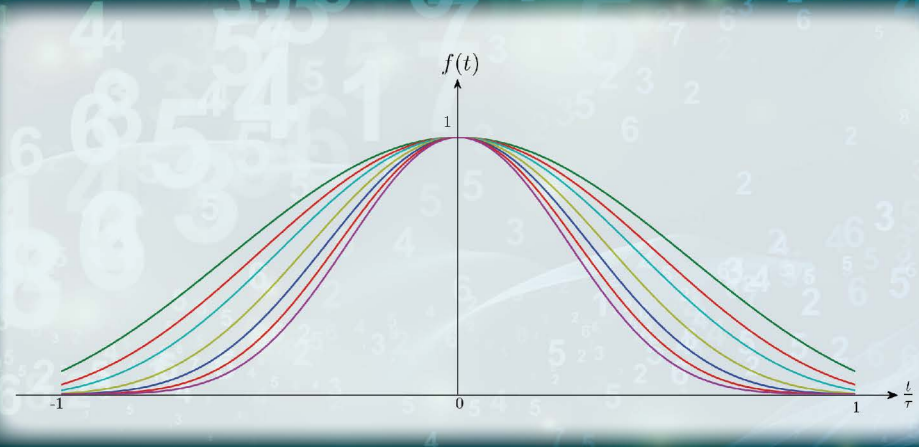


# Window Functions and Their Applications in Signal Processing



**K. M. M. PRABHU**



CRC Press  
Taylor & Francis Group



# **Window Functions and Their Applications in Signal Processing**



# Window Functions and Their Applications in Signal Processing

**K. M. M. PRABHU**



**CRC Press**  
Taylor & Francis Group  
Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2014 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper  
Version Date: 20130809

International Standard Book Number-13: 978-1-4665-1583-3 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

The Open Access version of this book, available at [www.taylorfrancis.com](http://www.taylorfrancis.com), has been made available under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 license.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

---

**Library of Congress Cataloging-in-Publication Data**

---

Prabhu, K. M. M.  
Window functions and their applications in signal processing / K. M. M. Prabhu.  
pages cm  
Includes bibliographical references and index.  
ISBN 978-1-4665-1583-3 (hbk. : alk. paper)  
1. Signal processing--Digital techniques. I. Title.

TK5102.9.P73 2014  
621.382'2--dc23

2013031807

---

Visit the Taylor & Francis Web site at  
<http://www.taylorandfrancis.com>

and the CRC Press Web site at  
<http://www.crcpress.com>

*To my parents and teachers*



---

# Contents

---

Foreword . . . . .	xiii
Preface . . . . .	xv
Acknowledgments . . . . .	xix
Abbreviations . . . . .	xxi
<b>1. Fourier Analysis Techniques for Signal Processing . . . . .</b>	<b>1</b>
1.1 Review of Basic Signals and Systems . . . . .	2
1.1.1 Basic Continuous-Time Signals . . . . .	2
1.1.2 Basic Discrete-Time Signals . . . . .	3
1.1.3 System and Its Properties . . . . .	4
1.1.4 LTI Systems . . . . .	5
1.2 Continuous-Time Fourier Transform . . . . .	6
1.2.1 Properties of the CTFT . . . . .	7
1.2.2 Examples of CTFT . . . . .	9
1.3 Discrete-Time Fourier Transform . . . . .	17
1.3.1 Properties of DTFT . . . . .	19
1.3.2 Examples of DTFT . . . . .	22
1.4 Z-Transform . . . . .	30
1.4.1 Examples of z-Transform . . . . .	34
1.5 Discrete Fourier Transform . . . . .	36
1.5.1 Properties of the DFT . . . . .	37
1.5.2 Examples of DFT . . . . .	40
1.6 Fast Fourier Transform . . . . .	50
1.6.1 Decimation-in-Time FFT (DIT-FFT) . . . . .	50
1.6.1.1 Computational Savings . . . . .	53
1.6.1.2 In-Place Computation . . . . .	54
1.6.2 Decimation-in-Frequency FFT (DIF-FFT) . . . . .	54
1.6.3 Inverse DFT from FFT . . . . .	56
1.6.4 Linear Convolution Using DIT-FFT and DIF-FFT . . . . .	57
References . . . . .	58
<b>2. Pitfalls in the Computation of DFT . . . . .</b>	<b>59</b>
2.1 Sampling, Reconstruction, and Aliasing . . . . .	59
2.1.1 WKS Sampling Theorem . . . . .	63
2.1.2 Reconstruction of Continuous-Time Signals from Discrete-Time Samples . . . . .	67
2.2 Frequency Leakage Effect . . . . .	69
2.2.1 Zero Leakage Case . . . . .	71
2.2.2 Maximum Leakage Case . . . . .	73



- 2.3 DFT as a Filter Bank . . . . . 76
- 2.4 Picket-Fence Effect or Scalloping Loss . . . . . 78
- 2.5 Zero-Padding and Frequency Resolution . . . . . 80
  - 2.5.1 Zero-Padding . . . . . 80
  - 2.5.2 Frequency Resolution . . . . . 84
- References . . . . . 85
  
- 3. Review of Window Functions . . . . . 87**
  - 3.1 Introduction . . . . . 87
  - 3.2 Characteristics of a Window Function . . . . . 87
  - 3.3 List of Windows . . . . . 88
    - 3.3.1 Rectangular (Box Car) Window . . . . . 89
    - 3.3.2 Triangular (Bartlett) Window . . . . . 89
    - 3.3.3 Cos( $x$ ) Window . . . . . 91
    - 3.3.4 Hann (Raised-Cosine) Window . . . . . 91
    - 3.3.5 Truncated Taylor Family . . . . . 94
    - 3.3.6 Hamming Window . . . . . 96
    - 3.3.7 Cos<sup>3</sup>( $x$ ) Window . . . . . 96
    - 3.3.8 Sum-Cosine Window . . . . . 97
    - 3.3.9 Cos<sup>4</sup>( $x$ ) Window . . . . . 101
    - 3.3.10 Raised-Cosine Family . . . . . 102
    - 3.3.11 Blackman Window . . . . . 106
    - 3.3.12 Optimized Blackman Window . . . . . 107
    - 3.3.13 Blackman–Harris Window . . . . . 109
    - 3.3.14 Parabolic Window . . . . . 110
    - 3.3.15 Papoulis Window . . . . . 111
    - 3.3.16 Tukey Window . . . . . 111
    - 3.3.17 Parzen (Jackson) Window . . . . . 113
    - 3.3.18 Dolph–Chebyshev Window . . . . . 114
    - 3.3.19 Kaiser’s Modified Zeroth-Order Bessel Window  
Function Family . . . . . 115
    - 3.3.20 Kaiser’s Modified First-Order Bessel Window  
Function Family . . . . . 116
  - 3.4 Rate of Fall-Off Side-Lobe Level . . . . . 118
    - 3.4.1 Theorem . . . . . 119
    - 3.4.2 Side-Lobe Fall-Off Rate in the Time-Domain . . . . . 121
  - 3.5 Comparison of Windows . . . . . 123
  - References . . . . . 126
  
- 4. Performance Comparison of Data Windows . . . . . 129**
  - 4.1 Definition of Window Parameters . . . . . 130
  - 4.2 Computation of Window Parameters . . . . . 132
  - 4.3 Discussion on Window Selection . . . . . 139
  - References . . . . . 142

<b>5. Discrete-Time Windows and Their Figures of Merit</b>	145
5.1 Different Classes of Windows	145
5.2 Discrete-Time Windows	148
5.2.1 Rectangular (Box Car) Window	148
5.2.2 Triangular (Bartlett) Window	148
5.2.3 $\text{Cos}^\alpha x$ Window Family	148
5.2.4 Hann Window	148
5.2.5 Truncated Taylor Family of Windows	149
5.2.6 Hamming Window	149
5.2.7 Sum-Cosine Window	149
5.2.8 Raised-Cosine Window Family	149
5.2.9 Blackman Window	150
5.2.10 Optimized Blackman Window	150
5.2.11 Tukey Window	150
5.2.12 Blackman–Harris Window	151
5.2.13 Nuttall Window Family	151
5.2.14 Flat-Top Window	152
5.2.15 Parabolic Window	153
5.2.16 Riemann Window	153
5.2.17 Poisson Window	153
5.2.18 Gaussian Window	153
5.2.19 Cauchy Window	154
5.2.20 Hann–Poisson Window	154
5.2.21 Papoulis (Bohman) Window	154
5.2.22 Jackson (Parzen) Window	155
5.2.23 Dolph–Chebyshev Window	155
5.2.24 Modified Zeroth-Order Kaiser–Bessel Window Family	155
5.2.25 Modified First-Order Kaiser–Bessel Window Family	156
5.2.26 Saramäki Window Family	156
5.2.27 Ultraspherical Window	157
5.2.28 Odd and Even-Length Windows	158
5.3 Figures of Merit	159
5.4 Time–Bandwidth Product	171
5.5 Applications of Windows	172
5.5.1 FIR Filter Design Using Windows	172
5.5.2 Spectral Analysis	177
5.5.3 Window Selection for Spectral Analysis	181
References	188
<b>6. Time-Domain and Frequency-Domain Implementations of Windows</b>	189
6.1 Time-Domain Implementation	189
6.2 A Programmable Windowing Technique	190

- 6.3 Computational Error in Time and Frequency-Domains . . . . . 192
- 6.4 Canonic Signed Digit Windowing . . . . . 195
  - 6.4.1 Window 1 . . . . . 197
  - 6.4.2 Window 2 . . . . . 197
  - 6.4.3 Window 3 . . . . . 198
  - 6.4.4 Window 4 . . . . . 202
  - 6.4.5 Window 5 . . . . . 202
  - 6.4.6 Window 6 . . . . . 202
  - 6.4.7 Window 7 . . . . . 203
  - 6.4.8 Window 8 . . . . . 205
  - 6.4.9 Window 9 . . . . . 207
  - 6.4.10 Window 10 . . . . . 209
  - 6.4.11 Window 11 . . . . . 209
  - 6.4.12 Window 12 . . . . . 209
  - 6.4.13 Window 13 . . . . . 210
  - 6.4.14 Window 14 . . . . . 211
- 6.5 Modified Zeroth-Order Kaiser–Bessel Window Family . . . . . 213
- 6.6 Summary . . . . . 216
- References . . . . . 217
  
- 7. FIR Filter Design Using Windows . . . . . 219**
  - 7.1 Ideal Filters . . . . . 219
    - 7.1.1 Lowpass Filter . . . . . 219
    - 7.1.2 Highpass Filter . . . . . 220
    - 7.1.3 Bandpass Filter . . . . . 221
    - 7.1.4 Bandstop Filter . . . . . 221
  - 7.2 Linear Time Invariant Systems . . . . . 222
  - 7.3 FIR Filters . . . . . 224
    - 7.3.1 Advantages of FIR Filters . . . . . 225
  - 7.4 IIR Filters . . . . . 226
    - 7.4.1 Properties of IIR Filters . . . . . 226
  - 7.5 Structure of an FIR Filter . . . . . 226
    - 7.5.1 Filter Specifications . . . . . 227
  - 7.6 FIR Filter Design . . . . . 228
    - 7.6.1 Linear-Phase Filters . . . . . 229
    - 7.6.2 Types of FIR Filters . . . . . 234
    - 7.6.3 Frequency Response of Type 1 FIR Filter . . . . . 236
    - 7.6.4 Design Procedure for Filters . . . . . 239
  - 7.7 Kaiser–Bessel Windows for FIR Filter Design . . . . . 243
    - 7.7.1 Filter Design Using Kaiser–Bessel Zeroth-Order ( $I_0$ -Sinh) Window . . . . . 243
    - 7.7.2 Filter Design Using Kaiser–Bessel First-Order ( $I_1$ -Cosh) Window . . . . . 249
  - 7.8 Design of Differentiator by Impulse Response Truncation . . . . . 252

7.9	Design of Hilbert Transformer Using Impulse Response Truncation . . . . .	253
	References . . . . .	256
<b>8.</b>	<b>Application of Windows in Spectral Analysis . . . . .</b>	<b>259</b>
8.1	Nonparametric Methods . . . . .	260
8.1.1	Periodogram PSD Estimator . . . . .	260
8.1.2	Modified Periodogram PSD Estimator . . . . .	265
8.1.3	Spectral Analysis Using Kaiser–Bessel Window . . . . .	271
8.1.4	Bartlett Periodogram . . . . .	278
8.1.5	Welch Periodogram Method . . . . .	280
8.1.6	Blackman–Tukey Method . . . . .	283
8.1.7	Daniel Periodogram . . . . .	292
8.1.8	Application of the FFT to the Computation of a Periodogram . . . . .	293
8.1.9	Short-Time Fourier Transform . . . . .	293
8.1.10	Conclusions . . . . .	297
	References . . . . .	301
<b>9.</b>	<b>Applications of Windows . . . . .</b>	<b>303</b>
9.1	Windows in High Range Resolution Radars . . . . .	303
9.1.1	HRR Target Profiling . . . . .	303
9.1.2	Simulation Results . . . . .	305
9.2	Effect of Range Side Lobe Reduction on SNR . . . . .	306
9.2.1	Introduction . . . . .	307
9.2.2	Loss Factor . . . . .	309
9.2.3	Weighting Function . . . . .	314
9.2.4	Results and Discussions . . . . .	316
9.3	Window Functions in Stretch Processing . . . . .	317
9.4	Application of Window Functions in Biomedical Signal Processing . . . . .	322
9.4.1	Biomedical Signal Processing . . . . .	323
9.4.2	FIR Filtering of Biomedical Signals . . . . .	324
9.4.3	Moving Average Filtering of Biomedical Signals . . . . .	328
9.4.4	QRS Detection in ECG Based on STFT . . . . .	333
9.5	Audio Denoising Using the Time–Frequency Plane . . . . .	336
9.5.1	Time–Frequency Plane . . . . .	336
9.5.2	Audio Denoising Using Time–Frequency Plane . . . . .	340
9.5.3	Block Thresholding . . . . .	341
9.5.4	Effect of Windows . . . . .	342
9.6	Effect of Windows on Linear Prediction of Speech . . . . .	343
9.6.1	Linear Prediction Coder . . . . .	344
9.6.2	Line Spectral Frequencies . . . . .	346
9.6.3	LSF Variation due to Windows . . . . .	347
9.7	Application of Windows in Image Processing . . . . .	349

- 9.7.1 Windows for ISAR Images . . . . . 350
- 9.7.2 Experimental Analysis . . . . . 352
- 9.7.3 Results and Conclusions . . . . . 356
- 9.8 Windows to Improve Contrast Ratio in Imaging Systems . . . . . 357
  - 9.8.1 Experimental Analysis . . . . . 361
  - 9.8.2 Results and Conclusions . . . . . 361
- References . . . . . 361
  
- Index . . . . . 365**

---

## *Foreword*

---

In the past decades, rapid advances in digital IC (integrated circuit) technology have caused a “digital revolution” in the field of signal processing. These days, almost any real-world signal is represented and processed digitally, from physiological vital signs via camera pictures, audio signals, video signals, and radar signals to the massive four-dimensional datasets produced by modern medical imaging equipment. The rapid growth of the fixed and mobile Internet, combined with the insatiable appetite of mankind for information, will further add fuel to this revolution.

When real-world signals are converted into a digital form, they are commonly split into discrete blocks for further processing. To avoid the edge effects across the blocks, the blocks are often weighted by a window function that tapers the signal off toward both ends of the block. A window function is a mathematical function that is zero-valued outside some chosen interval. When a signal is multiplied by a window function, the product is also zero-valued outside this interval. Effectively, we are viewing the signal through a “window,” hence the name of the function.

Window functions are explicitly or implicitly used in many, if not most, digital signal processing systems, and as such are genuinely important. Even so, the vast signal processing literature contains a few, if any, of books or monographs that are dedicated to this topic. This monograph is a welcome exception. To the best of my knowledge, it provides the most comprehensive treatment of window functions and their applications available to date. The author, Professor dr. ir. K.M.M. Prabhu, has been affiliated since the mid-1970s with the prestigious Indian Institute of Technology Madras, Chennai, India. He has made significant contributions to the development of window functions and their implementation intricacies in the mid-1970s and early 1980s and has maintained an active interest in window functions ever since. Hence, he is very well placed to provide an authoritative treatment on the topic.

Window functions have a strong impact on the spectrum of the signal and essentially permit a trade-off between time and frequency resolution. Accordingly, the monograph starts with a review of continuous and discrete-time Fourier analysis techniques and of key artifacts such as spectral aliasing and leakage. The core of the monograph consists of a survey and a detailed feature analysis of an extensive set of continuous and discrete-time window functions. This is supplemented by a treatment of efficient time- and frequency-domain window implementation approaches. The final chapters zoom in on the key applications of window functions, such as digital filter design, spectral analysis, and applications in fields such as radar

signal processing, biomedical engineering, and audio, speech, and image processing.

I would like to congratulate the author on this valuable addition to the signal processing literature.

**Professor dr. ir. J.W.M. Bergmans**

*Chairman, Signal Processing Systems Group*

*Eindhoven University of Technology*

*The Netherlands*

---

# *Preface*

---

This monograph presents an exhaustive and detailed account of window functions and their applications in signal processing. Window functions, otherwise known as weighting functions, tapering functions or apodization functions, are mathematical functions that are zero-valued outside the chosen interval. As a popular quote goes, there are as many numbers of windows as the number of people working in signal processing.

Chapter 1 deals with the Fourier analysis techniques. First, the basic signals and systems in the continuous time-domain are introduced, followed by the continuous-time Fourier transform (CTFT). Its properties and some examples are discussed next. We then move on to the discrete-time Fourier transform (DTFT), the first transform encountered in digital signal processing, to convert a discrete-time signal into its frequency-domain counterpart. The Fourier transform to handle sequences of finite length, called the discrete Fourier transform (DFT), is discussed next and its properties and applications are highlighted. Finally, the algorithms to compute the DFT faster, namely, the fast Fourier transform (FFT) based on decimation-in-time (DIT) and decimation-in-frequency (DIF), are described. This chapter concludes with an efficient technique to compute linear convolution via circular convolution using the DIT and DIF algorithms.

In Chapter 2, we discuss the pitfalls in the computation of the DFT. There are two processes involved while computing the DFT of an analog (continuous-time) signal: sampling and truncation. While sampling introduces a distortion called aliasing, the truncation operation due to the finite length data introduces two other effects known as the frequency leakage and picket-fence effect. In this chapter, these effects and the manner in which they can be eliminated/reduced are also detailed. The DFT functioning as a bank of band-pass filters is also demonstrated.

Chapter 3 introduces the commonly used window functions in the continuous-time-domain, rather than in the discrete-time-domain. The characteristics which qualify a function to be called as a window function are given next. The plots of the window functions are provided in the time-domain as well as in the frequency-domain. The two near-optimum Kaiser–Bessel window function families are also discussed in detail. The main characteristics of a window function such as normalized half-main-lobe width (NHMLW), first side-lobe level (FSLL), maximum side-lobe level (MSLL), ratio of main-lobe energy to the total energy (MLE), and rate of fall-off of side-lobe levels (RFSLL) are also enlisted for all the windows considered here. This chapter concludes with a rigorous comparison of all the window parameters.



Chapter 4, titled as the performance comparison of data windows, defines a number of parameters, once again in the continuous-time-domain. These parameters are computed either analytically or numerically for all the windows which were introduced in Chapter 3. Finally, directions for the choice of an appropriate window function for specific applications are provided.

Discrete-time windows and their figures of merit are discussed in Chapter 5. The four different classifications of windows are presented. The discrete-time versions of all the windows from Chapter 3, as well as some more popular windows discussed in the literature, are reviewed here. Definitions of window parameters and a discussion on the window selection process are also outlined. Finally, the chapter concludes with the two important applications of windows; namely, finite impulse response (FIR) digital filter design and spectral analysis.

The two implementation strategies of window functions in the time-domain and frequency-domain are dealt with in Chapter 6. A novel scheme to implement certain types of windows in the frequency-domain is derived and a structure called binary windowing structure to implement it is also presented. The computational error performance in terms of signal-to-computational error ratio (SCER) in both the domains is tabulated and their performances are compared. Finally, novel binary windowing structures called canonic signed digit (CSD) windowing are presented for all the binary windows considered in this chapter.

FIR filter design using windows is considered in Chapter 7. This chapter deals with the different types of ideal filters: lowpass, highpass, bandpass, and bandstop filters. A discussion on linear phase filters, followed by the four types of filters is presented next. A clear design procedure is given for FIR filters. Furthermore, FIR filter design using zeroth-order and first-order Kaiser–Bessel windows is presented. These use closed-form expressions in determining the filter order as well as the window shape parameter. The design of differentiators and Hilbert transformers are also outlined.

Window functions are vital in nonparametric methods of spectral estimation as well. They are classified as: periodogram PSD estimators, modified periodogram PSD estimators, and correlogram estimators. These methods and the requirement for window functions are discussed in detail in Chapter 8. This chapter also gives the application of Kaiser–Bessel window in spectral analysis. Closed-form expressions are available to compute the window length and the variable parameter  $\alpha$  of the Kaiser–Bessel window. Besides, we introduce short-time Fourier transform (STFT), which is also known as time-dependent Fourier transform, in analyzing nonstationary signals, such as speech. Several examples are discussed which clearly brings out the power of window functions in nonparametric spectral analysis.

Chapter 9 discusses well-known applications of window functions in the fields of radar, sonar, biomedical signal analysis, audio processing, and synthetic aperture radar. In the context of radar, the cases considered are high-range resolution radars, the effect of range side-lobe reduction on SNR and in

stretch processing. In biomedical signal processing, we consider FIR/moving average filtering of biomedical signals, QRS detection of ECG signals using STFT, and so on. Audio de-noising using time–frequency plane, effect of windows on linear prediction of speech, and so on are dealt with in the audio-processing section. Finally the chapter concludes with topics such as the effect of windows in ISAR (inverse synthetic aperture radar) images and the usage of windows in improving the contrast ratio in imaging systems.



---

## Acknowledgments

---

When Dr. Gagandeep Singh of Taylor & Francis Group/CRC Press asked me to write a book for them, I agreed to consider his request. Since my time was limited, I said that I could write a monograph on *Window Functions and Their Applications in Signal Processing*, instead of a textbook, on the condition that he should get the opinion on this topic from a few experts in the signal processing area. I provided two sample chapters and the proposed table of contents of the monograph. These details were sent to six professors for their feedback with a detailed questionnaire. To my surprise, all the feedback (except one), were quite positive and this enthused me to write this monograph.

I profusely thank Professor Fred Harris, San Diego State University, USA, Professor Jonathon Chambers, Loughborough University, UK, Professor Palaniappan Ramaswamy, University of Essex, UK, Professor Wai Lok Woo, Newcastle University, UK, and an anonymous reviewer, for giving me encouragement and suggestions/criticisms on the proposed monograph.

I thank the authorities of the Indian Institute of Technology (IIT) Madras, India, for granting me one semester sabbatical during July–November 2011, as well as providing me with all the necessary facilities in bringing this monograph to its present form. I profusely thank Prof. J.W.M. Bergmans, Chairman of the Signal Processing Systems group, Technical University Eindhoven (TU/e), Netherlands, for providing me travel grant to visit TU/e at different times.

I wish to extend my thanks to dr. ir. A.W.M. van den Enden, who was earlier with Philips Research and Philips Semiconductors, Eindhoven, The Netherlands (as a lead signal processing expert). He is now an independent consultant where he works closely for Philips Research, NXP, Catena Radio Design, SRON, and others. He tirelessly and meticulously read through all the chapters and we had detailed discussions on each chapter when I visited him at TU/e, Eindhoven last year. He also introduced me to the topic of canonic signed digit (CSD) representation. I express my whole-hearted appreciation to him for spending his valuable time overlooking his own personal work.

I wish to thank my research students who were an integral part of this project. They are Basil Abraham, S. Abhijith Varma, H.N. Bharath, Sunit Sivasankaran, P. Vikram Kumar, Sanjay Kumar Sindhi, and T. Srinivas Rao, all from the Department of Electrical Engineering, IIT Madras, India. I also thank my summer intern, Swathi Nagaraj from M.I.T., Manipal, India. All of them diligently worked toward achieving the goal to my satisfaction. I profusely thank all of them for their dedication toward perfection. I also thank my long-time friend, philosopher and guide, Prof. Umapathi Reddy, presently

Honorary Professor, IIT Hyderabad, and Fellow, Imagination Technologies, Hyderabad, India. I always cherish my long association with him.

I wish to thank my daughter Sahana and my wife Kanthi for proofreading and editing the entire manuscript.

The effort and hardwork put in by my parents in educating their children is still vivid in my memory. I have emulated the meticulous approach which my father took while solving every life-related problem. He has been a source of inspiration to me throughout my life.

---

## *Abbreviations*

---

ACF	Autocorrelation function
ADC	Analog-to-digital converter
AP	Action potential
ATP	Average transmitted power
AWGN	Additive white Gaussian noise
BIBO	Bounded-input bounded-output
BPF	Band-pass filter
BSF	Band-stop filter
BT	Blackman–Tukey
BW	Bandwidth
CR	Contrast ratio
CSD	Canonic signed digit
CTFT	Continuous-time Fourier transform
DFT	Discrete Fourier transform
DIF	Decimation-in-frequency
DIT	Decimation-in-time
DSP	Digital signal processing/processor
DTFT	Discrete-time Fourier transform
ECG	Electrocardiogram
EEG	Electroencephalogram
EGG	Electrogastrogram
EMG	Electromyogram
ENBW	Equivalent noise bandwidth
ENG	Electroneurogram
EOG	Electro-oculogram
ERG	Electroretinogram
ERP	Event-related potential
FDI	Frequency-domain implementation
FFT	Fast Fourier transform
FIR	Finite impulse response
FM	Frequency modulation
FSL	First side-lobe level
FT	Fourier transform
HPF	High-pass filter
HRR	High-range resolution
IDFT	Inverse discrete Fourier transform
IDTFT	Inverse discrete-time Fourier transform
IFFT	Inverse fast Fourier transform

IIR	Infinite impulse response
IRT	Impulse response truncation
ISAR	Inverse synthetic aperture radar
LCCDE	Linear constant coefficient difference or differential equation
LF	Loss factor
LFM	Linear frequency modulation
LPC	Linear predictive coding
LPF	Low-pass filter
LSF	Line spectral frequencies
LSI	Linear shift invariant
LTI	Linear time invariant
MA	Moving average
MLE	Main lobe energy
MSE	Mean square error
MSLL	Maximum side-lobe level
MSR	Main-lobe to side-lobe ratio
NHMLW	Normalized half-main-lobe width
PC	Pulse compression
PCG	Phonocardiogram
PM	Phase modulation
PRI	Pulse repetition interval
PSD	Power spectral density
PSF	Point spread function
PSLL	Peak side-lobe level
RCS	Radar cross-section
RFSLL	Rate of fall-off side-lobe level
RMS	Root mean square
ROC	Region of convergence
SAR	Synthetic aperture radar
SCER	Signal-to-computational ratio
SFW	Stepped frequency waveform
SLE	Side-lobe energy
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
SURE	Stein unbiased risk estimate
TDI	Time-domain implementation
TF	Time–frequency
VAG	Vibroarthogram
VMG	Vibromyogram
WCPL	Worst-case processing loss

# 1

---

## *Fourier Analysis Techniques for Signal Processing*

---

There are several methods to transform a time-domain signal into frequency-domain. The motivation for transforming a signal from one domain to another is that the characteristics of a signal are visible directly and can be easily extracted from such a representation. For instance, from a signal represented in time-domain, we can only extract some features such as the exact starting time instant of the signal or the duration for which the signal existed. However, other useful features such as bandwidth and frequency occupied by the signal are not directly visible unless we convert it into the frequency-domain. Another advantage of transforming a signal into the Fourier domain is that the convolution operation gets simplified to multiplication. We can use Fourier analysis techniques to identify and separate the frequency bands of interest from noisy observations. Transforms are vital in many speech applications (recognition, synthesis, and coding), radio communications, vibration analysis, and so on. Some specific areas where Fourier transforms (FTs) are applied include steady-state and resonance analysis of signals, modulation, filter design, sampling rate selection, stability analysis, correlations by block processing, and pitch period estimation.

In this chapter, we begin by describing the continuous-time Fourier transform (CTFT) technique for continuous-time (CT) signals and then proceed to the discrete-time Fourier transform (DTFT) for discrete-time (DT) sequences. The DTFT has been developed from the CTFT by utilizing the similarities between analyzing continuous- and discrete-time signals. The concept of  $z$ -transform is introduced next, which is useful in analyzing and synthesizing discrete-time signals and systems. However, for present-day applications, the DTFT is not amenable to digital computations, since in the forward DTFT, we require infinitely many number of computations; while in the inverse DTFT (IDTFT), we have an integral notation to deal with.

Therefore, we resort to the discrete Fourier transform (DFT) which is a uniformly sampled version of the DTFT. In the case of DFT, both the forward and inverse DFT (IDFT) expressions are discrete as well as finite. Therefore the DFT and IDFT remove the restrictions associated with the DTFT and IDTFT, respectively. Finally, we discuss the fast Fourier transform (FFT), which is a computationally efficient tool to compute the DFT of a signal with a reduced number of arithmetic operations. The FFT is commonly used in all digital signal processors (DSPs) and general purpose digital computers.



## 1.1 Review of Basic Signals and Systems

In this section, we start our study of signals which are commonly encountered in signal processing. We also enlist the important properties of systems.

### 1.1.1 Basic Continuous-Time Signals

1. *Unit step signal*: This signal is defined for a time instant ‘ $t$ ’ as follows:

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0. \end{cases} \quad (1.1)$$

This signal is shown in Figure 1.1.

2. *Unit impulse function*: It is not appropriate to give a duration for the impulse function; instead we can say that the area under the unit impulse is unity. It can be graphically represented as shown in Figure 1.2. It can be assumed as a limiting case of the delta function:

$$\delta(t) = \lim_{\Delta \rightarrow 0} \delta_{\Delta}(t). \quad (1.2)$$

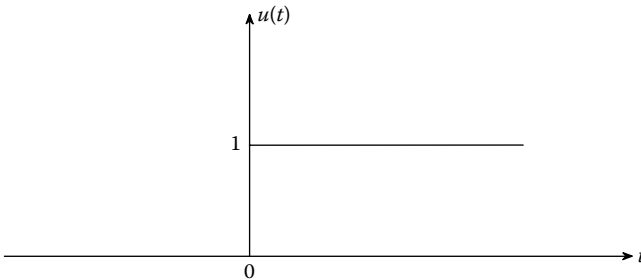
We note that the unit impulse function can be related to the unit step signal as:

$$\delta(t) = \frac{du(t)}{dt}. \quad (1.3)$$

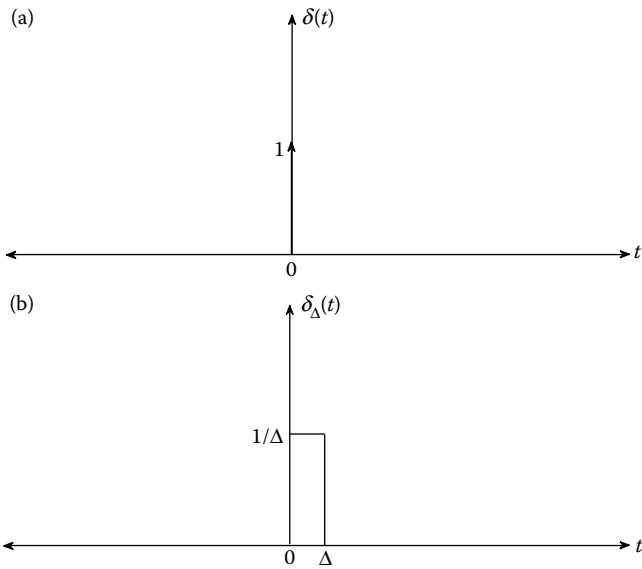
3. *Complex exponential signal*: This is represented by the following function:

$$x(t) = ce^{at}, \quad (1.4)$$

where  $c$  and  $a$  can represent complex numbers, in general.



**FIGURE 1.1**  
Unit step signal.

**FIGURE 1.2**

(a) Unit impulse signal. (b) Unit delta function.

4. *Sinusoidal signal*: The sine function with amplitude  $A$  and frequency of oscillation  $\Omega_0$  is given by:

$$x(t) = A \sin(\Omega_0 t + \phi), \quad (1.5)$$

where  $\phi$  represents the phase of the sinusoidal signal.

### 1.1.2 Basic Discrete-Time Signals

These signals are similar to their CT counterparts, but defined with respect to an integer-valued variable 'n'.

1. *Unit impulse sequence*:

$$\delta[n] = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases} \quad (1.6)$$

2. *Unit step sequence*:

$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases} \quad (1.7)$$

3. *Complex exponential:*

$$x[n] = c\alpha^n \quad (1.8)$$

where  $c$  and  $\alpha$  are, in general, complex numbers.

4. *Sinusoidal signal:*

$$x[n] = A \cos[\omega_0 n + \phi] \quad (1.9)$$

where  $\omega_0$  is the frequency,  $\phi$  represents the phase and  $A$  is the amplitude of the sinusoidal signal. The difference between the CT sinusoidal signal and its DT domain is that the equivalent frequency in the DT domain lies in the range  $[-\pi, \pi]$ , whereas in the CT domain, it varies in the broad range  $[-\infty, \infty]$ .

### 1.1.3 System and Its Properties

A discrete-time system can be defined as a transformer from the input space to a transformed space. It can schematically be described as shown in Figure 1.3 and it is mathematically described by the following relation:

$$y[n] = T\{x[n]\}. \quad (1.10)$$

Some useful properties of general systems are described below.

1. *Linearity property:* If we consider two output sequences  $y_1[n]$  and  $y_2[n]$  which are defined as the transformations of  $x_1[n]$  and  $x_2[n]$ :

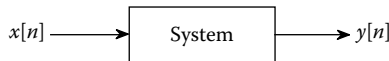
$$x_1[n] \xleftrightarrow{T} y_1[n] = T(x_1[n]) \quad (1.11)$$

$$x_2[n] \xleftrightarrow{T} y_2[n] = T(x_2[n]) \quad (1.12)$$

Then, the system is said to be linear if

$$ax_1[n] + bx_2[n] \xleftrightarrow{T} y_3[n] = T(ax_1[n] + bx_2[n]) \quad (1.13)$$

and if  $y_3[n] = ay_1[n] + by_2[n]$ . Here,  $a$  and  $b$  are any arbitrary constants.



**FIGURE 1.3**  
System.

2. *Time invariance property*: If the transformation of a signal does not vary with time, then the system is said to be time invariant. Let us consider a signal  $x[n]$  and its delayed version  $x[n - n_0]$  as

$$x[n] \xleftrightarrow{T} y[n] \quad (1.14)$$

$$x[n - n_0] \xleftrightarrow{T} y_1[n]. \quad (1.15)$$

If  $y_1[n] = y[n - n_0]$ , then the discrete-time system is said to be time invariant.

3. *Causality property*: If the output of a system at any time instant depends only on the present and past values of the input (and past values of the output), then the system is said to be causal. Hence, the output at any time does not depend on future values of input and output.
4. *Stability property*: We consider bounded-input bounded-output (BIBO) stability. When a bounded-input is applied to the system and if the output is bounded, then the system is said to be BIBO stable.

$$|x[n]| \leq B_x \quad \forall n, \quad 0 \leq B_x < \infty \quad (1.16)$$

and

$$|y[n]| \leq B_y \quad \forall n, \quad 0 \leq B_y < \infty. \quad (1.17)$$

It should be noted that only a causal and stable system is physically realizable.

### 1.1.4 LTI Systems

An important class of systems which obeys the linearity and time-invariance properties is called the linear time-invariant (LTI) system. An LTI system is uniquely represented by its impulse response  $h(t)$  or  $h[n]$  (in CT domain and in DT domain), which is the output of the system to an unit impulse signal. For an LTI system, the output of the system can be uniquely expressed as a convolution of the input with the impulse response of the system. The continuous-time convolution integral is defined as

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau) d\tau \quad (1.18)$$

and the discrete-time convolution sum is defined as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k]. \quad (1.19)$$

The above-defined convolution operations are called linear convolution operations and, in general, are represented in short-hand notations as

$$y(t) = x(t) * h(t) \quad (1.20)$$

$$y[n] = x[n] * h[n]. \quad (1.21)$$

This section has provided a brief overview of the signals and systems needed to understand the following chapters. In case the reader is interested in more details, refer to Refs. [1–4].

## 1.2 Continuous-Time Fourier Transform

We can represent the output of an LTI system to an input signal in terms of the shifted orthogonal basis signals for the ease of mathematical calculations as well as for visualization. The continuous-time periodic signal can be simply characterized as a sum of harmonically related sine and cosine waveforms. This is popularly known as the Fourier series expansion of the signal, which involves the decomposition of the periodic signals into their frequency components. To analytically represent an aperiodic signal in a similar manner, we need to make an assumption that an aperiodic signal is actually a periodic signal with infinite period. This type of time-domain to frequency-domain transformation is called the continuous-time Fourier transform (CTFT). There are few restrictions, known as the Dirichlet conditions, which a given signal should satisfy to be represented in the Fourier domain. These are stated as follows: it is sufficient that the signal be absolutely integrable (i.e., bounded signal) and it should have a finite number of maxima, minima and discontinuities in finite time. This encompasses a wide variety of signals which can be decomposed into a superposition integral of exponentials of infinite duration. Complex exponential functions,  $e^{jk\Omega_0 t}$ , are common periodic signals that can be used as the orthogonal basis functions. The complex exponential formula for continuous-time Fourier series of a periodic signal,  $\tilde{x}(t)$  (with period  $T$ ) is given by

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\Omega_0 t}, \quad \Omega_0 = \frac{2\pi}{T} \quad (1.22)$$

$$a_k = \frac{1}{T} \int_{-T/2}^{T/2} \tilde{x}(t) e^{-jk\Omega_0 t} dt, \quad (1.23)$$

where  $a_k$  denotes the Fourier series coefficients, and  $\Omega_0$  is the fundamental angular frequency in rad/s. This transform pair is quite a significant tool, as it states that even arbitrary discontinuous signals can be expressed in terms of simple smooth basis functions. Discontinuous parts of the signal are represented by the higher-order harmonics in the Fourier series.

This can analogously be extended to aperiodic signals by assuming that the period  $T$  is infinity. This generalization enables frequency-domain conversion of a much wider class of signals of interest. Increasing the  $T$  results in two effects: (i) the magnitude of the spectrum decreases by an order of  $1/T$ , and (ii) the spacing between the line spectra decreases with respect to  $\Omega_0$ . Let  $X(j\Omega)$  denote the CTFT of  $x(t)$ .

When  $T \rightarrow \infty$ ,  $X(j\Omega)$  approaches a continuous magnitude spectrum (rather than a line spectrum in the case of periodic signals). Note that  $X(j\Omega)$  is actually the envelope of  $Ta_k$ , which is defined as

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \quad (1.24)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega. \quad (1.25)$$

Equations (1.24) and (1.25) together are known as the continuous-time Fourier transform (CTFT) pair.

We can construct the FT of a periodic signal directly from its Fourier series representation. The resulting transform, given below, consists of a train of impulses in the frequency-domain.

$$X(j\Omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\Omega - k\Omega_0). \quad (1.26)$$

### 1.2.1 Properties of the CTFT

We now enlist some of the useful properties of the CTFT, which can simplify the solution of many problems. Let  $x(t)$  and  $y(t)$  be the time-domain signals and their CTFTs be  $X(j\Omega)$  and  $Y(j\Omega)$ , respectively. The FT pair is expressed as  $x(t) \xleftrightarrow{\mathcal{F}} X(j\Omega)$  and  $y(t) \xleftrightarrow{\mathcal{F}} Y(j\Omega)$ .

1. *Linearity property:* This results directly from the linearity property of integration.

$$ax(t) + by(t) \xleftrightarrow{\mathcal{F}} aX(j\Omega) + bY(j\Omega). \quad (1.27)$$

Here,  $a$  and  $b$  represent arbitrary constants. This simply means that the FT of a linear combination of two arbitrary signals is the same as the linear combinations of the transforms of individual components. It can be easily extended to linear sum of any arbitrary number of signals.

2. *Time and frequency shifting property:* The shift in time-domain by a duration  $t_0$  is given by

$$x(t - t_0) \xleftrightarrow{\mathcal{F}} e^{-j\Omega t_0} X(j\Omega). \quad (1.28)$$

It shows that shifting the time-domain signal by an amount  $t_0$  results in a phase shift in the Fourier domain, while the magnitude response remains unchanged. We also note that the higher the frequency, the greater the phase shift it experiences. This is obvious from the fact that, in the same span of time, a higher frequency signal covers more number of cycles. Similarly, we can state the corresponding dual property of shifting in the frequency-domain as

$$x(t)e^{j\Omega_0 t} \xleftrightarrow{\mathcal{F}} X(j(\Omega - \Omega_0)). \quad (1.29)$$

3. *Conjugation and conjugate symmetry property:* For a complex signal  $x(t)$ :

$$x^*(t) \xleftrightarrow{\mathcal{F}} X^*(-j\Omega). \quad (1.30)$$

This property directly follows from the evaluation of the complex conjugate of Equation (1.24).

4. *Differentiation and integration property:* The CTFT of  $dx(t)/dt$  can be found by differentiating Equation 1.25 with respect to  $t$ . After differentiation, we find that the higher frequency components of the signal become more pronounced. This property can be stated as follows:

$$\frac{dx(t)}{dt} \xleftrightarrow{\mathcal{F}} j\Omega X(j\Omega). \quad (1.31)$$

In a similar manner, integrating the time-domain signal results in the following:

$$\int_{-\infty}^t x(t) dt \xleftrightarrow{\mathcal{F}} \frac{1}{j\Omega} X(j\Omega) + \pi X(0)\delta(\Omega). \quad (1.32)$$

Integration attenuates the magnitude of the signal at higher frequencies and thus acts like a low-pass filter. If  $X(0)$  is nonzero, the signal contains a DC component, which introduces an impulse in the frequency-domain.

5. *Time and frequency scaling property:* For a scaling factor 'a', this property can be given as follows:

$$x(at) \xleftrightarrow{\mathcal{F}} \frac{1}{|a|} X\left(\frac{j\Omega}{a}\right). \quad (1.33)$$

For instance, if  $a = -1$  then

$$x(-t) \xleftrightarrow{\mathcal{F}} X(-j\Omega). \quad (1.34)$$

If the scaling factor has a magnitude greater than unity, then the signal is compressed in the time-domain, while its frequency spectrum gets expanded. For  $|a| < 1$ , exactly the converse happens, that is, the time-domain signal is expanded and the spectrum is scaled down.

6. *Duality property*: This property reveals the effect when we interchange the roles of  $t$  and  $\Omega$ . This helps in finding the CTFTs of some signals directly from a table of transforms. It simply states that every property of CTFT has a dual function, given as follows:

$$X(t) \xleftrightarrow{\mathcal{F}} 2\pi x(-j\Omega). \quad (1.35)$$

7. *Parseval's theorem*: This gives the relation between the energy ( $E$ ) of a signal in the time-domain and the frequency-domain. We can use this property to easily compute the energy of a signal by integrating the squared magnitude of its FT.

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(j\Omega)|^2 d\Omega. \quad (1.36)$$

8. *Convolution property*: Convolution in the time-domain is equivalent to multiplication in the frequency-domain and vice versa.

$$x(t) * h(t) \xleftrightarrow{\mathcal{F}} X(j\Omega)H(j\Omega). \quad (1.37)$$

This property is vital especially in the analysis of linear time-invariant (LTI) systems.

9. *Modulation property*: This is the dual of the convolution property stated above and can be given as follows:

$$x(t)y(t) \xleftrightarrow{\mathcal{F}} \frac{1}{2\pi} [X(j\Omega) * Y(j\Omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\theta)Y(j(\Omega - \theta)) d\theta.$$

Here, we must recall that multiplication of one signal by another amounts to modulation. The modulation property is extensively used in communications.

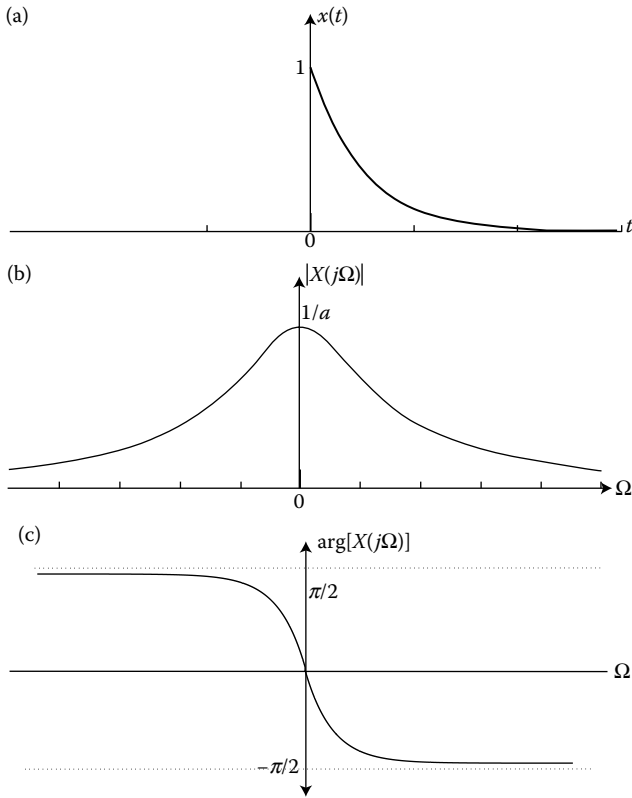
### 1.2.2 Examples of CTFT

1. Find the CTFT of a complex one-sided exponential signal given by:  $x(t) = e^{-at}u(t)$ ,  $a > 0$ .

The CTFT of this signal can be determined only if  $a > 0$ , since if  $a < 0$ , the signal fails to be absolutely integrable. Using the CTFT Equation (1.24), we obtain

$$\begin{aligned} X(j\Omega) &= \int_{-\infty}^{\infty} [e^{-at}u(t)]e^{-j\Omega t} dt = \int_0^{\infty} e^{-at}e^{-j\Omega t} dt \\ &= \int_0^{\infty} e^{-(a+j\Omega)t} dt = \frac{1}{a+j\Omega}, \quad a > 0. \end{aligned}$$





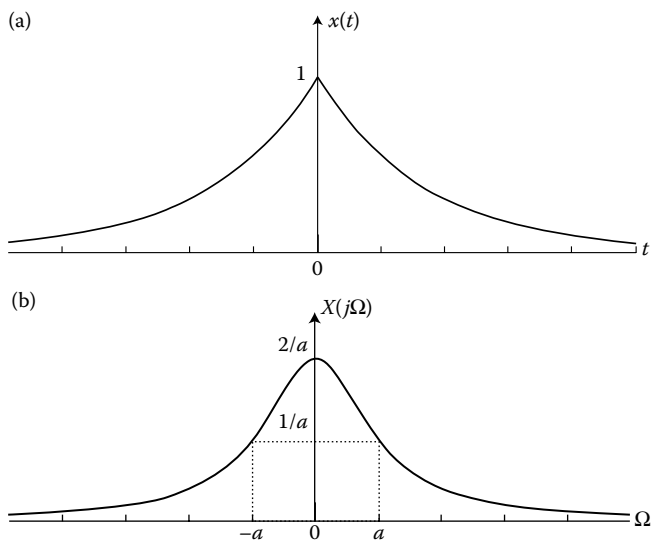
**FIGURE 1.4**

FT pair of  $e^{-at}u(t)$ . (a) Time-domain signal. (b) Frequency-domain magnitude plot. (c) Frequency-domain phase plot.

