



mobile communications series

SOFTWARE-DEFINED RADIO for ENGINEERS

TRAVIS F. COLLINS
ROBIN GETZ
DI PU
ALEXANDER M. WYGLINSKI

Software-Defined Radio for Engineers

For a listing of recent titles in the *Artech House
Mobile Communications*, turn to the back of this book.

Software-Defined Radio for Engineers

Travis F. Collins

Robin Getz

Di Pu

Alexander M. Wyglinski

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

A catalog record for this book is available from the British Library.

ISBN-13: 978-1-63081-457-1

Cover design by John Gomes

© 2018 Travis F. Collins, Robin Getz, Di Pu, Alexander M. Wyglinski

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House 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.

10 9 8 7 6 5 4 3 2 1

Dedication

To my wife Lauren
—Travis Collins

To my wonderful children, Matthew, Lauren, and Isaac, and my patient wife, Michelle—sorry I have been hiding in the basement working on this book. To all my fantastic colleagues at Analog Devices: Dave, Michael, Lars-Peter, Andrei, Mihai, Travis, Wyatt and many more, without whom Pluto SDR and IIO would not exist.
—Robin Getz

To my lovely son Aidi, my husband Di, and my parents Lingzhen and Xuexun
—Di Pu

To my wife Jen
—Alexander Wyglinski

Contents

Preface	xiii
CHAPTER 1	
Introduction to Software-Defined Radio	1
1.1 Brief History	1
1.2 What is a Software-Defined Radio?	1
1.3 Networking and SDR	7
1.4 RF architectures for SDR	10
1.5 Processing architectures for SDR	13
1.6 Software Environments for SDR	15
1.7 Additional readings	17
References	18
CHAPTER 2	
Signals and Systems	19
2.1 Time and Frequency Domains	19
2.1.1 Fourier Transform	20
2.1.2 Periodic Nature of the DFT	21
2.1.3 Fast Fourier Transform	22
2.2 Sampling Theory	23
2.2.1 Uniform Sampling	23
2.2.2 Frequency Domain Representation of Uniform Sampling	25
2.2.3 Nyquist Sampling Theorem	26
2.2.4 Nyquist Zones	29
2.2.5 Sample Rate Conversion	29
2.3 Signal Representation	37
2.3.1 Frequency Conversion	38
2.3.2 Imaginary Signals	40
2.4 Signal Metrics and Visualization	41
2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR	42
2.4.2 Eye Diagram	44
2.5 Receive Techniques for SDR	45
2.5.1 Nyquist Zones	47
2.5.2 Fixed Point Quantization	49

2.5.3	Design Trade-offs for Number of Bits, Cost, Power, and So Forth	55
2.5.4	Sigma-Delta Analog-Digital Converters	58
2.6	Digital Signal Processing Techniques for SDR	61
2.6.1	Discrete Convolution	61
2.6.2	Correlation	65
2.6.3	Z-Transform	66
2.6.4	Digital Filtering	69
2.7	Transmit Techniques for SDR	73
2.7.1	Analog Reconstruction Filters	75
2.7.2	DACs	76
2.7.3	Digital Pulse-Shaping Filters	78
2.7.4	Nyquist Pulse-Shaping Theory	79
2.7.5	Two Nyquist Pulses	81
2.8	Chapter Summary	85
	References	85
CHAPTER 3		
	Probability in Communications	87
3.1	Modeling Discrete Random Events in Communication Systems	87
3.1.1	Expectation	89
3.2	Binary Communication Channels and Conditional Probability	92
3.3	Modeling Continuous Random Events in Communication Systems	95
3.3.1	Cumulative Distribution Functions	99
3.4	Time-Varying Randomness in Communication Systems	101
3.4.1	Stationarity	104
3.5	Gaussian Noise Channels	106
3.5.1	Gaussian Processes	108
3.6	Power Spectral Densities and LTI Systems	109
3.7	Narrowband Noise	110
3.8	Application of Random Variables: Indoor Channel Model	113
3.9	Chapter Summary	114
3.10	Additional Readings	114
	References	115
CHAPTER 4		
	Digital Communications Fundamentals	117
4.1	What Is Digital Transmission?	117
4.1.1	Source Encoding	120
4.1.2	Channel Encoding	122
4.2	Digital Modulation	127
4.2.1	Power Efficiency	128
4.2.2	Pulse Amplitude Modulation	129

