Software Receiver Design

Build Your Own Digital Communications System in Five Easy Steps

Have you ever wanted to know how modern digital communications systems work? Find out with this step-by-step guide to building a complete digital radio that includes every element of a typical, real-world communication system. Chapter by chapter, you will create a MATLAB realization of the various pieces of the system, exploring the key ideas along the way, as well as analyzing and assessing the performance of each component. Then, in the final chapters, you will discover how all the parts fit together and interact as you build the complete receiver. Can you decode the messages hidden within the received signals?

In addition to coverage of crucial issues, such as timing, carrier recovery, and equalization, the text contains over 400 practical exercises, providing invaluable preparation for industry, where wireless communications and software radio are becoming increasingly important. Various extra resources are also provided online, including lecture slides and a solutions manual for instructors.

C. Richard Johnson, Jr. is the Geoffrey S. M. Hedrick Senior Professor of Engineering at Cornell University, where he has been on the faculty since 1981. He is a Fellow of the IEEE and co-author of *Telecommunication Breakdown* (2004, with William A. Sethares) and *Theory and Design of Adaptive Filters* (2001).

William A. Sethares is a Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin in Madison. He is the author of *Rhythm and Transforms* (2007) and *Tuning, Timbre, Spectrum, Scale* (2005).

Andrew G. Klein is an Assistant Professor at Worcester Polytechnic Institute. In addition to working in academia, he has also held industry positions at several wireless start-up companies.

Software Receiver Design

Build Your Own Digital Communications System in Five Easy Steps

C. RICHARD JOHNSON, JR.
Cornell University

WILLIAM A. SETHARES
University of Wisconsin in Madison

ANDREW G. KLEIN
Worcester Polytechnic Institute

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107007529

First published 2011

A catalogue record for this publication is available from the British Library

ISBN 978-1-107-00752-9 Hardback
ISBN 978-0-521-18944-6 Paperback

Additional resources for this publication at www.cambridge.org/9781107007529

To the Instructor . . .

. . . though it's OK for the student to listen in.

Software Receiver Design helps the reader build a complete digital radio that includes each part of a typical digital communication system. Chapter by chapter, the reader creates a MATLAB® realization of the various pieces of the system, exploring the key ideas along the way. In the final chapters, the reader "puts it all together" to build fully functional receivers, though as MATLAB code they are not intended to operate in real time. **Software Receiver Design** explores telecommunication systems from a very particular point of view: the construction of a workable receiver. This viewpoint provides a sense of continuity to the study of communication systems.

The three basic tasks in the creation of a working digital radio are

1. building the pieces,
2. assessing the performance of the pieces,
3. integrating the pieces.

In order to accomplish this in a single semester, we have had to strip away some topics that are commonly covered in an introductory course and emphasize some topics that are often covered only superficially. We have chosen not to present an encyclopedic catalog of every method that can be used to implement each function of the receiver. For example, we focus on frequency division multiplexing rather than time or code division methods, and we concentrate only on pulse amplitude modulation and quadrature amplitude modulation. On the other hand, some topics (such as synchronization) loom large in digital receivers, and we have devoted a correspondingly greater amount of space to these. Our belief is that it is better to learn one complete system from start to finish than to half-learn the properties of many.

Whole Lotta Radio

Our approach to building the components of the digital radio is consistent throughout **Software Receiver Design**. For many of the tasks, we define a "performance" function and an algorithm that optimizes this function. This approach provides a unified framework for deriving the AGC, clock recovery, carrier recovery, and equalization algorithms. Fortunately, this can be accomplished using only the mathematical tools that an electrical engineer (at the level of a college junior) is likely to have, and **Software Receiver Design** requires no more than knowledge of calculus, matrix algebra, and Fourier transforms. Any of the fine texts cited for further reading in Section 3.8 would be fine.

Software Receiver Design emphasizes two ways to assess the behavior of the components of a communication system: by studying the performance functions and by conducting experiments. The algorithms embodied in the various components can be derived without making assumptions about details of the constituent signals (such as Gaussian noise). The use of probability is limited to naive ideas such as the notion of an average of a collection of numbers, rather than requiring the machinery of stochastic processes. The absence of an advanced probability prerequisite for **Software Receiver Design** makes it possible to place it earlier in the curriculum.

The integration phase of the receiver design is accomplished in Chapters 9 and 15. Since any real digital radio operates in a highly complex environment, analytical models cannot hope to approach the "real" situation. Common practice is to build a simulation and to run a series of experiments. **Software Receiver Design** provides a set of guidelines (in Chapter 15) for a series of tests to verify the operation of the receiver. The final project challenges the digital radio that the student has built by adding many different kinds of imperfections, including additive noise, multipath disturbances, phase jitter, frequency inaccuracies, and clock errors. A successful design can operate even in the presence of such distortions.

It should be clear that these choices distinguish **Software Receiver Design** from other, more encyclopedic texts. We believe that this "hands-on" method makes **Software Receiver Design** ideal for use as a learning tool, though it is less comprehensive than a reference book. In addition, the instructor may find that the order of presentation of topics in the five easy steps is different from that used by other books. Section 1.3 provides an overview of the flow of topics, and our reasons for structuring the course as we have.

Finally, we believe that **Software Receiver Design** may be of use to nontraditional students. Besides the many standard kinds of exercises, there are many problems in the text that are "self-checking" in the sense that the reader will know when/whether they have found the correct answer. These may also be useful to the self-motivated design engineer who is using **Software Receiver Design** to learn about digital radio.

