

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

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

© Thomas Holton 2021

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

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.

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

To my parents, Gerald and Nina Holton.

CONTENTS

Preface xxi

- 1 Discrete-time signals and systems** 1
 - Introduction 1
 - 1.1 Two signal processing paradigms 1
 - 1.2 Advantages of digital signal processing 3
 - 1.3 Applications of DSP 5
 - 1.4 Signals 6
 - 1.4.1 Signal classification 7
 - 1.4.2 Discrete-time signals 8
 - 1.5 Basic operations on signals 10
 - 1.5.1 Shift 10
 - 1.5.2 Flip 11
 - 1.5.3 Flip and shift 12
 - 1.5.4 Time decimation 13
 - 1.5.5 Time expansion 14
 - 1.5.6 Operation on multiple sequences 14
 - 1.6 Basic sequences 15
 - 1.6.1 Impulse 15
 - 1.6.2 Unit step 19
 - 1.6.3 Pulse 21
 - 1.6.4 Power-law sequences 22
 - 1.6.5 Sinusoidal sequences 22
 - 1.6.6 Complex exponential sequences 26
 - 1.6.7 Sequence classification 28
 - 1.7 Systems 32
 - 1.7.1 Discrete-time scalar multiplier 32
 - 1.7.2 Offset 34
 - 1.7.3 Squarer 34
 - 1.7.4 Shift 35
 - 1.7.5 Moving-window average 36
 - 1.7.6 Summer 37
 - 1.7.7 Switch 38
 - 1.7.8 Linear constant-coefficient difference equation (LCCDE) 38
 - 1.8 Linearity 39
 - 1.8.1 The additivity property 39
 - 1.8.2 The scaling property 40
 - 1.8.3 Discrete-time scalar multiplier 41
 - 1.8.4 Offset 41
 - 1.8.5 Squarer 42
 - 1.8.6 Shift 42
 - 1.8.7 Moving-window average 43
 - 1.8.8 Summer 43
 - 1.8.9 Switch 44

1.8.10	Linear constant-coefficient difference equation	44
1.8.11	The “zero-in, zero-out” property of linear systems	46
1.9	Time invariance	46
1.9.1	Discrete-time scalar multiplier	47
1.9.2	Offset	47
1.9.3	Squarer	47
1.9.4	Shift	47
1.9.5	Moving-window average	47
1.9.6	Summer	48
1.9.7	Switch	48
1.9.8	Linear constant-coefficient difference equation (LCCDE)	49
1.10	Causality	49
1.10.1	Discrete-time scalar multiplier	50
1.10.2	Offset	50
1.10.3	Squarer	50
1.10.4	Shift	50
1.10.5	Moving-window average	50
1.10.6	Summer	50
1.10.7	Switch	51
1.10.8	Linear constant-coefficient difference equation (LCCDE)	51
1.11	Stability	51
1.11.1	Discrete-time scalar multiplier	51
1.11.2	Offset	51
1.11.3	Squarer	51
1.11.4	Shift	52
1.11.5	Moving-window average	52
1.11.6	Summer	52
1.11.7	Switch	52
1.11.8	Linear constant-coefficient difference equation (LCCDE)	52
	Summary	52
	Problems	53
2	Impulse response	63
	Introduction	63
2.1	FIR and IIR systems	63
2.1.1	Finite impulse response (FIR) systems	63
2.1.2	Infinite impulse response (IIR) systems	65
2.1.3	Response of a system to a flipped and shifted impulse	66
2.2	Convolution	68
2.2.1	Direct-summation method	69
2.2.2	Flip-and-shift method	71
2.2.3	Convolution examples	72
2.3	Properties of convolution	78
2.3.1	The commutative property	78
2.3.2	The associative property	81
2.3.3	The distributive property	82

