

# Practical Models in C++



TN93  
R787

# SIMULATING WIRELESS COMMUNICATION SYSTEMS

Companion Software Website  
<http://authors.phptr.com/rorabaugh/>

C. Britton Rorabaugh



E200500037



**PRENTICE HALL**  
Professional Technical Reference  
Upper Saddle River, NJ 07458  
[www.phptr.com](http://www.phptr.com)

## Library of Congress Cataloging-in-Publication Data

A CIP catalog record for this book can be obtained from the Library of Congress.

Companion Software Website <http://authors.phptr.com/rorabaugh/>

Editorial/Production Supervision: *Patti Guerrieri*

Cover Design Director: *Jerry Votta*

Cover Design: *Anthony Gemmellaro*

Art Director: *Gail Cocker-Bogusz*

Manufacturing Buyer: *Maura Zaldivar*

Publisher: *Bernard Goodwin*

Editorial Assistant: *Michelle Vincenti*

Marketing Manager: *Dan DePasquale*



© 2004 Pearson Education, Inc.

Publishing as Prentice Hall Professional Technical Reference

Upper Saddle River, New Jersey 07458

**Prentice Hall PTR offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales, 1-800-382-3419, [corpsales@pearsontechgroup.com](mailto:corpsales@pearsontechgroup.com). For sales outside of the U.S., please contact: International Sales, 1-317-581-3793, [international@pearsontechgroup.com](mailto:international@pearsontechgroup.com).**

Company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

First Printing

ISBN 0-13-022268-2

Pearson Education Ltd.

Pearson Education Australia Pty., Limited

Pearson Education South Asia Pte. Ltd.

Pearson Education Asia Ltd.

Pearson Education Canada, Ltd.

Pearson Educación de México, S.A. de C.V.

Pearson Education — Japan

Pearson Malaysia SDN BHD

# SIMULATING WIRELESS COMMUNICATION SYSTEMS

## About Prentice Hall Professional Technical Reference

---

With origins reaching back to the industry's first computer science publishing program in the 1960s, and formally launched as its own imprint in 1986, Prentice Hall Professional Technical Reference (PH PTR) has developed into the leading provider of technical books in the world today. Our editors now publish over 200 books annually, authored by leaders in the fields of computing, engineering, and business.

Our roots are firmly planted in the soil that gave rise to the technical revolution. Our bookshelf contains many of the industry's computing and engineering classics: Kernighan and Ritchie's *C Programming Language*, Nemeth's *UNIX System Administration Handbook*, Horstmann's *Core Java*, and Johnson's *High-Speed Digital Design*.

PH PTR acknowledges its auspicious beginnings while it looks to the future for inspiration. We continue to evolve and break new ground in publishing by providing today's professionals with tomorrow's solutions.



*To Joyce, Geoff, Amber, and Eleanor*

---

---

# PREFACE

*M*odern communications systems and the devices operating within these systems would not be possible without simulation, but practical information specific to the simulation of communications systems is relatively scarce. My motive for writing this book was to collect and capture in a useful form the techniques that can be used to simulate a wireless communication system using C++. It has been my experience that organizations newly confronted with a need to simulate a communication system are in a rush to get started. Consequently, these organizations will purchase a commercial simulation package like SPW or MATLAB Simulink without even considering the alternative of constructing their own simulation using C++. In the beginning, progress comes quickly as simple systems are configured from standard library models. Only when they begin to model the more complex proprietary parts of their systems do these organizations begin to realize how much control and flexibility they sacrificed in going with a commercial package. It is not possible for any library of precoded models to be absolutely complete. There will always be a need to build a highly specialized model or make modifications to existing models. A user attempting to do either, using a commercial package, usually spends more time dealing with the rules and limitations of the simulation infrastructure than with the details of the model algorithms themselves.

In the mid 1990s, I was the architect and lead designer for a proprietary simulation package that was used to simulate the wireless data communication links in several very large U.S. defense systems. This package wasn't perfect—software never is—but I drew upon this experience, and while writing this book, I developed a simpler simulation package that avoids many of the complexities and objectionable features of my earlier effort. This new package is called `PracSim`, which is short for Practical Simulation. All of the source code for the models and infrastructure comprising the `PracSim` package is provided on the Prentice Hall Web site (<http://authors.phptr.com/rorabaugh/>). Examples of this code are pre-

sented and discussed throughout the book, but there is far too much code to include it all in the text. The library of `PracSim` models is not intended to be complete, but rather to provide a foundation that users can modify or build upon as needed to capture the nuances of the particular systems they are attempting to model.

I didn't keep accurate records, but I'm sure that construction of the `PracSim` software took far more time than the actual writing of the text. I would like to thank my wife Joyce, son Geoffrey, daughter Amber, and mother-in-law Eleanor for not complaining too much about all the time I spent on this project and for dealing with all of the household problems that I never seemed to have time for. I would also like to thank my editor, Bernard Goodwin, for his patience despite the numerous times that I postponed delivery of the final manuscript.



