for interested readers.

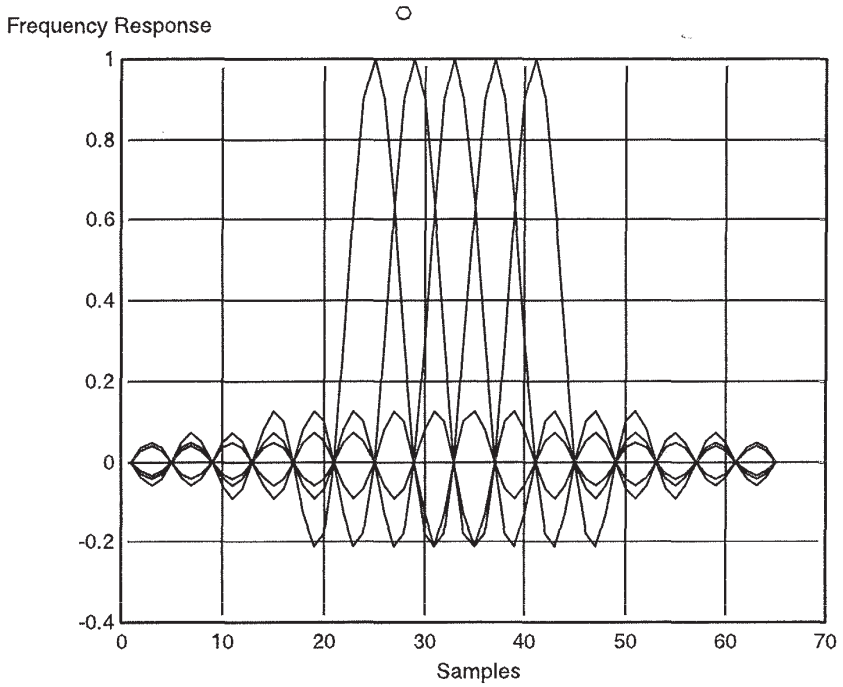


FIGURE 1.12
Overlapping orthogonal carriers.

TABLE 1.3 World Standards for OFDM WLANs

Standard	*Region of Operation*
IEEE 802.11a	Europe and North America
HiperLAN/2	Europe and North America
MMAC	Japan

The MAC frame structure for HiperLAN/2 shown in Figure 1.13 comprises time slots for broadcast control (BCH), frame control (FCH), access feedback control (ACH), and data transmission in downlink (DL), uplink (UL), and direct link (DiL) phases. These control frames are allocated dynamically depending on the need for transmission resources. To access the network, a mobile terminal (MT) first has to request capacity from the access point (AP) in order to send data. Access is granted via the random access channel (RCH), where contention for the same time slot is allowed. Downlink, uplink, and direct link phases consists of two types of protocol data units (PDUs): long PDUs and short PDUs. The long PDUs shown in Figure 1.14 have a size of 54 bytes and contain control or user data. They may contain resource requests, automatic repeat

request (ARQ) messages, and so on, and they are referred to as the short transport channel (SCH). The payload is 49.5 bytes and the remaining 4.5 bytes are used for the 2 bits PDU type, 10 bits sequence number (SN), 24-bit cyclic redundancy check (CRC-24). Long PDUs are referred to as the long transport channel (LCH). Short PDUs contain only control data and have a size of 9 bytes. Traffic from multiple connections to/from one MT can be multiplexed onto one PDU train, which contains long and short PDUs. A physical burst shown in Figure 1.15 is composed of the PDU train payload and a preamble and is the unit to be transmitted via the physical layer.

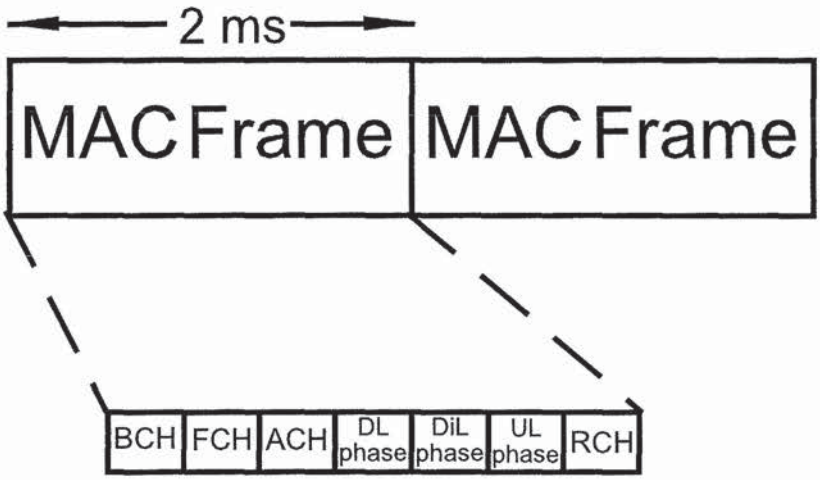


FIGURE 1.13
The HiperLAN/2 MAC frame.

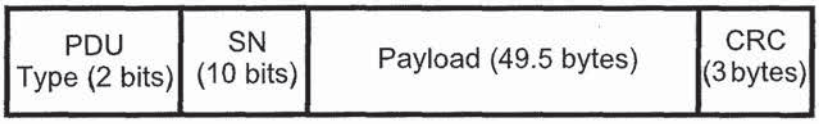


FIGURE 1.14
Format of the long PDU.



FIGURE 1.15
HiperLAN/2 physical burst format.

The MAC frame structure for IEEE 802.11a is shown in Figure 1.16. As stated earlier, the IEEE 802.11a uses a CSMA/CA protocol. An MT must sense the medium for specific time interval and ascertain whether the medium is available. This process is referred as *clear channel assessment* (CCA). Algorithms for performing CCA are described in Chapter 2, "Synchronization." If the channel is unavailable, the transmission is suspended and a random delay for re-assessing the channel is assigned. After the delay expires, the MT can access the channel again. Once a packet has been transmitted, the MT waits for an acknowledgement (ACK) frame. Unlike wireline communications, collisions are undetectable in a wireless environment since corruptions of the data can be caused by either collisions or fading. Thus, an ACK frame is necessary. If an ACK frame is not received, the MT retransmit the packet. Figure 1.16 shows the format of a complete packet. The header contains information about the length of the payload and the transmission rate, a parity bit, and six zero tail bits. The header is transmitted using binary phased shifted keying (BPSK), the lowest rate transmission mode, to ensure reliable reception. The rate field conveys information about the type of modulation and coding rate used in the rest of the packet. The length field takes a value between 1 and 4095 and specifies the number of bytes in the Physical Layer Service Data Unit (PSDU). The six tail bits are used to flush the convolutional encoder and terminate the code trellis in the decoder. The first 7 bits of the service field are set to zero and are used to initialize the descrambler. The remaining nine bits are reserved for future use.

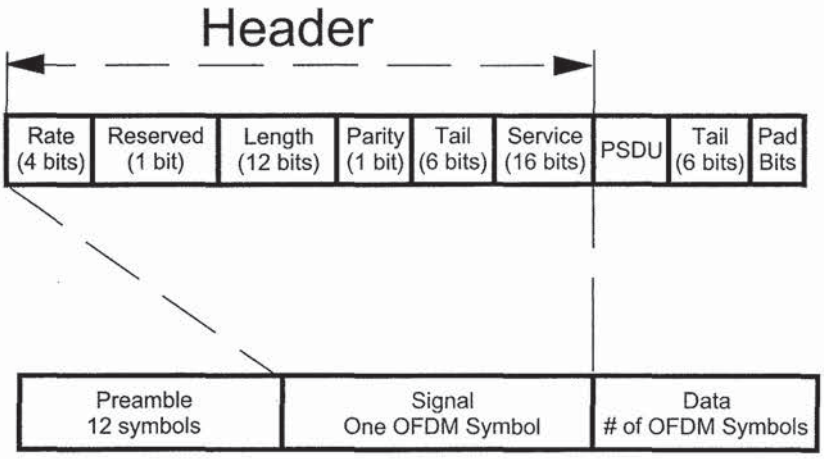


FIGURE 1.16
(PPDU) in 802.11a, including the preamble, header, and PSDU.

At the time this book was published, members of IEEE 802.11 and ETSI/BRAN standard committees were working to form a global 5 GHz WLAN standard. The task group within IEEE responsible for the harmonization process is 5GSG. The issues being addressed by the committee are co-existence, interworking, a single global standard, and regulatory issues. Co-existence issues are focused on frequency sharing and interference robustness. Interworking pertains to the ability of the two standards to communicate to each other. The new jointly developed global standard is being developed based on scenarios and application functional requirements. Finally, regulatory issues are centered around HiperLAN/2 need to restrict the frequency range and power to be compliant with the FCC regulations; for IEEE 802.11, what additional adoptions are needed for it to be operable outside of the USA.

Physical Layer Specifications for WLAN Standards

The physical layers of all the standards are very similar and are based on an OFDM baseband modulation. A list of the key parameters for the system is listed in Table 1.4. With each of the standards, OFDM was selected to combat frequency selective fading and to randomize the burst errors caused by a wideband fading channel. Selection of the transmission rate is determined by a *link adaptation scheme*, a process of selecting the best coding rate and modulation scheme based on channel conditions. The WLAN standards, however, do not explicitly specify the scheme. Data for transmission is supplied to the physical layer via a PDU train. The PDU train contains a sequence of 1s and 0s. Preparation for transmission and data recovery are performed by the functional blocks shown in Figure 1.17. Mux is a serial to parallel operation; demux (demultiplex) is a parallel to serial operation. It should be noted a length 127 pseudo random sequence is used to scramble the data out of the binary source prior to the convolutional encoder although it is not explicitly shown in the Figure. The purpose of scrambler is to prevent a long sequence of 1s or 0s. This helps with the timing recovery at the receiver. Besides that, the remaining functions in the transceiver are unaffected by the scrambling operation.

TABLE 1.4 Key Parameters of the OFDM Standards

Data Rate	*6, 9, 12, 18, 24, 36, 48, 54 Mbps*
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Coding Rates	1/2, 9/16, 2/3, 3/4
Number of Subcarriers	52
Number of Pilot Tones	4
OFDM Symbol Duration	4 μ sec
Guard interval	800 η sec, 400 η sec (optional)
Subcarrier Spacing	312.5 kHz
Signal Bandwidth	16.66 MHz
Channel Spacing	20 MHz

HiperLAN/2 and IEEE 802.11a standards specify different initialization for the scrambler. The scrambled data sequence is encoded with a de facto standard (2,1,7) convolutional encoder. The other rates shown in Table 1.4 are achieved by puncturing the output of this encoder. Puncturing involves deleting coded bits from output data sequence such that the ratio of uncoded bits to coded bits is greater than the mother code. For example, to achieve a 2/3 code rate one bit out of every four bits is deleted from the coded sequence. Next, the coded bits are interleaved to prevent error bursts from being fed into the convolutional decoder since the decoder does not work very well with burst errors. The interleaved coded bits are grouped together to form symbols. The symbols are modulated with one of the scheme given in Table 1.4. A BPSK symbol is comprised of one bit per symbol; a QPSK symbol is comprised of two bits per symbol; a 16-QAM symbol is comprised of four bits per symbol; and a 64-QAM symbol is comprised of 6 bits per symbol. The interested reader is referred to Chapter 3 for more details on modulation scheme mentioned here.

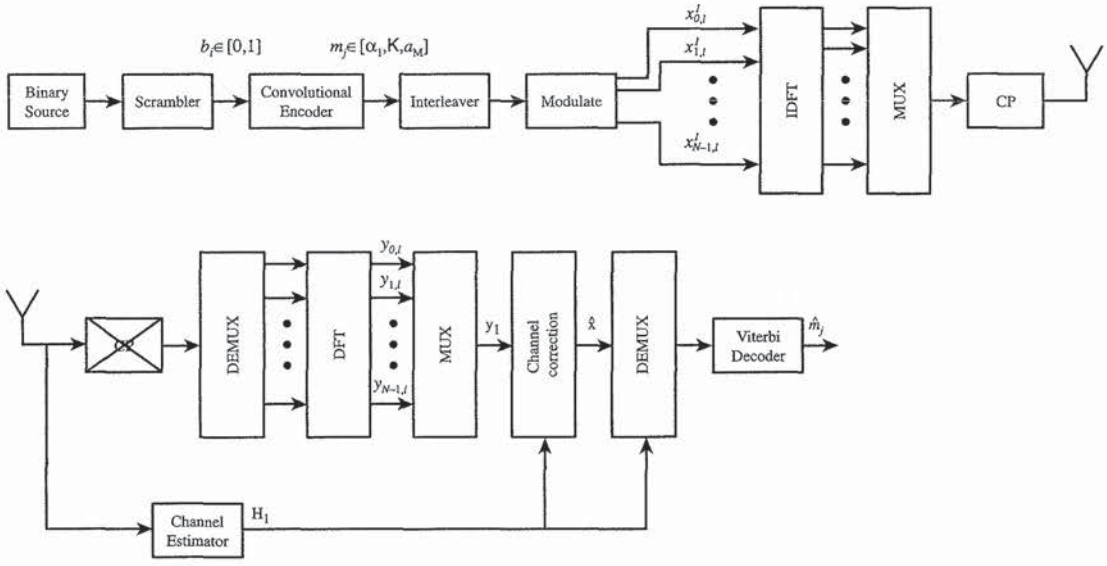


FIGURE 1.17

A simplified block diagram of the IEEE 802.11a transceiver.

The modulation symbols are mapped to the subcarrier of the 64 point IDFT, hence creating an OFDM symbol. Note that because of bandwidth limitations, only 48 subcarriers are used for modulation and four subcarriers are reserved for pilot tones.The remaining 12 subcarriers are not used. The pilot tones are used at the receiver to estimate any residual phase error. The output of the IDFT is converted to a serial sequence and a guard interval or CP is added. Thus, total duration of the OFDM symbol is the sum of the CP or guard duration plus the useful symbol duration. The guard or CP is considered overhead in the OFDM frame along with the preamble. After the CP has been added, the entire OFDM symbol is transmitted across the channel. As long as the

duration of the CP is longer than the channel impulse response, ISI is eliminated, which is evident from Equation 1.77 in the previous section.

After the CP has been removed, the receiver is responsible for performing the inverse operations of the transmitter in reverse order. Before any receiver algorithms can be employed, timing must first be recovered; that is, the system clock at the receiver must become synchronized with that at the transmitter while taking into account the delay associated with propagation across the channel. Chapter 2 provides a full treatment of synchronization algorithms used to recovery timing in OFDM systems.

In addition to timing recovery, the receiver must also compute *automatic gain control* (AGC) for the A/D converter. The purpose of the AGC is to maintain a fixed signal power to the A/D converter to prevent signals from saturating or clipping the output of the A/D converter. Since OFDM is a frequency domain modulation technique, it is essential to have accurate estimates of the frequency offset, caused by oscillator instability, at the receiver. It is obvious from Equation 1.77 that channel estimation is needed as well to demodulate the symbols. Training sequences are provided in the preamble for these specific functions mentioned above. To reduced the uncertainty in the channel estimation, two OFDM symbols containing training sequences are provided: short training and long training. The short training is used to provide coarse and fine estimation of time and frequency errors. The long training sequence is used to estimate the channel impulse response or *channel state information* (CSI). With the CSI, the received signal can be demodulated, de-interleaved, and fed to a Viterbi algorithm for decoding. As far as the physical layer specifications are concerned, there are only a few minor differences between the standards. These differences are

- HiperLAN/2 uses extra puncturing to accommodate the tail bits in order to keep an integer number of OFDM symbols in 54 byte packets required for ATM transmission. For example, the 16-QAM mode of HiperLAN/2 uses a rate 9/16 convolutional encoder rather than the rate 1/2 convolutional encoder used by IEEE 802.11a and MMAC. The rate 9/16 is generated by puncturing 2 out of every 18 coded bits.
- HiperLAN/2 uses different training sequences. The long training is the same as IEEE 802.11, but the preceding sequence of short training symbols is different. A downlink transmission starts with 10 short symbols as IEEE 802.11, but the first 5 symbols are different in order to detect the start of the downlink frame. Uplink packets may use 5 or 10 identical short symbols, with the last short symbol inverted.
- HiperLAN/2 has an optional mode that uses a 400 η sec cyclic prefix.

Single Carrier Versus OFDM Comparison

We conclude this chapter with a comparison of single carrier systems versus OFDM systems with the same data rate. The main difference between the systems are their robustness to fading

and synchronization errors—both frequency and timing. Under the assumption of perfect synchronization, the performance of a single carrier system and an OFDM system is equivalent for AWGN and flat or frequency non-selective channels. Consider a received signal for a single carrier system

$$r_i = \gamma s_i + n_i, \quad 0 \le i \le L-1 \tag{1.78}$$

where γ is a complex random variable, s_i is the baseband representation of the ith modulation symbol, and n_i is complex additive white Gaussian noise sample in the ith signal interval. The equivalent OFDM received signal for a circulant channel is

$$R_k = \gamma X_k + N_k, \quad 0 \le k \le L-1 \tag{1.79}$$

where R_k, X_k, and N_k are the frequency domain representations of r_i, x_i, and n_i respectively. Since the noise power of n_i and N_k are equivalent by Parseval's Theorem [7, 10, 18], there is no inherent advantage in detecting the signal using either Equation 1.78 or Equation 1.79. Now, consider reception of a signal over a circulant frequency-selective channel. For the single carrier system, the received signal becomes

$$\mathbf{r}(i) = \Gamma(i) \circledast \mathbf{s}(i) + \mathbf{n}(i) \tag{1.80}$$

where

$$\begin{aligned}
\mathbf{r}(i) &= \left[r_i, r_{i+1}, \cdots, r_{L+i-1}\right]^{\dagger} \\
\Gamma(i) &= \left[\gamma_i, \gamma_{i+1}, \cdots, \gamma_{p+i-1}\right]^{\dagger} \\
\mathbf{s}(i) &= \left[s_i, s_{i+1}, \cdots, s_{L+i-1}\right]^{\dagger} \\
\mathbf{n}(i) &= \left[n_i, n_{i+1}, \cdots, n_{L+i-1}\right]^{\dagger}
\end{aligned}$$

and † denotes vector transpose. To recover s_i, the single carrier system requires an equalizer $\mathbf{g}(i)$ to compensate for the channel effect. The equalizer performs deconvolution on the received signal, i.e.,

$$\begin{aligned}
\mathbf{r}_{eq}(i) &= \mathbf{g}(i) \circledast \mathbf{r}(i) = \mathbf{g}(i) \circledast \left(\Gamma(i) \circledast \mathbf{s}(i) + \mathbf{n}(i)\right) \\
\mathbf{r}_{eq}(i) &= \mathbf{s}(i) + \Gamma(i) \circledast \mathbf{n}(i)
\end{aligned} \tag{1.81}$$

In practical systems, the equalizer does not perfectly inverse the effects of the channel, there is always some residual error $\varepsilon(i)$,

$$\mathbf{r}_{eq}(i) = \mathbf{s}(i) + \Gamma(i) \circledast \mathbf{n}(i) + \varepsilon(i) \tag{1.82}$$

In addition to the residual error term $\varepsilon(i)$, the deconvolution process correlates the noise samples and enhances the noise amplitude in some samples. Further discussion of equalization shall be limited since equalization is well researched in the literature. An excellent text [11] summarizes some of the popular equalization techniques used in systems. In comparison, the OFDM system performs equalization in the same manner as the case of the frequency non-selective channel, i.e.,

$$\mathbf{R}(k) = \Gamma(k)\mathbf{S}(k) + \mathbf{N}(k) \tag{1.83}$$

$$\mathbf{R}_{eq}(k) = \mathbf{S}(k) + \Psi(k)\mathbf{N}(k) \tag{1.84}$$

where

$$\Psi(k) = \left[1/\gamma_i, 1/\gamma_{i+1}, \cdots, 1/\gamma_{p+i-1}\right]^{\dagger}$$

Equalization in OFDM systems is subject to the same impairments such as residual error and noise enhancement as the single carrier system; thus, theoretically, the two systems have equivalent performance. Yet, the complexity of the equalizer for the OFDM system is substantially less than that for the single carrier system. The reason is that OFDM systems employ a bank of single-tap equalizers while single carrier systems employ multi-tap equalizers. Further, the complexity of the equalizer grows as the square of the number of taps.

Synchronization Errors

Synchronization errors can be either timing, frequency, or both. The single carrier system is much more sensitive to timing errors than the OFDM system. On the other hand, the OFDM system is more sensitive to frequency errors. In the paragraphs to follow, the synchronization error performance for both systems will be examined.

Timing Errors

Even with the training interval, the demodulator reference circuitry may not be able to recover completely the timing at the transmitter. When the receiver is not symbol synchronized to the incoming data stream, the SNR at the output of the detection filter is degraded. In other words, for a particular sampling time $T_{optimal}$, the output SNR is given by [22]

$$\zeta = \Lambda(\tau)/\Lambda(0) \tag{1.85}$$

where Λ is the autocorrelation function and τ is the delay between the optimum sampling instance $T_{optimal}$ and the associated symbol timing for the received signal. Note that the parameter τ is random variable since it is estimated in the presence of noise and the variability of τ is usually referred to as *timing jitter*. The two special cases of interest, baseband time-limited signals and band-limited signals with normalized autocorrelation functions [26], are shown in Equations 1.86 and 1.87, respectively,

$$\Lambda(\tau) = \left(1 - \frac{|\tau|}{T_{Symbol}}\right) \tag{1.86}$$

$$\Lambda(\tau) = \frac{1}{N}\left(\frac{\sin(\pi NW\tau)}{\sin(\pi W\tau)}\right), \tag{1.87}$$

where W is the bandwidth of the band-limited signal. For the purpose of our discussion, the single carrier system is best described as a band-limited signal while the OFDM system is best described as a time-limited signal. For single carrier systems, the timing error or jitter manifests as a noisy phase reference for the bandpass signal. In the case of OFDM systems, pilot tones are transmitted along with the data bearing carrier to estimate residual phase errors. Chapter 2 outlines algorithms for estimating residual phase and compensation for the distortion. The single carrier system does not have a mechanism to achieve this compensation. The degradation resulting from phase error is developed later in this section.

Effects of Frequency Errors

When there is relative motion between the transmitter and receiver, a Doppler shift of the RF carrier results and introduces a frequency error. Also, there can be a residual frequency error caused by frequency instabilities in the oscillators at the transmitter and receiver. In either case, there are well-known carrier recovery scheme available for single carrier system such as a first order Costa loop [5]. The interested reader is encouraged to review [2], for more details. The important result is that although the carrier is recovered, the phase may be unknown and random.

Effects of Noisy Phase References

The effects of noisy phase references on the performance of the demodulator is developed in this section. We limit our treatment to binary phase shift keying (BPSK), quaternary phase shift keying (QPSK) modulation, and offset-QPSK. Consider first the baseband representation of a BPSK signal received in additive Gaussian noise

$$r(t) = \sqrt{2}\sum_{n} a_n h(t - nT)\cos\omega_c t + v(t) \tag{1.88}$$

where the signal amplitude has been normalized for unity power, $h(t)$ is the pulse shaping filter used at the transmitter, a_n refers to the binary digit transmitted in the nth interval, and $v(t)$ is the additive white Gaussian noise component. In general, $h(t)$ can be any band-limited shaping filter; however, for the sake of simplicity, we consider a finite duration unity amplitude rectangular pulse defined over the interval $(0, T]$. The receiver provides a reference carrier

$$R(t) = \sqrt{2}\cos(\omega_c t + \theta) \tag{1.89}$$

where θ is the phase error. Multiplying Equation 1.88 by Equation 1.89 and ignoring double-frequency terms yields

$$y(t) = r(t)R(t) = \sum_n a_n h(t - nT)\cos\theta + v(t)R(t) \tag{1.90}$$

For the case of rectangular pulse shaping, the optimum matched filter is an integrate and dump, and its output corresponding to the nth bit is

$$y_n(T) = \frac{1}{T}\int_0^T y(t)\partial t = a_n \cos\theta + \frac{1}{T}\int_0^T v(t)R(t)\partial t \tag{1.91}$$

The noise component of Equation 1.91 can be shown to be Gaussian with zero mean and variance $\sigma^2 = N_0 / 2$. The probability of error $P_b(\theta)$ conditioned on θ is easily derived and equal to

$$P_b(\theta) = Q\left[\sqrt{\frac{2E_b}{N_0}}\cos\theta\right] \tag{1.92}$$

where Q is complementary error function. The average error rate is obtained by integrating $P_b(\theta)$, i.e.,

$$P_{BPSK} = \int_{-\pi}^{\pi} P_b(\theta) f(\theta)\partial\theta \tag{1.93}$$

where $f(\theta)$ is the pdf of θ. For the case of timing jitter, the phase error is modelled as a uniform distribution between $[-\omega_c\Delta t_1, -\omega_c\Delta t_2]$, where Δt_1 and Δt_2 are the minimum and maximum timing errors, respectively. The average error rate must be evaluated numerical. For first order Costa loops, Viterbi showed in [26] that the pdf with an unmodulated carrier input is given by

$$f(\theta) = \frac{\exp(\alpha\cos\theta)}{2\pi I_0(\alpha)} \tag{1.94}$$

where α is the phase reference signal-to-noise ratio (SNR) and $I_0(\alpha)$ refers to the modified Bessel function of order zero and argument α. For large SNR, Equation 1.94 is well approximated by

$$f(\theta) \approx \frac{\exp[\alpha(\cos\theta - 1)]}{(2\pi/\alpha)^{1/2}} \tag{1.95}$$

Next, we consider a noisy phased corrupted QPSK signal. Recall that for QPSK, data is transmitted on quadrature channels; thus, the receiver must provide two local references

$$R_I(t) = \sqrt{2}\cos(\omega_c t + \theta)$$
$$R_Q(t) = \sqrt{2}\sin(\omega_c t + \theta) \tag{1.96}$$

for the receiver filters. For any nonzero phase error θ, each channel interferes with the other. An analysis similar to that for BPSK shows that the conditional error for QPSK is given by

$$P_q(\theta) = 1/2\left\{Q\left[\sqrt{\frac{2E_b}{N_0}}(\cos\theta + \sin\theta)\right] + Q\left[\sqrt{\frac{2E_b}{N_0}}(\cos\theta + \sin\theta)\right]\right\} \tag{1.97}$$

With offset QPSK one channel is offset with respect to the other channel by T, where T is the bit duration. As a result, a transition in one channel occurs at the midpoint of the signaling interval of the other channel. So, if a data transition occurs, the mutual interference from the first half interval exactly cancels the interference from the second half interval. If no transition occurs, the interference remains constant and the conditional is the same as that for QPSK. Thus, the conditional error rate for offset QPSK is

$$P_{O_q}(\theta) = 1/2[P_b(\theta) + P_q(\theta)] \tag{1.98}$$

For OFDM systems, the frequency offset error must be estimated with training sequence. Unfortunately, a residual frequency error might exist even after frequency correction has been applied. In the following development, we qualify the effects of frequency offset error in OFDM systems. Since the sampled output of the (I,Q) match filters are sufficient statistics for the receiver, it is instructive to examine these in the presence of frequency errors. Again assuming no timing error, the noiseless output values of the I channel and Q channel match filters are

$$y_n^I = \sqrt{E_b}\,h(\cos[2\pi\Delta f t + \theta])\big|_{t=nT}$$
$$y_n^Q = \sqrt{E_b}\,h(\sin[2\pi\Delta f t + \theta])\big|_{t=nT} \tag{1.99}$$

For time-limited signals or rectangular pulse shaping, $h(t)$ is an integration over one symbol period. Hence, the expressions in Equation 1.88 expand to

$$y_n^I = \frac{\sqrt{E_b}}{T}\int_0^T \cos[2\pi\Delta f t + \theta]\partial t$$
$$y_n^Q = \frac{\sqrt{E_b}}{T}\int_0^T \sin[2\pi\Delta f t + \theta]\partial t \tag{1.100}$$

which have closed form solutions

$$y_n^I = \sqrt{E_b}\left(\frac{\sin(2\pi\Delta fT)\cos\theta}{2\pi\Delta fT} + \frac{[-1+\cos(2\pi\Delta fT)]\sin\theta}{2\pi\Delta fT}\right)$$
$$y_n^Q = \sqrt{E_b}\left(\frac{[1-\cos(2\pi\Delta fT)]\cos\theta}{2\pi\Delta fT} + \frac{\sin(2\pi\Delta fT)\sin\theta}{2\pi\Delta fT}\right) \quad (1.101)$$

Thus, the signal power at the output of the matched filters is

$$Z = \left(y_n^I\right)^2 + \left(y_n^Q\right)^2 \quad (1.102)$$

which simplifies to

$$Z = E_b\left(\frac{\sin(2\pi\Delta fT)}{2\pi\Delta fT}\right)^2 + E_b\left(\frac{[1-\cos(2\pi\Delta fT)]}{2\pi\Delta fT}\right)^2$$
$$Z = E_b\frac{2-2\cos(2\pi\Delta fT)}{(2\pi\Delta fT)^2} \quad (1.103)$$
$$Z = E_b\left(\frac{\sin(2\pi\Delta fT)}{\pi\Delta fT}\right)^2$$

Hence, the degradation of the signal due to a frequency error of Δf hertz is

$$L(\Delta f) = \left(\frac{\sin(\pi\Delta fT)}{\pi\Delta fT}\right)^2 \quad (1.104)$$

Equation 1.104 can also be used to compute the intercarrier interference (ICI) from adjacent carriers. Thus, frequency offset errors not only result in a loss in SNR, but also create ICI, which can severely degrade the performance of the system.

To summarize, single carrier systems are fairly robust to frequency offset errors and are more appropriate for mobile environments that experience large frequency offset errors. Yet, the performance of single carrier systems is degraded as a result of SNR loss caused by timing errors. In constrast, OFDM systems are fairly robust to timing errors compared to single carrier systems, yet their performance is similarly affected by the loss in SNR caused by frequency offset errors. Intuitively, this is easily understood from the fact that the OFDM symbol duration is N times longer than its single carrier counterpart operating at the same data rate. In terms of fading, the performances of the two systems are similar. The complexity of the equalizer for single carrier systems, however, is much greater than OFDM systems.

Bibliography

[1] P. A. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. on Communication Systems,* vol. CS-11, Dec. 1963.

[2] V. K. Bhargava, D. Haccoun, R. Matyas, and P. P. Nuspl, *Digital Communications by Satellite: Modulation, Multiple Access, and Coding*. Krieger Publishing Company, 1991.

[3] R.W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission, *Bell System Tech. Journal,* 45:1775–1796, Dec. 1966

[4] E. P. Cunningham, *Digital Filtering: An Introduction*, Houghton Mifflin Company, 1992.

[5] A. N. D'Andrea and U. Mengali, *Synchronization Techniques for Digital Receivers,* Plenum Press, New York, 1997.

[6] O. Edfors, M. Sandell, J. van de Beek, D. Landström, and F. Sjöberg, *An introduction to orthogonal frequency-division multiplexing,* Technical Report, September 1996, Division of Signal Processing, Luleå University.

[7] D. R. Fannin, W. H. Tranter, and R. E. Ziemer, *Signal and Systems: Continuous and Discrete,* MacMillian Publishing Co., New York, 1983.

[8] J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet, "Speech coding," *IEEE Trans. on Commun.,* COM-27, 1979, pp. 710–737.

[9] S. Hara and J.P. Linnartz, *Wireless Personal Communication,* No.1-2, Kluwer, 1996.}

[10] M. H. Hayes, *Statistical Digital Signal Processing and Modeling,* John Wiley & Sons, Inc, New York, 1996.

[11] S. S. Haykin, *Adaptive Filtering Theory*: Fourth Editions, Prentice Hall, Inc., Englewood Cliffs, N.J., 1995.

[12] W. C. Jakes, *Microwave Communications,* IEEE Press, N.J. 1974.

[13] N. S., Jayant, and P., Noll, *Digital Coding of Waveforms,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1984.

[14] Cover and Thomas, *Elements of Information Theory,* JohnWiley & Sons, Inc, New York, 1991.

[15] T. J. Lynch, *Data Compression Techniques and Applications,* Lifetime Learning Publications, New York, 1985.

[16] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky and W. Papke, "The land mobile satellite communication channel - recording, statistics, and channel model," *IEEE Trans. Veh. Tech.*, vol. 40, no. 2, May 1991.

[17] B. Noble and J. Daniel, *Applied Linear Algebra Third Edition*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[18] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentic Hall, Englewood Cliffs, New Jersey, 1989.

[19] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms, *In Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 964–967, 1980.

[20] A. Papoulis, *Probability, Random Variables, and Stochastic Processes, 2nd Edition,* McGraw-Hill, 1984.

[21] J. G. Proakis, *Digital Communications Fourth Edition.* New York, McGraw-Hill, 2000

[22] B. Sklar, *Digital Communications: Fundamental and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[23] B. R. Saltzberg, "Performance of an efficient parallel data transmission system," IEEE *Trans. on Commun.,* COM-15(6), Dec., 1967, pp. 805–811.

[24] J. D. Terry, *Blind Adaptive Array Techniques for Mobile Satellite Communications,* UMI Company, Ann Arbor, MI, 1999.

[25] A. J. Viterbi, and J. K. Omura, *Principles of Digital Communication and Coding,* McGraw-Hill Book Company, New York, 1979.

[26] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication.* Addison Wesley Publishing Company, 1996.

[27] ETSI TS 101 761-1 V1.2.1 (2000-11), Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions

[28] ETSI TS 101 761-2 V1.2.1 (2001-04), Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC) sublayer

[29] ETSI TS 101 475 V1.2.2 (2001-02), Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer

Synchronization

CHAPTER

2

IN THIS CHAPTER

- Timing Estimation 49
- Frequency Synchronization 66
- Channel Estimation 77
- Clear Channel Assessment 81
- Signal Quality 82
- Bibliography 83

Synchronization is an essential task for any digital communication system. Without accurate synchronization algorithms, it is not possible to reliably receive the transmitted data. From the digital baseband algorithm design engineer's perspective, synchronization algorithms are the major design problem that has to be solved to build a successful product.

OFDM is used for both broadcast type systems and packet switched networks, like WLANs. These two systems require somewhat different approach to the synchronization problem. Broadcast systems transmit data continuously, so a typical receiver, like European Digital Audio Broadcasting (DAB) or Digital Video Broadcasting (DVB) system receivers, can initially spend a relatively long time to acquire the signal and then switch to tracking mode. Speth et al. [24] and [25] analyze synchronization techniques for broadcast receivers. On the other hand, WLAN systems typically have to use so called "single-shot" synchronization; that is, the synchronization has to be acquired during a very short time after the start of the packet. This requirement comes from the packet switched nature of WLAN systems and also from the high data rates used. To achieve good system throughput, it is mandatory to keep the receiver training information overhead to the minimum. To facilitate the single-shot synchronization, current WLAN standards include a preamble, like the IEEE 802.11a [9] preamble shown in Figure 2.4 or the various HiperLAN/2 [7] preambles shown in Figure 2.7, in the start of the packet. The length and the contents of the preamble have been carefully designed to provide enough information for good synchronization performance without any unnecessary overhead.

The OFDM signal waveform makes most of the synchronization algorithms designed for single carrier systems unusable, thus the algorithm design problem has to be approached from the OFDM perspective. This distinction is especially visible on the sensitivity difference to various synchronization errors between single carrier and OFDM systems. The frequency domain nature of OFDM also allows the effect of several synchronization errors to be explained with the aid of the properties of the Discrete Fourier Transform (DFT). Another main distinction with single carrier systems is that many of the OFDM synchronization functions can be performed either in time- or frequency-domain. This flexibility is not available in single carrier systems. The tradeoffs on how to perform the synchronization algorithms are usually either higher performance versus reduced computational complexity.

The scope of this book is WLAN systems; hence this chapter concentrates on WLAN synchronization algorithms. Occasionally we comment on differences between WLAN and broadcast system synchronization algorithms.

The order of the algorithms described in this chapter follows to some extent the order of how an actual receiver would perform the synchronization. The main assumption usually made when WLAN systems are designed is that the channel impulse response does not change significantly during one data burst. This assumption is justified by the quite short time duration of transmitted packets, usually a couple milliseconds at maximum and that the transmitter and receiver in most

applications move very slowly relatively to each other. Under this assumption most of the synchronization for WLAN receivers is done during the preamble and need not be changed during the packet.

Timing Estimation

Timing estimation consists of two main tasks: packet synchronization and symbol synchronization.

The IEEE 802.11 MAC protocol is essentially a random access network, so the receiver does not know exactly when a packet starts. The first task of the receiver is to detect the start of an incoming packet. HiperLAN/2 medium access architecture is a hybrid of both random access and time scheduled networks. Thus HiperLAN/2 receivers also have to be able to reliably detect the start of incoming packets, without prior knowledge.

Broadcast systems naturally do not require packet detection algorithms, because transmission is always on. However, for a packet oriented network architecture, finding the packets is obviously of central importance for high network performance.

Packet Detection

Packet detection is the task of finding an approximate estimate of the start of the preamble of an incoming data packet. As such it is the first synchronization algorithm that is performed, so the rest of the synchronization process is dependent on good packet detection performance. Generally packet detection can be described as a binary hypothesis test. The test consists of two complementary statements about a parameter of interest. These statements are called the null hypothesis, H_0, and the alternative hypothesis, H_1. In the packet detection test, the hypotheses assert whether a packet is present or not. The test is set up as shown below.

$$H_0 : \textit{Packet not present}$$
$$H_1 : \textit{Packet present}$$

The actual test is usually of the form that tests whether a decision variable m_n exceeds a predefined threshold Th. The packet detection case is shown below.

$$H_0 : m_n < Th \Rightarrow \textit{Packet not present}$$
$$H_1 : m_n \geq Th \Rightarrow \textit{Packet present}$$

The performance of the packet detection algorithm can be summarized with two probabilities: probability of detection P_D and probability of false alarm P_{FA}. P_D is the probability of detecting a packet when it is truly present, thus high P_D is a desirable quality for the test. P_{FA} is the probability that the test incorrectly decides that a packet is present, when actually there is none, thus P_{FA} should be as small as possible. In general, increasing P_D increases P_{FA} and decreasing P_{FA}

decreases P_D, hence the algorithm designer must settle for a some balanced compromise between the two conflicting goals. The general hypothesis test problem is discussed in several statistical inference and detection theory books [3, 10].