Figure 1.4 shows the signal  $x(t)$ , the magnitude, and phase responses of the CTFT of  $e^{-at}u(t)$ .

- Determine the CTFT of a two-sided exponential signal given by:  
 $x(t) = e^{-a|t|}$ ,  $a > 0$ .

$$\begin{aligned}
 X(j\Omega) &= \int_{-\infty}^{\infty} e^{-a|t|} e^{-j\Omega t} dt \\
 &= \int_{-\infty}^0 e^{at} e^{-j\Omega t} dt + \int_0^{\infty} e^{-at} e^{-j\Omega t} dt \\
 &= \int_{-\infty}^0 e^{(a-j\Omega)t} dt + \int_0^{\infty} e^{-(a+j\Omega)t} dt \\
 &= \frac{1}{a-j\Omega} + \frac{1}{a+j\Omega} \\
 &= \frac{2a}{a^2 + \Omega^2}.
 \end{aligned}$$



**FIGURE 1.5** FT pair of  $e^{-a|t|}$ . (a) Time-domain. (b) Frequency-domain.

The signal  $x(t)$  and the magnitude response of the FT of  $e^{-a|t|}$  are depicted in Figure 1.5. The phase is zero, since the signal is even symmetric in the time-domain.

3. Find the Fourier transform of the sinusoidal signals:

a.  $x(t) = \cos(\Omega_0 t)$

b.  $x(t) = \sin(\Omega_0 t)$

Note that signals such as sinusoids that exist for all time are not absolutely integrable. Absolute integrability is a sufficient condition for the existence of FT, but it is not a necessary condition. These difficulties can be solved by introducing Dirac delta function in the frequency-domain as detailed below.

a.  $x(t) = \cos(\Omega_0 t)$

$$x(t) = \frac{1}{2} [e^{j\Omega_0 t} + e^{-j\Omega_0 t}] = \frac{1}{2} e^{j\Omega_0 t} + \frac{1}{2} e^{-j\Omega_0 t}.$$

On comparing the above equation with the expression of the Fourier series of a continuous-time periodic signal, we obtain the Fourier series coefficients of  $x(t)$  as

$$a_1 = a_{-1} = 0.5, \text{ for } k = \pm 1 \text{ and } a_k = 0, \text{ for } k \neq \pm 1. \quad (1.38)$$

Therefore, using Equation 1.26

$$\begin{aligned} X(j\Omega) &= 2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\Omega - k\Omega_0) \\ &= \pi \delta(\Omega - \Omega_0) + \pi \delta(\Omega + \Omega_0). \end{aligned}$$

The above CTFT result is depicted in Figure 1.6(a).

b.  $x(t) = \sin(\Omega_0 t)$

$$x(t) = \frac{1}{2j} [e^{j\Omega_0 t} - e^{-j\Omega_0 t}] = \frac{1}{2j} e^{j\Omega_0 t} - \frac{1}{2j} e^{-j\Omega_0 t}.$$

From the above results, the Fourier series coefficients are

$$a_1 = \frac{1}{2j}, \quad a_{-1} = -\frac{1}{2j} \text{ for } k = \pm 1 \text{ and } a_k = 0, \text{ for } k \neq \pm 1. \quad (1.39)$$

Therefore,

$$\begin{aligned} X(j\Omega) &= 2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\Omega - k\Omega_0) \\ &= \frac{\pi}{j} \delta(\Omega - \Omega_0) - \frac{\pi}{j} \delta(\Omega + \Omega_0). \end{aligned}$$

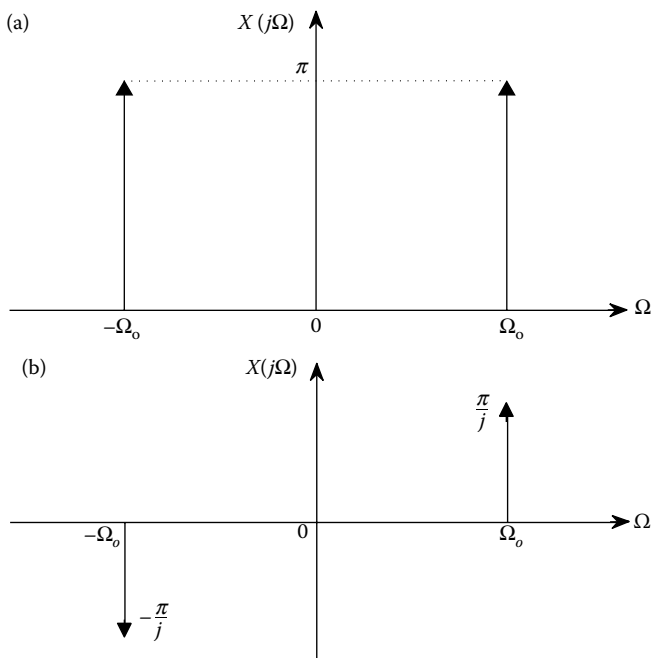
The CTFT of the sine signal is shown in Figure 1.6(b).

4. Determine the CTFT of a rectangular signal given by

$$x(t) = \begin{cases} 1, & |t| \leq \frac{\tau}{2} \\ 0, & |t| > \frac{\tau}{2}. \end{cases}$$

$$\begin{aligned} X(j\Omega) &= \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt = \frac{e^{j\Omega\tau/2} - e^{-j\Omega\tau/2}}{j\Omega} \\ &= 2 \frac{\sin(\Omega\tau/2)}{\Omega} = \tau \operatorname{sinc}\left(\frac{\Omega\tau}{2\pi}\right). \end{aligned}$$

Here, we have used the fact that  $\operatorname{sinc}(t) = \sin(\pi t)/(\pi t)$ . The time-domain and frequency-domain plots are shown in Figure 1.7.

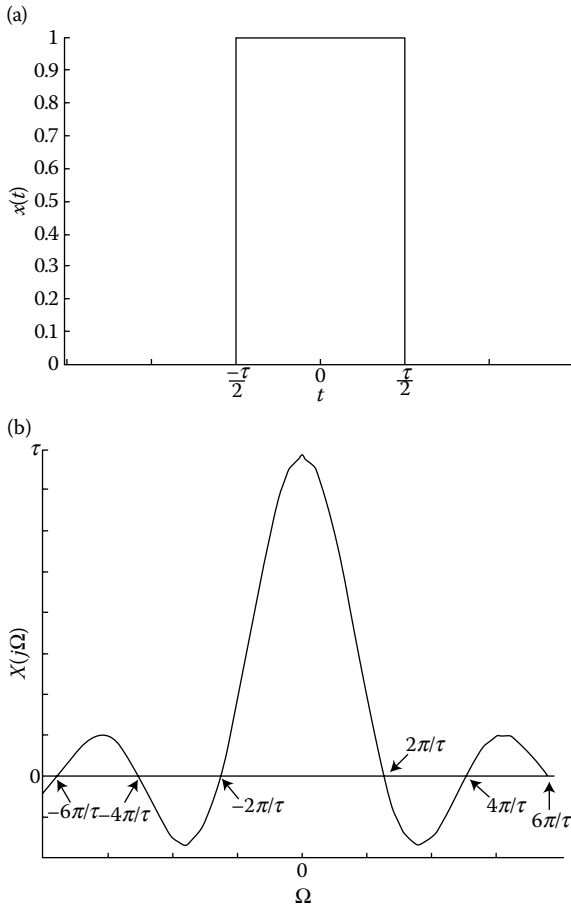


**FIGURE 1.6** CTFT of sinusoidal signals. (a) CTFT of cosine signal. (b) CTFT of sine signal.

5. Determine the CTFT of a shifted rectangular pulse signal

$$\begin{aligned}
 x(t) &= \begin{cases} 1, & 0 \leq t \leq \tau \\ 0, & \text{otherwise.} \end{cases} \\
 X(j\Omega) &= \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt = \int_0^{\tau} 1 \cdot e^{-j\Omega t} dt = \frac{e^{-j\Omega\tau/2}}{j\Omega} [e^{j\Omega\tau/2} - e^{-j\Omega\tau/2}] \\
 &= 2e^{-j\Omega\tau/2} \frac{\sin(\Omega\tau/2)}{\Omega} \\
 &= \tau e^{-j\Omega\tau/2} \text{sinc}\left(\frac{\Omega\tau}{2\pi}\right).
 \end{aligned}$$

Here also, we have used the fact that  $\text{sinc}(t) = \sin(\pi t)/(\pi t)$ . By comparing the above result with the result of the previous example, it can be observed that there is only a phase shift involved in the FT. The same result can also be obtained by using the time-shifting property of the CTFT.



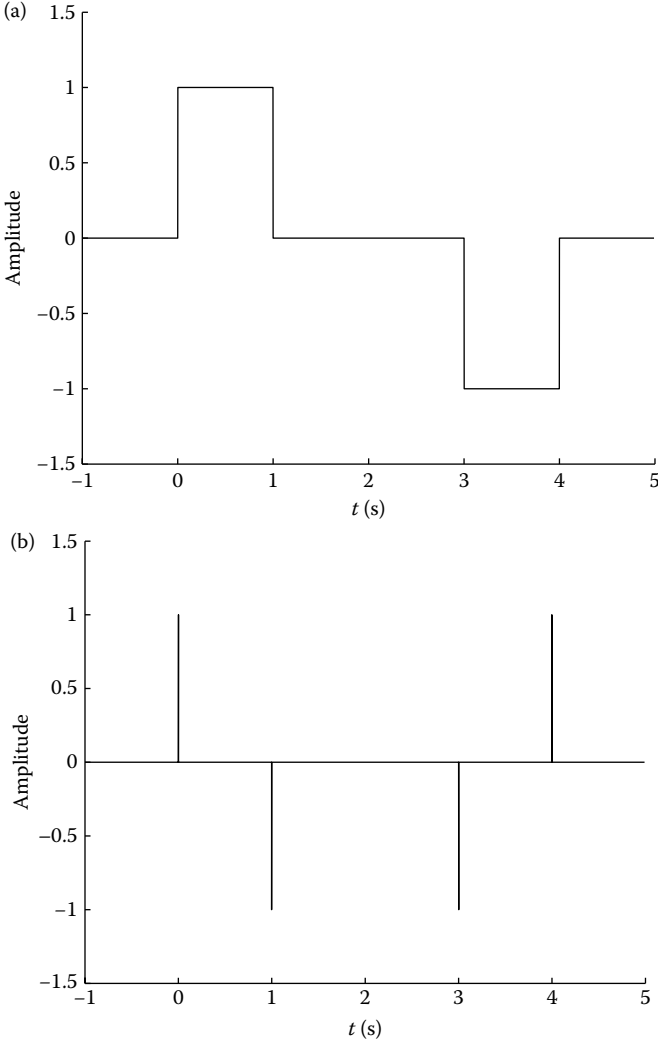
**FIGURE 1.7** FT of a rectangular pulse. (a) Time-domain. (b) Frequency-domain.

6. Consider a signal whose CTFT is given by

$$X(j\Omega) = \begin{cases} 1, & |\Omega| \leq W \\ 0, & |\Omega| > W. \end{cases}$$

Then, using the synthesis relation given in Equation (1.25), we can obtain the time-domain signal as

$$x(t) = \frac{1}{2\pi} \int_{-W}^W e^{j\Omega t} d\Omega = \frac{1}{2\pi} \left[ \frac{2 \sin(Wt)}{t} \right]$$



**FIGURE 1.8** Figure for Example 7. (a) Input  $x(t)$ . (b)  $dx(t)/dt$ .

$$\begin{aligned} &= \frac{\sin(Wt)}{\pi t} \\ &= \frac{W}{\pi} \operatorname{sinc}\left(\frac{Wt}{\pi}\right). \end{aligned}$$

We can observe the duality of this result with that of Example 4.

**7. An application of differentiation property:** Consider a signal  $x(t)$  displayed in Figure 1.8(a).

Differentiating the signal  $x(t)$ , we get an impulse train given in Equation 1.40 below. When we take the derivative of  $x(t)$  which is a piecewise constant signal, the constant region goes to zero. The infinite slope that occurs during the transition is represented by  $a \cdot \delta(t)$ , where  $a$  is the magnitude of transition. In this example,  $a = 1$  for all the transitions. Therefore,

$$\frac{dx(t)}{dt} = [\delta(t) - \delta(t - 1) - \delta(t - 3) + \delta(t - 4)]. \quad (1.40)$$

The impulse train is shown in Figure 1.8(b). Now taking the CTFT on either side of the above relation, and applying the differentiation property, we obtain

$$\begin{aligned} j\Omega X(j\Omega) &= [1 - e^{-j\Omega} - e^{-j3\Omega} + e^{-j4\Omega}] \\ &= e^{-j2\Omega} [e^{j2\Omega} - e^{j\Omega} - e^{-j\Omega} + e^{-j2\Omega}] \\ &= e^{-j2\Omega} [e^{j2\Omega} + e^{-j2\Omega} - (e^{j\Omega} + e^{-j\Omega})]. \end{aligned}$$

Upon simplification, the FT of  $x(t)$  turns out to be

$$X(j\Omega) = 2e^{-j2\Omega} \left[ \frac{\cos(2\Omega) - \cos(\Omega)}{j\Omega} \right].$$

Thus, the differentiation property can be used to simplify computations.

8. **An application of duality property:** Consider the following signal:

$$x(t) = \frac{2}{t^2 + 1}.$$

Using the result of Example 2 and recalling the duality property (Equation 1.35), we obtain (see Table 1.1)

$$e^{-a|t|} \xleftrightarrow{\mathcal{F}} \frac{2a}{a^2 + \Omega^2}.$$

Now with  $a = 1$  and using the duality property, we get

$$\frac{2}{t^2 + 1} \xleftrightarrow{\mathcal{F}} 2\pi e^{-|\Omega|}.$$

In order to summarize the results of this section, Table 1.1 contains a list of some useful CTFT pairs.

**TABLE 1.1**  
Basic CTFT Pairs

Signal	Fourier Transform
$\sum_{k=-\infty}^{\infty} a_k e^{jk\Omega_0 t}$	$2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\Omega - k\Omega_0)$
$e^{j\Omega_0 t}$	$2\pi \delta(\Omega - \Omega_0)$
$\cos(\Omega_0 t)$	$\pi [\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)]$
$\sin(\Omega_0 t)$	$\frac{\pi}{j} [\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$
$x(t) = 1$	$2\pi \delta(\Omega)$
$\delta(t)$	1
$\delta(t - t_0)$	$e^{-j\Omega t_0}$
$\sum_{n=-\infty}^{\infty} \delta(t - nT)$	$\frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - \Omega_0)$
$x(t) = \begin{cases} 1, &  t  \leq \frac{\tau}{2} \\ 0, &  t  > \frac{\tau}{2} \end{cases}$	$\tau \operatorname{sinc}\left(\frac{\Omega\tau}{2\pi}\right)$
$x(t) = \begin{cases} 1, & 0 \leq t \leq \tau \\ 0, & \text{otherwise} \end{cases}$	$\tau e^{-j\Omega\tau/2} \operatorname{sinc}\left(\frac{\Omega\tau}{2\pi}\right)$
$\frac{W}{\pi} \operatorname{sinc}\left(\frac{Wt}{\pi}\right)$	$X(j\Omega) = \begin{cases} 1, &  \Omega  \leq W \\ 0, &  \Omega  > W \end{cases}$
$u(t)$	$\frac{1}{j\Omega} + \pi \delta(\Omega)$
$e^{-at} u(t)$	$\frac{1}{a + j\Omega}, \quad \operatorname{Re}\{a\} > 0$
$te^{-at} u(t)$	$\frac{1}{(a + j\Omega)^2}, \quad \operatorname{Re}\{a\} > 0$
$\frac{t^{n-1}}{(n-1)!} e^{-at} u(t)$	$\frac{1}{(a + j\Omega)^n}, \quad \operatorname{Re}\{a\} > 0$
$e^{-a t }$	$\frac{2a}{a^2 + \Omega^2}, \quad \operatorname{Re}\{a\} > 0$
$x(t) = \begin{cases} -1, & t < 0 \\ 0, & t = 0 \\ 1, & t > 0 \end{cases}$	$\frac{2}{j\Omega}$
$\frac{2}{t^2 + 1}$	$2\pi e^{- \Omega }$

### 1.3 Discrete-Time Fourier Transform

The DTFT is the counterpart of CTFT for handling discrete-time signals. The basic concepts of FT are common to both the continuous- and discrete-time signals. In discrete time also, any periodic signal  $\tilde{x}[n]$  (with period  $N$ ) can be represented in terms of its discrete Fourier series representation, given by the



sum of complex exponentials,

$$\tilde{x}[n] = \sum_{k=\langle N \rangle} a_k e^{j\frac{2\pi}{N}kn} \quad (1.41)$$

$$a_k = \frac{1}{N} \sum_{n=\langle N \rangle} \tilde{x}[n] e^{-j\frac{2\pi}{N}kn}. \quad (1.42)$$

Here,  $k = \langle N \rangle$  denotes that  $k$  can take any  $N$  consecutive values. The terms in the series are harmonics of the fundamental frequency,  $2\pi/N$ . For the discrete-time case, the Fourier series is always convergent, since it is only a finite summation.

We now extend the frequency-domain representation to include more general aperiodic signals. Let us define a function  $X(e^{j\omega})$  as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}. \quad (1.43)$$

Comparing Equations 1.42 and 1.43, we can see that  $a_k$ 's are the samples of  $X(e^{j\omega})$ , spaced at  $\omega_0 = 2\pi/N$  in the frequency-domain as

$$a_k = \frac{1}{N} X(e^{jk\omega_0}). \quad (1.44)$$

As  $N \rightarrow \infty$ ,  $\omega_0$  becomes infinitesimally small. This indicates that  $X(e^{j\omega})$  is sampled with spacing  $\omega_0 \rightarrow 0$ . Thus,  $X(e^{j\omega})$  can be viewed as a continuous function. Similarly, the summation in Equation 1.41 is carried out over  $N$  consecutive intervals of width  $\omega_0 = 2\pi/N$  and the total interval of integration has a width of  $2\pi$ . Hence, unlike the CTFT where  $\Omega$  range is over the whole real axis, the DTFT requires only  $\omega$  values in the interval  $[0, 2\pi]$ .

The DTFT can be derived by taking the CTFT of a sampled signal. The IDTFT and the DTFT expressions are given by

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega. \quad (1.45)$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}. \quad (1.46)$$

Equation 1.45 represents the inverse DTFT (IDTFT), also known as the synthesis formula, while Equation 1.46 gives the DTFT, known as the analysis formula. The discrete-time periodic signals can be included within the framework of DTFT by interpreting the transform of a periodic signal as an impulse train in the frequency-domain as

$$X(e^{j\omega}) = \sum_{l=-\infty}^{\infty} \sum_{k=\langle N \rangle} 2\pi a_k \delta\left(\omega - \frac{2\pi}{N}k - 2\pi l\right). \quad (1.47)$$

The DTFT of  $x[n]$  is said to exist only if Equation 1.46 converges in some sense. There are two types of convergence (also known as summability) which are defined as follows.

- *Absolute summability:* If  $x[n]$  is an absolute summable sequence, then

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty. \quad (1.48)$$

Absolute summability is a sufficient condition for the existence of the DTFT.

- *Square summability:* Some sequences may not be absolutely summable, but they may be square summable, that is,

$$\lim_{M \rightarrow \infty} \int_{-\pi}^{\pi} |X(e^{j\omega}) - X_M(e^{j\omega})|^2 d\omega = 0, \quad (1.49)$$

wherein  $X_M(e^{j\omega})$ , we consider only a finite sequence of length  $M$ . Thus, the sequence is square summable if the mean square error between  $X(e^{j\omega})$  and  $X_M(e^{j\omega})$  tends to zero as  $M \rightarrow \infty$ . The DTFT of a sequence can exist under square summability condition as well.

### 1.3.1 Properties of DTFT

Many of the DTFT properties are exact parallels of the properties of continuous-time case, except for a few differences (which we shall indicate). The commonly used properties of DTFT are described in this section. Let  $x[n]$  and  $y[n]$  be time-domain signals and their corresponding DTFTs be  $X(e^{j\omega})$  and  $Y(e^{j\omega})$ , respectively. The DTFT pairs can be expressed as

$$x[n] \xleftrightarrow{\mathcal{F}} X(e^{j\omega}) \quad \text{and} \quad y[n] \xleftrightarrow{\mathcal{F}} Y(e^{j\omega}). \quad (1.50)$$

1. *Periodicity property:* Since the DTFT is periodic in  $\omega$  with a period  $2\pi$ , we can write

$$X(e^{j(\omega+2\pi)}) = X(e^{j\omega}). \quad (1.51)$$

2. *Linearity property:* For any arbitrary constants  $a$  and  $b$ , the DTFT of the weighted sum of two sequences is

$$ax[n] + by[n] \xleftrightarrow{\mathcal{F}} aX(e^{j\omega}) + bY(e^{j\omega}). \quad (1.52)$$

3. *Time shifting property:*

$$x[n - n_0] \xleftrightarrow{\mathcal{F}} e^{-j\omega n_0} X(e^{j\omega}). \quad (1.53)$$

However, note that both  $X(e^{j\omega})$  and  $e^{-j\omega n_0} X(e^{j\omega})$  have the same magnitude responses with only a phase shift introduced due to the time-shifting property. Therefore, delaying the time sequence has the effect of shifting the phase of its transformed version, while the magnitude response remains the same.

4. *Frequency shifting property:*

$$e^{j\omega_0 n} x[n] \xleftrightarrow{\mathcal{F}} X(e^{j(\omega - \omega_0)}). \quad (1.54)$$

5. *Time reversal property:* When the sequence  $x[n]$  is flipped (i.e., mirror image over  $-n$ ), then

$$x[-n] \xleftrightarrow{\mathcal{F}} X(e^{-j\omega}) \quad \text{and} \quad (1.55)$$

$$x[-n] \xleftrightarrow{\mathcal{F}} X^*(e^{j\omega}), \quad \text{only if } x[n] \text{ is real.} \quad (1.56)$$

6. *Convolution property:* The convolution of two sequences corresponds to multiplication of their corresponding DTFTs.

$$x[n] * y[n] \xleftrightarrow{\mathcal{F}} X(e^{j\omega})Y(e^{j\omega}). \quad (1.57)$$

We will now proceed to illustrate that the time-shifting property is a special case of the convolution property. According to Equation 1.46, the shifted impulse has the following frequency response:

$$\delta[n - n_d] \xleftrightarrow{\mathcal{F}} e^{-j\omega n_d}. \quad (1.58)$$

For any arbitrary input signal  $x[n]$  and the impulse response  $h[n]$  of an LTI system defined as

$$h[n] = \delta[n - n_d] \xleftrightarrow{\mathcal{F}} H(e^{j\omega}) = e^{-j\omega n_d}, \quad (1.59)$$

then the output is

$$y[n] = x[n] * \delta[n - n_d] \xleftrightarrow{\mathcal{F}} Y(e^{j\omega}) = e^{-j\omega n_d} X(e^{j\omega}). \quad (1.60)$$

7. *Differentiation in the frequency-domain:* Multiplication of  $x[n]$  by  $n$  results in

$$nx[n] \xleftrightarrow{\mathcal{F}} j \frac{d}{d\omega} X(e^{j\omega}). \quad (1.61)$$

8. *Windowing or modulation property:* Let  $w[n]$  be a window sequence, while  $x[n]$  is the input.

$$\text{If } w[n] \xleftrightarrow{\mathcal{F}} W(e^{j\omega}) \quad (1.62)$$

$$\text{and } y[n] = x[n].w[n] \quad (1.63)$$

$$\text{then, } Y(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})W(e^{j(\omega-\theta)})d\theta. \quad (1.64)$$

Equation 1.64 represents a periodic convolution, that is, convolution of two periodic functions with the limits of integration extending over one period (either  $-\pi$  to  $\pi$  or 0 to  $2\pi$ ). The duality in the FT theorems is evident when we compare the convolution and modulation theorems. In the continuous-time case, this duality is complete. We have stated that the convolution in the time-domain is equivalent to multiplication in the frequency-domain (and vice versa) for continuous-time signals. However, in the discrete-time case, this gets slightly modified. In the discrete time, fundamental differences arise because the DTFT is a sum, whereas the inverse transform is an integral over a continuous-time period ( $-\pi$  to  $\pi$ ) or (0 to  $2\pi$ ). The convolution of two sequences is equivalent to multiplication of the corresponding *periodic* FTs. Conversely, the multiplication of discrete-time sequences leads to the *periodic* convolution of their individual DTFTs. This is an essential distinction between the properties of the CTFT and the DTFT.

9. *Parseval's Theorem:* This theorem essentially relates the energy ( $E$ ) in the time- and the frequency-domains. For a sequence  $x[n]$ ,

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega. \quad (1.65)$$

The generalized Parseval's relation for two signals  $x[n]$  and  $y[n]$  is given by

$$\sum_{k=-\infty}^{\infty} x[k]y^*[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})Y^*(e^{j\omega})d\omega. \quad (1.66)$$

10. *Symmetry properties:* Some of the symmetry properties of the DTFT are presented below:

$$x^*[n] \xleftrightarrow{\mathcal{F}} X^*(e^{-j\omega}). \quad (1.67)$$

$$x^*[-n] \xleftrightarrow{\mathcal{F}} X^*(e^{j\omega}). \quad (1.68)$$

$$\text{Real part: } \mathcal{Re}\{x[n]\} = \frac{x[n] + x^*[n]}{2} \xleftrightarrow{\mathcal{F}} X_e(e^{j\omega}) = \frac{X(e^{j\omega}) + X^*(e^{-j\omega})}{2}. \quad (1.69)$$

$$\text{Imaginary part: } j\mathcal{Im}\{x[n]\} = \frac{x[n] - x^*[n]}{2} \xleftrightarrow{\mathcal{F}} X_o(e^{j\omega}) = \frac{X(e^{j\omega}) - X^*(e^{-j\omega})}{2}. \quad (1.70)$$

$$\text{Even part: } x_e[n] = \frac{x[n] + x^*[-n]}{2} \xleftrightarrow{\mathcal{F}} X_R(e^{j\omega}) = \frac{X(e^{j\omega}) + X^*(e^{j\omega})}{2}. \quad (1.71)$$

$$\text{Odd part: } x_o[n] = \frac{x[n] - x^*[-n]}{2} \xleftrightarrow{\mathcal{F}} jX_I(e^{j\omega}) = \frac{X(e^{j\omega}) - X^*(e^{j\omega})}{2}. \quad (1.72)$$

When  $x[n]$  is real, its DTFT exhibits the following characteristics:

$$X(e^{j\omega}) = X^*(e^{-j\omega}). \quad (1.73)$$

$$\text{Real part: } X_R(e^{j\omega}) = X_R(e^{-j\omega}), \quad \text{Imaginary part: } X_I(e^{j\omega}) = -X_I(e^{-j\omega}). \quad (1.74)$$

$$\text{Magnitude: } |X(e^{j\omega})| = |X(e^{-j\omega})|, \quad \text{Phase: } \angle X(e^{j\omega}) = -\angle X(e^{-j\omega}). \quad (1.75)$$

### 1.3.2 Examples of DTFT

1. Find the DTFT of the signal  $x[n] = \cos(\omega_0 n)$ .  
The given signal can be expanded as

$$x[n] = \frac{1}{2}e^{j\omega_0 n} + \frac{1}{2}e^{-j\omega_0 n}.$$

Comparing the above equation with the Fourier series expansion in Equation (1.22), we observe that  $a_1 = \frac{1}{2}$  and  $a_{-1} = \frac{1}{2}$ . Thus, the DTFT of  $x[n]$  can be written as a sum of weighted and shifted impulse trains as follows:

$$X(e^{j\omega}) = \sum_{l=-\infty}^{\infty} \pi[\delta(\omega - \omega_0 + 2\pi l) + \delta(\omega + \omega_0 + 2\pi l)].$$

2. Find the DTFT of the signal  $x[n] = \sin(\omega_0 n)$ .  
Now  $x[n]$  can be expanded as

$$x[n] = \frac{1}{2j} e^{j\omega_0 n} - \frac{1}{2j} e^{-j\omega_0 n}.$$

We obtain the Fourier series coefficients as  $a_1 = 1/2j$  and  $a_{-1} = -1/2j$ .  
Thus, the DTFT of  $x[n]$  can be written as

$$X(e^{j\omega}) = \sum_{l=-\infty}^{\infty} \frac{\pi}{j} [\delta(\omega - \omega_0 + 2\pi l) - \delta(\omega + \omega_0 + 2\pi l)].$$

3. Determine the DTFT of a rectangular pulse signal defined as

$$x[n] = \begin{cases} 1, & |n| \leq N_1 \\ 0, & |n| > N_1 \end{cases}.$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = \sum_{n=-N_1}^{N_1} e^{-j\omega n}.$$

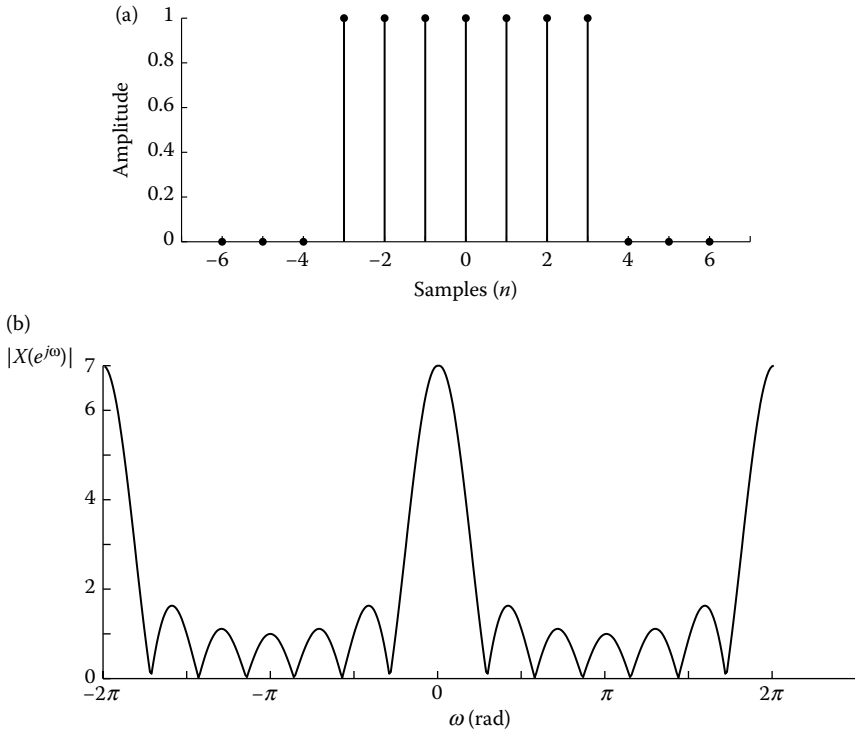
Let  $m = (n + N_1)$  then, we can write  $n = (m - N_1)$  and substitute as

$$\begin{aligned} X(e^{j\omega}) &= \sum_{m=0}^{2N_1} e^{-j\omega m} e^{j\omega N_1} = e^{j\omega N_1} \frac{[1 - e^{-j\omega(2N_1+1)}]}{1 - e^{-j\omega}} \\ &= \frac{e^{j\omega(N_1+\frac{1}{2})} - e^{-j\omega(N_1+\frac{1}{2})}}{e^{j\omega/2} - e^{-j\omega/2}} \\ &= \frac{\sin(\omega(2N_1 + 1)/2)}{\sin(\omega/2)}. \end{aligned}$$

The time function  $x[n]$  and the DTFT are shown in Figure 1.9.

4. Find the DTFT of a shifted rectangular pulse signal defined as

$$x[n] = \begin{cases} 1, & 0 \leq n \leq 2N_1 \\ 0, & \text{otherwise.} \end{cases}$$



**FIGURE 1.9** Rectangular pulse and its DTFT for  $N_1 = 3$ . (a) Input  $x(n]$ . (b) DTFT  $X(e^{j\omega})$ .

We can write,

$$\begin{aligned}
 X(e^{j\omega}) &= \sum_{n=0}^{2N_1} e^{-j\omega n} \\
 &= 1 + e^{-j\omega} + \dots + e^{-j\omega(2N_1)} \\
 &= \frac{1 - e^{-j\omega(2N_1+1)}}{1 - e^{-j\omega}} \\
 &= e^{-j(\omega N_1)} \frac{\sin(\omega(2N_1 + 1)/2)}{\sin(\omega/2)}.
 \end{aligned}$$

Here,  $(2N_1 + 1)$  is the length of the sequence and  $N_1$  is the symmetry point. The closed-form expression is obtained by using the geometric summation formula. We could also obtain the same result by applying the time-shifting property of the DTFT to the signal considered in Example 3.

5. Determine the DTFT of a triangular pulse signal

$$x[n] = \begin{cases} n + 1, & 0 \leq n \leq N_1 - 1 \\ 2N_1 - 1 - n, & N_1 \leq n \leq 2N_1 - 1 \\ 0, & \text{otherwise.} \end{cases}$$

A sample triangular pulse signal for  $N_1 = 6$  is shown in Figure 1.10. We can simplify the calculations by expressing this sequence as the self-convolution of a rectangular pulse with itself, given as follows:

$$x_1[n] = \begin{cases} 1, & 0 \leq n \leq N_1 - 1 \\ 0, & \text{otherwise} \end{cases}$$

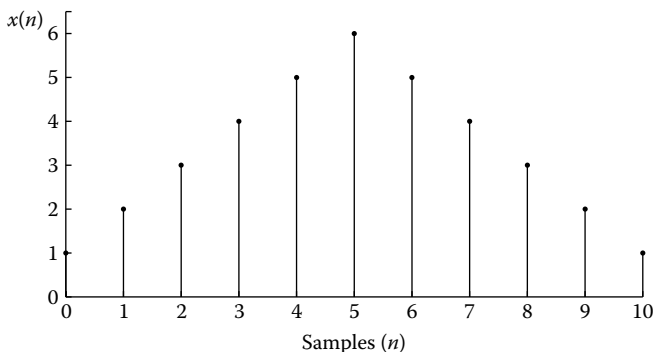
and  $x[n] = x_1[n] * x_1[n] \Leftrightarrow X(e^{j\omega}) = X_1(e^{j\omega})X_1(e^{j\omega})$ . (1.76)

Applying the result considered in Example 4 (shifted rectangular pulses), we obtain

$$X_1(e^{j\omega}) = e^{-j\omega(N_1-1)/2} \frac{\sin(\omega N_1/2)}{\sin(\omega/2)}. \quad (1.77)$$

Using the result from Equation 1.77 and substituting into Equation 1.76, we can obtain  $X(e^{j\omega})$  as

$$X(e^{j\omega}) = e^{-j\omega(N_1-1)} \left[ \frac{\sin(\omega N_1/2)}{\sin(\omega/2)} \right]^2.$$



**FIGURE 1.10**  
Figure for Example 5.



6. Consider the signal  $x[n] = a^n u[n]$ ,  $|a| < 1$ . Its DTFT is given by

$$\begin{aligned} X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} a^n u[n] e^{-j\omega n} = \sum_{n=0}^{\infty} (ae^{-j\omega})^n \\ &= \frac{1}{1 - ae^{-j\omega}}. \end{aligned}$$

7. Consider the signal  $x[n] = a^{|n|}$ ,  $|a| < 1$ . The DTFT of this signal is evaluated as follows:

$$\begin{aligned} X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} a^{|n|} e^{-j\omega n} \\ &= \sum_{n=-\infty}^{-1} a^{-n} e^{-j\omega n} + \sum_{n=0}^{\infty} a^n e^{-j\omega n} \\ &= \sum_{n=1}^{\infty} a^n e^{j\omega n} + \sum_{n=0}^{\infty} a^n e^{-j\omega n} \\ &= \frac{1}{1 - ae^{j\omega}} - 1 + \frac{1}{1 - ae^{-j\omega}} \\ &= \frac{(1 - a^2)}{1 - 2a \cos(\omega) + a^2}. \end{aligned}$$

8. **An application of differentiation property:** Find the DTFT of the signal

$$y[n] = (n + 1)a^n u[n], \quad |a| < 1.$$

Let  $x[n] = a^n u[n]$ , then  $y[n]$  can be written in terms of  $x[n]$  as

$$y[n] = nx[n] + x[n].$$

We know from Example 6 that  $X(e^{j\omega}) = 1/(1 - ae^{-j\omega})$ . Then, from the differentiation property given in Equation 1.61, we can obtain  $Y(e^{j\omega})$  as

$$\begin{aligned} nx[n] &\xleftrightarrow{\mathcal{F}} j \frac{d}{d\omega} X(e^{j\omega}) = \frac{ae^{-j\omega}}{(1 - ae^{-j\omega})^2} \\ Y(e^{j\omega}) &= \frac{ae^{-j\omega}}{(1 - ae^{-j\omega})^2} + \frac{1}{(1 - ae^{-j\omega})} \\ &= \frac{1}{(1 - ae^{-j\omega})^2}. \end{aligned}$$

9. Compute the DTFT of a finite-length exponential sequence (with  $|a| < 1$ ).

$$x[n] = \begin{cases} a^n, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases}$$

We can rewrite  $x[n]$  as the difference of two unit step functions as follows:

$$\begin{aligned} x[n] &= a^n u[n] - a^n u[n-N] \\ &= a^n u[n] - a^N a^{n-N} u[n-N]. \end{aligned}$$

Now computing the DTFT for  $|a| < 1$  and applying the result of Example 6, as well as the time-shifting property, we get

$$\begin{aligned} X(e^{j\omega}) &= \frac{1}{1 - ae^{-j\omega}} - \frac{a^N e^{-j\omega N}}{1 - ae^{-j\omega}} \\ &= \frac{1 - a^N e^{-j\omega N}}{1 - ae^{-j\omega}}. \end{aligned}$$

10. Compute the impulse response of an ideal discrete-time differentiator whose frequency response is given by

$$H_d(e^{j\omega}) = j\omega, \quad |\omega| \leq \pi.$$

Now using the synthesis equation (IDTFT), we get

$$\begin{aligned} h_d[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} j\omega e^{j\omega n} d\omega \\ &= \frac{j}{2\pi} \left[ \omega \frac{e^{j\omega n}}{jn} - \frac{e^{j\omega n}}{(jn)^2} \right]_{-\pi}^{\pi} \\ &= \frac{j}{2\pi} \left[ \pi \frac{e^{j\pi n}}{jn} + \frac{e^{j\pi n}}{n^2} + \pi \frac{e^{-j\pi n}}{jn} - \frac{e^{-j\pi n}}{n^2} \right] \\ &= \frac{\cos(\pi n)}{n} - \frac{\sin(\pi n)}{\pi n^2} \\ h_d[n] &= \begin{cases} \frac{(-1)^n}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases}. \end{aligned}$$

which is the impulse response of an ideal differentiator.

11. Compute the impulse response of an ideal Hilbert transformer defined by

$$H(e^{j\omega}) = \begin{cases} j, & -\pi \leq \omega \leq 0 \\ -j, & 0 \leq \omega \leq \pi \end{cases}.$$

Using the IDTFT equation

$$\begin{aligned}
 h[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \\
 &= \frac{1}{2\pi} \int_{-\pi}^0 j e^{j\omega n} d\omega - \frac{1}{2\pi} \int_0^{\pi} j e^{j\omega n} d\omega \\
 &= \frac{1}{2\pi} \left[ \frac{j e^{j\omega n}}{jn} \right]_{-\pi}^0 - \frac{1}{2\pi} \left[ \frac{j e^{j\omega n}}{jn} \right]_0^{\pi} \\
 &= \frac{1}{2\pi} \left[ \frac{2}{n} - \frac{2 \cos(\pi n)}{n} \right] \\
 &= \frac{1 - \cos(\pi n)}{\pi n}.
 \end{aligned}$$

$$h[n] = \begin{cases} \frac{2}{n\pi}, & n = \text{odd} \\ 0, & n = \text{even} \end{cases}.$$

which gives the impulse response of an ideal Hilbert transformer.

12. Determine the DTFT of the following sequence. (Note: This represents *Hann window* which will be discussed later.)

$$w_H[n] = 0.5 \left[ 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right], \quad 0 \leq n \leq N-1. \quad (1.78)$$

This sequence can be represented as

$$w_H[n] = 0.5 \left[ 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right] w_R[n],$$

where the rectangular window is  $w_R[n] = 1$ , is defined in the range  $0 \leq n \leq N-1$ . The DTFT of the rectangular window can be obtained from Example 4 as

$$W_R(e^{j\omega}) = e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)}.$$

Now applying the frequency-shifting property of the DTFT to  $W_R(e^{j\omega})$ , we can compute the DTFT of Equation 1.78 as

$$W_H(e^{j\omega}) = 0.5W_R(e^{j\omega}) - 0.25W_R\left(e^{j\left(\omega - \frac{2\pi}{N-1}\right)}\right) \\ - 0.25W_R\left(e^{j\left(\omega + \frac{2\pi}{N-1}\right)}\right)$$

$$\text{where, } W_R\left(e^{j\left(\omega - \frac{2\pi}{N-1}\right)}\right) = e^{-j\left(\omega - \frac{2\pi}{N-1}\right)\frac{(N-1)}{2}} \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]} \\ = e^{-j\omega\frac{(N-1)}{2}} e^{j\pi} \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}$$

$$W_R\left(e^{j\left(\omega - \frac{2\pi}{N-1}\right)}\right) = -e^{-j\omega\frac{(N-1)}{2}} \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}.$$

$$\text{Similarly, } W_R\left(e^{j\left(\omega + \frac{2\pi}{N-1}\right)}\right) = -e^{-j\omega\frac{(N-1)}{2}} \frac{\sin\left[\frac{N}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]}.$$

On adding the above terms, and applying the linearity property, we obtain the final expression for the DTFT of a Hann window as

$$W_H(e^{j\omega}) = 0.5e^{-j\omega(N-1)/2} \left\{ \frac{\sin(\omega N/2)}{\sin(\omega/2)} - 0.5 \left[ \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]} \right. \right. \\ \left. \left. + \frac{\sin\left[\frac{N}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]} \right] \right\}.$$

13. Determine the DTFT of the following sequence, which represents Hanning window,

$$w_H[n] = \left[ 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \right], \quad 0 \leq n \leq N-1.$$

Using the results of Example 12, we can directly write the DTFT of the sequence as

$$W_H(e^{j\omega}) = e^{-j\omega(N-1)/2} \left\{ 0.54 \frac{\sin(\omega N/2)}{\sin(\omega/2)} - 0.23 \left[ \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N-1}\right)\right]} \right. \right. \\ \left. \left. + \frac{\sin\left[\frac{N}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega + \frac{2\pi}{N-1}\right)\right]} \right] \right\}.$$

This expression is different from Example 12 only in terms of the coefficients.

**TABLE 1.2**  
Basic DTFT Pairs

Signal	Fourier Transform
$\sum_{k=(N)} a_k e^{jk\omega_0 n}$	$2\pi \sum_{l=-\infty}^{\infty} \sum_{k=<N>} a_k \delta(\omega - \omega_0 k - 2\pi l)$
$e^{j\omega_0 n}$	$2\pi \sum_{l=-\infty}^{\infty} \delta(\omega - \omega_0 - 2\pi l)$
$\cos(\omega_0 n)$	$\pi \sum_{l=-\infty}^{\infty} [\delta(\omega - \omega_0 + 2\pi l) + \delta(\omega + \omega_0 + 2\pi l)]$
$\sin(\omega_0 n)$	$\frac{\pi}{j} \sum_{l=-\infty}^{\infty} [\delta(\omega - \omega_0 + 2\pi l) - \delta(\omega + \omega_0 + 2\pi l)]$
$x[n] = 1$	$2\pi \sum_{l=-\infty}^{\infty} \delta(\omega - 2\pi l)$
$\delta[n]$	1
$\delta[n - n_0]$	$e^{-j\omega n_0}$
$\sum_{k=-\infty}^{\infty} \delta[n - kN]$	$\frac{2\pi}{N} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0)$
$x[n] = \begin{cases} 1, &  n  \leq N_1 \\ 0, &  n  > N_1 \end{cases}$	$\frac{\sin[\omega(N_1 + \frac{1}{2})]}{\sin(\omega/2)}$
$x[n] = \begin{cases} 1, & 0 \leq n \leq N_1 - 1 \\ 0, & \text{otherwise} \end{cases}$	$e^{-j\omega \frac{(N_1-1)}{2}} \frac{\sin(\omega N_1/2)}{\sin(\omega/2)}$
$\frac{W}{\pi} \text{sinc}\left(\frac{Wn}{\pi}\right)$	$X(e^{j\omega}) = \begin{cases} 1, &  \omega  \leq W \\ 0, & W <  \omega  \leq \pi \end{cases}$
$u[n]$	$\frac{1}{1 - e^{-j\omega}} + \pi \sum_{l=-\infty}^{\infty} \delta(\omega - 2\pi l)$
$a^n u[n],  a  < 1$	$\frac{1}{1 - ae^{-j\omega}}$
$a^{ n },  a  < 1$	$\frac{1 - a^2}{1 - 2a \cos(\omega) + a^2}$
$(n + 1)a^n u[n],  a  < 1$	$\frac{1}{(1 - ae^{-j\omega})^2}$
$h_d[n] = \begin{cases} \frac{(-1)^n}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases}$	$H_d(e^{j\omega}) = j\omega, \quad  \omega  \leq \pi$
$h[n] = \begin{cases} \frac{2}{n\pi}, & n = \text{odd} \\ 0, & n = \text{even} \end{cases}$	$H(e^{j\omega}) = \begin{cases} j, & -\pi \leq \omega \leq 0 \\ -j, & 0 \leq \omega \leq \pi \end{cases}$

The FTs of the commonly used signals are presented in Table 1.2.