2.4	Stability and causality	83
2.4.1	Stability	84
2.4.2	Causality	88
2.5	★Convolution reinterpreted	89
2.5.1	Convolution as polynomial multiplication	89
2.5.2	Convolution using Matlab	90
2.5.3	Convolution as matrix multiplication	91
2.6	★Deconvolution	93
2.7	★Convolution of long sequences	97
2.7.1	Overlap-add method	97
2.7.2	Overlap-save method	99
2.8	Implementation issues	100
	Summary	101
	Problems	101
3	Discrete-time Fourier transform	113
	Introduction	113
3.1	Complex exponentials and sinusoids	113
3.1.1	Response of LTI systems to complex exponentials	113
3.1.2	Response of linear time-invariant systems to sinusoids	116
3.2	Discrete-time Fourier transform (DTFT)	118
3.2.1	Orthogonality of complex exponential sequences	118
3.2.2	Definition and derivation	119
3.2.3	Notation of the DTFT	120
3.2.4	Existence of the DTFT	121
3.2.5	The system function (again)	122
3.2.6	Periodicity of the DTFT	122
3.2.7	DTFT of finite-length sequences	122
3.2.8	DTFT of infinite-length sequences	127
3.3	Magnitude and phase description of the DTFT	129
3.3.1	Magnitude and phase of the DTFT of an impulse	129
3.3.2	Essential phase discontinuities	131
3.4	Important sequences and their transforms	135
3.5	Symmetry properties of the DTFT	135
3.5.1	Time reversal	136
3.5.2	Conjugate symmetry and antisymmetry	136
3.5.3	Even and odd symmetry	138
3.5.4	Consequences of symmetry	139
3.5.5	★ Complex sequences	140
3.5.6	Symmetry summary	142
3.6	Response of a system to sinusoidal input	143
3.7	Linear-phase systems	145
3.7.1	Causal symmetric sequences	145
3.7.2	Causal antisymmetric sequences	147
3.7.3	Time delay and group delay	148
3.8	The inverse discrete-time Fourier transform	152

- 3.9 Using Matlab to compute and plot the DTFT 156
- 3.10 DTFT properties 157
 - 3.10.1 Linearity 157
 - 3.10.2 Delay (shifting) property 158
 - 3.10.3 Complex modulation (frequency shift) property 159
 - 3.10.4 Convolution 166
 - 3.10.5 Using the convolution property of the DTFT to do filtering 167
 - 3.10.6 Deconvolution and system identification using the convolution property 170
 - 3.10.7 Convolution properties 172
 - 3.10.8 Understanding filtering in the frequency domain 173
 - 3.10.9 Multiplication (windowing) property 178
 - 3.10.10 ★ Time- and band-limited systems 181
 - 3.10.11 ★ Spectral and temporal ambiguity 183
 - 3.10.12 Time-reversal property 184
 - 3.10.13 Differentiation property 186
 - 3.10.14 Parseval's theorem 186
 - 3.10.15 DC- and π -value properties 187
 - 3.10.16 ★ Using the DTFT to solve linear constant-coefficient difference equations 187
 - 3.10.17 Summary of DTFT properties 189
- 3.11 ★ The relation between the DTFT and the Fourier series 189
 - Summary 190
 - Problems 190
- 4 z-transform 211**
 - Introduction 211
 - 4.1 The z -transform 212
 - 4.2 The singularities of $H(z)$ 213
 - 4.2.1 Pole-zero plots 214
 - 4.2.2 Left-sided sequences 217
 - 4.2.3 Relation between the z -transform and DTFT 218
 - 4.2.4 Multiple poles and zeros 219
 - 4.2.5 Finding the z -transform from the pole-zero plot 227
 - 4.2.6 Complex poles and zeros 227
 - 4.2.7 Some important transforms 231
 - 4.2.8 Finite-length sequences 231
 - 4.2.9 Plotting pole-zero plots with Matlab 234
 - 4.3 ★ Linear-phase FIR systems 234
 - 4.3.1 Complex zeros 237
 - 4.3.2 Real zeros 237
 - 4.4 The inverse z -transform 240
 - 4.4.1 All-zero systems 240
 - 4.4.2 Distinct real poles 241
 - 4.4.3 Complex poles 244
 - 4.4.4 Multiple (repeated) poles 245
 - 4.4.5 Improper rational functions 247
 - 4.4.6 Using Matlab to compute the inverse z -transform 250