Generally it can be said that a false alarm is a less severe error than not detecting a packet at all. The reason is that after a false alarm, the receiver will try to synchronize to nonexistent packet and will detect its error at the first received data integrity check. On the another hand, not detecting a packet always results in lost data. A false alarm can also result in lost data, in case an actual data packet starts during the time the receiver has not yet detected its mistake. In this case, the receiver will not be able to catch that packet. The probability of this occurring depends on several issues like the network load and the time it takes for the receiver to detect its mistake. In conclusion, a little higher P_{FA} can be tolerated to guarantee good P_D. In the next sections, we introduce several approaches to the packet detection test design problem.

Received Signal Energy Detection

The simplest algorithm for finding the start edge of the incoming packet is to measure the received signal energy. When there is no packet being received, the received signal r_n consists only of noise $r_n = w_n$. When the packet starts, the received energy is increased by the signal component $r_n = s_n + w_n$, thus the packet can be detected as a change in the received energy level. The decision variable m_n is then the received signal energy accumulated over some window of length L to reduce sensitivity to large individual noise samples.

$$m_n = \sum_{k=0}^{L-1} r_{n-k} r_{n-k}^* = \sum_{k=0}^{L-1} |r_{n-k}|^2 \tag{2.1}$$

Calculation of m_n can be simplified by noting that it is a moving sum of the received signal energy. This type of sum is also called a sliding window. The rationale for the name is that at every time instant n, one new value enters the sum and one old value is disgarded. This structure can be used to simplify the computation of Equation 2.1. Equation 2.2 shows how to calcute the moving sum recursively.

$$m_{n+1} = m_n + |r_{n+1}|^2 - |r_{n-L+1}|^2 \tag{2.2}$$

Thus the number of complex multiplications is reduced to one per received sample; however, more memory is required to store all the values of $|r_n|^2$ inside the window. The response of this algorithm is shown in Figure 2.1. The figure shows the value m_n for IEEE 802.11a packet with 10dB Signal to Noise Ration (SNR) and sliding window length $L = 32$. The true start of the packet is at $n = 500$, thus in this case the threshold could be set between 10 and 25. The value of the threshold defines the P_D and P_{FA} of the test.

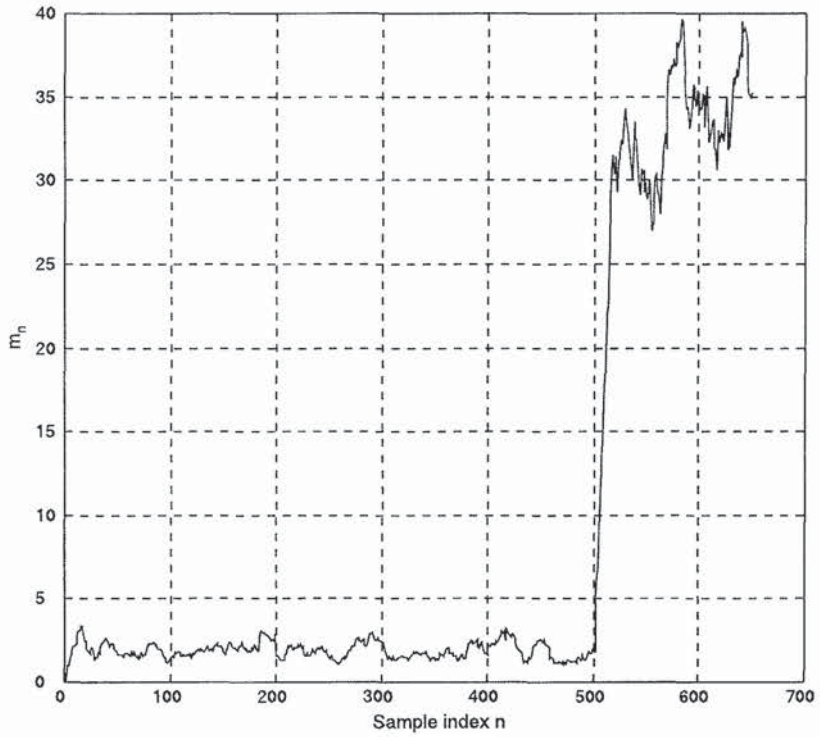


FIGURE 2.1
Received signal energy based packet detection algorithm.

This simple method suffers from a significant drawback; namely, the value of the threshold depends on the received signal energy. When the receiver is searching for an incoming packet, the received signal consists of only noise. The level of the noise power is generally unknown and can change when the receiver adjusts its Radio Frequency (RF) amplifier settings or if unwanted interferers go on and off in the same band as the desired system. When a wanted packet is incoming, its received signal strength depends on the power setting of the transmitter and on the total path loss from the transmitter to the receiver. All these factors make it quite difficult to set a fixed threshold, which could be used to decide when an incoming packet starts. The next section describes an improvement to the algorithm that alleviates the threshold value selection problem.

Double Sliding Window Packet Detection

The double sliding window packet detection algorithm calculates two consecutive sliding windows of the received energy. The basic principle is to form the decision variable m_n as a ratio of the total energy contained inside the two windows. Figure 2.2 shows the windows A and B and the response of m_n to a received packet. In Figure 2.2, the A and B windows are considered stationary relative to the packet that slides over them to the right. It can be seen that when only noise

is received the response is flat, since both windows contain ideally the same amount of noise energy. When the packet edge starts to cover the A window, the energy in the A window gets higher until the point where A is totally contained inside the start of the packet. This point is the peak of the triangle shaped m_n and the position of the packet in Figure 2.2 corresponds to this sample index n. After this point B window starts to also collect signal energy, and when it is also completely inside the received packet, the response of m_n is flat again. Thus the response of m_n can be thought of as a differentiator, in that its value is large when the input energy level changes rapidly. The packet detection is declared when m_n crosses over the threshold value Th.

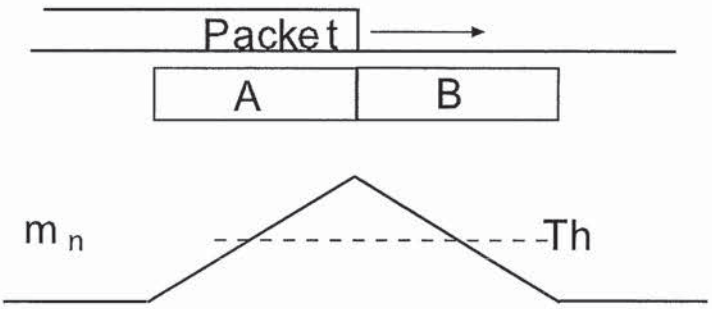


FIGURE 2.2
The response of the double sliding window packet detection algorithm.

Equation 2.3 shows the calculation of the A window value and Equation 2.4 the calculation for B window.

$$a_n = \sum_{m=0}^{M-1} r_{n-m} r_{n-m}^* = \sum_{m=0}^{M-1} |r_{n-m}|^2 \tag{2.3}$$

$$b_n = \sum_{l=1}^{L} r_{n+l} r_{n+l}^* = \sum_{l=0}^{L} |r_{n+l}|^2 \tag{2.4}$$

Both a_n and b_n are again sliding windows, thus the computation can be simplified in the same recursive manner as for the energy detection window. Then the decision variable is formed by dividing the value of the a_n by b_n

$$m_n = \frac{a_n}{b_n} \tag{2.5}$$

A simulated response of the variable m_n is shown in Figure 2.3. The figure is again for the IEEE 802.11a preamble with 10dB SNR. The figure clearly shows how the value of m_n does *not* depend on the total received power. After the peak, the response levels off to the same value as before the peak, although the received energy level is much higher. An additional benefit of this

approach is that, at the peak point of m_n the value of a_n contains the sum of signal energy S and noise energy N and the b_n value is equal to noise energy N, thus the value of m_n at the peak point can be used to estimate the received SNR. Equation 2.6 is the ratio a_n and b_n at the peak point and Equation 2.7 is the SNR estimate calculated from the ratio.

$$m_{peak} = \frac{a_{peak}}{b_{peak}} = \frac{S+N}{N} = \frac{S}{N} + 1 \tag{2.6}$$

$$\widehat{SNR} = m_{peak} - 1 \tag{2.7}$$

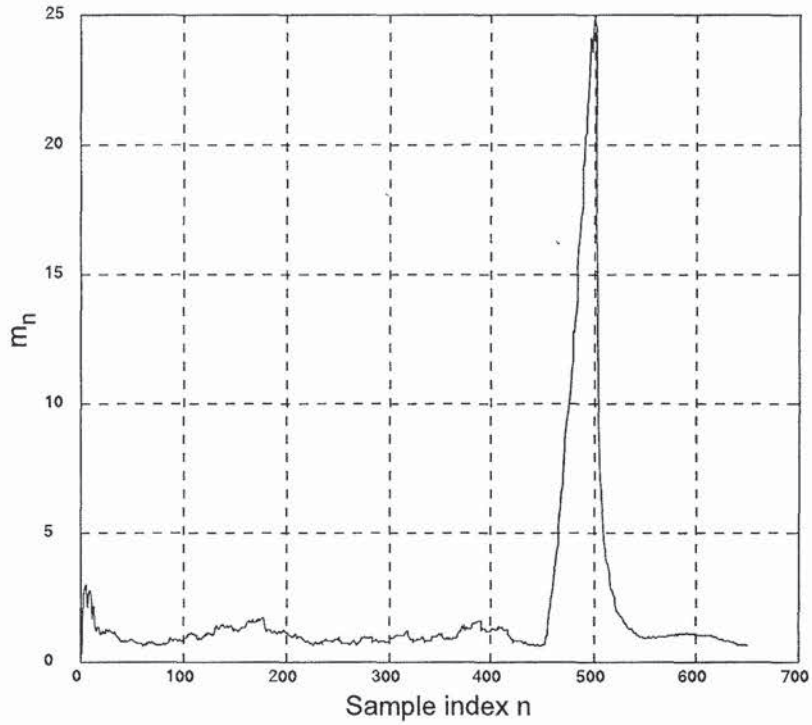


FIGURE 2.3
Double sliding window packet detection.

Using the double sliding window algorithm is a good approach, if the receiver does not have additional information about the received data. However, if the receiver does have additional information, more can be done as is explained in the following section.

Using the Structure of the Preamble for Packet Detection

A general communications system engineering principle is that the receiver should use all the available information to its advantage. In the packet detection algorithm, this means that the

known structure of the preamble should be incorporated into the algorithm. The preambles of both the IEEE 802.11a and HiperLAN/2 systems have been designed to help the detection of the start edge of the packet. The IEEE 802.11a standard gives guidelines on how to use the various segments of the preamble to perform the necessary synchronization functions. The preamble structure and the guidelines are illustrated in Figure 2.4. The parts from A_1 to A_{10} are short training symbols, that are all identical and 16 samples long. *CP* is a 32-sample cyclic prefix that protects the long training symbols C_1 and C_2 from intersymbol interference (ISI) caused by the short training symbols. The long training symbols are identical 64 samples long OFDM symbols. However, the guidelines are not binding requirements of the standard. The design engineer has the freedom to use any other available method or develop new algorithms.

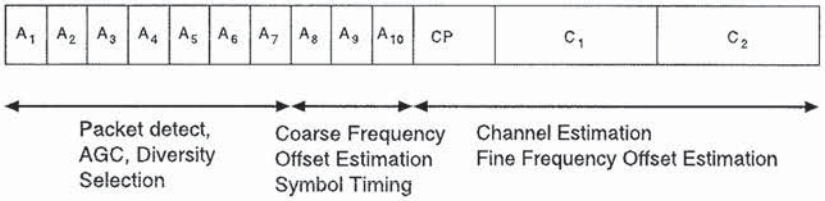


FIGURE 2.4
*The IEEE 802.11a standard preamble.**

The structure of the WLAN preamble enables the receiver to use a very simple and efficient algorithm to detect the packet. The following approach was presented in Schimdl and Cox [21] for acquiring symbol timing, but the general method is applicable to packet detection. Overall the method resembles the double sliding window algorithm introduced in the previous section, but it takes advantage of the periodicity of the short training symbols at the start of the preamble. This approach is called the delay and correlate algorithm. The name and the relation to the double sliding window method is illustrated in Figure 2.5, which shows the signal flow structure of the delay and correlate algorithm. The Figure shows two sliding windows *C* and *P*. The *C* window is a crosscorrelation between the received signal and a delayed version of the received signal, hence the name delay and correlate. The delay z^{-D} is equal to the period of the start of the preamble; for example, IEEE 802.11a has $D = 16$, the period of the short training symbols. The *P* window calculates the received signal energy during the crosscorrelation window. The value of the *P* window is used to normalize the decision statistic, so that it is not dependent on absolute received power level.

The value c_n is calculated according to Equation 2.8, and the value of p_n according to Equation 2.9.

*From IEEE Std. 802.11a-1999. Copyright 1999 IEEE. All rights reserved.

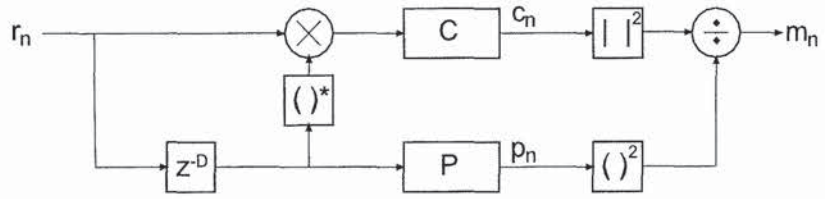


FIGURE 2.5

Signal flow structure of the delay and correlate algorithm.

$$c_n = \sum_{k=0}^{L-1} r_{n+k} r_{n+k+D}^{*} \tag{2.8}$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+D} r_{n+k+D}^{*} = \sum_{k=0}^{L-1} |r_{n+k+D}|^2 \tag{2.9}$$

Then the decision statistic m_n is calculated from Equation 2.10

$$m_n = \frac{|c_n|^2}{(p_n)^2} \tag{2.10}$$

c_n and p_n are again sliding windows, so the general recursive procedure can be used to reduce computational workload.

Figure 2.6 shows an example of the decision statistic m_n for IEEE 802.11a preamble in 10dB SNR. The overall response is restricted between $[0,1]$ and the step at the start of the packet is very clear. The difference of the response to the double sliding window response in Figure 2.3 can be explained with the behavior of the c_n value. When the received signal consists of only noise, the output c_n of the delayed crosscorrelation is zero-mean random variable, since the crosscorrelation of noise samples is zero. This explains the low level of m_n before the start of the packet. Once the start of the packet is received, c_n is a crosscorrelation of the identical short training symbols, which causes m_n to jump quickly to its maximum value. This jump gives a quite good estimate of the start of the packet.

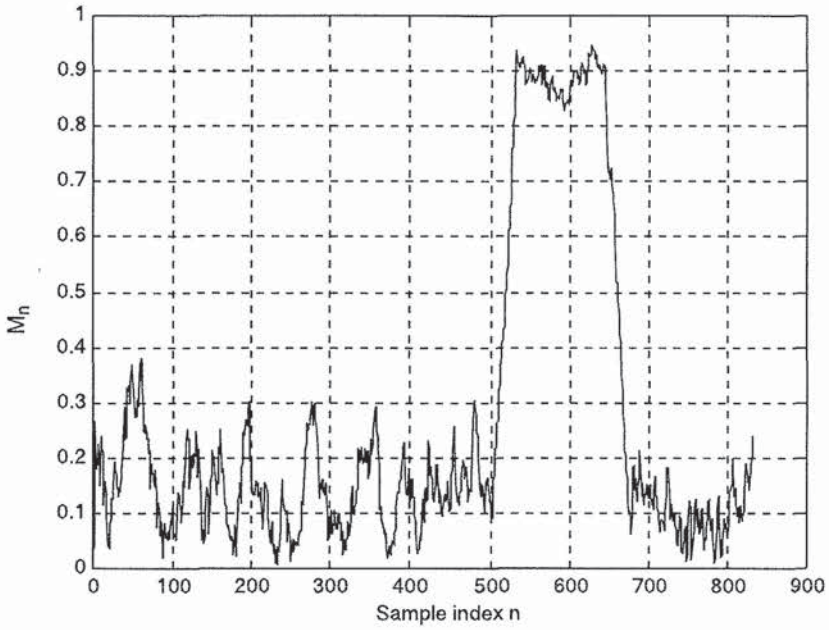


FIGURE 2.6
Response of the delay and correlate packet detection.

Exercise 1 *The simulation tool implements the delay- and correlate packet detection algorithm. Write a Matlab script for the double sliding window algorithm and compare its performance with the original algorithm.*

HiperLAN/2 Preambles

HiperLAN/2 has been designed with several different preambles, in contrast to the IEEE 802.11a that has only one preamble. The rationale for HiperLAN/2's several preambles is the different MAC architecture of HiperLAN/2. The centralized control approach of HiperLAN/2 enables some optimizations of the preamble design; namely, the preamble length is variable. This reduces overhead for some packets, which in turn increases system capacity. The different preamble structures of HiperLAN/2 are shown in Figure 2.7. The letters inside the figure designate different waveforms that are used during the preamble, the *I* preceding a letter means that the signal is inverted; for example *IA* is equal to $-A$. Preamble (a) is used for broadcast packets from the Access Point (AP); it is a full-length structure, since all the stations have to be able to receive the packet. The preamble (b) is a downlink preamble; it does not contain the short training symbols, because the all the stations know the starting point of the packet and frequency synchronization information. Preamble (c) is intented for generel uplink use. Preamble (d) can be used instead of (c), if the AP supports it. The main reason for the longer (d) preamble is a possibility to use

switch antenna diversity, this technique is discussed in more detail in Chapter 4, "Antenna Diversity." The longer short training symbol section allows the receiver to measure the received signal power for two antennas before the long training symbols, and make a decision of which antenna to use.

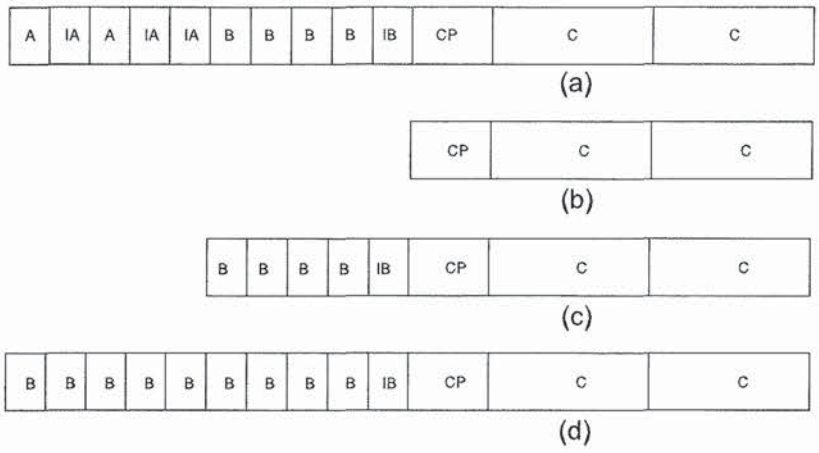


FIGURE 2.7
*Different preambles of the HiperLAN/2 standard.**

The different patterns for the short training symbols, when used with the delay and correlate algorithm, allow the receiver to distinguish between the different types of bursts. The response of m_n to preambles (c) and (d) is the same as for the IEEE 802.11a preamble, and is shown in Figure 2.8 (b). The response to preamble (a), however, is a zig-zag curve as the windows slide through multiple sections A, IA, B, and BI, this is shown in Figure 2.8 (a). This zig-zag response designates the broadcast packets that every station should decode. However, if a station receives a flat response, and it is not expecting a packet, decoding is not necessary.

Symbol Timing

Symbol timing refers to the task of finding the precise moment of when individual OFDM symbols start and end. The symbol timing result defines the DFT window; i.e., the set of samples used to calculate DFT of each received OFDM symbol. The DFT result is then used to demodulate the subcarriers of the symbol. The approach to symbol timing is different for WLAN and broadcast OFDM systems. A WLAN receiver cannot afford to spend any time beyond the preamble to find the symbol timing, whereas a broadcast transmission receiver can spend several symbols to acquire an accurate symbol timing estimate.

© ETSI 2000. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from publication@etsi.fr and http://www.etsi.org/eds/home.htm.

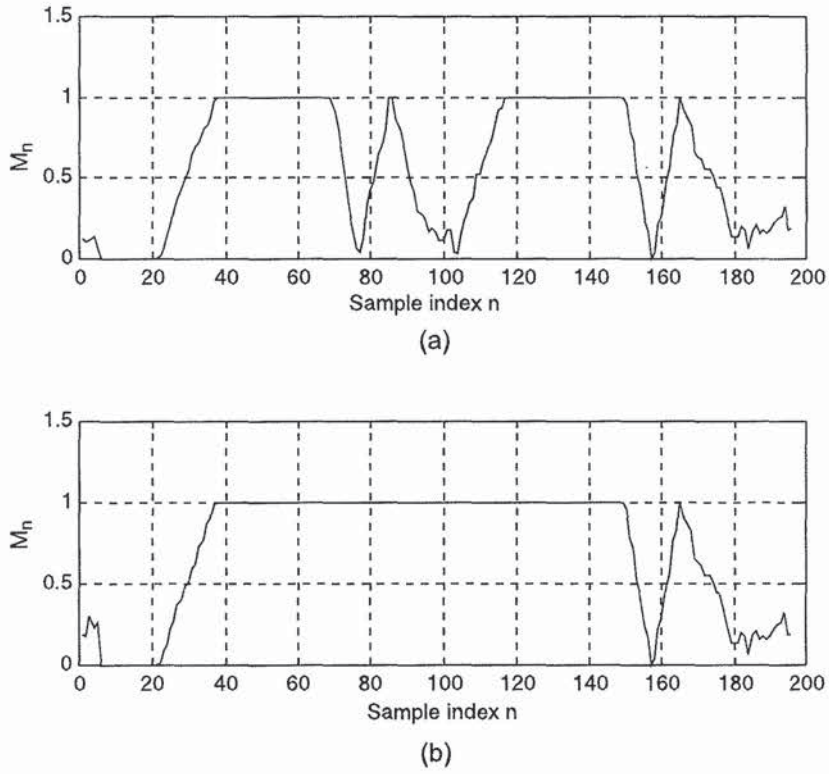


FIGURE 2.8

Response of delay and correlate to HiperLAN/2 preambles (a) and (d).

Symbol Timing for a WLAN Receiver

WLAN receivers have knowledge of the preamble available to them, which enables the receiver to use simple crosscorrelation based symbol timing algorithm. After the packet detector has provided an estimate of the start edge of the packet, the symbol timing algorithm refines the estimate to sample level precision. The refinement is performed by calculating the crosscorrelation of the received signal r_n and a known reference t_k; for example, the end of the short training symbols or the start of the long training symbols, to find the symbol timing estimate. Equation 2.11 shows how to calculate the crosscorrelation. The value of n that corresponds to maximum absolute value of the crosscorrelation is the symbol timing estimate.

$$\hat{t}_s = \arg\max_n \left| \sum_{k=0}^{L-1} r_{n+k} t_k^* \right|^2 \tag{2.11}$$

In Equation 2.11, the length L of the crosscorrelation determines the performance of the algorithm. Larger values improve performance, but also increase the amount of computation required.

In hardware implementations, it is possible to use only the sign of the reference and received signals, effectively quantizing them to one-bit accuracy. This greatly simplifies the hardware implementation, since no actual multiplications are necessary. Figure 2.9 shows the output of the crosscorrelator that uses the first 64 samples of the long training symbols of the IEEE 802.11a standard as the reference signal. The simulation was run in Additive White Gaussian Noise (AWGN) channel with 10dB SNR. The high peak at $n = 77$ clearly shows the correct symbol timing point.

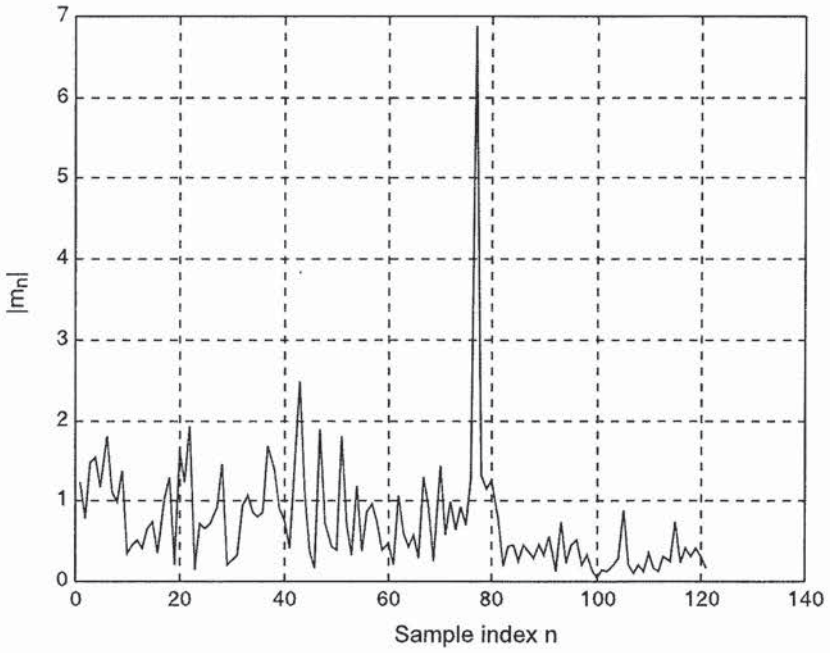


FIGURE 2.9
Response of the symbol timing crosscorrelator.

Optimization of Symbol Timing in a Multipath Channel

The performance of the symbol timing algorithm directly influences the effective multipath tolerance of a OFDM system. Characteristics of multipath channels were introduced in Chapter 1, "Background and WLAN Overview." An OFDM receiver achieves maximum multipath tolerance when symbol timing is fixed to the first sample of an OFDM symbol. Figure 2.10 shows three consecutive OFDM symbols, their respective cyclic prefixes (CP), and the effect of symbol timing on the DFT window. The ideal timing for symbol 2 is shown as the ideal DFT window in Figure 2.10 (a). In this case the DFT window does not contain any samples from the CP, hence the maximum length of the channel impulse response (CIR) that does not cause intersymbol interference (ISI) is equal to the CP length. This is the maximum possible ISI free multipath

length for an OFDM system. In practice, it is impossible to fix the symbol timing point perfectly to the first sample of the OFDM symbol. There will always be some variability in the symbol timing estimate around its mean value. In addition to the ideal symbol timing, Figure 2.10 (a) shows the probability density function (PDF) of a realistic symbol timing estimate with the mean value equal to the ideal symbol timing. The PDF shows how the actual symbol timing estimate can be before or after the ideal value. The effect on the DFT window is shown as the DFT window range; that is, the range within which the DFT window will fall. When the symbol timing point is estimated before the ideal value, the start of the DFT window will contain samples from the CP and the last samples of the symbol are not used at all. This case does not cause serious problems, because the CP is equal to the last samples of the symbol the circular convolution property of DFT is still satisfied as shown in Equation 1.74. It is possible that some ISI is caused by the early symbol timing estimate, if the CIR is long enough to reach the first samples of the DFT window. For a properly designed OFDM system, the amount of this interference is negligible, because the last taps of the CIR are small. Next consider the case when the symbol timing estimate is after the ideal value. In this case, the start of the DFT window will be after the first sample of the symbol and the last samples are taken from the beginning of the CP of the next symbol. For example, in Figure 2.10 (a), this would mean the the DFT window for symbol 2 would contain samples from the CP 3. When this happens, significant ISI is created by the samples form CP of the next symbol. Additionally the circular convolution property required for the orthogonality of the subcarriers is no longer true, hence intercarrier interference is generated. The end result of a late symbol timing estimate is a significant performance loss. Fortunately, there is a simple solution for this problem. Since early symbol timing does not create significant problems, the mean value of the symbol timing point can be shifted inside the CP. This is shown in Figure 2.10 (b). After the shift, the PDF of the symbol timing estimate is entirely contained inside the CP 2, hence the DFT range is also within CP 2 and symbol 2. This means that the circular convolution is preserved and no ISI is caused by the samples from CP 3. The optimal amount of the shift depends on the OFDM system paremeters and the performance of the symbol timing algorithm. A rule of thumb for an IEEE 802.11a system is 4-6 samples. The drawback of the shift of the mean symbol timing point is reduced multipath tolerance of the OFDM system. The shift effectively shortens the CP by the amount of the shift, because some samples of the CP are always used for the DFT window. This means that a CIR shorter that the CP can cause ISI; however, as was mentioned earlier, the last taps of the CIR are quite small, so the effect on performance is not serious. Nevertheless the OFDM system designer should keep the symbol timing in mind and put some samples in to the CP in addition to the maximum expected CIR length.

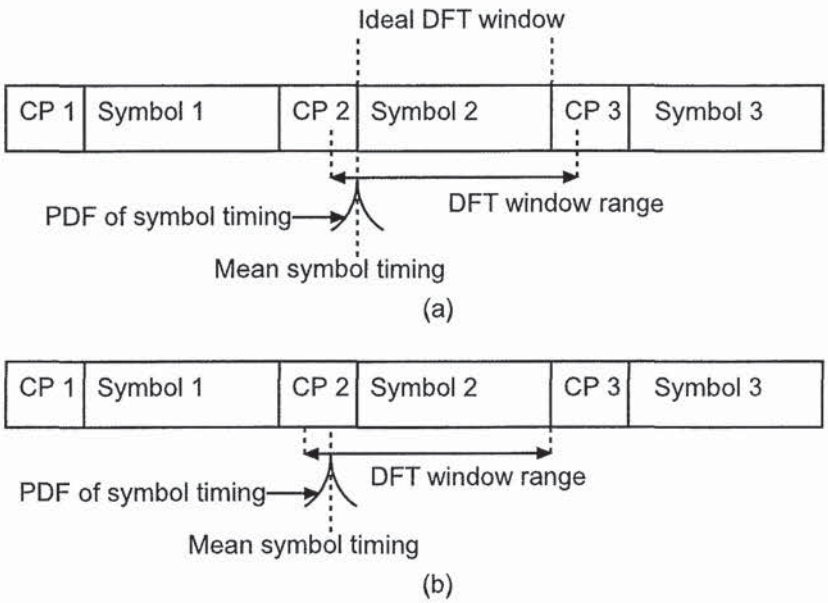


FIGURE 2.10
Effect of variance in symbol timing.

Exercise 2 *The simulation tool allows to change the amount of the symbol timing shift. Experiment with different values with a multipath channel model, and observe the performance difference.*

In a multipath environment, the symbol timing can be further optimized, if an estimate of the multipath taps h_n of the channel impulse response is available to the receiver. This approach is discussed in more detail in [15]. Figure 2.11 shows a sample of a channel impulse response. In the sample, the strongest tap occurs several taps later than the first one. The crosscorrelation symbol timing estimator would pick the strongest tap 3 as the timing point; however, this will mean that the taps 0 to 2 will not contribute to the signal energy inside the DFT window. This means that some signal energy is lost due to the symbol timing estimate. This loss can be recovered by changing the timing point such that the energy of h_n inside the DFT window is maximized. For the CIR sample in Figure 2.11, this would mean shifting the timing point to the left to cover the taps that occur before the strongest tap.

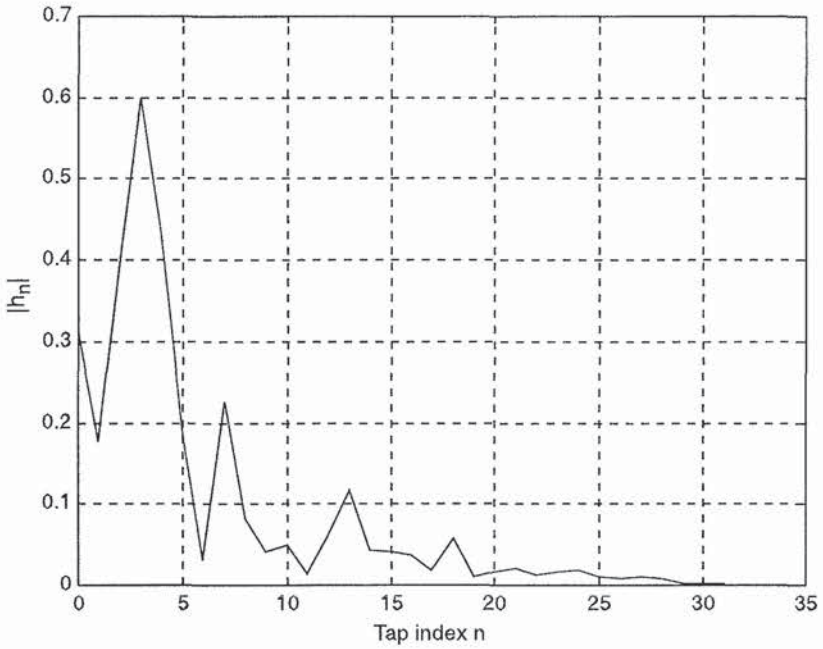


FIGURE 2.11
A sample of a channel impulse response.

Continuous Transmission System Symbol Timing

The different symbol timing algorithms for continuous transmission receivers can coarsely be divided into data-aided and nondata-aided methods. To facilitate data-aided methods broadcast OFDM systems like DVB, periodically insert known training symbols into the transmitted signal. These symbols are called *pilot symbols*. The nondata-aided methods use the cyclic prefix structure of OFDM waveform to perform symbol timing. In this sense, they are similar to the preamble structure packet detection algorithm; the algorithm presented in Schimdl and Cox [21] was developed for symbol timing, but the approach could also be used packet detection. Several authors have investigated symbol timing based on the cyclic prefix [1, 2]. Okada et al. [16] develop a technique for jointly estimating the symbol timing and frequency offset of the received signal. The underlying theme in all these techniques is the use of correlation to detect the repetition of the cyclic prefix in the received signal.

Sample Clock Tracking

A quite different timing estimation problem is tracking the sampling clock frequency. The oscillators used to generate the Digital to Analog Converter (DAC) and Analog to Digital Converter (ADC) sampling instants at the transmitter and receiver will never have exactly the same period.

Thus the sampling instants slowly shift relative to each other. The problem has been analyzed by several authors [6, 11, 17, 19].

The sampling clock error has two main effects: A slow shift of the symbol timing point, which rotates subcarriers, and a loss of SNR due to the intercarrier interference (ICI) generated by the slightly incorrect sampling instants, which causes loss of the orthogonality of the subcarriers. In [23] the normalized sampling error is defined as

$$t_\Delta = \frac{T' - T}{T} \tag{2.12}$$

where T and T' are the transmitter and receiver sampling periods, respectively. Then the overall effect, after DFT, on the received subcarriers $R_{l,k}$ is shown as

$$R_{l,k} = e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} X_{l,k} \operatorname{sinc}(\pi k t_\Delta) H_{l,k} + W_{l,k} + N_{t_\Delta}(l,k) \tag{2.13}$$

where l is the OFDM symbol index, k is the subcarrier index, T_s and T_u are the duration of the total OFDM symbol and the useful data portion, $W_{l,k}$ is additive white noise and the last term $N_{t_\Delta}(l,k)$ is the additional interference due to the sampling frequency offset. The power of the last term is approximated by

$$P_{t_\Delta} \approx \frac{\pi^2}{3}(k t_\Delta)^2 \tag{2.14}$$

hence the degradation grows as the square of the product of the offset t_Δ and the subcarrier index k. This means that the outermost subcarriers are most severely affected. The degradation can also be expressed directly as SNR loss in decibels. The following approximation [19] is derived

$$D_n \approx 10 \log_{10}\left(1 + \frac{\pi^2}{3}\frac{E_s}{N_0}(k t_\Delta)^2\right) \tag{2.15}$$

WLAN OFDM systems typically have relatively small number of subcarriers and quite small t_Δ, hence $k t_\Delta \ll 1$, so the interference caused by sampling frequency offset can usually be ignored. Equation 2.13 also shows the more significant problem caused by the offset, namely the term

$$e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} \tag{2.16}$$

This term shows the amount of rotation angle experienced by the different subcarriers as an OFDM signal is received. The angle depends on both the subcarrier index k and the OFDM symbol index l. Hence the angle is largest for the outermost subcarriers and increases with consecutive OFDM symbols. The term t_Δ is usually quite small, but as l increases, the rotation will

eventually be so large that correct demodulation is no longer possible. This necessitates tracking the sampling frequency offset.

Estimating the Sampling Frequency Error

The majority of the published approaches to estimating the sampling frequency offset rely on *pilot subcarriers*. The pilot subcarriers are used to transmit known data, called *pilot symbols*, that the receiver can use to perform synchronization functions. Sampling frequency error estimation algorithms usually assume the pilots to be symmetrically distributed around a middle subcarrier. The following method was presented in Speth et al. [25] and Fechtel [6]. The pilot subcarriers are divided into two sets; C_1 corresponds to pilots on negative subcarriers, and C_2 to pilots on positive subcarriers. The sampling frequency offset is estimated by using the knowledge of the linear relationship between the phase rotation caused by the offset and the pilot subcarrier index. The received pilot subcarriers, in a simplified form, are

$$R_{l,k} = H_k P_{l,k} e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} \tag{2.17}$$

Then calculate the rotation of the pilots from one symbol to the next

$$Z_{l,k} = R_{l,k} R^*_{l-1,k} \tag{2.18}$$

$$= H_k P_{l,k} e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} \left(H_k P_{l-1,k} e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} \right)^* \tag{2.19}$$

$$= |H_k|^2 |P_{l,k}|^2 e^{\left\{ j2\pi k t_\Delta l \frac{T_s}{T_u} \right\} \left\{ -j2\pi k t_\Delta (l-1) \frac{T_s}{T_u} \right\}} \tag{2.20}$$

$$= |H_k|^2 |P_{l,k}|^2 e^{j2\pi k t_\Delta l \frac{T_s}{T_u}} \tag{2.21}$$

Next calculate the cumulative phases of $Z_{l,k}$ for the two sets C_1 and C_2 as

$$\phi_{1,l} = \angle \left[\sum_{k \in C_1} Z_{l,k} \right] \tag{2.22}$$

$$\phi_{2,l} = \angle \left[\sum_{k \in C2} Z_{l,k} \right] \tag{2.23}$$

The sampling frequency offset is then estimated by

$$\hat{t}_\Delta = \frac{1}{2\pi} \frac{T_u}{T_s} \frac{1}{\min_{k \in C_2}(k) + \max_{k \in C_2}(k)} \left(\phi_{2,l} - \phi_{1,l} \right) \tag{2.24}$$

where the normalization factor $\min_{k \in C_2}(k) + \max_{k \in C_2}(k)$ assumes that the pilot indexes k are evenly distributed.

Correcting the Sampling Frequency Error

The rotation caused by the sampling frequency offset can be corrected with two main approaches. Namely, the problem can be corrected at its source by adjusting the sampling frequency of the receiver DAC—the receiver structure is shown on top of Figure 2.12. Secondly, the rotation can be corrected after the DFT processing by derotating the subcarriers, the lower receiver structure in Figure 2.12. Both of these solutions are analyzed in Pollet et al. [19], where the first method is referred to as synchronized sampling and the second as non-synchronized sampling.