### 1.4 Z-transform

As we have discussed in the previous sections, the transform-domain analysis of a signal is important in any signal processing application. In this section, we

introduce another transform-domain, which is a complex frequency-domain, called the  $z$ -plane. The transformation of a signal into the  $z$ -plane is called  $z$ -transform. The  $z$ -transform is a more generalized transformation when compared to the DTFT and is applicable to broader classes of signals. The  $z$ -transform of a signal  $x[n]$  is defined as

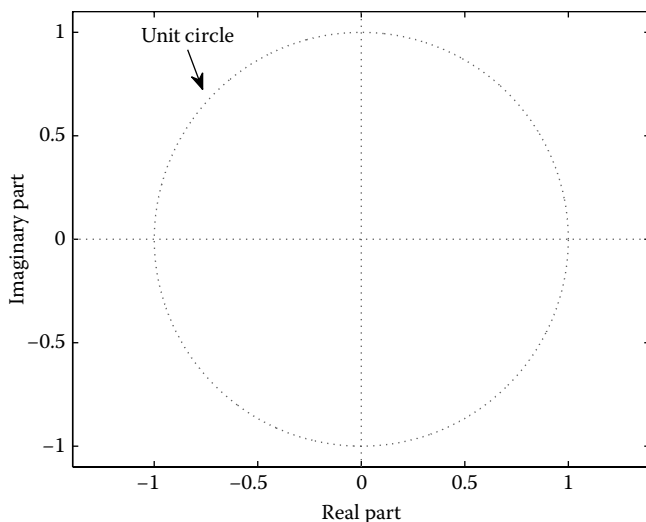
$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad (1.79)$$

where  $z = re^{j\omega}$ . Another advantage of  $z$ -transform is that it allows us to bring in the power of complex variable theory on problems of discrete-time signals. The primary roles of the  $z$ -transform in engineering practice are the study of system characteristics and the derivation of computational structures for implementing discrete-time systems on computers.

Figure 1.11 shows the  $z$ -plane which extends from  $-\infty < |z| < \infty$ . The circle shown in the  $z$ -plane is called the unit circle, where  $|z| = 1$ . On this circle,  $z = e^{j\omega}$ , hence the  $z$ -transform evaluated on the unit circle converges to the FT of the signal. Equation 1.79 is a power series, hence the power series converges under the criterion

$$\sum_{n=-\infty}^{\infty} |x[n]r^{-n}| < \infty. \quad (1.80)$$

The region in the  $z$ -plane where Equation 1.80 converges or  $z$ -transform converges is called the region of convergence (ROC). Owing to the multiplication



**FIGURE 1.11**  
z-plane.

of the sequence by the real exponential  $r^{-n}$ , it is possible for the  $z$ -transform to converge even if the corresponding DTFT does not. The FT can be considered as a special case of  $z$ -transform and it exists only if the ROC contains the unit circle. The  $z$ -transform is always specified with a ROC. Consider a signal  $a^n u[n]$  whose  $z$ -transform is given by

$$\begin{aligned} X(z) &= \sum_{n=0}^{\infty} (az^{-1})^n \\ &= 1 + az^{-1} + (az^{-1})^2 + (az^{-1})^3 + \dots \\ &= \frac{1}{1 - az^{-1}}, \quad |z| > |a|. \end{aligned} \quad (1.81)$$

Now consider another signal  $-a^n u(-n-1)$ , the  $z$ -transform of which is given by

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{-1} -(az^{-1})^n \\ &= -a^{-1}z - (a^{-1}z)^2 - (a^{-1}z)^3 \dots \\ &= \frac{1}{1 - az^{-1}}, \quad |z| < |a|. \end{aligned} \quad (1.82)$$

By comparing the  $z$ -transforms in the above two examples, we can see that the expressions are the same and the only difference is in their ROCs. Hence, ROC is required to uniquely represent the  $z$ -transform of a signal. The convergence of Equation 1.79 is dependent only on  $|z|$ , since

$$|X(z)| < \infty, \quad \text{if} \quad \sum_{n=-\infty}^{\infty} |x[n]| |z^{-n}| < \infty. \quad (1.83)$$

Hence, the ROC of the  $z$ -transform consists of all the values of  $z$  in the complex plane where the inequality in Equation 1.83 is satisfied. As a consequence of this, the ROC will be an annular region of the entire complex  $z$ -plane given by

$$R_{x-} < |z| < R_{x+}. \quad (1.84)$$

The lower limit  $R_{x-}$  may be zero and  $R_{x+}$  could possibly be  $\infty$ .

The inverse  $z$ -transform of  $X(z)$  can be obtained by the expression

$$x[n] = \frac{1}{2\pi j} \oint_c X(z) z^{n-1} dz, \quad (1.85)$$

TABLE 1.3

Basic z-Transform Pairs

Signal	z-Transform	ROC
$\delta[n]$	1	Entire z-plane
$u[n]$	$\frac{1}{1-z^{-1}}$	$ z  > 1$
$-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$ z  < 1$
$\delta[n-m]$	$z^{-m}$	All z, except at $z=0$ or $z=\infty$
$a^n u[n]$	$\frac{1}{1-az^{-1}}$	$ z  >  a $
$-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$ z  <  a $
$na^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z  >  a $
$-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z  <  a $
$\cos(\omega_0 n)u[n]$	$\frac{1 - \cos(\omega_0)z^{-1}}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}$	$ z  > 1$
$\sin(\omega_0 n)u[n]$	$\frac{\sin(\omega_0)z^{-1}}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}$	$ z  > 1$
$r^n \cos(\omega_0 n)u[n]$	$\frac{1 - r\cos(\omega_0)z^{-1}}{1 - 2r\cos(\omega_0)z^{-1} + r^2z^{-2}}$	$ z  >  r $
$r^n \sin(\omega_0 n)u[n]$	$\frac{r\sin(\omega_0)z^{-1}}{1 - 2r\cos(\omega_0)z^{-1} + r^2z^{-2}}$	$ z  >  r $
$a^n, 0 \leq n \leq N-1$	$\frac{1 - a^N z^{-N}}{1 - az^{-1}}$	Entire z-plane

where the symbol  $\oint_c$  denotes a contour integral in the z-plane over a counter-clockwise arbitrary closed path in the region of convergence and enclosing the origin  $z=0$ . In practice, we will not evaluate this integral directly, since that would require the knowledge of complex-function theory. Instead, we will evaluate the inverse z-transform by inspection using the one-to-one relationship between  $x[n]$  and  $X(z)$ . To facilitate this, the z-transform of some standard sequences are tabulated in Table 1.3.

A class of z-transform called the rational transforms—which are very important in signal processing applications—can be represented in the form

$$X(z) = \frac{P(z)}{Q(z)}, \quad (1.86)$$

where  $P(z)$  and  $Q(z)$  are polynomials in  $z$ . These systems can be represented by linear constant coefficient difference equations (LCCDEs). The roots of the denominator polynomial  $Q(z)$  are called poles and the roots of the numerator polynomial  $P(z)$  are called zeros. In this class of systems, the properties of the system can be completely interpreted in terms of the position of the poles



TABLE 1.4

z-Transform Properties

Signal	z-Transform	ROC
$ax_1[n] + bx_2[n]$	$aX_1(z) + bX_2(z)$	$R_{x_1} \cap R_{x_2}$
$x[n - n_0]$	$z^{-n_0}X(z)$	$R_x$ except at $z = 0$ or $z = \infty$
$z_0^n x[n]$	$X\left(\frac{z}{z_0}\right)$	$ z_0 R_x$
$nx[n]$	$-z \frac{dX(z)}{dz}$	$R_x$
$x^*[n]$	$X^*(z^*)$	$R_x$
$\mathcal{R}e\{x[n]\} = \frac{1}{2}[x[n] + x^*[n]]$	$\frac{1}{2}[X(z) + X^*(z^*)]$	Contains $R_x$
$\mathcal{I}m\{x[n]\} = \frac{1}{2j}[x[n] - x^*[n]]$	$\frac{1}{2j}[X(z) - X^*(z^*)]$	Contains $R_x$
$x[-n]$	$X\left(\frac{1}{z}\right)$	$\frac{1}{R_x}$
$x_1[n] * x_2[n]$	$X_1(z)X_2(z)$	$R_{x_1} \cap R_{x_2}$

and zeroes in the  $z$ -plane. These concepts are extensively used in different domains of signal processing. In the chapter dealing with filter design which we will come across in more detail.

The *properties of the z-transform* closely follow those of the DTFT. In addition, we also need to specify the ROC of the resulting signal. Let  $x_1[n]$  and  $x_2[n]$  be two arbitrary signals with the  $z$ -transforms  $X_1(z)$  and  $X_2(z)$ , respectively, and let their ROCs be  $R_{x_1}$  and  $R_{x_2}$ , respectively. Let  $a$  and  $b$  be two arbitrary constants. The properties of the  $z$ -transform are tabulated in Table 1.4.

The other two properties are:

- *Initial-value theorem*: For a causal signal  $x[n]$ , it turns out that

$$x[0] = \lim_{z \rightarrow \infty} X(z). \quad (1.87)$$

- *Parseval's theorem*: This relates the power or energy of  $x[n]$  to that of its  $z$ -transform.

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi j} \oint_c X(z)X^*\left(\frac{1}{z^*}\right)z^{-1}dz. \quad (1.88)$$

### 1.4.1 Examples of z-Transform

1. Consider a signal that is the sum of two real exponentials:

$$x[n] = \left(\frac{1}{2}\right)^n u[n] + \left(\frac{1}{3}\right)^n u[n] \quad (1.89)$$

The z-transform is given by

$$\begin{aligned}
 X(z) &= \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{1}{2}\right)^n u[n] + \left(\frac{1}{3}\right)^n u[n] \right\} z^{-n} \\
 &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2}\right)^n u[n] z^{-n} + \sum_{n=-\infty}^{\infty} \left(\frac{1}{3}\right)^n u[n] z^{-n} \\
 &= \sum_{n=0}^{\infty} \left(\frac{1}{2} z^{-1}\right)^n + \sum_{n=0}^{\infty} \left(\frac{1}{3} z^{-1}\right)^n \\
 &= \frac{1}{1 - \frac{1}{2} z^{-1}} + \frac{1}{1 - \frac{1}{3} z^{-1}} \\
 &= \frac{2 - \frac{5}{6} z^{-1}}{\left(1 - \frac{1}{2} z^{-1}\right) \left(1 - \frac{1}{3} z^{-1}\right)} \\
 &= \frac{2z \left(z - \frac{5}{12}\right)}{\left(z - \frac{1}{2}\right) \left(z - \frac{1}{3}\right)}. \tag{1.90}
 \end{aligned}$$

For the convergence of  $X(z)$ , both sums must converge. This requires that  $\left|\frac{1}{2}z^{-1}\right| < 1$  and  $\left|\frac{1}{3}z^{-1}\right| < 1$ , which implies  $|z| > \frac{1}{2}$  and  $|z| > \frac{1}{3}$ , respectively. Thus, from the properties of z-transform, the ROC is the region of overlap of both terms. Hence, the ROC of  $X(z)$  is given by

$$|z| > \frac{1}{2}. \tag{1.91}$$

2. Let us consider another signal

$$x[n] = -\left(\frac{1}{2}\right)^n u[-n-1] + \left(\frac{1}{3}\right)^n u[n]. \tag{1.92}$$

The z-transform can also be obtained in a more straightforward manner as given below: Note that the first sequence grows exponentially as  $n \rightarrow -\infty$ . From Table 1.3, it follows that

$$\begin{aligned}
 \left(\frac{1}{3}\right)^n u[n] &\stackrel{z}{\longleftrightarrow} \frac{1}{1 - \frac{1}{3} z^{-1}}, \quad |z| > \frac{1}{3} \\
 -\left(\frac{1}{2}\right)^n u[-n-1] &\stackrel{z}{\longleftrightarrow} \frac{1}{1 - \frac{1}{2} z^{-1}}, \quad |z| < \frac{1}{2}.
 \end{aligned}$$

By the linearity property of the z-transform,

$$\begin{aligned} X(z) &= \frac{1}{1 - \frac{1}{3}z^{-1}} + \frac{1}{1 - \frac{1}{2}z^{-1}}, \quad |z| > \frac{1}{3}, \quad |z| < \frac{1}{2}. \\ &= \frac{2 - \frac{5}{6}z^{-1}}{\left(1 - \frac{1}{2}z^{-1}\right)\left(1 - \frac{1}{3}z^{-1}\right)}. \end{aligned}$$

In this case, the ROC is the annular region  $\frac{1}{3} < |z| < \frac{1}{2}$ . Note that the ROC does not contain the unit circle; hence its FT does not exist (as it is not absolutely summable).

## 1.5 Discrete Fourier Transform

In one of the preceding sections, we have discussed the definition and properties of the DTFT, where  $\omega$  is a continuous variable. It is apparent that it is not possible to implement the FT pair for DTFT given in Equations 1.45 and 1.46 on general purpose digital computers or digital signal processors (DSPs). We see that Equation 1.45 has an integral sign, while Equation 1.46 requires infinitely several computations. Owing to these two difficulties, we define a new transform known as the discrete Fourier transform (DFT), which is amenable to digital implementation.

The DFT is defined for  $N$  samples of  $x[n]$  at  $N$  equally spaced frequencies  $\omega_k$ . Given the sequence  $x[n]$  for all  $n$ , its DTFT was defined in Equation 1.46. We now consider a periodic signal with period  $N$  and compute only  $N$  samples of  $X(e^{j\omega})$  for  $\omega = k\omega_0$ , for  $k = 0, 1, 2, \dots, N - 1$ , with  $\omega_0 = 2\pi/N$ . Then, we can define DFT as

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots, N - 1. \quad (1.93)$$

The inverse DFT (IDFT) can also be defined in a similar way as

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \dots, N - 1. \quad (1.94)$$

Notice that Equations 1.93 and 1.94 have similar forms, except for a scale factor and different signs of the exponential terms.

*Relation between DTFT and DFT:* DTFT of a sequence of length  $N$  defined over the range  $[0, N - 1]$  can be obtained from DFT using the relation

$$\begin{aligned}
 x[n] &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn} \\
 X(e^{j\omega}) &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \phi \left( \omega - \frac{2\pi k}{N} \right), \tag{1.95}
 \end{aligned}$$

$$\text{where } \phi(\omega) = e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)}.$$

*Proof:*

$$\begin{aligned}
 x[n] &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn} \\
 X(e^{j\omega}) &= \sum_{n=0}^{N-1} x[n] e^{-j\omega n} = \sum_{n=0}^{N-1} \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn} e^{-j\omega n} \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \sum_{n=0}^{N-1} e^{-j(\omega - \frac{2\pi}{N} k)n} \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left[ \frac{1 - e^{-j(\omega - \frac{2\pi}{N} k)N}}{1 - e^{-j(\omega - \frac{2\pi}{N} k)}} \right] \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{-j(\omega - \frac{2\pi}{N} k)(\frac{N-1}{2})} \frac{\sin \left[ (\omega - \frac{2\pi}{N} k) \frac{N}{2} \right]}{\sin \left[ (\omega - \frac{2\pi}{N} k) \frac{1}{2} \right]} \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \phi \left( \omega - \frac{2\pi k}{N} \right), \tag{1.96}
 \end{aligned}$$

$$\text{where, } \phi(\omega) = e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)}.$$

Equation 1.96 is called the DFT interpolation formula and  $\phi(\omega)$  is known as the interpolation function.

### 1.5.1 Properties of the DFT

The DFT of two finite duration sequences of length  $N$ ,  $x[n]$  and  $y[n]$ , is represented as

$$x[n] \xleftrightarrow{\mathcal{DFT}} X[k] \quad \text{and} \quad y[n] \xleftrightarrow{\mathcal{DFT}} Y[k]. \tag{1.97}$$

Let  $a$  and  $b$  be any two arbitrary constants.

1. *Linearity property:* If two finite-length sequences  $x_1[n]$  and  $x_2[n]$  are linearly combined as

$$x_3[n] = ax_1[n] + bx_2[n], \quad (1.98)$$

then the DFT of the resultant  $X_3[k]$  (i.e., the DFT of  $x_3[n]$ ) can be written as

$$X_3[k] = aX_1[k] + bX_2[k]. \quad (1.99)$$

If the individual sequences  $x_1[n]$  and  $x_2[n]$  are not of equal lengths, but are of lengths  $N_1$  and  $N_2$ , respectively, then the length of  $x_3[n]$  will have to be  $N_3 = \max(N_1, N_2)$ . It makes sense only if the DFTs are computed with the same lengths, that is,  $N \geq N_3$ . If  $N_2 > N_1$ , then  $X_1[k]$  is the DFT of the sequence  $x_1[n]$  padded with  $(N_2 - N_1)$  zeros. Therefore, we have to pad that sequence which is smaller in length with zeros, such that the lengths of the individual sequences are made equal.

2. *Circular shifting property:* When  $x[n]$  is shifted by  $m$ ,

$$x[(n - m)_N] \xleftrightarrow{\text{DFT}} e^{-j\frac{2\pi}{N}km} X[k]. \quad (1.100)$$

We note that  $n$  and  $k$  must be in the range  $0 \leq n < N - 1$ ,  $0 \leq k < N - 1$ . Here,  $(n - m)_N$  denotes modulo  $N$ . This type of shift is known as circular shift.

3. *Duality property:*

$$X[n] \xleftrightarrow{\text{DFT}} Nx[(-k)_N], \quad 0 \leq k \leq N - 1. \quad (1.101)$$

4. *Conjugation property:*

$$x^*[n] \xleftrightarrow{\text{DFT}} X^*[(-k)_N]. \quad (1.102)$$

5. *Time reversal property:* Here, we consider flipping the sequence with modulo  $N$ .

$$x[(-n)_N] \xleftrightarrow{\text{DFT}} X[-k], \quad 0 \leq n \leq N - 1. \quad (1.103)$$

6. *Symmetry properties:* If the even and odd parts of a signal  $x[n]$  are

$$x_e[n] = \frac{1}{2}[x[n] + x^*[(-n)_N]] \quad (1.104)$$

$$x_o[n] = \frac{1}{2}[x[n] - x^*[(-n)_N]], \quad (1.105)$$

then, the following relations hold:

$$\mathcal{R}e\{x[n]\} \xleftrightarrow{\mathcal{DFT}} X_e[k] = \frac{1}{2}[X[(k)_N] + X^*[(-k)_N]]. \quad (1.106)$$

$$j\mathcal{I}m\{x[n]\} \xleftrightarrow{\mathcal{DFT}} X_o[k] = \frac{1}{2}[X[(k)_N] - X^*[(-k)_N]]. \quad (1.107)$$

$$x_e[n] \xleftrightarrow{\mathcal{DFT}} \mathcal{R}e\{X[k]\}. \quad (1.108)$$

$$x_o[n] \xleftrightarrow{\mathcal{DFT}} j\mathcal{I}m\{X[k]\}. \quad (1.109)$$

When  $x[n]$  is a real sequence, then

$$X[k] = X^*[(-k)_N] \quad (1.110)$$

$$\mathcal{R}e\{X[k]\} = \mathcal{R}e\{X[(-k)_N]\} \quad \text{and} \quad \mathcal{I}m\{X[k]\} = -\mathcal{I}m\{X[(-k)_N]\}. \quad (1.111)$$

$$|X[k]| = |X[(-k)_N]| \quad \text{and} \quad \angle\{X[k]\} = -\angle\{X[(-k)_N]\}. \quad (1.112)$$

7. *Circular convolution property:* If  $x_1[n]$  and  $x_2[n]$  are  $N$  length sequences and

$$x_1[n] \xleftrightarrow{\mathcal{DFT}} X_1[k] \quad \text{and} \quad x_2[n] \xleftrightarrow{\mathcal{DFT}} X_2[k], \quad (1.113)$$

then their circular convolution (denoted by  $\circledast$ ) can be expressed as

$$x_3[n] = x_1[n] \circledast x_2[n] \quad (1.114)$$

$$= \sum_{m=0}^{N-1} x_1[(m)_N]x_2[(n-m)_N] \quad (1.115)$$

$$= x_1[n] \circledast x_2[n] \xleftrightarrow{\mathcal{DFT}} X_3[k] = X_1[k]X_2[k]. \quad (1.116)$$

8. *Multiplication of two sequences:* This is the converse of the previous property, which can be stated as follows:

$$x_1[n]x_2[n] \xleftrightarrow{\mathcal{DFT}} \frac{1}{N} \sum_{l=0}^{N-1} X_1[l]X_2[(k-l)_N]. \quad (1.117)$$

9. *Linear convolution using circular convolution:* Let  $x_1[n]$  be a sequence of length  $L$  and  $x_2[n]$  be a sequence of length  $P$ , then the length of the resultant linearly convolved sequence will be  $N_1 = L + P - 1$ . The linear convolution expression is given by

$$x_3[n] = \sum_{m=0}^{N-1} x_1[m]x_2[n-m]. \quad (1.118)$$

Their circular convolution expression was already given in Equation 1.115. In the DTFT domain, the linear convolution can be represented in terms of the multiplication of the individual DTFTs of  $x_1(n)$  and  $x_2(n)$  as follows:

$$X_3(e^{j\omega}) = X_1(e^{j\omega})X_2(e^{j\omega}). \quad (1.119)$$

Therefore, we can define the DFT as

$$X_3[k] = X_3(e^{j\frac{2\pi}{N}k}) = X_1(e^{j\frac{2\pi}{N}k})X_2(e^{j\frac{2\pi}{N}k}) = X_1[k]X_2[k], \quad 0 \leq k \leq N-1 \quad (1.120)$$

where  $X_1[k]$  and  $X_2[k]$  are  $N$ -point DFTs of  $x_1[n]$  and  $x_2[n]$ , respectively. Now upon reconstruction of  $x_3[n]$ , we get the following periodic sequence:

$$x_{3_p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n - rN], & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases} \quad (1.121)$$

Hence, the circular convolution that corresponds to  $X_1[k]X_2[k]$  is identical to the linear convolution corresponding to  $X_1(e^{j\omega})X_2(e^{j\omega})$ , if  $N$ , the lengths of  $X_1[k]X_2[k]$ , satisfies the condition  $N \geq N_1$ . Otherwise, there will be aliasing in  $x_{3_p}[n]$ . If both the sequences are padded with zeros, such that the total length of each sequence becomes  $(N_1 = L + P - 1)$ , then the circular convolution would be equivalent to the linear convolution. We require linear convolution in digital signal processing and, therefore, circular convolution can be used to compute linear convolution.

### 1.5.2 Examples of DFT

1. Compute the  $N$ -point DFT of a rectangular pulse,

$$x[n] = 1, \quad 0 \leq n \leq 4. \quad (1.122)$$

The period of  $x[n]$  is not mentioned and, therefore, we consider the following two cases.

Case 1: Period  $N = 5$

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^4 e^{-j\frac{2\pi}{5}nk} \\ &= \frac{1 - e^{-j2\pi k}}{1 - e^{-j\frac{2\pi}{5}k}} \end{aligned} \quad (1.123)$$

$$= \begin{cases} 5, & k = 0 \\ 0, & \text{otherwise} \end{cases} \quad (1.124)$$

$$= \{5, 0, 0, 0, 0\} \quad (1.125)$$

Case 2: Now let the period be  $N = 10$ . Then,

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^4 e^{-j\frac{2\pi}{10}nk} \quad (1.126)$$

$$= \frac{1 - e^{-j\pi k}}{1 - e^{-j\frac{2\pi}{10}k}}. \quad (1.127)$$

The time-domain sequences of cases 1 and 2 are given in Figure 1.12 and their corresponding DFTs are given in Figure 1.13.

For comparing the DFTs of both the sequences, we can consider the interpretation of the DFT as the periodic sampling of the DTFT. The DTFT of  $x[n]$  is given by

$$X(e^{j\omega}) = e^{-j2\omega} \frac{\sin(5\omega/2)}{\sin(\omega/2)}. \quad (1.128)$$

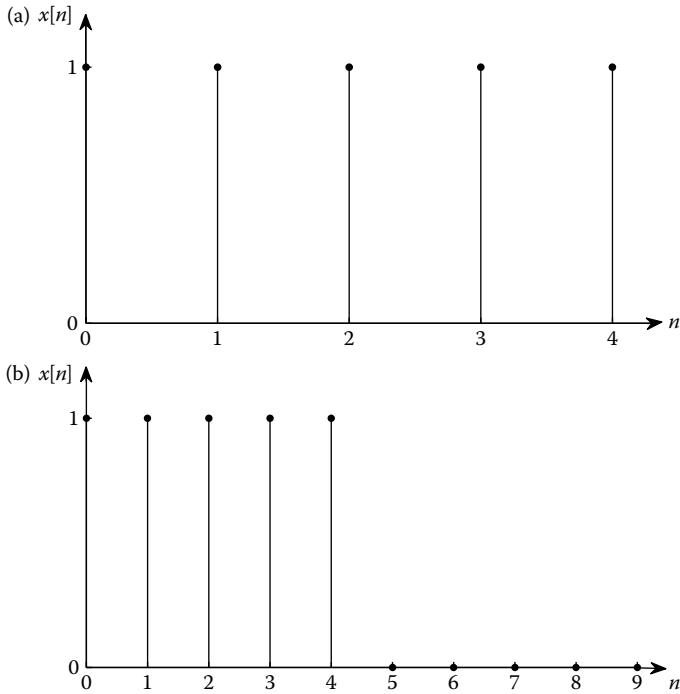
For  $N = 5$ ,  $X(e^{j\omega})$  is sampled at five equidistant points around the unit circle; if  $N$  is doubled ( $N = 10$ ), it is sampled at 10 equidistant points around the unit circle. Hence, the second case can be considered as  $x[n]$  padded with five zeros, so as to make  $N = 10$ . From this result, we can conclude that the effect of zero padding of  $x[n]$  does not improve the resolution, but only gives a better picture of the DFT spectrum.

Case 3: Now consider another sequence

$$x_1[n] = \{1, 0, 1, 0, 1, 0, 1, 0\}. \quad (1.129)$$

Find the DFT for  $N = 10$ .





**FIGURE 1.12** Two time-domain sequences. (a) Case 1:  $N = 5$ . (b) Case 2:  $N = 10$ .

The transform is defined as

$$X_1[k] = \sum_{n=0}^9 x_1[n] e^{-j\frac{2\pi}{10}nk} = \sum_{n=0}^9 x_1[n] W_N^{nk}, \tag{1.130}$$

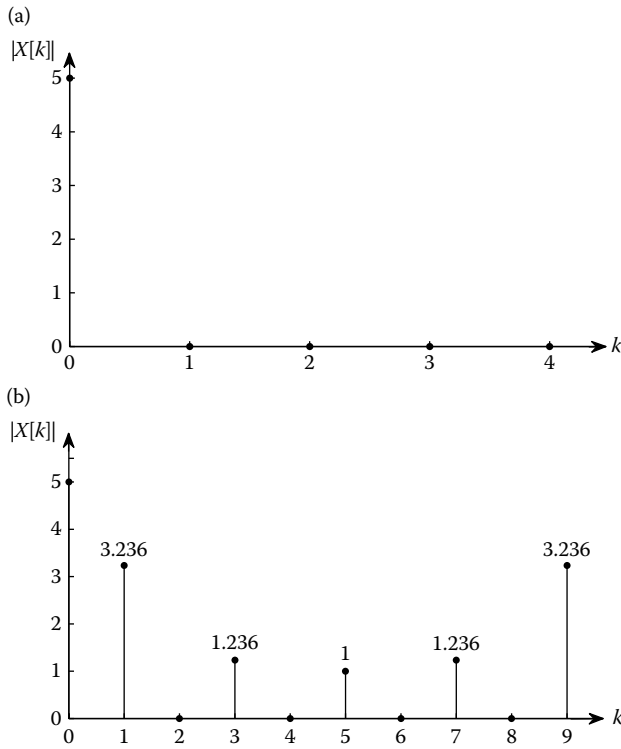
where the twiddle factor  $W_N = e^{-j(\frac{2\pi}{10})}$ . On expanding, we obtain

$$\begin{aligned} X_1[k] = & 1.W_{10}^{0k} + 0.W_{10}^{1k} + 1.W_{10}^{2k} + 0.W_{10}^{3k} + 1.W_{10}^{4k} + 0.W_{10}^{5k} + 1.W_{10}^{6k} \\ & + 0.W_{10}^{7k} + 1.W_{10}^{8k} + 0.W_{10}^{9k}. \end{aligned} \tag{1.131}$$

Hence, the alternate terms become zero. By substituting the values of  $k$ , we obtain the complete DFT sequence as

$$X_1[k] = \{5, 0, 0, 0, 0, 5, 0, 0, 0, 0\}. \tag{1.132}$$

Note that the given signal  $x_1[n]$  is actually a zero-interpolated version of  $x[n]$ . It is interesting to note that  $X_1[k]$  is a repetition of  $X[k]$  (in case 1), that is,  $X_1[k] = \{X[k], X[k]\}$ .



**FIGURE 1.13**  
 N-point DFT of a rectangular pulse. (a) Case 1:  $N = 5$ . (b) Case 2:  $N = 10$ .

2. The DFT of  $x[n]$  is  $X[k] = \{1, 2, 3, 4, 5\}$ . Find the DFT of the zero-interpolated signal  $g[n]$ , as defined in Equation 1.134: Given that

$$x[n] \xleftrightarrow{\text{DFT}} \{1, 2, 3, 4, 5\}, \tag{1.133}$$

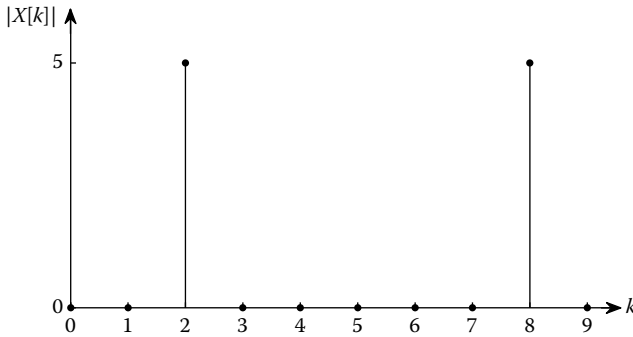
let  $g[n]$  be defined as

$$g[n] = \begin{cases} x[n/2], & \text{if } n \text{ multiple of } 2 \\ 0, & \text{otherwise} \end{cases} \tag{1.134}$$

We can deduce from the previous example that

$$g(n) \xleftrightarrow{\text{DFT}} \{1, 2, 3, 4, 5, 1, 2, 3, 4, 5\}. \tag{1.135}$$

In other words, the  $N$ -fold zero-interpolation of  $x[n]$  yields a corresponding replication in the DFT domain.

**FIGURE 1.14**

DFT of a periodic cosine sequence with length  $N = 10$ ,  $r = 2$ .

3. The DFT of a periodic cosine sequence with period  $N$  defined as

$$x[n] = \cos\left(\frac{2\pi}{N}rn\right), \quad 0 \leq n \leq N-1 \quad (1.136)$$

$$\text{can be given by, } X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad 0 \leq k \leq N-1 \quad (1.137)$$

where  $W_N = e^{-j(2\pi/N)}$  represents the twiddle factor.

$x[n]$  can be expanded in terms of the exponentials as

$$x[n] = \frac{1}{2} [W_N^{-rn} + W_N^{rn}] \quad (1.138)$$

$$X[k] = \frac{1}{2} \left[ \sum_{n=0}^{N-1} W_N^{-rn} W_N^{kn} + \sum_{n=0}^{N-1} W_N^{rn} W_N^{kn} \right] \quad (1.139)$$

$$= \frac{1}{2} \sum_{n=0}^{N-1} W_N^{(k-r)n} + \frac{1}{2} \sum_{n=0}^{N-1} W_N^{(k+r)n}. \quad (1.140)$$

Then, from the orthogonality property of the DFT, we obtain

$$X[k] = \begin{cases} \frac{N}{2}, & \text{if } k = r \\ \frac{N}{2}, & \text{if } k = N - r. \\ 0, & \text{otherwise} \end{cases} \quad (1.141)$$

The plot of  $X[k]$  is shown in Figure 1.14.

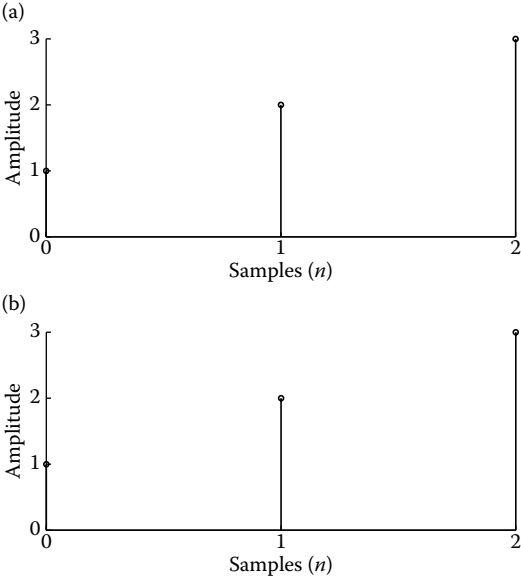
4. Illustration of the comparison between linear and circular convolution.

Figure 1.15 shows two signals  $x[n]$  and  $h[n]$  of length 3 each. The linear convolution can be performed as follows:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k]. \tag{1.142}$$

The linear convolution can be performed just like ordinary multiplication as illustrated in Table 1.5 [5].

Find the linear and circular convolutions of two sequences  $x[n] = \{1, 2, 3\}$  and  $h[n] = \{1, 2, 3\}$ .



**FIGURE 1.15** Inputs. (a) Input  $x(n)$ . (b) Impulse response  $h(n)$ .

**TABLE 1.5**  
Linear Convolution

		1	2	3
	*	1	2	3
		3	6	9
	2	4	6	
1	2	3		
1	4	10	12	9

The result of the linear convolution is shown in Figure 1.16(a) and the sequence is

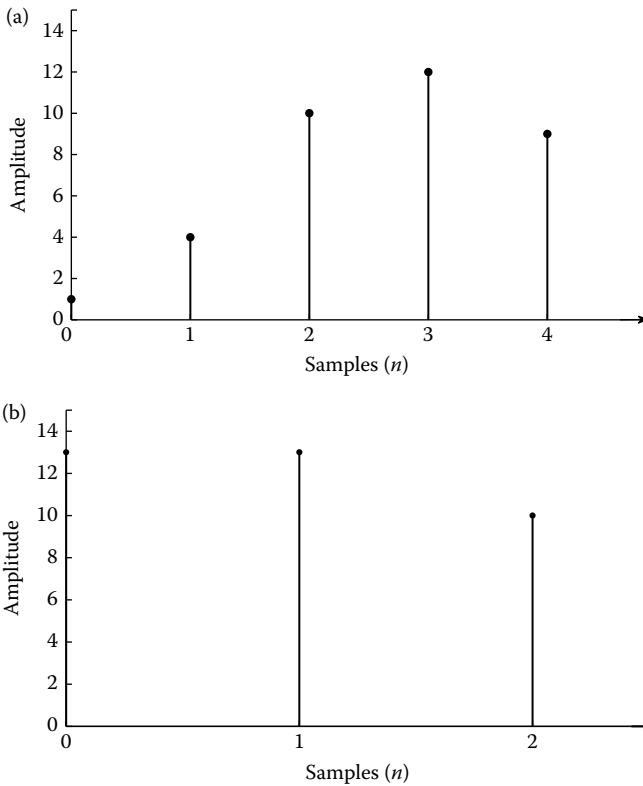
$$x[n] = \{1, 4, 10, 12, 9\}.$$

The circular convolution of the two sequences is given in Figure 1.16(b). The circular convolution can be computed by the two concentric circle approach as illustrated in Figure 1.17.

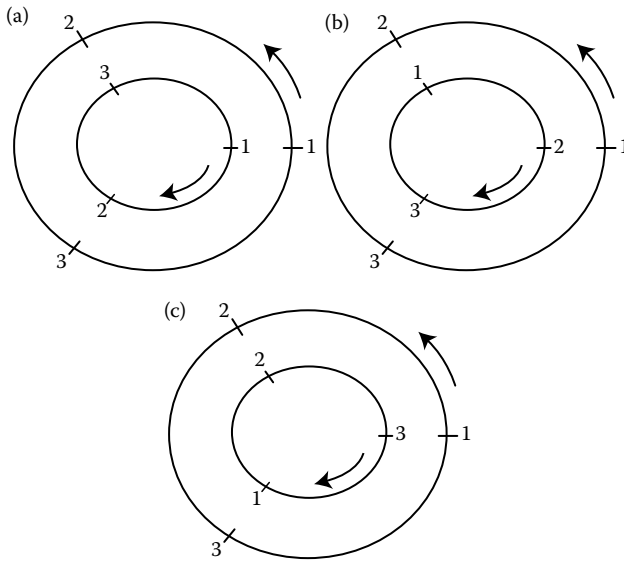
- (a) First, overlay the sequence  $x[n]$  on an outer circle in an anticlockwise direction and then overlay the sequence  $h[n]$  in an inner circle in a clockwise direction as illustrated in Figure 1.17(a). Perform point-by-point multiplication and add to give:

$$y'[0] = 1 + 6 + 6 = 13. \tag{1.143}$$

- (b) Then, rotate the inner circle in an anticlockwise direction by one sample and then again do the point-by-point multiplication as



**FIGURE 1.16** Linear and circular convolutions of  $x[n]$  and  $h[n]$ . (a) Linear convolution. (b) Circular convolution.



**FIGURE 1.17**  
Circular convolution of  $x[n]$  and  $h[n]$ . (a) Stage 1. (b) Stage 2. (c) Stage 3.

shown in Figure 1.17(b), which will give us:

$$y'[1] = 2 + 2 + 9 = 13. \tag{1.144}$$

(c) Then, once again, rotate the inner circle (anticlockwise), multiply and add the result as shown in Figure 1.17(c) to get:

$$y'[2] = 3 + 4 + 3 = 10. \tag{1.145}$$

Therefore, the circular convolution of  $x[n]$  and  $h[n]$  gives the following result:

$$y'[n] = x[n] \circledast h[n] = \{13, 13, 10\}. \tag{1.146}$$

Upon comparing  $y[n]$  and  $y'[n]$ , we can see that not only the numerical values of the convolution are different but also the length of the sequences are not as same as the linear convolution. Suppose, we augment the two sequences with zeros, such that the length of each sequence is  $(N_1 + N_2 - 1)$ , that is, in this example  $(3 + 3 - 1 = 5)$ , then each sequence can be thought as extended

sequences,  $x_e[n]$  and  $h_e[n]$ , as follows:

$$x_e[n] = \{1, 2, 3, 0, 0\} \quad (1.147)$$

$$h_e[n] = \{1, 2, 3, 0, 0\}. \quad (1.148)$$

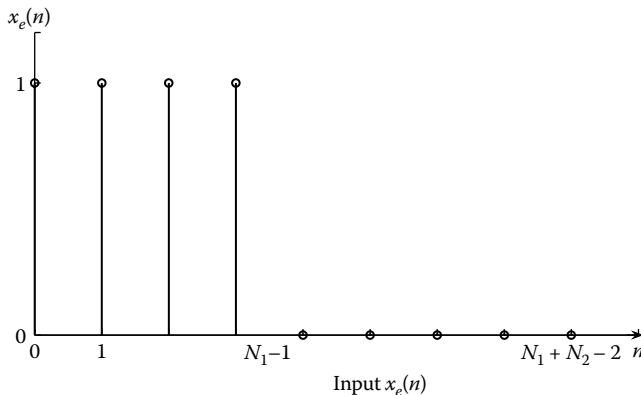
If we now perform the circular convolution operation using the two concentric circle approach discussed above, we obtain the following result:

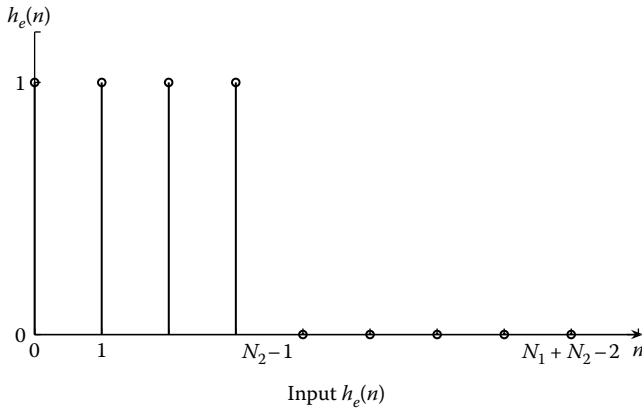
$$y[n] = \{1, 4, 10, 12, 9\}, \quad (1.149)$$

which is the same as linear convolution. The fact that circular convolution equals linear convolution is extremely important in many signal processing applications. The linear convolution of a sequence  $x[n]$  having  $N_1$  (i.e., 0 to  $(N_1 - 1)$ ) samples with a sequence of  $N_2$  samples will result in a sequence of  $(N_1 + N_2 - 1)$  samples in length. Thus, the linear convolution will have all of its nonzero values in the interval  $0 \leq n \leq (N_1 + N_2 - 2)$  points.

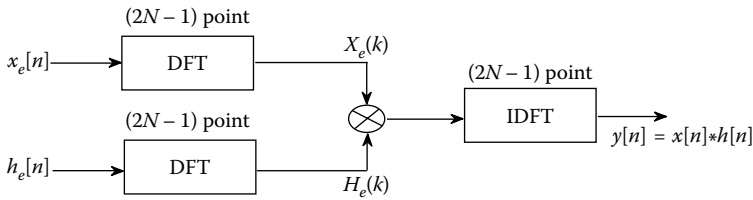
We can conclude from the above example that if a sequence of length  $N_1$  is followed by  $(N_2 - 1)$  zero-valued sequence, then the resulting sequence which has  $(N_1 + N_2 - 1)$  points can be circularly convolved with another sequence of length  $N_2$ , augmented by  $(N_1 - 1)$  zeros. The result thus obtained will be the same if we perform linear convolution. Linear convolution can be obtained via circular convolution, provided a proper choice is made for the number of points (in circular convolution). Therefore, both the sequences should be padded with zeros such that the total length of each sequence ( $x_e[n]$  and  $h_e[n]$ ) becomes  $(N_1 + N_2 - 1)$  samples. If  $N_1 = N_2 = N$ , then the total length of the sequence is  $(2N - 1)$ . We can compute the linear convolution of  $x[n]$  and  $h[n]$  via DFT as follows:

$$y[n] = x_e[n] \otimes h_e[n] = IDFT [X_e[k].H_e[k]] = DFT^{-1} [X_e[k].H_e[k]], \quad (1.150)$$





where  $X_e[k]$  and  $H_e[k]$  are the DFTs of  $x_e[n]$  and  $h_e[n]$ , respectively. Equation 1.150 can be represented in terms of a block schematic as given in Figure 1.18. The output  $y[n]$  will be the linear convolution of  $x[n]$  and  $h[n]$ . The procedure discussed above is attractive since there is an algorithm called the fast Fourier transform (FFT) which can compute the DFT much faster and it is efficient, especially if the



**FIGURE 1.18**  
Linear convolution via DFT and IDFT.

**TABLE 1.6**

Summary of Properties of DFT

Finite-Length Sequence (Length $N$ )	$N$ -Point DFT (Length $N$ )
$x[N + n] = x[n]$	$X[k + N] = X[k]$
$ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
$x[(n - m)_N]$	$e^{-j\frac{2\pi}{N}km} X[k]$
$X[n]$	$Nx[(-k)_N]$
$x^*[n]$	$X^*[(-k)_N]$
$x[(-n)_N]$	$X[-k]$
$x_1[n] \otimes x_2[n]$	$X_1[k]X_2[k]$
$x_1[n]x_2[n]$	$\frac{1}{N} \sum_{l=0}^{N-1} X_1[l]X_2[(k - l)_N]$



**TABLE 1.7**

Summary of Symmetry Properties of DFT

Finite-Length Sequence (Length $N$ )	$N$ -Point DFT (Length $N$ )
$\mathcal{R}e\{x[n]\}$	$\frac{1}{2}[X[(k)_N] + X^*[(-k)_N]]$
$j\mathcal{I}m\{x[n]\}$	$\frac{1}{2}[X[(k)_N] - X^*[(-k)_N]]$
$x_e[n] = \frac{1}{2}[x[n] + x^*[(-n)_N]]$	$\mathcal{R}e\{X[k]\}$
$x_o[n] = \frac{1}{2}[x[n] - x^*[(-n)_N]]$	$j\mathcal{I}m\{X[k]\}$
When $x[n]$ is real	$X[k] = X^*[(-k)_N]$
	$\mathcal{R}e\{X[k]\} = \mathcal{R}e\{X[(-k)_N]\}$
	$\mathcal{I}m\{X[k]\} = -\mathcal{I}m\{X[(-k)_N]\}$
	$ X[k]  =  X[(-k)_N] $
	$\angle\{X[k]\} = -\angle\{X[(-k)_N]\}$

sequence length is large. The FFT algorithm is discussed in the next section.

The properties of DFT are summarized and presented in Tables 1.6 and 1.7.