4.5	Properties of the z -transform	254
4.5.1	Linearity	255
4.5.2	Shifting property	255
4.5.3	Differentiation property	256
4.5.4	Time reversal property	257
4.5.5	Convolution property	259
4.5.6	Applications of convolution	262
4.5.7	Initial-value theorem	263
4.5.8	Final-value theorem	263
4.6	Linear constant-coefficient difference equations (LCCDE)	265
4.6.1	LCCDE of FIR systems	265
4.6.2	LCCDE of IIR systems	265
4.6.3	Relation between LCCDE and $H(z)$	266
4.6.4	Using Matlab to solve LCCDEs	267
4.6.5	Inverse filter	268
4.7	★ The unilateral z -transform	274
	Summary	274
	Problems	275
5	Frequency response	287
	Introduction	287
5.1	The computation of $H(\omega)$ from $H(z)$	287
5.2	Systems with a single real zero	287
5.2.1	Direct computation	288
5.2.2	Graphical method	288
5.3	Systems with a single real pole	295
5.4	Multiple real poles and zeros	303
5.5	Complex poles and zeros	306
5.6	★3-D visualization of $H(\omega)$ from $H(z)$	308
5.7	Allpass filter	310
5.7.1	Real allpass filter	311
5.7.2	Multiple poles and zeros	313
5.7.3	Allpass filters with complex poles and zeros	314
5.7.4	General allpass filter	315
5.7.5	Systems with the same magnitude	316
5.7.6	Practical applications of allpass filters	320
5.8	Minimum-phase-lag systems	321
	Summary	323
	Problems	323
6	A/D and D/A conversion	331
	Introduction	331
6.1	Overview of A/D and D/A conversion	331
6.2	Analog sampling and reconstruction	333
6.2.1	Analog sampling	334
6.2.2	The sampling theorem	335
6.2.3	The Nyquist sampling criterion	342

6.2.4	Oversampling, undersampling and critical sampling	343
6.2.5	★ Sampling a cosine	348
6.3	Conversion from continuous time to discrete time and back	352
6.3.1	The continuous-to-discrete (C/D) converter	352
6.3.2	Spectrum of the discrete-time sequence	353
6.3.3	The discrete-to-continuous (D/C) converter	355
6.3.4	Summary	356
6.4	Anti-aliasing and reconstruction filters	357
6.4.1	The anti-aliasing filter	357
6.4.2	A digital recording application	359
6.4.3	Reconstruction filter	361
6.4.4	Revised model of D/A conversion	361
6.5	Downsampling and upsampling	364
6.5.1	Downsampling	364
6.5.2	Decimation and aliasing	368
6.5.3	Oversampling A/D converter in a digital recording application	371
6.5.4	Upsampling	372
6.5.5	★ Upsampling a cosine	375
6.5.6	Upsampling D/A converter in a digital recording application	377
6.5.7	Resampling	378
6.6	Matlab functions for sample-rate conversion	381
6.7	★ Quantization	382
6.7.1	Model of quantization	383
6.7.2	Quantization error	386
6.7.3	Noise reduction by oversampling	388
6.7.4	★ Noise-shaping A/D converters	391
6.7.5	★ Sigma-delta A/D converters	398
6.8	★ A/D converter architecture	401
6.9	★ D/A converter architecture	401
	Summary	402
	Problems	402
7	Finite impulse response filters	409
	Introduction	409
7.1	Linear-phase FIR filters	410
7.1.1	Types of linear-phase filters	410
7.1.2	Basic properties of linear-phase filters	413
7.1.3	★ Time-aligned and zero-phase FIR filters	415
7.2	Preliminaries of filter design	415
7.2.1	Specification of filter characteristics	415
7.2.2	The ideal lowpass filter	417
7.2.3	The optimum least-square-error FIR filter	417
7.2.4	Even- and odd-length causal filters	419
7.3	Window-based FIR filter design	420
7.3.1	Rectangular window filter	420

