Digital Signal Processing

Principles and Applications

Combining clear explanations of elementary principles, advanced topics and applications with step-by-step mathematical derivations, this textbook provides a comprehensive yet accessible introduction to digital signal processing. All the key topics are covered, including discrete-time Fourier transform, *z*-transform, discrete Fourier transform and FFT, A/D conversion, and FIR and IIR filtering algorithms, as well as more advanced topics such as multirate systems, the discrete cosine transform and spectral signal processing. Over 600 full-color illustrations, 200 fully worked examples, hundreds of end-of-chapter homework problems and detailed computational examples of DSP algorithms implemented in Matlab® and C aid understanding and help put knowledge into practice. A wealth of supplementary material accompanies the book online, including interactive programs for instructors, a full set of solutions and Matlab® laboratory exercises, making this the ideal text for senior undergraduate and graduate courses on digital signal processing.

Online resources available at www.cambridge.org/holton

- 600+ full-color figures
- Interactive programs for instructors
- A full set of solutions
- Extensive supplementary material, including 100+ pages of text and 70+ illustrations
- Matlab® laboratory exercises

Thomas Holton is Professor of Electrical and Computer Engineering at San Francisco State University with interests in speech and audio processing.

"Professor Holton has done a great service to faculty who teach digital signal processing. The material is developed in a clear and thorough manner with an excellent range of topics, from elementary to advanced and from theoretical to applied. Many insightful analytical and computational examples and homework problems are included, with Matlab intelligently integrated. This textbook is the clear frontrunner in a crowded field."

Howard Weinert, Johns Hopkins University

"... a student-friendly book, making learning DSP a fun journey."

Professor Xiyi Hang, California State University–Northridge

"The *Digital Signal Processing* (DSP) textbook written by Thomas Holton is an excellent textbook for undergraduate as well as graduate students. It is well written, very clearly defined and presents all DSP topics, using many examples including the use of Matlab from chapter 1. In 14 chapters, Dr. Holton covers all necessary materials for a thorough understanding of DSP concepts and practical applications of a subject which is very mathematical. There are 3 appendices, including a tutorial on Matlab. In addition, the text has a website for additional reference materials. It is without any reservations that I strongly endorse and recommend the DSP book by Professor Holton."

Mousavinezhad Hossein, Idaho State University

"The Holton text includes technical materials that a practicing engineer needs to know to prototype a fixed-coefficient DSP system architecture using Matlab. There are many unique features of this textbook, including a full chapter on visualizing frequency response from pole-zero plots, multi-color plots for better comprehension, rigorous derivation of all formulas, and up-to-date hardware- and software-based implementation ideas for the benefit of novice and practicing engineers. I strongly recommend its adoption."

Kalyan Mondal, Fairleigh Dickinson University

DIGITAL SIGNAL PROCESSING

PRINCIPLES AND APPLICATIONS

Thomas Holton
San Francisco State University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, 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

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108418447
DOI: 10.1017/9781108290050

First published 2021

Printed in Singapore by Markono Print Media Pte Ltd 2021

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

ISBN 978-1-108-41844-7 Hardback

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

To my parents, Gerald and Nina Holton.

CONTENTS

PREFACE

Digital signal processing (DSP) is still a comparatively young field. The first textbooks arrived in the 1970s and were relatively scholarly books for graduate students that assumed a level of mathematical sophistication and facility that was appropriate to the intended audience at the time. Over the last 50 years, DSP has evolved from an elective subject aimed at advanced students into a subject that is required in many undergraduate programs. However, I feel that many textbooks still owe their approach and their expectations of the abilities of their audience to books of an earlier era.

The purpose of this book is simple: to lay out the basic principles of digital signal processing, to excite you about the possibility of applying DSP to your own applications, and to give you the tools to do so. The book was originally written for my students, and for the bulk of students like them in engineering programs everywhere. My students have a range of mathematical abilities, from those who could appreciate an advanced book to those who sometimes struggle with basic mathematical material. I suspect this describes the profile of students in many programs around the country and the world. It has been a challenge to engage all of my students and get them to understand the material. This book evolved as my response to that challenge. It started, as many books do, as a series of supplementary notes to an undergraduate DSP course that I have been teaching for a number of years, and has now evolved into a full-featured text that not only covers the basics but also includes coverage of a range of topics to satisfy more advanced students and perhaps even practitioners in the field.