## 1.6 Fast Fourier Transform

So far, we have seen how to compute the DFT of a signal. Now we will introduce an efficient tool for the computation of the DFT termed as FFT. This can be done by exploiting the periodicity and symmetry properties of the twiddle factors  $W_N^{kn}$ , which are given as follows:

1.  $W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$ , complex conjugate property
2.  $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$ , periodicity in  $n$  and  $k$

There are several algorithms available for the efficient computation of the DFT and these have come to be known as fast Fourier transform (FFT) algorithms. All these algorithms are based on the fundamental principle of decomposing the computation of the DFT of a sequence of length  $N$  into successively smaller DFTs (known as the “divide and conquer” approach).

In this section, we will discuss in detail two such popular algorithms:

1. Decimation-in-time FFT algorithm
2. Decimation-in-frequency FFT algorithm

### 1.6.1 Decimation-in-Time FFT (DIT-FFT)

The decimation-in-time FFT (DIT-FFT) algorithm is based on decomposing the time sequence  $x[n]$  into successively smaller sub-sequences and hence its

name. Here, we consider only a radix-2 FFT algorithm, where  $N$  is a power of 2. We now explain the process of DIT-FFT in detail. The DFT of  $x[n]$  is

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk}, \quad k = 0, 1, \dots, N-1. \quad (1.151)$$

We can separate  $x[n]$  into even- and odd-numbered samples as

$$X[k] = \sum_{n=\text{even}} x[n]W_N^{nk} + \sum_{n=\text{odd}} x[n]W_N^{nk}. \quad (1.152)$$

On substituting the variables  $n = 2r$  for  $n$ -even and  $n = 2r + 1$  for  $n$ -odd, we obtain

$$X[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r]W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]W_N^{(2r+1)k} \quad (1.153)$$

$$= \sum_{r=0}^{\frac{N}{2}-1} x[2r](W_N^2)^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x[2r+1](W_N^2)^{rk}. \quad (1.154)$$

But, we can prove that  $(W_N^2)^{kn} = W_{\frac{N}{2}}^{kn}$ . By substituting this result into Equation 1.154, we get

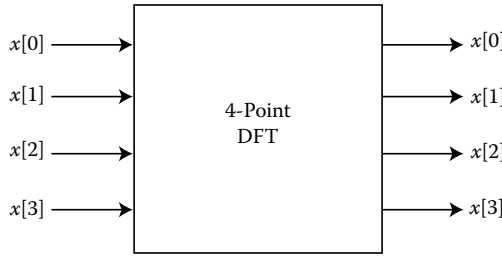
$$X[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r]W_{\frac{N}{2}}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]W_{\frac{N}{2}}^{rk} \quad (1.155)$$

$$= X_e[k] + W_N^k X_o[k], \quad k = 0, 1, \dots, N-1 \quad (1.156)$$

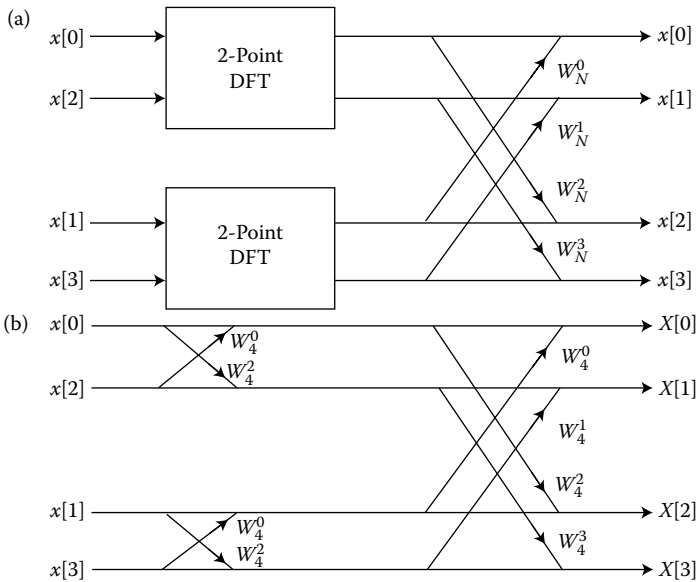
where  $X_e[k]$  and  $X_o[k]$  are  $N/2$ -point DFTs of the even-numbered and the odd-numbered samples of  $x[n]$ , respectively. This procedure of decomposing the sequence into smaller sequences can be continued, since the new  $N/2$ -point DFT blocks generated are again periodic in  $k$  with period  $N/2$ . Hence, these blocks can be further divided into two  $N/4$ -point DFT blocks (by decomposing again into even and odd parts). We proceed in this way till the blocks are reduced to two input blocks.

We will now consider an example for a sequence of length  $N = 4$ , which is shown in Figure 1.19. Let us apply the DIT-FFT algorithm by decomposing the inputs of Figure 1.19 into two  $N/2$ -point DFT computations of even- and odd-numbered samples, as shown in Figure 1.20.

For a generalized case, we can draw the elementary computation, called a butterfly, as shown in Figure 1.21(a). From the symmetry and periodicity



**FIGURE 1.19**  
Example of a 4-point DFT.

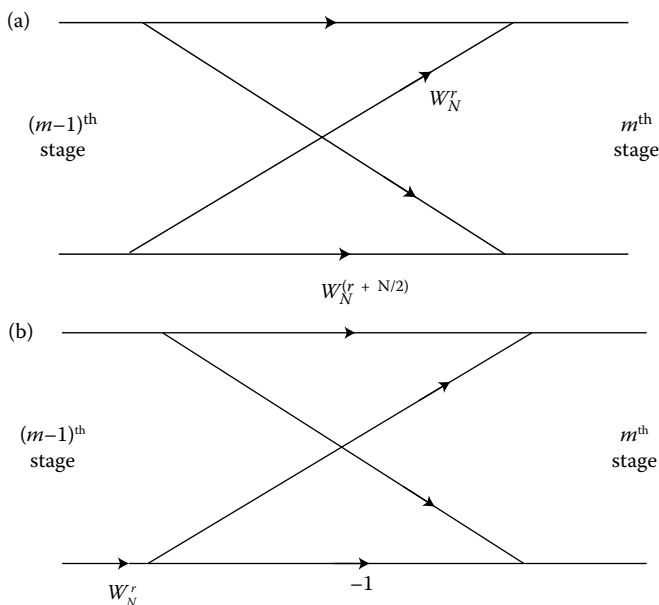


**FIGURE 1.20**  
Decomposition of a 4-point FFT into two 2-point FFT blocks. (a) Block diagram. (b) Butterfly structure.

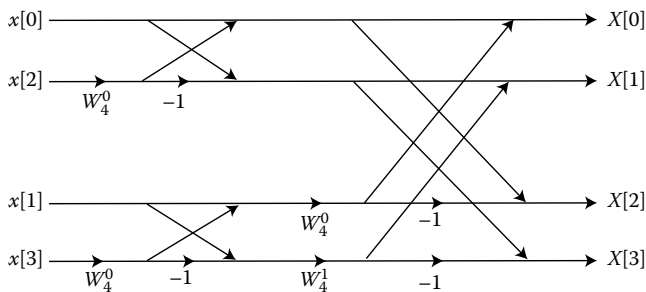
properties of  $W_N^r$ , we can deduce that

$$W_N^{r+\frac{N}{2}} = W_N^r W_N^{\frac{N}{2}} = -W_N^r. \tag{1.157}$$

In particular, the number of complex multiplications has been reduced by half when compared to the number presented in Figure 1.21(a). With this observation, the butterfly computation of Figure 1.21(a) can be simplified to a form shown in Figure 1.21(b), which requires only one complex multiplication instead of two (see Equation 1.157). The basic signal flow graph of Figure 1.21(b) is an efficient replacement for the butterflies of the form of



**FIGURE 1.21** Butterfly structures. (a) Butterfly structure. (b) Efficient butterfly structure.



**FIGURE 1.22** Complete butterfly diagram for a 4-point FFT.

Figure 1.21(a). We can obtain the signal flow graph of Figure 1.22 from Figure 1.20(b) and 1.21(b). Now using this, we complete the butterfly diagram for a 4-point DFT as shown in Figure 1.22.

### 1.6.1.1 Computational Savings

Let us now examine the computational savings provided by the FFT technique. In the case of an  $N$ -point DFT as given by Equation 1.93, the

computation of each term requires the sum of  $N$  products. Hence, for the computation of  $N$ -point DFT, we require  $N^2$  complex multiplications and  $N(N - 1)$  complex additions.

In the case of  $N$ -point FFT, there are  $r = \log_2 N$  stages. Each stage requires  $N/2$  complex multiplications by twiddle factors and  $N$  complex additions. Hence, the total number of complex multiplications is of the order of  $(N/2 \log_2 N)$  and the total number of complex additions is of the order of  $(N \log_2 N)$ .

**1.6.1.2 In-Place Computation**

In the case of DFT computations using FFT, the complex coefficients are stored in memory and read out for multiplication when required. Similarly, the intermediate stage results also need to be stored for the computation of the output of the next stage. An intermediate stage is shown in Figure 1.23.

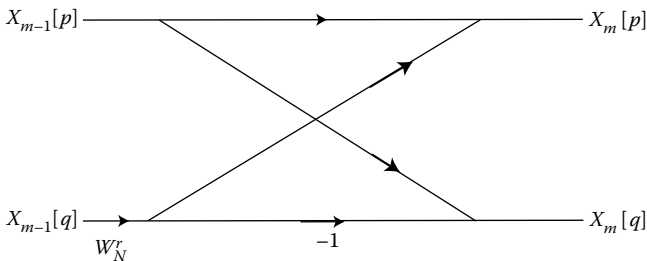
$$X_m[p] = X_{m-1}[p] + W_N^r X_{m-1}[q] \tag{1.158}$$

$$X_m[q] = X_{m-1}[p] - W_N^r X_{m-1}[q] \tag{1.159}$$

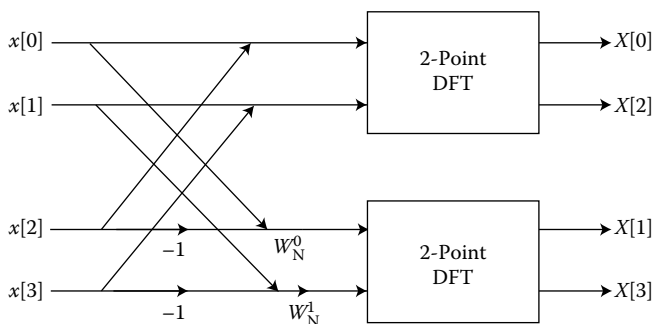
However, the intermediate results  $X_{m-1}[p]$  and  $X_{m-1}[q]$  are used only for the computation of the next stage  $X_m[p]$  and  $X_m[q]$ , and they are never used later. Hence, instead of using new memory locations for  $X_m[p]$  and  $X_m[q]$ , we can keep these results in place of  $X_{m-1}[p]$  and  $X_{m-1}[q]$ , and thus the memory can be saved. This is called *in-place* computation.

**1.6.2 Decimation-in-Frequency FFT (DIF-FFT)**

In the previous subsection, for the case of DIT-FFT algorithm, we have divided the input sequence  $x[n]$  into smaller sequences. Now in the case of decimation-in-frequency (DIF-FFT) algorithm, we decompose the output sequence  $X[k]$  into smaller subsequences in an analogous manner and hence its name.



**FIGURE 1.23**  
 $r$ th Stage butterfly.



**FIGURE 1.24**  
4-point FFT using DIF-FFT.

Here, we consider computing the even- and odd-numbered frequency samples separately as follows:

$$X[2r] = \sum_{n=0}^{\frac{N}{2}-1} \left[ x[n] + x \left[ n + \frac{N}{2} \right] \right] W_{\frac{N}{2}}^{rn}, \quad r = 0, 1, \dots, \frac{N}{2} - 1 \quad (1.160)$$

$$X[2r + 1] = \sum_{n=0}^{\frac{N}{2}-1} \left[ x[n] - x \left[ n + \frac{N}{2} \right] \right] W_N^{(2r+1)n}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left[ x[n] - x \left[ n + \frac{N}{2} \right] \right] W_N^n W_{\frac{N}{2}}^{rn}, \quad r = 0, 1, \dots, \frac{N}{2} - 1. \quad (1.161)$$

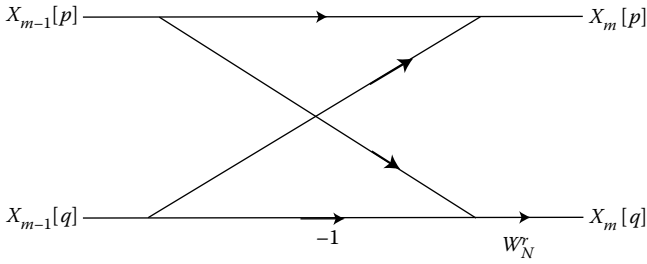
Hence, we can decompose the output sequence as shown in Figure 1.24 for a 4-point case. Now we can proceed on similar lines by dividing the output sequence into progressively smaller blocks (just as was done in the case of DIT-FFT for the input sequence,  $x[n]$ ). The elementary butterfly structure for DIF-FFT is presented in Figure 1.25.

The concept of in-place computation can be used in the DIF-FFT as well, since

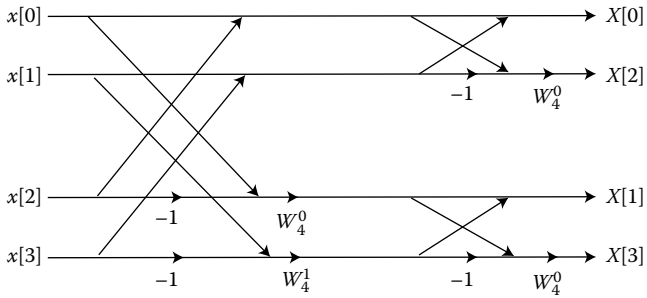
$$X_m[p] = X_{m-1}[p] + X_{m-1}[q] \quad (1.162)$$

$$X_m[q] = [X_{m-1}[p] - X_{m-1}[q]]W_N^r. \quad (1.163)$$

The complete butterfly diagram for a 4-point DIF-FFT is given in Figure 1.26. Now on comparing the 4-point butterfly structures for DIT-FFT and DIF-FFT, we can observe that in DIT-FFT, the inputs were given in the bit-reversed order; whereas for the latter, the outputs appear in the bit-reversed order.



**FIGURE 1.25**  
 mth Stage butterfly of DIF-FFT.



**FIGURE 1.26**  
 Flow graph of a 4-point DIF-FFT.

We can also observe that DIT-FFT and DIF-FFT flowgraphs are transposes of each other (see Figures 1.22 and 1.26).

**1.6.3 Inverse DFT from FFT**

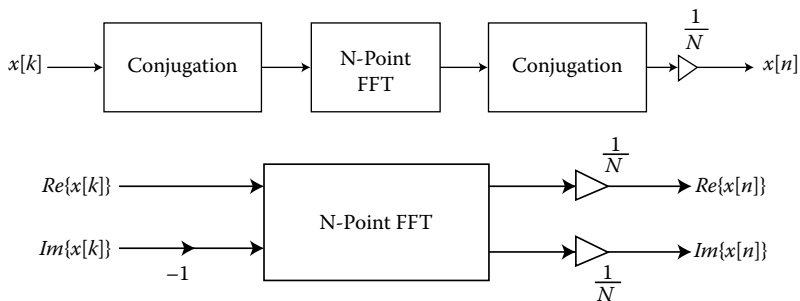
The expression for the DFT and the inverse DFT (IDFT) are reproduced below (as discussed in Section 1.5):

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad k = 0, 1, \dots, N - 1 \tag{1.164}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-kn}, \quad n = 0, 1, \dots, N - 1 \tag{1.165}$$

where  $W_N = e^{-j(2\pi/N)}$  is called the twiddle factor.

We note that in both Equations 1.164 and 1.165, the expressions differ only in the sign of the power of  $W_N$  and the presence of a scale factor  $1/N$  in the



**FIGURE 1.27** Block diagram for inverse DFT using FFT. (a) Inverse DFT. (b) Simplified inverse DFT.

IDFT expression. Thus, it is possible to compute the inverse DFT using the forward FFT technique with some modifications of the input and output.

From Equation 1.165, we can write

$$x^*[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^*[k] W_N^{kn}, \quad n = 0, 1, \dots, N - 1 \quad (1.166)$$

$$Nx^*[n] = \sum_{k=0}^{N-1} X^*[k] W_N^{kn} \quad (1.167)$$

$$= DFT\{X^*[k]\}. \quad (1.168)$$

Therefore, we can recover  $x[n]$  as follows:

$$x[n] = \frac{1}{N} [DFT\{X^*[k]\}]^*, \quad n = 0, 1, \dots, N - 1. \quad (1.169)$$

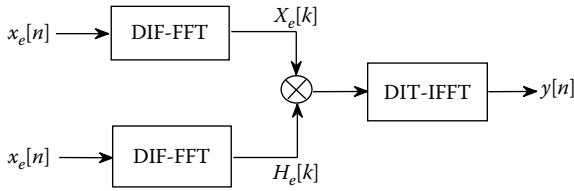
The basic block diagrams are given in Figure 1.27.

### 1.6.4 Linear Convolution Using DIT-FFT and DIF-FFT

Since FFT computes the DFT faster, we can use the FFT algorithm to compute the linear convolution. It has been shown earlier that the input to the DIT-FFT is in bit-reversed order, while the output is in natural order. However, in the case of DIF-FFT, it is just the reverse, that is, the input will be in natural order while the output will be in bit-reversed order. We can use these two algorithms to compute the linear convolution very efficiently as detailed below.

Since the DIF-FFT takes the input in natural order and gives the output in bit-reversed order, this scheme can be used to compute the forward DFTs of the two sequences. Since the outputs  $X_e[k]$  and  $H_e[k]$  are in bit-reversed order, we use DIT-IFFT to compute the inverse DFT after the point-by-point





**FIGURE 1.28**  
Linear convolution via. DIF-FFT and DIT-IFFT.

multiplication. This is illustrated in Figure 1.28. Therefore, if we use the above scheme, we need not do any bit reversal in computing the linear convolution via the FFT approach. Here,  $x_e[n]$  and  $h_e[n]$  represent the extended sequences, since we will have to pad the original sequences  $x[n]$  and  $h[n]$  with appropriate number of zeros, in order to obtain linear convolution from circular convolution.

In this chapter, we have considered different Fourier analysis methods, their properties and applications. We have considered the digital implementation of Fourier transform in detail.

---

## References

1. A.V. Oppenheim, A.S. Willsky, and S.H. Nawab, *Signals and Systems*, 2nd Edn., Prentice-Hall, Upper Saddle River, NJ, 1997.
2. A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, 3rd Edn., Prentice-Hall, Upper Saddle River, NJ, 2010.
3. S.S. Soliman and M.D. Srinath, *Continuous and Discrete Signals and Systems*, 2nd Edn., Prentice-Hall, India, 2004.
4. S.K. Mitra, *Digital Signal Processing: A Computer Based Approach*, 4th Edn., McGraw-Hill, New York, NY, 2011.
5. J.W. Pierre, A novel method for calculating the convolution sum of two finite length sequences, *IEEE Transactions on Education*, vol. 39, no. 1, pp. 77–80, February 1996.

# 2

---

## *Pitfalls in the Computation of DFT*

---

In this chapter, we turn our attention to the issues in spectral estimation and FIR filter design. Quite often, the DFT is used as an approximation to the CTFT. However, we should be aware of the fact that there are several pitfalls associated with this approximation. The discrepancies between the DFT and the CTFT arise because DFT requires sampling and truncation. Improper sampling gives rise to “aliasing errors.” Aliasing refers to the distortion of the signal spectrum due to the introduction of spurious low-frequency components owing to a combination of a very low sampling rate and an improper anti-aliasing filter. Section 2.1 briefly describes sampling, reconstruction, and the associated distortion called aliasing.

The other two types of pitfalls arise due to the truncation of the data while applying the DFT. These distortions are called (i) frequency leakage (or just leakage) and (ii) picket-fence effect (or scalloping loss). These are discussed in detail in Sections 2.2 and 2.4, respectively. However, before we discuss the picket-fence effect, we will introduce the representation of the DFT as a bank of bandpass filters, in Section 2.3. DFT resolution, zero-padding, and frequency-domain sampling are discussed in Section 2.5.

---

### **2.1 Sampling, Reconstruction, and Aliasing**

The concepts of sampling, reconstruction, and aliasing can be demonstrated by means of a familiar example. While capturing motion pictures, the camera converts the dynamic scene into a sequence of frames. These frames are usually taken at regular time intervals of 24 frames/second. The frame rate has been chosen by taking into account the persistence of vision of the human eye. Sampling essentially selects a set of finite data points as a representation of the continuous-time signal at the corresponding time duration. Movie frames thus take samples of the scene information during each second. When it is played, our eyes and brain fill the missing data between the frames and thus provide the illusion of a continuously varying video. This operation of filling the breaks between the sampled data points is called reconstruction.

In general terms, reconstruction converts a sampled sequence back into a continuous-time signal. It generates an infinite amount of data from the samples. In the motion picture example, the reconstructed signal is a hallucination of our brain. What are the problems associated with sampling? Naturally, one cannot expect the reconstructed signal to be an exact copy of the original. Sampling often yields a type of distortion that is called as aliasing. Let us explain this distortion by an example. Imagine the scene of a clockwise rotating wheel. As long as its speed of rotation is less than half the number of frames/second, we perceive it correctly. However, when the speed increases beyond this value, the wheel actually appears to rotate anti-clockwise, that too at a reduced speed! Its apparent speed is now the number of frames/second minus its real speed. Another observation we can make is that, when the speed is exactly equal to the number of frames/second, the wheel is seemingly stationary. This occurs because the wheel is now sampled at an identical position. Now, if the speed is increased further, the wheel apparently rotates clockwise again, but at a reduced speed. To generalize, the wheel always appears to rotate at a speed not higher than half the number of frames/second (in either direction).

Sampling is the basis of DSP and, hence, a thorough understanding of sampling is necessary for practical applications. Minimizing the phenomenon of aliasing is one of the vital problems. Engineering applications often provide the continuous-time signal in the form of a voltage wave, and sampling is carried out using electronic circuitry. Reconstruction is also performed in a similar manner. Further, the distortion of the sequence is caused by the physical limitations of electronic circuitry.

*History of sampling theorem:* The sampling theorem is usually attributed to Shannon who introduced it in the field of information theory in 1949 [1]. However, a Russian scientist named Kotelnikov had found it independently around the same time. The credit for first discovering the theorem and its importance should be given to E.T. Whittaker who published a remarkable paper [2] on the sampling theorem in 1915, wherein he also discloses a formula for reconstructing the waveform from its samples. This laid the foundation for modern digital signal processing. We thereby refer to sampling as the Whittaker Kotelnikov Shannon or **WKS sampling theorem**, using the first letters of all three surnames.

There are several ways to sample an analog or a continuous-time signal,  $x_c(t)$ . We will consider the most general method called periodic or uniform sampling. This is described by the following relationship:

$$x_c(t)|_{t=nT_s} = x_c(nT_s) = x[n], \quad (2.1)$$

where  $x[n]$  represents the uniformly sampled discrete-time signal obtained after sampling a continuous-time signal  $x_c(t)$  and  $T_s$  represents the sampling period or sampling time in seconds. Here,  $F_s = \frac{1}{T_s}$  is called the sampling

rate (in samples per second) or the sampling frequency, expressed in Hertz (Hz). Therefore, the uniform sampling establishes a relationship between the time variables  $t$  and  $n$  of the continuous-time and discrete-time signals, respectively, as follows:

$$t = nT_s = \frac{n}{F_s}. \quad (2.2)$$

In view of Equation 2.2, there exists a linear relationship between the continuous-time frequency variable  $F_s$  (or  $\Omega_s = 2\pi F_s$ , where the unit of  $\Omega_s$  is radians/second) in Hertz and the discrete-time frequency variable  $f$  (or  $\omega$ , where the unit of  $\omega$  is radians/sample) is dimensionless. This relation can be established by considering a continuous-time signal given by

$$x_c(t) = \sin(\Omega t + \theta) = \sin(2\pi F t + \theta). \quad (2.3)$$

If  $x_c(t)$  is sampled uniformly at a rate of  $F_s = \frac{1}{T_s}$  samples/second, we obtain

$$\begin{aligned} x_c(t)|_{t=nT_s} &= x_c(nT_s) \equiv x[n] \\ &= \sin(\Omega t + \theta)|_{t=nT_s} = \sin(2\pi F n T_s + \theta) \\ &= \sin\left(\frac{2\pi n F}{F_s} + \theta\right). \end{aligned} \quad (2.4)$$

A discrete-time sinusoidal signal can be represented as

$$x[n] = \sin[\omega n + \theta] = \sin[2\pi f n + \theta], \quad (2.5)$$

where  $n$  is an integer variable called the sample number,  $\theta$  is the phase in radians, and  $\omega$  is the frequency in radians/sample (or if we use  $f$ , then in cycles/sample).

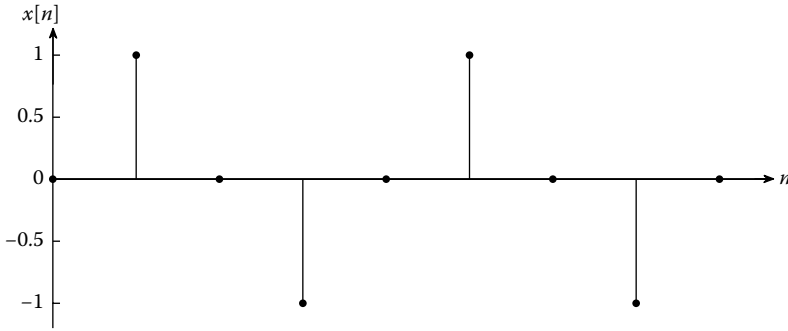
Now, if we compare Equations 2.4 and 2.5, we note that the frequency variables  $F$  and  $f$  are related by

$$f = \frac{F}{F_s}, \quad (2.6)$$

or, equivalently,  $\Omega$  and  $\omega$  are related by the following:

$$\omega = \Omega T_s. \quad (2.7)$$

Therefore, the relationship given in Equation 2.6 justifies the name relative or normalized frequency, which is sometimes used to describe the frequency variable  $f$ . As implied by Equation 2.6, we can use  $f$  to determine the frequency  $F$  (in hertz) only if the sampling frequency  $F_s$  is known.



**FIGURE 2.1**  
Sampled data.

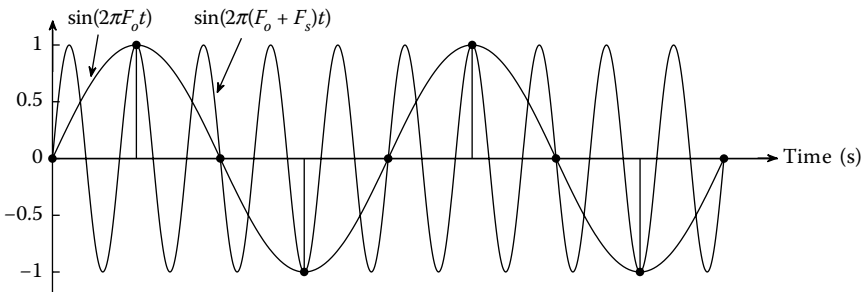
Now, let us consider a continuous-time signal:

$$x_c(t) = \sin(2\pi F_o t), \quad -\infty < t < \infty.$$

If this signal is sampled at a rate of  $F_s$  (or  $\frac{1}{T_s}$ ) samples/second, the resulting discrete-time sequence,  $x[n]$  can be represented as in Figure 2.1. Now, if we are asked to reconstruct the signal from these samples, we may end up with many possibilities. Two of such representations are given in Figure 2.2. In this figure, the frequencies of the two sinusoids are  $(F_o + F_s)$  and  $F_o$ . These two sinusoids share the same samples since, by expansion, we obtain

$$\begin{aligned} \sin[2\pi(F_o + F_s)nT_s] &= \sin[2\pi F_o nT_s + 2\pi F_s nT_s] \\ &= \sin[2\pi F_o nT_s + 2\pi n] \\ &= \sin[2\pi F_o nT_s]. \end{aligned}$$

In fact, any sinusoid with frequency  $(F_o + lF_s)$ , where  $l$  is an integer, fits exactly into these sampled values. Thus, an infinite number of continuous-time



**FIGURE 2.2**  
Possible signal reconstructions.

sinusoids can be obtained from the same set of discrete-time signal samples. Equivalently, we can say that the frequencies  $F_k = (F_o + lF_s)$ ,  $-\infty < l < \infty$ , are indistinguishable from the frequency  $F_o$  after sampling and hence they are the aliases of  $F_o$ . Hence, sampling a continuous-time signal introduces a distortion called *aliasing*. This uncertainty arises because the sinusoid with frequency  $(F_o + F_s)$  does not have enough number of samples per period. Therefore, to uniquely reconstruct the signal the sampling frequency should be selected such that the sinusoid with maximum frequency (denoted as  $F_{\max}$ ) has at least two samples per period. In other words, if the sampling period  $T_s < \frac{T_o}{2}$ , where  $T_o$  is the time period of the signal, then we must choose the sampling frequency  $F_s > 2F_{\max}$ . This ensures that we can reconstruct the original signal without aliasing.

### 2.1.1 WKS Sampling Theorem

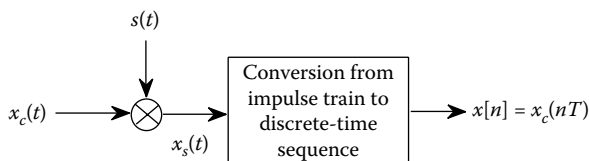
Further insight can be gained by representing Equation 2.1 as a two-stage process, as depicted in Figure 2.3 [3]. The first stage is the modulation process, where the continuous-time signal  $x_c(t)$  is multiplied with the impulse train  $s(t)$  given by

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT). \quad (2.8)$$

The output of the modulator (or multiplier) is

$$\begin{aligned} x_s(t) &= x_c(t)s(t) \\ &= x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT). \end{aligned} \quad (2.9)$$

The conversion from impulse train  $x_s(t)$  to discrete-time sequence  $x[n]$  is given in Figure 2.3 only to make the process complete [3]. In practice, the entire system is replaced with an analog-to-digital converter (ADC). Since  $x_s(t)$  is the product of  $x_c(t)$  and  $s(t)$ , the Fourier transform of  $x_s(t)$  is the convolution of their Fourier transforms,  $X_c(j\Omega)$  and  $S(j\Omega)$ , respectively. The Fourier



**FIGURE 2.3**  
C/D converter.

transform of  $s(t)$  is again an impulse train given by

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s). \quad (2.10)$$

where  $\Omega_s$  is the sampling frequency.

Therefore, using Equations 2.9 and 2.10, we get

$$\begin{aligned} X_s(j\Omega) &= \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) \\ &= \frac{1}{2\pi} X_c(j\Omega) * \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \\ &= \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)). \end{aligned} \quad (2.11)$$

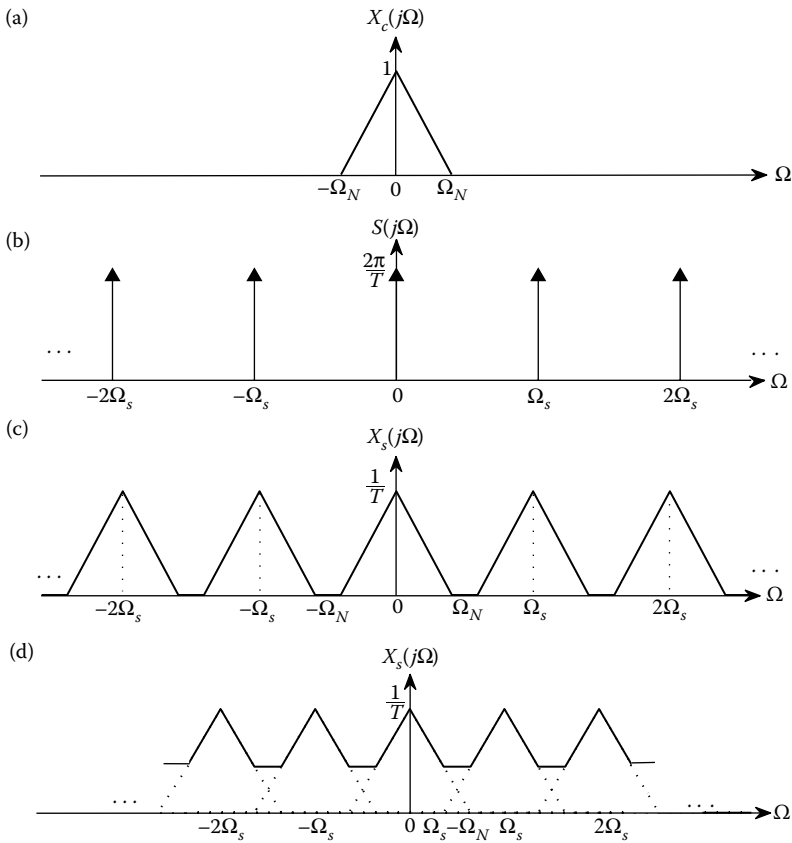
We note that Equation 2.11 gives the mathematical relationship between the Fourier transform of a continuous-time signal  $x_c(t)$  and the Fourier transform of the sampled signal  $x_s(t)$ . If we assume that  $X_c(j\Omega)$  is a lowpass signal band-limited to  $\Omega_N$  rad/sec, then  $X_s(j\Omega)$  contains the replicas of  $X_c(j\Omega)$  placed at integer multiples of the sampling frequency  $\Omega_s$ . The entire operation is illustrated in Figure 2.4, where  $X_c(j\Omega)$ ,  $S(j\Omega)$ , and  $X_s(j\Omega)$  are shown in Figures 2.4(a), (b), and (c), respectively. From Figure 2.4(c), we can observe that the replicas of  $X_c(j\Omega)$  are placed at  $\dots, -2\Omega_s, -\Omega_s, 0, \Omega_s, 2\Omega_s, \dots$ . These replicas do not overlap if  $(\Omega_s - \Omega_N) > \Omega_N$  or equivalently,  $(\Omega_s > 2\Omega_N)$ . In such a case, the continuous-time signal can be exactly reconstructed without any loss of information using an ideal lowpass filter, as depicted in Figure 2.5. The cutoff frequency of such a lowpass filter ( $\Omega_c$ ) should be carefully selected such that

$$\Omega_N < \Omega_c < (\Omega_s - \Omega_N). \quad (2.12)$$

On the other hand, if the sampling frequency is selected such that  $\Omega_s < 2\Omega_N$ , then the replicas of  $X_c(j\Omega)$  overlap as shown in Figure 2.4(d), thereby producing a distortion called aliasing. Here, the high-frequency components get folded into the lower frequencies of other replicas. In this case, the continuous-time signal cannot be reconstructed completely. To avoid this, we should ensure that  $x_c(t)$  is band-limited and is sampled with a frequency  $\Omega_s > 2\Omega_N$ . Here,  $\Omega_N$  is called the Nyquist frequency, while  $2\Omega_N$  is referred to as the Nyquist rate. The above discussion leads us to the Nyquist–Shannon sampling theorem that is stated as follows.

### Nyquist–Shannon Sampling Theorem

If  $X_c(j\Omega)$ , the CTFT of  $x_c(t)$ , is band-limited, that is,  $X_c(j\Omega) = 0$  for  $|\Omega| > \Omega_N$ , then  $x_c(t)$  can be exactly reconstructed from its samples  $x[n] = x_c(nT)$ ,  $n = 0, \pm 1, \pm 2, \dots$ , if we choose  $\Omega_s = \frac{2\pi}{T} > 2\Omega_N$ .



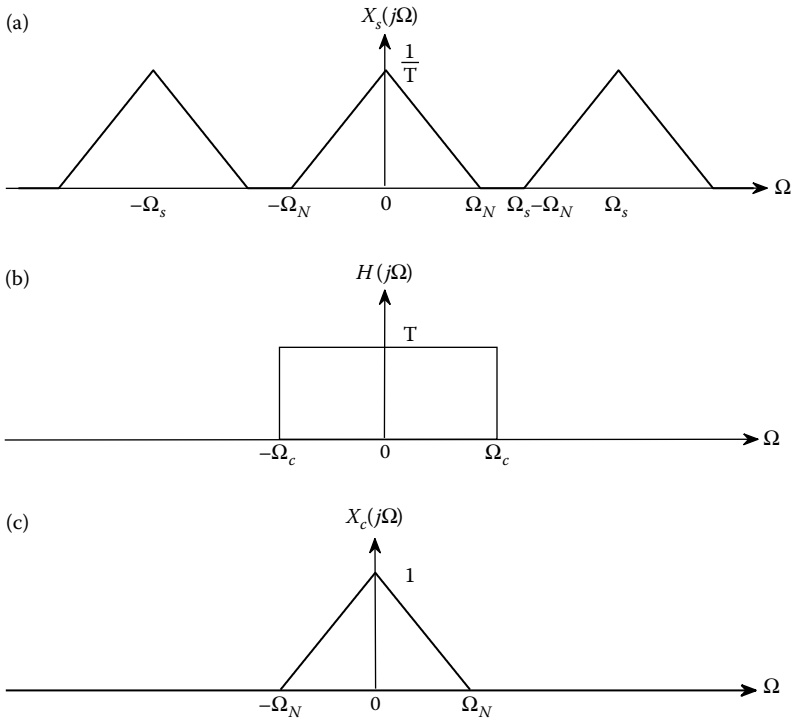
**FIGURE 2.4** (a) FT of  $x_c(t)$ , (b) FT of the impulse train, and (c) and (d) FTs of  $x_s(t)$  with sampling frequencies  $\Omega_s > 2\Omega_N$  and  $\Omega_s < 2\Omega_N$ , respectively.

**Relationship between the Input and Output of C/D Converter**

To obtain the relation between  $X_s(j\Omega)$ ,  $X_c(j\Omega)$ , and  $X(e^{j\Omega T})$  (which is the DTFT of  $x[n]$ ), we start by taking the CTFT of  $x_s(t)$  as follows [3]. From the earlier discussion, we can write

$$\begin{aligned}
 X_s(j\Omega) &= \int_{t=-\infty}^{\infty} x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) e^{-j\Omega t} dt \\
 &= \int_{t=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_c(t) e^{-j\Omega t} \delta(t - nT) dt.
 \end{aligned}
 \tag{2.13}$$





**FIGURE 2.5**

(a) FT of sampled signal, (b) frequency response of an ideal lowpass filter, and (c) frequency response of the reconstructed signal.

From the sifting theorem, we have

$$\int_{t=-\infty}^{\infty} x(t)\delta(t - \tau)d\tau = x(\tau). \tag{2.14}$$

If we interchange the integration and summation in Equation 2.13 and if we use the sifting theorem, we get

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-j\Omega nT}. \tag{2.15}$$

Note that the summation and integration in Equation 2.13 can be interchanged only if the infinite summation converges uniformly for all values of  $t$  [3].

We know that the DTFT of a sequence  $x[n]$  is defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \tag{2.16}$$

We can relate  $X(e^{j\omega})$  and  $X_s(j\Omega)$  using Equations 2.15 and 2.16 as

$$X_s(j\Omega) = X(e^{j\omega})|_{\omega=\Omega T} = X(e^{j\Omega T}). \quad (2.17)$$

Therefore, from Equations 2.11 and 2.17, we can relate  $X(e^{j\omega})$  and  $X_c(j\Omega)$  as follows:

$$X(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)) \quad (2.18)$$

or equivalently

$$X(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c \left( j \left( \frac{\omega}{T} - \frac{2\pi k}{T} \right) \right). \quad (2.19)$$

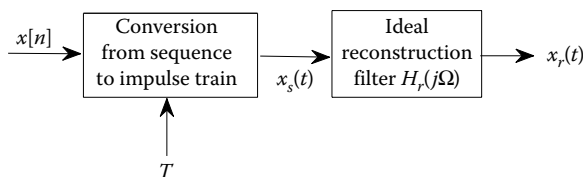
From Equation 2.19, we can observe that  $X(e^{j\omega})$  is the frequency-scaled version of  $X_s(j\Omega)$ . This is because all the samples in the discrete-time signal are spaced by unity, irrespective of the sampling period  $T$ . The discrete-time signal can be obtained by time-scaling the impulse-modulated signal  $x_s(t)$  by  $T$ . Consequently, the frequency axis is scaled by a factor of  $(\frac{1}{T})$ .

### 2.1.2 Reconstruction of Continuous-Time Signals from Discrete-Time Samples

When we generate the discrete-time signal from the impulse-modulated signal  $x_s(t)$ , the implicit time period information present in the signal is lost. Therefore, we need both the discrete-time sequence  $x[n]$  and the sampling frequency  $F_s$  for the reconstruction of the continuous-time signal from its samples. The reconstruction process is depicted as a two-stage process as shown in Figure 2.6 [3].

The first step is the conversion of the sequence to an impulse train by using the information of the sampling period  $T$ . This process can be mathematically represented as in Equation 2.20 below:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT). \quad (2.20)$$



**FIGURE 2.6**  
Reconstruction of  $x_c(t)$ .

If  $x_c(t)$  is band-limited and sampled with a frequency greater than the Nyquist rate, then the continuous-time signal  $x_c(t)$  can be exactly reconstructed without any loss of information by passing  $x_s(t)$  through an ideal lowpass filter with frequency response  $H_r(j\Omega)$ . The cutoff frequency  $\Omega_c$  of the lowpass filter should be selected such that  $(\Omega_N < \Omega_c \leq \Omega_s - \Omega_N)$ . The impulse response of the ideal lowpass reconstruction filter with the cutoff frequency  $\frac{\pi}{T}$  is given by

$$h_r(t) = \frac{\sin(\pi t/T)}{(\pi t/T)}. \quad (2.21)$$

We shall see the characteristics of  $h_r(t)$  given by Equation 2.21. First, we note that

$$h_r(0) = 1. \quad (2.22)$$

This directly follows from the small angle approximation. Second

$$h_r(nT) = 0, \quad \text{for } n = \pm 1, \pm 2, \dots \quad (2.23)$$

The output of the reconstruction filter will be the convolution of  $x_s(t)$  with  $h_r(t)$ . This is given by the following relationship:

$$\begin{aligned} x_r(t) &= \int_{\tau=-\infty}^{\infty} x_s(\tau)h_r(t - \tau)d\tau \\ &= \int_{\tau=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\delta(\tau - nT)h_r(t - \tau)d\tau. \end{aligned} \quad (2.24)$$

Interchanging the order of integration and summation and using the sifting theorem given in Equation 2.14 in the above equation (Equation 2.24), we obtain

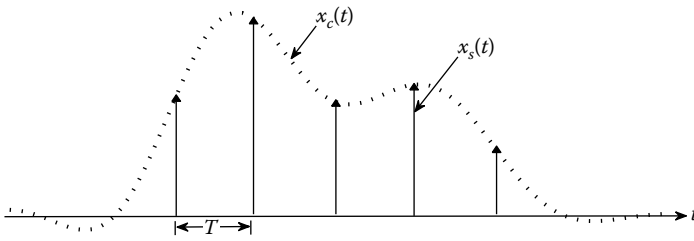
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT). \quad (2.25)$$

Substituting  $h_r(t)$  (see Equation 2.21) in Equation 2.25, we get

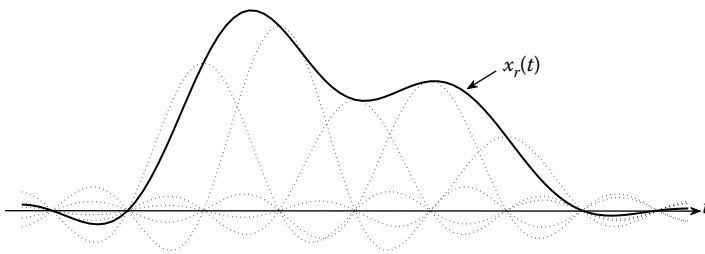
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin(\pi(t - nT)/T)}{\pi(t - nT)/T}. \quad (2.26)$$

The filtering process is very obvious in the frequency-domain, where the filter allows exactly one replica as shown in Figure 2.5(c). In the time-domain, the same process can be explained in terms of sinc interpolation, as illustrated below.

Figure 2.7 shows an impulse train obtained by assigning the strength of impulse at  $nT$  to  $n^{\text{th}}$  discrete sample (shown as dotted lines). This signal is convolved with the impulse response of an ideal lowpass filter (sinc function)



**FIGURE 2.7**  
Impulse train-modulated signal.



**FIGURE 2.8**  
Reconstruction of continuous-time signal.

to obtain the continuous-time signal. The impulse response of the ideal low-pass filter is shifted by integer multiples of  $T$  and scaled by the corresponding strengths of the impulses in  $x_s(t)$ . This is shown in Figure 2.8. From this plot, we can observe that exactly one sinc function contributes to the interpolated continuous-time signal at the sampling points.

The Fourier transform of the reconstructed signal  $x_r(t)$  is related to the transformed input signal  $x_c(t)$  as

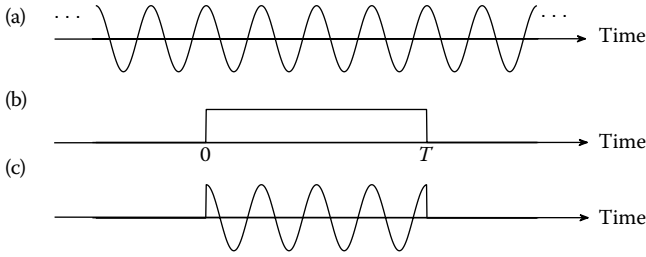
$$X_r(j\Omega) = X_c(j\Omega)H_r(j\Omega). \tag{2.27}$$

If the reconstruction filter is ideal and the cutoff frequency is selected appropriately, then the reconstructed signal,  $X_r(j\Omega)$ , will be the same as the input signal,  $X_c(j\Omega)$ .

## 2.2 Frequency Leakage Effect

The effect of frequency leakage can be illustrated using a continuous-time cosine signal given by

$$x(t) = \cos(\Omega_0 t), \quad -\infty < t < \infty.$$



**FIGURE 2.9** Rectangular data window implied when a finite record of data is analyzed. (a) Signal, (b) rectangular window, and (c) finite-length record.