7.3.2	Raised cosine window filters	426
7.3.3	Kaiser window	436
7.4	Highpass, bandpass and bandstop FIR filters	437
7.5	Matlab implementation of window-based FIR filters	444
7.5.1	Matlab functions that implement FIR filtering	446
7.6	★ Spline and raised-cosine FIR filters	449
7.6.1	FIR filters designed using splines	449
7.6.2	FIR filters designed using raised cosines	451
7.7	★ Frequency-sampled FIR filter design	453
7.7.1	Inverse DFT	454
7.7.2	Frequency sampling as interpolation	458
7.7.3	Design formulas	460
7.7.4	Simultaneous equations	460
7.7.5	Matlab implementation of frequency-sampled filters	463
7.8	★ Least-square-error FIR filter design	463
7.8.1	Discrete least-square-error FIR filters	464
7.8.2	★ Integral least-square-error FIR filter design	470
7.9	★ Optimal lowpass filter design	470
7.10	Multiband filters	471
7.11	★ Differentiator	472
7.12	★ Hilbert transformer	474
7.12.1	Derivation of the Hilbert transformer	474
7.12.2	FIR implementation of a Hilbert transformer	477
7.12.3	FFT implementation of a Hilbert transformer	479
7.12.4	Applications of the Hilbert transformer	479
	Summary	481
	Problems	482
8	Infinite impulse response filters	499
	Introduction	499
8.1	Definition of the IIR filter	500
8.2	Overview of analog filter design	501
8.2.1	Parameter definitions	502
8.2.2	Butterworth filter	504
8.2.3	★ Chebyshev filter	512
8.2.4	★ Inverse Chebyshev filter	520
8.2.5	★ Elliptic filter	525
8.2.6	Summary	532
8.3	★ Impulse invariance	533
8.3.1	Impulse-invariance approach	533
8.3.2	Impulse-invariance design procedure	535
8.3.3	Mapping of s -plane to z -plane	541
8.4	Bilinear transformation	544
8.4.1	Forward-difference approximation	546
8.4.2	Backward-difference approximation	547

8.4.3	Bilinear transformation	548
8.4.4	Bilinear-transformation procedure	549
8.4.5	Cascade of second-order sections	556
8.5	★ Spectral transformations of IIR filters	562
8.5.1	Lowpass-to-lowpass transformation	563
8.5.2	Lowpass-to-highpass transformation	567
8.5.3	Lowpass-to-bandpass transformation	570
8.5.4	Lowpass-to-bandstop transformation	573
8.6	★ Zero-phase IIR filtering	575
	Summary	578
	Problems	578
9	Filter architecture	597
	Introduction	597
9.1	Signal-flow graphs	597
9.2	Canonical filter architecture	599
9.2.1	First-order filters	600
9.2.2	Canonical filter architecture	602
9.3	Transposed filters	603
9.4	Cascade architecture	606
9.4.1	Allpass filters	609
9.4.2	Using Matlab to design cascade filters	610
9.5	Parallel architecture	611
9.5.1	Using Matlab to design parallel filters	612
9.6	FIR filters	613
9.7	★ Lattice and lattice-ladder filters	614
9.7.1	FIR lattice filters	615
9.7.2	Specialized FIR lattice filters	620
9.7.3	IIR lattice filters	621
9.7.4	Allpass lattice filters	625
9.7.5	Lattice-ladder IIR filters	625
9.7.6	Stability of IIR filters revisited	628
9.8	★ Coefficient quantization	630
9.8.1	Systems with poles	630
9.8.2	Systems with zeros	636
9.8.3	Systems with poles and zeros	638
9.8.4	Pairing poles and zeros	640
9.8.5	Coefficient quantization of lattice filters	641
9.9	★ Implementation issues	642
9.9.1	Software implementation	642
9.9.2	Hardware implementation	647
	Summary	652
	Problems	652
10	Discrete Fourier transform (DFT)	657
	Introduction	657
10.1	Derivation of the DFT	658

10.1.1	The inverse discrete Fourier transform (IDFT)	662
10.1.2	Orthogonality of complex exponential sequences	664
10.1.3	Periodicity of the DFT	665
10.1.4	★ Conditions for the reconstruction of a sequence from the DFT	665
10.2	DFT of basic signals	665
10.2.1	DFT of an impulse	665
10.2.2	DFT of a pulse	666
10.2.3	DFT of a constant	667
10.2.4	DFT of a complex exponential sequence	667
10.2.5	DFT of a sinusoid	667
10.2.6	Resolution and frequency mapping of the DFT	669
10.2.7	Summary (so far)	670
10.3	Properties of the DFT	670
10.3.1	Linearity	671
10.3.2	Complex conjugation	671
10.3.3	Symmetry properties of the DFT	671
10.3.4	Circular time shifting	676
10.3.5	Circular time reversal	680
10.3.6	Circular frequency shift	682
10.3.7	Circular convolution	682
10.3.8	Multiplication	687
10.3.9	Parseval's theorem	689
10.3.10	Summary of DFT properties	690
10.4	★ Matrix representation of the DFT	690
10.5	★ Using the DFT to increase resolution in the time and frequency domains	692
10.5.1	Increasing frequency resolution by zero-padding in the time domain	692
10.5.2	Upsampling in the time domain by zero-padding in the frequency domain	694
10.5.3	★ Recovery of the DTFT from the DFT	697
	Summary	697
	Problems	697
11	Fast Fourier transform (FFT)	707
	Introduction	707
11.1	Radix-2 FFT transforms	708
11.1.1	Decimation-in-time FFT	709
11.1.2	Computational gain	715
11.1.3	Bit reversal	717
11.1.4	★ Decimation-in-frequency FFT	719
11.2	★ Radix-4 FFT	721
11.2.1	The radix-4 decomposition	721
11.2.2	The radix-4 transform as a combination of radix-2 transforms	723
11.3	★ Composite (mixed-radix) FFT	725
11.3.1	FFTs of composite size	726
11.3.2	Mixed radix-2 and radix-4 transform	729
11.3.3	Transposed and split-radix transforms	729
11.4	Inverse FFT	730