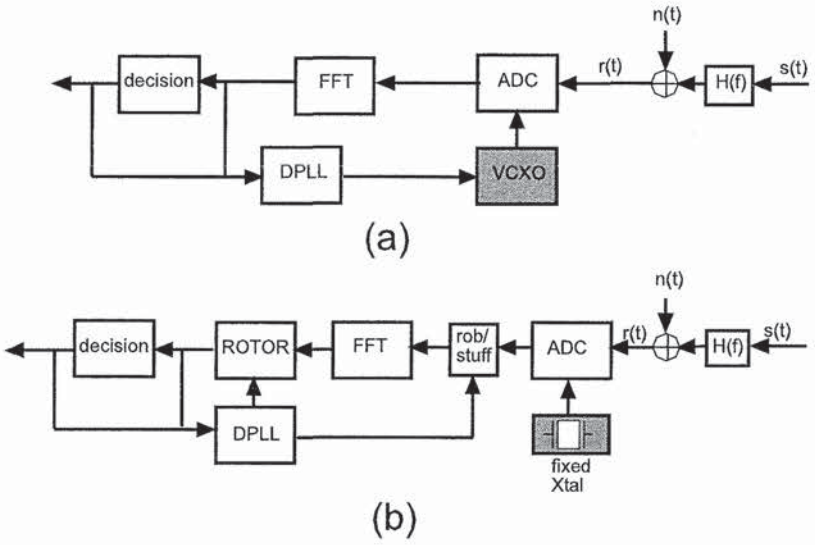


FIGURE 2.12

Adapted from Pollet et. al. [19]: Receiver structures for correcting sampling frequency error. © 1994 IEEE.

The adjustment of the clock of the ADC naturally perfectly removes the sampling frequency offset, provided the estimate of the offset is accurate. In this sense, it is the optimal way of performing the correction. However, during recent years, the trend in receiver design has been towards all digital receivers, that do not attempt to adjust the sampling clock. This is illustrated in the lower receiver structure in Figure 2.12 by the fixed crystal that controls the ADC. The rationale for this trend is the desire to simplify the analog part of the receiver, as the analog components are relatively costly compared to digital gates. Hence by using a fixed crystal, instead of a controllable one, the number of analog components can be reduced, and as a result the cost of the receiver can be decreased. The non-synchronized sampling receiver in Figure 2.12 (b) shows an additional

block named "rob/stuff," just after the ADC. This block is required, because the drift in sampling instant will eventually be larger than the sampling period. When this happens, the "rob/stuff" block will either "stuff" a duplicate sample or "rob" one sample from the signal, depending on whether the receiver clock is faster or slower than the transmitter clock. This process prevents the receiver sampling instant from drifting so much that the symbol timing would be incorrect. The "ROTOR" block performs the required phase corrections with the information provided by the Digital Phase-Locked Loop (DPLL) that estimates the sampling frequency error.

Frequency Synchronization

One of the main drawbacks of OFDM is its sensitivity to carrier frequency offset. The effect of carrier frequency error on SNR was introduced in Chapter 1, where the main differences between multicarrier and singlecarrier systems were discussed. The degradation is caused by two main phenomena: reduction of amplitude of the desired subcarrier and ICI caused by neighboring carriers. The amplitude loss occurs because the desired subcarrier is no longer sampled at the peak of the sinc-function of DFT. The sinc-function is defined as $\text{sinc}\,(x) = \frac{\sin(x)}{x}$. Adjacent carriers cause interference, because they are not sampled at the zero-crossings of they sinc-functions. The overall effect on SNR is analyzed in Pollet et al. [18], and for relatively small frequency errors, the degradation in dB was approximated by

$$SNR_{Loss} = \frac{10}{3\ln 10}(\pi T f_{\Delta})^2 \frac{E_s}{N_0}\,\text{dB} \tag{2.25}$$

where f_{Δ} is the frequency error as a fraction of the subcarrier spacing and T is the sampling period. The performance effect varies strongly with the modulation used; naturally, constellations with fewer points can tolerate larger frequency errors than large constellations. Figure 2.13 illustrates the effect for QPSK and 64-QAM constellations. The figure shows that the 64QAM, the largest constellation in IEEE802.11a, cannot tolerate more than 1% error in the carrier frequency for a negligible SNR loss of 0.5dB, whereas QPSK modulation can tolerate up to 5% error for the same SNR loss.

The analysis above would seem to indicate that large constellations are exceedingly difficult to use with an OFDM system. However, keep in mind that a large constellation automatically implies higher operating SNR than with a small constellation. This directly improves the performance of the frequency error estimators, as is shown in Equations 2.41 and 2.57.

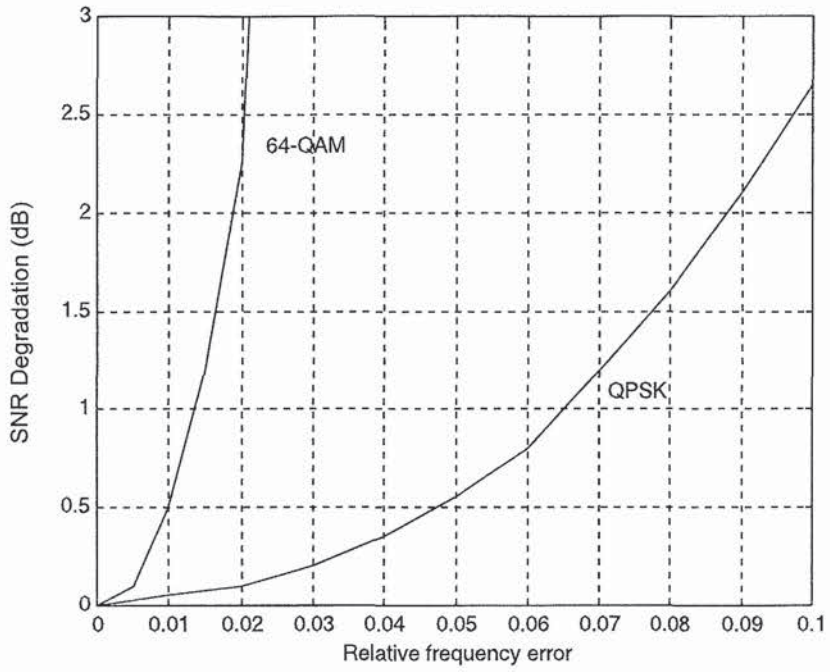


FIGURE 2.13

Symbol error rate (SER) degradation due to frequency offset at SER = 10^{-4}. *© 1995 IEEE. Adapted from Daffara and Adami [5].*

In Hsieh and Wei [8], the various algorithms that have been developed to estimate carrier frequency offsets in OFDM systems are divided into three types:

Type 1 Data-aided algorithms; these methods are based on special training information embedded into the transmitted signal.

Type 2 Nondata-aided algorithms that analyze the received signal in frequency domain.

Type 3 Cyclic prefix based algorithms that use the inherit structure of the OFDM signal provided by the cyclic prefix.

For WLAN applications, type 1 is the most important. The preamble allows the receiver to use efficient maximum likelihood algorithms to estimate and correct for the frequency offset, before the actual information portion of the packet starts. The algorithms belonging to types 2 and 3 are better suited for broadcast or continuous transmission type OFDM systems. Owing to the nature of this book, we will focus on data-aided frequency synchronization algorithms in this section.

Time Domain Approach for Frequency Synchronization

We first derive a data-aided maximum-likelihood estimator that operates on the received time domain signal. This method has been presented in several papers in slightly varying forms [2, 21]. The training information required is at least two consecutive repeated symbols. As a note,

the preamble of the WLAN standards satisfies this requirement for both the short and long training symbols.

Let the transmitted signal be s_n, then the complex baseband model of the passband signal y_n is

$$y_n = s_n e^{j2\pi f_{tx} n T_s} \tag{2.26}$$

where f_{tx} is the transmitter carrier frequency. After the receiver downconverts the signal with a carrier frequency f_{rx}, the received complex baseband signal r_n, ignoring noise for the moment, is

$$r_n = s_n e^{j2\pi f_{tx} n T_s} e^{-j2\pi f_{rx} n T_s} \tag{2.27}$$

$$= s_n e^{j2\pi (f_{tx} - f_{rx}) n T_s} \tag{2.28}$$

$$= s_n e^{j2\pi f_\Delta n T_s} \tag{2.29}$$

where $f_\Delta = f_{tx} - f_{rx}$ is the difference between the transmitter and receiver carrier frequencies. Let D be the delay between the identical samples of the two repeated symbols. Then the frequency offset estimator is developed as follows, starting with an intermediate variable z

$$z = \sum_{n=0}^{L-1} r_n r_{n+D}^* \tag{2.30}$$

$$= \sum_{n=0}^{L-1} s_n e^{j2\pi f_\Delta n T_s} \left(s_{n+D} e^{j2\pi f_\Delta (n+D) T_s} \right)^* \tag{2.31}$$

$$= \sum_{n=0}^{L-1} s_n s_{n+D}^* e^{j2\pi f_\Delta n T_s} e^{-j2\pi f_\Delta (n+D) T_s} \tag{2.32}$$

$$= e^{-j2\pi f_\Delta D T_s} \sum_{n=0}^{L-1} |s_n|^2 \tag{2.33}$$

Equation 2.33 is a sum of complex variables with an angle proportional to the frequency offset. Finally, the frequency error estimator is formed as

$$\hat{f}_\Delta = -\frac{1}{2\pi D T_s} \angle z \tag{2.34}$$

where the $\angle z$ operator takes the angle of its argument.

Properties of the Time Domain Frequency Synchronization Algorithm

An important feature of the present method is its operating range. The operating range defines how large frequency offset can be estimated. The range is directly related to the length of the

repeated symbols. The angle of z is of the form $-2\pi f_\Delta DT_s$, which is unambiguously defined only in the range $[-\pi, \pi)$. Thus if the absolute value of the frequency error is larger than the following limit

$$|f_\Delta| \geq \frac{\pi}{2\pi DT_s} = \frac{1}{2DT_s} \tag{2.35}$$

the estimate will be incorrect, since z has rotated an angle larger than π. This maximum allowable frequency error is usually normalized with the subcarrier spacing f_s. If the delay D is equal to the symbol length, then

$$\frac{1}{2DT_s} = \frac{1}{2} f_s \tag{2.36}$$

Thus the frequency error can be at most a half of the subcarrier spacing. It should be noted that if the repeated symbols include a cyclic prefix, the delay is longer than the symbol length, and hence the range of the estimator is reduced.

As an example we can calculate the value of this limit for the IEEE 802.11a system for both the short and long training symbols. For the short training symbols, the sample time is 50ns, and the delay $D = 16$. Thus the maximum frequency error that can be estimated is

$$f_{\Delta\max} = \frac{1}{2DT_s} \tag{2.37}$$

$$= \frac{1}{2 \cdot 16 \cdot 50 \cdot 10^{-9}} \tag{2.38}$$
$$= 625kHz$$

This should be compared with the maximum possible frequency error in an IEEE802.11a system. The carrier frequency is approximately 5.3 GHz, and the standard specifies a maximum oscillator error of 20 parts per million (ppm). Thus if the transmitter and receiver clocks have the maximum allowed error, but with opposite signs, the total observed error will be 40ppm. This amounts to a frequency error of $f_\Delta = 40 \cdot 10^{-6} \cdot 5.3 \cdot 10^9 = 212kHz$. Hence the maximum possible frequency error is well within the range of the algorithm. Now consider the long training symbols. The only significant difference is that the delay $D = 64$ is four times longer. Hence the range is

$$f_{\Delta\max} = \frac{1}{4 \cdot 2DT_s} \tag{2.39}$$

$$= 156.25kHz \tag{2.40}$$

Observe that this is less than the maximum possible error defined in the standard. Thus this estimator would not be reliable if only the long training symbols were used.

Beek et al. [2] show that in an AWGN channel the estimator $\hat{f}_\Delta$ is a maximum-likelihood estimate of the frequency offset. Additionally, under the same AWGN assumption, Schimdl and Cox [21] do an analysis of the performance of the algorithm and show that at high SNR the variance $\sigma^2_{\hat{f}_\Delta}$ of the estimator is proportional to

$$\sigma^2_{\hat{f}_\Delta} \sim \frac{1}{L \cdot SNR} \tag{2.41}$$

Hence the more samples in the sum, the better the quality of the estimator will be, as one would expect.

Post DFT Approach to Frequency Error Estimation

Frequency offset estimation can also be performed after the DFT processing [14]. As with the time domain algorithm, at least two consecutive repeated symbols are required. The preamble structure of WLAN standards has this property during the short and long training symbols, thus allowing the receiver to perform the frequency error estimation also after the DFT processing. Following the analysis in Moose [14], a maximum-likelihood frequency error estimator is derived as follows. The received signal during two repeated symbols is, ignoring noise for convenience,

$$r_n = \frac{1}{N}\left[\sum_{k=-K}^{K} X_k H_k e^{\frac{j2\pi n(k+f_\Delta)}{N}}\right] \qquad n = 0,1,\ldots,2N-1 \tag{2.42}$$

where X_k are the transmitted data symbols, H_k is the channel frequency response for subcarrier k, K is the total number of subcarriers, and f_Δ is the fractional frequency error normalized to the subcarrier spacing. Then calculating the DFT for the first received symbol, the value of kth subcarrier is

$$R_{1,k} = \sum_{n=0}^{N-1} r_n e^{\frac{-j2\pi kn}{N}} \qquad k = 0,1,\ldots,N-1 \tag{2.43}$$

and the DFT of the second symbol is

$$R_{2,k} = \sum_{n=N}^{2N-1} r_n e^{\frac{-j2\pi kn}{N}} \tag{2.44}$$

$$= \sum_{n=0}^{N-1} r_{n+N} e^{\frac{-j2\pi kn}{N}} \qquad k = 0,1,\ldots,N-1 \tag{2.45}$$

Now the crucial observation, from Equation 2.42, is that

$$r_{n+N} = \frac{1}{N}\left[\sum_{k=-K}^{K} X_k H_k e^{\frac{j2\pi(n+N)(k+f_\Delta)}{N}}\right] \tag{2.46}$$

$$= \frac{1}{N}\left[\sum_{k=-K}^{K} X_k H_k e^{\frac{j2\pi n(k+f_\Delta)}{N}} e^{\frac{j2\pi(k+f_\Delta)N}{N}}\right] \tag{2.47}$$

$$= \frac{1}{N}\left[\sum_{k=-K}^{K} X_k H_k e^{\frac{j2\pi n(k+f_\Delta)}{N}} e^{j2\pi(k+f_\Delta)}\right] \tag{2.48}$$

$$= \frac{1}{N}\left[\sum_{k=-K}^{K} X_k H_k e^{\frac{j2\pi n(k+f_\Delta)}{N}}\right] e^{j2\pi f_\Delta} \tag{2.49}$$

$$= r_n e^{j2\pi f_\Delta} \tag{2.50}$$

since $e^{j2\pi k} = 1$. Equation 2.50 implies that

$$R_{2,k} = R_{1,k} e^{j2\pi f_\Delta} \tag{2.51}$$

Thus every subcarrier experiences the same phase shift that is proportional to the frequency offset. Hence the frequency error can be estimated from this phase shift. Using an intermediate variable z again

$$z = \sum_{k=-K}^{K} R_{1,k} R_{2,k}^* \tag{2.52}$$

$$= \sum_{k=-K}^{K} R_{1,k}\left(R_{1k} e^{j2\pi f_\Delta}\right)^* \tag{2.53}$$

$$= e^{-j2\pi f_\Delta} \sum_{k=-K}^{K} R_{1,k} R_{1,k}^* \tag{2.54}$$

$$= e^{-j2\pi f_\Delta} \sum_{k=-K}^{K} |R_{1,k}|^2 \tag{2.55}$$

Thus z is a complex variable, for which the angle is defined by the frequency error. Finally the estimator is

$$\hat{f}_\Delta = -\frac{1}{2\pi} \angle z \tag{2.56}$$

which is quite similar in form to the time domain version.

Properties of the Post DFT Frequency Error Estimation Algorithm

The estimator uses the angle of the complex variable as the basis for the estimate, thus the same $[-\pi, \pi)$ unique range applies as for the time domain estimator. The frequency error f_Δ was defined as a fractional value normalized to the subcarrier spacing, hence the maximum estimable frequency error is again a half of the subcarrier spacing. Otherwise the angle of z will rotate over the $\pm\pi$ limit. This implies that the frequency range is the same for both the time domain and frequency domain estimators.

A somewhat counterintuitive property of this estimator is that the ICI introduced by calculating the DFT of the signal with frequency offset is actually useful information for the estimator. Thus there is no performance penalty compared to the time domain algorithm. In fact, the variance $\sigma^2_{\hat{f}_\Delta}$ is shown [14] to be proportional to

$$\sigma^2_{\hat{f}_\Delta} \sim \frac{1}{L \cdot SNR} \tag{2.57}$$

which is the same result as for the time domain algorithm.

Exercise 3 *The simulation tool implements the time domain frequency error estimation algorithm. Write a Matlab script for the frequency domain algorithm and compare the performance of the algorithms.*

Comments on Frequency Error Estimation Algorithms

Figure 2.4 suggests a two-step frequency estimation process with a coarse frequency estimate performed from the short training symbols and a fine frequency synchronization from the long training symbols. The rationale for this can be explained with the estimation range of the time domain algorithm and the post DFT algorithm. As was shown in Equation 2.38, either estimator could not reliably estimate the frequency error only from the long training symbols. The two-step process would then proceed by first acquiring a coarse estimate of the frequency error from the short training symbols, and then correcting the long training symbols with this estimate. The accuracy of the coarse estimate should easily be better than the 156.25kHz range of the estimator during the long training symbols. Hence a second estimation step could be done from the long training symbols to improve the estimate. Whether the second step is necessary depends on the accuracy of the first estimate. If enough data samples are used to calculate the first estimate from the short symbols, a satisfactory accuracy can usually be reached. Hence the second estimation would be unnecessary.

The main disadvantage of frequency domain estimation is that the DFT has to be calculated for both repeated symbols. Compared to the time domain estimator, the DFT operations mean additional computations without any advantages. Thus the time domain method is preferable for a

WLAN receiver, which in general has very little time to complete all the necessary synchronization functions during the preamble.

Alternative Techniques for Frequency Error Estimation

Hsieh and Wei [8] present a technique to remedy the ±0.5 subcarrier estimation range. The idea is to take advantage of the correlation of the channel frequency response between adjacent subcarriers. The algorithm first uses training data to estimate the channel frequency response. The next step is to calculate the autocorrelation function of the channel estimate. Since the frequency offset shifts the received signal in frequency, the autocorrelation function will have its maximum at the lag corresponding to the frequency offset. The authors show that in an OFDM system with 1,024 subcarriers frequency offsets up to ±200 subcarrier spacings can be estimated, which is a significant improvement from the ±0.5 subcarrier spacing limit.

Traditional single carrier receivers usually employ some kind of phase locked loop to track the carrier phase. The main difficulty in using phase locked loops for OFDM carrier phase tracking is the nature of the OFDM waveform. It is not easy to design a phase error detector, a main component of a phase locked loop, for an OFDM system. A carrier frequency tracking technique for OFDM that resembles single carrier frequency recovery loop has been proposed [13]. The proposed method is composed of two steps: acquisition and tracking. The acquisition step is performed from known synchronization symbols. Tracking mode is a frequency control loop that has been modified for OFDM waveform. The main advantage of this method is its quite large acquisition range, compared to the two estimators presented above. This advantage is significant for broadcast systems like DVB and DAB, which place subcarriers very close to each other; for example, the minimum subcarrier spacing for DAB is 1 kHz. Thus a half subcarrier spacing acquisition range of the estimator practically renders the presented time domain and post DFT methods useless. In the paper, the crucial phase error detection is performed by detecting and remodulating the received data, and then comparing the remodulated signal to the received signal. A similar technique is also investigated in Daffara and Chouly [4]. The structure of the phase error detector of the previous two papers is simplified [5] by exploiting the cyclic prefix to detect the phase error.

Carrier Phase Tracking

Frequency estimation is not a perfect process, so there is always some residual frequency error. The SNR loss due to the ICI generated should not be a problem if the estimator has been designed to reduce the frequency error below the limit required for a negligible performance loss for the used modulation. The main problem of the residual frequency offset is constellation rotation. The analysis of the post DFT frequency error estimator also shows that the constellation rotation is the same for all subcarriers. To illustrate the effect in an IEEE 802.11a system, Figure 2.14 shows how much a QPSK constellation rotates during 10 OFDM symbols with a 3kHz frequency error.

This error corresponds to only 1% of the subcarrier spacing, thus the effect on SNR is negligible as shown earlier. The figure shows that after only 10 symbols, the constellation points have just rotated over the decision boundaries shown as solid lines, thus correct demodulation is no longer possible.

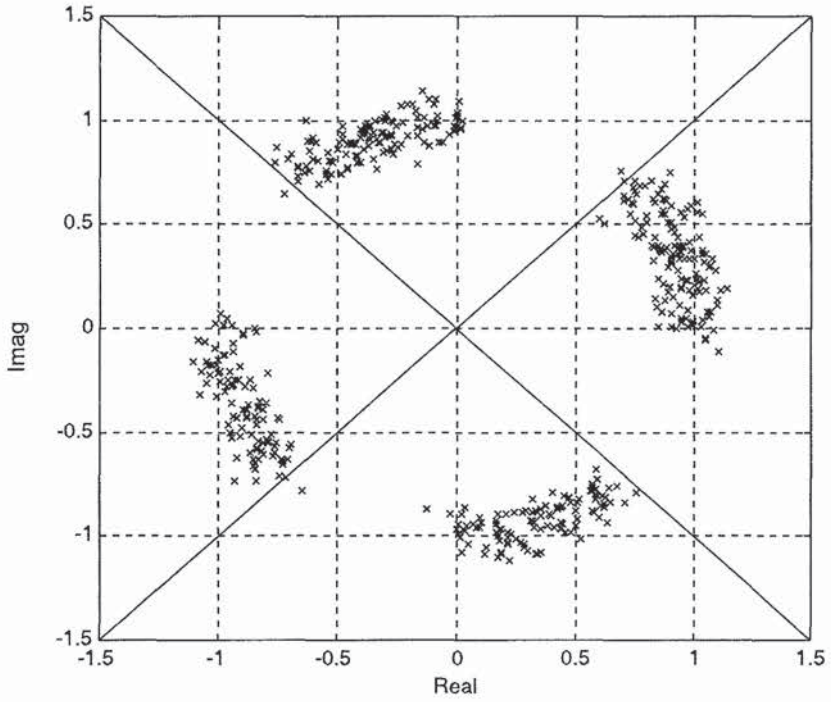


FIGURE 2.14
Constellation rotation with 3kHz frequency error during 10 symbols.

This effect forces the receiver to track the carrier phase while data symbols are received.

Data-Aided Carrier Phase Tracking

The simplest method is data-aided tracking of the carrier phase. IEEE 802.11a and HiperLAN/2 include four predefined subcarriers among the transmitted data. These special subcarriers are referred to as *pilot* subcarriers. The main purpose of these pilots is exactly to help the receiver to track the carrier phase. After the DFT of the nth received symbol, the pilot subcarriers $R_{n,k}$ are equal to the product of the channel frequency response H_k and the known pilot symbol $P_{n,k}$, rotated by the residual frequency error.

$$R_{nk} = H_k P_{nk} e^{j2\pi n f_\Delta} \tag{2.58}$$

Assuming an estimate $\hat{H}_k$ of the channel frequency response is available, the phase estimate is

$$\hat{\Phi}_n = \angle\left[\sum_{k=1}^{N_p} R_{n,k}\left(\hat{H}_k P_{n,k}\right)^*\right] \tag{2.59}$$

$$= \angle\left[\sum_{k=1}^{N_p} H_k P_{n,k} e^{j2\pi n f_\Delta}\left(\hat{H}_k P_{n,k}\right)^*\right] \tag{2.60}$$

If we assume that the channel estimate is perfectly accurate, we get the estimator

$$\hat{\Phi}_n = \angle\left[\sum_{k=1}^{N_p} |H_k|^2 |P_{n,k}|^2 e^{j2\pi n f_\Delta}\right] \tag{2.61}$$

$$= \angle\left[e^{j2\pi n f_\Delta} \sum_{k=1}^{N_p} |H_k|^2\right] \tag{2.62}$$

In this case we need not worry about the $[-\pi, \pi)$ phase range, because the pilot data are known, thus the phase ambiguity is automatically resolved correctly. Note that in practice the channel estimates are not perfectly accurate, thus they contribute to the noise in the estimate.

Nondata-Aided Carrier Phase Tracking

The phase error can be estimated without the help of the pilot symbols. Recall that the phase error resulting from the frequency offset is identical for all the subcarriers; this property (identical frequency offset) allows the development of a nondata-aided estimator for the phase error. Figure 2.15 shows how the data subcarriers have rotated for a single BPSK modulated OFDM symbol after the channel effect has been corrected. The angle Φ shown in the figure is the result of the frequency offset.

The angle Φ can be estimated without any knowledge of the data by performing hard decisions $\hat{X}_{n,k}$ on the received data symbols $R_{n,k} = H_k X_{nk} e^{j2\pi n f_\Delta}$ after they are corrected for the channel effect. Then the angle between a hard decision and the corresponding received symbol is used as an estimator.

$$\hat{\Phi} = \angle\left[\sum_{k=-K}^{K} R_{n,k}\left(\hat{X}_k \hat{H}_k\right)^*\right] \tag{2.63}$$

$$= \angle\left[\sum_{k=-K}^{K} H_k \hat{H}_k^* X_{n,k} \hat{X}_{n,k}^* e^{j2\pi n f_\Delta}\right] \tag{2.64}$$

$$= \angle\left[\sum_{k=-K}^{K} |H_k|^2 |X_{n,k}|^2 e^{j2\pi n f_\Delta}\right] \tag{2.65}$$

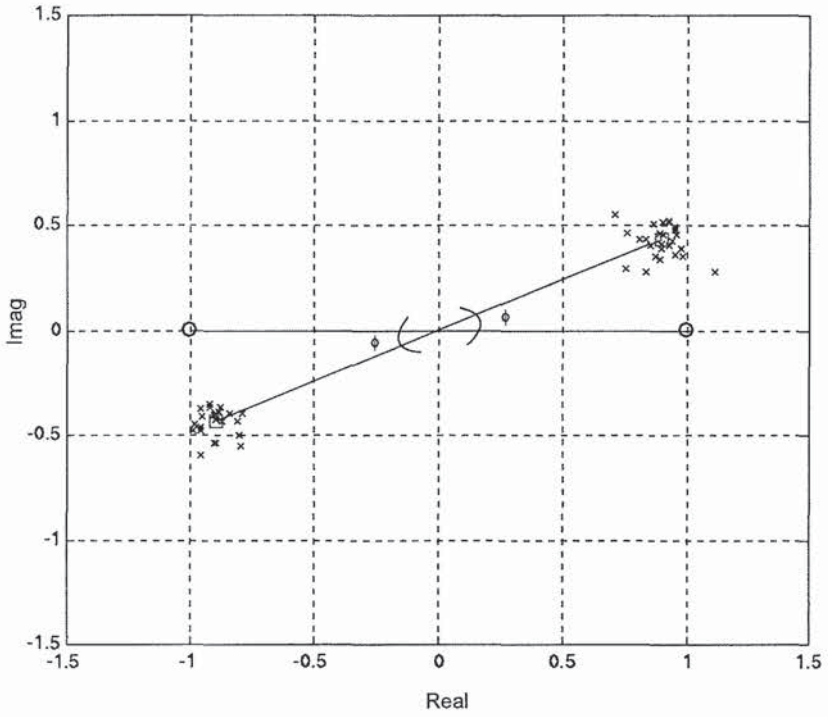


FIGURE 2.15
BPSK single symbol rotation.

In Equations 2.63-2.65, we have assumed that all the hard decisions are correct and that the channel estimates are perfect. In reality, neither condition is true, but if the number of data carriers is large the effect of incorrect hard decisions will not have a significant impact.

It should be noted that the total rotation angle increases from symbol to symbol, hence it is necessary to compensate this angle before the hard decisions are performed. Without the compensation the rotation angle will eventually cross the decision boundaries of the constellation, and then all the hard decisions will be incorrect. As a result, the phase estimate will also be worthless. This sounds like a circular argument: How can the angle be compensated before it has been estimated? A simple tactic is to keep track of the total rotation up to the previous symbol, and correct the data with this value, before the hard decisions are performed. The rotation from symbol to symbol is quite small, thus the probability of incorrect hard decisions is not increased significantly. After the angle has been estimated, the total rotation angle is updated for the next symbol.

Exercise 4 *The simulation tool implements the data-aided carrier phase tracking algorithm. Write a Matlab script for the nondata-aided algorithm and compare the performance of the algorithms. Try to implement a method to combine both algorithms to achieve improved performance.*

Channel Estimation

Channel estimation is the task of estimating the frequency response of the radio channel the transmitted signal travels before reaching the receiver antenna. The impulse response of a time varying radio channel is usually [20] represented as a discrete time FIR filter

$$h(\tau;t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t)) \tag{2.66}$$

WLAN applications generally assume that the channel is *quasistationary*, that is, the channel does not change during the data packet. With this assumption the time dependency in Equation 2.66 can be dropped

$$h(\tau) = \sum_n \alpha_n e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n) \tag{2.67}$$

Then the discrete time frequency response of the channel is the Fourier transform of the channel impulse response

$$H_k = DFT\{h_n\} \tag{2.68}$$

Hence the channel estimation process outputs the $\hat{H}_k$, an estimate of the channel frequency response for each subcarrier.

Channel estimation is mandatory for OFDM systems that employ coherent modulation schemes. Otherwise correct demodulation would not be possible. Knowledge of the channel can also improve the performance OFDM with noncoherent modulation schemes, although the $\hat{H}_k$ are not needed for demodulation in this case. The improvement can be achieved when an error control code is used in the system, in which case the channel knowledge can help the code decoder do a better job.

Frequency Domain Approach for Channel Estimation

Frequency domain channel estimation can be performed with two main approaches. The first method uses training data transmitted on every subcarrier. This method is applicaple to WLAN systems. The second method uses correlation properties of the frequency response of a multipath channel, and training information that is transmitted on a subset of the subcarriers. This approach is used in broadcast OFDM systems.

Channel Estimation Using Training Data

The long training symbols in the WLAN preamble, Figure 2.4, facilitate an easy and efficient estimate of the channel frequency response for all the subcarriers. The contents of the two long training symbols are identical, so averaging them can be used to improve to quality of the

channel estimate. DFT is a linear operation, hence the average can be calculated before the DFT. Then only one DFT operation is needed to calculate the channel estimate. After the DFT processing, the received training symbols $R_{1,k}$ and $R_{2,k}$ are a product of the training symbols X_k and the channel H_k plus additive noise $W_{l,k}$.

$$R_{l,k} = H_k X_k + W_{l,k} \tag{2.69}$$

Thus the channel estimate can be calculated as

$$\hat{H}_k = \frac{1}{2}(R_{l,k} + R_{2,k})X_k^* \tag{2.70}$$

$$= \frac{1}{2}(H_k X_k + W_{1,k} + H_k X_k + W_{2,k})X_k^* \tag{2.71}$$

$$= H_k|X_k|^2 + \frac{1}{2}(W_{1,k} + W_{2,k})X_k^* \tag{2.72}$$

$$= H_k + \frac{1}{2}(W_{1,k} + W_{2,k})X_k^* \tag{2.73}$$

where the training data amplitudes have been selected to be equal to one. The noise samples $W_{1,k}$ and $W_{2,k}$ are statistically independent, thus the variance of their sum divided by two is a half of the variance of the individual noise samples.

Channel Estimation Using Pilot Subcarriers and Interpolation

If the subcarriers are very close to each other, like in broadcast standards (e.g. DVB) the frequency domain correlation of the channel frequency response can be used to estimate the channel, even if training data is not available for all the subcarriers. The *coherence bandwidth* of the channel is usually approximated [20] as the inverse of the total length of the channel impulse response

$$(\Delta f) \approx \frac{1}{\tau_{\max}} \tag{2.74}$$

When the subcarrier spacing is much less than the coherence bandwidth, the channel frequency response on neighboring subcarriers will be practically identical. For example, the DVB standard takes advantage of this by not sending training information on all subcarriers, hence more data can be transmitted. The pilot subcarriers in DVB are divided into two sets. The first set is regular pilot subcarriers that are similar to the WLAN pilots subcarriers; known data is constantly transmitted on them. The second set is called *scattered* pilots; known data is transmitted only intermittently transmitted on the subcarrier. Then the channel estimation is performed by interpolating the frequency response of the subcarriers that do not have training information from the known

subcarriers. We do not describe this algorithm in detail, a description of a method can be found in Speth et. al. [25].

Time Domain Approach for Channel Estimation

The channel estimation can also be performed using the time domain approach, before DFT processing of the training symbols. In this case, the channel impulse response, instead of the channel frequency response, is estimated. The following derivation of the channel estimator uses IEEE 802.11a standard training symbols as an example, but the same approach can be applied to any OFDM system that includes training symbols. The received time domain signal during the two long training symbols is

$$r_{1,n} = h * x_n + w_{1,n} \tag{2.75}$$

The time domain convolution can be expressed as a matrix vector multiplication. The circular convolution matrix is formed from the training data as

$$X = \begin{bmatrix} x_1 & x_{64} & x_{63} & \cdots & x_{64-L+2} \\ x_2 & x_1 & x_{64} & & x_{64-L+3} \\ \vdots & & \vdots & & \vdots \\ x_{63} & x_{62} & & & x_{64-L} \\ x_{64} & x_{63} & & \cdots & x_{64-L+1} \end{bmatrix} \tag{2.76}$$

The parameter L defines the maximum length of the impulse response that can be estimated, and X is in general a rectangular matrix. The channel impulse response vector is

$$h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_L \end{bmatrix} \tag{2.77}$$

Then the convolution is expressed as

$$r_{l,n} = h * x_n + w_{l,n} = Xh + w_l \tag{2.78}$$

The channel impulse response estimate can then be formed by

$$\hat{h} = \frac{1}{2} X^{\dagger}(r_{1,n} + r_{2,n}) \tag{2.79}$$

$$= \frac{1}{2} X^{\dagger}(Xh + w_1 + Xh + w_2) \tag{2.80}$$

$$= X^{\dagger}Xh + \frac{1}{2} X^{\dagger}(w_1 + w_2) \tag{2.81}$$

$$= h + \frac{1}{2} X^{\dagger}(w_1 + w_2) \tag{2.82}$$

where $X^{\dagger}$ denotes Moore-Penrose [22] generalized inverse of X. The channel frequency response estimate is then formed by calculating the DFT of the impulse response estimate

$$\hat{H} = DFT\{\hat{h}\} \tag{2.83}$$

Exercise 5 *The simulation tool implements the frequency domain channel estimation algorithm. Write a Matlab script for the time-domain algorithm and compare the performance of the algorithms.*

Exercise 6 *The simulation tool allows to run simulations with perfect synchronization or with partial or full synchronization. Experiment how the performance of the system degrades compared to perfect synchronization as the various synchronization algorithms are activated.*

Analysis of the Time Domain and Frequency Domain Approaches for Channel Estimation

The advantage of the time domain approach is improved performance, when the maximum length of the impulse response can be limited to a number significantly less than the number of subcarriers. The rationale is that the frequency domain estimator has to simultaneously estimate all the subcarriers, whereas the time domain estimator needs to estimate only the taps of the impulse response. When the number of subcarriers is large compared to the number of channel taps, the signal energy used to estimate each H_k is significantly less than the signal energy used to estimate each h_n. In other words, it is easier to estimate fewer parameters given a fixed amount of data. For example, in the IEEE 802.11a system the number of subcarriers is 52, and the maximum length of the channel can be assumed to be less than the cyclic prefix length of 16 samples. Thus the frequency domain algorithm estimates more than three times the number of parameters than the time domain algorithm.

The drawback of the time domain method is that additional computations are required. The $\frac{1}{2}X^{\dagger}(r_{1,n}+r_{2,n})$ operation requires $64 \cdot L$ multiplications, which are not needed at all in the frequency domain estimator. This is the usual engineering trade-off; better performance usually implies higher costs in one form or another.

Enhancing the Channel Estimate

The correlation of the channel frequency response between different subcarriers can be used to further improve the quality of the channel estimate. Figure 2.16 shows an example of the channel amplitude response for a IEEE802.11a system. It can be seen that the neighboring subcarriers are highly correlated. We will not go to the details of this method, but refer to [12] for a full discussion. The paper shows that in some conditions performance very close to ideal channel estimation can be achieved. Unfortunately, this improvement does not come without a price; significantly more computations are required than to calculate the simple frequency domain channel estimator.

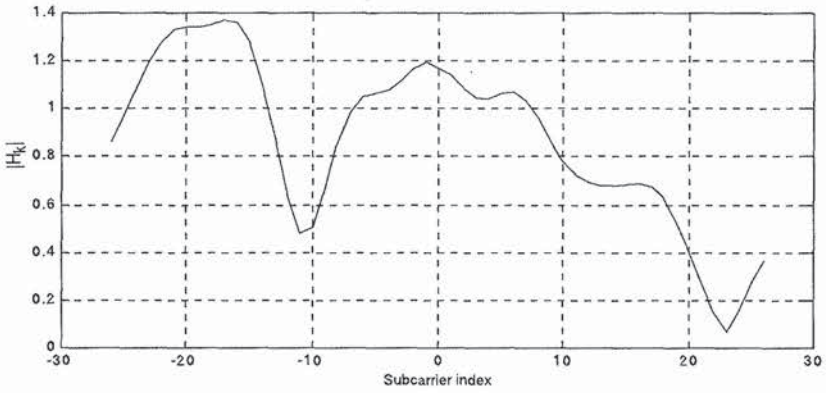


FIGURE 2.16
Example of a channel amplitude response.

Clear Channel Assessment

Clear Channel Assessment (CCA) is not strictly an OFDM synchronization algorithm, but a network synchronization method. However, it is closely related to packet detection problem and thus it is included in this chapter. The overall system performance of IEEE 802.11 WLAN is crucially dependent on high quality CCA algorithm, so the algorithm has to be carefully designed. The IEEE 802.11a standard has two different requirements for the CCA algorithm. The first is the detection probability when a preamble is available and the second is the detection probability when the preamble is not available. The requirement for the first case is that the CCA algorithm shall indicate a *Busy* channel with >90% probability within 4μs observation window, if a signal is

received at –82 dBm level. For the second case, when the known preamble structure is not available, the requirement is relaxed by 20dB. Thus a signal detection probability of >90% within $4\mu s$ observation for a received signal level of –62dBm is required.