This is sketched in Figure 2.9a. The CTFT of this signal is given by

$$X(j\Omega) = \pi [\delta(\Omega + \Omega_0) + \delta(\Omega - \Omega_0)]. \tag{2.28}$$

The CTFT of this signal can be considered as two impulses located at  $\pm\Omega_0$  and weighted by  $\pi$ , as shown in Figure 2.10(a). While computing the finite Fourier transform, we are forced to take only a finite number of data samples covering a time duration of  $T$  seconds and neglect everything that has happened before (and after) this period. In effect, the infinitely ranged signal  $x(t)$  is multiplied by a rectangular window  $w(t) = 1, 0 \leq t \leq T$  (shown in Figure 2.9(b)) to obtain

$$x_w(t) = x(t)w(t) = \cos(\Omega_0 t), \quad 0 \leq t \leq T. \tag{2.29}$$

The resulting signal  $x_w(t)$  is shown in Figure 2.9(c). Multiplication in the time-domain is equivalent to the convolution in the frequency-domain. Hence, the finite Fourier transform of any finite record of data is equivalent to convolving the CTFT of the actual signal with the CTFT of the rectangular window. The transform of a rectangular window is given by

$$W(j\Omega) = T \operatorname{sinc} \left( \frac{\Omega T}{2\pi} \right) e^{-\frac{j\Omega T}{2}}, \quad -\infty < \Omega < \infty. \tag{2.30}$$

Thus, the CTFT of an infinitely ranged pure cosine wave,  $x(t)$ , gives rise to two impulses at frequencies  $\pm\Omega_0$ , as shown in Figure 2.10(a). However, in the case of the finite Fourier transform of a cosine wave, the impulse function is convolved with the infinitely ranged Fourier transform of the rectangular window. The resulting Fourier transform of  $x_w(t)$  is given by

$$\begin{aligned} X_w(j\Omega) &= X(j\Omega) * W(j\Omega) \\ &= \pi [\delta(\Omega + \Omega_0) + \delta(\Omega - \Omega_0)] * T \operatorname{sinc} \left( \frac{\Omega T}{2\pi} \right) e^{-\frac{j\Omega T}{2}} \end{aligned}$$

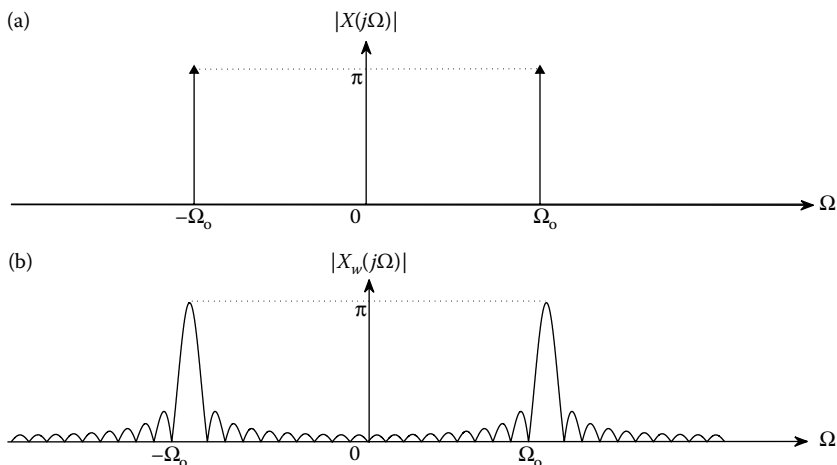


FIGURE 2.10

(a) Magnitude response of a cosine signal. (b) Leakage in finite extent data.

$$= \pi T [\text{sinc}((\Omega + \Omega_0)T/2\pi) + \text{sinc}((\Omega - \Omega_0)T/2\pi)] e^{-\frac{j\Omega T}{2}},$$

$$-\infty < \Omega < \infty. \quad (2.31)$$

The magnitude response of the resulting function  $|X_w(j\Omega)|$ , shown in Figure 2.10(b), are two sinc functions centered at  $\pm\Omega_0$ . This function is not localized on the frequency axis and in fact has a series of spurious peaks called side lobes, that decay quite slowly ( $-6$  dB/octave). This effect is due to the truncation, which is unavoidable while applying the finite DTFT. Owing to these side lobes, it is possible for the finite DTFT to exhibit a number of frequency components instead of only one. To localize the contribution of a given frequency, the usual approach is to apply a different data window to the time series that has lower side lobes in the frequency-domain than that of a rectangular window. However, this will give rise to yet another effect in terms of loss of frequency resolution.

Alternatively, if the input frequency components are integer multiples of the reciprocal of the sample length, then the leakage will be zero (only in discrete-time case). The response is zero at adjacent points, because the zeros of the  $(\frac{\sin x}{x})$  response exactly coincide with the location of the DFT output points.

### 2.2.1 Zero Leakage Case

We will now show that the leakage is produced due to the combination of the sinc side-lobe amplitudes and the transform of the sinusoidal components,

whose frequencies are *not* integer multiples of the reciprocal of the sample length  $T$  [4].

Let the input sequence to the DFT be  $x[n] = Ae^{j\omega_0 n \Delta T}$ , where  $\Delta T = T/N$ , while  $T$  is the sampling interval (in seconds) and  $N$  is the sequence length. The DFT can be computed as follows:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad k = 0, 1, \dots, (N-1) \\ &= \sum_{n=0}^{N-1} A e^{j\omega_0 n \Delta T} e^{-j2\pi nk/N} \\ &= A \sum_{n=0}^{N-1} e^{j(n\alpha)}, \quad \text{where } \alpha = \left( \omega_0 \Delta T - \frac{2\pi k}{N} \right). \end{aligned}$$

The above expression is actually a geometric summation. Therefore, this can be written as a quotient of two terms as follows:

$$\begin{aligned} &= A \frac{1 - e^{jN\alpha}}{1 - e^{j\alpha}} = A \frac{e^{j\frac{N\alpha}{2}} \left[ e^{-j\frac{N\alpha}{2}} - e^{j\frac{N\alpha}{2}} \right]}{e^{j\frac{\alpha}{2}} \left[ e^{-j\frac{\alpha}{2}} - e^{j\frac{\alpha}{2}} \right]} \\ &= A e^{j\frac{(N-1)\alpha}{2}} \cdot \frac{\sin\left(\frac{N\alpha}{2}\right)}{\sin\left(\frac{\alpha}{2}\right)}. \end{aligned} \quad (2.32)$$

Substituting back the value of  $\alpha$ , we obtain the following expression:

$$X[k] = A e^{j\left[\frac{(N-1)}{2} \left(\omega_0 \Delta T - \frac{2\pi k}{N}\right)\right]} \cdot \frac{\sin\left[\frac{N}{2} \left(\omega_0 \Delta T - \frac{2\pi k}{N}\right)\right]}{\sin\left[\frac{1}{2} \left(\omega_0 \Delta T - \frac{2\pi k}{N}\right)\right]}. \quad (2.33)$$

If  $\omega_0 \Delta T = \frac{2\pi m}{N}$  or  $\omega_0 = \frac{2\pi m}{T}$ , then  $f_0$  ( $\omega_0 = 2\pi f_0$ ) is an integer multiple of the reciprocal of the sample length  $T$ . Then, we find that  $X[k]$  can be expressed as follows:

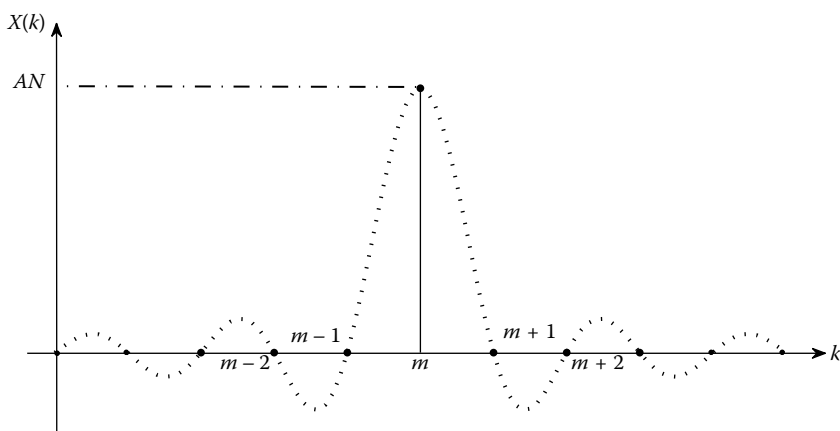
$$X[k] = A e^{j\left[\frac{(N-1)}{2} \pi(m-k)\right]} \cdot \frac{\sin[\pi(m-k)]}{\sin\left[\frac{\pi}{N}(m-k)\right]}. \quad (2.34)$$

As  $m \rightarrow k$ , the expression for  $X[k]$  becomes

$$X[k] = A \lim_{m \rightarrow k} \frac{\sin(\pi(m-k))}{\sin(\pi(m-k)/N)}. \quad (2.35)$$

By applying L'Hospital's rule, we obtain

$$X[k] = A \lim_{m \rightarrow k} N \frac{\cos(\pi(m-k))}{\cos(\pi(m-k)/N)}. \quad (2.36)$$



**FIGURE 2.11**  
DFT output response to demonstrate zero leakage case.

Therefore

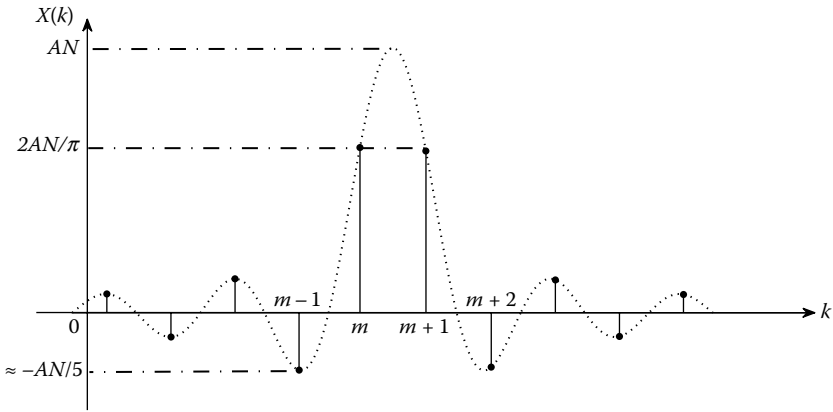
$$X[k] = \begin{cases} AN, & k = m \\ 0, & k \neq m. \end{cases} \quad (2.37)$$

The above result produces the response in the frequency-domain as shown in Figure 2.11. For the sake of comparison, we have shown the sinc function response as dotted lines. Note that the response is zero at adjacent output points. This is true because the zeros of the  $\left(\frac{\sin x}{x}\right)$  response exactly coincide with the location of the DFT output points.

### 2.2.2 Maximum Leakage Case

If  $f_0$  is not an integer ( $m$ ) multiple of the reciprocal of the sample length, but rather  $(m + \frac{1}{2})$ , then  $f_0 = \left(\frac{m+\frac{1}{2}}{T}\right)$  and we obtain the plot of  $X[k]$  as demonstrated in Figure 2.12. From Figure 2.12, it can be seen that the amplitude of  $X[m-1]$  and  $X[m+2]$  is approximately equal to  $\frac{AN}{5}$ . Calculating the ratio of the amplitude at these points and the peak center point, we get,  $20 \log \left|\frac{AN/5}{AN}\right| \simeq -13$  dB. Therefore, the quantity  $-13$  dB corresponds to the peak side-lobe level of a rectangular window. In this case, it can be seen that the response at the adjacent points (near the main output point) is nonzero. In this example, the zeros of the sinc response no longer line up with the adjacent output points, since the input frequency is not an integer multiple of the reciprocal of the sample length. Consequently, we observe that in the overall DFT, the output has energy at frequencies that are not present in the DFT input. This spreading or smearing of energy is referred to as *frequency leakage* or simply *leakage*. It can be shown that for a single frequency component





**FIGURE 2.12**  
Spectral leakage.

input, at a frequency midway between two DFT bins, the leakage is significant ( $-25$  dB) for about eight output points (on either side of the DFT bins) [4].

From the above discussion, we see that to reduce or minimize the leakage problem, we must either limit the frequency components of the input signal to be only integer multiples of  $\frac{1}{T}$  or reduce the side-lobe amplitudes by using appropriate windows, having lower side-lobe amplitudes in the frequency-domain. It is obvious that the first choice is not practical and, therefore, we must find a way to reduce the side-lobe levels.

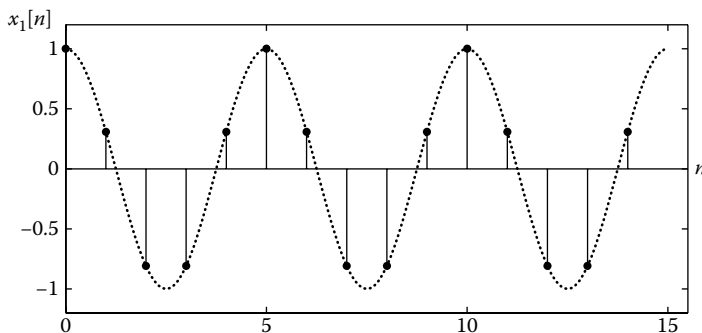
In the above example, we took a signal having a single frequency component. However, the leakage problem increases significantly as the input waveform progresses from a single frequency component to a waveform with many frequency components having differing amplitudes buried in noise. For this case, the leakage will produce nonzero Fourier coefficients throughout the output band. Therefore, these leakage components could be of sufficient amplitudes so as to mask the desired low-amplitude frequency components that are present in the input signal. Hence, the problem of leakage is quite significant in spectral analysis.

**Example**

The following example illustrates the leakage effect. Consider the following two discrete-time sequences:

1.  $x_1[n] = \cos\left(\frac{2\pi n}{5}\right), \quad 0 \leq n \leq 4.$
2.  $x_2[n] = \cos\left(\frac{2\pi n(1.5)}{5}\right), \quad 0 \leq n \leq 4.$

The periodic extension of these two sequences are given in Figures 2.13 and 2.14. From the last section, we have seen that there are only  $N$  distinguishable frequencies for which we get zero leakage when we take the



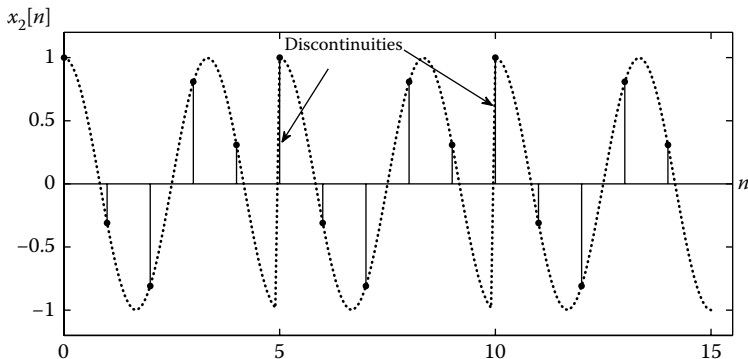
**FIGURE 2.13**  
 $x_1[n]$  extended up to three periods.

$N$ -point DFT of the sequence. One set of these frequencies is

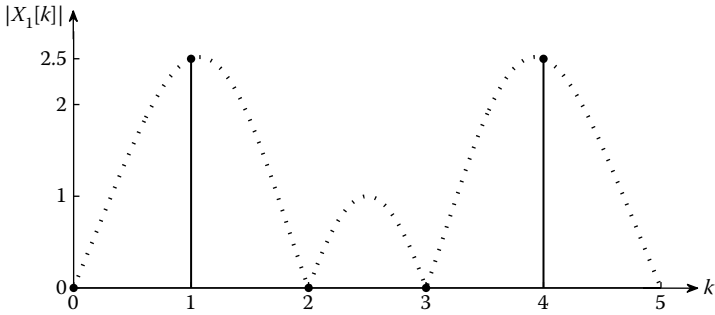
$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, 1, 2, \dots, N - 1. \quad (2.38)$$

In this example,  $x_1[n]$  has a frequency component corresponding to one of these frequencies, resulting in no-leakage case as shown in Figure 2.15. On the other hand, frequency of  $x_2[n]$  does not correspond to any of the DFT bins, resulting in the spread of energy throughout the spectrum, as given in Figure 2.16. In Figures 2.15 and 2.16, the dotted lines represent the DTFT of the sequence. We can see that, owing to the proper choice of  $N$ , the DFT of  $x_1[n]$  obtained by sampling the DTFT has only zeros at all the frequencies, other than at the signal frequency components,  $\frac{2\pi}{5}$  and  $\frac{8\pi}{5}$  or  $\frac{2\pi k}{5}$ , with  $k = 1$  and  $k = 4$ .

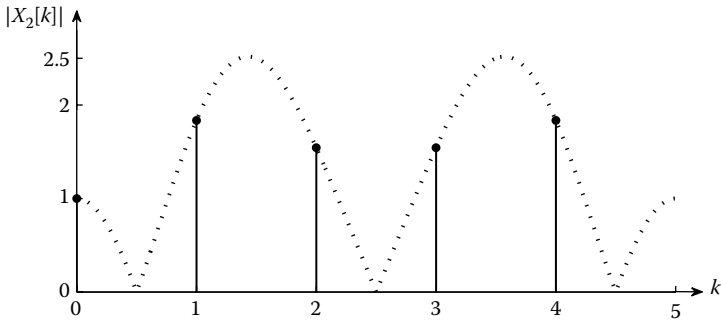
Another intuitive approach to study the effect of leakage is to observe the periodic repetition of the sequences. The periodic repetition of  $x_2[n]$  shows clear discontinuities (Figure 2.14) at  $n = 5, 10, 15, \dots$ , unlike



**FIGURE 2.14**  
 $x_2[n]$  extended up to three periods.



**FIGURE 2.15**  
DFT of  $x_1[n]$  (zero leakage case).



**FIGURE 2.16**  
DFT of  $x_2[n]$  (leakage case).

the extended signal of  $x_1[n]$  (Figure 2.13). These discontinuities are responsible for the nonzero leakage in the case of  $x_2[n]$ .

### 2.3 DFT as a Filter Bank

In an earlier chapter, we have seen how we can relate the DTFT with the DFT of a finite-length sequence  $x[n]$ . If the DTFT of a sequence  $x[n]$ , represented as  $X(e^{j\omega})$ , is a function of a continuous variable  $\omega$ , then the DFT itself is a sequence. They can be related as follows:

$$\begin{aligned}
 X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \Big|_{\omega = \frac{2\pi k}{N}} \\
 &= \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{N}}, \quad k = 0, 1, \dots, N - 1.
 \end{aligned}
 \tag{2.39}$$

Therefore, the DFT of a sequence is a sampled form of the DTFT. The DFT can be considered as the bank of bandpass filters [5], tuned to frequencies corresponding to the DFT bins. Let the discrete-time signal be  $x[n] = e^{j\omega n}$ . Let us observe the DFT output sequences,  $X(0), X(1), \dots, X(N-1)$ , as  $\omega$  is varied from 0 to  $2\pi$ .

$$\begin{aligned} X(0) &= \sum_{n=0}^{N-1} x[n]W_N^0 = \sum_{n=0}^{N-1} e^{j\omega n} = \frac{1 - e^{j\omega N}}{1 - e^{j\omega}} \\ &= \frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \cdot e^{j(N-1)\omega/2}. \end{aligned}$$

In Figure 2.17(a), we show the plot of  $|X(0)|$  versus  $\omega$ , assuming that  $N = 5$ . Now, let us compute  $X(1)$  as illustrated below:

$$\begin{aligned} X(1) &= \sum_{n=0}^{N-1} e^{j\omega n} \cdot W_N^n = \sum_{n=0}^{N-1} e^{j\omega n} \cdot e^{-j\frac{2\pi}{N}n} \\ &= \sum_{n=0}^{N-1} e^{j\left(\omega - \frac{2\pi}{N}\right)n} = \frac{1 - e^{j\left(\omega - \frac{2\pi}{N}\right)N}}{1 - e^{j\left(\omega - \frac{2\pi}{N}\right)}} \\ &= \frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2} - \frac{\pi}{N}\right)} e^{j\omega\left(\frac{N-1}{2}\right)} e^{j\frac{\pi}{N}}. \end{aligned}$$

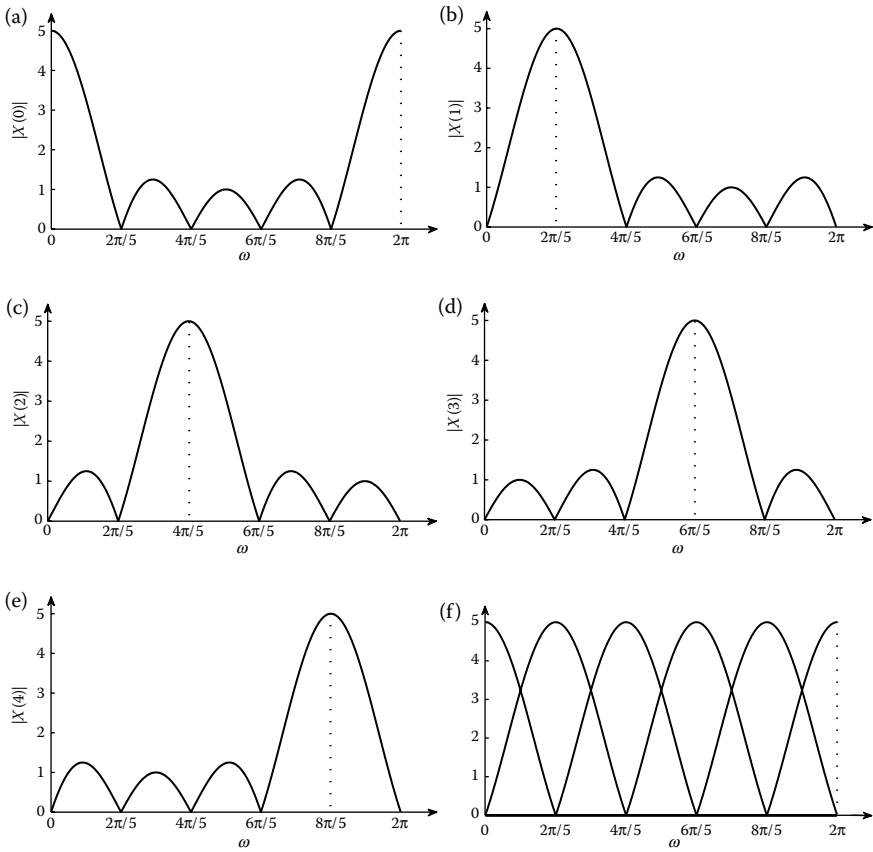
The corresponding plot is given in Figure 2.17(b) for  $N = 5$ . In general, we can express  $X[k]$  as follows:

$$X[k] = \frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2} - \frac{\pi k}{N}\right)} e^{j\omega\left(\frac{N-1}{2}\right)} e^{j\frac{\pi k}{N}}, \quad k = 0, 1, \dots, (N-1). \quad (2.40)$$

If we choose  $N = 5$ , then the magnitude responses of  $X(0), X(1), \dots, X(4)$  can be sketched as given in Figures 2.17(a) through (e).

Figure 2.17(f) combines all the elemental plots of Figures 2.17(a) through (e), but the side lobes are omitted for the sake of clarity. Therefore, from Figure 2.17, we can conclude that the DFT represents (or is analogous to) a bank of bandpass filters. Hence, the DFT can be interpreted in the following three ways:

1. A reversible transformation that converts one complex sequence into another.
2. It corresponds to samples of the  $z$ -transform of the sequence, equispaced on the unit circle (i.e.,  $z = e^{j\frac{2\pi}{N}k}$ ).
3. A bank of bandpass filters.



**FIGURE 2.17** Magnitude response plots at DFT bins as  $\omega$  is varied from 0 to  $2\pi$ . (a)  $|X(0)|$ . (b)  $|X(1)|$ . (c)  $|X(2)|$ . (d)  $|X(3)|$ . (e)  $|X(4)|$ . (f) Main lobes of  $X(k)$ .

## 2.4 Picket-Fence Effect or Scalloping Loss

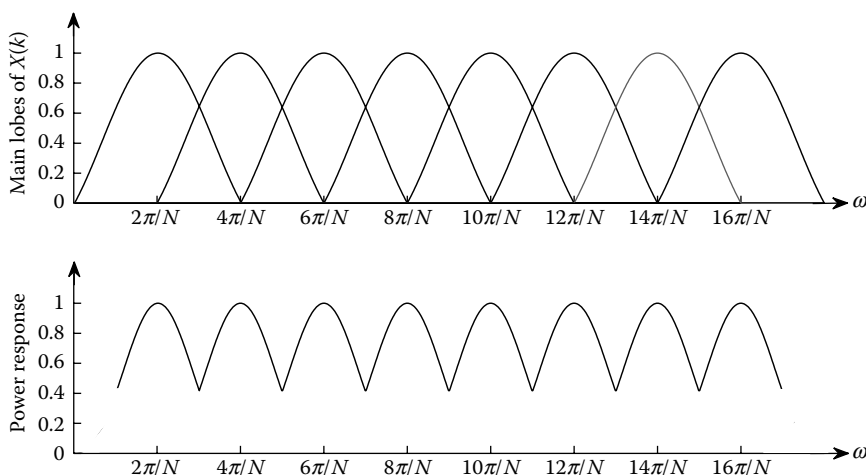
The picket-fence effect is produced by the inability of the DFT to observe the spectrum as a continuous function since the computation of the spectrum is limited to integer multiples of the fundamental frequency [5,6]. From the discussion of the previous section, we can say that the Fourier transformation of the discrete data can be viewed as passing the data through a bank of bandpass filters. Ideally, each Fourier coefficient would act as a complex filter (the absolute values of all the coefficients are equal to one). However, because of the finite-length data, which is equivalent to multiplying the data by a rectangular window, the amplitude response of the filter is in the form of the main lobes of  $(\frac{\sin x}{x})$  functions. The normalized frequency response is shown

in Figure 2.18 (side lobes are not shown for the sake of clarity). Note that the amplitude axis is normalized to unity.

Therefore, at the frequencies computed, these main lobes appear to be independent filters, with unity magnitude response. However, if the data consist of a spectral component that lies, say, for example, between the fourth and fifth harmonic frequencies, then the component is seen by both the filters centered at the fourth and fifth harmonics, but at a value less than unity. In the worst case, when the spectral component lies exactly half-way between the computed harmonics, the amplitude of the signal is reduced to 0.637 [7], which represents the worst case. When this value is squared, the apparent peak power of the signal is only 0.406. Thus, the power spectrum seen by this set of bandpass filters has a ripple that varies by a factor of 2.5 to 1. The rippled curve (shown in the second plot of Figure 2.18), also known as the picket-fence effect, is responsible for the processing loss of input frequencies between the bin centers. Therefore, one seems to be viewing the true spectrum (using the DFT) through a “picket-fence” [7], since we can observe the exact behavior only at discrete points. This effect can be reduced by applying a data window that has a broader main-lobe width (in the frequency-domain) but a larger attenuation in the side lobes than that of a rectangular window.

The picket-fence effect occurs because the  $N$ -point DFT cannot resolve the spectral components any closer than the spacing

$$\Delta f = f_o = \frac{1}{T_o} = \frac{f_s}{N}.$$



**FIGURE 2.18**

Picket-fence effect: DFT coefficients as a set of bandpass filters. (Redrawn from G.D. Bergland, A guided tour of the fast fourier transform, *IEEE Spectrum*, Vol. 6, pp. 41–52, July 1969.)

If the sampling rate is fixed,  $\Delta f$  can be decreased by increasing the number of points in the DFT.

---

## 2.5 Zero-Padding and Frequency Resolution

If the data length is limited to  $T$  s, where  $T < T_o$ , then the data may be extended to  $T_o$  by adding additional zero-valued sampling points. This is called zero-padding and is explained below.

### 2.5.1 Zero-Padding

In this section, we look at the necessity and applications in which zero-padding is employed [8]. It is used in the following scenarios:

1. *Filling the sequence with zeros to utilize the radix-2 FFT algorithm:* Sufficient number of zeros ( $N_z$ ) are added to the  $N$ -point data to satisfy the requirement that

$$(N + N_z) = 2^m \quad (2.41)$$

for a radix-2 FFT, where  $m$  is an integer that represents the number of stages of the FFT algorithm. Besides, the harmonics of the frequency

$$\frac{1}{(N + N_z)T}$$

(where  $T$  is the sampling interval) coincides with the signal frequencies.

2. *Implementing linear convolution through circular convolution:* The circular convolution of the two sequences  $x_1[n]$  and  $x_2[n]$  is computed as follows:

$$y_c[n] = \sum_{m=0}^{N-1} x_1[(m)_N]x_2[(n - m)_N] \quad (2.42)$$

where  $(-)_N$  denotes the modulo  $N$  operation.

As stated in the properties of the DFT, the circular convolution is directly related to the DFTs of  $x_1[n]$  and  $x_2[n]$  as

$$Y_c[k] = X_1[k]X_2[k]. \quad (2.43)$$

Therefore, the circular convolution (denoted as  $x_1[n] \otimes x_2[n]$ ) can be computed as

$$y_c[n] = x_1[n] \otimes x_2[n] = \text{IDFT}\{X_1[k]X_2[k]\}, \quad (2.44)$$

where  $X_1[k]$  and  $X_2[k]$  are the DFTs of  $x_1[n]$  and  $x_2[n]$ , respectively. It was found that the circular convolution calculated from the DFTs is computationally efficient rather than using Equation 2.42 directly. This will be evident if sequences of very large lengths are considered.

The linear convolution of sequences  $x_1[n]$  and  $x_2[n]$  is defined as

$$y_l[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m]. \quad (2.45)$$

If one sequence is of length  $N_1$  and the other sequence is of length  $N_2$ , then  $y_l[n]$  will be of length  $L = (N_1 + N_2 - 1)$ . Circular convolution can be considered as the aliased version of the linear convolution. In fact, the circular and linear convolutions are related (see Section 1.5.2) by

$$y_c[n] = \sum_{k=-\infty}^{\infty} y_l[n - kN], \quad n = 0, 1, \dots, N - 1. \quad (2.46)$$

where  $N$  is the length of each sequence and  $L$  is the length of linear convolution. From the relation,  $y_c[n]$  will be equal to  $y_l[n]$  when the shifts  $N$  is equal to  $L$ .

If the sequences are padded with sufficient number of zeros, i.e.,  $(L - N)$  then we can use IDFT to compute the linear convolution. This is illustrated by the following example.

### Example

Consider the sequences  $x_1[n] = x_2[n] = \{1, 1, 1\}$ .

Here,  $N_1 = N_2 = 3$ . The linear convolution of these two sequences is

$$y_l[n] = \{1, 2, 3, 2, 1\},$$

and the result of the circular convolution is

$$y_c[n] = \{3, 3, 3\}.$$

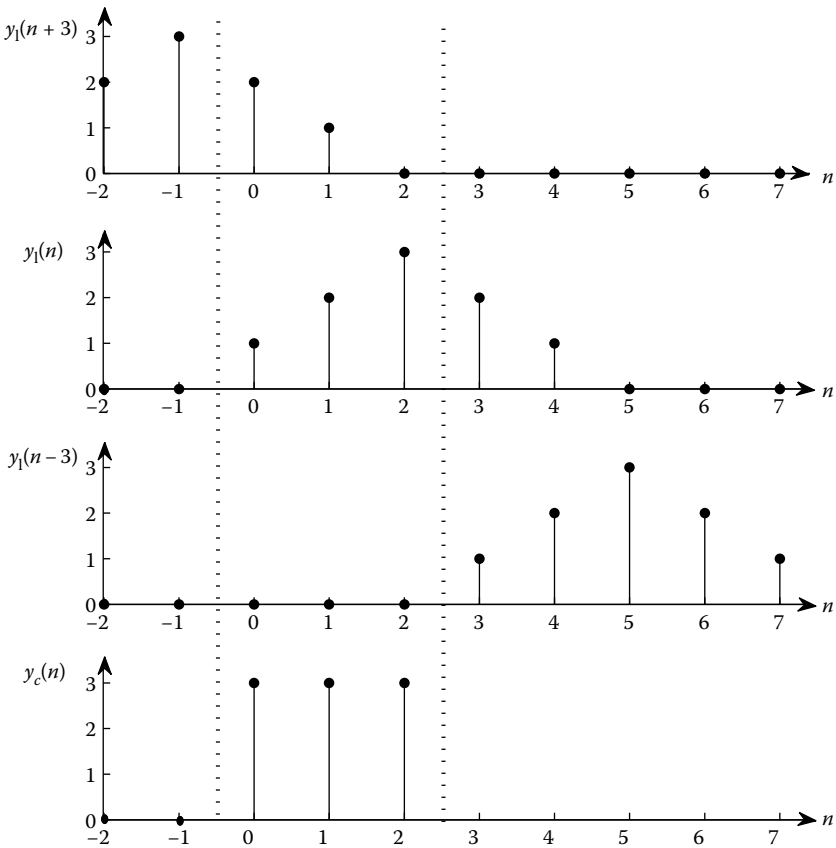
In this example,  $N = 3$ . The circular convolution obtained using the linear convolution from Equation 2.46 is graphically depicted in Figure 2.19. We now pad the sequences  $x_1[n]$  and  $x_2[n]$  with  $(N_2 - 1)$  and  $(N_1 - 1)$  zeros, respectively, to make them equal to  $(N_1 + N_2 - 1)$  in length. We then compute the circular convolution to obtain the linear convolution. From Figure 2.20, we can see that there is no aliasing, and the circular convolution is the same as the linear convolution for the zero-padded case. Therefore, once we pad the sequences with the required number of



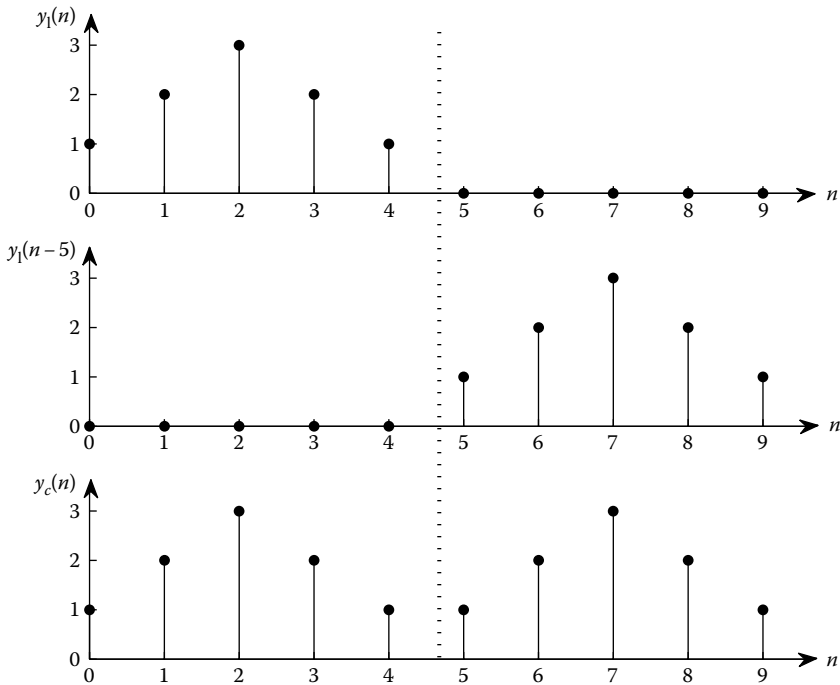
zeros, we can compute the circular convolution using the IDFT of the product of the DFTs of individual sequences to get the linear convolution.

3. *Providing a better display of the spectrum of a finite-length sequence:* We consider an example where  $N_z$  zeros are added to a sequence (which originally had a nonzero length of  $N_f$ ) to give a sequence of  $N$  values, and then an  $N$ -point DFT is computed. The sampled values of the DFT spectrum are spaced  $\left(\frac{2\pi}{N_f+N_z}\right)$  apart. For a sequence  $x[n]$  defined as

$$x[n] = \begin{cases} 1, & n = 0, 1, 2, 3 \\ 0, & \text{otherwise,} \end{cases} \tag{2.47}$$



**FIGURE 2.19** Circular convolution as aliased form of linear convolution.



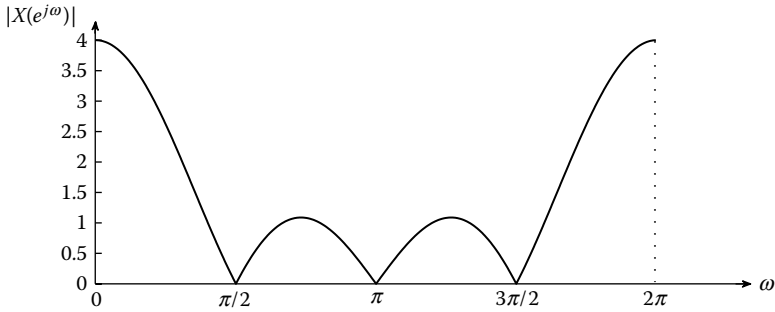
**FIGURE 2.20**  
Circular convolution by linear convolution (after zero-padding).

the DTFT magnitude response  $|X(e^{j\omega})|$  of the above sequence  $x[n]$  is shown in Figure 2.21. While Figure 2.22(a) shows the DFT of a four-point sequence without any zero-padding, Figures 2.22(b) and 2.22(c) show the 8-point and 16-point DFTs of the zero-padded sequences, respectively. From these plots, we can observe that as more number of zeros are added, the DFT provides closely spaced samples of its Fourier transform of the original sequence and thus generates a better-looking display [9]. However, it must be mentioned here that we do not have any additional information that could be obtained by sampling the *interpolation formula*, which is given by

$$X(e^{j\omega}) = \sum_{k=0}^{N-1} X[k] \phi \left( \omega - \frac{2\pi k}{N} \right), \tag{2.48}$$

where

$$\phi(\omega) = \frac{e^{-j\omega(\frac{N-1}{2})}}{N} \cdot \frac{\sin(\omega N/2)}{(\omega/2)}. \tag{2.49}$$



**FIGURE 2.21**  
DTFT of  $x[n]$ .

Here, Equations 2.48 and 2.49 provide the interpolation formula and the interpolation function, respectively. We emphasize that zero-padding does not actually improve the frequency resolution but only gives a better-looking display of the available data (without additional calculations) to evaluate the values of the DFT through the interpolation formula.

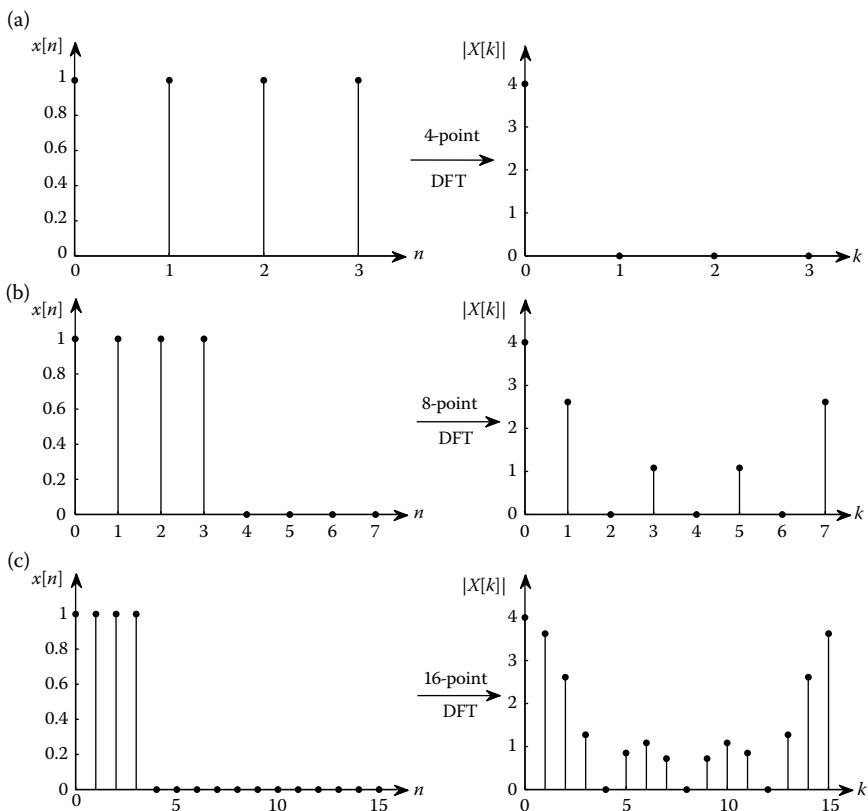
### 2.5.2 Frequency Resolution

The sequence  $x[n]$  is often produced by sampling a continuous-time signal  $x_c(t)$ , resulting in the sequence  $x_c(nT)$ . The DFT coefficients for this sampled signal can be written as  $X[k\Delta f]$  rather than simply  $X[k]$ , where  $\Delta f$  represents the frequency spacing of the coefficients. This is the same as  $T$  in  $x(nT)$  representing the time spacing of the sampled signal. Therefore, the frequency components in the DFT are spaced apart according to the following relations:

$$\Delta f = \frac{F_s}{N} \quad \text{or} \quad \Delta f = \frac{1}{NT} = \frac{1}{T_o}, \tag{2.50}$$

where  $F_s$  is the sampling frequency (or  $T$  is the sampling interval,  $T = \frac{1}{F_s}$ ) and  $N$  is the period of the resulting sequence,  $\Delta f$  is the frequency spacing (also called as frequency resolution) and  $T_o$  is the record length ( $T_o = NT$ ). To resolve closely spaced frequencies, or in other words to increase the resolution, we need to make  $\Delta f$  appropriately smaller.

As discussed before, the application of DFT to a finite-length data gives rise to leakage and picket-fence effects. These effects can be reduced by weighting the data with suitable windows. However, the use of data windows (other than rectangular window) affects the bias, variance, and frequency resolution of the spectral estimates. In general, the variance of the estimates increases with the use of windows. An estimate is said to be consistent if both the bias and the variance of the estimate tend to zero as the number of observations



**FIGURE 2.22**  $x[n]$  and its DFTs: (a) No zero-padding. (b) Padding with four zeros. (c) Padding with 12 zeros.

is increased. Hence, the issues associated with the spectral estimation of a random data by the DFT technique reduce to the problem of establishing efficient data windows or data-smoothing schemes. This topic shall be discussed in detail in a forthcoming chapter.

## References

1. C.E. Shannon, Communication in the presence of noise, *Proceedings of IRE*, vol. 37, no. 1, pp. 10–21, January 1949.
2. E.T. Whittaker, Expansion of the interpolation theory, *Proceedings of the Royal Society of Edinburgh*, vol. 3, p. 181, 1915.
3. A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, 3rd Edn., Prentice-Hall, Upper Saddle River, NJ, 2010.

4. D. Gingras, *Time Series Windows for Improving Discrete Spectra Estimation*, Naval Undersea Research and Development Center, San Diego, California, NUC TN715, April 1972.
5. F.J. Harris, On the use of windows for harmonic analysis with the discrete fourier transform, *Proceedings of IEEE*, vol. 66, no. 1, pp. 51–83, January 1978.
6. R.O. Nielsen, *Sonar Signal Processing*, Artech House, Inc. Norwood, MA, 1991.
7. G.D. Bergland, A guided tour of the fast Fourier transform, *IEEE Spectrum*, vol. 6, pp. 41–52, July 1969.
8. R.G. Lyons, *Understanding Digital Signal Processing*, 3rd Edn., Prentice-Hall, Upper Saddle River, NJ, 2011.
9. L.C. Ludeman, *Fundamentals of Digital Signal Processing*, John Wiley & Sons, New York, NY, 1986.

# 3

---

## *Review of Window Functions*

---

---

### 3.1 Introduction

This chapter presents a concise review of all popular window functions that are commonly employed in digital signal processing. Since no window is the best in all aspects, it should be selected according to the user's requirements. First, the characteristics that qualify a function to be a window function are outlined in Section 3.2. In Section 3.3, almost all the window functions are cataloged, along with their time-domain and frequency-domain representations. The Fourier domain representations of these windows are also discussed at length. Section 3.4 provides theorems relating to the rate of fall-off side-lobe levels (RFSLL) of windows. We then describe the various basic parameters of the windows that are useful in choosing an efficient window for a particular application. Finally, a comprehensive comparison of all the windows in terms of their computed parameters is provided in Section 3.5.

---

### 3.2 Characteristics of a Window Function

In this section, we list some of the desirable characteristics of a window function. For ease of discussion, we describe the window functions in the continuous domain in time and frequency. Here,  $f(t)$  represents the window of length  $2\tau$  in the time-domain and  $F(j\Omega)$  represents its continuous Fourier transform, that is,

$$F(j\Omega) = \int_{-\tau}^{\tau} f(t)e^{-j\Omega t} dt. \quad (3.1)$$

In the rest of this chapter, the main lobes and the side lobes refer to those of the Fourier transform  $F(j\Omega)$  of the window. Every window,  $f(t)$ , and its Fourier transform,  $F(j\Omega)$ , are required to possess the following properties:

1.  $f(t)$  should be real and nonnegative.
2.  $f(t)$  should be an even function, that is,  $f(t) = f(-t)$  and therefore  $F(j\Omega)$  is real.
3.  $f(t)$  should attain its maximum at  $t = 0$ , that is,  $|f(t)| \leq f(0)$ , for all  $t$  and  $f(t) = 0$  for  $|t| > \tau$ , where  $\tau$  represents the one-sided duration of the window.
4.  $F(j\Omega)$  should have a main lobe about the origin and side lobes on either side.
5. The main-lobe width should be as narrow as possible.
6. The main lobe must contain a large part of the total energy.
7. The maximum side-lobe level (MSLL) should be as small as possible, relative to the main-lobe peak.
8. If the  $m$ th derivative of  $f(t)$  is impulsive, then the peak of the side lobes of  $|F(j\Omega)|$  decays asymptotically as  $6m$  dB/octave. The proof of this property is presented in Section 3.4.

On the basis of the above characteristics, a number of windows have been proposed by researchers and are detailed below. Each window is illustrated by the plots of the time function and its Fourier transform.