---

---

# CONTENTS

<b>PREFACE</b>	<b>xv</b>
<b>1 SIMULATION: BACKGROUND AND OVERVIEW</b>	<b>1</b>
1.1 Communication Systems	2
1.2 Simulation Process	2
1.3 Simulation Programs	3
<b>2 SIMULATION INFRASTRUCTURE</b>	<b>4</b>
2.1 Parameter Input	4
2.1.1 Individual Parameter Values	5
2.1.2 Parameter Arrays	5
2.1.3 Enumerated Type Parameters	7
2.1.4 System Parameters	7
2.1.5 Signal-Plotting Parameters	8
2.2 Signals	9
2.2.1 Signal Management Strategy	10
2.2.2 SMS Implementation	20
2.3 Controls	29
2.4 Results Reporting	30
<b>2A EXAMPLE SOURCE CODE</b>	<b>33</b>
2A.1 PracSimModel	33
2A.2 GenericSignal	38
<b>3 SIGNAL GENERATORS</b>	<b>44</b>
3.1 Elementary Signal Generators	44
3.1.1 Unit Step	44
3.1.2 Rectangular Pulse	45
	<b>vii</b>

3.1.3	Unit Impulse	46
3.1.4	Software Implementation	47
3.2	Tone Generators	49
3.2.1	Software Implementation	50
3.3	Sampling Baseband Signals	51
3.3.1	Spectral View of Sampling	53
3.4	Baseband Data Waveform Generators	54
3.4.1	NRZ Baseband Signaling	55
3.4.2	Biphase Baseband Signaling	57
3.4.3	Delay Modulation	58
3.4.4	Practical Issues	59
3.5	Modeling Bandpass Signals	61
<b>3A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>64</b>
3A.1	MultipleToneGener	64
3A.2	BasebandWaveform	69
<b>4</b>	<b>RANDOM PROCESS MODELS</b>	<b>78</b>
4.1	Random Sequences	78
4.1.1	Discrete Distributions	79
4.1.2	Discrete-Time Random Processes	82
4.2	Random Sequence Generators	83
4.2.1	Linear Congruential Sequences	84
4.2.2	Software Implementations	90
4.2.3	Evaluating Random-Number Generators	92
4.3	Continuous-Time Noise Processes	93
4.3.1	Continuous Random Variables	94
4.3.2	Random Processes	97
4.4	Additive Gaussian Noise Generators	99
4.4.1	Gaussian Distribution	99
4.4.2	Error Function	100
4.4.3	Spectral Properties	101
4.4.4	Noise Power	102
4.4.5	Gaussian Random Number Generators	102
4.5	Bandpass Noise	104
4.5.1	Envelope and Phase	104
4.5.2	Rayleigh Random Number Generators	109
4.6	Parametric Models of Random Processes	109
4.6.1	Autoregressive Noise Model	110

<b>4A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>112</b>
4A.1	AdditiveGaussianNoise	112
<b>5</b>	<b>DISCRETE TRANSFORMS</b>	<b>119</b>
5.1	Discrete Fourier Transform	119
5.1.1	Parameter Selection	120
5.1.2	Properties of the DFT	120
5.2	Decimation-in-Time Algorithms	123
5.2.1	Software Notes	126
5.3	Decimation-in-Frequency Algorithms	131
5.4	Small - $N$ Transforms	136
5.5	Prime Factor Algorithm	138
5.5.1	Software Notes	138
<b>5A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>141</b>
5A.1	FFT Wrapper Routines	141
5A.2	FFT Engines	141
<b>6</b>	<b>SPECTRUM ESTIMATION</b>	<b>146</b>
6.1	Sample Spectrum	146
6.1.1	Software Implementation	147
6.2	Daniell Periodogram	148
6.2.1	Software Implementation	149
6.3	Bartlett Periodogram	151
6.3.1	Software Implementation	152
6.4	Windowing and Other Issues	153
6.4.1	Triangular Window	154
6.4.2	Software Considerations	155
6.4.3	von Hann Window	157
6.4.4	Hamming Window	160
6.4.5	Software Implementation	161
6.5	Welch Periodogram	167
6.5.1	Software Implementation	167
6.6	Yule-Walker Method	167
6.6.1	Software Implementation	168
<b>6A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>171</b>
6A.1	BartlettPeriodogramWindowed	171
6A.2	GenericWindow	177