The first requirement can be approached using any of the packet detection algorithms presented in this chapter. The methods that use the known preamble structure are capable of reaching the performance level required in the standard. The only possible approach for the second requirement is to use the received signal energy detection method. The reason is the maximum allowed length of $4\mu s$ for the observation time. This is equal to the symbol length in the IEEE802.11a system. Thus, during the $4\mu s$ time, the receiver might receive one whole OFDM symbol or the end of one symbol and the start of the next one. In this case, there is no available signal structure that could be taken advantage of to improve the detection probability. The OFDM waveform itself resembles white noise, and inside $4\mu s$ windows it cannot be guaranteed that a whole OFDM symbol is received, which would allow to use the cyclic prefix in the CCA algorithm. As a conclusion, the receiver can only measure the total received energy and test whether it is above the specified limit.

Signal Quality

Signal quality estimation is not strictly a synchronization algorithm, but a network performance issue. WLAN systems define several data rates that can be used to transmit data; thus, to achieve the best overall network performance, data should be transmitted on the highest rate that can be reliably transmitted under the current channel conditions. For the transmitter to be able to decide what data rate to use, an accurate estimate of the quality of the channel has to be available. This is the task of the signal quality algorithm.

The double sliding window algorithm showed how its output could be used to estimate the received *SNR* in Equation 2.7. This is a quite simple method if the receiver implements the double sliding window detector. However, the form of the estimator does not take into account fading channel conditions. Only the total received signal energy is considered. In some cases, this behavior can lead to an overly optimistic estimate of the signal quality. For example, the total channel impulse response energy may be quite high, indicating high SNR, but the frequency response of the channel can have very deep nulls. In this case, the actual quality of the channel could be significantly less than the estimate given by double sliding window algorithm.

A quite different approach to the signal quality problem can be taken by using the convolutional error control code specified in the IEEE802.11a system. The convolutional error correcting code used in IEEE 802.11a standard is described in Chapter 3, "Modulation and Coding." The standard method to perform convolutional code decoding is the Viterbi algorithm that is described in Chapter 1, and in more detail in [20]. We will not describe the algorithm, but assume knowledge of its basic principles. Essentially the Viterbi algorithm calculates the distance between the

maximum-likelihood transmitted code word and the received data. This distance can be used as an indication of the quality of the channel. When the channel is in good condition, the distance between the transmitted and received signal is small; this is indicated by the Viterbi algorithm as a small cumulative minimum distance at the end of the decoding process. When the channel is in a bad state, the distance between the transmitted and received signal is larger, and the cumulative minimum distance will be large. Figure 2.17 shows the probability density functions of the cumulative minimum distance for several different *SNRs*. The *SNR* estimator is formed by dividing the cumulative minimum distance into ranges according to the value of the PDFs. The *SNR* value corresponding to the PDF that has the largest value at a given cumulative minimum distance is the estimate of the signal quality. For example, a cumulative minimum distance between 2.2 and 2.9 indicates a 13dB SNR. From Figure 2.17 it can be seen that for a large range of received *SNRs* a very good accuracy of 1dB resolution can be achieved.

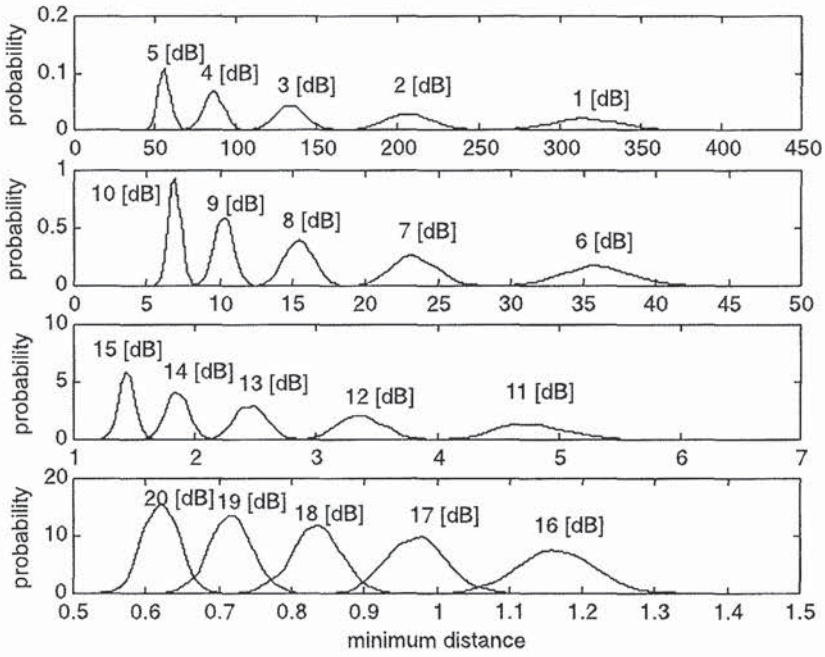


FIGURE 2.17

Cumulative minimum distance in Viterbi decoding.

Bibliography

[1] J.-J. van de Beek, M. Sandell, M. Isaksson and P.O. Borjesson, "Low-complex frame synchronization in OFDM systems," in Proceedings of IEEE Internatial Conference Universal Personal Communications, Toronto, Canada, Sept. 27–29, 1995, pp. 982–986

[2] J-J. van de Beek, M. Sandell, P. O. Börjesson, "ML Estimation of Time and Frequency Offset in OFDM systems," IEEE Transactions on Signal Processing, Vol. 45, No. 7, July 1997

[3] G. Casella, R. Berger, "Statistical Inference," Duxbury Press, California, 1990

[4] F. Daffara, A. Chouly, "Maximum Likelihood Frequency Detectors for Orthogonal Multicarrier Systems," Proceedings of IEEE International Conference on Communications, Geneva, Switzerland, May 22–26, 1993, pp.766–771

[5] F. Daffara, O. Adami, "A New Frequency Detector for Orthogonal Multicarrier Transmission Techniques," Proceedings of IEEE Vehicular Technology Conference, Chicago, IL, July 25–28, 1995, pp.804–809

[6] S. A. Fechtel, "OFDM Carrier and Sampling Frequency Synchronization and its Performance on Stationary and Mobile Channels," IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, August 2000, pp. 438–441

[7] HiperLAN/2, "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer," ETSI TS 101 475 V1.2.1 (2000–11)

[8] Meng-Han Hsieh and Che-Ho Wei, "A Low-Complexity Frame Synchronization and Frequency Offset Compesation Scheme for OFDM Systems over Fading Channels," in IEEE Transactions on Vehicular Technology, Vol. 48, No. 5, September 1999

[9] IEEE 802.11a, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHZ Band," IEEE Std 802.11a-1999

[10] Steven M. Kay, "Fundamentals of Statistical Signal Processing: Detection Theory," Prentice Hall, New Jersey, 1998

[11] D. K. Kim, S.H. Do, H. B. Cho, H. J. Choi, K. B. Kim, "A New Joint Algorithm of Symbol Timing Recovery and Sampling Clock Adjustment for OFDM Systems," IEEE Transactions on Consumer Electronics, Vol. 44, No. 3, August 1998, pp. 1142–1149

[12] Y. Li, L. J. Cimini, N. R. Sollenberger, "Robust Channel Estimation for OFDM Systems with Rapid Dispersive Fading Channels," IEEE Transactions on Communications, Vol. 46, No. 7, July 1998

[13] M. Luise, R. Reggiannini, "Carrier Frequency Acquisition and Tracking for OFDM Systems," IEEE Transactions on Communications, Vol. 44, No. 11, November 1996

[14] P. H. Moose, "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction," IEEE Transactions on Communications, Vol. 42, No. 10, October 1994

[15] R. van Nee and Ramjee Prasad, "OFDM for Wireless Multimedia Communications," Artech House Publishers, 2000

[16] M. Okada, S. Hara, S. Komaki, N. Morinaga, "Optimum Synchronization of Orthogonal Multi-Carrier Modulated Signals," Proceedings IEEE PIMRC, Taipei, Taiwan, Oct. 15–18, 1996, pp. 863–867

[17] T. Pollet, M. Peeters, "Synchronization with DMT Modulation," IEEE Communications Magazine, April 1999, pp.80–86

[18] T. Pollet, M. van Bladel, M. Moeneclaey, "BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Wiener Phase Noise," IEEE Transactions on Communications, Vol. 43, Issue 2, Part 3, February, March, April 1995, pp. 191–193

[19] T. Pollet, P. Spruyt, M. Moeneclaey, "The BER Performance of OFDM Systems using Non-Synchronized Sampling," IEEE Global Telecommunications Conference 1994, pp. 253–257

[20] J. G. Proakis, "Digital Communications," McGraw-Hill, Boston, 3ed., 1995

[21] T. M. Schmidl, D.C. Cox, "Low-Overhead, Low-Complexity [Burst] Synchronization for OFDM," IEEE International Conference on Communications, Vol. 3., 1996, pp 1301–1306

[22] S. R. Searle, "Matrix Algebra Useful for Statistics," John Wiley & Sons, New York 1982

[23] M. Speth, D. Daecke, H. Meyr, "Minimum Overhead Burst Synchronization for OFDM based Broadband Transmission," IEEE Global Telecommunications Conference, Vol. 5, 1998, pp. 2777–2782

[24] M. Speth, S. A. Fechtel, G. Fock, H. Meyr, "Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM-Part I," IEEE Transactions on Communications, Vol. 47, No. 11, November 1999, pp. 1668–1677

[25] M. Speth, S. Fechtel, G. Fock, H. Meyr, "Optimum Receiver Design for OFDM-Based Broadband Transmission-Part II: A Case Study," IEEE Transactions on Communications, Vol. 49, No. 4, April 2001, pp. 571–578

Modulation and Coding

CHAPTER

3

IN THIS CHAPTER

- Modulation 88
- Interleaving 102
- Channel Codes 106
- Bibliography 120

Modulation and channel coding are fundamental components of a digital communication system. Modulation is the process of mapping the digital information to analog form so it can be transmitted over the channel. Consequently every digital communication system has a *modulator* that performs this task. Closely related to modulation is the inverse process, called *demodulation*, done by the receiver to recover the transmitted digital information. The design of optimal demodulators is called *detection* theory. An OFDM system performs modulation and demodulation for each subcarrier separately, although usually in a serial fashion, to reduce complexity.

Channel coding, although not strictly speaking mandatory for digital communication, is essential for good performance. After Claude Shannon published his groundbreaking paper [16] that started *information theory* as a scientific discipline and established the fundamental law equation 3.1 of the capacity C of an AWGN communication channel with bandwidth W and signal to noise ratio SNR, communication scientists and engineers have been searching for methods on how to reach the capacity of the channel.

$$C = W \log_2(1 + SNR) \tag{3.1}$$

Shannon's proof of the channel capacity formula was nonconstructive, and thus did not provide a method to actually reach the capacity C. As a consequence, *coding theory* got started as a discipline that research methods to design a communication system that has performance close to the fundamental limit C. The methods developed under coding theory have different names; channel coding, forward error correction (FEC), and error control coding are the most usual. Channel coding and basic types of FEC codes were introduced in "Channel Coding" in Chapter 1. Note that channel coding and source coding, introduced in the "Source Coding" section in Chapter 1, although related, are quite different topics.

Our goal in this chapter is to discuss modulation, detection, channel coding and interleaving from an OFDM system of point of view. Channel coding is such an important component of OFDM systems that the term Coded Orthogonal Frequency Division Multiplexing (COFDM) is sometimes used. The reasons for this emphasis on channel coding for OFDM systems will become apparent in the channel coding section of this chapter. Modulation and demodulation or detection are discussed in depth in several books [11, 17, 18]. Channel coding is the topic of a wide variety of books [10, 20] and also books that concentrate on specific types of FEC codes [7, 9].

Modulation

As was discussed in "Modulation" in Chapter 1, modulation can be done by changing the amplitude, phase or frequency of transmitted Radio Frequency (RF) signal. For an OFDM system, the first two methods can be used. Frequency modulation cannot be used because subcarriers are orthogonal in frequency and carry independent information. Modulating subcarrier frequencies would destroy the orthogonality property of the subcarriers; this makes frequency modulation unusable for OFDM systems.

The main design issue of the modulator is the *constellation* used. Constellation is the set of points that can be transmitted on a single symbol. The used constellation affects several important properties of a communication system; for example, bit error rate (BER), peak to average power ratio (PAPR), and RF spectrum shape. The single most important parameter of a constellation is *minimum distance*. Minimum distance ($d_{\min}$) is the smallest distance between any two points in the constellation. Therefore $d_{\min}$ determines the least amount of noise that is needed to generate a decision error. Actual BER or P_b of a constellation can in many cases be calculated using the Q-function defined in Equation 3.2. The value of the Q-function is equal to the area under the tail of the Probability Density Function (PDF) of a zero mean and unit variance normal random variable.

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \quad x \geq 0 \tag{3.2}$$

P_b is a function of energy per bit to noise ratio $\frac{E_b}{N_o}$ or signal energy to noise ratio $\frac{E_s}{N_o}$; these ratios are related by a simple scaling factor that depends on the number of bits k transmitted per symbol $E_s = kE_b$. The usefulness of $\frac{E_b}{N_o}$ as a measure of the signal quality will became apparent when different coding schemes are discussed in "Channel Codes" later in this chapter. Then a general form of P_b equations for several important constellations in terms of the Q-function is shown in Equation 3.3.

$$P_b \sim Q\left(\sqrt{\frac{E_b}{N_o}}\right) \tag{3.3}$$

The value of the Q-function gets smaller for larger arguments, hence Equation 3.3 shows that large $\frac{E_b}{N_o}$ implies better performance. Different constellations have different $d_{\min}$ values for the same $\frac{E_b}{N_o}$. The constellation with the largest $d_{\min}$ for a given $\frac{E_b}{N_o}$ has the best performance. In P_b equations like 3.3, this effect results in scaling of the value and the argument of the Q-function.

Minimum distance depends on several factors: number of points M, average power P_{ave} and shape of the constellation. The most important is the number of points in the constellation, which is directly dependent on the number of bits k transmitted in one symbol $M = 2^k$. Average power P_{ave} scales the constellation smaller or larger, depending on the transmitted power. Consequently, to make comparisons between different constellations fair, P_{ave} is usually normalized to one when $d_{\min}$ of a constellation is calculated. Average power of a constellation is evaluated simply by averaging the power of all the M points c_k of the constellation as shown in Equation 3.4.

$$P_{ave} = \frac{1}{M} \sum_{k=1}^{M} |c_k|^2 \tag{3.4}$$

Another important factor that influences in $d_{\min}$ is the shape of the constellation. A constellation is basically a set of points located in one dimensional or larger space spanned by the transmitted symbols. Traditional constellations are either one or two dimensional. In principle, these points can be located in any shape whatsoever. Naturally, for communications purposes care must taken to ensure good performance and practical implementation. There are two main goals when constellation shapes are designed. The first is constant amplitude of the transmitted signal. In this case, all the points in the constellation are required to have equal amplitude. The second goal is to improve $d_{\min}$. This is an optimization problem that tries to place all the points such that $d_{\min}$ is maximized for a finite P_{ave}. For the usual case of a two dimensional constellation with large number of points, this results in a circular constellation. Forney and Wei [4] give a through analysis of the behavior of $d_{\min}$ for different constellation types.

Coherent Modulations

Coherent modulation can be used by a communication system that maintains a phase lock between the transmitter and receiver RF carrier waves. Coherent modulation improves performance, but requires more complex receiver structure compared to *non-coherent* systems that are discussed later in this chapter. The performance gain of coherent modulation is significant when the system uses large constellations. High speed communication systems, like IEEE 802.11a, are usually coherent. The following sections describe the most common coherent modulations and their performance.

Amplitude Shift Keying

Amplitude Shift Keying (ASK) modulation transmits information by changing the amplitude of the carrier. Equation 3.5 shows the carrier waveform of ASK modulation. The A_k term does the actual modulation by multiplying the carrier wave by the selected amplitude level. The transmitted bits determine which of the possible symbols $\{A_1, \ldots, A_M\}$ is selected.

$$s(t) = A_k \cos(\omega_c t) \tag{3.5}$$

Figure 3.1 shows several different ASK modulations; (a) is a 2-ASK or BPSK, (b) is a 4-ASK, and (c) is a 8-ASK. In the figure, all the modulations have the same minimum distance $d^2_{\min} = 4$, however the average power of the constellations is not equal. To perform fair comparison of the ASK constellations, Table 3.1 shows the normalized $d^2_{\min}$ and the required increase in *SNR* to maintain the same BER when changing to a one-step-larger constellation. The *SNR* increase converges to about 6 dB for each additional bit in the constellation; for this reason, large ASK modulations are rarely used in practice. The 2-ASK modulation does not have an *SNR* increase value, because it is the smallest possible constellation. The IEEE 802.11a system uses only the smallest 2-ASK modulation; for larger constellations, Phase Shift Keying and Quadrature Amplitude Modulations are used in practice. These two methods are the topics of the next sections.

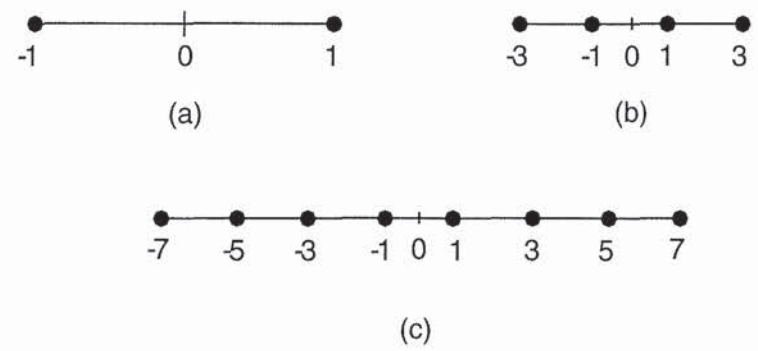


FIGURE 3.1
ASK modulations, (a) 2-ASK, (b) 4-ASK, (c) 8-ASK.

TABLE 3.1 Distance Properties of ASK Modulations

Modulation	P_{ave}	$d^2_{\min}$ *Normalized*	*SNR Increase*
2-ASK	1	4	–
4-ASK	5	$\frac{4}{5}$	6.99 dB
8-ASK	21	$\frac{4}{21}$	6.23 dB

Bit error rate P_b of binary ASK is the most basic error probability (see Equation 3.6). The P_b of 2-ASK is plotted in Figure 3.2. It is useful to memorize a reference point from this curve, like $P_b = 2 \cdot 10^{-4}$ at 8.0 dB $\frac{E_b}{N_o}$ or $P_b = 10^{-5}$ at 9.6 dB $\frac{E_b}{N_o}$. These points can serve as a quick check to validate simulation results. As the number of points in the constellation increases, exact analytic P_b equations become quite complicated, therefore either formulas for symbol error rate P_s or approximations to P_b are more commonly used.

$$P_b = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{3.6}$$

Symbol error rate P_s of M-ary ASK modulations is equal to Equation 3.7. Bit error rate P_b can be approximated by dividing P_s by the number of bits per symbol as in Equation 3.8. The formula for P_b is an approximation, because generally the number of bit errors that occur for each symbol error can be more than one. Equation 3.8 assumes that each symbol error causes only one bit error. However, with the help of Gray coding, which is discussed in the section "Labeling Constellation Points," for most symbol errors the number of bit errors is equal to one. The quality of this approximation improves at high signal to noise ratios.

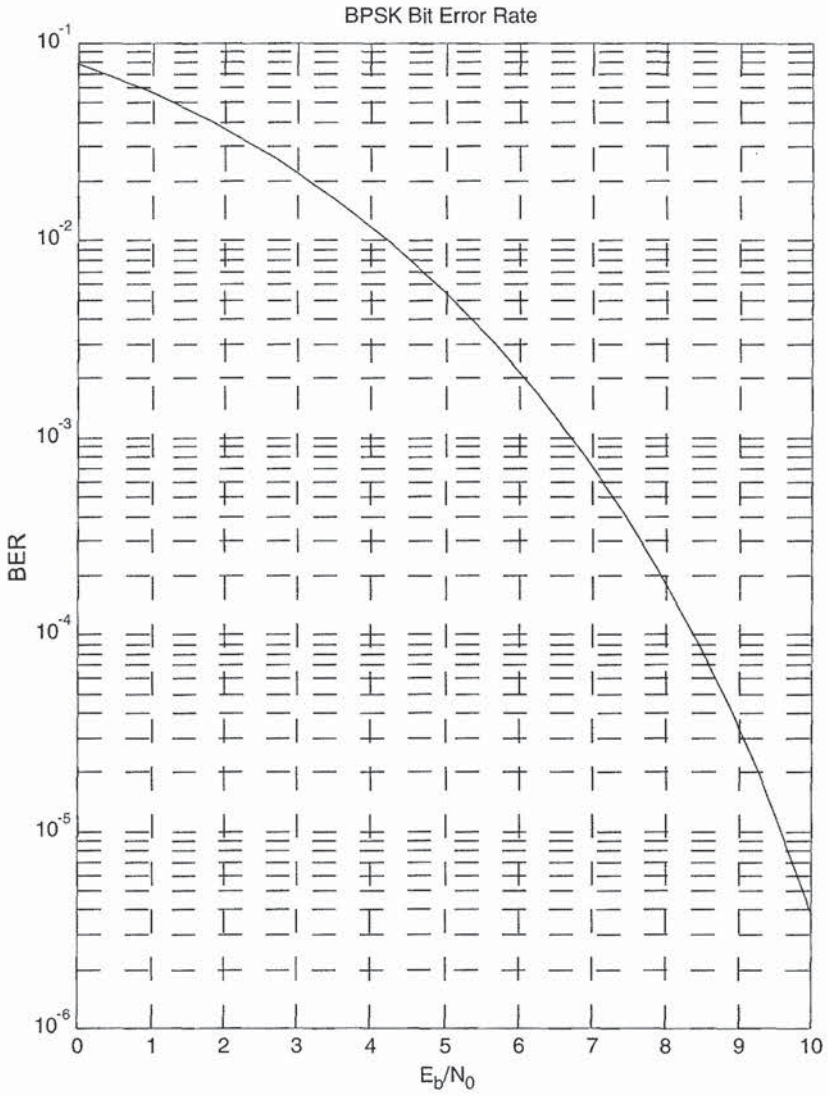


FIGURE 3.2
BPSK bit error rate.

$$P_s = 2\frac{M-1}{M}Q\left(\sqrt{\frac{A^2}{2N_0}}\right) \tag{3.7}$$

In Equation 3.7, A is the amplitude difference between symbol levels. For example, in the constellation in Figure 3.1, $A = 2$.

$$P_b \approx \frac{P_s}{\log_2 M} = \frac{P_s}{k} \tag{3.8}$$

Phase Shift Keying

Phase Shift Keying (PSK) modulations transmit information by changing the phase of the carrier, the amplitude is kept constant; hence PSK modulations are also called *constant amplitude modulations*. Equation 3.9 shows the carrier waveform of PSK signal. Modulation is done by the ϕ_k term.

$$s(t) = \cos(\omega_c t + \phi_k) \tag{3.9}$$

The main benefit of constant amplitude modulation is the peak-to-average power ratio (PAPR) that is practically equal to one. However, this is only true for a single carrier systems. An OFDM signal that is a sum of several modulated subcarriers does not have a constant amplitude, even if the individual subcarriers do. The main benefit of constant amplitude modulation is simplified RF chain design for both the transmitter and receiver. An OFDM signal can never have a constant amplitude, hence having a constant amplitude constellation does *not* benefit OFDM systems. Several techniques to mitigate the large amplitude variations of OFDM signals are discussed in Chapter 5.

Figure 3.3 shows the three smallest PSK constellations, (a) is 2-PSK or binary PSK (BPSK), (b) is 4-PSK or quadrature PSK (QPSK), and (c) is 8-PSK. Note that BPSK is identical to 2-ASK. The figure shows how PSK modulations use both the *Inphase (I)* and *Quadrature (Q)* carrier waves, hence the modulation is two dimensional. This increase in dimensionality of the constellation improves the behavior of $d^2_{\min}$ as the number of points in the constellation increases.

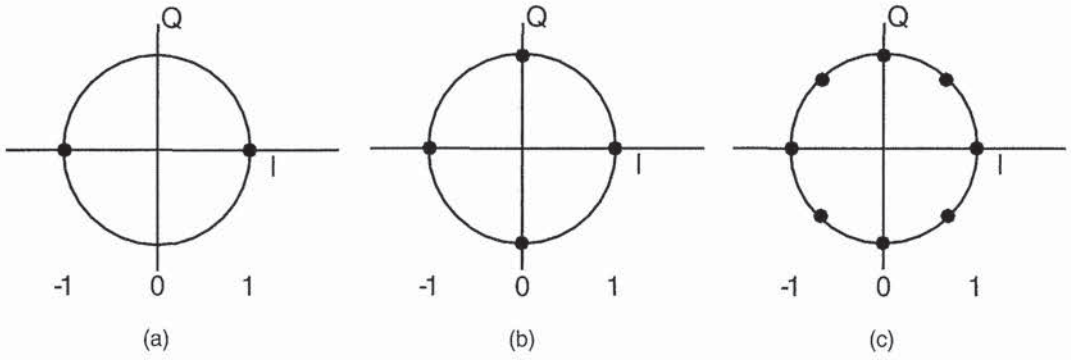


FIGURE 3.3
The three smallest PSK modulations, (a) BPSK, (b) QPSK, (c) 8-PSK.

Table 3.2 shows the required increase in *SNR* for 2- to 16-point PSK constellations. Note particularly that the increase from BPSK to QPSK is only 3 dB, compared to 2-ASK to 4-ASK increase of 6.99 dB. However, as the number of points is increased to 16, *SNR* increase for one additional bit per symbol converges towards the 6 dB figure as for ASK modulation. This behavior can be attributed to the fact that although PSK constellation is two dimensional, it still has only one degree of freedom: the phase of the carrier. Thus PSK does not fully use the two-dimensional space to locate the points of the constellation. Therefore larger than 8-point PSK modulations are not commonly used.

TABLE 3.2 Distance Properties of PSK Modulations

Modulation	P_{ave}	$d^2_{\min}$ *Normalized*	*SNR Increase*
BPSK	1	4.00	–
QPSK	1	2.00	3.00 dB
8-PSK	1	0.5858	5.33 dB
16-PSK	1	0.1522	5.85 dB

Symbol error rate of PSK modulations can be derived exactly for BPSK and QPSK; the former case was already shown in Equation 3.6. For QPSK, the P_s expression is somewhat more complicated than BPSK as shown in Equation 3.10.

$$P_s = 2Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\left[1-\frac{1}{2}Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\right] \tag{3.10}$$

Higher order PSK modulation P_s can be approximated by Equation 3.11. Bit error rate can again be approximated by dividing P_s by the number of bits $k = \log_2 M$ in the constellation.

$$P_s = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right)\sin\left(\frac{\pi}{M}\right) \tag{3.11}$$

Quadrature Amplitude Modulation

Quadrature Amplitude Modulation (QAM) changes both the amplitude and phase of the carrier, thus it is a combination of both ASK and PSK. Equation 3.12 shows the QAM signal in so-called *IQ*-form that presents the modulation of both the *I*- and *Q*-carriers. QAM can also be described by Equation 3.13 that shows how amplitude and phase modulations are combined in QAM.

$$s(t) = I_k \cos(\omega_c t) - Q_k \sin(\omega_c t) \tag{3.12}$$

$$= A_k \cos(\omega_c t + \phi_k) \tag{3.13}$$

The amplitude and phase terms of Equation 3.13 are calculated from Equations 3.14 and 3.15.

$$A_k = \sqrt{I_k^2 + Q_k^2} \tag{3.14}$$

$$\phi_k = \tan^{-1}\left(\frac{Q_k}{I_k}\right) \tag{3.15}$$

Figure 3.4 shows three QAM constellations: (a) 4-QAM or QPSK, (b) 8-QAM, and (c) 16-QAM. Especially QPSK and 16-QAM are very common modulations. On the other hand, 8-QAM is not used as often, probably due to its somewhat inconvenient shape; instead 8-PSK is commonly used when 8-point constellations are required. Figure 3.4 (b) is not the only possible shape for a 8-QAM constellation.

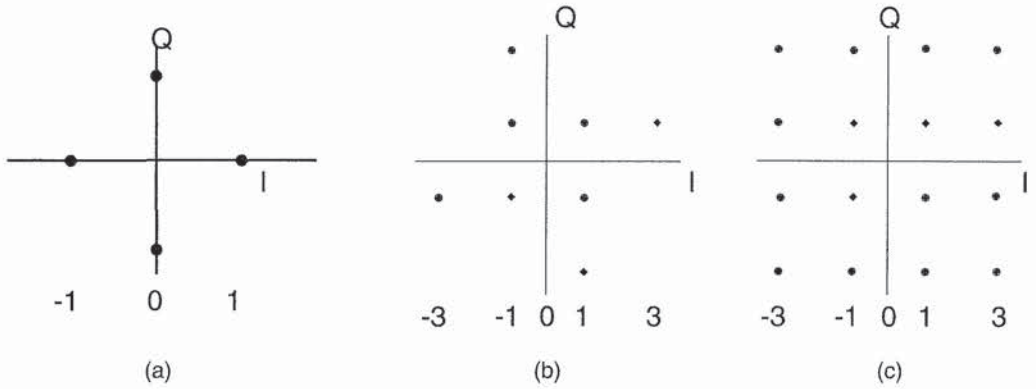


FIGURE 3.4
QAM constellations, (a) QPSK, (b) 8-QAM, (c) 16-QAM.

Table 3.3 shows how the $d_{\min}^2$ of QAM behaves as a function of the constellation size. The step from QPSK to 8-QAM is somewhat anomalous, and it is actually possible to improve the minimum distance of our example 8-QAM modulation. However, we have used the present example because of its regular form. The step from 16-QAM to 32-QAM shows a 3 dB increase in required SNR to maintain constant BER. This 3 dB SNR increase for each additional bit per symbol is a general rule for QAM constellations.

Symbol error rate of QAM modulation is approximated by considering QAM as two independent ASK modulation on both I- and Q-carriers. This is shown in Equation 3.16.

$$P_s \approx 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right) \tag{3.16}$$

TABLE 3.3 Distance Properties of QAM Modulations

Modulation	P_{ave}	d^2_{min} *Normalized*	*SNR Increase*
QPSK	1	2.00	–
8-QAM	6	0.67	4.77 dB
16-QAM	10	0.40	2.22 dB
32-QAM	20	0.20	3.01 dB

Labeling Constellation Points

Labeling constellation points means assigning a bit pattern to all the points. The selected labeling scheme has an impact on the performance and thus must be optimized. Figure 3.5 shows two common ways to assign bit patterns to 16-QAM constellation points: (a) is called natural order, and (b) is called Gray coding. Figure 3.5 shows the decimal value of the label above the point and binary value below the point. Natural ordering has appeal in its straightforward assignment of labels using decimal numbers from 0 to 15 to the points. The disadvantage of natural ordering is in the actual bit patterns representing constellation points.

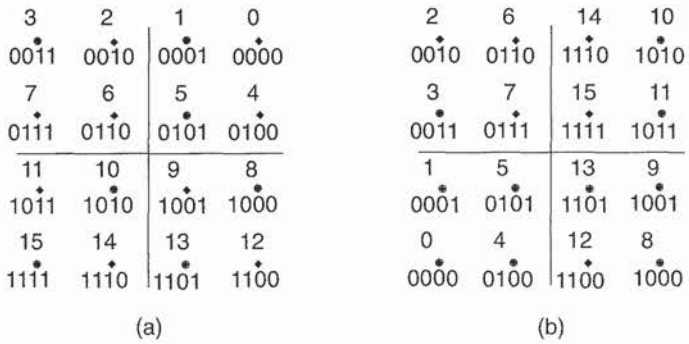


FIGURE 3.5
(a) QAM natural order and (b) Gray coded labeling schemes.

For example, consider an error where point number 1 was transmitted, but it was received as 2. This mistake between neighboring points is the most common type error the receiver does. Now look at the bit patterns representing 1 (0001) and 2 (0010); two bits have changed resulting in two bit errors for one symbol error. Gray coding eliminates all these two-bit errors for symbol errors between neighboring points, thus reducing bit error rate for the same symbol error rate. In the Gray coded constellations the points of the previous example are labeled 14 (1110) and 6 (0110); note that only left most bit is different. All the constellations in IEEE 802.11a and HiperLAN/2 standards are Gray coded.

Exercise 1 *Plot the different bit error probability functions corresponding to the modulations used in IEEE 802.11a and compare the curves with simulation results.*

Exercise 2 *Compare the performance difference predicted by minimum distace values of different constellations to the analytical bit error probability curves.*

Detection of Coherent Modulations

After the receiver has performed all the required synchronization operations, discussed in Chapter 2, the next job is to decide what was actually received. This is done by a detector or demodulator that decides for each received symbol the most likely transmitted bits. The decisions are divided into *hard* and *soft* decisions, depending on how much information about each transmitted bit is produced. Both types of detectors are discussed next.

Hard Decision Detection

A hard decision *demodulator* or *slicer* makes a definite determination of whether a zero or one bit was transmitted, thus the output of the demodulator are zeros and ones. A hard decision demodulator is also defined based on the number of possible inputs to the modulator in the transmitter. If this number is equal to the number of possible outputs from the demodulator in the receiver, the demodulator uses hard decision. If the demodulator can output more information than the input to the modulator, the system uses soft decision, which is discussed later. Figure 3.6 shows the *decision boundaries* for QPSK constellation. The decision boundaries determine how received symbols are mapped to bits. Essentially, the maximum-likelihood decision is the constellation point that is closest to the received symbol, and the hard decisions are the bits assigned to that constellation point. For example, in Figure 3.6, the x marks the location of the received symbol; it is closest to the constellation point on the positive I-axis, hence the hard decision bits are 00.

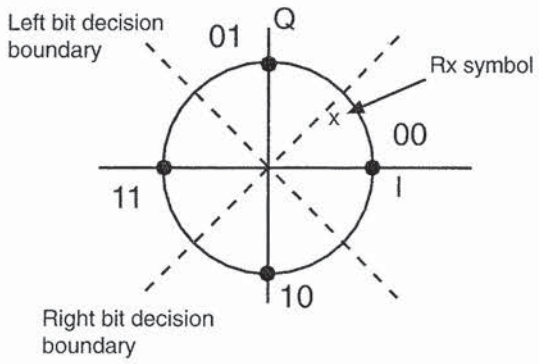


FIGURE 3.6
Hard decision boundaries for QPSK constellation.

Soft Decision Detection

Soft decision demodulator outputs "*soft*" bits, that in addition to indicating a zero or one bit retain information about the reliability of the decision. This additional information can greatly improve the performance of channel coding schemes; the effect will be discussed in "Channel Codes." To perform soft decisions, the demodulator has to consider the received bits individually. Consider again the received symbol in Figure 3.6. It is located very close to the decision boundary between symbols corresponding to bits 00 and 01. For the left bit to change to 1, the symbol would have to move over the left bit decision boundary. The received symbol is quite far from this boundary, hence the left bit is quite reliable. On the other hand, the right bit changes from 0 to 1; if the received symbol would be above the right bit decision boundary, instead of just below it. This flip does not require much additional noise, hence the right bit is quite unreliable. To reflect this difference of reliability of the soft decision, the soft bits have different values: a large absolute value for the first bit and a small value for the second bit. The sign of the soft decision indicates a 0 or 1 bit. The absolute value of each soft decision is the distance to the decision boundary. We will show the performance effect of using either hard or soft decision with the convolutional error correcting code used in IEEE 802.11a in "Channel Codes."

Non-Coherent Modulations

Non-coherent modulations can be used by a communication system that does not maintain a phase lock between transmitter and receiver, or have knowledge of the amplitude change of the transmitted symbol caused by the channel. This means that the received symbols are rotated and scaled arbitrarily compared to the transmitted symbol. Therefore the ASK, PSK, or QAM modulations cannot be used because they require the received symbol phase and amplitude to be very close to that transmitted phase and amplitude. The solution is to use differential PSK (DPSK) or differential APSK (DAPSK) modulation. Differential modulations encode the transmitted information to a phase, or phase and amplitude change from one transmitted symbol to the next. This encoding introduces memory to the signal, because transmitted symbols depend on previous symbols. As a consequence, the demodulator has to consider two consecutive symbols when making decisions. The next two sections describe these two modulation methods.

The main benefit of differential encoding is significantly simplified receiver structure. Several of the synchronization algorithms presented in Chapter 2 are not needed in a non-coherent receiver. Specifically, phase tracking and channel estimation are not needed, because absolute knowledge of carrier phase and the channel effect is not needed. Carrier frequency estimation could also be removed, if the system can tolerate the performance due to intercarrier interference caused by lost orthogonality between the subcarriers. Regardless of these simplifications in receiver design, non-coherent modulations have not achieved popularity among high speed communications systems. All the IEEE 802.11a and HiperLAN/2 modulations are coherent. The main reason is unavoidable performance loss associated with differential approach. In contrast, low data rate

systems do use differential techniques, mainly DPSK modulations. For example, the European Digital Audio Broadcasting (DAB) system uses differential modulation.

Differential Phase Shift Keying

Differential phase shift keying changes the carrier phase from its current value according to the data bits. For binary DPSK, the encoding operation can be expressed at bit level, as in Equation 3.17, which shows how the transmitted bit b_n is calculated from the data bits d_n using binary XOR operation.

$$b_n = d_n \oplus d_{n-1} \tag{3.17}$$

Alternatively, encoding can be expressed directly as carrier phase change as in Equation 3.18

$$s(t) = \cos(\omega_c t + \Delta\phi_n + \phi) \tag{3.18}$$

where ϕ is unknown phase difference between transmitter and receiver and $\Delta\phi_n = \phi_n - \phi_{n-1}$ is the phase difference of two PSK modulated signals. DPSK is used in communications systems that require simple receiver structures. The constellation for M-DPSK modulation is identical with the corresponding M-PSK constellation; only the mapping of bits to constellation points is changed.