Noteworthy features of the book

Accessible, comprehensive text

In this book, I have striven for three things: *simplicity*, *clarity* and *thoroughness* of explanation. In writing the book, I had in my mind's eye explaining material to an average student, one-on-one, during office hours. Hence, the tone of the book is conversational, with an occasional dash of humor to keep the reader interested (I hope).

The content of the book is comprehensive, starting with patient, clear explanations of elementary principles and proceeding to more advanced topics, some contained in "starred (★) sections" in the text or as supplementary material on the web. A lot of effort went into the early chapters dealing with the impulse response, convolution, the discrete-time Fourier transform and the z-transform. They are particularly detailed, with a lot of examples, because it is essential that the students master this material before trying to move forward. While I have resisted the temptation to cover every single topic in this book, I have included a range of useful material not usually covered, particularly in the later chapters: Chapter 7 (Finite impulse response filters), Chapter 8 (Infinite impulse response filters), Chapter 12 (Discrete cosine transform), Chapter 13 (Multirate systems) and Chapter 14 (Spectral analysis). While some of these topics may be of more interest to advanced readers, my intention has been to make these topics accessible to everyone who has mastered the earlier material.

Good balance of theory and application

In any text such as this, there is also a trade-off between theory and practical discussion. Here, I have followed the famous advice of Yogi Berra[1]: "When you come to a fork in the road, take it." In other words, I have tried to give the reader a good balance of the two. In terms of theory, I have not assumed that I can safely leave steps out of important derivations. So, where necessary, derivations are worked out fully in the text, sometimes using more than one approach. Some of the more algebraically exhausting and tedious derivations have been relegated to the problems, but I have nevertheless tried to structure those problems in such a manner that readers can follow the gist of the derivations even if they do not wish to work them out themselves.

In terms of practical applications, I am mindful of the fact that many (most?) engineering students are likely aiming to get jobs in industry. If they are tasked to use their DSP knowledge, it will most likely be to implement existing algorithms or to use existing design tools to design filters and transforms for specific applications. Accordingly, most chapters have sections that discuss details of hardware or software implementations and applications, including computational examples. Some chapters, for example the supplementary sections of Chapter 6 (A/D and D/A conversion), discuss relevant hardware in considerable detail, something that is often not done in DSP texts. Many students may not see this material in any other course and they ought to in order to understand how DSP theory is put into practice.

Multi-level integration of Matlab®

Matlab is a standard computing language and computing environment for DSP algorithm development and simulation. So, it has become *de rigueur* to include Matlab in any text, sometimes (I feel) gratuitously. I have thoroughly integrated Matlab into the text in several ways:

1. **Language basics**: In early chapters and in Appendix C, the book provides an introduction to the features of the basic Matlab language, and presents a series of examples that indicate features and pitfalls of the language of which the student and practitioner must be aware.
2. **DSP toolbox functions**: While I have resisted cataloging every relevant Matlab function, in later chapters, the text describes the syntax of essential Matlab functions of the signal processing toolbox and gives examples of their use. For example, in Chapter 7 on finite impulse response (FIR) filters, I show how to use Matlab toolbox functions for designing window-based, frequency-sampling, least-square-error and optimum filters, including `rectwin`, `hamming`, `hann`, `hanning`, `blackman`, `kaiserord`, `kaiser`, `window`, `fir1`, `fir2`, `firls`, `firpm` and `firpmord`. But…
3. **DSP algorithms from scratch**: While providing a summary of Matlab's functions is necessary, it isn't adequate for a couple of reasons. First, simply giving the syntax of Matlab's basic filter design functions (such as those listed above) and a quick example or two of their use doesn't give students any clue how to design or implement a filter for themselves, and also weds them to the Matlab toolbox functions. So, the text includes examples of filter algorithms written *from scratch* without those functions. For example, in Chapter 7

[1] Lorenzo Pietro ("Yogi") Berra (1925–2015) was one of the greatest catchers in American baseball history, a winner of 10 World Series Championships in 18 seasons with the New York Yankees, a team he also managed later in his career. Berra was known for his many unintentionally amusing yet wise aphorisms, such as this timeless one, "It ain't over 'till it's over."