---

### 3.3 List of Windows

We shall now briefly present some of the well-known window functions used in the signal processing literature. Each window is described by its functional form in the continuous time-domain,  $f(t)$ , and its CTFT,  $F(j\Omega)$ . In the following discussion,  $\tau$  represents the one-sided duration of the window in the time-domain. Figures 3.1 through 3.16 (except for Figures 3.5 and 3.8) present the plots of window functions in the time-domain and their CTFTs. The time-domain function is plotted as a function of the normalized time parameter,  $(t/\tau)$ . The normalized magnitude of the Fourier transform is plotted as a function of the normalized frequency parameter  $(\Omega\tau/2\pi)$ , where the normalization is carried out with respect to the amplitude at the origin,  $F(0)$ . The normalized magnitude is expressed in dB scale, that is,  $20 \log_{10} \left| \frac{F(j\Omega)}{F(0)} \right|$ . We also provide vital parameters such as (i) normalized half main-lobe width (NHMLW), (ii) first side-lobe level (FSL), (iii) maximum SLL (MSLL), (iv) ratio of main-lobe energy to total energy (MLE), and (v) rate of fall-off side-lobe level (RFSLL). The same terminology is followed for all the windows discussed in this chapter. In Table 3.1, all the main properties of the window functions described in Sections 3.3.1 through 3.3.18 are summarized.

### 3.3.1 Rectangular (Box Car) Window

The rectangular window [1], which is also called a uniform window or a box car window due to its shape, is defined as follows:

$$f(t) = \begin{cases} 1, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.2)$$

and its CTFT is

$$F(j\Omega) = \frac{2\tau \sin(\Omega\tau)}{\Omega\tau}, \quad -\infty < \Omega < \infty. \quad (3.3)$$

Therefore, the CTFT of a rectangular window function represents a sinc function.

The rectangular window results from the direct truncation of the signal. The Fourier transform of this window shows that the first side lobe of this window is about one-fifth of the main-lobe peak and the side lobes fall at a rate of  $\frac{1}{\Omega}$ . The time-domain (i.e.,  $f(t)$  vs.  $t/\tau$ ) and the frequency-domain (i.e.,  $F(j\Omega)$  vs.  $\Omega\tau/2\pi$ ) plots of a rectangular window are given in Figure 3.1(a). In the frequency-domain plot, the abscissa ( $x$ -axis) is given by  $(\frac{\Omega\tau}{2\pi})$  and the ordinate ( $y$ -axis) is given by  $20 \log_{10} \left| \frac{F(j\Omega)}{F(0)} \right|$ . Thus, both the amplitude and the frequency axes display normalized values. Therefore, the normalized half main-lobe width (NHMLW) of the rectangular window is 0.5. The MSL of this window, which is also the same as the first side-lobe level (FSL), is about  $-13$  dB.

### 3.3.2 Triangular (Bartlett) Window

The triangular window [1] is derived by linearly convolving two rectangular windows of half the duration (i.e.,  $\tau/2$ ). This window is specified by the time-domain function

$$f(t) = \begin{cases} 1 - \frac{|t|}{\tau}, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.4)$$

and its corresponding Fourier transform is

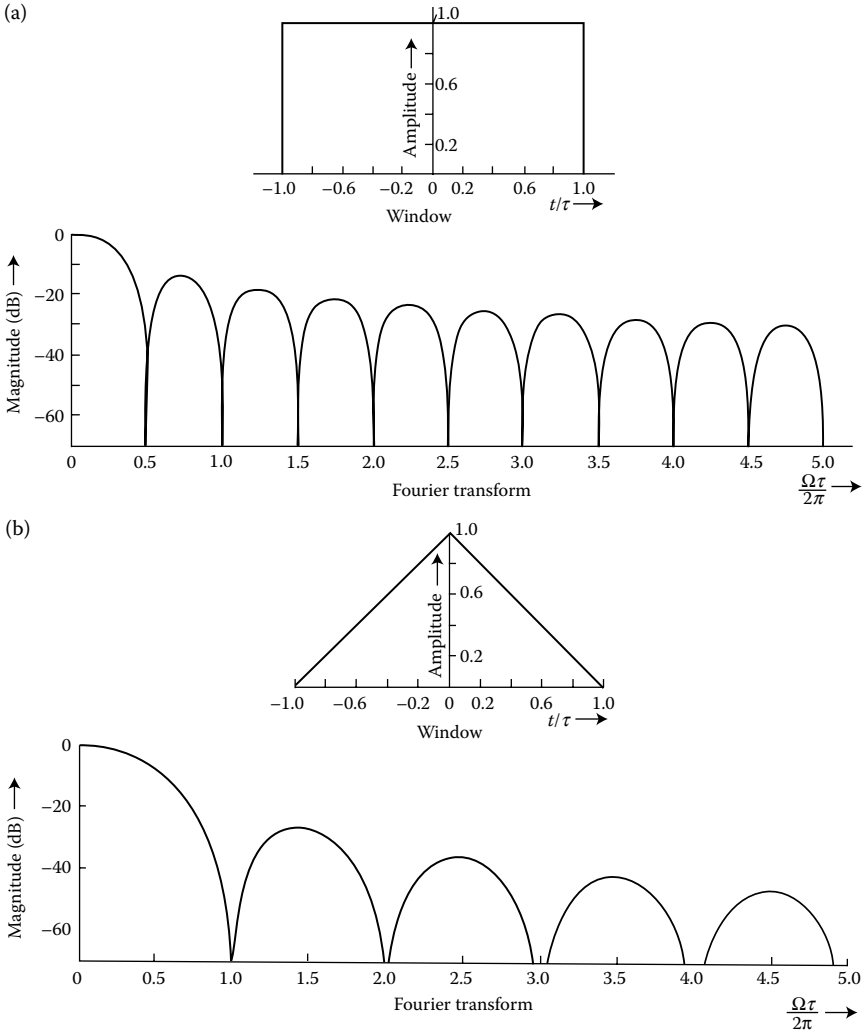
$$F(j\Omega) = \tau \left[ \frac{\sin\left(\frac{\Omega\tau}{2}\right)}{\left(\frac{\Omega\tau}{2}\right)} \right]^2, \quad -\infty < \Omega < \infty. \quad (3.5)$$



Therefore, the Fourier transform pair of the Bartlett window can be represented as follows:

$$\left(1 - \frac{|t|}{\tau}\right)_{|t| \leq \tau} \xleftrightarrow{\mathcal{F}} \frac{4 \sin^2\left(\frac{\Omega\tau}{2}\right)}{(\Omega^2\tau)} \tag{3.6}$$

The relevant plots are shown in Figure 3.1(b). We can observe that the half main-lobe width of this window is twice that of the rectangular window (i.e.,



**FIGURE 3.1** Window functions (time- and frequency-domain plots). (a) Rectangular (box car) window. (b) Triangular (Bartlett) window.

1.0) and that the side lobes fall at the rate of  $\frac{1}{\Omega^2}$  (since  $F(j\Omega) \simeq \frac{4}{\Omega^2\tau}$  as  $\Omega \rightarrow \infty$ ). It can be noted that this is the simplest window that exhibits a nonnegative Fourier transform. This happens due to the self-convolution property, which can always be achieved by convolving any window by itself.

### 3.3.3 Cos(x) Window

The time function of this window [2] and its Fourier transform are given by

$$f(t) = \begin{cases} \cos\left(\frac{\pi t}{2\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.7)$$

and

$$F(j\Omega) = \frac{\sin\left(\left(\Omega + \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{2\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{2\tau}\right)},$$

$$-\infty < \Omega < \infty. \quad (3.8)$$

The Fourier transform pair for this window can be expressed as follows:

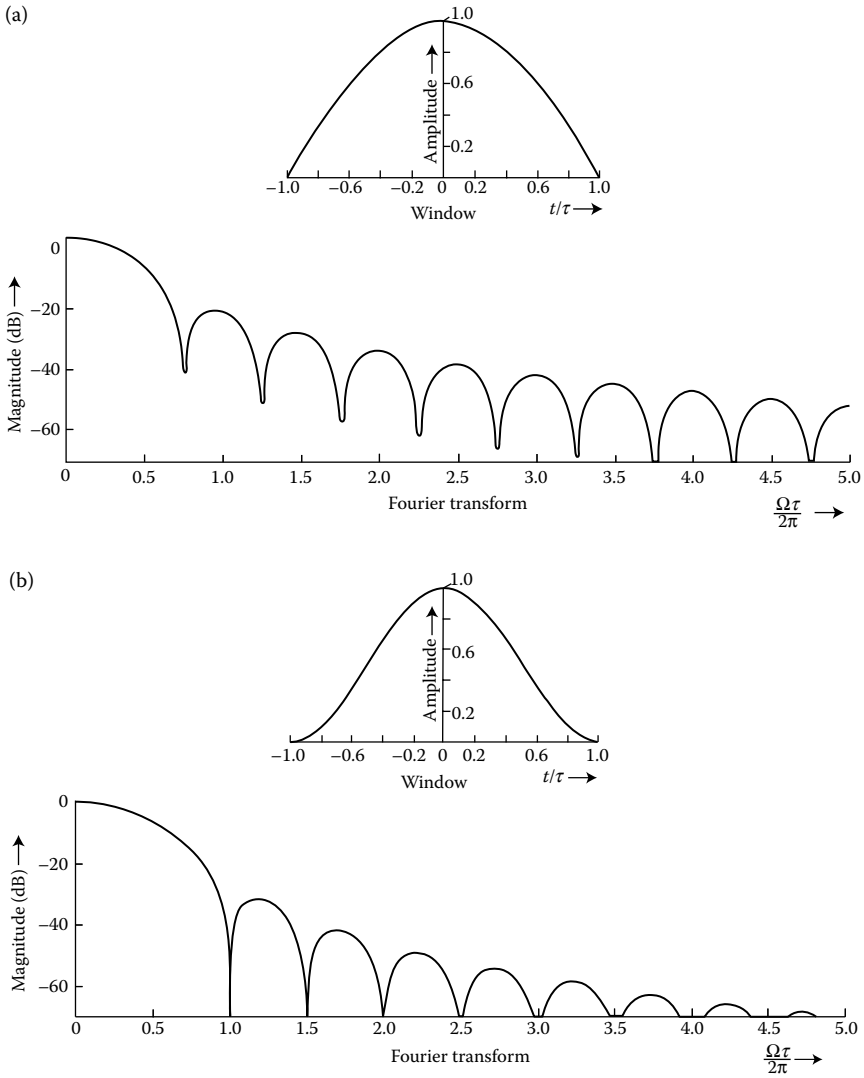
$$\left(\cos\left(\frac{\pi t}{2\tau}\right)\right)_{|t| \leq \tau} \xleftrightarrow{\mathcal{F}} 4\pi\tau \frac{\cos(\Omega\tau)}{(\pi^2 - 4\Omega^2\tau^2)}. \quad (3.9)$$

The advantage of this window is the ease with which the term can be generated. This window is depicted in Figure 3.2(a). Note that the NHMLW of this window is 0.75, (i.e., one and a half times that of the rectangular window), while the RFSLL is  $\frac{1}{\Omega^2}$  (see Equation 3.9). As shown in Table 3.1, the FSL is about 23 dB down. The side-lobe levels for all the window functions discussed in the rest of the section are also given in this table.

### 3.3.4 Hann (Raised-Cosine) Window

This window was proposed by the Austrian meteorologist Julius von Hann. It is also known by different names: the raised-cosine, von Hann window [1,2], and so on. It is defined by the Fourier transform pair

$$f(t) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.10)$$



**FIGURE 3.2** Window functions (time- and frequency-domain plots). (a) Cosine window. (b) Hann window.

and

$$F(j\Omega) = \frac{\sin(\Omega\tau)}{\Omega} + 0.5 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right],$$

$$-\infty < \Omega < \infty. \tag{3.11}$$

**TABLE 3.1**  
Parameters of Different Data Windows

Window	Normalized Half Main-Lobe Width (NHMLW)	First Side-Lobe Level (FSLL) (dB)	Maximum Side-Lobe Level (MSLL) (dB)	Ratio of Main-Lobe Energy to Total Energy (MLE)	Rate of Fall-off Side-Lobe Levels (RFSLL)
Rectangular	0.5	-13.27	-13.27	0.902820	$1/\Omega$
Parabolic	0.708	-21.28	-21.28	0.992084	$1/\Omega^2$
Cosine	0.75	-23.0	-23.0	0.994949	$1/\Omega^2$
Triangular	1.0	-26.53	-26.53	0.997057	$1/\Omega^2$
Hann	1.0	-31.47	-31.47	0.999485	$1/\Omega^3$
Truncated Taylor family ( $K = 0.072$ )	1.0	-42.05	-42.05	0.999698	$1/\Omega$
Hamming	1.0	-44.05	-42.69	0.999632	$1/\Omega$
Raised-cosine ( $b = 0.0113$ )	1.104	-42.02	-42.02	0.999947	$1/\Omega$
Raised-cosine ( $b = 0.0138$ )	1.15	-47.46	-47.46	0.9999608	$1/\Omega$
Raised-cosine ( $b = 0.0155$ )	1.198	-54.12	-52.81	0.9999533	$1/\Omega$
Raised-cosine ( $b = 0.0165$ )	1.245	-61.94	-51.79	0.999945	$1/\Omega$
$\text{Cos}^3x$	1.25	-39.30	-39.30	0.999925	$1/\Omega^4$
Sum-cosine ( $B = 0.100$ )	1.25	-58.62	-52.25	0.999987	$1/\Omega^2$
Sum-cosine ( $B = 0.103$ )	1.25	-54.01	-54.01	0.9999862	$1/\Omega^2$
Blackman	1.5	-58.12	-58.12	0.999990	$1/\Omega^3$
Optimized Blackman	1.5	-68.38	-62.58	0.9999899	$1/\Omega^3$
Papoulis	1.5	-46.01	-46.01	0.999990	$1/\Omega^4$
$\text{Cos}^4x$	1.5	-46.75	-46.75	0.999976	$1/\Omega^5$
Blackman-Harris	2.0	-91.98	-91.98	0.999981	$1/\Omega$
Parzen	2.0	-53.05	-53.05	0.999976	$1/\Omega^4$

The Hann window can be represented in a short-hand notation as

$$\left(\frac{1}{2} + \frac{1}{2} \cos\left(\frac{\pi t}{\tau}\right)\right)_{|t| \leq \tau} \xleftrightarrow{\mathcal{F}} \frac{\pi^2 \sin(\Omega\tau)}{\Omega(\pi^2 - \Omega^2\tau^2)}. \tag{3.12}$$

From Equation 3.11, it is obvious that the Fourier transform of the Hann window is the sum of three  $\frac{\sin(x)}{x}$  terms. The function at the origin is assigned a weight of unity and the other two are shifted sinc functions on either side of the origin by  $(\pm \frac{\pi}{\tau})$  with an assigned weight of 0.5 each. This yields significantly reduced side lobes over that of the rectangular window, but at the expense of the main-lobe width (which is twice that of the rectangular window). The side lobes fall off at a rate  $\frac{1}{\Omega^3}$  (since  $F(j\Omega) \simeq \frac{\pi^2}{\tau^2\Omega^3}$  as  $\Omega \rightarrow \infty$ ). The corresponding plots of the Hann window are shown in Figure 3.2(b). This window is also called a  $\cos^2(x)$  window, since it results by just expanding the square of a cosine function.

### 3.3.5 Truncated Taylor Family

Taylor functions are obtained by adding a weighted-cosine series to a constant (called a pedestal). A simpler form of these functions can be obtained by dropping some of the higher-order terms in the Taylor series expansion. If all other terms, except for the first two significant ones, are dropped, a truncated Taylor function is obtained, which can be expressed as [3]

$$f(t) = \begin{cases} \frac{(1+k)}{2} + \frac{(1-k)}{2} \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \tag{3.13}$$

whose Fourier transform is

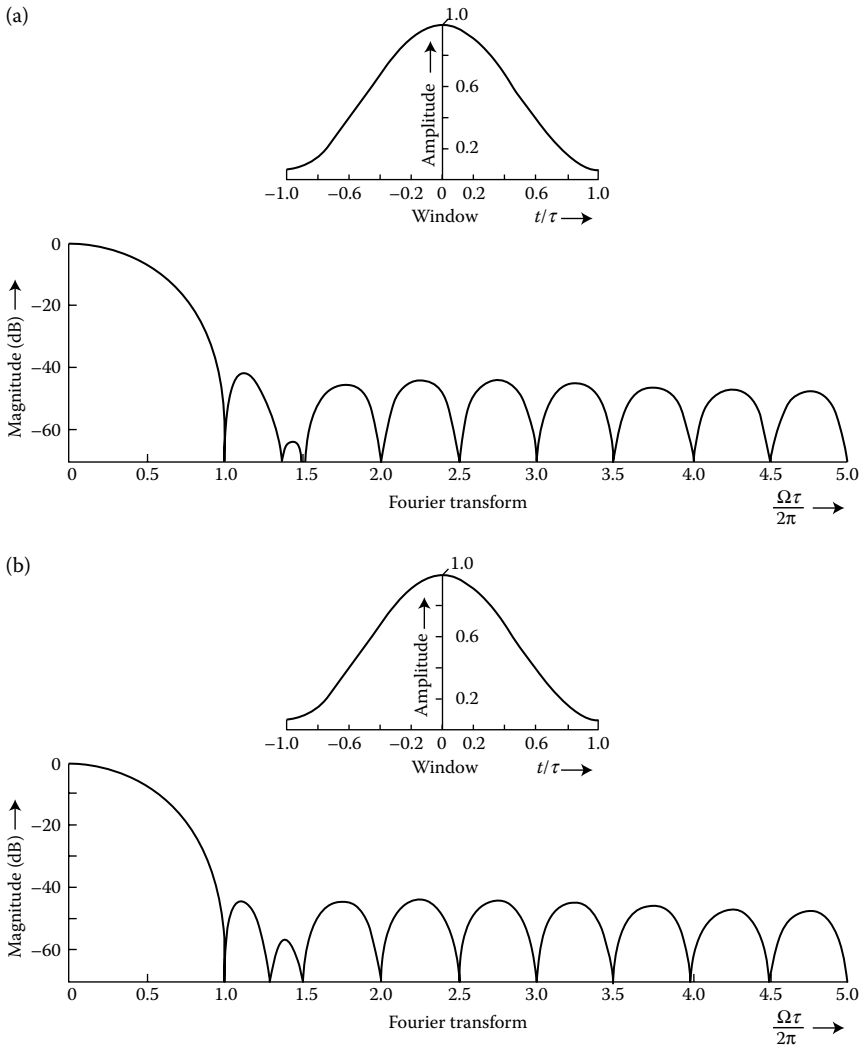
$$F(j\Omega) = (1+k) \frac{\sin(\Omega\tau)}{\Omega} + \frac{(1-k)}{2} \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right],$$

$$-\infty < \Omega < \infty. \tag{3.14}$$

Equation 3.13 can be interpreted as a cosine-squared response, weighted by a factor  $(1 - k)$  and sitting on a pedestal of height  $k$ , where  $k \leq 1$ . Equation 3.14 can be simplified and written as follows:

$$F(j\Omega) = \frac{[\pi^2(1+k) - 2k\Omega^2\tau^2] \sin(\Omega\tau)}{\Omega(\pi^2 - \Omega^2\tau^2)}. \tag{3.15}$$

It can be seen that  $F(j\Omega) \rightarrow 0$  with the rate of  $\simeq \left(\frac{2k}{\Omega}\right)$ , as  $\Omega \rightarrow \infty$ .



**FIGURE 3.3**

Window functions (time-domain and frequency-domain plots). (a) Truncated Taylor window with  $k = 0.072$ . (b) Hamming window.

The range of side-lobe levels varies as a function of the pedestal height (see Figures 3.2(b) and 3.3(a)). Figures 3.2(b) and 3.3(a) represent the plots for the values of  $k = 0$  (Hann window) and  $k = 0.072$ , respectively. It is interesting to point out that the truncated Taylor family is closely related to two other functions: the Hann window results when  $k = 0$  and the Hamming window (discussed below), which is a truncated Taylor window with  $k = 0.08$  (see Figure 3.3(b)).

### 3.3.6 Hamming Window

This window can be thought of as an optimized form of the Hann window, and it was proposed by Hamming [1,4]. The coefficients of this window are optimized so as to obtain the minimum FSL. The Hamming window finds applications in optics for apodization, which smoothens the input intensity or transmission profile, such that it approaches almost zero at the edges. Its functional form is represented by

$$f(t) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.16)$$

whose Fourier transform can be written as follows:

$$F(j\Omega) = 1.08 \left[ \frac{\sin(\Omega\tau)}{\Omega} \right] + 0.46 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right],$$

$$-\infty < \Omega < \infty. \quad (3.17)$$

Equations 3.16 and 3.17 together can be represented by a Fourier transform pair as follows:

$$\left(0.54 + 0.46 \cos\left(\frac{\pi t}{\tau}\right)\right)_{|t| \leq \tau} \xleftrightarrow{\mathcal{F}} \frac{[1.08\pi^2 - 0.16\Omega^2\tau^2] \sin(\Omega\tau)}{\Omega(\pi^2 - \Omega^2\tau^2)}. \quad (3.18)$$

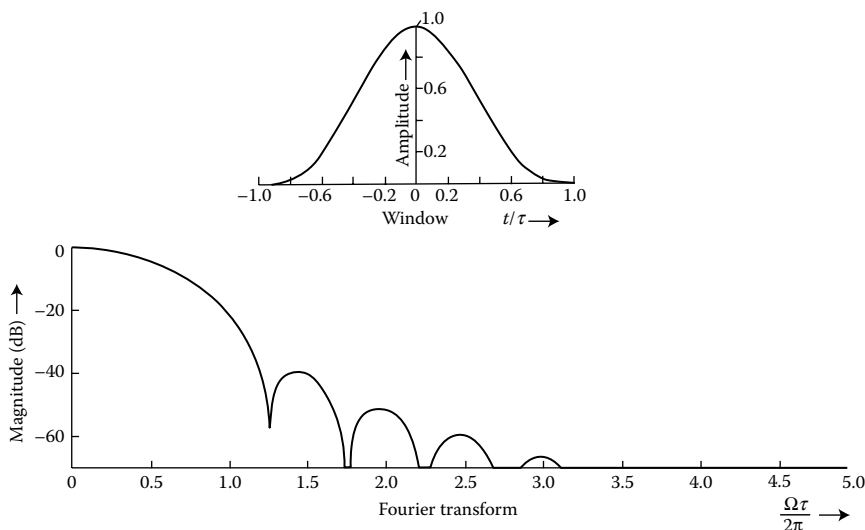
It can be seen that the asymptotic attenuation in the case of Hamming window is  $\left(\frac{0.16}{\Omega}\right)$ .

The time- and Fourier-domain plots are shown in Figure 3.3(b). From these plots, we can see that the MSLL is about  $-42$  dB and the side lobes fall at the rate of  $\frac{1}{\Omega}$ . This slow fall-off rate is due to the small discontinuity (0.08) at the edges of the window. However, the FSL of this window is about  $-44$  dB. Following the discussion in Section 3.3.5, we must point out here that the Hamming window yields the lowest side-lobe levels for the truncated Taylor class of window functions (with  $k = 0.08$ ).

### 3.3.7 $\cos^3(x)$ Window

The time function of this window [2] is obtained by expanding the  $\cos^3(x)$  function (hence its name) and it can be rewritten as

$$f(t) = \begin{cases} 0.75 \cos\left(\frac{\pi t}{2\tau}\right) + 0.25 \cos\left(\frac{3\pi t}{2\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere.} \end{cases} \quad (3.19)$$



**FIGURE 3.4**  
 $\text{Cos}^3(x)$  Window (time and frequency-domain).

and its Fourier domain representation is expressed as

$$\begin{aligned}
 F(j\Omega) = & 0.75 \left[ \frac{\sin \left( \left( \Omega + \frac{\pi}{2\tau} \right) \tau \right)}{\left( \Omega + \frac{\pi}{2\tau} \right)} + \frac{\sin \left( \left( \Omega - \frac{\pi}{2\tau} \right) \tau \right)}{\left( \Omega - \frac{\pi}{2\tau} \right)} \right] \\
 & + 0.25 \left[ \frac{\sin \left( \left( \Omega + \frac{3\pi}{2\tau} \right) \tau \right)}{\left( \Omega + \frac{3\pi}{2\tau} \right)} + \frac{\sin \left( \left( \Omega - \frac{3\pi}{2\tau} \right) \tau \right)}{\left( \Omega - \frac{3\pi}{2\tau} \right)} \right], \quad -\infty < \Omega < \infty.
 \end{aligned}
 \tag{3.20}$$

The NHMLW of this window is 1.25 and the side-lobe fall-off rate can be proven to be of the order of  $\frac{1}{\Omega^4}$  (see Figure 3.4).

### 3.3.8 Sum-Cosine Window

This function can be considered as an optimized form of the  $\text{cos}^3(x)$  window (similar to that of the Hamming and Hann windows) [5]. It is expressed as

$$f(t) = \begin{cases} (1 - 2B) \cos \left( \frac{\pi t}{2\tau} \right) + 2B \cos \left( \frac{3\pi t}{2\tau} \right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases}
 \tag{3.21}$$



where  $B$  is a constant and the Fourier representation of  $f(t)$  is

$$F(j\Omega) = (1 - 2B) \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{2\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{2\tau}\right)} \right] \\ + 2B \left[ \frac{\sin\left(\left(\Omega + \frac{3\pi}{2\tau}\right)\tau\right)}{\left(\Omega + \frac{3\pi}{2\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{3\pi}{2\tau}\right)\tau\right)}{\left(\Omega - \frac{3\pi}{2\tau}\right)} \right], \quad -\infty < \Omega < \infty. \quad (3.22)$$

We will now discuss in detail how a sum-cosine window is synthesized [5,6]. Recall the time function of the rectangular window:

$$f_r(t) = \begin{cases} 1, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.23)$$

and that of the cosine window:

$$f_c(t) = \begin{cases} \cos\left(\frac{\pi t}{2\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere.} \end{cases} \quad (3.24)$$

The Fourier transforms of the rectangular and cosine windows are given by

$$F_r(j\Omega) = \frac{2 \sin(\Omega\tau)}{\Omega}, \quad -\infty < \Omega < \infty. \quad (3.25)$$

and

$$F_c(j\Omega) = \frac{\sin\left(\left(\Omega + \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{2\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{2\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{2\tau}\right)}, \quad -\infty < \Omega < \infty, \quad (3.26)$$

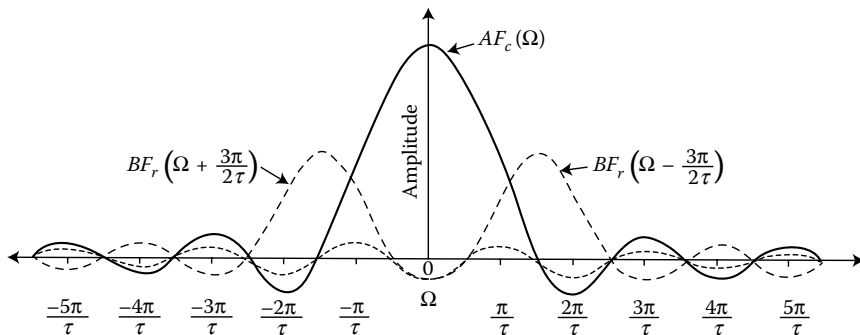
respectively.

Consider  $A$  and  $B$  to be two real constants and let the Fourier transform of the rectangular window be shifted to the right (and left) of the origin by an amount  $\left(\frac{3\pi}{2\tau}\right)$  [5]. The resulting waveforms of the three elemental Fourier transforms, namely,  $BF_r(j(\Omega + \frac{3\pi}{2\tau}))$ ,  $BF_r(j(\Omega - \frac{3\pi}{2\tau}))$ , and  $AF_c(j\Omega)$ , are displayed in Figure 3.5. Therefore, a linear combination of these three waveforms gives

$$F_{sc}(j\Omega) = AF_c(j\Omega) + BF_r\left(j\left(\Omega + \frac{3\pi}{2\tau}\right)\right) + BF_r\left(j\left(\Omega - \frac{3\pi}{2\tau}\right)\right), \quad (3.27)$$

whose inverse CTFT can be shown to be

$$f_{sc}(t) = A \cos\left(\frac{\pi t}{2\tau}\right) + 2B \cos\left(\frac{3\pi t}{2\tau}\right), \quad |t| \leq \tau. \quad (3.28)$$



**FIGURE 3.5**  
Fourier transforms of cosine and rectangular windows.

Since the window functions are required to be even, without the loss of generality,  $f_{sc}(t)$  can be assumed to be unity at the origin (i.e.,  $A + 2B = 1$ ), in which case Equation 3.28 reduces to

$$f_{sc}(t) = (1 - 2B) \cos\left(\frac{\pi t}{2\tau}\right) + 2B \cos\left(\frac{3\pi t}{2\tau}\right), \quad |t| \leq \tau. \quad (3.29)$$

It can be noticed from Figure 3.5 that the side-lobe ripples from the functions  $BF_r(\Omega \pm \frac{3\pi}{2\tau})$  tend to cancel the side-lobe ripples from  $AF_c(\Omega)$ , thereby considerably reducing the overall side-lobe levels of the resulting Fourier transform,  $F_{sc}(j\Omega)$ . However, this will be at the expense of the main-lobe width. To achieve the minimum side-lobe level, constant  $B$  is evaluated under two different constraints, specified by the following criterion:

$$\left| \frac{F_{sc}(0)}{F_{sc}(j\Omega_1)} \right| = \text{Maximum} \quad (3.30)$$

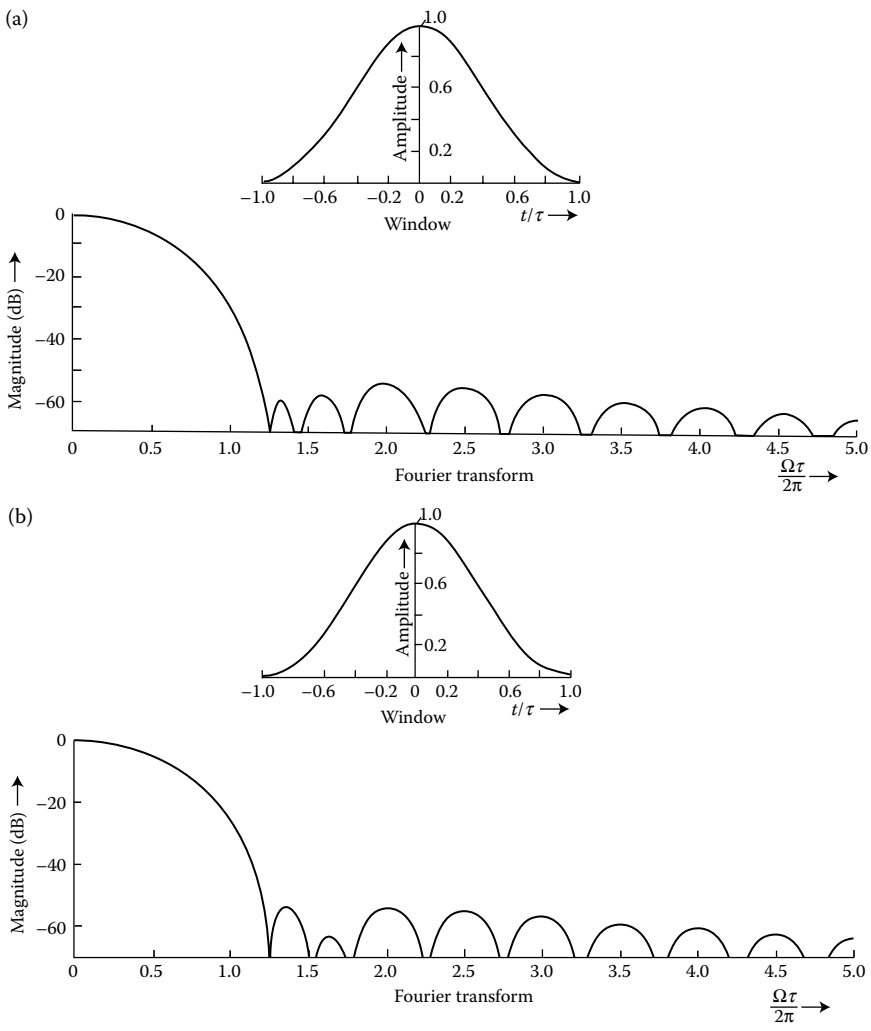
and

$$\left| \frac{F_{sc}(0)}{F_{sc}(j\Omega_p)} \right| = \text{Maximum}, \quad (3.31)$$

where  $|F_{sc}(j\Omega_1)|$  is the peak magnitude of the first side-lobe of Equation 3.27, that is, its peak magnitude in the interval  $(\frac{5\pi}{2\tau}, \frac{7\pi}{2\tau})$ . Here  $|F_{sc}(j\Omega_p)|$  is the largest peak magnitude of the first two side lobes of Equation 3.27 [5]. The synthesis problem can now proceed in two ways:

- i. Choose the value of  $B$  in Equation 3.29 to satisfy the condition of Equation 3.30.
- ii. Select  $B$  in Equation 3.29 such that it satisfies Equation 3.31.

Consequently, two values of  $B$  are obtained using the conditions in (i) and (ii). Using numerical techniques, the values of  $B$  are determined to be 0.100 and 0.103, respectively, corresponding to the above two conditions [5]. Figures 3.6(a) and (b) show the plots of normalized magnitudes of the Fourier transforms of the sum-cosine window for these values of  $B$ , together with their corresponding time functions. The graphs shown are drawn as a function of the normalized time and frequency parameters.



**FIGURE 3.6** Sum-cosine windows (time- and frequency-domain plots). (a) Sum-cosine window with  $B = 0.100$ . (b) Sum-cosine window with  $B = 0.103$ .

The Fourier transform pair in the case of the sum-cosine window is

$$\begin{aligned} & \left( (1 - 2B) \cos\left(\frac{\pi t}{2\tau}\right) + 2B \cos\left(\frac{3\pi t}{2\tau}\right) \right)_{|t| \leq \tau} \\ & \xleftrightarrow{\mathcal{F}} \frac{[4\Omega^2\tau^3(8B - 1) + 3\pi^2(3 - 8B)]4\pi\tau \cos(\Omega\tau)}{[16\Omega^4\tau^4 - 40\Omega^2\pi^2\tau^2 + 94\pi^4]}. \end{aligned} \quad (3.32)$$

The asymptotic attenuation of side lobes for large  $\Omega$  is given by  $\simeq \frac{\pi(8B-1)}{\Omega^2}$ . The distinct feature of the sum-cosine window is its simple form (similar to that of the Hamming window). From the plots of Figure 3.6(b) and the corresponding results given in Table 3.1, it is clear that further modifications to Equation 3.31 to include more side lobes still yields a value of  $B = 0.103$ . For this family of windows, the side-lobe fall-off rate in both the cases is  $\frac{1}{\Omega^2}$ , which is better than the rectangular window, but not as good as the  $\cos^3(x)$  window discussed earlier. The performance comparison of this window with the near-optimum window is done in Section 3.5 of this chapter.

### 3.3.9 $\cos^4(x)$ Window

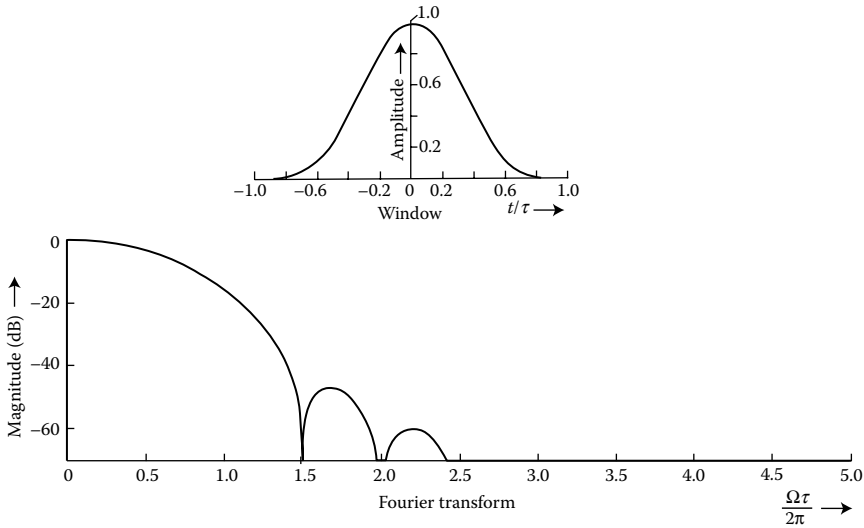
It should be noted that this window is the product of two Hann windows [2]. Therefore, the  $\cos^4(x)$  window (see Figure 3.7) is defined by

$$f(t) = \begin{cases} 0.375 + 0.5 \cos\left(\frac{\pi t}{\tau}\right) + 0.125 \cos\left(\frac{2\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere,} \end{cases} \quad (3.33)$$

whose Fourier transform can be expressed as

$$\begin{aligned} F(j\Omega) = & 0.75 \frac{\sin(\Omega\tau)}{\Omega} + 0.5 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right] \\ & + 0.125 \left[ \frac{\sin\left(\left(\Omega + \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{2\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{2\pi}{\tau}\right)} \right], \quad -\infty < \Omega < \infty. \end{aligned} \quad (3.34)$$

The plots for this window function are shown in Figure 3.7. The side lobes fall at a rate of  $\frac{1}{\Omega^3}$  from the main lobe. However, the FSL of this window is only about  $-47$  dB and the NHMLW is 1.5.



**FIGURE 3.7**  
 $\cos^4(x)$  Window (time- and frequency-domain plots).

**3.3.10 Raised-Cosine Family**

The normalized main-lobe widths of the synthesized window functions [5,7] detailed below fall in between those of the Hamming and sum-cosine windows. This family of window functions is defined as

$$f(t) = \begin{cases} \frac{(1 - 2D)}{2} \left( 1 + \cos\left(\frac{\pi t}{\tau}\right) \right) + 2D \cos\left(\frac{2\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere,} \end{cases} \quad (3.35)$$

with its Fourier representation given by the expression

$$F(j\Omega) = (1 - 2D) \frac{\sin(\Omega\tau)}{\Omega} + \frac{(1 - 2D)}{2} \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right] + 2D \left[ \frac{\sin\left(\left(\Omega + \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{2\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{2\pi}{\tau}\right)} \right], \quad -\infty < \Omega < \infty. \quad (3.36)$$

The raised-cosine set of windows [7] is synthesized by adopting a procedure similar to the one presented in Section 3.3.8 (for the sum-cosine window) and is detailed below:

Consider the rectangular window

$$f_r(t) = \begin{cases} 1, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.37)$$

and the raised-cosine (Hann) window

$$f_{rc}(t) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.38)$$

whose Fourier transforms, respectively, are given by

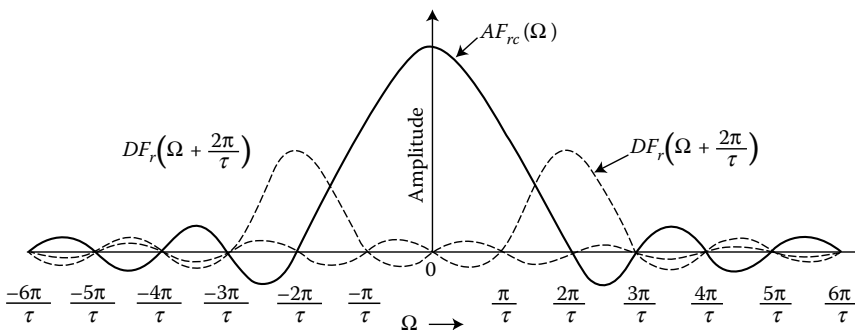
$$F_r(j\Omega) = \frac{2 \sin(\Omega\tau)}{\Omega}, \quad -\infty < \Omega < \infty. \quad (3.39)$$

and

$$F_{rc}(j\Omega) = \frac{\sin(\Omega\tau)}{\Omega} + 0.5 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right], \quad -\infty < \Omega < \infty. \quad (3.40)$$

Consider Figure 3.8, which shows the scaled and shifted waveforms of  $AF_{rc}(j\Omega)$ ,  $DF_r(j(\Omega + \frac{2\pi}{\tau}))$  and  $DF_r(j(\Omega - \frac{2\pi}{\tau}))$  [5]. Here,  $A$  and  $D$  are real constants. Adding these terms, we obtain

$$F_{ref}(j\Omega) = AF_{rc}(j\Omega) + DF_r\left[j\left(\Omega + \frac{2\pi}{\tau}\right)\right] + DF_r\left[j\left(\Omega - \frac{2\pi}{\tau}\right)\right], \quad (3.41)$$



**FIGURE 3.8**  
Fourier transforms of raised-cosine and rectangular windows.

whose inverse Fourier transform can be shown to be

$$f_{ref}(t) = \frac{A}{2} \left[ 1 + \cos \left( \frac{\pi t}{\tau} \right) \right] + 2D \cos \left( \frac{2\pi t}{\tau} \right), \quad |t| \leq \tau. \quad (3.42)$$

By restricting the function  $f_{ref}(t)$  to be unity at the origin (i.e.,  $(A + 2D) = 1$ ), which is in accordance with the basic properties of windows, we slightly modify Equation 3.42 as follows:

$$f_{ref}(t) = (0.5 - D) \left[ 1 + \cos \left( \frac{\pi t}{\tau} \right) \right] + 2D \cos \left( \frac{2\pi t}{\tau} \right), \quad |t| \leq \tau. \quad (3.43)$$

We now need a scheme to choose an optimum value of  $D$  [5]. Let  $\phi$  be the NHMLW of the window to be synthesized. Select  $D$  in Equation 3.43 in accordance with the following set of conditions [5]. Let  $F_{ref}(j\Omega)$  be the Fourier transform of  $f_{ref}(t)$ .

- i. The difference between the first zero of  $F_{ref}(j\Omega)$  and  $\phi$  is less than a small specified value  $\delta$ .
- ii.

$$\left| \frac{F_{ref}(0)}{F_{ref}(j\Omega_p)} \right| = \text{Maximum}, \quad (3.44)$$

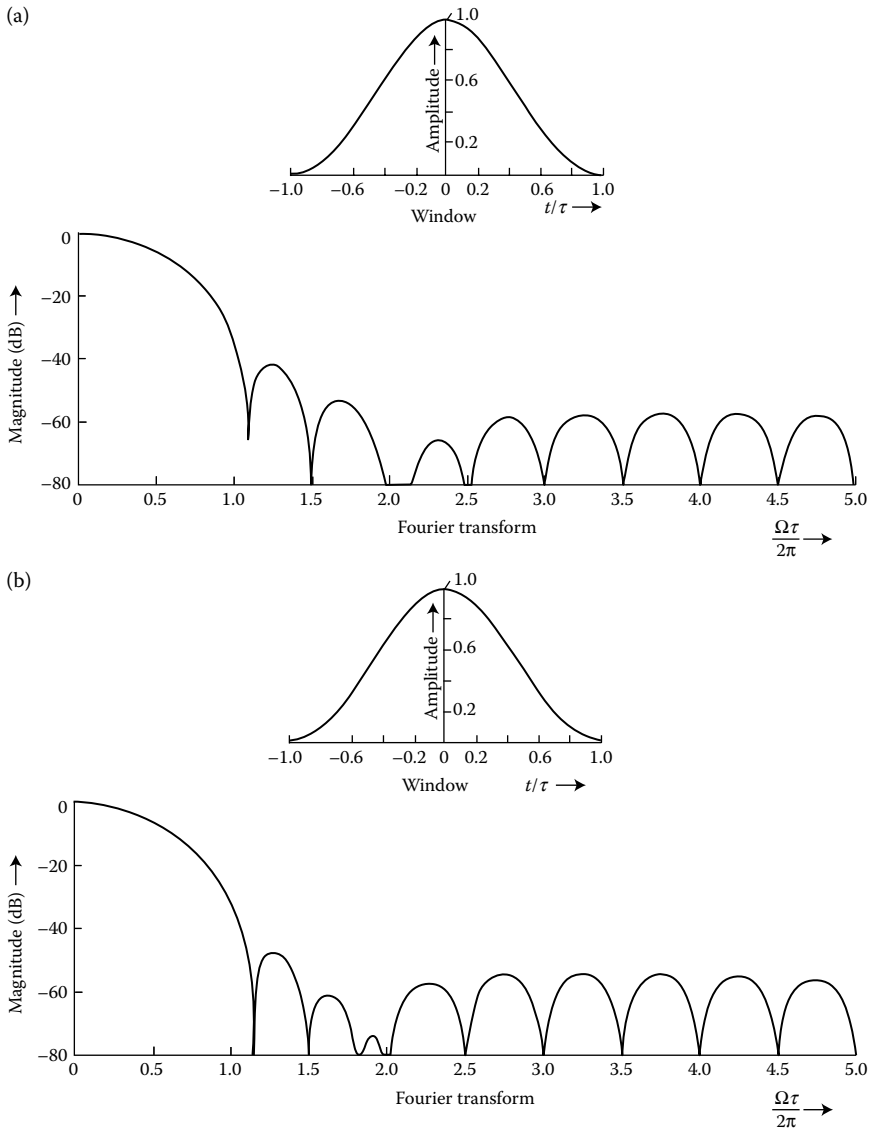
where  $F_{ref}(j\Omega_p)$  is the peak magnitude of  $F_{ref}(j\Omega)$  in the interval  $(\phi, \frac{5\pi}{\tau})$ .

Using numerical techniques, the values of  $D$  are determined for the normalized values of  $\phi = 1.10, 1.15, 1.20,$  and  $1.245$  (while  $\delta = 0.005$ ) [5]. This set of windows is named as raised-cosine family, since one of the functions used is the raised-cosine pulse. The plots of Figures 3.9(a) through (d) depict the normalized magnitudes of the Fourier transforms of raised-cosine family for four different values  $D = 0.0113, 0.0138, 0.0155,$  and  $0.0165$ , respectively, together with their corresponding normalized time functions. For the above-mentioned values of  $D$ , the NHMLW are  $1.10, 1.15, 1.20,$  and  $1.245$ , respectively.