11.5	Matlab implementation	731
11.6	FFT of real sequences	732
11.6.1	Properties of DFTs (revisited)	732
11.6.2	N -point FFT of two real N -point sequences	733
11.6.3	N -point IFFT of two N -point transforms	735
11.6.4	★ $N/2$ -point FFT of a real N -point sequence	736
11.6.5	★ $N/2$ -point IFFT of an N -point transform	737
11.7	FFT resolution	739
11.7.1	Increasing the resolution of the FFT	740
11.7.2	Decreasing the resolution of the FFT	740
11.8	Fast convolution using the FFT	741
11.8.1	Convolution of fixed-length input sequences	741
11.8.2	Block convolution using the FFT	743
11.8.3	★ Using both DIT and DIF transforms	746
11.8.4	Matlab support for convolution using the FFT	747
11.9	★ The Goertzel algorithm	747
11.10	Iterative and recursive implementations	751
11.10.1	Iterative implementation	751
11.10.2	Recursive implementation	753
11.11	Implementation issues	755
	Summary	757
	Problems	757
12	Discrete cosine transform (DCT)	761
	Introduction	761
12.1	The DCT	761
12.1.1	★ Periodically extended sequences	762
12.1.2	“The” discrete cosine transform (DCT-II)	764
12.1.3	The inverse discrete cosine transform (IDCT)	766
12.1.4	The four principal DCT variants	769
12.1.5	Properties of the DCT	770
12.1.6	★ Matrix form of the DCT and IDCT	771
12.1.7	★ Energy compaction of the DCT	773
12.1.8	★ Implementation of the DCT-II and IDCT-II	774
12.1.9	★ The modified discrete cosine transform (MDCT)	776
12.2	MPEG audio compression	778
12.2.1	The MP3 encoder	780
12.2.2	Hybrid filter bank and MDCT	781
12.2.3	The psychoacoustic model	781
12.2.4	Bit allocation and quantization	787
12.2.5	Minimum entropy coding	788
12.3	JPEG image compression	792
12.3.1	Color processing	793
12.3.2	DCT transformation and quantization	796
12.3.3	Coefficient encoding	801
12.3.4	Implementation of 2D-DCT	802

Summary	802
Problems	803
13 Multirate systems	815
Introduction	815
13.1 Polyphase downsampling	816
13.1.1 Review of downsampling	816
13.1.2 Polyphase implementation of downsampling	818
13.1.3 Downsampling summary	822
13.2 Polyphase upsampling	826
13.2.1 Review of upsampling	826
13.2.2 Polyphase implementation of upsampling	828
13.2.3 Upsampling summary	832
13.3 ★ Polyphase resampling	833
13.4 Transform analysis of polyphase systems	833
13.4.1 Basic decimation and expansion identities	833
13.4.2 Multirate identities of downsampling and upsampling	835
13.4.3 Transform analysis of polyphase downsampling	840
13.4.4 Transform analysis of polyphase upsampling	841
13.4.5 Transform analysis of polyphase resampling	842
13.4.6 ★ Matlab implementation of polyphase sample-rate conversion algorithms	848
13.5 Multistage systems for downsampling and upsampling	848
13.5.1 Multistage downsampling	849
13.5.2 Multistage upsampling	855
13.6 Multistage and multirate filtering	858
13.6.1 Multistage interpolated FIR (IFIR) filters	858
13.6.2 Multirate lowpass filtering	863
13.7 Special filters for multirate applications	865
13.7.1 Half-band filters	865
13.7.2 Polyphase downsampling and upsampling using half-band filters	870
13.7.3 <i>L</i> -band (Nyquist) filters	873
13.8 ★ Multirate filter banks	875
Summary	875
Problems	875
14 Spectral analysis	883
Introduction	883
14.1 Basics of spectral analysis	884
14.1.1 Spectral effects of windowing	884
14.1.2 Effect of window choice	886
14.1.3 Spectral spread and leakage	888
14.1.4 Spectral effect of sampling	892
14.2 The short-time Fourier transform (STFT)	895
14.2.1 Constant overlap-add criterion	896
14.2.2 The spectrogram	897
14.2.3 ★ Implementation of the discrete STFT in Matlab	901