(Finite impulse response filters), you will find many Matlab examples of the design of window-based, frequency-sampling and least-square-error designs written in basic Matlab that a student can easily test. In Chapter 8 (Infinite impulse response filters), example code for the design for all filters is provided. In Chapter 12 (Discrete cosine transform), Chapter 13 (Fast Fourier transform) and Chapter 14 (Spectral analysis), there are numerous pieces of code that exemplify every major point.

Providing basic code that does not rely on the Matlab toolboxes is also important because basic Matlab is similar enough to other languages (e.g., Python) that users who understand the basic Matlab code in the chapters can easily port the algorithms to the language of their choice. This is significant, because there are ongoing efforts, particularly in the academic community, to move away from Matlab, which is proprietary and expensive, to open-source languages, either Matlab clones such as Octave, or general-purpose languages such as Python.

4. **End-of-chapter laboratory exercises**: In my DSP course, there are mandatory, weekly, open-ended, project-based laboratory exercises in which the students use Matlab to design, code, debug and test algorithms to solve specific problems related to the material they are learning in lecture. I have adapted a number of these exercises into special sections of Matlab laboratory exercises that appear as supplementary content at www.cambridge.org/holton. I have made an effort to make these laboratory exercises relevant to real applications, including DTMF-to-text decoding and audio compression.

Examples and illustrations

The chapters contain over 200 fully worked theoretical and practical Matlab-based examples, and over 600 illustrations, many in color, designed to complement the text. Each chapter has end-of-chapter problems, ranging from basic exercises to more complex examples that extend the discussion of the text.

Detailed description of the book's chapters

The book has 14 chapters of which the first five comprise the essential, core material upon which everything else depends. The remaining chapters contain a variety of topics, a selection of which would serve for a one-semester course when added to the core chapters. For example, one could add the first sections of Chapter 6 (A/D and D/A conversion), up to and including the discussion of upsampling and downsampling, the first sections of Chapter 7 (Finite impulse response filters) covering window-based filters, the first sections of Chapter 9 (Filter architecture) covering cascade and parallel architectures, much of Chapter 10 (Discrete Fourier transform), and the first section of Chapter 11 (Fast Fourier transform), which deals with the decimation-in-time (DIT) transform. The later sections of many chapters, such as Chapter 7 (Finite impulse response filters), Chapter 8 (Infinite impulse response filters) as well as all of Chapter 12 (Discrete cosine transform), Chapter 13 (Multirate systems) and Chapter 14 (Spectral analysis) contain a variety of more-or-less advanced topics and applications that might provide interesting supplements to the basic course material.

Here is a description of the content of each of the chapters:

Chapter 1 (Discrete-time signals and systems) introduces the fundamental concepts of signals and systems in the discrete-time domain that will be required in the remaining chapters. It starts with a description of the basic signals – impulses, steps, sinusoids and exponentials, both real and complex – and the basic operations upon them, such as flip and shift. Then, using a series of examples, the chapter describes important system properties: linearity, time invariance, causality and stability.

Chapter 2 (Impulse response) defines the notion of the impulse response – the response of the system to an impulse – and shows that for linear time-invariant (LTI) systems, the impulse response is all that is necessary to compute the response of the system to any input through the operation of convolution. The properties of convolution are presented with examples from finite impulse response (FIR) and infinite impulse response (IIR) systems. We demonstrate simple tests for system causality and stability based on the impulse response. Deconvolution is described as well as two methods of convolving infinitely long sequences: the overlap-add and overlap-save methods.

Chapter 3 (Discrete-time Fourier transform) deals with the basic discrete-time Fourier transform (DTFT) material. The chapter describes the response of LTI systems to complex exponential sequences and shows how the orthogonality of these sequences leads to the definition of the DTFT and its inverse. The chapter features a catalog of the important signals and their transforms and a comprehensive treatment of the main properties of the DTFT. This chapter also introduces the idea of linear-phase systems, which are central to the development of FIR filters in Chapter 7.

Chapter 4 (*z*-transform) develops the *z*-transform as a generalization of the DTFT that is designed to handle a larger variety of signals and systems. There is an extended discussion of the visualization of the singularities – poles and zeros – and the region of convergence of the *z*-transform by means of pole-zero plots. The following section of the chapter builds upon the discussion of linear-phase systems started in Chapter 3 and shows that the poles and zeros of linear-phase systems can only occur at constrained locations on the *z*-plane. The properties of the *z*-transform are developed and applied to a number of examples, including the solution of linear constant-coefficient difference equations (LCCDEs). Finally, there is a brief discussion of the unilateral *z*-transform and its application to the solution of LCCDEs with initial conditions.