Differential Amplitude Phase Modulation

Differential Amplitude Phase Modulation (DAPSK) combines differential phase and differential amplitude modulation. Figure 3.7 shows a 32-point DAPSK constellation that uses three bits or eight levels for phase modulation and two bits or four levels for amplitude modulation. The differential phase modulation is analogous to the regular DPSK. Differential amplitude modulation, on the other hand, has to change the constellation shape compared to coherent amplitude modulation. The reason is the unknown scaling of the amplitude of the transmitted symbol caused by the channel. The receiver cannot correctly detect the symbols unless this unknown scaling is removed. Equation 3.19 shows a received OFDM subcarrier symbol R_k when the transmitted symbol was X_k and the subcarrier frequency response is H_k.

$$R_k = H_k X_k \tag{3.19}$$

To recover the amplitude modulation the receiver is only concerned with the amplitude of the received symbol, as shown in Equation 3.20.

$$|R_k| = |H_k X_k| = |H_k||X_k| \tag{3.20}$$

A general assumption when differential modulation is used is that the channel and carrier phase are constant during two consecutive symbols. Therefore we can cancel the effect of the channel by dividing two consecutive symbols, as shown in Equations 3.21 – 3.23.

$$Y_k = \frac{|H_k||X_k|}{|H_{k-1}||X_{k-1}|} \tag{3.21}$$

$$= \frac{|H_k||X_k|}{|H_k||X_{k-1}|} \tag{3.22}$$

$$= \frac{|X_k|}{|X_{k-1}|} \tag{3.23}$$

The multiplicative scaling of the channel prevents using additive differential amplitude modulation, whereas for DPSK the channel effect on received symbol phase is additive, hence it can be canceled by subtraction instead of division. The next question is how the amplitude levels of the constellation are selected. This is illustrated in Figure 3.7, which shows amplitude levels $1, a, a^2, a^3$. The information is encoded as a jump from one amplitude level to another; for example, bits 01 could mean a jump of one amplitude level. Differential amplitude encoding done with multiplication such as

$$Y_k = D_k Y_{k-1} \tag{3.24}$$

where $D_k \in \left\{\frac{1}{a^3}, \frac{1}{a^2}, \frac{1}{a}, 1, a, a^2, a^3\right\}$. The fractional values are required, because amplitude multiplication is not naturally periodic like phase addition in the case of DPSK modulation. The actual value of D_k depends on the input bits and the value of Y_{k-1}. The value of D_k is selected such that the amplitude jumps wrap around after a^3. For example, if $Y_{k-1} = a^2$ and a two amplitude level jump is needed, then $D_k = \frac{1}{a^2}$, so $Y_k = a^2 \frac{1}{a^2} = 1$. The two amplitude level jump is then $a^2 \rightarrow a^3 \rightarrow 1$.

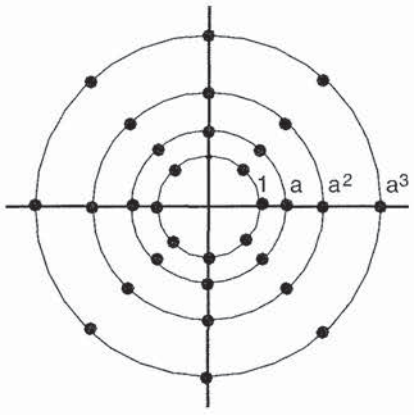


FIGURE 3.7
32-point differential amplitude and phase modulation.

The multiplication encoding method forces the points in the middle of the constellation quite close to each other. This is an unavoidable disadvantage of differential amplitude modulation. The effect is reduced minimum distance for the points in the middle, hence they experience a performance loss compared to the points on the outer amplitude levels.

Detection of Differential Modulations

Detection of differential modulation is done in two steps. First the differential encoding is removed from the signal and then a normal demodulation is performed as explained for regular PSK and QAM constellations. Equations 3.25 and 3.26 show how the differential encoding is removed in DPSK detection by calculating the phase difference of two consecutive symbols. Equation 3.26 also show the reason for the performance loss of differential encoding compared to coherent modulations. Differential detection has to use two symbols, therefore the amount of noise per detected symbol is approximately doubled.

$$\hat{\theta}_k = \angle(R_k + N_k) - \angle(R_{k-1} + N_{k-1}) \tag{3.25}$$

$$= \theta_k - \theta_{k-1} + \tilde{N}_k + \tilde{N}_{k-1} \tag{3.26}$$

The performance loss of DPSK compared to coherent modulation varies with the size of the modulation [11], for DBPSK it is between 1-2dB, for DQPSK about 2.3dB and for larger constellations 3dB.

Linear and Nonlinear Modulation

Different modulation methods are generally divided into linear and nonlinear modulations. The differentiating property is whether the transmitted signal can be expressed as a sum of the individual symbols a_k with pulse shape $p(t)$ as in Equation 3.27.

$$s(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT) \tag{3.27}$$

Nonlinear modulation usually contains memory in the modulator, like differential modulation, so that the transmitted symbols are dependent and cannot be simply delayed and summed together to form the transmitted waveform. Differential modulation is one application of nonlinear modulation; another common one is *continuous phase modulation* (CPM). The main goal of CPM is to control the spectrum of the transmitted signal by forcing the phase of the carrier wave to change continuously between symbols. This narrows the spectrum of the signal, because the change between symbols is less abrupt than without carrier phase control. CPM is not very useful for OFDM, because the spectrum shape is determined by the multicarrier nature and has a brick-like shape, even with linear modulation. Another difficulty in applying CPM to OFDM is that the signal would have to have a continuous phase for all the subcarriers. This is difficult or impossible to

achieve, because of the cyclic prefix added to OFDM symbols. Therefore CPM is not used with OFDM systems.

Interleaving

Interleaving aims to distribute transmitted bits in time or frequency or both to achieve desirable bit error distribution after demodulation. What constitutes a desirable error distribution depends on the used FEC code. What kind of interleaving pattern is needed depends on the channel characteristics. If the system operates in a purely AWGN environment, no interleaving is needed, because the error distribution cannot be changed by relocating the bits. Communication channels are divided into *fast* and *slow* fading channels. A channel is fast fading if the impulse response changes approximately at the symbol rate of the communication system, whereas a slow fading channel stays unchanged for several symbols.

WLAN systems generally assume a very slowly fading channel, also called *quasi-stationary*, that does not change during one packet. Another characterization of communications channels is as *flat* or *frequency selective* fading channels. A channel is flat fading if the frequency response is constant over the whole bandwidth of the transmitted signal. A frequency selective channel changes significantly within the band of the signal. WLAN systems are wide bandwidth systems, and therefore usually experience frequency selective fading channel. OFDM technology is well suited for communication over slow frequency selective fading channels.

Diversity is the single most important technique to improve the performance of a communication system in a fading channel. Diversity generally refers to methods that take advantage of several different realizations of the channel to make a decision on each received information bit. The more independent channel realizations can be used, the better the performance improvement will be. This is a consequence of the channel fading statistics. The more individual samples of the channel are combined, the less likely it is that all of them are in a faded state. There are several sources of diversity that can be used: time, frequency, and space are the most common ones. OFDM wireless LANs mainly exploit frequency diversity due to their wideband nature. Time diversity cannot be used, because of the slowly fading characteristics of the channel. When the channel is in a bad condition, it will remain so for the duration of the packet, and there is nothing the receiver can do about it. Space diversity is achieved by using multiple transmit and/or receiver antennas. This technique is discussed in detail in Chapter 4.

Interleaving necessarily introduces delay into the system because bits are not received in the same order as the information source transmits them. The overall communication system usually dictates some maximum delay the system can tolerate, hence restricting the amount of interleaving that can be used. For example, cellular telephone systems usually use time diversity, because the channels are fast fading. However the maximum phone to phone delay is usually constrained

to 20ms or less, to prevent noticeable degradation in call quality. This means the maximum interleaving delay must be much less than 20ms to allow for other delay sources in the system.

Block Interleaver

Block interleaving operates on one block of bits at a time. The number of bits in the block is called *interleaving depth*, which defines the delay introduced by interleaving. A block interleaver can be described as a matrix to which data is written in columns and read in rows, or vice versa. For example, Figure 3.8 shows a 8×6 block interleaver, hence interleaving depth is 48. The input bits are written in columns as $[b_0, b_1, b_2, b_3, \ldots]$ and the interleaved bits are read by rows as $[b_0, b_8, b_{16}, b_{24}, \ldots]$. Block interleaver is simple to implement using random access memory (RAM) on current digital circuits.

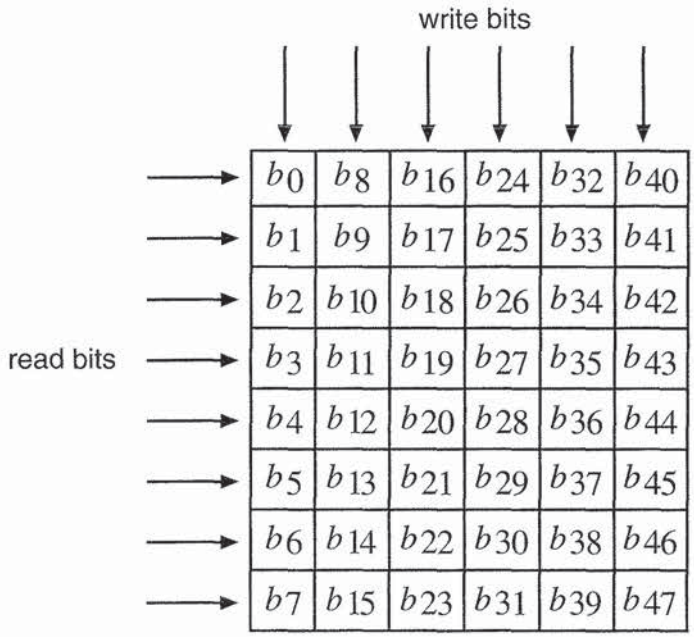


FIGURE 3.8
Bit write and read structure for a 8×6 block interleaver.

Deinterleaving is the opposite operation of interleaving; that is, the bits are put back into the original order. The deinterleaver corresponding to Figure 3.8 is simply a transpose of the original 8×6 matrix. The deinterleaver is a 6×8 matrix to which the received bits are written into columns and read from rows.

Convolutional Interleaver

A convolutional interleaver is another possible interleaving solution that is most suitable for systems that operate on continuous stream of bits. This interleaver structure was published by Ramsey [12]. Figure 3.9 shows the basic structure of a convolutional interleaver. The interleaver operates by writing the bits into the commutator on the left, and reading bits out from the commutator on the right. The delay elements D are clocked after each cycle of the commutators is completed; that is, after the last delay line has been written and read. The main benefit of a convolutional interleaver is that it requires approximately half of the memory required by a block interleaver to achieve the same interleaving depth. This saving can be significant for long interleaver depths.

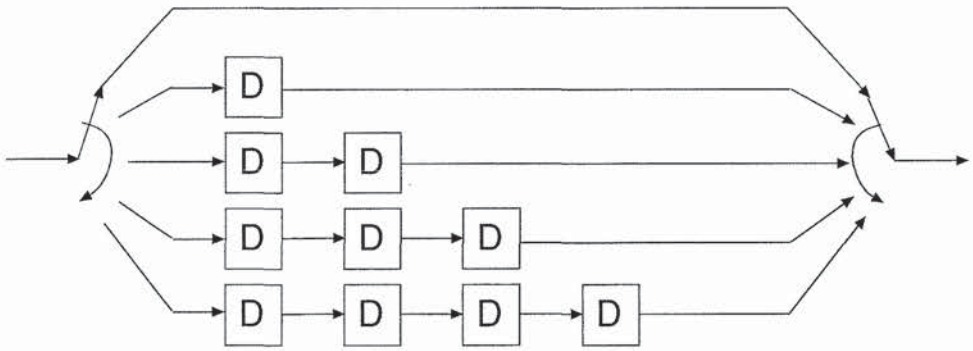


FIGURE 3.9
Basic structure of a convolutional interleaver.

Deinterleaving of convolutional interleaver is achieved by flipping the interleaver along its horizontal axis. The structure is otherwise identical except the longest delay line is at the top, and the no delay line is last. Otherwise the deinterleaving operation is identical to interleaving.

Interleaving in IEEE 802.11a

Interleaving is a very important component of the IEEE 802.11a standard. Interleaving depth has been selected to be equal to one OFDM symbol. Therefore it is naturally a block interleaver. The performance effect of interleaving in IEEE 802.11a is a consequence of frequency diversity. IEEE 802.11a is a wideband communications system and experiences a flat fading channel very rarely. This is an essential requirement to be able to exploit frequency diversity. The combined effect of interleaving and convolutional channel coding takes advantage of the frequency diversity provided by the wideband nature of the transmitted signal.

Interleaving depth is only one OFDM symbol, because the channel is assumed to be quasi-static; that is, the channel is assumed to stay essentially the same for the duration of a transmitted

packet. Therefore no additional diversity gain can be achieved by interleaving in time. Additionally increasing the interleaving depth would increase the delay of baseband processing; the maximum delay possible is constrained by the IEEE 802.11 MAC level protocol's acknowledgement packet short interframe spacing (SIFS) timing requirements. The SIFS time is equal to 16μs, hence after the packet ends, processing it has to be completed in less time. This is one of the most demanding implementation requirements of the IEEE 802.11a standard.

The interleaving depth measured in bits changes according the used modulation: BPSK, QPSK, 16-QAM, and 64-QAM have interleaving depths of 48, 96, 192, and 288 bits, respectively. The interleaving depth for each modulation is calculated by multiplying the number of data subcarriers by the number of bits per symbol.

Figures 3.10 and 3.11 show BER and PER curves for IEEE 802.11a 12 Mbits/s mode in a Rayleigh fading channel that has 75ns root mean square (rms) delay spread. The effect of the interleaver is most striking for PER, at 1% the gain from interleaver is approximately 5 dB. Figure 3.11 also shows how the slope of the curve is steeper with interleaving; this is a consequence of the diversity gain achieved.

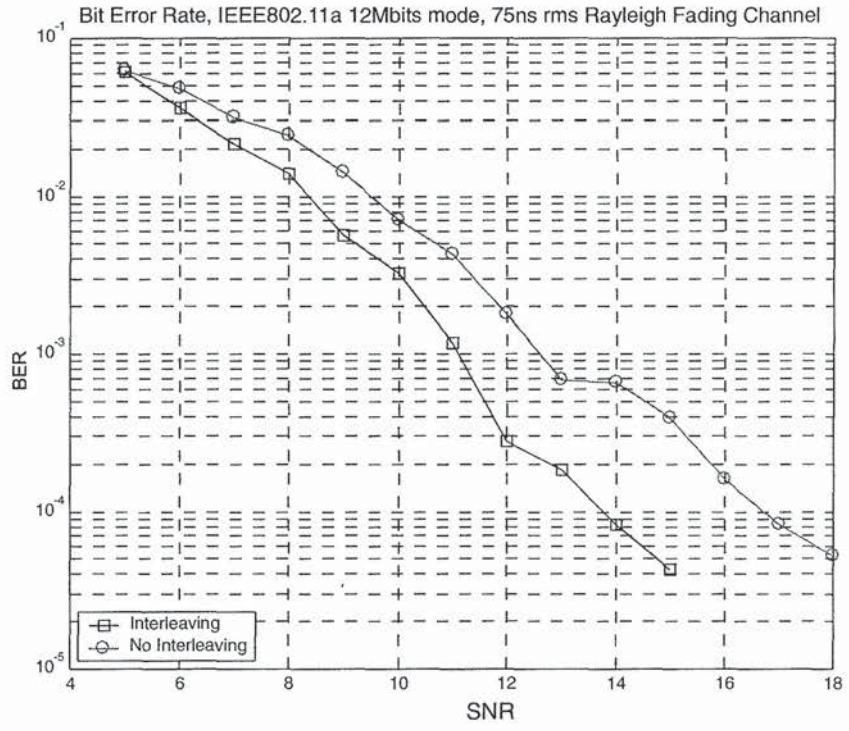


FIGURE 3.10

Bit error rate of IEEE 802.11a 12 Mbits mode with and without interleaving in 75ns rms delay spread Rayleigh fading channel.

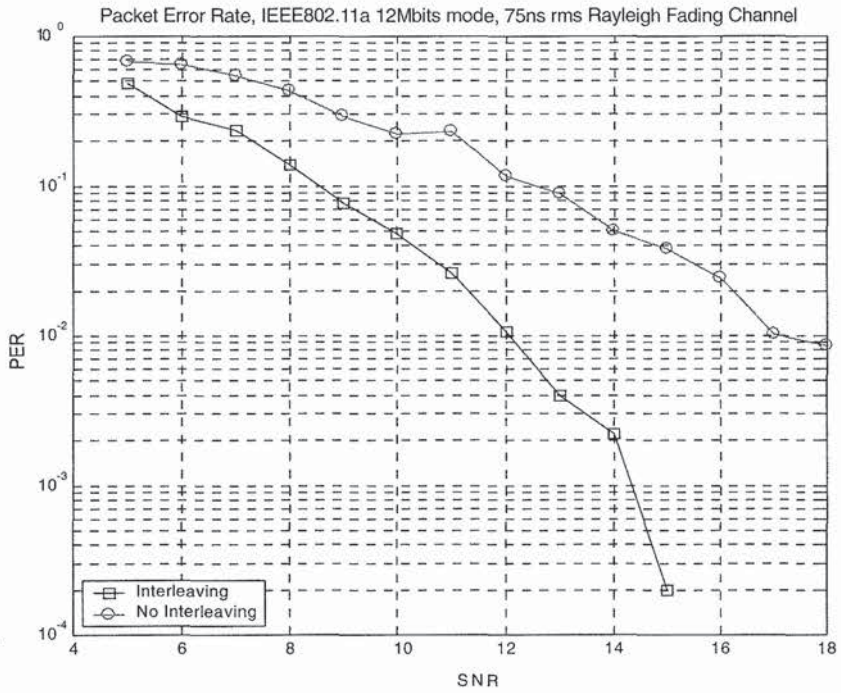


FIGURE 3.11

Packet error rate of IEEE 802.11a 12 Mbits mode with and without interleaving in 75ns rms delay spread Rayleigh fading channel.

Channel Codes

Channel codes are a very important component of any modern digital communication system, and they make today's effective and reliable wireless communications possible. There are several books that concentrate on different channel coding approaches [7, 9, 10, 17, 20]. The basic measure of channel coding performance is *coding gain*, which is usually measured in dBs as the reduction of required $\frac{E_b}{N_o}$ to achieve a certain bit error rate in AWGN channel. As an example, IEEE 802.11a could use two methods to achieve 12 Mbits/s data rate. The simplest way would be to use uncoded BPSK modulation on each subcarrier, therefore each OFDM symbol would carry 48 bits worth of information. The symbol time is $4\mu s$ or 250000 symbols per second, hence the overall data is 250000×48 = 12Mbits/s.

Another way to achieve the same data rate is to use channel coding. The IEEE 802.11a 12Mbits/s mode uses QPSK and rate 1/2 convolutional code. This results in a significantly lower $\frac{E_b}{N_o}$ or *SNR* to achieve a good BER performance. This is illustrated in Figure 3.12, which shows a coding gain of approximately 5.5dB at bit error rate of 10^{-5}. This means that to achieve the same performance, the system that does not use channel coding has to spend 5.5dB more energy for each

transmitted bit than the system that uses channel coding. Figure 3.12 shows another typical behavior of channel codes, the uncoded and coded BER curves cross over at some point. However this happens at so high BER that channel coding practically always provides an improvement.

To make an even more striking demonstration of channel coding in action, Figure 3.13 shows packet error rate curves in a fading channel with the same parameters as the previous example. IEEE 802.11a is a packet-oriented system, hence PER is the most commonly used measure of its performance. In the figure, the curve without channel coding does *not* produce usable PER at all, hence the price paid for implementing channel coding is handsomely paid back. Over the 9dB increase in $\frac{E_b}{N_o}$, the performance of the coded system improves by two orders of magnitude, whereas the performance of the uncoded system only improves from 100% PER to about 80% PER. Flatness of the uncoded PER curves is explained by noting that even one bit error causes a packet error. The system is operating in a fading channel, hence typically some subcarriers experience very low SNR, and bit errors on these bad subcarriers cause packet errors.

The previous examples of channel coding options show another important parameter of channel coding, namely the *rate* of the code. Code rate is the ratio of bits input, called the *message word,* to the encoder to the bits output, called the *code word,* from the encoder. This ratio is always less than one; in the previous example, rate 1/2 means that twice as many bits are output from the encoder than were input. Therefore the coded system had to use QPSK modulation, instead of BPSK to achieve the same data rate. Channel coding always forces the system to use a larger constellation to keep the same data rate as an uncoded system. Going to a larger constellation reduces $d_{\min}$; this implies higher BER at the output of the demodulator. However, at the output of the channel code decoder, the bit error rate is significantly reduced.

The performance of channel codes is ultimately limited by the channel capacity formula, Equation 3.1. After about 50 years of research, *Turbo-codes* [7] have finally emerged as a class of codes that can approach the ultimate limit in performance. Another innovation, and a very active research area, are Low Density Parity Check (LDPC) codes, which also have performance very close to the capacity.

Convolutional codes have been the most widely used channel code in wireless systems for the past decades, hence this section concentrates on them and the performance of IEEE 802.11a system. Additional channel codes are Trellis-Coded Modulation (TCM) that is closely related to convolutional codes and algebraic Reed-Solomon (RS) codes. These coding schemes are described in sections "Trellis Coded Modulation" and "Block Codes," respectively.

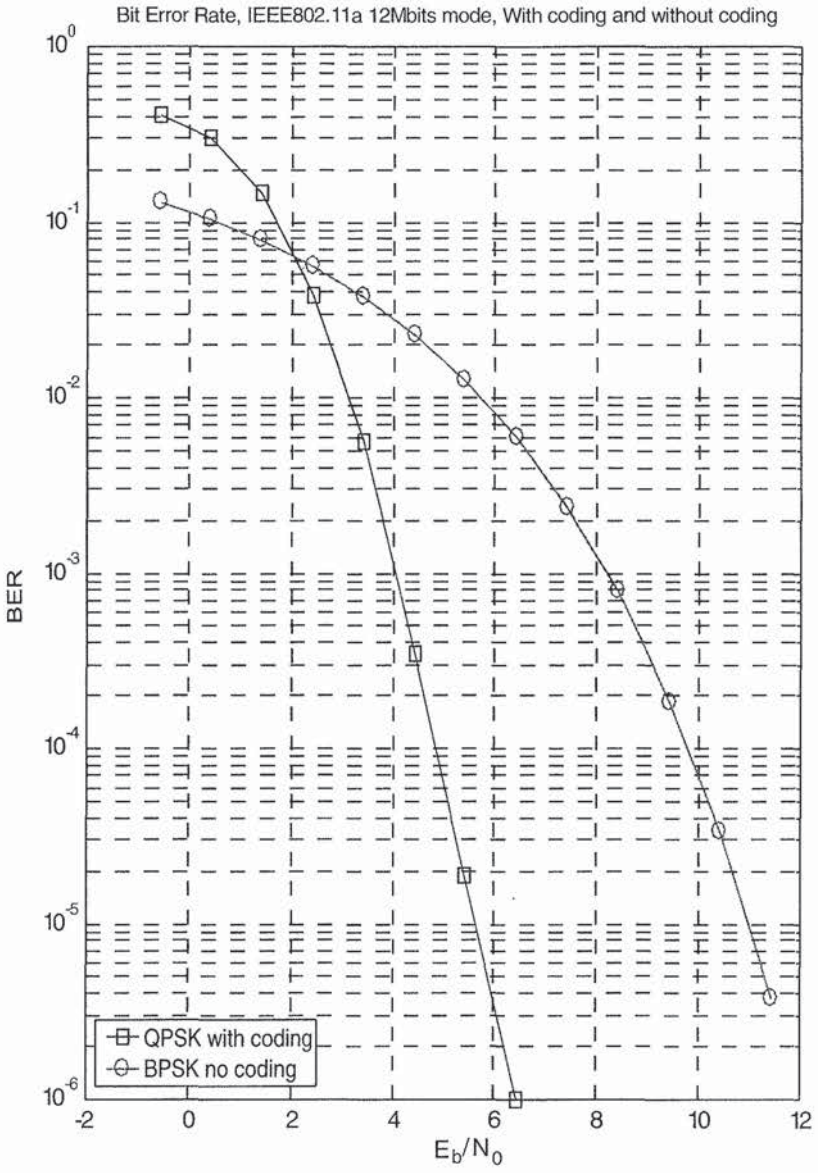


FIGURE 3.12
Coding gain in BER of 64 state rate 1/2 convolutional code in AWGN channel.

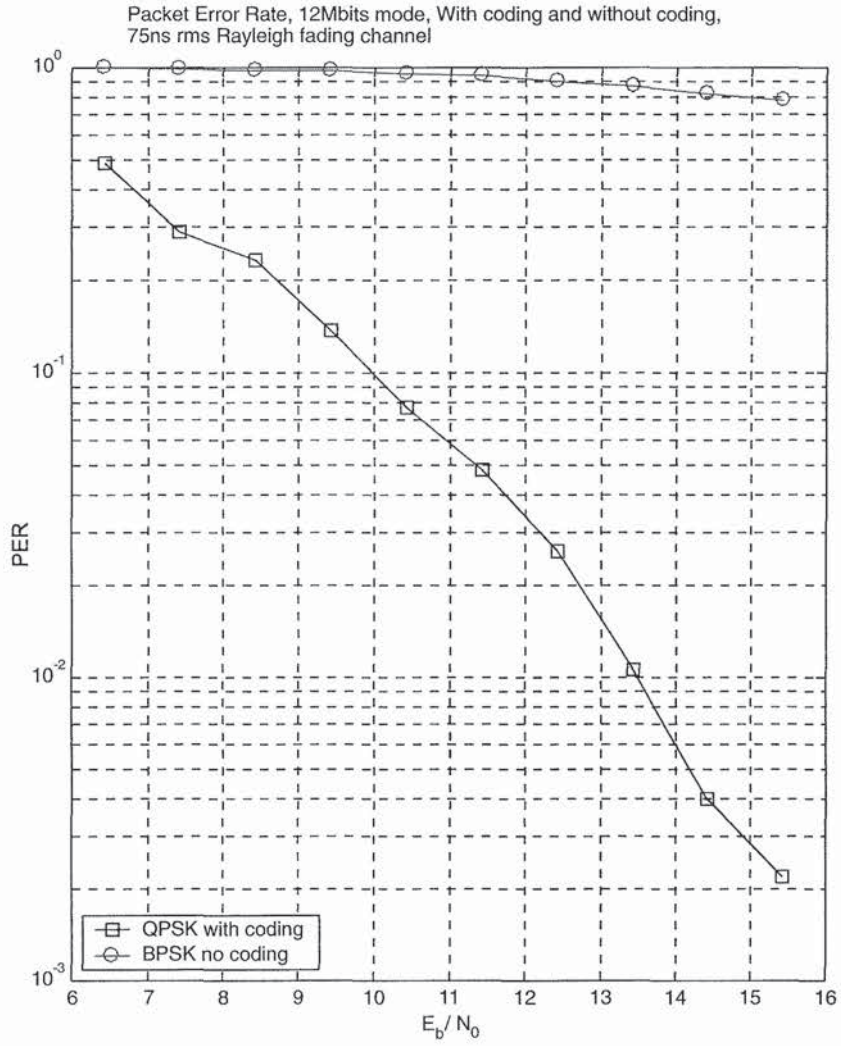


FIGURE 3.13

Coding gain in PER of 64 state rate 1/2 convolutional code in fading channel.

Convolutional Codes

Convolutional codes are one of the mostly widely used channel codes in today's systems; all the major cellular systems (GSM, IS-95) in use today use convolutional channel codes. IEEE 802.11a and HiperLAN/2 WLAN standards also use convolutional error correcting code, and IEEE 802.11b includes an optional mode that uses them. Convolutional codes owe their popularity to good performance and flexibility to achieve different coding rates.

A convolutional code is defined by a set of connections between stages of one or more shift registers and the output bits of the encoder. The number of shift registers k is equal to the nominator of the code rate and the number of output bits n is equal to the denominator. The rate of a convolutional code is generally expressed as $\frac{k}{n}$. The number of connections required to describe a code is equal to the product of k and n. For each output bit there are k connections that define how the value of the output bit is calculated from the state of the shift registers.

For example, Figure 3.14 shows the convolutional encoder used in IEEE 802.11a. This is a rate $\frac{1}{2}$ code with connections 133_8 and 171_8. The connections are defined as octal numbers, the binary representations are $001\,011\,011_2$ and $001\,111\,001_2$. The octal notation is used to shorten the expressions, when connections for different convolutional codes are tabulated. From the binary notation, the structure of the encoder is easily constructed. The connections are aligned to the end of the shift register, and a value of 1 means that the shift register stage output is connected to one of the output bits of the encoder using a binary XOR operation. In Figure 3.14 the connection 133_8 that defines values the even indexed bits b_{2n} and the connection 171_8 defines the values of the odd indexed bits b_{2n+1}.

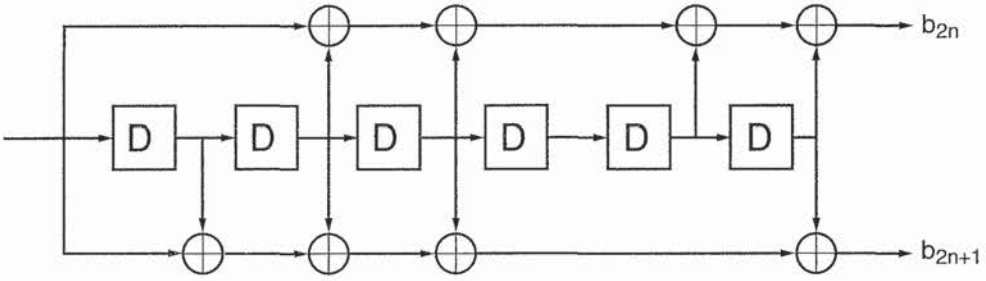


FIGURE 3.14
64 state rate 1/2 convolutional encoder used in IEEE 802.11a.

The end of shift register alignment for defining connection values is not universal. Several sources report the connection values by aligning them to the start of the shift register input. Using this notation the connection for the code in Figure 3.14 are 554_8 and 744_8 or in binary $101\,101\,100_2$ and $111\,100\,100_2$. The length of the binary representation is extended with zeros to be a multiple of three, so that the octal values are well defined. Connection values for several different convolutional codes can be found in coding theory text books [9, 10, 20]. These codes were all found using an exhaustive computer search for all possible convolutional codes of the specific rate and number of shift register elements.

The number of shift register elements determines how large a coding gain the convolutional code can achieve. The longer the shift registers, the more powerful the code is; unfortunately, the decoding complexity of the maximum likelihood Viterbi algorithm grows exponentially with the

number of shift register elements. This complexity growth limits the currently used convolutional codes to eight shift register elements, and IEEE 802.11a uses only six, due to its very high speed data rates.

The performance of a convolutional code is determined by the minimum *free distance* of the code. Free distance is defined using the *Hamming distance* that is equal to the number of position in which two code words are different. Free distance of a convolutional code is the minimum Hamming distance between two different code words. An asymptotic coding gain at high SNR for a convolutional code can be calculated from the free distance and the rate of the code, as in Equation 3.1.

$$coding\ gain = 10\log_{10}(rate \cdot free\ distance) \tag{3.28}$$

For example, the code used in IEEE 802.11a has a free distance of 10 and rate $\frac{1}{2}$ that gives an asymptotic coding gain of $10\log_{10}\left(\frac{1}{2}\cdot 10\right) = 7.0dB$. However, this is an asymptotic result; Figure 3.12 showed that at 10^{-5} BER, the coding is equal to 5.5dB, and less for lower $\frac{E_b}{N_o}$. The asymptotic result is reached at such a high SNR that practical systems usually do not operate at that SNR region.

Puncturing Convolutional Codes

Usually communications systems provide a set of possible data rates; for example, IEEE 802.11a has eight different data rates: 6, 9, 12, 18, 24, 36, 48, and 54 Mbits/s. Now if the system could only change the data rate by adjusting the constellation size, and not the code rate, a very large number of different rates would be difficult to achieve as the number of constellations and the number of points in the largest constellation would grow very quickly. Another solution would be to implement several different convolutional encoders with different rates and change both the convolutional code rate and constellation. However this approach has problems in the receiver that would have to implement several different decoders for all the codes used.

Puncturing is a very useful technique to generate additional rates from a single convolutional code. Puncturing was first discover by Cain, Clark, and Geist [3], and subsequently the technique was improved by Hagenauer [6]. The basic idea behind puncturing is not to transmit some of the bits output by the convolutional encoder, thus increasing the rate of the code. This increase in rate decreases the free distance of the code, but usually the resulting free distance is very close to the optimum one that is achieved by specifically designing a convolutional code for the punctured rate. The receiver inserts dummy bits to replace the punctured bits in the receiver, hence only one encoder/decoder pair is needed to generate several different code rates.

The bits that are not transmitted are defined by a *puncturing pattern* . Puncturing pattern is simply a set of bits that are not transmitted within a certain period of bits. Figure 3.15 shows the two different puncturing patterns of IEEE 802.11a. The pattern (a) is used to generate rate $\frac{3}{4}$ code

from the rate $\frac{1}{2}$ mother convolutional code. This puncturing pattern has a period of six bits, and bits 3 and 4 are punctured (not transmitted) from each period. The *puncturing rate* is equal to $\frac{4}{6} = \frac{2}{3}$ and the overall code rate is equal to $\frac{1}{2\frac{2}{3}} = \frac{3}{4}$, since $\frac{2}{3}$ of the original encoded bits are output from the puncturer. The other punctured code rate of IEEE 802.11a is a rate $\frac{2}{3}$ code. The puncturing pattern is shown in Figure 3.15 (b) has a period of 4 bits, and the fourth bit is punctured, the puncturing rate is $\frac{3}{4}$, hence the overall code rate is $\frac{1}{2\frac{3}{4}} = \frac{2}{3}$.

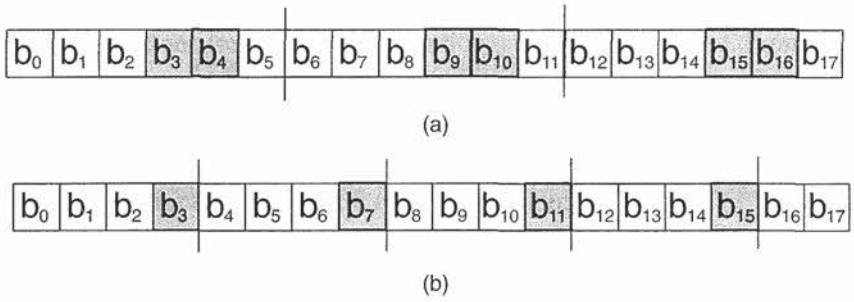


FIGURE 3.15
Puncturing patterns of IEEE 802.11a.

Table 3.4 shows the free distances and the asymptotic coding gains of the three code rates used in IEEE 802.11a. The table also shows the optimum rate $\frac{3}{4}$ and $\frac{2}{3}$ codes; as you can see, the performance loss due to using punctured codes, instead of the optimum ones, is very small. The rate $\frac{1}{2}$ is naturally the optimum code, because the original code is a rate $\frac{1}{2}$ code. Therefore the table does not show the punctured free distance and coding gain values for this rate.

TABLE 3.4 Free Ddistances of the 64 State Convolutional Codes Used in IEEE 802.11a

Code Rates	*Punctured Free Distance*	*Punctured Coding Gain*	*Optimum Free Distance*	*Optimum Coding Gain*
$\frac{1}{2}$	–	–	10	7.0 dB
$\frac{2}{3}$	6	6.0 dB	7	6.7 dB
$\frac{3}{4}$	5	5.7 dB	6	6.5 dB

Before the punctured code can be decoded, the removed bits have to be inserted back into the bit stream, because the decoder of the original code expects them to be there. Depuncturing is done

by simply inserting dummy bits into the locations that were punctured in the transmitter. The values of the dummy bits depend on whether the system uses hard or soft decisions. A hard decision system should insert randomly one and zero bits into the punctured locations. A soft decision receiver inserts a soft decision value of zero. For the usual case of decoding with the Viterbi algorithm, the zero-valued dummy bit does not have any effect on the outcome of the decoder.

Decoding Convolutional Codes

There are several algorithms that can be used to decode convolutional codes. The algorithms vary in complexity and performance, and more complexity means better performance. During recent years, the Viterbi algorithm has reached a dominant position as the method to decode convolutional codes, especially in wireless applications; other methods are practically nonexistent. The reason is that the Viterbi algorithm is a maximum-likelihood code word estimator; it provides the best possible estimate of the transmitted code word. The Viterbi algorithm was introduced in Chapter 1, and it is discussed in detail in any of the coding theory text books mentioned in the start of this section, so we assume you are familiar with it.

The Viterbi algorithm can easily be implemented using either hard or soft decision demodulation. Figures 3.16 and 3.17 show the performance improvement in BER and PER gained by using soft decisions. The gain is approximately 2dB at 10^{-3} BER and 10^{-1} PER. The gain increases to a little bit more than 2dB for higher SNRs. Therefore soft decision decoding is the recommended method to use with Viterbi decoding because the performance improvement it provides does not cost any communications resources.

A coherent OFDM system has to estimate the frequency response of the channel. The information about the amplitudes of the individual subcarriers can be incorporated into the Viterbi algorithm to provide performance improvement. Equation 3.29 shows how the path metrics p_n in the Viterbi algorithm are calculated by weighting the squared Euclidian distance between the soft decision $\hat{b}_n$ and the reference value b_n by the squared amplitude of the subcarrier k on which the bit was transmitted.

$$p_n = |H_k|^2 \left|\hat{b}_n - b_n\right|^2 \tag{3.29}$$

The performance effect of this weighting is significant in fading channel. Figure 3.18 shows the PER of IEEE 802.11a 12Mbits/s mode when the convolutional code decoding is done with using weighting in Viterbi or by using equal weight for all the path metrics. The impact of weighting on performance is very large. The reason is diversity; when the subcarrier is faded, its amplitude is small and the path metric will be scaled down to almost zero. This means that the bits that were transmitted on bad subcarriers have very little impact on the decision the decoder makes, hence improved performance.