4.2.3	Quadrature Amplitude Modulation	131
4.2.4	Phase Shift Keying	133
4.2.5	Power Efficiency Summary	139
4.3	Probability of Bit Error	141
4.3.1	Error Bounding	145
4.4	Signal Space Concept	148
4.5	Gram-Schmidt Orthogonalization	150
4.6	Optimal Detection	154
4.6.1	Signal Vector Framework	155
4.6.2	Decision Rules	158
4.6.3	Maximum Likelihood Detection in an AWGN Channel	159
4.7	Basic Receiver Realizations	160
4.7.1	Matched Filter Realization	161
4.7.2	Correlator Realization	164
4.8	Chapter Summary	166
4.9	Additional Readings	168
	References	169

CHAPTER 5

	Understanding SDR Hardware	171
5.1	Components of a Communication System	171
5.1.1	Components of an SDR	172
5.1.2	AD9363 Details	173
5.1.3	Zynq Details	176
5.1.4	Linux Industrial Input/Output Details	177
5.1.5	MATLAB as an IIO client	178
5.1.6	Not Just for Learning	180
5.2	Strategies For Development in MATLAB	181
5.2.1	Radio I/O Basics	181
5.2.2	Continuous Transmit	183
5.2.3	Latency and Data Delays	184
5.2.4	Receive Spectrum	185
5.2.5	Automatic Gain Control	186
5.2.6	Common Issues	187
5.3	Example: Loopback with Real Data	187
5.4	Noise Figure	189
	References	190

CHAPTER 6

	Timing Synchronization	191
6.1	Matched Filtering	191
6.2	Timing Error	195
6.3	Symbol Timing Compensation	198

6.3.1	Phase-Locked Loops	200
6.3.2	Feedback Timing Correction	201
6.4	Alternative Error Detectors and System Requirements	208
6.4.1	Gardner	208
6.4.2	Müller and Mueller	208
6.5	Putting the Pieces Together	209
6.6	Chapter Summary	212
	References	212
CHAPTER 7		
	Carrier Synchronization	213
7.1	Carrier Offsets	213
7.2	Frequency Offset Compensation	216
7.2.1	Coarse Frequency Correction	217
7.2.2	Fine Frequency Correction	219
7.2.3	Performance Analysis	224
7.2.4	Error Vector Magnitude Measurements	226
7.3	Phase Ambiguity	228
7.3.1	Code Words	228
7.3.2	Differential Encoding	229
7.3.3	Equalizers	229
7.4	Chapter Summary	229
	References	230
CHAPTER 8		
	Frame Synchronization and Channel Coding	231
8.1	O Frame, Where Art Thou?	231
8.2	Frame Synchronization	232
8.2.1	Signal Detection	235
8.2.2	Alternative Sequences	239
8.3	Putting the Pieces Together	241
8.3.1	Full Recovery with Pluto SDR	242
8.4	Channel Coding	244
8.4.1	Repetition Coding	244
8.4.2	Interleaving	245
8.4.3	Encoding	246
8.4.4	BER Calculator	251
8.5	Chapter Summary	251
	References	251
CHAPTER 9		
	Channel Estimation and Equalization	253
9.1	You Shall Not Multipath!	253

9.2	Channel Estimation	254
9.3	Equalizers	258
9.3.1	Nonlinear Equalizers	261
9.4	Receiver Realization	263
9.5	Chapter Summary	265
	References	266
CHAPTER 10		
	Orthogonal Frequency Division Multiplexing	267
10.1	Rationale for MCM: Dispersive Channel Environments	267
10.2	General OFDM Model	269
10.2.1	Cyclic Extensions	269
10.3	Common OFDM Waveform Structure	271
10.4	Packet Detection	273
10.5	CFO Estimation	275
10.6	Symbol Timing Estimation	279
10.7	Equalization	280
10.8	Bit and Power Allocation	284
10.9	Putting It All Together	285
10.10	Chapter Summary	286
	References	286
CHAPTER 11		
	Applications for Software-Defined Radio	289
11.1	Cognitive Radio	289
11.1.1	Bumblebee Behavioral Model	292
11.1.2	Reinforcement Learning	294
11.2	Vehicular Networking	295
11.3	Chapter Summary	299
	References	299
APPENDIX A		
	A Longer History of Communications	303
A.1	History Overview	303
A.2	1750–1850: Industrial Revolution	304
A.3	1850–1945: Technological Revolution	305
A.4	1946–1960: Jet Age and Space Age	309
A.5	1970–1979: Information Age	312
A.6	1980–1989: Digital Revolution	313
A.7	1990–1999: Age of the Public Internet (Web 1.0)	316
A.8	Post-2000: Everything comes together	319
	References	319