Chapter 5 (Frequency response) complements the preceding chapter on the *z*-transform. It shows how to visualize the magnitude and phase of the frequency response directly from the *z*-transform in many simple cases, and even design basic filters by inspection. This chapter also introduces allpass filters and minimum-phase systems.

Chapter 6 (A/D and D/A conversion) covers the basic material of analog-to-digital (A/D) and digital-to-analog (D/A) conversion that is required to understand how to apply DSP to real-world analog signals. The chapter first goes over analog sampling and reconstruction and then adds the idea of the "continuous-to-discrete" converter. Anti-aliasing filters on the A/D and anti-imaging filters on the D/A are discussed in the context of a digital recording application. The chapter includes coverage of basic sample-rate conversion: upsampling, downsampling and resampling. One section of the chapter is devoted to the effects of quantization, which is inherent in sampled-data systems, and an explanation of how quantization noise can be reduced by oversampling and noise-shaping A/D converters. Supplementary material to this chapter includes a survey of A/D and D/A conversion hardware.

Chapter 7 (Finite impulse response filters) deals with many varieties of FIR filters, and is one of the longest in the book. It describes a number of methods of designing FIR filters to meet specific design criteria. The first half of the chapter is devoted to window-based filters, which are the most important to understand and implement. The windows (e.g., Hamming, Hann and Kaiser) are necessary not only for filter design, but for windowing data in spectral applications, a topic which will appear again in Chapter 14 (Spectral analysis). The second half of the chapter covers an array of more advanced filtering topics, including frequency-sampling, least-square and optimum (Parks–McClellan) filter design. The chapter is complemented by multiple Matlab programs for the design of filters from scratch.

Chapter 8 (Infinite impulse response filters) describes the design of discrete-time filters that are based on analog filter prototypes. The first half of the chapter describes in detail the theory of design of classical analog Butterworth, Chebyshev, inverse Chebyshev and elliptic filters, complemented by multiple pieces of Matlab code that implement these filters. The chapter then describes several methods of converting from an analog prototype to a discrete-time realization, including the impulse-invariance and bilinear transformations, and shows why the bilinear-transformation method is preferred. The next section of the chapter describes spectral transformations that can be used to convert a prototype discrete-time lowpass filter into a highpass, bandpass or bandstop filter. The last section of the chapter describes how to perform zero-phase filtering to compensate for the nonlinear phase delay of an IIR filter in a non-real-time application.

Chapter 9 (Filter architecture) is concerned with the manner in which the filters described in Chapters 7 and 8 can be implemented in software or hardware. The first section introduces the signal-flow graph, which is an implementation-independent way of representing the basic architecture of FIR and IIR filters. The discussion covers two basic ways of implementing these filters: cascade and parallel configurations in both canonical and transpose variants, as well as lattice and lattice-ladder filters, which are useful in a number of special applications, particularly in speech and audio signal processing. The subsequent section discusses the effect of the quantization of filter coefficients on the filter's response. Finally, the last section considers a number of issues that arise in practical implementations in software and special-purpose hardware implementations.

Chapter 10 (Discrete Fourier transform) begins with the derivation and implementation of the discrete Fourier transform (DFT) from the point of view of the DTFT of periodic sequences. The chapter presents the DFT of basic signals and a description of the fundamental properties of the DFT, with particular attention being paid to circular convolution and to its interpretation as linear convolution followed by time-domain aliasing. Additional topics include the use of the DFT to increase the resolution in the frequency domain by zero-padding in the time domain and the corollary operation of upsampling in the time-domain by zero-padding in the frequency domain.

Chapter 11 (Fast Fourier transform) deals with the fast Fourier transform (FFT), which is the practical implementation of the DFT covered in Chapter 10. The chapter first derives in detail the most common variant of the transform: the radix-2 decimation-in-time (DIT) transform. There follow sections discussing radix-4 and composite (mixed-radix) transforms, the decimation-in-frequency (DIF) transform and transposed forms of the DIT and DIF transforms. The chapter then shows how considerable computational savings can be achieved by exploiting the symmetry of the DFT/FFT in the calculation of real sequences. The chapter concludes with a discussion of recursive and non-recursive strategies for implementing the FFT.