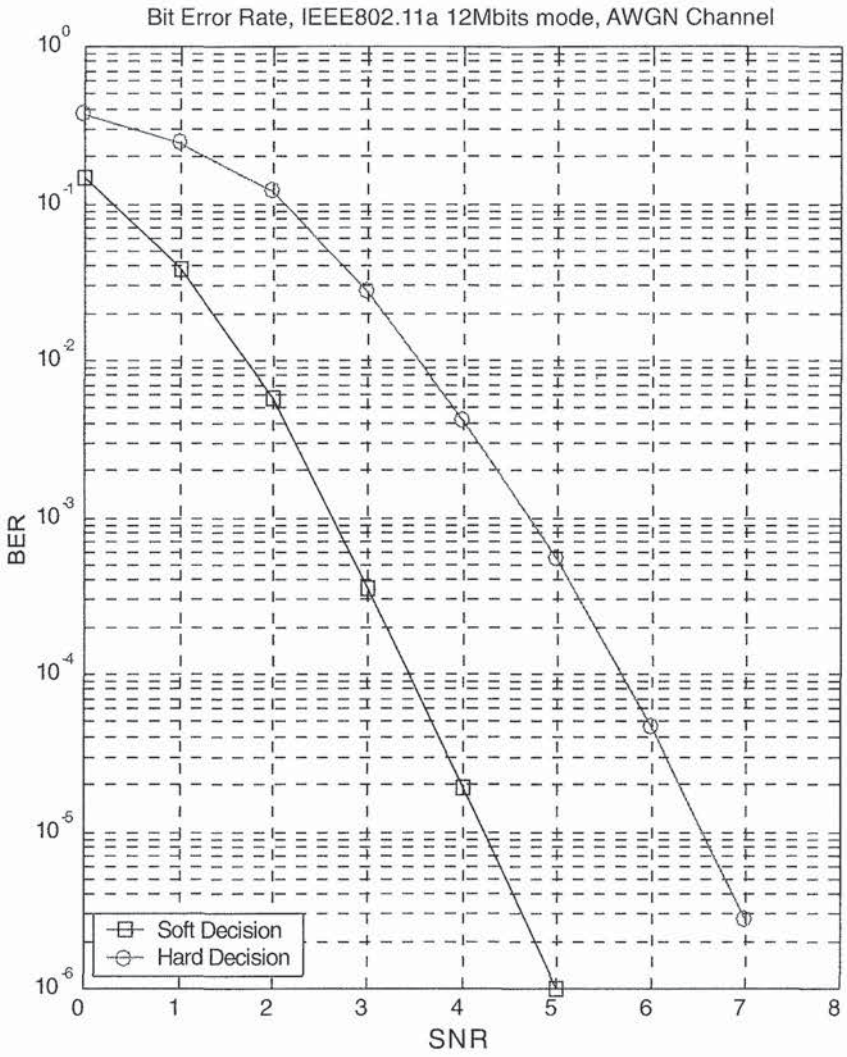


FIGURE 3.16

Bit error rate of IEEE 802.11a 12Mbits/s mode in AWGN channel using soft and hard decision decoding.

Performance of IEEE 802.11a

IEEE 802.11a offers eight different data rates from 6 Mbits/s to 54 Mbits/s with increasingly higher SNR required as the data rate increases. Figure 3.19 shows the SNR required for the 6 Mbits/s and 54 Mbits/s modes in AWGN channel. The total increase in SNR at 1% PER is about 17dB or about fifty fold increase in transmitted power. Another way to interpret the SNR increase is to look at the change of range of the system at different data rates. The range depends on the

path loss the transmitted signal experiences, before it reaches the receiver antenna. The overall path loss L_p depends on many factors [18]. The range dependent factor is proportional to the distance d between the transmitter and receiver. The impact of d on path loss depends on path loss coefficient c, that determines how fast the signal power attenuates as it travels. This relationship is shown in Equation 3.30.

$$L_p \sim \frac{1}{d^c} \tag{3.30}$$

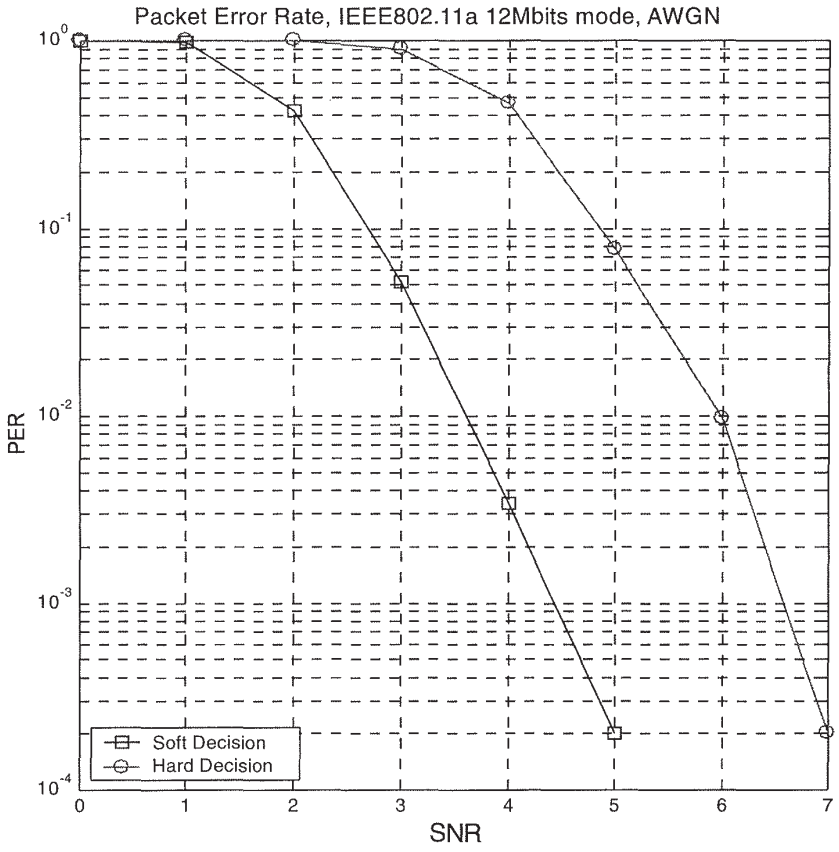


FIGURE 3.17

Packet error rate of IEEE 802.11a 12Mbits/s mode in AWGN channel using soft and hard decision decoding.

The value of the path loss coefficient is dependent on the environment and a large number of studies have been performed to determine its value in various cases [13]. The basic case is free space where $c = 2$; in an indoor environment, a value $c = 3$ can be used. We can use these two values to get an estimate of the range difference between the lowest and highest data rate of IEEE 802.11a. When $c = 2$, every doubling of range decreases the signal power to $\frac{1}{4}$ or by 6 dB, hence

the 17 dB SNR range means approximately 7.2 factor in range. Therefore the 6 Mbits/s mode can function for a distance 7.2 times larger than 54 Mbits/s. The absolute range depends on many things, like transmitter power and receiver design. When we assume the indoor value $d = 3$, the transmitted power decreases to $\frac{1}{8}$ or by 9 dB for every doubling of range. This implies that range difference from 6 to 54 Mbits is about 3.7.

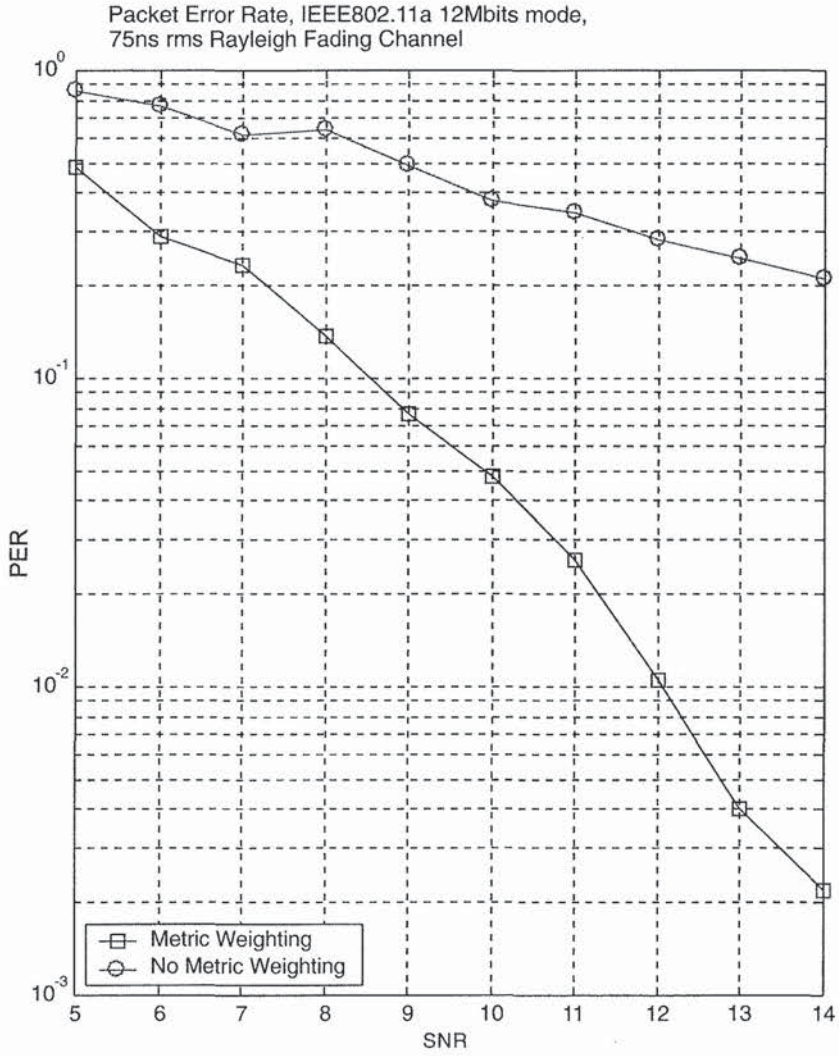


FIGURE 3.18

Packet error rate effect of metric weighting in Viterbi decoding of convolutional code for 12Mbits/s mode.

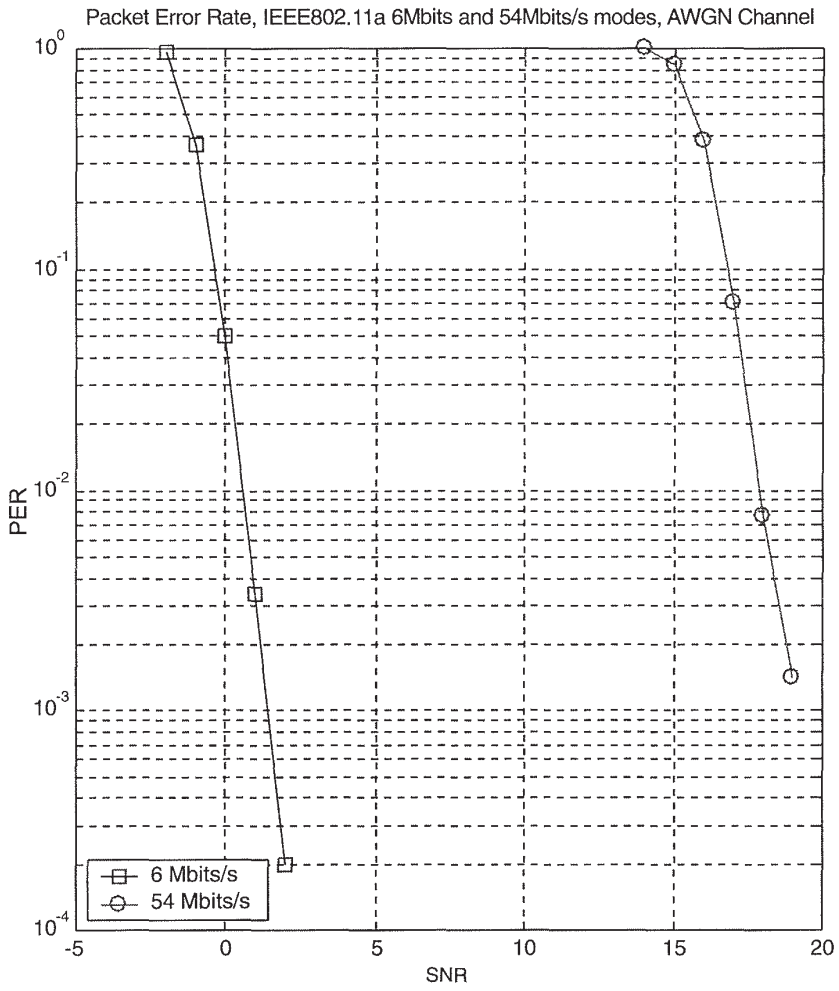


FIGURE 3.19
Packet error rate of IEEE 802.11a 6Mbits/s and 54 Mbits/s modes.

Exercise 3 *The performance of an OFDM system depends on the length of the impulse response of the channel. Simulate different rms delay spread values and note the PER performance. What is the explanation for the results?*

Exercise 4 *Simulate the 9 Mbits/s and 12 Mbits/s modes in both AWGN and fading channel. Compare the results and explain them using the convolutional code free distance and minimum distance of the constellations.*

Trellis Coded Modulation

Trellis coded modulation (TCM) was discovered by Ungerboeck [19]. Trellis Coded Modulation is the main subject of the book by Biglieri et al. [2]. The main idea of TCM is to merge channel coding and modulation into a single integrated component. The benefit of this approach is that the code design is optimized for the used constellation. The most significant benefits of this approach are reached in AWGN channel and with high spectral efficiencies; in other words, with large constellations. One of the first commercial applications of TCM was high speed telephone modems that use very large constellation sizes to reach data rates up 33 kbits/s over the telephone line.

A Trellis Coded Modulation encoder consists of two parts: a convolutional encoder and a modulator. Figure 3.20 shows a basic structure of a TCM encoder. The input bits to encoder are b_0, b_1, and b_2.The convolutional encoder in the figure has rate $\frac{2}{3}$, it encodes input bits b_0 and b_1. In general the convolutional codes used by TCM are of rate $\frac{k}{k+1}$. The three bits, c_0, c_1, and c_2, output from the convolutional encoder enter the modulator with one additional uncoded bit c_3, therefore the overall rate of the TCM code is $\frac{3}{4}$. In general the number of uncoded bits can vary from zero to several bits. These uncoded bits are TCM's Achilles' heel for using it in fading channels with packet radio system. Figure 3.13 showed how the PER really suffers in a fading channel when there are bits, like c_3 in Figure 3.20, that are not protected with channel coding. An error in a single uncoded bit causes a packet error, and the loss of the whole packet. For a practical TCM system to achieve the 54 Mbits/s data rate, it would have to include some uncoded bits in the encoder, thus TCM was not selected as the coding method for IEEE 802.11a. The detrimental effect of the uncoded bits can be reduced by increasing the diversity level of the system. This is discussed in more detail in Chapter 4.

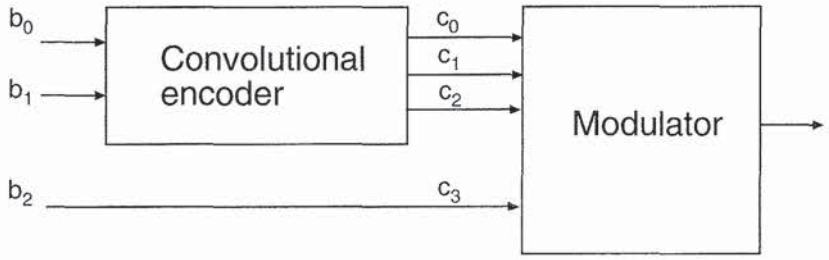


FIGURE 3.20
Trellis coded modulation encoder.

Block Codes

The first error correcting code, discovered by Hamming [5], was a block code. Block codes are different from convolutional codes in the sense that the code has a definite code word length n, instead of variable code word length like convolutional codes. The most popular class of block codes are Reed-Solomon (RS) codes discovered by Reed and Solomon [15]. Another important difference between block codes and convolutional codes is the block codes are designed using algebraic properties of polynomials or curves over finite fields, whereas convolutional codes are designed using exhaustive computer searches. Block codes have not gained wide acceptance in wireless systems, their performance does not reach the level of convolutional codes. The most famous application of Reed-Solomon codes is probably for compact discs that use RS codes to combat burst errors.

Concatenated Codes

Concatenated codes used to be the best performing error correcting codes, before the introduction of turbo codes that are discussed in next section. Concatenated codes are built by combining an outer code and an inner code, shown in Figure 3.21. The outer code is usually a Reed-Solomon block code and the inner code a convolutional code. Concatenated codes have reached performance that is only 2.2 dB from the channel capacity limit.

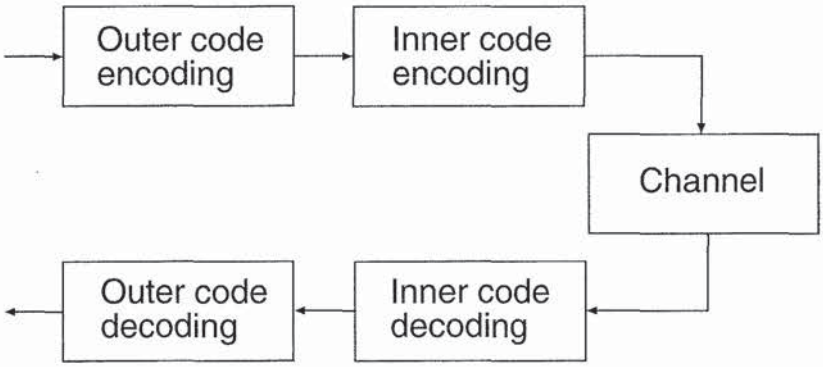


FIGURE 3.21
Basic structure of a concatenated channel coding system.

Turbo Codes

Turbo codes were a major discovery in 1993, when Berrou, Glavieux and Thitimajshima [1] released their results of a code that had performance only 0.6 dB from the channel capacity. Turbo codes are a combination of recursive systematic convolutional (RSC) codes, interleaving, and iterative decoding. Heegard and Wicker [7] provide a thorough treatment of these high performance codes. Figure 3.22 shows a basic structure of a rate $\frac{1}{3}$ turbo code encoder. The encoder

outputs systematic bits b_{3n} that are identical to input bits, and two sets of encoded bits b_{3n+1} and b_{3n+2} that are encoded by different RSC encoders RSC1 and RSC2. The input to the RSC2 is an interleaved version of the input to the RSC1. The interleaver is an important component of a turbo code, and the code performance improves as the size of the interleaver is increased.

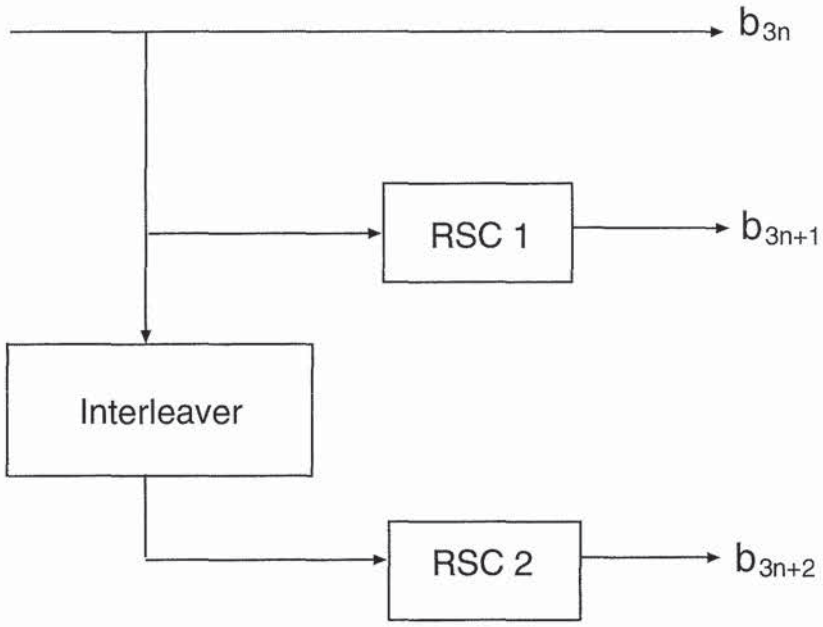


FIGURE 3.22
Rate 1/3 turbo code encoder structure.

Hoshyar, Jamali, and Bahai [8] studied a packet radio system with the same OFDM parameters as IEEE 802.11a, but replaced the channel code with turbo code. Surprisingly the results in a fading channel did not show similar performance gains as AWGN results would suggest, BER was stated to have equal performance, and PER performance was improved by 2.0 dB compared to the convolutional code.

Bibliography

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," 1993 IEEE International Conference on Communications, Geneva, Volume: 2, 1993

[2] E. Biglieri, D. Divsalar, P. J. McLane, M. K. Simon, "Introduction to Trellis-Coded Modulation with Applications," Macmillan Publishing Company, New York, 1991

[3] J. B. Cain, G. C. Clark Jr., and J.M. Geist, "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Maximum Likelihood Decoding," IEEE Transactions of Information Theory, Vol. IT-25, pp. 97-100, January 1979

[4] D. Forney, L.-F. Wei, "Multidimensional constellations. I. Introduction, figures of merit, and generalized cross constellations," IEEE Journal on Selected Areas in Communications, Volume: 7 Issue: 6, Aug. 1989, pp: 877–892

[5] R. W. Hamming, "Error Detecting and Error Correcting Codes," Bell Systems Technical Journal, Vol. 29, pp. 147–160, 1950

[6] J. Hagenauer, "Rate Compatible Punctured Convolutional Codes and Their Applications," IEEE Transactions on Communications, Vol. COM-36, pp. 389-400, April 1988

[7] C. Heegard and S. B. Wicker, "Turbo codes," Kluwer Academic Publishers, 1999

[8] R. Hoshyar, S. H. Jamali, A.R.S. Bahai, "Turbo coding performance in OFDM packet transmission," IEEE 51st Vehicular Technology Conference Proceedings, Tokyo, Volume: 2, pp. 805–810, 2000

[9] L. H. Lee, "Error-Control Convolutional Coding," Artech House, 1997

[10] S. Lin and D. J. Costello, "Error Control Coding: Fundamentals and Applications," Prentice-Hall, New Jersey, 2.ed, 1983

[11] J. G. Proakis, "Digital Communications," McGraw-Hill, Boston, 3ed., 1995

[12] J. L. Ramsey, "Realization of Optimum Interleavers," IEEE Transactions on Information Theory, Vol. IT-16, pp. 338-345, May 1970

[13] T. S. Rappaport, "Wireless Communications: Principles and Practice," Prentice Hall, New Jersey, 1995

[14] H. Rohling and K. Brüninhaus, "High Rate OFDM-Modem with quasicoherent DAPSK," IEEE 47th Vehicular Technology Conference, Volume: 3, pp: 2055–2059, 1997

[15] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," SIAM Journal on Applied Mathematics, Vol. 8, pp. 300–304, 1960

[16] C. E. Shannon, "A Mathematical theory of communication," Bell Systems Technical Journal, 27, pp. 379–423, 623-656, 1948

[17] M. K. Simon, W. C. Lindsey, S. M. Hinedi, "Digital Communication Techiques: Signal Design and Detection," Simon & Schuster Trade, 1994

[18] B. Sklar, "Digital Communications: Fundamentals and Applications," Prentice-Hall, New Jersey, 2.ed, 2001

[19] G. Ungerboeck and I. Csajka, "On Improving Data Link Performance by Increasing the Channel Alphabet and Introducing Sequence Coding," 1976 International Symposium on Information Theory, Ronneby, Sweden, June 1976

[20] S. B. Wicker, "Error control coding," Prentice-Hall, New Jersey, 1995

Antenna Diversity

CHAPTER

4

IN THIS CHAPTER

- Background 124
- Receive Diversity 131
- Transmit Diversity 136
- Bibliography 163

In this chapter, we explore the most underutilized resource in current wireless LAN systems, antenna diversity. Recent information theoretic results suggest that there is tremendous capacity potential for wireless communication systems using *antenna diversity*. Diversity is defined as multiple independent channels between transmitters and receivers; hence, antenna diversity or polarization discrimination occurs when the independent paths are spatial in nature. That is, there is sufficient spacing between the antenna elements at the transmitters and receivers such that there is no or very little correlation amongst their respective signals. The antenna diversity can be used to either improve the link performance of a signal or increase data throughput. The latter has been the subject of considerable research since it is generally believed that near Shannon capacity [61] might be achievable. The former has been and continues to be the study of research in error rate improvement. Both of these research areas have focused of various forms of antenna diversity techniques, which combines the multiple paths in such a way to improve the performance—either error rate or data rate—over that for a single transmit and receiver configuration. Throughout the remainder of this chapter, single transmit and receiver configuration shall be referred to as single-input, single-output (SISO) system.

The remainder of this chapter is organized as follows. First, the "Background" section provides the necessary relevant information required for a detailed discussion on antenna diversity. In the section, notation used throughout the remainder of the chapter is introduced. Most importantly, the fundamental limits of capacity that is achievable for various transmitter and receiver configurations are given. Remaining sections of the chapter are dedicated to antenna diversity techniques. More specifically, receive diversity and transmit diversity are discussed in the context of OFDM systems, respectively. The concepts presented in those sections are not easily grasped. Thus, in an effort to help the learning process, several examples are given throughout the sections to demonstrate particular concepts. This is a slight change from what has been heretofore presented in this text, but the complexity and the importance of this topic warranted additional learning aids. To maintain consistency with Chapters 2, 3, and 5, there are exercises provided that utilize the simulation tool at the Web site. As a final remark, the transmit diversity section is given more attention since additional hardware cost can be more easily accommodated at the access points (AP) rather than the mobile terminal (MT).

Background

In this section, we review the limits of capacity for fading environments, the channel model for multiple-input, multiple-output (MIMO) systems, and the concept of diversity. Each of these topics is needed to help in the understanding of the techniques used in antenna diversity. We begin with a treatment of capacity for fading environments. Foschini and Gans [20] have noted that OFDM systems are particularly well-suited for enhanced capacity using antenna diversity techniques. The report is summarized in the next section.

Limits of Capacity in Fading Environments

The report in [20] was motivated by the need for fundamental understanding of the ultimate limits of bandwidth efficient delivery of higher bit-rates in digital wireless communications. In addition, methods for achieving these limits are examined. The analysis assumes extremely limited mobility and a very slowly changing propagation environment such as a user at a desk within an office space. For this model, the channel is considered "quasi-static"; that is, the channel remains fixed during a burst and is allowed to randomly change from burst to burst. The burst is assumed to be of long enough duration that information theory offers meaningful results. Further, the results presented therein did not assume known channel characteristics at the transmitter, but the receiver knows the channel characteristic perfectly. The analysis is restricted to Rayleigh fading flat channels. Capacity is expressed in units of bits per second per Hertz (bps/Hz), or, equivalently, bits/cycle. The Rayleigh fading assumption stems from the fact that there are a large number of scatters within an indoor environment. Independence of the paths is assumed from the decorrelate on antennas spaced at least an half wavelength apart (polarization discrimination). Moreover, capacity in these analyses is treated as a random variable, and, as such, the *cumulative distribution functions* (cdf) are appropriate for assessing its performance. For instance, if we desire a 99% probability of achieving a particular performance, the cdf graphs provide the capacity curves and one can read off the SNR needed for a 1% outage probability. We briefly define the notation and basic assumptions that will be used throughout the remainder of this chapter:

- Number of transmit antennas and receive antennas are N and M, respectively
- Transmitted signal $s(t)$ is constrained to a total power P regardless of the value N
- The noise at each receiver $v_i(t)$ are independent identical distributed (i.i.d.) additive white Gaussian processes, whose power is equal to N_0
- The average SNR at each receiver is $\gamma = P / N_0$ independent of N
- Matrix channel impulse response $h(t)$ has N columns and M rows. $\mathbf{H}(f)^*$ will be used to denote the Fourier transform of $h(t)$
- Superscript T for vector transpose, superscript † for conjugate transpose, *det* for determinant, *tr* for trace of a matrix, and I_n for the $n \times n$ identity matrix

The received signal $r(t)$ at the ith antenna is given by

$$r_i(t) = h_i(t) \bigstar s_i(t) + v_i(t) \tag{4.1}$$

*Within the text, it will be convenient to drop the dependence of the power transfer characteristic on frequency for notational simplicity. The reader should assume that the discussion pertains to a particular frequency and uniformly applies to the other frequencies as well.

where ★ denotes linear convolution. The discretized versions of continuous variables such as $r(t)$, $s(t)$, and $h(t)$ will be denoted by sample time indexed variables such as $r(l)$, $s(l)$, and $h(l)$, respectively. For notational convenience, the dependence on the sampling time has been suppressed, e.g., $r(lT) \rightarrow r(l)$, where T is the sampling period.

Returning to the main discussion of this section, the standard capacity formula given by Shannon [61] can be expressed in bps/Hz as

$$C = \log_2\left(1 + \gamma|\mathbf{H}|^2\right) \quad bps\ /\ Hz \tag{4.2}$$

where H is just a complex scalar in this case.

For N statistically independent data streams with equal power and Gaussian distributed, the capacity becomes

$$C = \log_2\left[\det\left(I_M + \frac{\gamma}{N}\mathbf{H}\mathbf{H}^{\dagger}\right)\right] \quad bps\ /\ Hz \tag{4.3}$$

It is instructive to examine Equation 4.3 when $H = I_n$, i.e., parallel independent Gaussian channels; hence, the AWGN channel is found to be

$$C = n\log_2\left(1 + \frac{\gamma}{n}\right) \rightarrow \gamma\ /\ \ln(2) \quad as\ n \rightarrow \infty \tag{4.4}$$

Unlike the fading channels model (see Equation 4.2), the capacity scales linearly with increasing SNR for AWGN channels rather than logarithmically. Equation 4.4 hints at the benefits of parallel orthogonal transmission of independent information. One should, however, be careful not to forget that, in general, the various paths in a fading environment might not be orthogonal. In fact, there will tend to be correlation amongst the paths and thus they will interfere with each other. The equality in Equation 4.4 can be achieved by simply dividing the power equally among n independent paths and sending n equal rate independent signals.

When the channel is not AWGN but rather fading, diversity can be employed at the receiver; see the section on "Receive Diversity" for more details. It can be shown that multiple reception from a single transmitter can increase capacity. At the receiver, a linear combination of the antenna outputs are performed such that information contained in the input signal is maximized. This algorithm is called the *maximal ratio combining* (MRC) and is discussed in the "Maximal Ratio Combining" section of this chapter. The capacity resulting from MRC techniques is given by

$$C = \log_2\left(1 + \gamma\sum_{i=1}^{M}|\mathbf{H}_i|^2\right) \quad bps\ /\ Hz \tag{4.5}$$

where $\mathbf{H}_i$ is the channel matrix for the ith receive antenna.

Another receive diversity technique, which is commonly used but not optimum, is *selection diversity*. Selection diversity selects the receive antenna with the highest SNR The capacity offered by selection diversity is

$$C = \max_m \log_2\left[1+\gamma|\mathbf{H}_m|^2\right] = \log_2\left[1+\gamma \max_m |\mathbf{H}_m|^2\right], \quad m \in \{1, M\} \tag{4.6}$$

As implied previously, selection diversity is inferior to MRC diversity, yet it is much simpler to implement. Hence, it is attractive to use in that sense.

A brief point of clarification is in order here. Throughout the analysis presented heretofore in this section, the random channels considered have been Rayleigh channel models, which means that the elements of the M x N channel matrix $\mathbf{H}$ are i.i.d. complex, zero-mean, unit-variance Gaussian random variables. Further, it is well-known that the sum of the squares of k i.i.d. squared zero-mean unit-variance Gaussian random variables is a chi-squared random variable with k degrees of freedom.

The following equations examine some special Rayleigh channels using chi-squared variates.

Example 1 No Diversity: $M = N = 1$.

$$C = \log_2\left[1+\gamma\chi_2^2\right] \tag{4.7}$$

So, the parameter γ multiplies a chi-squared variate with two degrees of freedom.

Example 2 Receive Diversity: $M = \mathrm{n}$, $N = 1$.

$$C = \log_2\left[1+\gamma\chi_{2n}^2\right] \tag{4.8}$$

Contrast Equation 4.8 with the capacity for selection diversity expressed using chi-squared variates,

$$C = \log_2\left[1+\gamma \max\left(\chi_2^2\right)\right] \tag{4.9}$$

Example 3 Transmit Diversity: $M = 1$, $N = n$.

$$C = \log_2\left[1+\frac{\gamma}{n}\chi_{2n}^2\right] \tag{4.10}$$

Compared with Equation 4.7, one can see that transmit diversity DOES NOT improve the SNR by $10\log_{10}(n)$, which is a common misconception. This would be true if we had not constrained

the overall transmit power independent of the number of transmitting antennas N. In other words, the average SNR over all channel realizations is the same at each receive antenna.

Example 4 Combine Transmit and Receive Diversity: It is required only that $N = M$, however, $N \geq M$ is easily accommodated, so it is included as well. The lower bound on the combined transmit and receive diversity case is given by

$$C = \sum_{k=N-(M-1)}^{N} \log_2\left[1 + \frac{\gamma}{N}\chi^2_{2k}\right] \tag{4.11}$$

Contrast Equation 4.11 with

$$C_{optimum} = \sum_{i=1}^{N} \log_2\left[1 + \frac{\gamma}{N}\chi^2_{2(M)_i}\right] \tag{4.12}$$

where i indexes statistically independent chi-squared variates each with $2M$ degrees of freedom. The capacity indicated by Equation 4.12 can only be achieved when the channel state information is known perfectly at both the transmitter and receiver simultaneously. The subject will be further pursued in the section "Water-Filling for Multi-Antenna Systems."

Example 5 Cyclic Delay Diversity: For this case, one transmitter of N transmit antennas is used at a time and detected by M receive antennas. All N transmit antennas are cycled through with period N. The cyclic mechanism avoids catastrophic interference and is simple to implement. The capacity formula for the cyclic delay diversity is given by

$$C = \frac{1}{N}\sum_{i=1}^{N} \log_2\left[1 + \gamma\chi^2_{2(M)_i}\right] \tag{4.13}$$

The first three examples, Equations 4.7–4.10, are simple applications of the generalized capacity formula and are easily derived. Equations 4.11 and 4.13 are more complicated, and interested readers are directed to reference [20] for the details of the proofs of the two formulas.

Exercise 1 *First, using the simulation tool provided at the Web site, determine the required SNR needed to transmit QPSK symbols encoded with the rate 1/2 convolutional code over a Rayleigh fading with reliability of 10^{-6} BER rate. Second, using Equation 4.2, determine the capacity in a Rayleigh fading for that SNR. Third, again using the capacity formula just mentioned, determine the SNR needed to obtain 1 bps/Hz. Finally, what can be said about the coding scheme, which corresponds to the 12 Mbps mode of the IEEE 802.11a standard, in terms of its efficiency with regards to optimum capacity?*

Exercise 2 *Which of the IEEE 802.11a standard modes is closest to optimum capacity in Rayleigh fading? Which is closest in AWGN?*

Channel Model for Multiple-Input/Multiple-Output (MIMO) Systems

In this section, we provide sufficient detail to generally characterize the channel for any multiple-input multiple-output (MIMO) systems employing an OFDM modulation; that is, we shall fully develop the expression given in Equation 4.1 of the previous section. The final form for the channel model shall be denoted in matrix/vector form for notational convenience. Now let's consider a communication link comprising N transmitter antennas and M receiver antennas that operates in a MIMO channel. Each receiver antenna responds to each transmitter antenna through a statistically independent fading path. The received signals are corrupted by additive noise that is statistically independent among the M receiver antennas and the transmission periods. For ease of presentation, discrete baseband notation shall be used; that is, at sample time index l, the complex symbol, $s_i(l)$, sent by the N transmit antennas and, subsequently, detected by kth receive antenna is denoted as $r_k(l)$. Then $r_k(l)$ can be expressed as

$$r_k(l) = \sqrt{\frac{P}{N}} \sum_{i=1}^{N} h_{ki}(l) \bigstar s_i(l) + v_k(l), \quad 1 \le k \le M \tag{4.14}$$

where $h_{ik}(l)$ is the fading path between the ith transmitter antenna and the kth receive antenna at time index l. The noise samples at time index i, $v_k(l)$, are complex zero-mean spatially and temporally white Gaussian random variable with variance $N_0 / 2$ per dimension. It is further assumed that the transmitted signals $s_i(l)$ are normalized such that average energy for the signal constellation is unity. Recall in "OFDM WLAN Overview," in Chapter 1, for OFDM signal, the channel is made circulant by prepending a cyclic prefix (CP) to the data sequence prior to transmission. This simple mechanism allows for replacing the linear convolution in Equation 4.14 with a circular one. Hence, the frequency domain representation of Equation 4.14 using an L-point FFT is given by

$$R_k(m) = \sqrt{\frac{P}{N}} \sum_{i=1}^{N} \mathbf{H}_{ki}(m)\mathbf{S}_i(m) + V_k(m), \quad 1 \le m \le L \tag{4.15}$$

where $R_k(m)$, $\mathbf{H}_{ki}(m)$, $\mathbf{S}_i(m)$, and $V_k(m)$ denote the frequency domain representations of mth subcarrier of the received signal for the kth antenna, complex channel gains between ith transmitter antenna and the kth receive antenna, transmitted signal for the ith antenna, and noise signal for the kth receive antenna, respectively. If the channel gains are slowly fading, then it can be assumed that during a period of T time index the gains are constant and approximate the channel as a block fading channel. Thus, the received signal vector for the mth subcarrier using matrix notation is given by

$$\mathbf{R}(m) = \sqrt{\frac{P}{N}}\mathbf{H}(m)\mathbf{S}(m) + \mathbf{V}(m) \tag{4.16}$$

where

$$\begin{aligned}
\mathbf{R}(m) &= [R_1(m), R_2(m), \cdots, R_M(m)]^T \\
\mathbf{H}(m) &= \begin{bmatrix} H_{11}(m) & H_{12}(m) & \cdots & H_{1N}(m) \\ H_{21}(m) & \ddots & \cdots & H_{2N}(m) \\ \vdots & & \ddots & \vdots \\ H_{M1}(m) & \cdots & & H_{MN}(m) \end{bmatrix} \\
\mathbf{S}(m) &= [S_1(m), S_2(m), \cdots, S_N(m)]^T \\
\mathbf{V}(m) &= [V_1(m), V_2(m), \cdots, V_M(m)]^T
\end{aligned} \tag{4.17}$$

Hence, total received signal vector, again using matrix notation, can be written as

$$\tilde{\mathbf{R}} = [\mathbf{R}(1), \mathbf{R}(2), \cdots, \mathbf{R}(L)] \tag{4.18}$$

which is simply a collection of the frequency domain received signals into a matrix.

Given this necessary mathematical framework, we can proceed to describe popular antenna diversity techniques.

Introducing Diversity

One way to improve the system performance over fading channels is to change its statistical characteristics. This is generally done by a technique called *diversity*. In fact, the effects of fading can be substantially mitigated through the use of diversity techniques in such systems via appropriately designed signal processing algorithms at both the transmitter and receiver. Practical, high-performance systems require that such diversity techniques be efficient in their use of resources such as power, bandwidth, and hardware cost; in addition, they often must meet stringent computational and delay constraints.

Three main forms of diversity are traditionally exploited in communication systems for fading channels: temporal, spectral, and spatial diversity. Temporal diversity is effective when the fading is time-selective. The degree to which this form of diversity can be exploited depends on delay constraints in the system relative to the coherence time of the fading process, which, in turn, is a function of, e.g., vehicle speeds in mobile applications [53]. "Interleaving" in Chapter 3 provides a detailed discussion of time diversity techniques.