<b>7</b>	<b>SYSTEM CHARACTERIZATION TOOLS</b>	<b>182</b>
7.1	Linear Systems	182
7.1.1	Characterization of Linear Systems	183
7.1.2	Transfer Functions	184
7.1.3	Computer Representation of Transfer Functions	186
7.1.4	Magnitude, Phase, and Delay Responses	189
7.2	Constellation Plots	192
7.2.1	Eye Diagrams	193
<b>7A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>199</b>
7A.1	CmpxIqPlot	200
7A.2	HistogramBuilder	203
<b>8</b>	<b>FILTER MODELS</b>	<b>207</b>
8.1	Modeling Approaches	207
8.1.1	Numerical Integration	207
8.1.2	Sampled Frequency Response	208
8.1.3	Digital Filters	208
8.2	Analog Filter Responses	209
8.2.1	Magnitude Response Features of Lowpass Filters	210
8.2.2	Filter Transformations	210
8.3	Classical Analog Filters	217
8.3.1	Butterworth Filters	217
8.3.2	Chebyshev Filters	218
8.3.3	Elliptical Filters	222
8.3.4	Bessel Filters	227
8.4	Simulating Filters via Numerical Integration	229
8.4.1	Biquadratic Form	231
8.4.2	Software Design	232
8.5	Using IIR Digital Filters to Simulate Analog Filters	234
8.5.1	Properties of IIR Filters	236
8.5.2	Mapping Analog Filters into IIR Designs	237
8.5.3	Software Design	240
8.6	Filtering in the Frequency Domain	242
8.6.1	Fast Convolution	242
8.6.2	Software Design	244
<b>8A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>247</b>
8A.1	Classical Filters	247

<b>9</b>	<b>MODULATION AND DEMODULATION</b>	<b>262</b>
9.1	Simulation Issues	262
9.1.1	Using the Recovered Carrier	263
9.2	Quadrature Phase Shift Keying	264
9.2.1	Nonideal Behaviors	266
9.2.2	Quadrature Modulator Models	269
9.2.3	Correlation Demodulator Models for QPSK	270
9.2.4	Quadrature Demodulator Models	273
9.2.5	QPSK Simulations	275
9.2.6	Properties of QPSK Signals	279
9.2.7	Offset QPSK	282
9.3	Binary Phase Shift Keying	286
9.3.1	BPSK Modulator Models	286
9.3.2	BPSK Demodulation	287
9.3.3	BPSK Simulations	289
9.3.4	Properties of BPSK Signals	290
9.3.5	Error Performance	292
9.4	Multiple Phase Shift Keying	293
9.4.1	Ideal $m$ -PSK Modulation and Demodulation	293
9.4.2	Power Spectral Densities of $m$ -PSK Signals	295
9.4.3	Error Performance	298
9.5	Frequency Shift Keying	299
9.5.1	FSK Modulators	303
9.6	Minimum Shift Keying	306
9.6.1	Nonideal Behaviors	306
9.6.2	MSK Modulator Models	309
9.6.3	Properties of MSK Signals	312
<b>9A</b>	<b>EXAMPLE SOURCE CODE</b>	<b>315</b>
9A.1	MskModulator	316
9A.2	MpskOptimalDemod	320
<b>10</b>	<b>AMPLIFIERS AND MIXERS</b>	<b>325</b>
10.1	Memoryless Nonlinearities	326
10.1.1	Hard Limiters	326
10.1.2	Bandpass Amplifiers	327
10.2	Characterizing Nonlinear Amplifiers	342
10.2.1	AM/AM and AM/PM	342
10.2.2	Swept-Frequency Response	343

10.3 Two-Box Nonlinear Amplifier Models	344
10.3.1 Filter Measurements	344
<b>10A EXAMPLE SOURCE CODE</b>	<b>350</b>
10A.NonlinearAmplifier	350
<b>11 SYNCHRONIZATION AND SIGNAL SHIFTING</b>	<b>356</b>
11.1 Shifting Signals in Time	356
11.1.1 Delaying Signals by Multiples of the Sampling Interval	357
11.1.2 Advancing Signals by Multiples of the Sampling Interval	360
11.1.3 Continuous-Time Delays via Interpolation	368
11.2 Correlation-Based Delay Estimation	385
11.2.1 Software Implementation	387
11.3 Phase-Slope Delay Estimation	388
11.4 Changing Clock Rates	393
<b>11A EXAMPLE SOURCE CODE</b>	<b>398</b>
11A.DiscreteDelay	398
<b>12 SYNCHRONIZATION RECOVERY</b>	<b>406</b>
12.1 Linear Phase-Locked Loops	407
12.2 Digital Phase-Locked Loops	412
12.2.1 Phase-Frequency Detector	412
12.3 Phase-Locked Demodulators	424
12.3.1 Squaring Loop	424
12.3.2 Costas Loop	426
<b>12A EXAMPLE SOURCE CODE</b>	<b>430</b>
12A.DigitalPLL	430
<b>13 CHANNEL MODELS</b>	<b>440</b>
13.1 Discrete Memoryless Channels	440
13.1.1 Binary Symmetric Channel	440
13.1.2 Other Binary Channels	441
13.1.3 Nonbinary Channels	443
13.2 Characterization of Time-Varying Random Channels	449
13.2.1 System Functions	449
13.2.2 Randomly Time-Varying Channels	455
13.3 Diffuse Multipath Channels	459
13.3.1 Uncorrelated Tap Gains	460