The Fourier transform pair of this family of window is given as follows:

$$\begin{aligned} & \left( (0.5 - D) \left[ 1 + \cos \left( \frac{\pi t}{\tau} \right) \right] + 2D \cos \left( \frac{2\pi t}{\tau} \right) \right)_{|t| \leq \tau} \\ & \xleftrightarrow{\mathcal{F}} \frac{[4\pi^4(1 - 2D) - \pi^2\Omega^2\tau^2(1 + 2D) + 4D\Omega^4\tau^4] \sin(\Omega\tau)}{\Omega(\Omega^4\tau^4 - 5\pi^2\Omega^2\tau^2 + 4\pi^4)}. \end{aligned} \quad (3.45)$$

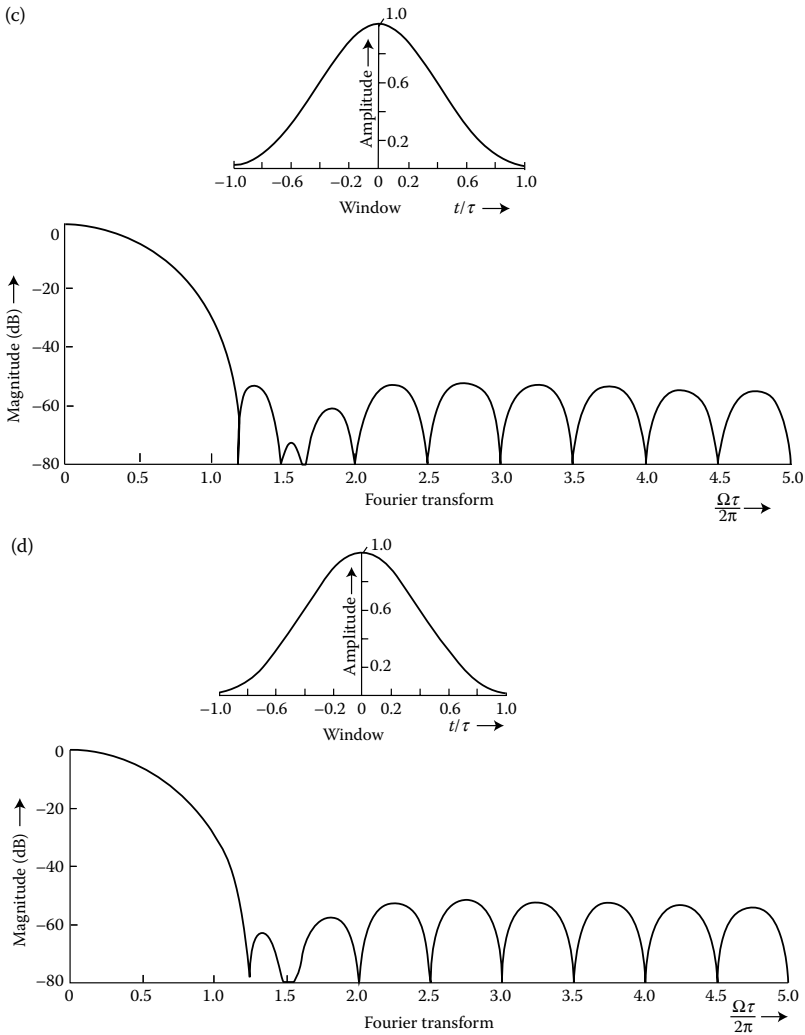
The asymptotic attenuation of  $F(j\Omega)$  for large values of  $\Omega$  is given by  $(\frac{4D}{\Omega})$ . From Equation 3.43, it is clear that the Hann window is a special case of the raised-cosine family when  $D = 0$ . The above family of windows, apart



**FIGURE 3.9** Raised-cosine window family (time- and frequency-domain plots). (a) Raised-cosine window with  $D = 0.0113$ . (b) Raised-cosine window with  $D = 0.0138$ .

from being simple in form, has the slight advantage of obtaining a variable main-lobe width (though to a limited extent) by choosing different values of the constant  $D$ . A comparison of how this family of windows performs with respect to the near-optimum window family is made in Section 3.5.





**FIGURE 3.9** (Continued). Raised-cosine window family (time- and frequency-domain plots). (c) Raised-cosine window with  $D = 0.0155$ . (d) Raised-cosine window with  $D = 0.0165$ .

### 3.3.11 Blackman Window

The Blackman window [1,2] is defined by the following Fourier transformation pair:

$$f(t) = \begin{cases} 0.42 + 0.5 \cos\left(\frac{\pi t}{\tau}\right) + 0.08 \cos\left(\frac{2\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.46)$$

and

$$\begin{aligned}
 F(j\Omega) = & 0.84 \left[ \frac{\sin(\Omega\tau)}{\Omega} \right] + 0.5 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right] \\
 & + 0.08 \left[ \frac{\sin\left(\left(\Omega + \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{2\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{2\pi}{\tau}\right)} \right], \quad -\infty < \Omega < \infty.
 \end{aligned} \tag{3.47}$$

The Fourier transform pair of the Blackman window can be given as follows:

$$\begin{aligned}
 & \left( 0.42 + 0.5 \cos\left(\frac{\pi t}{\tau}\right) + 0.08 \cos\left(\frac{2\pi t}{\tau}\right) \right)_{|t| \leq \tau} \\
 & \xleftrightarrow{\mathcal{F}} \frac{(3.36\pi^4 - 0.36\pi^2\Omega^2\tau^2) \sin(\Omega\tau)}{\Omega(\Omega^4\tau^4 - 5\pi^2\Omega^2\tau^2 + 4\pi^4)}.
 \end{aligned} \tag{3.48}$$

Therefore,  $F(j\Omega) \simeq \left(\frac{0.36\pi^2}{\Omega^3\tau^2}\right)$  as  $\Omega \rightarrow \infty$ . The plots for this window are shown in Figure 3.10(a). We note here that the MSLR of this window is as low as 0.001 of the main-lobe peak (which is about  $-60$  dB). The side-lobe fall-off rate is  $\frac{1}{\Omega^3}$  (see Equation 3.48). However, the main-lobe width of this window is thrice that of the rectangular window. This window can be considered as a special case of the  $\cos^4(x)$  window which was described in Section 3.3.9.

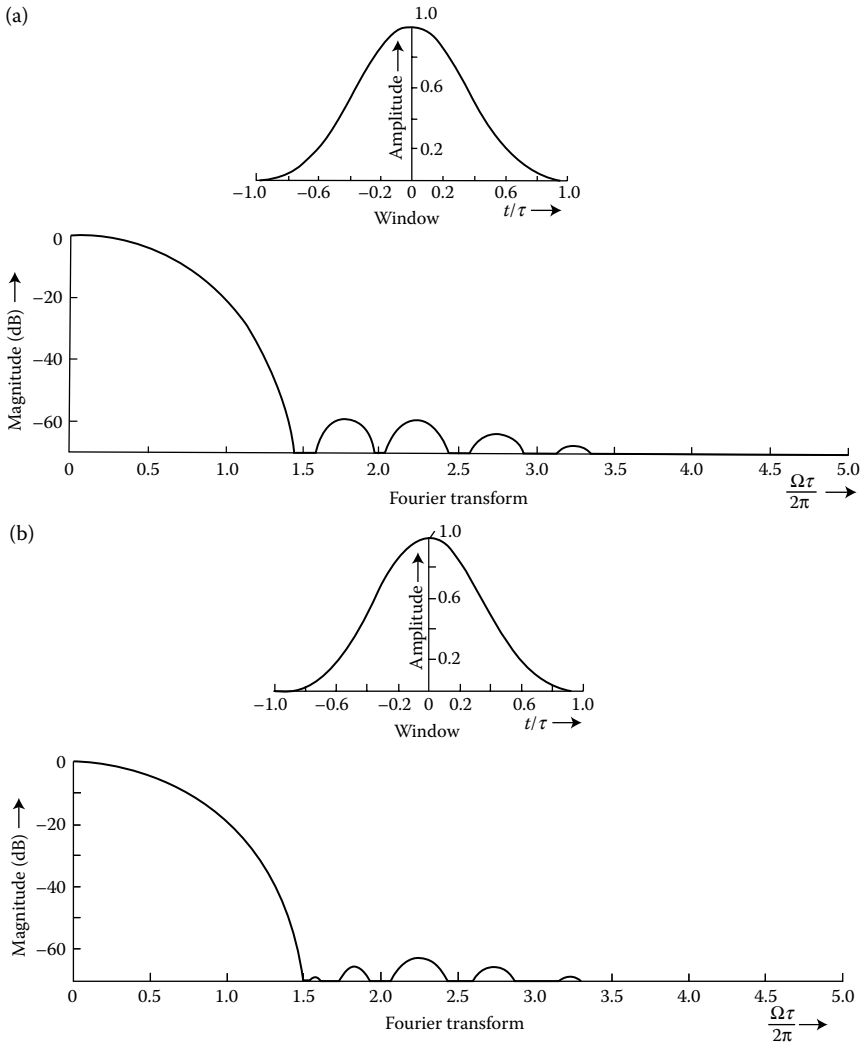
### 3.3.12 Optimized Blackman Window

It can be seen that the window function described in Section 3.3.11 has not been optimized. We discuss the optimization procedure here [5,8]. A generalized form of the Blackman window [1,2] can be represented by

$$f(t) = \begin{cases} A + 2B \cos\left(\frac{\pi t}{\tau}\right) + 2C \cos\left(\frac{2\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \tag{3.49}$$

where  $A$ ,  $B$ , and  $C$  are real constants. Restricting the function to be unity at the origin, but keeping the constant  $B = 0.25$ , Equation 3.49 gives

$$f(t) = (0.5 - 2C) + 0.5 \cos\left(\frac{\pi t}{\tau}\right) + 2C \cos\left(\frac{2\pi t}{\tau}\right), \quad |t| \leq \tau \tag{3.50}$$



**FIGURE 3.10** Blackman window functions (time- and frequency-domain plots). (a) Blackman window. (b) Optimized Blackman window.

whose Fourier transform is

$$\begin{aligned}
 F(j\Omega) = & (1 - 4C) \frac{\sin(\Omega\tau)}{\Omega} + 0.5 \left[ \frac{\sin\left(\left(\Omega + \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{\pi}{\tau}\right)} \right] \\
 & + 2C \left[ \frac{\sin\left(\left(\Omega + \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega + \frac{2\pi}{\tau}\right)} + \frac{\sin\left(\left(\Omega - \frac{2\pi}{\tau}\right)\tau\right)}{\left(\Omega - \frac{2\pi}{\tau}\right)} \right], \quad -\infty < \Omega < \infty.
 \end{aligned}
 \tag{3.51}$$

We set the coefficient  $B = 0.25$  (which is the same as in the case of the Blackman window) itself, since it is easier to implement this coefficient in the frequency-domain [5]. The implementation of windows is discussed in Chapter 6. The optimization of the constant  $C$  is performed, such that the ratio

$$\left| \frac{F(0)}{F(j\Omega_1)} \right| = \text{Maximum}, \quad (3.52)$$

where  $F(j\Omega_1)$  is the peak magnitude of the first side lobe [5]. Using numerical techniques, the value of  $C$  was found to be 0.044, which satisfies Equation 3.52. The important parameters of the Blackman and the optimized Blackman windows are provided in Table 3.1. Figure 3.10(b) shows the plots of the normalized log-magnitude of  $F(j\Omega)$  as a function of the normalized frequency, along with the normalized time function of the window.

The results given in Table 3.1 show that the new coefficients of the *optimized* Blackman window yield about 10 dB improvement in the FSL and 4.5 dB improvement in the MSL, over that of the Blackman window. However, the main-lobe width remains the same as in the case of the Blackman window. Therefore, if an application demands immediate side-lobe rejection, this optimized window offers a better solution.

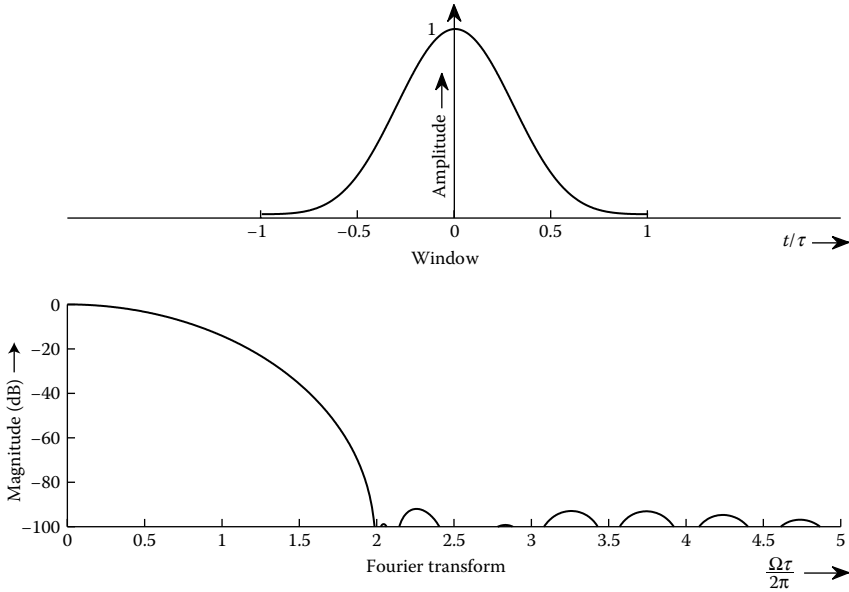
### 3.3.13 Blackman–Harris Window

The Blackman–Harris window [2] is defined as the sum of four terms (as against three terms in the Blackman window). It is described below:

$$f(t) = \begin{cases} 0.35875 + 0.48829 \cos\left(\frac{\pi t}{\tau}\right) \\ \quad + 0.14128 \cos\left(\frac{2\pi t}{\tau}\right) + 0.01168 \cos\left(\frac{3\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.53)$$

whose Fourier transform is given by the expression

$$F(j\Omega) = 0.7175 \frac{\sin(\Omega\tau)}{\Omega} + 0.48829 \left[ \frac{\sin((\Omega + \pi/\tau)\tau)}{(\Omega + \pi/\tau)} + \frac{\sin((\Omega - \pi/\tau)\tau)}{(\Omega - \pi/\tau)} \right] \\ + 0.14128 \left[ \frac{\sin((\Omega + 2\pi/\tau)\tau)}{(\Omega + 2\pi/\tau)} + \frac{\sin((\Omega - 2\pi/\tau)\tau)}{(\Omega - 2\pi/\tau)} \right]$$



**FIGURE 3.11** Blackman–Harris window function (time- and frequency-domain plots).

$$\begin{aligned}
 &+ 0.01168 \left[ \frac{\sin ((\Omega + 3\pi/\tau)\tau)}{(\Omega + 3\pi/\tau)} + \frac{\sin ((\Omega - 3\pi/\tau)\tau)}{(\Omega - 3\pi/\tau)} \right], \\
 &-\infty < \Omega < \infty.
 \end{aligned}
 \tag{3.54}$$

This window achieves a trade-off between the main-lobe width and the side lobe level. It exhibits side lobes just shy of  $-92$  dB from the main lobe. However, the main-lobe width of this window is four times that of the rectangular window, while the side lobes fall at the same rate as the original Blackman window (in Section 3.3.11). The reader can refer to Figure 3.11 for the corresponding plots.

**3.3.14 Parabolic Window**

The parabolic window [2] is similar to the cosine lobe discussed in Section 3.3.3. This window has the time-domain form

$$f(t) = \begin{cases} 1 - \left(\frac{|t|}{\tau}\right)^2, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases}
 \tag{3.55}$$

and the corresponding spectral window is given by

$$F(j\Omega) = \frac{4\tau}{(\Omega\tau)^2} \left( \frac{\sin(\Omega\tau)}{\Omega\tau} \right) - \cos(\Omega\tau). \quad (3.56)$$

This exhibits a discontinuous first derivative at the boundaries (see Section 3.4). Owing to this property, its transform has a fall-off side-lobe rate of the order of  $1/\Omega^2$  only (see Equation 3.56). The FSL of this window is about  $-22$  dB from the main-lobe peak (see Figure 3.12(a)).

### 3.3.15 Papoulis Window

The time-limited function of the Papoulis window [9] is represented by

$$f(t) = \begin{cases} \frac{1}{\pi} \left| \sin\left(\frac{\pi t}{\tau}\right) \right| + \left(1 - \frac{|t|}{\tau}\right) \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.57)$$

whose Fourier transform yields the following optimum spectral window:

$$F(j\Omega) = 4\pi^2\tau \frac{(1 + \cos \Omega\tau)}{(\pi^2 - \Omega^2\tau^2)^2}, \quad -\infty < \Omega < \infty. \quad (3.58)$$

In our context, the “optimum” window is meant in the sense that it has the largest energy content in the main lobe of its Fourier transform (similar to Kaiser’s modified zeroth-order Bessel window function family, to be discussed in Section 3.3.19), while the side lobes contain less energy. The main-lobe width of this window is almost the same as that of the Blackman window. The FSL of this window is  $-46$  dB. Nevertheless, the side lobes fall at a much faster rate of  $\frac{1}{\Omega^4}$  (since  $F(j\Omega) \simeq \left(\frac{4\pi^2}{\Omega^4\tau^3}\right)$ , as  $\Omega \rightarrow \infty$ ) than in any other case, as can be observed from Figure 3.12(b).

### 3.3.16 Tukey Window

This is also known as the cosine-tapered window and can be expressed as a cosine-lobe convolved with a rectangular window. This window [10] is defined as

$$f(t) = \begin{cases} 1, & |t| \leq \beta\tau \\ 0.5 + 0.5 \cos \frac{\pi(|t| - \beta\tau)}{(1 - \beta)\tau}, & \beta\tau \leq |t| \leq \tau \\ 0, & \text{elsewhere.} \end{cases} \quad (3.59)$$

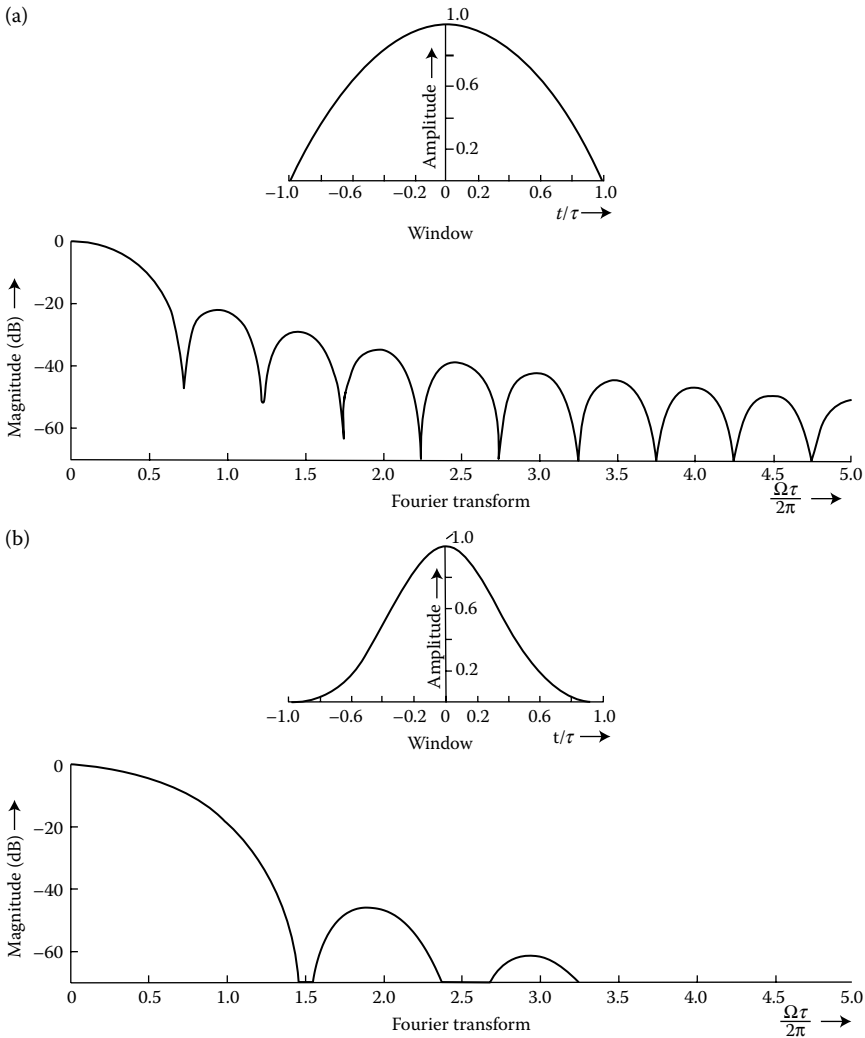


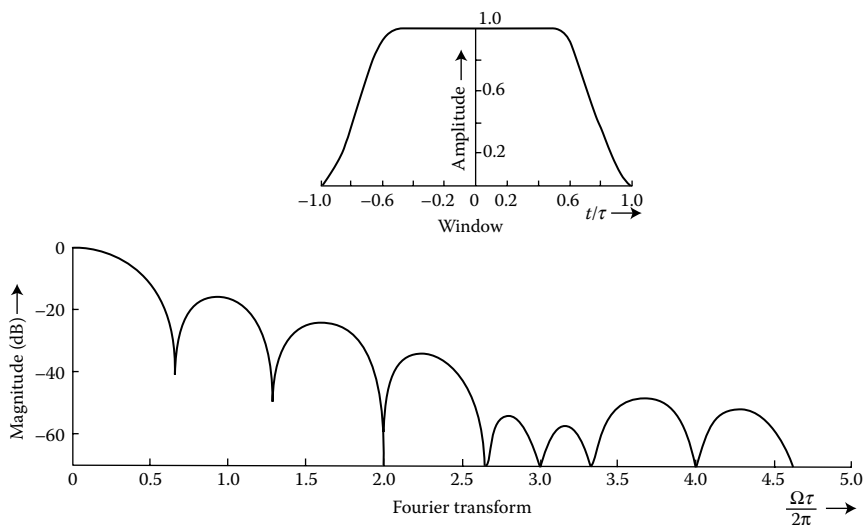
FIGURE 3.12

Window functions (time- and frequency-domain plots). (a) Parabolic window. (b) Papoulis window.

The resultant Fourier transform is consequently the product of two individual transforms given below:

$$F(j\Omega) = \frac{\sin [\Omega(1 + \beta)/2] \cos [\Omega(1 - \beta)/2]}{\Omega [1 - (1 - \beta)^2(\Omega/\pi)^2]}. \tag{3.60}$$

This window represents an attempt to smoothly set the data to zero at the boundaries (see Figure 3.13). The value of  $\beta$  used in both the time- and



**FIGURE 3.13** Tukey window function (time- and frequency-domain plots, with  $\beta = 0.5$ ).

frequency-domain plots is 0.5. The window evolves from the rectangular to the Hann window, as the parameter  $\beta$  varies from zero to unity ( $0 \leq \beta \leq 1$ ) (Figure 3.13).

### 3.3.17 Parzen (Jackson) Window

The Parzen window [10] function is defined in three different ranges and is presented as follows:

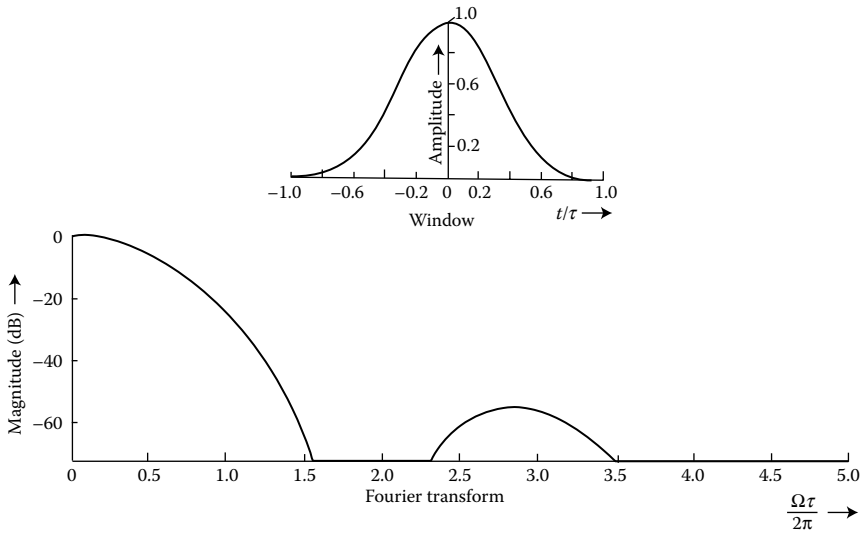
$$f(t) = \begin{cases} 1 - 6 \left(\frac{|t|}{\tau}\right)^2 \left(1 - \frac{|t|}{\tau}\right), & |t| \leq \frac{\tau}{2} \\ 2 \left(1 - \frac{|t|}{\tau}\right)^3, & \frac{\tau}{2} \leq |t| \leq \tau \\ 0, & \text{elsewhere.} \end{cases} \quad (3.61)$$

Its Fourier transform can be shown to be

$$F(j\Omega) = \frac{3\tau}{4} \left[ \frac{\sin\left(\frac{\Omega\tau}{4}\right)}{\frac{\Omega\tau}{4}} \right]^4, \quad -\infty < \Omega < \infty. \quad (3.62)$$

The NHMLW of this window is two, which is four times that of the rectangular window. The side-lobe fall-off rate is  $\frac{1}{\Omega^4}$  (since  $F(j\Omega) \simeq \left(\frac{192}{\Omega^4\tau^3}\right)$  as





**FIGURE 3.14** Parzen window function (time- and frequency-domain plots).

$\Omega \rightarrow \infty$ ), similar to the Papoulis window discussed in Section 3.3.15 (see Figures 3.12(b) and 3.14 for details). Since the side-lobe fall-off rate of the Parzen window is quite rapid, we cannot observe other side lobes (beyond the first) in Figure 3.14. It is a nonnegative window and it is similar to the Bartlett window by virtue of its self-convolution construction.

### 3.3.18 Dolph–Chebyshev Window

The class of window functions with the minimum main-lobe width for a given side-lobe amplitude is known as the continuous-time Dolph–Chebyshev weighting functions [11]. The Fourier transform of the weights is chosen to be the Dolph–Chebyshev function given by

$$F(j\Omega) = \cos \left[ P \cos^{-1} \left( \frac{\cos \pi \Omega}{\cos \pi B/2} \right) \right], \tag{3.63}$$

where  $\Omega$  is the normalized frequency, such that  $|\Omega| \leq 1/2$ . Then the side lobes of the Fourier transform of any sine wave in the input will be *minimax* [12]. In Equation 3.63,  $P$  is one less than the number of weights and  $B$  is the normalized bandwidth [13]. The discrete-time Dolph–Chebyshev window function is discussed in Chapter 5 (Section 5.2.23).

### 3.3.19 Kaiser's Modified Zeroth-Order Bessel Window Function Family

*Optimum window function:* The function  $f(t)$ , whose CTFT is  $F(j\Omega)$  that maximizes the energy inside some selected frequency interval  $(-\Omega_1, \Omega_1)$  with respect to the total energy can be represented by

$$\frac{\int_{-\Omega_1}^{\Omega_1} |F(j\Omega)|^2 d\Omega}{\int_{-\infty}^{\infty} |F(j\Omega)|^2 d\Omega} = \text{Maximum.} \quad (3.64)$$

Such a function  $f(t)$  is called the optimum window function. A window which obeys such a condition was derived by Slepian and Pollak (1961) and is called *prolate-spheroidal* wave function [14]. However, the two sets of modified Bessel window function families, namely, the modified zeroth-order (this section) and the modified first-order Kaiser–Bessel (Section 3.3.20), are simple approximations to these quite complicated functions. These windows were proposed by Kaiser [15] and are described in this section and the next.

Kaiser has introduced a set of windows that are relatively simple but closely approximate the zeroth-order prolate-spheroidal wave functions [14], which are known to be optimum spectral windows. The two sets of Kaiser windows described here and in Section 3.3.20 are known to be *near-optimum*. The first family of time-limited functions suggested by Kaiser is given by the Fourier transform pair [15]

$$f(t) = \begin{cases} \frac{I_0 \left[ \alpha \sqrt{1 - \left( \frac{t}{\tau} \right)^2} \right]}{I_0(\alpha)}, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases} \quad (3.65)$$

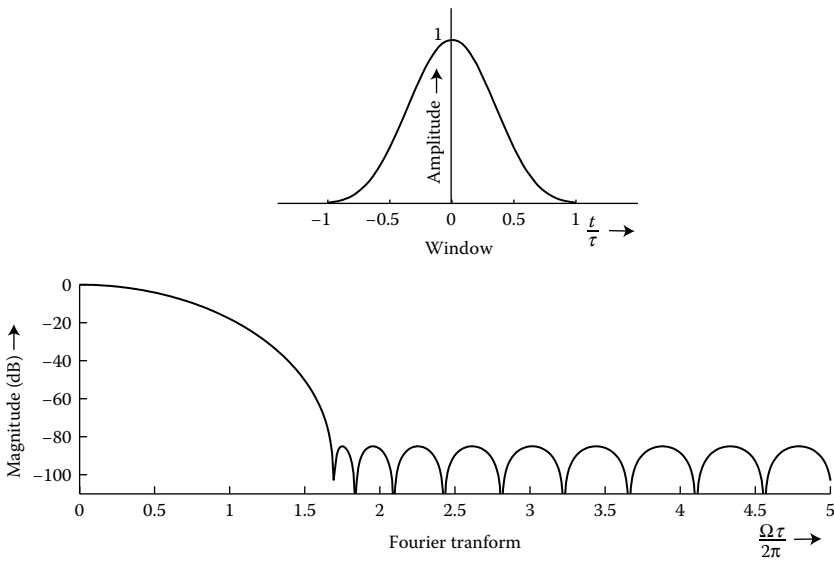
and

$$F(j\Omega) = \frac{\sinh \left[ \alpha \sqrt{1 - \left( \frac{\Omega}{\alpha} \right)^2} \right]}{(\sinh \alpha) \sqrt{1 - \left( \frac{\Omega}{\alpha} \right)^2}}, \quad -\infty < \Omega < \infty, \quad (3.66)$$

where  $I_0(x)$  is the modified zeroth-order Bessel function of the first kind and  $\alpha$  is a variable parameter. The function  $I_0(x)$  can be generated by means of the rapidly convergent series approximation:

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left[ \frac{1}{k!} \left( \frac{x}{2} \right)^k \right]^2. \quad (3.67)$$

In practice, we require no more than 15–25 terms of this series. It can be shown that by adjusting the parameter  $\alpha$  within the usual range of  $4 \leq \alpha \leq 9$



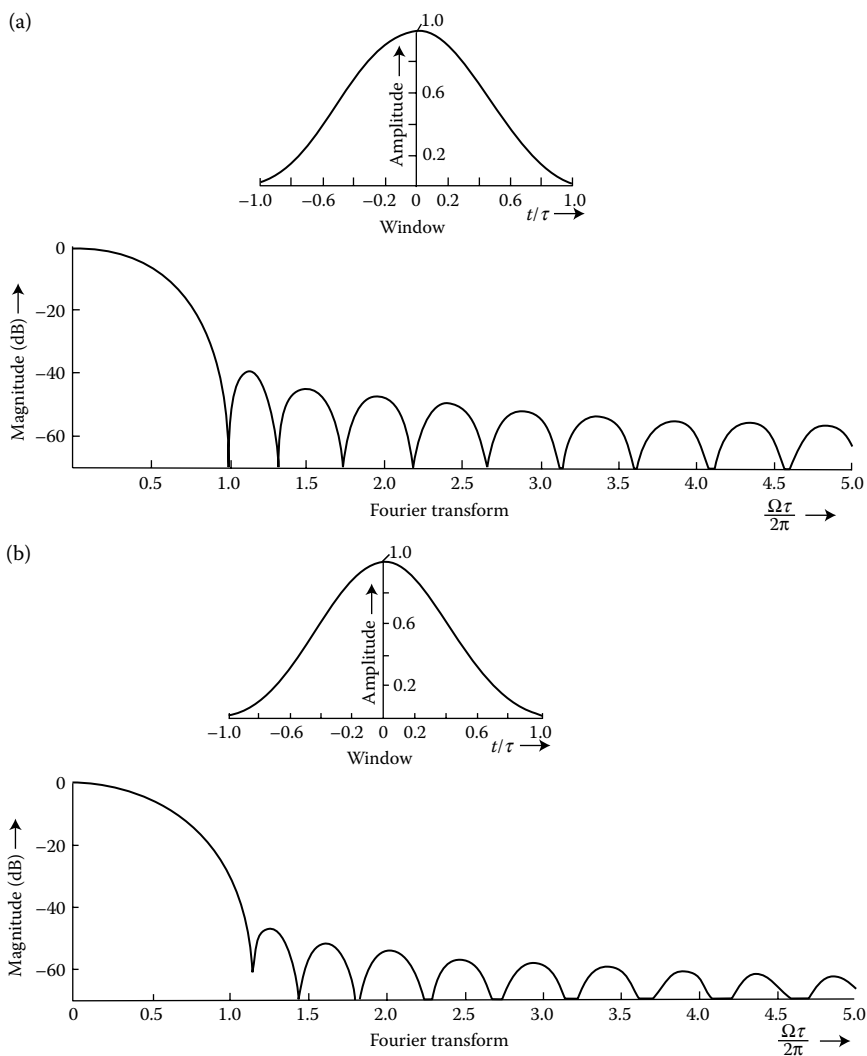
**FIGURE 3.15** Dolph-Chebyshev window (time- and frequency-domain plots).

for optimum results, the side-lobe levels can be minimized at the expense of the main-lobe widths [15–17]. The above-mentioned choice of  $\alpha$  values correspond to a range of maximum side-lobe amplitudes of about 30 dB down to 67 dB, with respect to the main-lobe peak. We have provided the plots for this window family for two different values of  $\alpha$  (see Figure 3.16). For this family of windows, the side-lobes fall at a rate of  $-6\text{dB/octave}$ . The figures presented give an idea of how the behavior of the window parameters changes with the  $\alpha$  values. Since the modified zeroth-order Bessel window family closely approximates the zeroth-order prolate-spheroidal wave functions (which are proven to be optimum) the zeroth-order Bessel family is also called the *near-optimum* window function family. However, the computational complexity of the modified zeroth-order Bessel function is relatively low, when compared to prolate-spheroidal wave functions [15]. The results of this window function family are summarized in Table 3.2.

**3.3.20 Kaiser’s Modified First-Order Bessel Window Function Family**

The second set of windows again propounded by Kaiser [15] is given by the time function

$$f(t) = \begin{cases} \frac{I_1 \left[ \gamma \sqrt{1 - \left(\frac{t}{\tau}\right)^2} \right]}{I_1(\gamma) \sqrt{1 - \left(\frac{t}{\tau}\right)^2}}, & |t| \leq \tau \\ 0, & \text{elsewhere,} \end{cases} \tag{3.68}$$



**FIGURE 3.16** Window functions (time- and frequency-domain plots). (a) Modified Bessel window of zeroth-order with  $\alpha = 5.4413981$ . (b) Modified Bessel window of zeroth-order with  $\alpha = 6.5$ .

and its corresponding Fourier transform is given by

$$F(j\Omega) = \frac{\cosh \left[ \gamma \sqrt{1 - \left( \frac{\Omega}{\gamma} \right)^2} \right] - \cos \Omega\tau}{(\cosh \gamma - 1)}, \quad -\infty < \Omega < \infty. \quad (3.69)$$

**TABLE 3.2**

Parameters of Kaiser’s Modified Zeroth-Order Bessel Family

$\alpha$ Value	Normalized			Ratio of Main-Lobe
	Half Main-Lobe Width (NHMLW)	First Side-Lobe Level (FSL) (dB)	Maximum Side-Lobe Level (MSLL) (dB)	Energy to Total Energy (MLE)
5	0.94	-36.73	-36.73	0.999741
$\pi\sqrt{3}$	1.0	-39.79	-39.79	0.999881
6	1.078	-43.82	-43.82	0.999956
$\pi\sqrt{3.84}$	1.100	-44.93	-44.93	0.9999654
6.5	1.149	-47.44	-47.44	0.999982
$\pi\sqrt{4.29}$	1.150	-47.49	-47.49	0.9999819
$\pi\sqrt{4.76}$	1.200	-50.07	-50.07	0.9999894
7	1.221	-51.15	-51.15	0.999993
$\pi\sqrt{5.25}$	1.250	-52.62	-52.62	0.9999944
7.5	1.294	-54.93	-54.93	0.999997
8	1.368	-58.67	-58.67	0.9999989
8.5	1.442	-62.55	-62.55	0.99999996

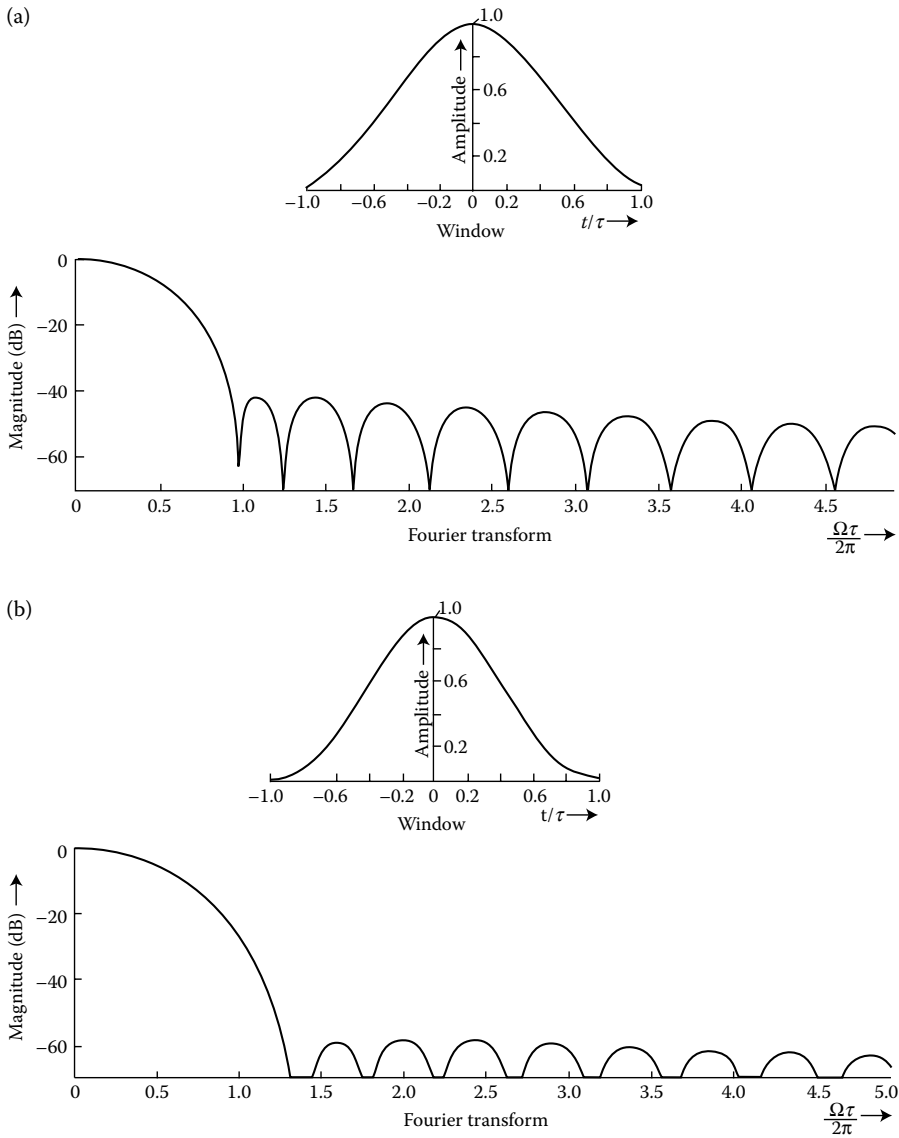
Here,  $I_1(x)$  represents the modified first-order Bessel function of the first kind and it is defined as

$$I_1(x) = \sum_{m=0}^{\infty} \frac{1}{m!(m + 1)!} \left(\frac{x}{2}\right)^{2m+1}. \tag{3.70}$$

Again, the side lobes can be varied by choosing different values of the window function parameter,  $\gamma$  [15]. The time- and frequency-domain plots of the first-order Kaiser–Bessel family for two different values of  $\gamma$  are shown in Figure 3.17. However, of the two families proposed by Kaiser, the modified zeroth-order Bessel family is closer to the optimum zeroth-order prolate-spheroidal wave functions [14]. The modified first-order Bessel family has the slight advantage of smaller first side lobes when compared to either zeroth-order Bessel window family or prolate-spheroidal wave functions, but its side-lobe fall-off rate is slower (see Figures 3.16 and 3.17) [15]. These observations can be easily verified from the plots given in Figures 3.16 and 3.17 as well as from Tables 3.2 and 3.3.

### 3.4 Rate of Fall-Off Side-Lobe Level

This is one of the vital parameters associated with the Fourier transform of a window in detecting weak harmonics. A theorem relating the time function  $f(t)$  and the RFSLL is stated and proved in the next section.



**FIGURE 3.17** Window functions (time- and frequency-domain plots). (a) Modified Bessel window of first-order with  $\gamma = 6.1296883$ . (b) Modified Bessel window of first-order with  $\gamma = 8.0451893$ .

### 3.4.1 Theorem

If the time function of a window  $f(t)$  is continuous and bounded for the *first*  $n$  derivatives, then its side lobes will fall off at the rate of  $\frac{1}{\Omega^{n+1}}$  [18].

**TABLE 3.3**

Parameters of Kaiser’s Modified First-Order Bessel Family

$\gamma$ Value	Normalized			Ratio of Main-Lobe
	Half Main-Lobe Width (NHMLW)	First Side-Lobe Level (FSLL) (dB)	Maximum Side-Lobe Level (MSLL) (dB)	Energy to Total Energy (MLE)
6.129688	0.976	-41.84	-41.79	0.999789
6.565842	1.046	-46.91	-45.14	0.999902
7.103980	1.139	-54.43	-49.67	0.999962
7.576740	1.227	-62.67	-54.05	0.999984
8.045189	1.320	-58.86	-57.88	0.999993
8.513068	1.418	-64.25	-61.93	0.999997
8.983568	1.511	-70.54	-66.08	0.9999986
9.458593	1.567	-78.21	-70.12	0.9999993

*Proof:*

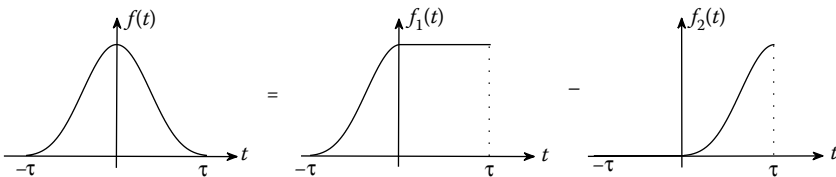
This theorem can be proved for the first derivative and can be extended to higher derivatives using the Fourier transform properties. If a function  $f(t)$  of bounded variation is Riemann integrable, then its transform  $F(j\Omega)$  falls at least as fast as  $\frac{1}{\Omega}$ .

$$\begin{aligned}
 F(j\Omega) &= \int_{-\tau}^{\tau} f(t)e^{-j\Omega t} dt \\
 &= \int_{-\tau}^{\tau} f(t) \cos(\Omega t) dt - j \int_{-\tau}^{\tau} f(t) \sin(\Omega t) dt.
 \end{aligned}
 \tag{3.71}$$

If  $f(t)$  is a monotonically increasing (or decreasing) function, then

$$\int_a^b f(t)g(t) dt = f(a) \int_a^\epsilon g(t) dt + f(b) \int_\epsilon^b g(t) dt,
 \tag{3.72}$$

where  $g(t)$  can be any arbitrary function. The window function  $f(t)$  can be written as the difference of two monotonically increasing functions,  $f_1(t)$  and  $f_2(t)$  as depicted in Figure 3.18. Hence, all the windows will satisfy



**FIGURE 3.18**

Decomposition of a window function into two monotonically increasing functions.

Equation 3.72. We consider only the first term, that is,  $\int_{-\tau}^{\tau} f(t) \cos(\Omega t) dt$  of Equation 3.71. Using Equation 3.72, we can rewrite the first term as

$$\int_{-\tau}^{\tau} f_1(t) \cos(\Omega t) dt = f_1(-\tau) \int_{-\tau}^{\varepsilon} \cos(\Omega t) dt + f_1(\tau) \int_{\varepsilon}^{\tau} \cos(\Omega t) dt \quad (3.73)$$

$$\left| \int_{\alpha}^{\beta} \cos(\Omega t) dt \right| = \frac{|\sin(\Omega \alpha) - \sin(\Omega \beta)|}{|\Omega|} \leq \frac{2}{|\Omega|} \quad (3.74)$$

$$\left| \int_{-\tau}^{\tau} f_1(t) \cos(\Omega t) dt \right| \leq \frac{4M}{|\Omega|}, \text{ where } M = \max\{f_1(-\tau), f_1(\tau)\}. \quad (3.75)$$

Now, considering both  $f_1(t)$  and  $f_2(t)$ , we obtain

$$\left| \int_{-\tau}^{\tau} f(t) \cos(\Omega t) dt \right| \leq \frac{4M}{|\Omega|}, \text{ where } M = \max\{f_1(-\tau), f_1(\tau), f_2(-\tau), f_2(\tau)\}. \quad (3.76)$$

Following a similar approach for the second term with  $\sin(\Omega t)$ , we obtain

$$\left| \int_{-\tau}^{\tau} f(t) \sin(\Omega t) dt \right| \leq \frac{4M}{|\Omega|}. \quad (3.77)$$

From Equations 3.76 and 3.77, it is clear that  $F(j\Omega)$  falls at least as fast as  $\frac{1}{\Omega}$ .

If the first derivative is bounded, then  $f^{(1)}(t)$  falls at the rate of  $\frac{1}{\Omega}$ , which in turn implies that  $F(j\Omega)$  falls at the rate of  $\frac{1}{\Omega^2}$  (since the Fourier transform of  $f^{(1)}(t)$  is  $j\Omega F(j\Omega)$ ). Therefore, if the function  $f(t)$  and its  $n$  derivatives are of bounded variation, then the above procedure can be repeated upto the  $n$ th derivative to obtain  $F(j\Omega)$ , which tends to zero at least as fast as  $\frac{1}{\Omega^{n+1}}$ , as  $|\Omega| \rightarrow \infty$ .