Spectral diversity is effective when the fading is frequency-selective, i.e., it varies as a function of frequency. This form of diversity can be exploited when the available bandwidth for transmission is large enough that individual multipath components can begin to be resolved. Examples of systems that take advantage of frequency diversity are direct-sequence spread-spectrum communication, frequency-hopped spread-spectrum communication, and—the subject of this book—OFDM systems. The benefits of frequency diversity for OFDM have been documented in earlier chapters—"OFDM WLAN Overview" in Chapter 1 and "Interleaving" in Chapter 3—of this text while the spread-spectrum systems, on the other hand, are beyond the scope of this book.

Even in situations where the fading channel is nonselective, i.e., neither time selective nor frequency selective, or when system constraints preclude the use of these forms of temporal or spectral diversity, spatial diversity can be used to provide substantial improvement in system performance. Spatial diversity involves the use of multiple antennas sufficiently well-separated at the receiver and/or the transmitter that the individual transmission paths experience effectively independent fading. The extent to which this form of diversity can be exploited depends on issues such as cost and physical size constraints.

The two broad categories for antenna diversity are transmit and receive diversity.

We treat receiver diversity first because it is the simpler of the two concepts, and the remainder of the chapter is dedicated to transmit diversity.

Receive Diversity

The use of multiple antennas at the receiver, which is referred to as receive diversity, is fairly easily exploited. In essence, multiple copies of the transmitted stream are received, which can be efficiently combined using appropriate signal processing techniques. As the number of antennas increases, the outage probability is driven to zero, and the effective channel approaches an additive Gaussian noise channel. The two most popular receive diversity techniques are selection and maximal ratio combining.

Selection Diversity

The simplest receive diversity is selection diversity. Because of its simplicity, current IEEE 802.11b WLAN products employ selection diversity at the mobile terminal (MT) and access point (AP). Given M receive antennas, selection diversity entails choosing the receive antenna with the largest SNR in each symbol interval. Another attractive feature of selection diversity is that it does not require any additional RF receiver chain. In other words, all the receive antennas share a single RF receiver chains, which is key drive in keeping the cost down for the MT equipment.

The performance of selection diversity is easily computed by assuming each receive antenna is subject to *i.i.d.* Gaussian noise. Thus, selecting the antenna with the largest SNR is tantamount to selecting the receive antenna with the largest instantaneous power. Further, it assumes that the average reliability of each of the paths is equal. Thus, average SNR $\bar{\gamma}$ amongst the receive antennas is given by

$$\bar{\gamma} = \frac{\bar{\varepsilon}}{N_0} = E\left(h_l^2\right)\frac{\varepsilon}{N_0}, \quad l = 1,\ldots,M \tag{4.19}$$

Naturally, the instantaneous SNR γ_l is therefore given by

$$\gamma = h_l^2\frac{\varepsilon}{N_0}, \quad l = 1,\ldots,M \tag{4.20}$$

The probability that the SNR for the lth receive antenna is lower than a threshold z is given by

$$\Pr(\gamma_l \le \mathsf{z}) = \int_0^{\mathsf{z}} f_{\gamma_l} \tag{4.21}$$

where $f_{\gamma_l}(\beta)$ denotes the probability density function (pdf) (see "Review of Stochastic Processes and Random Variables" in Chapter 1 for discussion) of γ_l, which is assumed to be the same for all antennas. With M independent receive antennas, the probability that all of them have an SNR below the threshold z is given by

$$\Pr(\gamma_1 \le \mathsf{z},\ldots,\gamma_M \le \mathsf{z}) = [\Pr(\gamma_l \le \mathsf{Z})]^M \tag{4.22}$$

and decreases as M increases. Not surprising, this is also the cdf of the random variable

$$\breve{\gamma} = \max\{\gamma_1,\ldots,\gamma_M\} \tag{4.23}$$

Hence, $\breve{\gamma}$ is less than z if and only if (iff) $\gamma_1,\ldots,\gamma_M$ are all less than z. Therefore the pdf follows directly from the derivative of the cdf with respect to z. This result can be used to obtain the error probability of a digital modulation scheme in the presence of selection diversity by integrating the conditional error probability with respect to the $\breve{\gamma}$. To assess the benefit of selection diversity, we consider a QPSK modulation signal in a five-tap Rayleigh fading environment with 1, 2, and, 4 receive antennas. The error probability as a function of SNR is given in Figure 4.1.

From Figure 4.1, it is clearly evident that error rate performance improves with increasing number of receive antennas. Note for a BER of 1%, the improvement in error rate performance is between 2-3 dB for two receive antennas and 3-4 dB for four receive antennas. The caveat to this performance improvement is that each path for the receive antennas must be independent of the others.

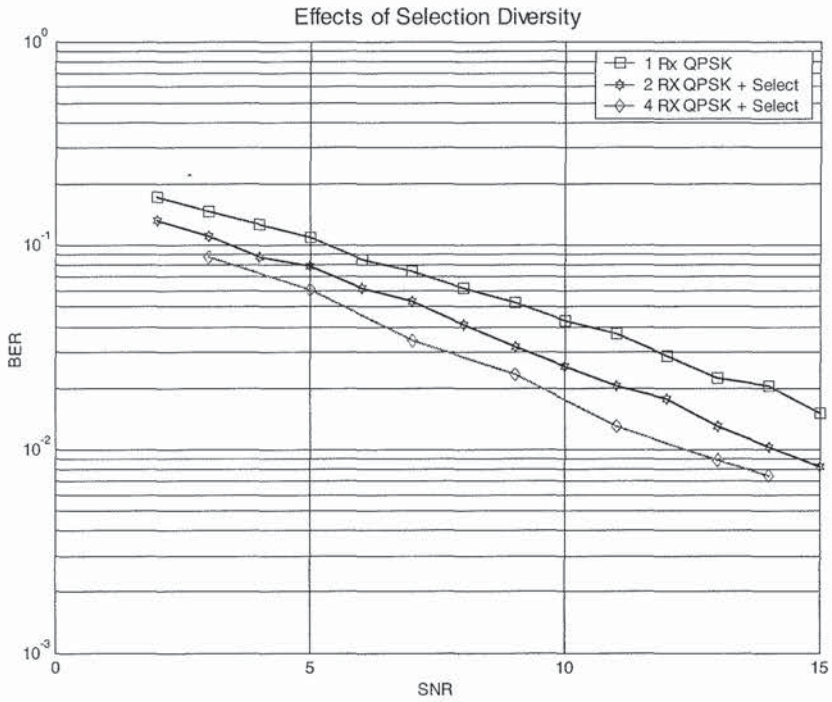


FIGURE 4.1
Error rate performance for selection diversity in Rayleigh fading.

To close, selection diversity is relatively easy to implement, as noted earlier, because it requires only a measure of the powers from each receive antenna and a switch. However, the simple observation that it disregards the information from all antennas but one leads us to conclude that is not an optimum combining technique.

Maximal Ratio Combining

In maximal ratio combining (MRC), the signals at the output of the M receive antennas are combined linearly so as to maximize the instantaneous SNR. The coefficients that yield the maximum SNR are found from straightforward application of optimization theory. Let's denote the received signal per antenna r_l as

$$r_l = h_l s_l + v_l \tag{4.24}$$

with independent noise samples v_l and have the same power spectral density $2N_0$. Further assume perfect channel state information at each antenna. Finally, the transmitted signal s_l has been normalized such that the average signal energy equal 2ε.

MRC consists of using the linear combination

$$y = \sum_{l=1}^{M} w_l^* r_l = \sum_{l=1}^{M} w_l^* h_l s_l + \sum_{l=1}^{M} w_l^* v_l \tag{4.25}$$

prior to detection. The power spectral density S_v of the noise after MRC is given by

$$S_v = 2N_0 \sum_{l=1}^{M} |w_l|^2$$

while the instantaneous signal energy is

$$2\varepsilon \left| \sum_{l=1}^{M} |w_l^* h_l| \right|^2$$

The ratio of these two quantities

$$\gamma = \frac{\varepsilon \left| \sum_{l=1}^{M} |w_l^* h_l| \right|^2}{N_0 \sum_{l=1}^{M} |w_l|^2}$$

can be maximized as follows. Recall the Cauchy-Schwartz inequality defined as

$$\left| \sum_{l=1}^{M} |a_l^* b_l| \right|^2 \leq \sum_{l=1}^{M} |a_l|^2 \sum_{l=1}^{M} |b_l|^2 \tag{4.26}$$

where the equality in Equation 4.26 is obtained for $w_l = h_l$ for all l, which provides the weighting coefficients for MRC. In plain English, each antenna is weighted proportionally by its corresponding fading attenuation. The heavily faded antennas, which are less reliable, are counted less than the less faded antennas, which are more reliable, and vice versa. The SNR provided by MRC is given by

$$\gamma_{MRC} = \frac{\varepsilon}{N_0} \sum_{l=1}^{M} |w_l|^2 \tag{4.27}$$

Noting that $\varepsilon |w_l|^2 / N_0$ is the SNR per antenna, then Equation 4.27 is just the sum of the SNRs for each antenna, which means γ_{MRC} can be large even when the individual SNRs are small. As an example of the improvement results from MRC, consider the same scenario as before with selection diversity. Rather than using selection diversity, we instead use MRC. The error performance of this system is shown in Figure 4.2.

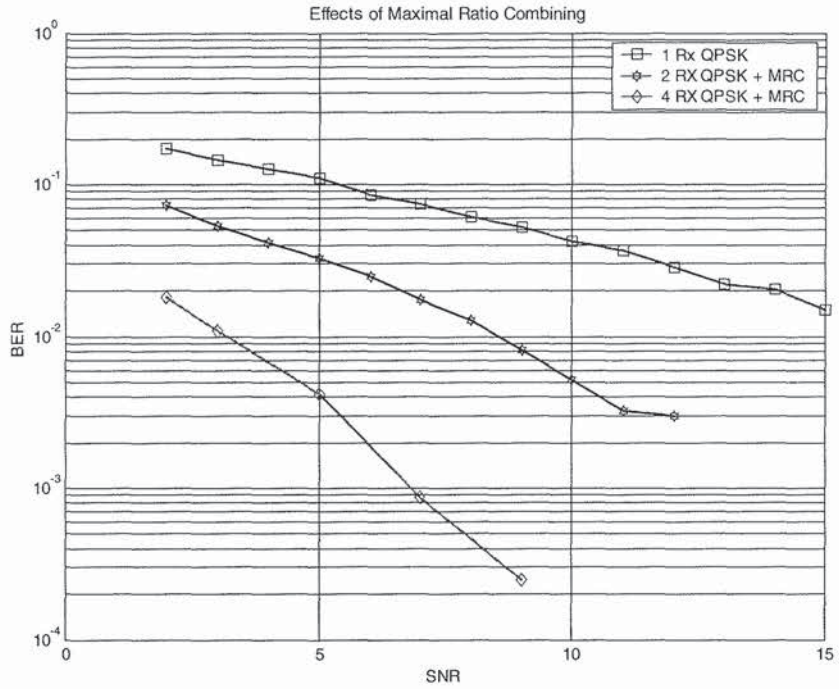


FIGURE 4.2
Error rate performance for MRC in Rayleigh fading.

The performance improvement is much more significant for MRC than selection diversity. For instance, at a BER of 1%, MRC with two receive antennas provides nearly 10 dB improvement, while for selection diversity, it was found to be between 2-3 dB. For four receive antennas, the improvement is nearly 15 dB!

In summary, the potential gain using MRC can be tremendous. This is not surprising since it is the optimum solution. The caveat to this is that perfect channel knowledge was assumed, which is not true in general. The accuracy of the channel estimation will depend somewhat on the operating SNR. The sensitivity of MRC to channel estimation error is left as an exercise to the readers.

Exercise 3 *Consider the case when the weights for the MRC are all equal to unity, referred to as equal gain diversity. How does this receive diversity technique compare to selection diversity and MRC? Modify the MRC module given at the Web site to perform this task and plot on the same graph selection diversity, equal-gain diversity, and MRC for the case scenario given in the selection diversity example.*

Transmit Diversity

In the previous section, receive diversity techniques were reviewed. However, receive diversity can be impractical in a number of applications. In such scenarios, the use of multiple antennas at the transmitter, which is referred to as transmit diversity, is significantly more attractive. Multiple element transmitter antenna arrays have an increasingly important role to play in emerging wireless LAN networks, particularly at the AP. Indeed, when used in conjunction with appropriately designed signal processing algorithms, such arrays can dramatically enhance performance. This last statement begs the question, "How are good transmit diversity schemes designed?"

Transmit Diversity Design Criteria for Fading Channels

In this section, we explore different design criteria for transmit diversity schemes. As will be explained later, the design criteria depend heavily on how much is known about the channel at the transmitter and receiver. In general, a wireless communication system is comprised of N transmit antennas and M receive antennas. The information data is encoded using a channel code to protect it against imperfections caused by the channel. The encoded data stream is split into N parallel streams, each of which is modulated and then transmitted using separate antennas. Each path for the separate antennas is assumed *i.i.d.* and quasi-static; that is, the complex gains of the paths are constant over each data frame but changes from frame to frame.

We begin our analysis of transmit diversity design criteria by examining the single antenna case and then generalize the results for the multiple antenna case in the Trellis Space-Time Codes section. To develop codes that perform well over fading channels, we choose to minimize the pairwise error probability. That is, the probability that codeword $\mathbf{c}$ is transmitted over the channel and a maximum-likelihood (ML) receiver decides in favor of a different codeword $\mathbf{e}$. The ML receiver makes decoding decisions based on a performance metric m ($\mathbf{r}$, $\mathbf{c}$, $\mathbf{h}$) provided that estimates of the fading amplitudes, $\mathbf{h}$, are available at the receiver. Formally, the maximum likelihood criterion for the optimum decoder requires the conditional probability of receiving $\mathbf{r}$ given that codeword $\mathbf{c}$ was transmitted to be greater than the probability of receiving $\mathbf{r}$ assuming any other codeword $\mathbf{e}$ was transmitted, i.e.,

$$P(\mathbf{c} \rightarrow \mathbf{e} \mid \mathbf{h}) = \Pr[m(\mathbf{r}, \mathbf{e}; \mathbf{h}) \geq m(\mathbf{r}, \mathbf{c}; \mathbf{h}) \mid \mathbf{h}] \tag{4.28}$$

where

$$m(r_i, e_i; h_i) = |r_i - h_i e_i|^2 \tag{4.29}$$

and $|\cdot|^2$ represents the squared Euclidean norm. Note that since the performance metric $m(\mathbf{r}, \mathbf{c}; \mathbf{h})$ is just the distance between vectors, the total metric is simply the sum of the distances per dimension $m(r_i, e_i; h_i)$. The pairwise error probability is found by taking the statistical expectation of

Equation 4.28. Rather than solving for the exact pairwise error probability, which can only be evaluated numerically, an upper bound can be found for Equation 4.28 by using the Chernoff bound techniques [25]

$$P(\mathbf{c} \to \mathbf{e}) \le \mathbf{E}_h\{\exp(\lambda[m(\mathbf{r}, \mathbf{e}; \mathbf{h}) - m(\mathbf{r}, \mathbf{c}; \mathbf{h})])\} \tag{4.30}$$

$$P(\mathbf{c} \to \mathbf{e}) \le \prod_{l=1}^{L} E_{h_l}\left\{\exp\left(\lambda\left[|r_l - h_l c_l|^2 - |r_l - h_l e_l|^2\right]\right)\right\} \tag{4.31}$$

where $\lambda \ge 0$ is the Chernoff parameter to be optimized for the tightest bound. The equality in Equation 4.31 is obtained by invoking the additive property of the metric. The optimum Chernoff parameter to form the tightest bound can be found by first rewriting the multiplicands in Equation 4.31 as

$$\exp\left(\lambda\left[|r_l - h_l c_l|^2 - |r_l - h_l e_l|^2\right]\right) = \exp\left[-\lambda h_l^2 (1 - N_0\lambda)|c_l - e_l|^2\right] \tag{4.32}$$

Now, the optimum Chernoff parameter λ_{opt} is found by taking the derivative of the argument in Equation 4.32 with respect to λ to yield

$$\lambda_{opt} = \left(\frac{1}{2N_0}\right) \tag{4.33}$$

Substituting Equation 4.33 into Equation 4.32 results in a optimized Chernoff bound

$$P(\mathbf{c} \to \mathbf{e}) \le \prod_{l=1}^{L} E_{h_l}\left\{\exp\left(-\frac{h_l^2}{4N_0}|c_l - e_l|^2\right)\right\} \tag{4.34}$$

Rician Channels

To evaluate the bound for a Rician channel, we average the result of Equation 4.34 over the Rician pdf [54] to obtain

$$P(\mathbf{c} \to \mathbf{e}) \le \prod_{l=1}^{L} \int_0^{\infty} 2h_l(1+\kappa)\exp\left(-\left[\kappa + h_l^2(1+\kappa) + h_l^2\delta^2\right]\right) \tag{4.35}$$

$$I_0\left(2h_l\sqrt{k(1+\kappa)}\right)\delta h_l \tag{4.36}$$

where

$$\delta^2 = \frac{|c_l - e_l|^2}{4N_0}$$

and κ is the Rician factor defined as the ratio of energy in the spectral path to the diffuse path. This integral can be evaluated using [34]

$$\int_0^\infty t\exp\left(-at^2\right)\cdot J_0(\lambda t)\partial t = \frac{1}{2\alpha}\exp\left(\frac{-\gamma^2}{4a}\right) \tag{4.37}$$

where

$$J_0(jt) = I_0(t)$$

and $I_0(t)$ is the zeroth order modified Bessel Function [34] and $j = \sqrt{-1}$.

Thus the pairwise error probability for a Rician channel is given by

$$P(\mathbf{c}\to\mathbf{e}) \le \prod_{l=1}^{L}\frac{1+\kappa}{1+\kappa+\frac{1}{4N_0}|c_l-e_l|^2}\exp\left[\frac{-\kappa\frac{1}{4N_0}|c_l-e_l|^2}{1+\kappa+\frac{1}{4N_0}|c_l-e_l|^2}\right] \tag{4.38}$$

which simplifies for high signal-to-noise ratios (SNR) to

$$P(\mathbf{c}\to\mathbf{e}) \le \prod_{l\in\eta}\frac{(1+\kappa)\exp[-\kappa]}{\frac{1}{4N_0}|c_l-e_l|^2} \tag{4.39}$$

where η is the set of all l for which $c_l \neq e_l$. Denoting the number of elements in η as L_η, Equation 4.39 can be rewritten as

$$P(\mathbf{c}\to\mathbf{e}) \le \frac{\left((1+\kappa)\exp[-\kappa]\right)^{L_\eta}}{\left(\frac{1}{4N_0}d_P^2(L_\eta)\right)^{L_\eta}} \tag{4.40}$$

where $d_P^2(L_\eta)$ is the so-called squared product distance along the error event $(\mathbf{c},\mathbf{e})$ path defined as

$$d_P^2(L_\eta) = \left(\prod_{l\in\eta}|c_l-e_l|^2\right)^{\frac{1}{L_\eta}} \tag{4.41}$$

and L_η is the *effective length* of the error event $(\mathbf{c},\mathbf{e})$.

Rayleigh Channels

For Rayleigh channels, κ is equal to zero and Equation 4.40 simplifies to

$$P(\mathbf{c}\to\mathbf{e}) \le \prod_{l\in\eta}\frac{1}{\frac{1}{4N_0}|c_l-e_l|^2} \tag{4.42}$$

Among all terms in Equations 4.40 and 4.42, the terms with the smallest effective length L_η and smallest product distance $d_P^2(L_\eta)$ dominate the error event probability at high SNR. Let's denote the minimum L_η by L and the corresponding squared product distance as $d_P^2(L)$ thus the error event probability for Rician channels is asymptotically approximated by

$$P(\mathbf{c} \to \mathbf{e}) \equiv \alpha\left(L, d_P^2(L)\right) \frac{\left((1+\kappa)\exp[-\kappa]\right)^L}{\left(\frac{1}{4N_0} d_P^2(L)\right)^L} \tag{4.43}$$

and, for Rayleigh channels,

$$P(\mathbf{c} \to \mathbf{e}) \equiv \alpha\left(L, d_P^2(L)\right) \frac{1}{\left(\frac{1}{4N_0} d_P^2(L)\right)^L} \tag{4.44}$$

where $\alpha(L,\ d_P^2(L))$ is the average of number of code sequences having effective length L and squared product distance $d_P^2(L)$. Equation 4.44 says that the error event probability has two dominant factors: the product distance and its multiplicity—average number of occurrences $\alpha(L,\ d_P^2(L))$. So, not only must the product distance $d_P^2(L)$ be maximized, the number of low symbol Hamming weight code sequences $\alpha(L,\ d_P^2(L))$ must be kept as small as possible as well.

Delay Diversity

Just as selection diversity is the simplest of the receive diversity techniques, delay diversity is the simplest of the transmit diversity techniques. Consider a repetition code of length of N—a symbol is repeated N times on N transmit antennas. The N symbols are transmitted over the N different antennas at non-overlapping time instants or frequency bins of a DFT, i.e.,

$$\mathbf{S} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ 0 & s_1 & s_2 & s_3 & s_4 \\ 0 & 0 & s_1 & s_2 & s_3 \end{bmatrix} \tag{4.45}$$

where the rows correspond to transmit antennas and the columns correspond to symbol intervals. The received signal $r(l)$ is given by

$$r(l) = \sum_{i=1}^{N} \sum_{j=1}^{p} h_i(j) S_i(l-j) + v(l) \tag{4.46}$$

where the subscript ith indexes the individual paths. If this diversity code is subject to a frequency-selective channel, a minimum mean squared error (MMSE) decision feedback equalizer or a maximum likelihood sequence estimator (MLSE) is needed to resolve the multipath

distortion. When implemented in an OFDM framework, the OFDM received signal for the kth subcarrier $R(k)$ becomes

$$R(k) = \sum_{i=1}^{N} H_i(k)S_i(k) + V(k) \tag{4.47}$$

where $H_i(k)$, and $S_i(k)$ are the frequency domain representations for the kth subcarrier and ith path of the channel impulse response and transmit signal, respectively; $V(k)$ is the additive noise process at the receiver. Assuming perfect knowledge of the channel at the receiver, the receiver must solve the following optimization problem for each subcarrier k,

$$\varepsilon_k = \min \left\| R(k) - \mathbf{H}(k)^{\dagger} \mathbf{S}(k) \right\|^2 \tag{4.48}$$

$$\mathbf{H}(k) = [H_1(k), H_2(k), \ldots, H_N(k)]^T \tag{4.49}$$

$$\mathbf{S}(k) = [S_1(k), S_2(k), \ldots, S_N(k)]^T \tag{4.50}$$

The MLSE requires a search over all possible transmit sequences and finds the sequence that yields the minimum of Equation 4.48.

Exercise 4 *Develop a Matlab script file using Equation 4.45 for a two-antenna delay diversity code within an OFDM frame with symbols drawn from a QPSK constellation. The receiver function shall be an MLSE, which requires that for every subcarrier, the sixteen possible combinations for two QPSK symbols are to be evaluated in Equation 4.48. Integrate the delay diversity encoder and decoder into the simulation environment at our Web site and compare the performance for this code with that of selection diversity and MRC with two receive antennas.*

Trellis Space-Time Codes

The foregoing analysis in the "Transmit Diversity Design Criteria for Fading Channels" section leads to a set of design parameters for good space-time codes (STC) over fading channels. To characterize these parameters, we consider the error event probability of STC over fading channels as the performance metric. Assuming ideal channel state information at the receivers and multiple transmit and receive antennas, the pairwise error probability, determined from the Chernoff bound in Equation 4.34, is given by

$$P(\mathbf{c} \to \mathbf{e}) \leq \exp\left(\frac{-d^2(\mathbf{c}, \mathbf{e})E_s}{4N_0} \right) \tag{4.51}$$

where $d^2(\mathbf{c}, \mathbf{e})$ represents the normalized accumulative distance between codewords $(\mathbf{c}, \mathbf{e})$ and is defined as

$$d^2(\mathbf{c},\mathbf{e}) = \sum_{j=1}^{m}\sum_{l=1}^{L}\left|\sum_{i=1}^{n} h_{i,j}(c_i(l) - e_i(l))\right|^2 \tag{4.52}$$

Using matrix notation, Equation 4.52 can be expressed as

$$d^2(\mathbf{c},\mathbf{e}) = \sum_{j=1}^{m} \mathbf{h}_j^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})\mathbf{h}_j \tag{4.53}$$

where h_j is again the channel coefficient vector defined as

$$\mathbf{h}_j = \left[h_{1,j}, h_{2,j}, \cdots, h_{n,j}\right] \tag{4.54}$$

$\mathbf{D}(\mathbf{e},\mathbf{c})$ is the codeword difference matrix defined as

$$\mathbf{D}(\mathbf{e},\mathbf{c}) = \begin{bmatrix} c_1(0) - e_1(0) & c_1(1) - e_1(1) & \cdots & c_1(L) - e_1(L) \\ \vdots & \ddots & \vdots & \vdots \\ c_n(0) - e_n(0) & c_n(1) - e_n(1) & \cdots & c_n(L) - e_n(L) \end{bmatrix}^T \tag{4.55}$$

T denotes matrix transpose. Thus, substituting Equation 4.53 into Equation 4.51, the pairwise error probability in matrix notation becomes

$$P(\mathbf{c} \to \mathbf{e}) \le \prod_{j=1}^{M} \exp\left(\frac{-\left[\mathbf{h}_j^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})\mathbf{h}_j\right]E_s}{4N_0}\right) \tag{4.56}$$

The matrix $\mathbf{D}(\mathbf{e},\mathbf{c})^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})$ is Hermitian; thus, it be can expressed in terms of a unitary matrix $\mathbf{V}$ whose columns are its eigenvectors $\mathbf{v}_i$ and a diagonal matrix Λ containing its eigenvalues, i.e.,

$$\mathbf{h}_j^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})^{\dagger}\mathbf{D}(\mathbf{e},\mathbf{c})\mathbf{h}_j = \sum_{i=1}^{n} \lambda_i \left|\mathbf{h}_j^{\dagger}\mathbf{v}_i\right| \tag{4.57}$$

We note that since channel coefficients $h_{i,j}$ are *i.i.d.* by assumption, an orthonormal transformation of the channel coefficients, $\beta_{i,j} = \mathbf{h}_j^{\dagger}\mathbf{v}_i$, is also *i.i.d.* Hence, we arrive at the upper bound on the average probability of error with respect to the $\beta_{i,j}$ for Rician and Rayleigh channels, respectively, by substituting Equation 4.57 into Equation 4.53 and using Equations 4.38 and 4.42,

$$P(\mathbf{c} \to \mathbf{e}) \le \prod_{j=1}^{m}\prod_{i=1}^{p} \frac{(1+\kappa_{i,j})}{1+\kappa_{i,j}\frac{E_s}{4N_0}\lambda_i} \exp\left[-\frac{\kappa_{i,j}\frac{E_s}{4N_0}\lambda_i}{1+\kappa_{i,j}\frac{E_s}{4N_0}\lambda_i}\right] \tag{4.58a}$$

$$P(\mathbf{c} \to \mathbf{e}) \le \left(\frac{E_s}{4N_0}\right)^{-pm} \left(\prod_{i=1}^{r} \lambda_i\right)^{-m} \tag{4.58b}$$

where p is the rank of the codeword difference matrix $\mathbf{D}(\mathbf{e},\mathbf{c})$. Inspection of (4.58b) leads to the design criteria we sought.

- **The Distance/Rank Criterion:** In order to achieve a diversity of pm in a rapid fading environment for any two codewords $\mathbf{c}$ and $\mathbf{e}$, the strings $c_1(l)\ c_2(l)\cdots c_n(l)$ and $e_1(l)\ e_2(l)\cdots e_n(l)$ must be different at least for p values of $1 \le l \le n$. In other words, the rank of $\mathbf{D}(\mathbf{e},\mathbf{c})$ must be p to achieve a diversity of pm in a quasi-static flat fading environment.
- **The Product/Determinant Criterion:** Recall that η is the set of all l for which $c_l \ne e_l$. and let $\left|\mathbf{c}_l - \mathbf{e}_l\right|^2 = \sum_{i=1}^{n}\left[c_i(l) - e_i(l)\right]$. Then to achieve the most coding gain in a rapid fading environment, the minimum squared product distance $d_P^2(L_\eta)$ over η, as defined in Equation 4.41, must be maximized. For the case of a static fading channel, the minimum of pth roots of the sum of determinants of all $p\,x\,p$ principal cofactors of $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})$ taken over all pairs of distinct codewords $\mathbf{c}$ and $\mathbf{e}$ corresponds to the coding gain.

Improvement of the STC Design Criterion

D.M. Ionescu [43] demonstrated that the determinant criterion can be strengthened by requiring that the eigenvalues of $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})$ be as close as possible, for any pair codewords $\mathbf{c}$, $\mathbf{e}$. Formally, the criterion was expressed as follows.

> **The Equal Eigenvalue Criterion:** For N-transmit antenna system operating in i.i.d. Rayleigh fading with perfect channel state information (CSI), an upper bound to the pairwise error probability is made as small as possible *iff*, for all pair codewords $\mathbf{c}$, $\mathbf{e}$, the squared Euclidean distance $tr\left[\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})\right]$ are made as large as possible. Further, the non-square matrices $\mathbf{D}(\mathbf{e},\mathbf{c})$ are semi-unitary—up to a scale factor—, i.e. $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c}) = \left(tr\left[\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})\right]/N\right) * I_N$. Essentially, maximizing $\min_{\mathbf{c},\mathbf{e}} \det\left[\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})\right]$, as specified by the Determinant Criterion, requires maximizing the minimum eigenvalue product over all $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})$.

Proof By Hadamard's theorem, the eigenvalue product for a square, positive definite matrix $\mathbf{A}$, with elements $a_{i,j}$ assumes its maximum value $\Pi_i a_{i,i}$, *iff* $\mathbf{A}$ is diagonal. Once $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})$ is diagonalized, the product of its diagonal elements is maximized *iff* they are rendered equal and their sum, $tr\left[\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger\mathbf{D}(\mathbf{e},\mathbf{c})\right]$, is maximized. Consider the arithmetic-mean geometric mean inequality [34], i.e.,

$$\sqrt[n]{\prod_{i=1}^{n} \lambda_i} \le \left(\frac{1}{n}\sum_{i=1}^{n} \lambda_i\right) \tag{4.59}$$

The astute reader will recognize that the left hand side of Equation 4.59 has the same form as the product distance given in Equation 4.41. Hence, the product distance $d_P^2(L)$ is upper bounded by the arithmetic mean of the eigenvalues of $\mathbf{D}(\mathbf{e},\mathbf{c})^\dagger \mathbf{D}(\mathbf{e},\mathbf{c})$ with equality achieved when all eigenvalues are equal.

D.M. Ionescu recognized that it might be difficult to enforce this condition for all pairs of codewords **c**, **e** and thus proposed a sub-optimal solution to enforce the condition of the codewords corresponding to the shortest error event paths in the code trellis. Shortly, we will see that this new Determinant Criterion relates to block STC.

Using the original design criteria for STC, Tarokh *et al* [65, 67, 69] were able to generate a family of codes, which have received considerable attention. Tarokh *et al* claimed that these codes achieve near optimum *outage capacity* (rather than Shannon's capacity). Outage capacity is defined as the maximum constant information rate that can be maintained in all fading conditions with some nonzero probability of outage, assuming perfect receiver CSI. For these environments, it is more appropriate to use outage capacity rather than Shannon's capacity since some channels yield zero capacity using Shannon's formula. Hence, it is not a meaningful measure of the performance of these environments. Examples of outage capacity are when all the phone lines at your job are busy and your cell phone cannot get service. Hopefully, this doesn't occur to you very often. In fact, most communication systems strive to have greater than 99% availability to its customer or less than 1% outage probability. Previously reported performance results in [67], compared to outage capacity are reprinted here for inspection. The trellis structures examined are shown in Figures 4.3–4.5. The performance curves for these trellises are given for inspection and their corresponding performance in Figures 4.6–4.9.

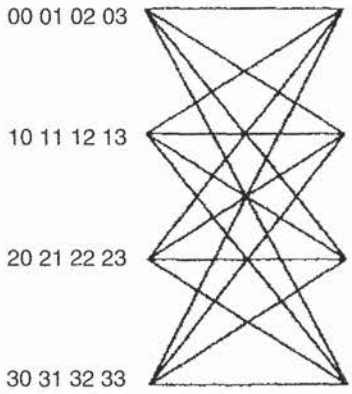


FIGURE 4.3
A fully connected four state STC using QPSK. © 1998 IEEE.

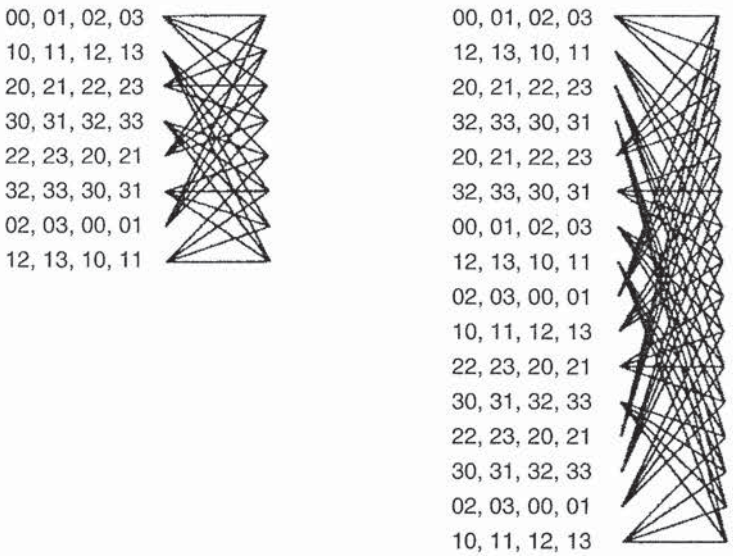


FIGURE 4.4
Trellises for 8 state and 16 state with diversity order 2. © 1998 IEEE.

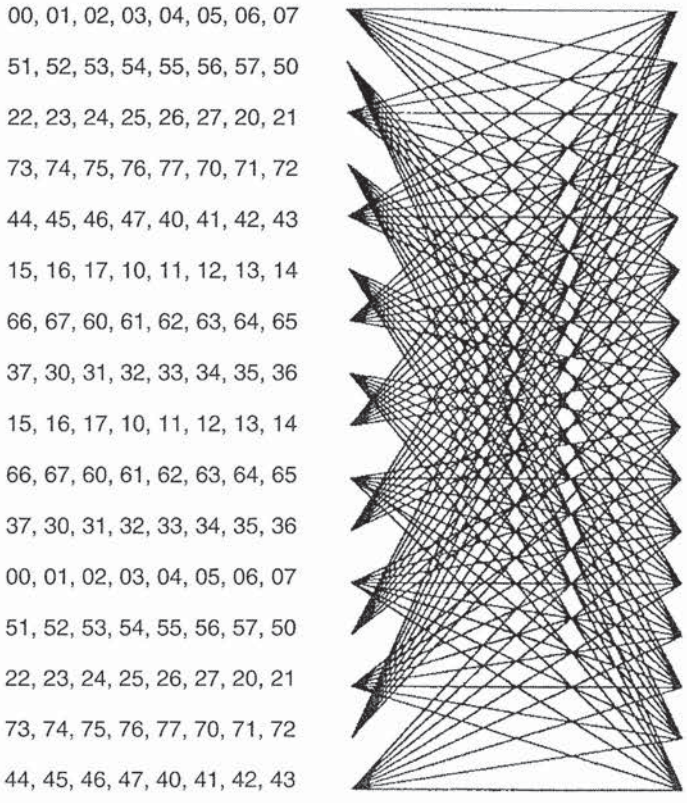


FIGURE 4.5
Trellis for 16 states, 3 bps/Hz STC. © 1998 IEEE.

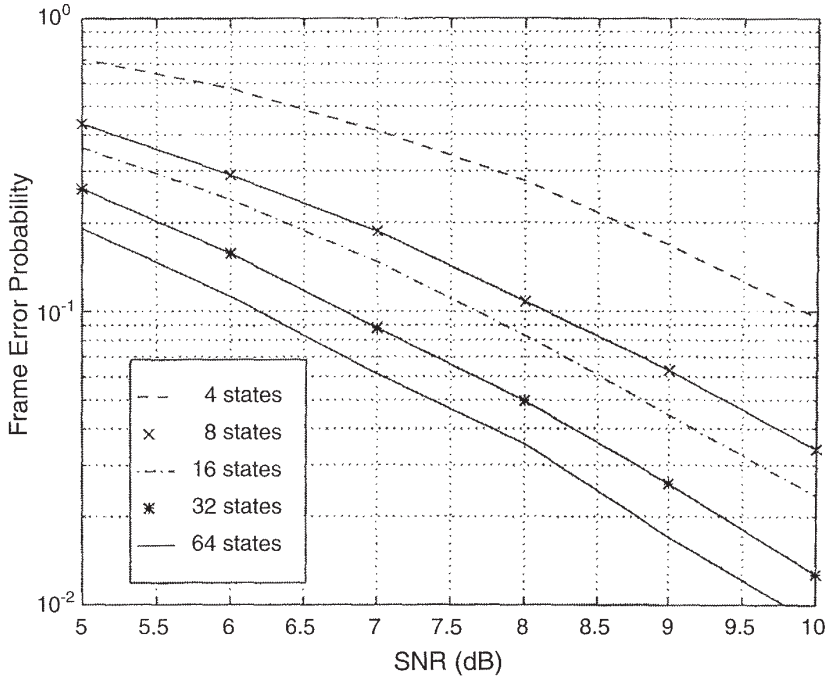


FIGURE 4.6
Codes for QPSK with rate 2 bps/Hz that achieve diversity order 4 with 2 rx and 2 tx antennas. © 1998 IEEE.