APPENDIX B

Getting Started with MATLAB and Simulink	327
B.1 MATLAB Introduction	327
B.2 Useful MATLAB Tools	327
B.2.1 Code Analysis and M-Lint Messages	328
B.2.2 Debugger	329
B.2.3 Profiler	329
B.3 System Objects	330
References	332

APPENDIX C

Equalizer Derivations	333
C.1 Linear Equalizers	333
C.2 Zero-Forcing Equalizers	335
C.3 Decision Feedback Equalizers	336

APPENDIX D

Trigonometric Identities	337
About the Authors	339
Index	341

Equalizer Derivations

C.1 Linear Equalizers

Suppose we assume a transceiver model where the information source produces amplitude values I_n applied to an infinite impulse train; namely,

$$s(t) = \sum_{n=-\infty}^{\infty} I_n \delta(t - nT), \quad (\text{C.1})$$

where $\delta(t)$ is the Dirac delta function. Applying a transmit pulse shaping filter $b_T(t)$ to the information signal $s(t)$, we obtain the transmitter's output signal:

$$v(t) = \sum_{n=-\infty}^{\infty} I_n b_T(t - nT), \quad (\text{C.2})$$

which is then sent through a propagation channel that is characterized by a channel filter $b_C(t)$ and an additive white Gaussian noise signal $z(t)$. The output of the channel filter yields a signal:

$$p(t) = \sum_{n=-\infty}^{\infty} I_n b(t - nT), \quad (\text{C.3})$$

where $b(t) = b_T(t) * b_C(t)$ is the channel impulse response. The signal intercepted by the receiver is expressed as

$$r(t) = p(t) + z(t) = \sum_{n=-\infty}^{\infty} I_n b(t - nT) + z(t). \quad (\text{C.4})$$

At the receiver, we would like to find the expression for the mean squared error (MSE) and minimize it. To achieve this objective, we choose the receive filter $b_R(t)$ to be matched to $b(t)$, yielding the following optimal result:

$$b_R(t) = b^*(-t), \quad (\text{C.5})$$

which results in the output of the receiver filter being equal to

$$y(t) = p(t) * b^*(-t) + z(t) * b^*(-t) = \sum_{n=-\infty}^{\infty} I_n g(t - nT) + z(t) * b^*(-t), \quad (\text{C.6})$$

where $g(t) = h(t) * h_R(t)$, and

$$x(t) = \phi_b(t) = h(t) * h^*(-t) = \int_{-\infty}^{\infty} h(t + \tau)h^*(\tau)d\tau. \quad (\text{C.7})$$

Supposed that $v(t) = z(t) * h^*(-t)$, then we get the following expression:

$$y(t) = \sum_{n=-\infty}^{\infty} I_n x(t - nT) + v(t). \quad (\text{C.8})$$

Now, let us sample the output of the receive filter such that

$$y_k = y(kT) = \sum_{n=-\infty}^{\infty} I_n x((k - n)T) + v(kT), \quad (\text{C.9})$$

$$= \sum_{n=-\infty}^{\infty} I_n x_{k-n} + v_k. \quad (\text{C.10})$$

Note that x_k is referred to as the channel autocorrelation. Furthermore, the v_k samples are not white due to the filtering by $h^*(t)$, which means that $E\{v_k v_{k+n}\} \neq \delta(n)$. In these circumstances, we have a couple of options to deal with this colored noise. One approach involves finding $E\{v_k v_l^*\}$ for the noise sequence v_k . In this approach, we know that $v_k(t)$ is Gaussian since $z(t)$ is Gaussian. Furthermore, we know that $z(t)$ is white; that is, $S_z(f) = N_0$. Thus, if $E\{z^{ast}(s)z(t)\} = N_0\delta(t - s)$, we can then solve $E\{v_k v_l^*\}$ as follows:

$$E\{v_k v_l^*\} = \int_{-\infty}^{\infty} ds \int_{-\infty}^{\infty} dt h(t - lT)h^*(s - kT)E\{z^*(s)z(t)\}, \quad (\text{C.11})$$

$$= N_0 \int_{-\infty}^{\infty} dt h(t - lT)h^*(t - kT), \quad (\text{C.12})$$

$$= N_0 \int_{-\infty}^{\infty} dt h(t + |k - l|T)h^*(t). \quad (\text{C.13})$$

Since we have:

$$x_k = \int_{-\infty}^{\infty} h^*(t)h(t + kT)dt, \quad (\text{C.14})$$

which then gives us the expression for $E\{v_k v_l^*\}$ as:

$$E\{v_k v_l^*\} = N_0 x_{k-l}, \quad (\text{C.15})$$

that can then be written as the autocorrelation function and the power spectral density of v_k ; namely,

$$R_v(k) = E\{v_n v_{k+n}^*\} = N_0 x_k, \quad (\text{C.16})$$

$$S_v(z) = \mathcal{Z}\{R_v(k)\} = N_0 X(z). \quad (\text{C.17})$$

The second approach for dealing with colored noise is to implement something referred to as a *whitening filter*. In this case, we try to reverse the effects of the receiver filter on the noise signal $z(t)$. In other words, since we have the power spectral density of v_k equal to $S_v(z) = N_0 X(z)$, we ultimately would like to have the whitened noise power spectral density only equal to N_0 . To achieve this, we assume that the z-transform of x_k , $X(z)$, can be represented by the following:

$$X(z) = F(z)F^*(1/z^*). \quad (\text{C.18})$$

Thus, in we have a whitening filter whose transfer function is equal to $1/F^*(1/z^*)$, the resulting power spectral density at the output of this whitening filter should be equal to:

$$S_n(z) = \frac{1}{F(z)F^*(1/z^*)} S_v(z) = \frac{N_0 X(z)}{X(z)} = N_0, \quad (\text{C.19})$$

which yields an output noise signal that is white.

C.2 Zero-Forcing Equalizers

Suppose we assume a discrete time model for the receiver that is equal to the following:

$$w_k = \sum_{n=-\infty}^{\infty} I_n f_{k-n} + n_k, \quad (\text{C.20})$$

where w_k is the output signal of the whitening filter, f_k is the impulse response of the whitening filter, and n_k is the whitened noise signal with power spectral density equal to N_0 .

In the *zero-forcing equalizer* (ZFE), we choose a transfer function $C(z)$ such that each ISI-compensated term $q_k = \delta_k$, which implies that there is no ISI present and the resulting equalized outputs $I_k + n'_k$ are subsequently quantized, yielding a probability of error equal to $P_e = Q(1/\sigma')$ for $I_k = \pm 1$. Although P_e is often used as a performance metric for a communication system, in the case of equalizer design we are going to use the minimum mean squared error (MMSE) as our metric; that is, $E\{|I_k - \hat{I}_k|^2\}$.

Suppose we take the z-transform of (C.20), thus obtaining:

$$W(z) = I(z)F(z) + N(z) \quad (\text{C.21})$$

and once this has been filtered by the ZFE equalizer we get the output equal to

$$W(z)C(z) = I(z)F(z)C(z) + N(z)C(z), \quad (\text{C.22})$$

with $Q(z) = F(z)C(z)$. The objective of our ZFE is to generate an output equal to

$$\mathcal{Z}^{-1}\{W(z)C(z)\} = I_k * q_k + n'_k, \quad (\text{C.23})$$

where $q_k = \delta_k$; that is, no ISI present in the signal. Consequently, the ISI term in this expression is $F(z)C(z)$ and thus we want it to be equal to $Q(z) = F(z)C(z) = 1$ for $q_k = \delta_k$. Rearranging the terms, our ZFE should be equal to $C(z) = 1/F(z)$.

C.3 Decision Feedback Equalizers

Although ZFE filters are conceptually straightforward, they almost always possess an infinite number of taps, thus making them impossible to implement in real world applications. Furthermore, when the signal resulting at the output of the whitening filter is equalized using the ZFE, any noise contributions contained within the equalized signal will no longer be white. As a result, there exists other equalizer designs that can be more readily implemented in hardware. One of these is the *decision feedback equalizer* (DFE), which consists of two filters: a feedforward filter $a(z)$ and a feedback filter $b(z) - 1$.

The *feedforward filter* $a(z)$ is anti-causal and has the following form:

$$a(z) = a_{-1}z + a_{-2}z^2 + a_{-3}z^3 + a_{-4}z^4 + \dots, \quad (\text{C.24})$$

while the *feedback filter* $b(z)$ is a causal filter possessing the form:

$$b(z) - 1 = b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + \dots \quad (\text{C.25})$$

Consequently, we can design $a(z)$ and $b(z)$ such that

$$a(z) = CF^*(1/z^*) \quad (\text{C.26})$$

$$b(z) = F(z) \quad (\text{C.27})$$

$$F(z)F^*(1/z^*) = X(z) + N_0. \quad (\text{C.28})$$