14.3	★ Nonparametric methods of spectral estimation	903
14.3.1	The periodogram	903
14.3.2	Bartlett's method	908
14.3.3	The modified periodogram	912
14.3.4	Averaged modified periodogram	913
14.3.5	Welch's method	914
14.3.6	Discrete-time periodograms	915
14.3.7	Matlab implementation of the periodogram functions	915
14.4	★ Parametric methods of spectral estimation	917
14.4.1	The ARMA model	917
14.4.2	The Levinson–Durbin algorithm	921
14.4.3	Matlab implementation of the Levinson–Durbin algorithm	925
14.5	Linear prediction	928
14.5.1	Predictor error and the estimation of model order	930
14.5.2	Linear predictive coding (LPC)	932
14.5.3	The source-filter model	933
14.5.4	Linear predictive coding architecture	933
14.5.5	★ Alternate formulations of linear prediction equations	936
	Summary	936
	Problems	937
Appendix A Linear algebra 943		
A.1	Systems of linear equations	943
A.2	Solution of an inhomogeneous system of equations	944
A.2.1	Unique solution	944
A.2.2	Infinite number of solutions	947
A.2.3	No solution	948
A.3	Solution of a homogeneous system of equations	948
A.3.1	Trivial solution	948
A.3.2	Infinite number of solutions	949
A.4	Least-square-error optimization	949
Appendix B Numeric representations 955		
B.1	Integer representation	955
B.1.1	Unsigned binary	955
B.1.2	Signed-magnitude binary	956
B.1.3	Two's-complement binary	956
B.1.4	Offset binary	958
B.1.5	Converting between binary formats	959
B.2	Fixed-point (fractional) representation	959
B.2.1	Rounding and truncation	961
B.2.2	Arithmetic of fractional numbers	964
B.3	Floating-point representation	964
B.4	Computer representation of numbers	966

Appendix C Matlab tutorial	969
C.1 Introduction to Matlab	969
C.1.1 What is Matlab?	969
C.1.2 What is Matlab <i>not</i> ?	970
C.2 The elements of Matlab	970
C.2.1 Calculator functions	970
C.2.2 Variables	970
C.2.3 Matlab functions	976
C.3 Programming in Matlab	981
C.3.1 Scripts	981
C.3.2 Functions	982
C.3.3 Conditionals and loops	984
C.3.4 Classes	984
C.4 Matlab help	987
C.5 Plotting	988
C.6 The Matlab environment	989
C.6.1 Command window and editor	989
C.6.2 Debugging and writing “clean” code	990
Appendix D Probability and random processes	991
D.1 Probability distribution and density functions	991
D.1.1 Discrete probability density function	991
D.1.2 Continuous probability density function	992
D.1.3 Joint, marginal and conditional probability distributions	994
D.1.4 Expected value and moments	996
D.1.5 Covariance and correlation	997
D.2 Random processes	999
D.2.1 Statistics of a random process	1000
D.2.2 Stationary random processes	1002
D.2.3 White noise	1004
D.2.4 Filtered random processes	1005
D.2.5 The Wold decomposition	1007
D.2.6 Estimators	1008
D.2.7 Ergodic processes	1010
D.3 Power spectral density	1012
D.3.1 Definition	1012
D.3.2 Power spectral density of a filtered random process	1013
D.3.3 Power spectral density of noise	1013
D.4 Matlab functions	1013
D.4.1 Random number generators	1013
D.4.2 Autocorrelation and crosscorrelation	1014
Problems	1014
<i>References</i>	1017
<i>Index</i>	1021

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 Berra¹: “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

¹Lorenzo 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).