---

13.3.2	Correlated Tap Gains	461
13.4	Discrete Multipath Channels	463
<b>14</b>	<b>MULTIRATE SIMULATIONS</b>	<b>465</b>
14.1	Basic Concepts of Multirate Signal Processing	465
14.1.1	Decimation by Integer Factors	466
14.1.2	Interpolation by Integer Factors	466
14.1.3	Decimation and Interpolation by Noninteger Factors	468
14.2	Filter Design for Interpolators and Decimators	469
14.2.1	Interpolation	471
14.2.2	Decimation	480
14.3	Multirate Processing for Bandpass Signals	487
14.3.1	Quadrature Demodulation	487
14.3.2	Quadrature Modulation	487
<b>15</b>	<b>MODELING DSP COMPONENTS</b>	<b>491</b>
15.1	Quantization and Finite-Precision Arithmetic	491
15.1.1	Coefficient Quantization	491
15.1.2	Signal Quantization	495
15.1.3	Finite-Precision Arithmetic	495
15.2	FIR Filters	496
15.3	IIR Filters	501
<b>16</b>	<b>CODING AND INTERLEAVING</b>	<b>506</b>
16.1	Block Codes	506
16.1.1	Cyclic Codes	507
16.2	BCH Codes	509
16.3	Interleavers	513
16.3.1	Block Interleavers	513
16.3.2	Convolutional Interleavers	514
16.4	Convolutional Codes	515
16.4.1	Trellis Representation of a Convolutional Encoder	518
16.4.2	Viterbi Decoding	519
16.5	Viterbi Decoding with Soft Decisions	525
<b>A</b>	<b>MATHEMATICAL TOOLS</b>	<b>532</b>
A.1	Trigonometric Identities	532
A.2	Table of Integrals	534
A.3	Logarithms	536

A.4	Modified Bessel Functions of the First Kind	536
A.4.1	Identities	537
<b>B</b>	<b>PROBABILITY DISTRIBUTIONS IN COMMUNICATIONS</b>	<b>538</b>
B.1	Uniform Distribution	538
B.2	Gaussian Distribution	538
B.3	Exponential Distribution	539
B.4	Rayleigh Distribution	540
B.4.1	Relationship to Exponential Distribution	541
B.5	Rice Distribution	541
B.5.1	Marcum $Q$ Function	542
<b>C</b>	<b>GALOIS FIELDS</b>	<b>543</b>
C.1	Finite Fields	543
C.1.1	Fields	545
C.2	Polynomial Arithmetic	545
C.3	Computer Generation of Extension Fields	551
C.3.1	Computer Representations for Polynomials	552
C.3.2	Using a Computer to Find Primitive Polynomials	552
C.3.3	Programming Considerations	557
C.4	Minimal Polynomials and Cyclotomic Cosets	559
<b>D</b>	<b>REFERENCES</b>	<b>563</b>
	<b>INDEX</b>	<b>566</b>



# SIMULATION: BACKGROUND AND OVERVIEW

*M*odern communications systems and the devices operating within these systems would not be possible without simulation. The expanded use of digital signal processing techniques has spawned cell phones and wireless transceivers that offer incredible performance and features at a per-unit cost that puts them within the reach of nearly everyone. However, these low per-unit costs are achieved through mass production of hundreds of thousands or even millions of units from a single design. The design of a new cell phone or wireless modem for a PDA is a very complex and expensive affair. Because of the complexity in such devices, it is not practical to breadboard prototypes for testing until after the design has been exhaustively tested and honed using simulation. Even after a new device has been prototyped, it is usually impractical to test it under every possible combination of operating conditions. For example, the nature of CDMA and GSM cellular phone systems is such that all of the phones in a given area unavoidably interfere with each other. The phones and base stations all include processing to mitigate this interference, but the severity of the interference and the effectiveness of the countermeasures depend upon the relative locations, with respect to the base station tower, of all the potentially interfering phones. Assessment of the interference is complicated by the fact that the phones can individually vary their transmit powers via power-control loops executing in the phones or in response to commands from the base station. Analysis is impossible and exhaustive testing is impractical. Simulation using carefully constructed models of the phones and base station is the only answer. In the design of nearly any type of communications equipment, simulation provides an inexpensive way to explore possibilities and design trades before the more expensive process of prototyping is initiated.