How We've Used Software Receiver Design

The authors have taught from (various versions of) this text for a number of years, exploring different ways to fit coverage of digital radio into a "standard" electrical engineering elective sequence.

Perhaps the simplest way is via a "stand-alone" course, one semester long, in which the student works through the chapters and ends with the final project as outlined in Chapter 15. Students who have graduated tell us that when they get to the workplace, where software-defined digital radio is increasingly important, the preparation of this course has been invaluable. After having completed this course plus a rigorous course in probability, other students have reported that they are well prepared for the typical introductory graduate-level class in communications offered at research universities.

At Cornell University, the University of Wisconsin, and Worcester Polytechnic Institute (the home institutions of the authors), there is a two-semester sequence in communications available for advanced undergraduates. We have integrated the text into this curriculum in three ways.

1. Teach from a traditional text for the first semester and use **Software Receiver Design** in the second.
2. Teach from **Software Receiver Design** in the first semester and use a traditional text in the second.
3. Teach from **Software Receiver Design** in the first semester and teach a project-oriented extension in the second.

All three work well. When following the first approach, students often comment that by reading **Software Receiver Design** they "finally understand what they had been doing the previous semester." Because there is no probability prerequisite for **Software Receiver Design**, the second approach can be moved earlier in the curriculum. Of course, we encourage students to take probability at the same time. In the third approach, the students were asked to extend the basic pulse amplitude modulation (PAM) and quadrature amplitude modulation (QAM) digital radios to incorporate code division multiplexing, to use more advanced equalization techniques, etc.

We believe that the increasing market penetration of broadband communications is the driving force behind the continuing (re)design of "radios" (wireless communications devices). Digital devices continue to penetrate the market formerly occupied by analog (for instance, digital television has now supplanted analog television in the USA) and the area of digital and software-defined radio is regularly reported in the mass media. Accordingly, it is easy for the instructor to emphasize the social and economic aspects of the "wireless revolution." The impact of digital radio is vast, and it is an exciting time to get involved.

Some Extras

The course website contains extra material of interest, especially to the instructor. First, we have assembled a complete collection of slides (in .pdf format) that may help in lesson planning. The final project is available in two complete forms, one that exploits the block coding of Chapter 14 and one that does not. In addition, there are several "received signals" on the website, which can be used for assignments and for the project. Finally, all the Matlab code that is presented in the text is available on the website. Once these are added to the Matlab path, they can be used for assignments and for further exploration.[1]

Mathematical Prerequisites

- G. B. Thomas and R. L. Finney, *Calculus and Analytic Geometry*, 8th edition, Addison-Wesley, 1992.
- B. Kolman and D. R. Hill, *Elementary Linear Algebra*, 8th edition, Prentice-Hall, 2003.
- J. H. McClellan, R. W. Schafer, and M. A. Yoder, *Signal Processing First,* Prentice-Hall, 2003.

[1] The `.m` scripts will run with either Matlab or GNU Octave, which is freely available at http://www.gnu.org/software/octave. When using the scripts with Matlab, the Signal Processing Toolbox is required; all scripts have been tested with Matlab v7.10/R2010a, but are expected to work with older versions of Matlab. For Octave, the scripts were tested with Octave v3.2.3 and the required Octave-Forge toolboxes `signal` v1.0.11, `specfun` v1.0.9, `optim` v1.0.12, `miscellaneous` v1.0.9, and `audio` v1.1.4. When using Octave, the script `firpm_octave.m` can be renamed `firpm.m` so that identical code will run in the two platforms.

Contents

With thanks to ...

Tom Robbins for his foresight, encouragement, and perseverance.

The professors who have taught from earlier versions of this book: Mark Andersland, Marc Buchner, Brian Evans, Edward Doering, Tom Fuja, Brian Sadler, Michael Thompson, and Mark Yoder.

The classes of students who have learned from earlier versions of this book: ECE436 and ECE437 at the University of Wisconsin, EE467 and EE468 at Cornell University, and ECE 3311 at Worcester Polytechnic Institute.

Teaching assistants and graduate students who have helped refine this material: Jai Balkrishnan, Rick Brown, Wonzoo Chung, Qingxiong Deng, Matt Gaubatz, Jarvis Haupt, Sean Leventhal, Yizheng Liao, Katie Orlicki, Jason Pagnotta, Adam Pierce, Johnson Smith, Charles Tibenderana, and Nick Xenias.

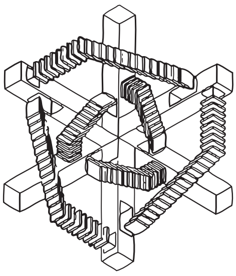
The companies that have provided data or access to engineers, and joined us in the field: Applied Signal Technology, Aware, Fox Digital, Iberium, Lucent Technologies, and NxtWave Communications (now AMD).

The many talented academics and engineers who have helped hone this material: Raul Casas, Tom Endres, John Gubner, Rod Kennedy, Mike Larimore, Rick Martin, Phil Schniter, John Treichler, John Walsh, Evans Wetmore, and Robert Williamson.

The staff of talented editors at Cambridge University Press: Phil Meyler, Sarah Finlay, Sarah Matthews, Steven Holt, and Joanna Endell-Cooper. Where would we have been without LaTeX guru Sehar Tahir?

Those with a well-developed sense of humor: Ann Bell and Betty Johnson.

Dread Zeppelin, whose 1999 performance in Palo Alto inspired the title *Telecommunication Breakdown.* In this new, improved, expanded (and more revealingly titled) version, the legacy continues: how many malapropic Led Zeppelin song titles are there?



Dedicated to ...

Samantha and Franklin