Chapter 12 (Discrete cosine transform) develops the discrete cosine transform (DCT) and its application to sound and image encoders that people use on a daily basis: their cellphones, music players and cameras. The chapter begins with a description of the four main variants of the DCT, and concentrates on the derivation and implementation of the most common variant, the DCT-II. Supplementary material for this chapter also covers the modified windowed DCT (MDCT), which is the main transform used in audio codecs such as the MP3 format. The last two main sections of the chapter give a fairly broad overview of compression of audio by the MP3 codec and compression of images by the JPEG standard of MP3 and JPEG compression. There is quite a bit of non-DSP detail here, such as a description of psychophysical auditory and visual models, color spaces and minimum entropy coding, but I feel that it is important to show how DSP interfaces with all these other disciplines in order to create workable systems.

Chapter 13 (Multirate systems) is somewhat more advanced than the preceding chapters, but polyphase concepts are now so pervasive in filtering and sample-rate conversion that it seemed necessary to create an accessible treatment. To do this, the chapter starts with a derivation of polyphase upsampling, downsampling and resampling from a time-domain perspective, which is perhaps the most intuitive way to introduce the subject. Then, the same material is reexamined using the z-transform, which makes the concepts such as the key multirate identities easier to derive and understand. Since polyphase methods are often paired with multirate filters, multistage filters and filter banks, this chapter is the logical place for that material. There is a section on multistage systems for decimation and interpolation, another section on multistage and multirate filtering and a section on half-band and L-band filters for downsampling and upsampling. Supplementary material associated with this chapter contains a discussion of multirate filter banks including quadrature mirror filters (QMF) and a variety of maximally decimated filter banks.

Chapter 14 (Spectral analysis) covers a wide range of topics relating to spectral analysis, which is the process of measuring, estimating and characterizing the frequency content of signals. The chapter starts with an explanation of two fundamental issues that affect the practical measurement of the frequency spectrum of signals: data windowing and frequency sampling. The following section develops the short-term Fourier transform (STFT), which is useful in many cases where signals are inherently time-varying, such as the analysis of speech and music. The remaining sections of the chapter are concerned with the spectral analysis of random signals. The chapter covers two broad categories of techniques for spectral measurement and estimation of probabilistic systems: nonparametric and parametric. Nonparametric spectral methods do not depend on knowing any specific information of the signal being analyzed. These include several variants of the periodogram (e.g., Bartlett's method and Welch's method). Parametric methods, which are covered next, are based on estimating the parameters of models that have been specifically designed to match the characteristics of the process being analyzed, such as speech. The chapter discusses the autoregressive (AR) model in detail, and derives the

Yule–Walker equations and their solution via the Levinson–Durbin algorithm. The chapter concludes with a discussion of linear prediction and its application to encoding and decoding speech sounds using the source-filter model of speech production.

Appendix A (Linear algebra) provides a brief review of linear algebra used throughout the book, including the solution of systems of linear equations in matrix form and a derivation of the normal equations.

Appendix B (Numeric representations) discusses how integers and floating-point numbers are represented in binary form.

Appendix C (Matlab tutorial) is a brief introduction to Matlab, which highlights features that make Matlab suited to the processing of arrays, and also indicates a number of potential pitfalls for programmers.

Appendix D (Probability and random processes) is a relatively comprehensive review of the topics that are prerequisite for understanding the material in Chapter 14 (Spectral analysis). The appendix covers probability distributions and density functions, basic random processes including discussions of stationary processes, filtered random processes and power spectral density.

Supplementary material In addition to the material published in the book, there are about 100 pages of supplementary material available at www.cambridge.org/holton containing an elaboration and extension of topics discussed in the chapters. Examples of this more advanced material include an extensive discussion of the hardware implementation of A/D and D/A converters (Chapter 6), derivation and examples of the Parks–McClellan algorithm for optimal FIR filtering (Chapter 7), development of the modified discrete cosine transform (MDCT) (Chapter 12) and a discussion of multirate filter banks, including quadrature and conjugate mirror filters, and complex-modulated filter banks (Chapter 13).