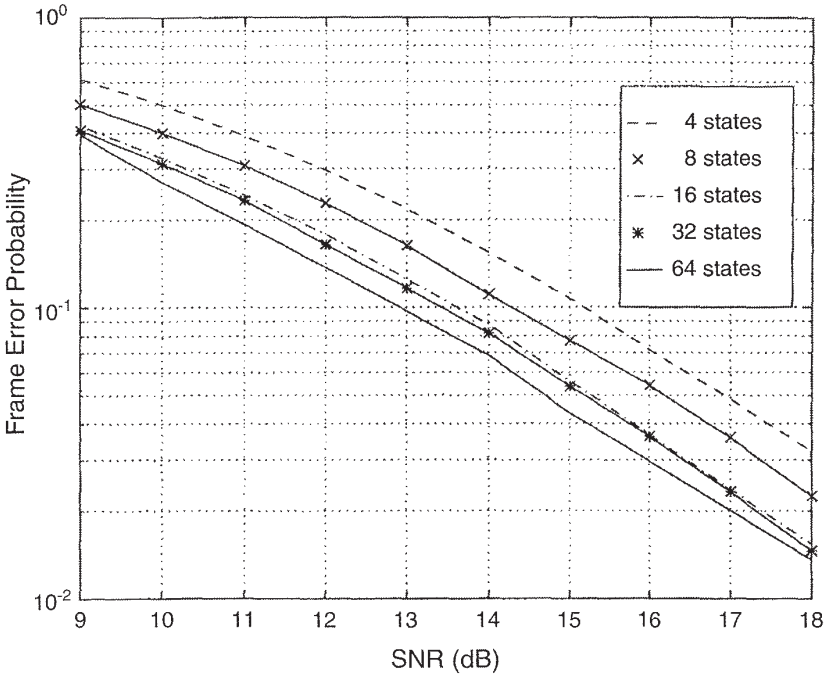


FIGURE 4.7
Codes for QPSK with rate 2 bps/Hz that achieve diversity order 2 with 1 rx and 2 tx antennas. © 1998 IEEE.

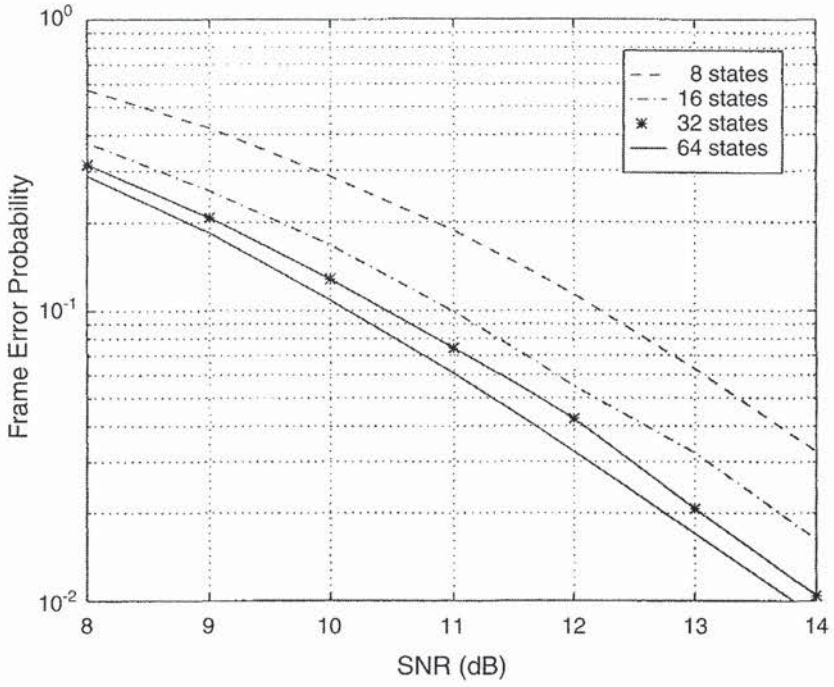


FIGURE 4.8
Codes for 8-PSK with rate 3 bps/Hz that achieve diversity order 4 with 2 tx and 2 rx antennas. © 1998 IEEE.

Layered Space-Time Codes

The Layered Space-Time Processing approach to STC was first introduced by Foschini [21] at Bell Labs. The basic concept behind layered STC is demultiplexing information bits into individual streams in a hierarchal manner, which are then fed into individual encoders. The outputs of the coders are modulated and fed to separate antennas, from which they are transmitted, using the same carrier frequency and symbol waveform. At the receiver, a spatial beam-forming process is used at the front end in order to separate the individual coded streams, and feed them to their individual decoders. After decoding the most important bits, they are used to decode less important bits and this goes till all the information is decoded.

Foschini considered this problem in [21]. He proposed a multilayered structure that in principle can achieve a tight lower bound on the capacity. If N transmit and N receive antennas are used, then at the receiver the transmitted signal from transmit antenna *1* is treated as the desired signal, while the signals transmitted from other transmit antennas are treated as interference. Linear processing is then used to suppress the interfering signals using N receive antennas, providing a diversity gain of one. Once the signal transmitted from antenna *1* is correctly detected, the signal transmitted from antenna *2* is treated as the desired signal while the signals transmitted from transmit antennas $3, 4, \cdots, N$ are treated as interference. Linear processing is then applied to suppress the interfering signals using N receive antennas. This provides a diversity gain of two. This

process is repeated until all the transmitted signals are detected. Clearly, the dominant diversity in this architecture is associated with the first estimated signal. In this light, long frames of data combined with powerful coding techniques are needed to achieve the lower bound on outage capacity. The diversity level can be improved albeit at the expense of losing half of the rate [21].

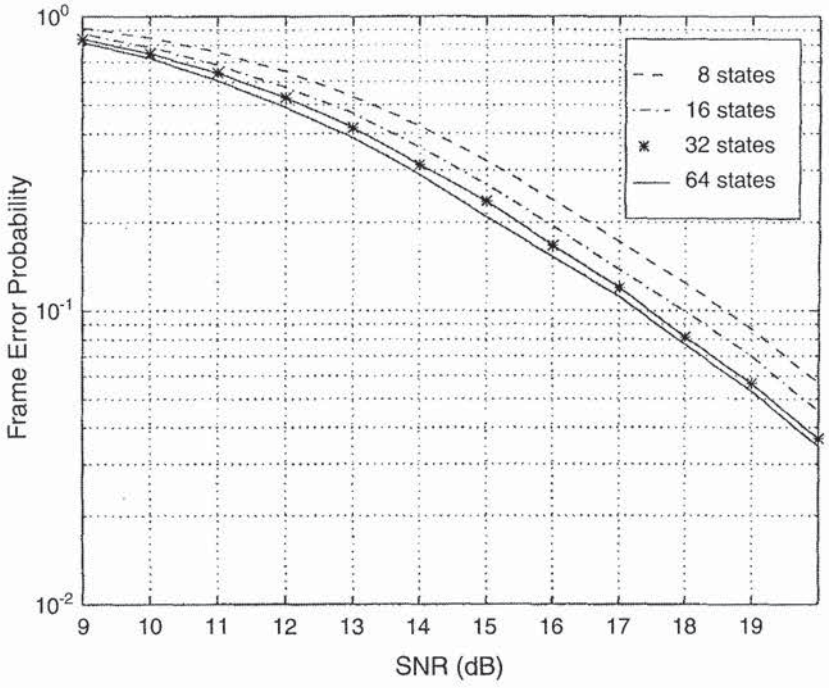


FIGURE 4.9
Codes for 8-PSK with rate 3 bps/Hz that achieve diversity order 2 with 1 rx and 2 tx antennas. © 1996 IEEE.

This approach is purely a signal processing one. Tarokh [65] attacked the problem from an array signal processing combined with channel coding perspective. Antennas at the transmitter are partitioned into small groups, and individual STCs, called the component codes, are used to transmit information from each group of antennas. At the receiver, an individual STC is decoded by a novel linear array processing technique, called the group interference suppression method [65], that suppresses signals transmitted by other groups of antennas by treating them as interference. Since the diversity in each decoding stage l is more than that of the previous decoding stage $l-1$, the transmit power out of each antenna at level l can be substantially less than that of the previous layer. Thus the transmitter should divide the available transmit power among different antennas in an unequal manner. Power allocation for this scenario is straightforward. In fact, powers at different levels could be allocated based on the diversity gains. In this way, the allocated powers may decrease geometrically in terms of the diversity gains.

This combination of array processing at the receiver and coding techniques for multiple transmit antennas provides reliable and very high data rate communication over wireless channels. One obvious advantage of the group interference suppression method over the architecture of Foschini [21] is that the number of receive antennas can be less than the number of transmit antennas [65].

Block Space-Time Codes

Trellis space-time coding combines signal processing at the receiver with coding techniques appropriate to multiple transmit antennas. Specific trellis space-time codes designed for two and four transmit antennas [67] perform extremely well in slow-fading environments (typical of indoor transmission) and come close to the outage capacity. However, when the number of transmit antennas is fixed, the decoding complexity of trellis STCs—determined by the number of trellis states in the decoder—increases exponentially with transmission rate.

In addressing the issue of decoding complexity, Alamouti [2] discovered a remarkable scheme for transmission using two transmit antennas. This scheme is much less complex than trellis space-time coding for two transmit antennas, but there is a loss in performance compared to trellis STCs. Despite this performance penalty, Alamouti's scheme is appealing in terms of simplicity and performance, and it motivated a search for similar schemes using more than two transmit antennas.

The novelty of Alamouti's block STC is the simple decoding system. Although it is shown [68] that the spectrum efficiency of this method is not as good as trellis STCs, the simplicity of the decoder is the motivation to use this method in practical systems. Maximum-likelihood decoding is achieved in a simple way through decoupling of the signals transmitted from different antennas rather than joint detection. This uses the orthogonal structure of the block STC and gives a maximum likelihood decoding algorithm, which is based only on linear processing at the receiver. Block STC are designed to achieve the maximum diversity order for a given number of transmit and receive antennas subject to the constraint of having a simple decoding algorithm.

A block STC is defined by an $N \times T$ transmission matrix X where its entries are linear combinations of the symbols $s_1, s_2, \cdots, s_k$ and their conjugates. It is designed in such a way that to get the maximum diversity gain of NM, but the coding gain is sacrificed to have a simple ML decoder. In fact, a block STC assigns a space-time signature to each symbol and its conjugate in such a way that these signatures generate the maximum diversity for all symbols. This condition results from the orthogonality property induced between signatures; hence, ML decoding is performed for symbols $s_1, s_2, \cdots, s_k$ independently. This results in a simplified decoder structure, which consists of a parallel bank of k simple ML decoders [66, 68].

The main properties of block space-time codes could be summarized as follows:

- There is no loss in bandwidth, in the sense that orthogonal designs provide the maximum possible transmission rate at full diversity.

- Because there is no coding gain obtained by block STC, the spectral efficiency of this method is less than trellis STC codes.
- There is an extremely simple maximum likelihood decoding algorithm which only uses linear combining at the receiver. The simplicity of the algorithm comes from the orthogonality of the columns of the orthogonal design. The above properties are preserved even if linear processing is performed at the transmitter. Therefore, the definition for orthogonal designs can be relaxed to allow linear processing at the transmitter. Signals transmitted from different antennas will now be linear combinations of constellation symbols.

Examples of real orthogonal designs are the 2×2

$$X = \begin{pmatrix} s_1 & s_2 \\ -s_2 & s_1 \end{pmatrix} \tag{4.60}$$

and the 4×4

$$X = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2 & s_1 & -s_4 & s_3 \\ -s_3 & s_4 & s_1 & -s_2 \\ -s_4 & -s_3 & s_2 & s_1 \end{pmatrix} \tag{4.61}$$

A point worth noting is that block STC guarantees satisfaction of the Equal Eigenvalue Criterion by M. Ionescu. Furthermore, the symbols $s_1, s_2, \cdots, s_k$ can be chosen to maximize the Euclidean distance for all pair codewords **c**, **e**. To compensate for poor spectral efficiency compared to trellis STC code, the symbols $s_1, s_2, \cdots, s_k$ can be generated from a spectrally efficient modulation such as trellis coded modulation (TCM) or block coded modulation (BCM). Researchers J. Terry *et al* [72] used this combination to produce a very high rate OFDM system. For OFDM systems, block STC can also be performed over the whole OFDM symbol. Consider the 2×2 complex block STC popularized by Alamouti [2], also known as the Radon Hurwitz (R-H) unitary transform defined as

$$R - H\begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{bmatrix} -\mathbf{s}_1^* & \mathbf{s}_2 \\ \mathbf{s}_2^* & \mathbf{s}_1 \end{bmatrix}$$

If two consecutive OFDM symbols are referred to as $\mathbf{S}_o$ and $\mathbf{S}_e$, then at the first antenna $\mathbf{S}_o$ is transmitted in the first time epoch followed by $\mathbf{S}_e$ in the second time epoch while at the second transmitter $-\mathbf{S}_e^*$ is transmitted in the first time epoch followed by $\mathbf{S}_o^*$ in the second time epoch. The appeal, as mentioned earlier, of the block STC schemes is the ease in which decoding can be performed compared to trellis STCs. Denote the diagonal matrices containing the discrete Fourier Transforms (DFT) of the channel response vectors for transmitters 1 and 2 as $\mathbf{H}_1$ and $\mathbf{H}_2$,

respectively. Assuming the channel is constant over the two consecutive time epochs, the demodulated received signals $(\mathbf{R}_1, \mathbf{R}_2)$ in the respective time slots are given by

$$\mathbf{R}_1 = -\mathbf{H}_1\mathbf{S}_0^* + \mathbf{H}_2\mathbf{S}_e^* \tag{4.62}$$

$$\mathbf{R}_2 = \mathbf{H}_1\mathbf{S}_e + \mathbf{H}_2\mathbf{S}_0 \tag{4.63}$$

Using simple substitution methods, the estimates of the consecutive OFDM symbols are

$$\hat{\mathbf{S}}_0 = \mathbf{H}_2^*\mathbf{R}_2 - \mathbf{H}_1\mathbf{R}_1^* \tag{4.64}$$

$$\hat{\mathbf{S}}_e = \mathbf{H}_1^*\mathbf{R}_2 + \mathbf{H}_2\mathbf{R}_1^* \tag{4.65}$$

Substituting Equations 4.62 and 4.63 into Equations 4.64 and 4.65 yield

$$\hat{\mathbf{S}}_0 = \left(|\mathbf{H}_1|^2 + |\mathbf{H}_2|^2\right)\mathbf{S}_0 \tag{4.66}$$

$$\hat{\mathbf{S}}_e = \left(|\mathbf{H}_1|^2 + |\mathbf{H}_2|^2\right)\mathbf{S}_e \tag{4.67}$$

Clearly from Equations 4.66–4.67, we understand why block STC technique improves the performance of TCM over fading channels. The envelope of the estimated symbols $\hat{\mathbf{S}}_0$ and $\hat{\mathbf{S}}_e$ are chi-squared variants with four degrees of freedom. Moreover, the chi-squared variates normalized by the number of degrees of freedom approach unity as the number of degrees of freedom approach infinity. In practice, this normalized chi-squared variates are nearly unity with four degrees of freedom. Another way to interpret this is that as the number of degrees of freedom increases, the randomness of the chi-squared variates decreases and, in the limit, approaches a constant. Thus, the code design should be targeted for an AWGN channel.

To summarize, both the Equal Eigenvalue Criterion and Rank Criterion are satisfied when block STC are concatenated with TCM. Since Ungerbock TCM codes are well known to be the best in terms of Euclidean distance, very powerful diversity codes are easily achieved using these two methods [3, 33].

For applications where good diversity performance on the base station link is required, trellis STCs seem to offer the best performance amongst the STC architectures. Yet, block space-time coding (STC) offers the best trade-off of performance versus complexity while layered STCs offer the simplest decoder complexity at the expense of poorest performance. This is largely due to the loss of diversity, which arises due to the antenna beamforming process, and error propagation effect.

Exercise 5 *Using the simulation tool at the Web site, compare the performances of a 2 × 2 block STC, i.e., Radon-Hurwitz, with QPSK modulation to the performances of two receive*

antenna selection diversity and MRC in five-tap Rayleigh fading chanels. Using Equation 4.61, modify the 2 × 2 block STC function and repeat the experiment above but now compare the performance to four receive antenna selection diversity and MRC.

Multidimensional Space-Time Codes

The design criteria given earlier in the "Transmit Diversity Design Criteria for Fading Channels" section were derived starting with the pairwise error probability. This led to a determinant criteria for code design. To improve the performance of the coding gain of a constituent STC, the minimum squared distance of the signal constellation Δ_0^2 must be increased without increasing the average energy for the signal constellation. One simple way to accomplish this is to increase the dimensionality of the signal constellation.

Example 6 Consider a fully connected, two-antenna space-time trellis with Q states and Q transitions. This simple trellis is designed such that transitions leaving a state differ in the first symbol, and transitions entering a state differ in the second symbol. The codeword difference matrix for the shortest error event path in this trellis is easily shown to be

$$\mathbf{D}(\mathbf{c},\mathbf{e}) = \begin{bmatrix} \Delta_0 & 0 \\ 0 & \Delta_0 \end{bmatrix} \tag{4.68}$$

Clearly, the outer product of matrix $\mathbf{D}(\mathbf{c},\mathbf{e})$ is also diagonal with eigenvalues and geometric mean equal to Δ_0^2. Thus, increasing the minimum distance between codewords also increases the coding gain proportionately for this simple trellis.

In addition to maintaining the average energy, it is also desirable to minimize the peak to average power ratio (PAPR) of the transmitted symbols. A large PAPR requires the input signal power to the amplifier, located at the transmitter, to be "backed-off" from its saturation level, resulting in poor overall DC power efficiency. The combination of these two requirements lead to spherical codes as a natural choice for an OFDM system.

Spherical Coding

A k-dimensional *spherical code* is a set of points in R^k that lie on the surface of a k-dimensional unit radius sphere. Mathematically, the points on the surface of a unit radius k-dimensional Euclidean sphere are defined as

$$\Omega_k = \left\{ \begin{array}{c} \mathbf{x} = (x_1, x_2 \ldots, x_k) \in \mathsf{R}^k: \\ \|\mathbf{x}\|^2 = x_1^2 + x_2^2 + \cdots + x_k^2 = 1 \end{array} \right\} \tag{4.69}$$

Collectively, $(x_1, x_2 \ldots, x_k)$ is referred to as a *k-tuple* . The parameters of interest are *dimension, minimum distance,* and *size.* The dimension is the smallest integer k such that $\Theta \subset \Omega_k$. Usually,

we describe the spherical code Θ with respect to the smallest possible basis, where the dimension equals the number of coordinates. The minimum squared distance of the code Θ is

$$d_{\min}^2(\Theta) = \min_{i \neq j} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \tag{4.70}$$

The maximum theoretically attainable minimum squared distance is the quantity

$$\left(d_{\min}^*\right)^2 = \max_{\Theta} \left(d_{\min}^2\right) \tag{4.71}$$

The size, finally, is simply the number of points in the set Θ denoted by M.

Remark 1 *Any signaling scheme used in code construction should differ in all or as many coordinates as possible to improve diversity. Rotation of a spherical code Θ about its coordinate axes does not change $d_{\min}^2$ for the code. Therefore, diversity for a particular spherical code Θ is optimized when all coordinates differ for each symbol in the k-dimensional space.*

Remark 2 *Ideal coordinate interleaving for a k-dimensional spherical code provides a k-fold improvement in diversity level compared to ideal symbol interleaving.*

Remarks 1 and 2 lead the authors in [71] to believe that concatenation of spherical coding with STC, termed *spherical STC* (SSTC), will provide substantial diversity gains.

SSTC in OFDM Framework

A simplified diagram of SSTC implemented in a conventional OFDM is shown in Figure 4.10. The spherical space-time encoder depicted in Figure 4.10 is comprised of the trellis STC encoder followed by a multidimensional symbol mapper. The latter function maps output code symbols of the trellis space-time encoder to points in R^k that lie on the surface of a k-dimensional unit radius sphere. L complex samples are formed from L pairs of coordinates from p output coded symbols, where

$$p = \left\lceil \frac{2L}{k} \right\rceil$$

and $\lceil \bullet \rceil$ represents the ceiling operation. The complex samples are interleaved and serially fed into the N–pt. IDFT, where $N \geq L$. After the IDFT operation, an cyclic prefix is prepended to the transformed symbols to form an OFDM symbol. Finally, each OFDM is transmitted over a separate antenna in every time epoch.

At the receiver, the cyclic prefix is removed. Next, the remaining complex samples are buffered and fed into a DFT. The output of the DFT is de-interleaved and the pairs of coordinates are

re-grouped into k-tuples. The k-tuples are passed to the spherical space-time decoder that outputs an estimate of the transmitted data stream.

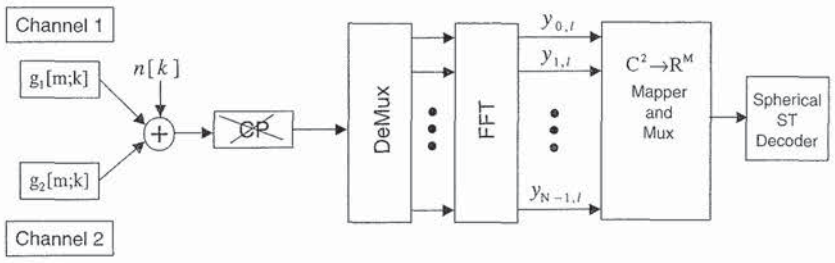


FIGURE 4.10

SSTC encoding in an OFDM framework.

To illustrate the benefits of SSTC, J. Terry and J. Heiskala compared the 24 Mbps mode of the IEEE 802.11a with a SSTC operating also at 24 Mbps. The system parameters for the SSTC were two transmit antennas, one receive antenna, 16-ary spherical code defined over four dimensions, and delay diversity STC trellis, i.e. Q states with Q transitions, which has no coding gain in AWGN channels. The spherical codes used for the signal mapper was found from a computer search based on maximizing the minimum Euclidean distance between symbols and symbol coordinates. The gain for this code is approximately 4.77 dB over uncoded. The systems are evaluated over an AWGN channel and a five-tap slowly fading indoor channel. The channel statistics follow a lognormal distribution and the taps are spaced 50 nanoseconds apart. Figure 4.11 demonstrates that the SSTC outperforms the considered 24 Mbits/sec mode of IEEE 802.11a for the error rates of interest. SSTC benefits from both coding gain and diversity gain as seen from the approximately 4 dB improvement over IEEE 802.11a in indoor fading channels and 2 dB over AWGN channels.

To summarize, the authors in [71] were able to demonstrate that a multidimensional signal mapper can be an effective method for providing both coding and diversity gains with STC. In general, as the dimensionality of the signal mapper is increased, the spectral efficiency decreases but the coding and diversity gains increase. Alternately, lattice codes rather than spherical codes can be investigated as multidimensional signal mappers. Based on the inherent structure of lattice codes, potentially simplified encoder and decoder structures can be conceived.

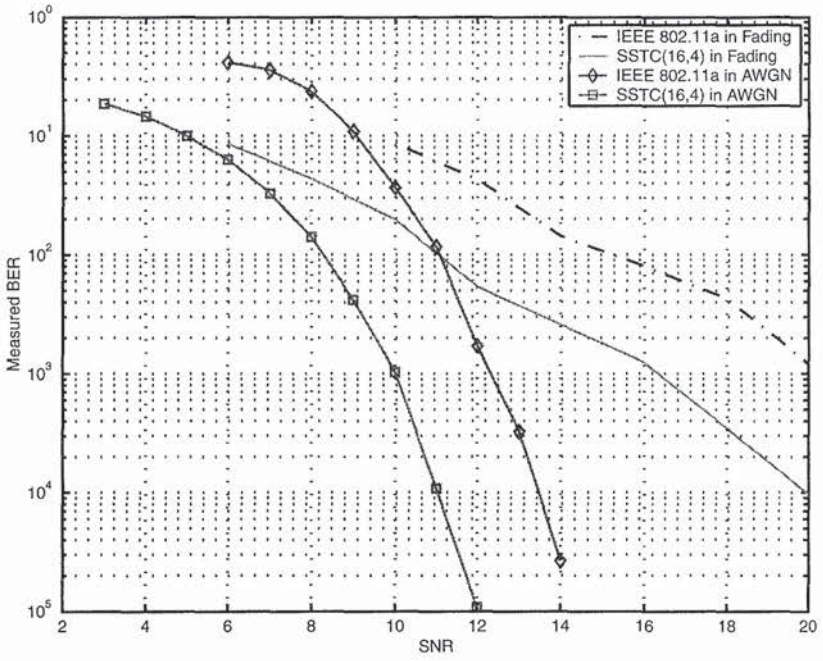


FIGURE 4.11

Performance comparison of 24 Mbit/sec modes for SSTC(16,4) and IEEE 802.11a.

Water-Filling for Single Antenna Systems

In this section, we consider a diversity technique referred to as *water-filling*, which assumes known channel state information at the transmitter. We begin with a review of the single antenna case in this section and generalize to the multi-antenna system in the next section. Note, when the channel frequency response is not ideal, the frequency components of transmitted signals observe different channel gain. This means that the frequency attenuation is not uniform for all the frequency components of the signal. Moreover, in channels which have severe nulls in their frequency response or there is powerful narrow-band noise, the signal frequency components are completely canceled out in those frequencies. It is obvious that transmitting energy in these frequencies is a waste of the power. Therefore, to approach the capacity of these channels, a kind of spectral shaping should be applied to transmitted signal. Correspondingly, new diversity schemes that are adapted to the frequency response of the channel are required. In other words, the output power and information should be distributed over frequency in such a way that achieve maximum capacity. This is the essence of water-filling. Typically, water-filling is performed in two steps: *power allocation* and *bit loading*.

Power Allocation

Let $P(f)$ and $N(f)$ denote the power spectral densities of the transmitted signal and the noise, respectively. Also, let $H(f)$ stand for the channel frequency response. The *channel SNR function* is defined as

$$\gamma(f) = \frac{|H(f)|^2}{N(f)} \tag{4.72}$$

Intuitively, the preferred transmission band is where $\gamma(f)$ is largest. Using information theory, the total capacity of the system is described by:

$$C = \int_B \log_2(1 + P(f)\gamma(f))df \tag{4.73}$$

where B is the channel bandwidth. Maximizing this capacity with respect to $P(f)$ subject to the following power constraint

$$\int_B P(f)df \le P \tag{4.74}$$

and by employing Lagrange multipliers, results in an adaptive power distribution which is called "spectral water-filling" or power allocation and defined by

$$P_{Alloc}(f) = \left[u - \frac{1}{\gamma(f)}\right]^+, \quad f \in B \tag{4.75}$$

where μ is found such that, $P_{Alloc}(f)$ satisfies the power constraint in Equation 4.74 with equality; the function $[\cdot]^+$ clamps all the negative values to zero. Substituting $P_{Alloc}(f)$ in Equation 4.73 results the optimal capacity

$$C_{optimum} = \int_B [\log_2(\mu\gamma(f))]^+ df \tag{4.76}$$

The capacity-achieving band B is the most important feature of the water-filling spectrum. For many practical channels, B is a single frequency interval. However, in other cases, e.g., in the presence of severe narrow-band interference, it may consist of multiple frequency intervals as in an OFDM system.

Bit Loading

Let's assume that the channel has been decomposed into N disjoint subchannels, i.e., $B = N\Delta f$. Let P_i be the power assigned to subcarrier i, $1 \le i \le N$. Then, the total transmitted power is

$$P = \sum_{i=1}^{N} P_i \tag{4.77}$$

It is obviously seen from Equation 4.75 that those subchannels which have higher SNR have been assigned more power and they are able to carry more information. The maximum amount of information in bps/Hz, which is possible to be sent reliably through each subchannel is determined from Equation 4.76. The total number of bits that can be supported by a particular channel realization $H(f)$ is $R_{Total} = C_{optimum}B$. Furthermore, since we are considering digital modulation schemes, we should find a set of integer bits per subchannel R_i such that the total number of bits sent for any channel realization is maximized, i.e.,

$$R_{Total} = \sum_{i=1}^{N} R_i \tag{4.78}$$

with respect to P_i subject to Equation 4.77. The algorithm, which maximizes Equation 4.78 under the stipulated constraints, is referred to as bit loading.

If an M-ary QAM modulation set is combined with a channel code C_i to be used with subcarrier i, it can be shown that for a given error rate ε_i, power P_i, and $\gamma(i)$ for each subchannel, the maximum number of bits that can be transmitted through subchannel i, satisfying the given requirements is defined by [9, 48]

$$R_i = \log_2\left(1 + \frac{\gamma(i)P_i}{\Gamma(C_i, \varepsilon_i)}\right) \tag{4.79}$$

where

$$\Gamma(C_i, \varepsilon_i) = \frac{3}{k_{eff}(C_i)\left[Q^{-1}\left(\frac{\varepsilon_i}{4}\right)\right]^2} \tag{4.80}$$

is called the *capacity gap* , where k_{eff} and ε_i are the effective coding gain and required error rate, respectively. Q in Equation 4.80 denotes the *complementary error function* defined as

$$Q\ (x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\tau^2}{2}} \partial\tau \tag{4.81}$$

Therefore, the *inverse complementary error function* Q^{-1} is given by

$$Q^{-1}(x) = \sqrt{\ln\left(\sqrt{2\pi}\,\frac{\partial Q}{\partial x}\right)^{-2}} \tag{4.82}$$

By increasing the reliability, the capacity gap increases and the spectral efficiency decreases.

To keep the overall error rate as small as possible, all the subcarriers should have the same error rate. Otherwise, the subchannel which has the maximum error rate will dominate. Assuming that the coding gain and error rate over all the subcarriers be the same and denoted by C and ε, respectively; let's define $\Gamma = \Gamma(C, \varepsilon)$ as the capacity gap over all subchannels. Now substituting Equation 4.79 in Equation 4.78, the total number of bits over all subchannels become

$$R_{Total} = \sum_{i=1}^{N} \log_2\left(1 + \frac{\gamma(i)P_i}{\Gamma}\right) \tag{4.83}$$

Maximizing this with respect to P_i's subject to the power constraint in Equation 4.77, we get [8, 19]

$$P_i = \left[\mu - \frac{\Gamma}{\gamma(i)}\right]^+, \quad i \in \{1, \cdots, N\} \tag{4.84}$$

This is similar to Equation 4.75 except the capacity gap that is defined by the required reliability and coding gain. It is analagous to saying that before applying water-filling, the SNR of all the subchannels $\gamma(i)$ should normalized by a constant Γ—indeed, prior even to selecting μ to satisfy the power constraint.

Water-Filling for Multi-Antenna Systems

Now consider a quasi static flat fading MIMO channel described by Equation 4.16. Recall, as mentioned in previous sections, when channel state information (CSI) is not available in transmitter (open loop systems), STCs should be used to get diversity and coding gain. Likewise to the single-input single-output (SISO) systems, when CSI is available in transmitter, we can increase the efficiency of the system optimal power and rate adaptation in transmitter.

Let the singular value decomposition of H be $H = U\Sigma V^H$, where U and V are unitary matrices and Σ be a diagonal matrix with positive real values on diagonal elements standing for the singular values of the channel. If the transmitted vector is premultiplied by V in transmitter and received vector is post multiplied by U^H in receiver, then (4.16) could be rewritten as

$$\tilde{\mathbf{y}} = \Sigma\mathbf{x} + \tilde{\eta} \tag{4.85a}$$

where $\tilde{\mathbf{y}}$ and $\tilde{\eta}$ denote the received vector and noise vector after postmultiplication by U. Note that because U is a unitary matrix, there is no noise amplification and the noise vectors remains spatially white. Thus, the error rate of maximum likelihood decoder remains the same as it was for decoding Equation 4.16, but its complexity is significantly reduced because the entries of $\mathbf{x}$ are decoded separately. If p denotes the rank of Σ, then this MIMO channel is a set of parallel of SISO channels

$$\tilde{y}_k = \sigma_k x_k + \tilde{\eta}_k \quad k = 1, \cdots, p \tag{4.85b}$$

where σ_k is the gain of kth channel and $\tilde{\eta}_k$ is white Gaussian noise.

The total capacity of the MIMO channel is equal to the aggregate capacity of all these SISO channels. Considering that if spatial water-filling is applied for this MIMO channel, the capacity of this MIMO channel is the sum of the capacities described by Equation 4.2. In the previous section, power allocation and bit loading were suggested for coding schemes achieving the maximum capacity over SISO fading channels. Here the capacity could be optimized by spatial-frequency waterfilling and bit loading. The spatial-frequency waterfilling would operate essentially as described in "Water-Filling for Single Antenna Systems." The next section explores potential bit loading algorithms.

Adaptive Modulation

Recall that in Chapter 3 several popular modulation and coding schemes were discussed. In that chapter, none of the modulation and coding techniques discussed adapted to fading conditions. These nonadaptive methods require a fixed link margin to maintain acceptable performance when the channel quality is poor. Thus, these systems are effectively designed for the worst case channel conditions, resulting in insufficient utilization of the full channel capacity. Adapting modulation to the signal fading allows the channel to be used more efficiently since power and rate can be allocated to take advantage of favorable channel conditions. For the purpose of our discussions, adaptive modulation shall function as the bit loading algorithm associated with multi-antenna water-filling techniques.

The basic idea behind adaptive transmission is to maintain a constant bit error rate by varying the transmitted power level, symbol transmission rate, constellation size, coding rate/scheme, or any combination of these parameters [29, 30]. Thus, without sacrificing bit error rate (BER), these schemes provide high average spectral efficiency by transmitting at high speeds under favorable channel conditions, and reducing throughput as the channel degrades. Adaptive coded modulation does not require interleaving since error bursts are eliminated by adjusting the power, size, and duration of the transmitted signal constellation relative to the channel fading. However, adaptive modulation does require accurate channel estimates at the receiver which are fed back to the transmitter with minimal latency.

There are several practical constraints that determine when adaptive modulation should be used. If the channel is changing faster than it can be estimated and fed back to the transmitter, adaptive techniques will perform poorly, and other means of mitigating the effects of fading should be used. It has been shown in [30] that, to achieve a low target BER, the BER remains at its target level as long as the total delay of the channel estimator and feedback path is less than Doppler period $T_{Doppler}$, i.e.,

$$T_{Doppler} = \frac{\lambda}{\upsilon}$$

where v is the vehicle speed and λ is the signal wavelength. However, a higher BER target loosens the delay constraint. Furthermore, hardware constraints and pulse shaping considerations may dictate how often the transmitter can change its rate and/or power.

Cellular systems exploit the power falloff with distance of signal propagation to reuse the same frequency channel at spatially separated locations. While frequency reuse provides more efficient use of the limited available spectrum within a given area, it also introduces co-channel interference, which ultimately determines the data rates and corresponding BERs available to each user. Thus, although adaptive modulation techniques increase the spectral efficiency (bits/s/Hz) of a single channel, these techniques may also increase co-channel interference levels in a cellular system, thereby requiring a higher reuse distance to mitigate this increased interference power. Adaptive modulation may therefore reduce the area spectral efficiency [30]. As stated before, the inherent problem in fading channels is that the transmitted signal undergo a variable gain in transmission; this results to a time variant SNR in receiver. As a result, the instantaneous capacity is a random variable. Assuming that the fading gain is an ergodic random process, we can expect that the temporal average capacity of the channel be equal to the expected value of the instantaneous capacity.

If the channel state information (CSI) is not available at the transmitter, the optimal power allocation is to distribute it uniformly over time or subchannels. As the result, the achievable channel capacity is described by [5]:

$$C_{\text{RCSI}} = E_{\gamma}\{\log_2(1+P_{av}\gamma)\} \tag{4.86}$$

$$= \int_0^{\infty} p_{\gamma}(\gamma)\log_2(1+P_{av}\gamma)d\gamma \tag{4.87}$$

where $\gamma = \frac{v}{\sigma_{\eta}^2}$ is the channel SNR function and E_{γ} is the statistical expectation with respect to γ. In contrast, if the CSI is available at the transmitter, it can adapt its output power P_x with respect to γ to increase the spectral efficiency. Therefore, under the power constraint $E\gamma\{P_x(\gamma)\} = P_{av}$, the maximum achievable capacity is achieved by temporal water-filling, which is described by

$$P_x^o(\gamma) = \left[\frac{1}{\gamma_0} - \frac{1}{\gamma}\right]^+ \tag{4.88}$$

where γ_0 is chosen so that $P_x^o(\gamma)$ satisfies the power constraint. This adaptation of the output power results in the following equation for channel capacity [5]:

$$C_{\text{TRCSI}} = E_{\gamma}\left(\left[\log_2\left(\frac{\gamma}{\gamma_0}\right)\right]^+\right) \tag{4.89}$$

Channel Inversion

An alternate suboptimal approach is to invert the channel at the transmitter, referred to *channel inversion.* Channel inversion is very common in spread—spectrum systems with near—far interference imbalances. It is also very simple to implement, since the encoder and decoder are designed for an AWGN channel, independent of the fading statistics. However, it can exhibit a large capacity penalty in extreme fading environments. For example, in Rayleigh fading, the capacity gap $E\left(\frac{1}{\gamma}\right)$ is infinite, and thus the capacity with channel inversion is zero. The power adaptation for channel inversion $P_x^I(\gamma)$ is given by [31]

$$P_x^I(\gamma) = \frac{P_{av}}{\gamma E\left(\frac{1}{\gamma}\right)} \tag{4.90}$$

and the resulting achievable capacity for channel inversion power allocation C_{TRCSI}^I is

$$C_{\text{TRCSI}}^I = \log_2\left(1 + \frac{P_{av}}{E\left(\frac{1}{\gamma}\right)}\right) \tag{4.91}$$

Consequently, a truncated channel inversion policy that only compensates for fading above a certain cutoff fade depth level γ_0 is defined as [31]

$$P_\gamma^{TI}(v) = \begin{cases} \frac{P_{av}}{\gamma \int_{\gamma_0}^{\infty} \frac{1}{\gamma} p_\gamma(\gamma) d\gamma}, & \gamma \geq \gamma_0 \\ 0, \gamma < \gamma_0 \end{cases} \tag{4.92}$$

For decoding this truncated policy, the receiver must know when $\gamma < \gamma_0$. The maximum capacity in this case is obtained by maximizing over all possible γ_0

$$C_{\text{TRCSI}}^{TI} = \max_{\gamma_0} \log_2\left(1 + \frac{P_{av}}{\int_{\gamma_0}^{\infty} \frac{1}{\gamma} p_\gamma(\gamma) d\gamma}\right) \tag{4.93}$$

A more detailed analysis and also numerical simulation for each of these methods has been done by Goldsmith in [31]. Assuming the flat fading channel described by Equation 4.14, the general system model for an adaptive modulator that transmits message m from transmitter to receiver is illustrated in Figure 4.12. The message is encoded into the codeword x, which is transmitted over the time variant channel as $x(i)$ at time index i. The channel gain $g(i)$ changes over the transmission of the codeword. The perfect instantaneous channel power gain $g(i)$ is assumed to be known in both transmitter and receiver. This allows the transmitter to adapt $x(i)$ to the channel gain at time index i, and is a reasonable model for a slowly varying channel with channel estimation for transmitter feedback.