### 3.4.2 Side-Lobe Fall-Off Rate in the Time-domain

We now look at the computation of side-lobe fall-off rate in the time-domain [19]. The continuous-time version of a generalized window function of interest sometimes can be expressed in a generalized form as

$$f(t) = \sum_{k=0}^K a_k \cos\left(\frac{\pi k t}{\tau}\right), \quad |t| \leq \tau \quad (3.78)$$

where  $a_k$  represents real constants. Without the loss of generality,  $f(0)$  can be set to be unity. In such a case, all the coefficients of a window will add up to unity as follows:

$$\sum_{k=0}^K a_k = 1. \quad (3.79)$$



We find from Equation 3.78 that

$$f(\pm\tau) = \lim_{|t| \rightarrow \tau} f(t) = \sum_{k=0}^K (-1)^k a_k. \quad (3.80)$$

If  $f(\pm\tau)$  is nonzero, then the weighting function  $f(t)$  is said to be discontinuous at  $t = \pm\tau$  and hence the asymptotic decay rate of  $F(j\Omega)$  will be according to  $\frac{1}{\Omega}$  for large  $\Omega$ . Conversely, if Equation 3.80 is zero, then  $f(t)$  is continuous for all  $t$ . Furthermore,  $f^{(1)}(t)$  is continuous for all  $t$ , since we have

$$f^{(1)}(t) = -\frac{\pi}{\tau} \sum_{k=0}^K k a_k \sin\left(\frac{\pi k t}{\tau}\right), \quad |t| < \tau, \quad (3.81)$$

and

$$\lim_{|t| \rightarrow \tau} f^{(1)}(t) = 0; \quad f^{(1)}(t) = 0, \quad |t| > \tau. \quad (3.82)$$

Thus, when the weighting function values  $f(\pm\tau)$  in Equation 3.80 are zero,  $f(t)$  and  $f^{(1)}(t)$  are both continuous for all  $t$ .

However,  $f^{(2)}(t)$  may not be continuous at  $t = \pm\tau$ . Therefore, we have from Equation 3.81 that

$$f^{(2)}(t) = -\frac{\pi^2}{\tau^2} \sum_{k=0}^K k^2 a_k \cos\left(\frac{\pi k t}{\tau}\right), \quad |t| < \tau, \quad (3.83)$$

and

$$\lim_{|t| \rightarrow \tau} f^{(2)}(t) = -\frac{\pi^2}{\tau^2} \sum_{k=0}^K (-1)^k k^2 a_k. \quad (3.84)$$

If Equation 3.84 is not zero, then  $f^{(2)}(t)$  is discontinuous at  $t = \pm\tau$  and  $F(j\Omega)$  will decay as  $\frac{1}{\Omega^3}$ , for large values of  $\Omega$ . However, if Equation 3.84 is zero, then  $f^{(2)}(t)$  is continuous for all  $t$ , and it follows that  $f^{(3)}(t)$  is continuous for all  $t$ . Consequently,  $F(j\Omega)$  decays at least as quick as  $\frac{1}{\Omega^5}$  for large  $\Omega$ , following the arguments presented above (similar to Equation 3.84). The side-lobe fall-off rate calculations for the Hamming and Hann windows using this method are presented below:

*Hamming Window:* The Hamming window is defined as

$$f(t) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{otherwise} \end{cases}.$$

Comparing Equation 3.78 with the Hamming window definition, we get

$$K = 1, a_0 = 0.54 \text{ and } a_1 = 0.46$$

The weighting function value at  $t = \pm\tau$  can be calculated using Equation 3.80 as

$$f(\pm\tau) = \lim_{|t| \rightarrow \tau} f(t) = a_0 - a_1 = 0.08. \quad (3.85)$$

The Hamming window has nonzero value at  $t = \pm\tau$ . Therefore, its side lobes decay at the rate of  $\frac{1}{\Omega}$  (i.e.,  $-6$  dB/octave) only.

*Hann Window:* As defined previously, the Hann window function is given by

$$f(t) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{\pi t}{\tau}\right), & |t| \leq \tau \\ 0, & \text{otherwise} \end{cases}.$$

The coefficients  $a_0$  and  $a_1$  in this case are 0.5 (see Equation 3.78).

Using Equation 3.80, we can calculate  $f(t)$  at  $t = \pm\tau$  as

$$f(\pm\tau) = \lim_{|t| \rightarrow \tau} f(t) = a_0 - a_1 = 0. \quad (3.86)$$

Hence the Hann window is continuous at the boundaries.

From Equations 3.81 and 3.82, we saw that the first derivative  $f^{(1)}(t)$  contains only sine terms and is equal to zero at  $t = \pm\tau$ . Therefore,  $f^{(1)}(t)$  is also continuous.

At  $t = \pm\tau$ , the second derivative values can be found using Equation 3.84 as follows:

$$f^{(2)}(\pm\tau) = -\frac{\pi^2}{\tau^2} \sum_{k=0}^K (-1)^k k^2 a_k = \frac{\pi^2}{2\tau^2}. \quad (3.87)$$

$f^{(2)}(t)$  is nonzero at  $t = \pm\tau$ , and therefore its side lobes decay at  $\frac{1}{\Omega^3}$  ( $-18$  dB/octave).

### 3.5 Comparison of Windows

We can now proceed to compare the various windows based on their parameters such as FSLL, MSLL, NHMLW, the ratio of the main-lobe energy to the total energy (MLE), and the RFSLL. The half main-lobe width refers to the width of half the central lobe of the Fourier transform of the window. Since a

window is real and even, its Fourier transform is also real and even. For any comparison to be meaningful, it would be appropriate if one of the parameters of the window is kept constant. Hence, we can use the NHMLW for the purpose of comparison. The ratio of the peak of the FSLL to that of the main lobe is yet another useful parameter. The magnitude of this ratio is referred to as the FSLL. MSLL refers to the ratio between the maximum of the peak magnitudes of all side lobes and the absolute value of the main-lobe peak itself. While the NHMLW is dimensionless (since it is normalized), the MSLL and the FSLL are expressed in decibels (dB), that is  $20 \log_{10} \left| \frac{F(j\Omega)}{F(0)} \right|$ .

The above-mentioned parameters are computed for almost all the windows described in the previous section and are listed in Tables 3.1 through 3.3. To make a proper comparison among different windows, it is necessary to keep the main-lobe widths of the windows the same. We now provide a detailed analysis of the parameters of some interesting windows. Since the modified zeroth-order Bessel windows with variable parameters are near-optimum, it is only natural to compare all the other windows with this family.

To find the value of  $\alpha$  corresponding to a particular value of the NHMLW, the following relationship has been arrived at by the author of this monograph, given by

$$\alpha^2 = (\Omega^2 - \pi^2) \quad (3.88)$$

where  $\Omega = 2\pi Y$  and  $Y$  is the required NHMLW. Therefore, to obtain the value of  $\alpha$  for  $Y = 1$  (NHMLW for the Hann or Hamming case), the value of  $\alpha^2 = (2\pi Y)^2 - \pi^2 = (4\pi^2 - \pi^2) = 3\pi^2$  or  $\alpha = \pi\sqrt{3} = 5.4413981$ . On similar lines, for an NHMLW,  $y = 1.5$ , the value of  $\alpha = \sqrt{(2\pi \times 1.5)^2 - \pi^2} = \sqrt{9\pi^2 - \pi^2} = \sqrt{8\pi^2} = \pi\sqrt{8}$ . Therefore, considering the Hamming window and the modified zeroth-order Bessel window with  $\alpha = 5.4413981$ , we find that both have the same NHMLW of unity (refer Figures 3.3(b) and 3.16(a)). The NHMLWs of the Hann, Hamming, and modified zeroth-order Bessel family with  $\alpha = 5.4413981$  can be verified from Figures 3.2(b), 3.3(b), and 3.16(a), respectively. From Tables 3.1 and 3.2, it is clear that the Hamming window yields the lowest FSLL and the lowest MSLL than that of the corresponding modified zeroth-order Bessel or Hann window. However, by observing the plots of Figures 3.3(b) and Figure 3.16(a), for the Hamming and zeroth-order Bessel window with  $\alpha = 5.4413981$ , respectively, it is seen that Kaiser's window has lower side-lobe peaks from the third lobe onwards. In contrast, the Hamming window (omitting the first two side lobes), continues to oscillate approximately sinusoidally, with slowly diminishing amplitudes. On the other hand, the amplitudes of the side lobes of Kaiser's modified zeroth-order Bessel window diminish much more rapidly. The large main-lobe energy of Kaiser's window explains this faster fall-off rate of the side lobes. Further, it can be shown that the RSFLL is  $\frac{1}{\Omega}$  for both the Hamming and Kaiser's zeroth-order Bessel family. However, in the case of the Hann window, the FSLL is only about  $-31.5$  dB, which is also the MSLL, while the rate of fall-off for

the side lobes is  $\frac{1}{\Omega^3}$  (refer Figure 3.2(b)). This rate of fall-off side lobes is much faster than either the Hamming or modified zeroth-order Bessel window with  $\alpha = \pi\sqrt{3}$ . Kaiser et al. have considered the use of the  $I_0$ -sinh window family for spectral analysis and non-recursive digital filter design, respectively, in Refs. [16,17].

To determine how good the synthesized sum-cosine windows are, we compare them with Kaiser's near-optimum modified zeroth-order Bessel family. For a meaningful comparison, we choose the parameter  $\alpha$ , such that the NHMLW of the Bessel window is also the same as that of the sum-cosine window, which is equal to 1.25. It can be shown that  $\alpha = \pi\sqrt{5.25}$  yields an NHMLW of 1.25 for the modified zeroth-order Bessel window. The important parameters of the sum-cosine window and the Bessel window are given in Tables 3.1 and 3.2, respectively. The results show that the FSLL of the sum-cosine window, with  $B = 0.100$ , is 6 dB lower than that of Kaiser's window, whereas the MSLL remains almost the same in both the cases. However, the main-lobe energy of the sum-cosine window is slightly smaller than that of Kaiser's zeroth-order Bessel window, but the difference is only about 0.00078%. Therefore, the energy ratios indicate that the side lobes of the modified zeroth-order Bessel window fall almost at the same rate as those of the sum-cosine windows. It is also seen that the sum-cosine window (with  $B = 0.103$ ) yields an improvement of nearly 1.5 dB in the FSLL (which is also equal to the MSLL in this case) over that of Kaiser's window. Nevertheless, the difference in the main-lobe energies is about the same as before.

To compare the performance of the raised-cosine family of windows with those of Kaiser's modified zeroth-order Bessel windows, the values of  $\alpha$  are determined such that the NHMLW are 1.10, 1.15, 1.20, and 1.25. The important parameters of the raised-cosine family and the Bessel windows are listed in Tables 3.1 and 3.2, respectively. These results show that the raised-cosine family of windows yield about 4–9 dB improvement in the FSLL and the improvement in the MSLL varies from +2.8 dB to –0.85 dB, when compared to Kaiser's windows. However, the window with  $D = 0.0113$  yields a loss of about 3 dB in both cases. The energy ratios indicate that the energy in the main lobe of the modified zeroth-order Bessel window is only slightly larger than that of the corresponding windows of the raised-cosine family (about 0.005%). This is similar to the case of the Hamming window, that is, the main-lobe energy of the Hamming window is less than that of the corresponding Kaiser–Bessel window.

Table 3.3 provides similar parameters for the modified first-order Bessel family of windows for different values of  $\gamma$ . The author has also developed a mathematical relationship to obtain the value of  $\gamma$  for a specified NHMLW. It is given by the relation  $\gamma = 2\pi\sqrt{2Y-1}$ , where  $Y$  is the required NHMLW. We consider two cases: (i) for an NHMLW of 1, the value of  $\gamma = 2\pi$ , (ii) for an NHMLW of 1.5, the value of  $\gamma$  is  $2\sqrt{2}\pi$  ( $\gamma = 8.885765$ ). As discussed earlier, for the same value of the NHMLW, this family of windows has the slight advantage of having lower FSLL than the corresponding zeroth-order

Bessel window family, but the RFSLL is slower. Besides, the computational complexity of the first-order Bessel family is much higher than the modified zeroth-order Bessel family of windows. To conclude, from the basic comparison presented in this chapter, we further point out from Table 3.1 that the Hann, Hamming, and the truncated Taylor windows belong to the same class of window functions. Among these, the Hamming window has the advantage of exhibiting the lowest side-lobe levels.

It is to be noted that the parameters of almost all the window functions described in Section 3.3 are given in Tables 3.1 through 3.3 in the increasing order of NHMLW. The window functions presented in Table 3.1 are called *fixed windows*, while those presented in Tables 3.2 and 3.3 are called *variable window function families*.

---

## References

1. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectra*, Dover Publications, New York, NY, 1958.
2. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proceedings*, vol. 66, no. 1, pp. 51–83, January 1978.
3. C.E. Cook and M. Bernfeld, *Radar Signals*, Academic Press, New York, NY, 1967.
4. J.F. Kaiser, Digital filters, in *System Analysis by Digital Computer*, Chapter 7, F.F. Kuo and J.F. Kaiser, Eds., Wiley, New York, NY, 1966.
5. K.M.M. Prabhu, *Data Windows in Digital Signal Processing*, PhD Thesis, IIT Madras, India, October 1980.
6. V.U. Reddy and K.M.M. Prabhu, Sum-cosine window, *Electronics Letters*, vol. 10, pp. 438–439, October 1974.
7. K.M.M. Prabhu, V.U. Reddy, and J.P. Agrawal, Synthesis of windows from a raised-cosine pulse, *Electronics Letters*, vol. 11, pp. 161–163, April 1975.
8. K.M.M. Prabhu and H. Renganathan, Optimised data windows, *Electronics Letters*, vol. 16, pp. 38–40, January 1980.
9. A. Papoulis, Minimum-bias windows for high-resolution spectral estimates, *IEEE Transactions on Information Theory*, vol. IT-19, pp. 9–12, January 1973.
10. N.C. Geckinli and D. Yavuz, Some novel windows and a concise tutorial comparison of window families, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP- 26, pp. 501–507, August 1978.
11. H.D. Helms, Digital filters with equiripple or minimax responses, *IEEE Transactions on Audio Electroacoustics*, vol. AU- 19, pp. 87–94, March 1971.
12. H.D. Helms, Non recursive digital filters: Design methods for achieving specifications on frequency response, *IEEE Transactions on Audio Electroacoustics*, vol. AU-16, pp. 336–342, September 1968.
13. C.L. Dolph, A current distribution for broad-side arrays which optimizes the relationship between beamwidth and side-lobe level, *Proceedings of IRE*, vol. 34, pp. 335–348, June 1946.
14. D. Slepian, H.T. Landau, and H.O. Pollak, Prolate spheroidal wave functions, fourier analysis, and uncertainty principle (I and II), *Bell System Technical Journal*, *BSTJ*, vol. 40, no. 1, pp. 43–80, January 1961.

15. J.F. Kaiser, *A family of window functions having nearly ideal properties*, Memorandum for File, Bell Telephone Laboratories, November 1964.
16. J.F. Kaiser and R.W. Schafer, On the use of the  $I_0$ -sinh window for spectrum analysis, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-28, no. 1, pp. 105–107, February 1980.
17. J.F. Kaiser, Nonrecursive digital filter design using the  $I_0$ -sinh window function, *Proceedings of IEEE International Symposium on Circuits and Systems*, San Francisco, CA, 1974.
18. A. Papoulis, *Signal Analysis*, McGraw-Hill Book Company, New York, NY, pp. 92–96, 1977.
19. A.H. Nuttall, Some windows with very good side-lobe behavior, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, pp. 84–91, February 1981.



# 4

---

## *Performance Comparison of Data Windows*

---

In this chapter, we compute several parameters of a window that are useful in choosing a suitable window for particular applications, such as power spectral estimation via discrete Fourier transform (DFT) and the design of FIR digital filters. A comprehensive comparison of the windows that were introduced in Chapter 3 is made based on the computed parameters. All the parameters listed are computed using the properties of windows in the continuous-time-domain.

The leakage that occurs in spectral estimation (via DFT) due to the prominent side lobes of the spectral window obviously degrades the accuracy of the results. Windows are weighting functions applied to the finite observation data to reduce the spectral leakage. There are four basic factors that need to be considered while choosing a window: (a) resolution or bandwidth, (b) stability, (c) leakage, and (d) smoothness. We shall now examine each of them in detail.

(a) *Resolution* refers to the ability of a spectrum estimate to represent fine structures in the frequency properties of the data, such as narrow peaks in the spectrum. Owing to the averaging involved in computing a spectrum estimate, a narrow peak in the periodogram is spread out into a broader peak. The width is roughly an image of the spectral window used in the estimate. Note that the width of the suitably defined spectral window is the bandwidth of the estimate. If the spectrum of a time series consists of two narrow peaks that are closer together than the bandwidth of the estimate used, we find that the two narrow peaks overlap, resulting in a single peak (which is broader). Thus, the estimate fails to resolve two narrow peaks that occur in close proximity to each other in the true spectrum.

(b) *Stability* of a spectrum estimate refers to the extent to which the estimates computed from different segments of a series concur, or the extent to which irrelevant fine structures in the periodogram are eliminated. Actually, resolution and stability are conflicting requirements since a high stability requires averaging over many periodograms, whereas this results in a reduced resolution.

(c) As discussed in one of the earlier chapters, *leakage* occurs because of the side lobes in the spectral window. This could be reduced by applying appropriate window functions. The smoothness of a spectrum is a less tangible property that would add a further conflict in requirements.



(d) *Smoothing* the input observations by a data window has three detrimental effects on the spectral estimates:

- i. Attenuation of the amplitudes
- ii. Loss of statistical stability
- iii. Loss in bandwidth of analysis

It is well known that applying windows on the data tapers the amplitude and thus introduces an attenuation of the spectral estimates [1–6]. A weighting factor, known as variance compensation factor ( $Q$ ), salvages against this loss and produces a window that has unit area in the frequency-domain. The effective attenuation of the spectrum values due to data smoothing is reduced by dividing the spectrum estimates (obtained from DFT) by the compensating factor. This can be computed easily (for a specific data window) and the procedure to obtain  $Q$  is described in the next section.

The loss of statistical accuracy due to windowing is automatically overcome by the cyclic nature of the algorithm [1], as this produces increasingly stable estimates. The reduction of bandwidth is unavoidable when applying data windows. A measure of this loss can be obtained *a priori* from the dispersion factor (to be defined later) and the frequency resolution is adjusted to provide a suitable bandwidth of analysis.

When a data window is applied, it causes an increase in the bandwidth of analysis. This, in turn, reduces the effective length of data over which it is applied. Therefore, the period of the data should be enhanced by some factor when the leakage is to be reduced and a specific 3 dB frequency has to be maintained. The half-power bandwidths of various windows are related to that of the rectangular window by means of the 3 dB ratios. These ratios also give a measure of the increase in the main-lobe width caused by data smoothing.

#### 4.1 Definition of Window Parameters

It is to be noted that all the parameters are defined and computed in the continuous-time and continuous-frequency-domains. The important window parameters (apart from the ones discussed in the previous chapter) in the context of performance, following the definitions of Refs. [1–6], are listed below:

- i. *Variance compensation factor* ( $Q$ ): The variance compensation factor of a window is computed by the following definition:

$$Q = \frac{1}{2\tau} \int_{-\tau}^{\tau} f^2(t) dt. \quad (4.1)$$

As is evident, this quantity is unity in the case of uniformly weighted data window and is less in the case of other windows.

The attenuation caused due to data smoothing can be compensated by normalizing the spectrum estimates obtained from the DFT by the value of the variance compensating factor,  $Q$ .

- ii. *Dispersion factor* ( $\eta$ ): This factor is defined as follows:

$$\eta = \frac{\int_{-\tau}^{\tau} f^2(t) dt}{\frac{1}{2\tau} \left[ \int_{-\tau}^{\tau} f(t) dt \right]^2}. \quad (4.2)$$

It is also known as the equivalent noise bandwidth of the window. A study of the dispersion factor indicates that windows that cause severe tapering of the data result in a spectral response, which has a wider main-lobe width when compared to the rectangular window (for which  $\eta = 1$ ). However, the side lobes are drastically reduced. Therefore, the application of a data window is a judicious compromise between the smearing effect caused by the window broadening and the leakage suppression that it provides.

- iii. *Coherent gain* ( $G$ ): The coherent gain factor is defined as

$$G = \frac{1}{2\tau} \int_{-\tau}^{\tau} f(t) dt. \quad (4.3)$$

Only in the case of a rectangular window, this factor is unity, while for all other windows the gain is reduced since the window smoothly tapers to zero near the boundaries. This reduction in proportionality is quite vital as it presents a known bias on spectral amplitudes.

- iv. *Total energy* ( $E$ ): The total energy in the time-domain is represented by the following definition:

$$E = \int_{-\tau}^{\tau} f^2(t) dt. \quad (4.4)$$

This parameter is also important since the variance of the smoothed spectral estimate depends on  $E$ .

- v. *Major-lobe energy* (MLE) *content*: The MLE content is defined as the ratio of the energy contained in the main lobe of the Fourier transform of the window to its total energy in the time-domain, computed with  $\tau = 1$ . It also provides an idea of the energy that is contained in the side lobes of the Fourier transform of the window.
- vi. *Half-power bandwidth*: This is also known as the 3 dB bandwidth and is yet another criterion that we should consider in the window selection process. It is the width of the window at half-power points. This criterion reflects the fact that two equal-strength main

lobes that are separated in frequency by an amount less than their corresponding 3 dB bandwidths will essentially exhibit a single spectral peak, and thus cannot be resolved as two distinct peaks. Therefore, this parameter indicates the resolution that can be obtained while applying a particular window.

- vii. *Peak side-lobe level (PSLL) or maximum side-lobe level (MSLL)*: It represents the ratio of the peak magnitude of the side lobe to the magnitude of the main lobe at zero frequency (DC). This parameter is expressed in dB.
- viii. *Normalized half-power bandwidth ( $\Delta BW$ )*: This can be defined as

$$\Delta BW = \frac{\text{Half-power bandwidth of the window under consideration}}{\text{Half-power bandwidth of the rectangular window}}.$$

- ix. *Normalized half main-lobe width ( $\Delta W$ )*: This ratio is defined as follows:

$$\Delta W = \frac{\text{Main-lobe width of the window under consideration}}{\text{Main-lobe width of the rectangular window}}.$$

- x. *6 dB bandwidth*.
- xi. *Rate of fall-off of side-lobe levels (RFSLL)*.
- xii. *Degradation loss ( $L$ )*: This is the reciprocal of the dispersion factor, which is expressed in dB.

Note that the half-power bandwidth ( $\Delta BW$ ), 6 dB bandwidth, MLE content, PSLL, normalized half main-lobe width ( $\Delta W$ ), and RFSLL are all parameters concerned with the Fourier transform of the data window. Therefore, these parameters are all computed numerically, except for the RFSLL. The procedure for computing the RFSLL was already detailed in Chapter 3. Other parameters such as variance compensation factor ( $Q$ ), dispersion factor ( $\eta$ ), coherent gain ( $G$ ), and total energy ( $E$ ) can be computed using the expressions given in Equations 4.1 through 4.4, respectively.

---

## 4.2 Computation of Window Parameters

We will now proceed to compute each of the parameters described in Equations 4.1 through 4.4 for the truncated Taylor family of windows [2,3] considered in the previous chapter.

The expression for the truncated Taylor family of windows [3] is reproduced below:

$$f(t) = \frac{(1+K)}{2} + \frac{(1-K)}{2} \cos\left(\frac{\pi t}{\tau}\right), \quad |t| \leq \tau. \quad (4.5)$$

From the definition of the variance compensation factor given in Equation 4.1, we obtain

$$(Q)_{TTF} = \frac{1}{2\tau} \int_{-\tau}^{\tau} \left[ \frac{(1+K)}{2} + \frac{(1-K)}{2} \cos\left(\frac{\pi t}{\tau}\right) \right]^2 dt. \quad (4.6)$$

Expanding Equation 4.6, we obtain

$$\begin{aligned} (Q)_{TTF} = \frac{1}{2\tau} \int_{-\tau}^{\tau} & \left[ \frac{(1+K)^2}{4} + \frac{(1-K)^2}{4} \cos^2\left(\frac{\pi t}{\tau}\right) \right. \\ & \left. + \frac{(1+K)(1-K)}{2} \cos\left(\frac{\pi t}{\tau}\right) \right] dt. \end{aligned} \quad (4.7)$$

Using trigonometric identities, we can simplify Equation 4.7 as follows:

$$\begin{aligned} (Q)_{TTF} = \frac{1}{2\tau} \int_{-\tau}^{\tau} & \left\{ \frac{(1+K)^2}{4} + \frac{(1-K)^2}{2} \cos\left(\frac{\pi t}{\tau}\right) \right. \\ & \left. + \frac{(1-K)^2}{8} \left[ 1 + \cos\left(\frac{2\pi t}{\tau}\right) \right] \right\} dt. \end{aligned}$$

Upon integration and simplification, we obtain the following result:

$$\begin{aligned} (Q)_{TTF} &= \frac{1}{\tau} \left[ \tau \frac{(1+K)^2}{4} + \tau \frac{(1-K)^2}{8} \right] \\ &= \frac{1}{\tau} \left[ \tau \frac{(3+2K+3K^2)}{8} \right] \\ &= \frac{(3+2K+3K^2)}{8}. \end{aligned} \quad (4.8)$$

Equation 4.8 provides the expression for the variance compensation factor ( $Q$ ). If we set  $K = 0$  in Equation 4.5, it represents the Hann window and its corresponding  $Q \simeq 0.375$ . The Hamming window results when  $K = 0.08$  and in this case,  $Q \simeq 0.4$ . The variance compensation factor for all the other windows considered in the previous chapter are presented in Tables 4.1 through 4.3. Now, consider the expression for the dispersion factor,  $\eta$ , given

**TABLE 4.1**  
Performance Comparison of Windows

Window	Dispersion Factor, $\eta$	Variance Compensation Factor, $Q$	Total Energy, $E$	Half-Power Bandwidth ( $\tau = 1$ )	AW $\Delta BW$	Coherent Gain, $G$	Degradation Loss, $L$ (dB)	Rate of Fall-off Side Lobe	6 dB Bandwidth	
<i>Truncated Taylor Family</i>										
$\left(\frac{1+K}{2}\right) + \left(\frac{1-K}{2}\right) \times \cos \frac{\pi t}{\tau}$	$\frac{(3+2K+3K^2)}{2(1+K)^2}$	$\frac{(3+2K+3K^2)}{8}$	$\frac{(3+2K+3K^2)\tau}{4}$	—	—	—	—	—	—	
Hamming ( $K = 0.08$ )	1.3628	0.3974	$0.7948\tau$	0.652	2.0	1.475	0.54	-1.34	-6	0.908
<i>Raised-cosine Family</i>										
$\left(\frac{1-2D}{2}\right) \left(1 + \cos \frac{\pi t}{\tau}\right) + 2D \cos \frac{2\pi t}{\tau}$	$\frac{(0.75-3D+7D^2)}{2(0.5-D)^2}$	$\frac{(0.75-3D+7D^2)}{2}$	$(0.75-3D+7D^2)\tau$	—	—	—	—	—	—	
$D = 0.0113$	1.5011	0.3585	$0.717\tau$	0.716	2.208	1.62	0.48	-1.76	-6	1.0
$D = 0.0138$	1.5016	0.3550	$0.71\tau$	0.716	2.30	1.62	0.49	-1.77	-6	1.0
$D = 0.0155$	1.5021	0.3526	$0.7052\tau$	0.716	2.396	1.62	0.49	-1.77	-6	1.0
$D = 0.0165$	1.5023	0.3512	$0.7024\tau$	0.716	2.49	1.62	0.48	-1.77	-6	1.0
Hanning ( $D = 0$ )	1.50	0.375	$0.75\tau$	0.720	2.0	1.629	0.50	-1.76	-18	1.0
<i>Sum-cosine</i>										
$(1-2B) \cos \frac{\pi t}{2\tau} + 2B \cos \frac{3\pi t}{2\tau}$	$\frac{18\pi^2(1-4B+8B^2)}{(12-32B)^2}$	$\frac{(1-4B+8B^2)}{2}$	$(1-4B+8B^2)\tau$	—	—	—	—	—	—	

$B = 0.100$	1.56	0.34	0.68 $\tau$	0.744	2.50	1.683	0.47	-1.93	-12	1.040
$B = 0.103$	1.5779	0.3364	0.6729 $\tau$	0.752	2.50	1.701	0.46	-1.98	-12	1.052
<b>Generalized Blackman</b>										
$(0.5 - 2C) + 0.5 \cos \frac{\pi t}{\tau}$	$\frac{(0.375 - 2C + 6C^2)}{(0.5 - 2C)^2}$	$(0.375 - 2C + 6C^2)$	$(0.375 - 2C + 6C^2)2\tau$	—	—	—	—	—	—	—
$+ 2C \cos \frac{2\pi t}{\tau}$										
Blackman	1.7268	0.3046	0.6092 $\tau$	0.822	3.0	1.860	0.42	-2.37	-18	1.150
Rectangular	1.0	1.0	2 $\tau$	0.442	1.0	1.0	1.0	0.0	-6	0.604
Parabolic	1.20	0.533	1.066 $\tau$	0.58	1.416	1.312	0.667	-0.792	-12	0.795
Cosine	1.234	0.50	$\tau$	0.594	1.50	1.34	0.64	-0.91	-12	0.820
Triangular	1.333	0.333	0.667 $\tau$	0.638	2.0	1.44	0.50	-1.25	-12	0.886
$\text{Cos}^3 x$	1.735	0.313	0.626 $\tau$	0.830	2.50	1.81	0.42	-2.39	-24	1.156
$\text{Cos}^4 x$	1.944	0.274	0.547 $\tau$	0.938	3.0	2.122	0.375	-2.889	-30	1.294
Optimized Blackman	1.759	0.299	0.597 $\tau$	0.838	3.0	1.896	0.412	-2.453	-18	1.17
Papoulis	1.786	0.293	0.535 $\tau$	0.852	3.0	1.92	0.41	-2.52	-24	1.19
Blackman-Harris	2.0	0.258	0.516 $\tau$	0.95	4.0	2.149	0.359	-3.02	-6	1.334
Parzen	1.92	0.27	0.539 $\tau$	0.91	4.0	2.06	0.38	-2.83	-24	1.276

**TABLE 4.2**  
Performance Comparison of Modified Zeroth-Order Bessel Family

Zeroth-Order		MSLL (dB)	Dispersion Factor, $\eta$	Variance Compensation Factor, $Q$	Total Energy, $E$	Half-Power Bandwidth ( $\tau = 1$ )	$\Delta W$	$\Delta BW$	Coherent		Degradation	
with $\alpha$	NHMLW								Gain, $G$	Loss, $L$ (dB)	6 dB Bandwidth	
5.0	0.940	-36.73	1.359	0.403	0.807 $\tau$	0.652	1.880	1.475	0.545	-1.33	0.906	
$\pi\sqrt{3}$	1.000	-39.79	1.407	0.386	0.772 $\tau$	0.674	2.0	1.525	0.524	-1.48	0.938	
6.0	1.078	-43.82	1.467	0.367	0.733 $\tau$	0.702	2.156	1.588	0.50	-1.66	0.978	
$\pi\sqrt{3.84}$	1.100	-44.93	1.483	0.362	0.724 $\tau$	0.708	2.20	1.602	0.494	-1.71	0.988	
6.5	1.149	-47.44	1.519	0.352	0.704 $\tau$	0.724	2.298	1.638	0.481	-1.81	1.012	
$\pi\sqrt{4.76}$	1.200	-50.07	1.554	0.342	0.685 $\tau$	0.742	2.40	1.679	0.469	-1.92	1.036	
7.0	1.221	-51.15	1.569	0.339	0.677 $\tau$	0.748	2.442	1.692	0.465	-1.96	1.046	
$\pi\sqrt{5.25}$	1.250	-52.62	1.589	0.334	0.668 $\tau$	0.758	2.50	1.715	0.458	-2.01	1.058	
7.5	1.294	-54.93	1.618	0.327	0.654 $\tau$	0.770	2.588	1.742	0.450	-2.09	1.078	
8.0	1.368	-58.67	1.667	0.316	0.633 $\tau$	0.792	2.736	1.792	0.436	-2.22	1.110	
8.5	1.442	-62.55	1.712	0.307	0.613 $\tau$	0.814	2.884	1.842	0.423	-2.34	1.140	

**TABLE 4.3**  
Performance Comparison of Modified First-Order Bessel Family

Zeroth-Order Bessel Window with $\alpha$	NHMLW	MSLL (dB)	Dispersion Factor, $\eta$	Variance		Total Energy, $E$	Half-Power Bandwidth ( $\tau = 1$ )	$\Delta W$	$\Delta BW$	Coherent		Degradation	
				Factor, $Q$	Compensation					Gain, $G$	Loss, $L$ (dB)	6 dB Bandwidth	
6.129688	0.976	-41.79	1.364	0.396	0.793r	0.654	1.952	1.480	0.54	-1.35	0.91		
6.565842	1.046	-45.14	1.411	0.380	0.760r	0.674	2.092	1.525	0.52	-1.50	0.94		
7.104398	1.139	-49.67	1.468	0.363	0.725r	0.70	2.278	1.584	0.50	-1.67	0.978		
7.576740	1.227	-54.05	1.517	0.349	0.698r	0.722	2.454	1.634	0.48	-1.81	1.01		
8.045189	1.320	-57.88	1.564	0.337	0.674r	0.744	2.64	1.683	0.464	-1.94	1.042		
8.513068	1.418	-61.93	1.611	0.326	0.652r	0.766	2.836	1.733	0.45	-2.07	1.072		
8.983568	1.511	-66.08	1.656	0.316	0.632r	0.788	3.022	1.783	0.437	-2.19	1.102		
9.458593	1.567	-70.12	1.701	0.307	0.614r	0.808	3.134	1.828	0.425	-2.31	1.132		



by Equation 4.2, which can be expressed as

$$\eta = \frac{2\tau Q}{\frac{1}{2\tau} \left[ \int_{-\tau}^{\tau} f(t) dt \right]^2} = \frac{2\tau Q}{\text{DEN}}. \quad (4.9)$$

The denominator (DEN) of Equation 4.9 is computed for the truncated Taylor family (using Equation 4.5) as follows:

$$\begin{aligned} \text{DEN} &= \frac{1}{2\tau} \left[ \int_{-\tau}^{\tau} \left( \frac{1+K}{2} + \frac{1-K}{2} \cos\left(\frac{\pi t}{\tau}\right) \right) dt \right]^2 \\ &= \frac{1}{2\tau} [(1+K)\tau]^2 \\ &= \frac{(1+K)^2\tau}{2}. \end{aligned} \quad (4.10)$$

The substitution of Equations 4.8 and 4.10 into Equation 4.9 yields the following expression for the dispersion factor for the truncated Taylor family:

$$(\eta)_{TF} = \frac{(3 + 2K + 3K^2)}{2(1 + K)^2}.$$

Noting that the energy,  $E = 2\tau Q$ , we have

$$(E)_{TF} = \frac{(3 + 2K + 3K^2)\tau}{4}.$$

We recall that the coherent gain  $G$  by definition is

$$G = \frac{1}{2\tau} \int_{-\tau}^{\tau} f(t) dt.$$

For the truncated Taylor family,  $G$  is given by

$$\begin{aligned} (G)_{TF} &= \frac{1}{2\tau} \int_{-\tau}^{\tau} \left[ \frac{(1+K)}{2} + \frac{(1-K)}{2} \cos\left(\frac{\pi t}{\tau}\right) \right] dt \\ &= \frac{1}{2\tau} [(1+K)\tau] \\ &= \frac{(1+K)}{2}. \end{aligned}$$

The rest of the parameters are computed numerically, assuming the value of  $\tau$  to be unity.

Following the above steps, all the parameters [4–6] are computed for the remaining data windows considered in Section 3.3 of the previous chapter. It must be emphasized that in the case of Kaiser's modified zeroth-order Bessel family [7] as well as Kaiser's modified first-order Bessel family [7], all the parameters have to be computed numerically. Tables 4.1 through 4.3 provide the computed values of the parameters. For the purpose of comparison, the corresponding values of the rectangular window are also included in Table 4.1.

Figure 4.1(a) shows the plots of NHMLW versus MSLL in dB, for various data windows, including that of Kaiser's modified zeroth-order and first-order Bessel families of windows. It is interesting to note that these plots are straight lines in the case of the Kaiser–Bessel windows. An approximate linear relationship between the side-lobe levels and the main-lobe widths is obtained for Kaiser's modified zeroth-order Bessel family and is given by

$$y \simeq -52x + 12.4,$$

where  $x$  represents the NHMLW and  $y$  gives the MSLL in dB. The main-lobe width for this class of windows can be changed by varying the value of  $\alpha$  (the parameter of the window family), which usually lies within the range  $4 \leq \alpha \leq 9$  for optimum results [7]. A similar relationship can be obtained for Kaiser's modified first-order Bessel family of windows, which can be given as follows:

$$y \simeq -47.68x + 4.68,$$

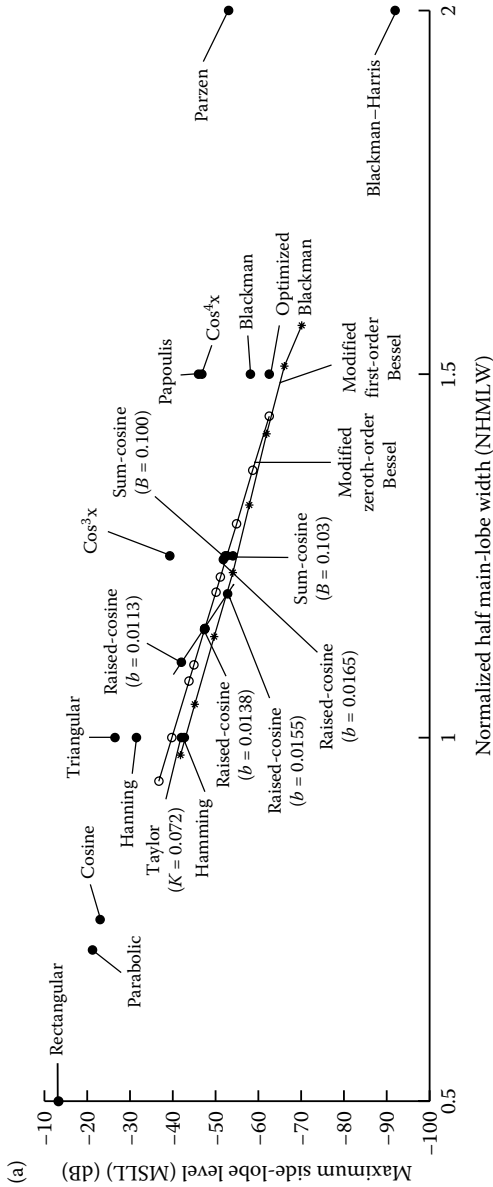
where  $x$  again represents the NHMLW of Kaiser's modified first-order Bessel family and  $y$  gives the corresponding MSLL in dB. The NHMLW and the MSLL for these two families of windows are also included in Tables 4.2 and 4.3 for the sake of comparison.

Figure 4.1(b) shows the plot of the NHMLW versus the side-lobe energy (SLE), for different data windows. Again, in the case of the modified zeroth-order Bessel family, this follows a regular shape. For a given main-lobe width, the modified zeroth-order Bessel window contains maximum energy in the main lobe of its Fourier transform and, consequently, the minimum in the side lobes of its Fourier transform. Therefore, Kaiser's modified zeroth-order Bessel windows are called near-optimum windows since they closely approximate the optimum prolate-spheroidal wave functions whose band-limiting properties are well known [8].

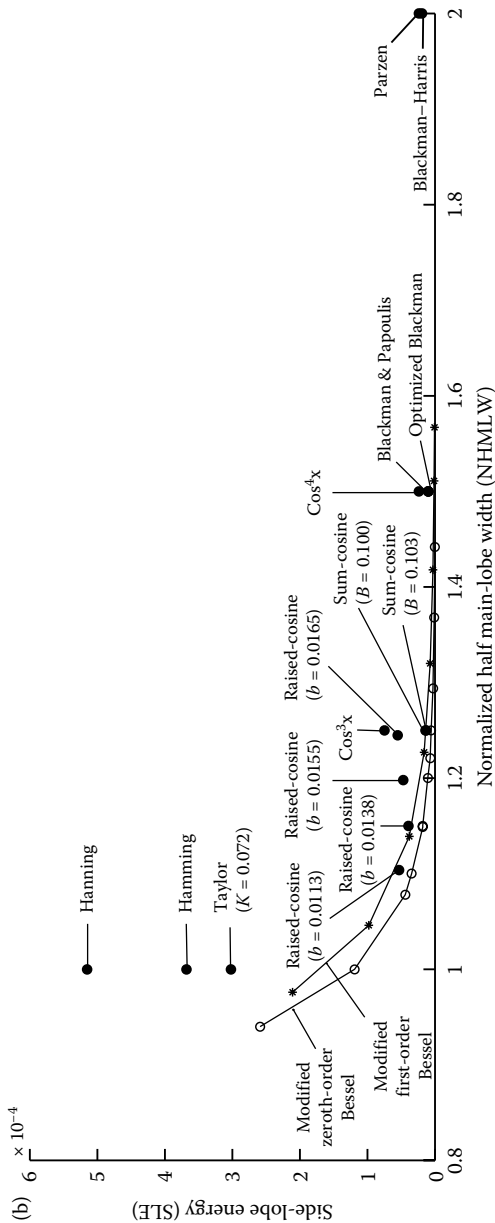
---

### 4.3 Discussion on Window Selection

In spectral estimation, a desirable window is the one that yields small values of the variance compensation factor ( $Q$ ), dispersion factor ( $\eta$ ), total energy ( $E$ ),



**FIGURE 4.1** Performance comparison of typical data windows. (a) NHMLW versus MSL (dB). (b) NHMLW versus SLE.



**FIGURE 4.1** Performance comparison of typical data windows. (a) NHMLW versus MSL (dB). (b) NHMLW versus SLE. (Continued).

PSLL,  $\Delta W$  and  $\Delta BW$ , and a large value of the main-lobe energy (MLE) content. However, the results of Tables 4.1 through 4.3 show that the decrease in the PSLL is associated with an increase in both  $\Delta W$  and  $\eta$ , which is a measure of the increase in the main-lobe width. Similarly, a smaller value of the PSLL leads to a higher leakage suppression. A rise in  $\eta$  implies a corresponding increase in the loss of frequency resolution. The variance of the smoothed spectral estimate is proportional to the total energy,  $E$ , when the data are simple functions of a Gaussian process. It is to be noted that the variance of any estimate is a measure of its reliability: the smaller the value of  $E$ , the higher is its reliability. Therefore, it is necessary to make a compromise between the variance and the bias of an estimator of a spectrum. The bias can only be made small by making the main lobe as narrow (and as close to a delta function) as possible. On the other hand, a narrow spectral window results in a large variance. Thus, the selection of an optimal data window for spectral estimation turns out to be a judicious compromise among the various parameters presented in these Tables.

For the purpose of designing FIR digital filters, a window with the smallest main-lobe width and the lowest PSLL is best suited for weighting the Fourier coefficients,  $h(n)$ . However, these conditions cannot be met simultaneously. Therefore, the selection of a window for this application is again a trade-off between these two parameters.

The graphs presented in Figures 4.1(a) and (b) are useful in selecting a data window that is closest to the near-optimum Kaiser–Bessel window, namely, the modified zeroth-order Bessel family. A window that lies very close to the straight line (corresponding to the zeroth-order Kaiser–Bessel window) of Figure 4.1(a) and the curve (again corresponding to the zeroth-order Kaiser–Bessel window) of Figure 4.1(b) is desirable. The graphs indicate that the *sum-cosine window* satisfies these conditions simultaneously. From the above discussions, it is clear that there is no unique window that has universal applications for optimum results.

---

## References

1. T.S. Durrani and J.M. Nightingale, Data windows for digital spectral analysis, *IEE Proceedings*, vol. 119, pp. 343–352, March 1972.
2. K.M.M. Prabhu, *Data Windows in Digital Signal Processing*, PhD Thesis, IIT Madras, India, October 1980.
3. C.E. Cook and M. Bernfeld, *Radar Signals*, Academic Press, New York, NY, 1967.
4. K.M.M. Prabhu, V.U. Reddy, and J.P. Agrawal, Performance-comparison of data windows, *Electronics Letters*, vol. 13, pp. 600–601, 1977.
5. K.M.M. Prabhu and V.U. Reddy, Data windows in digital signal processing: A review, *Journal of the Institution of Electronics and Telecommunication Engineers*, vol. 26, pp. 69–76, 1980.

6. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proceedings*, vol. 66, No. 1, pp. 51–83, January 1978.
7. J.F. Kaiser, *A Family of Window Functions Having Nearly Ideal Properties*, Memorandum for File, Bell Laboratories, November 1964.
8. D. Slepian, H.T. Landau, and H.O. Pollak, Prolate spheroidal wave functions, Fourier analysis, and uncertainty principle (Parts I and II), *Bell Systems Technical Journal*, vol. 40, No. 1, pp. 43–80, January 1961.



# 5

---

## *Discrete-Time Windows and Their Figures of Merit*

---

Having introduced a variety of popular windows in the continuous-time-domain in Chapter 3, we will now discuss their discrete-time counterparts, so that they can be implemented in several digital signal processing applications. To have a wider choice, we have also included few more windows in this chapter, the characteristics of which were not discussed in Chapter 3. We have also defined a number of parameters that will enable the user to select appropriate windows, depending on the application. Finally, based on the parameters computed, we compare the performance of the discrete-time window functions. Section 5.1 presents different classes of window functions. Discrete-time windows are discussed in Section 5.2. Different figures of merit, which will enable proper window selection, are defined in Section 5.3. The concept of time–bandwidth product is introduced in Section 5.4. Finally, Section 5.5 gives applications of windows in finite impulse response filter design and power spectral estimation.

---

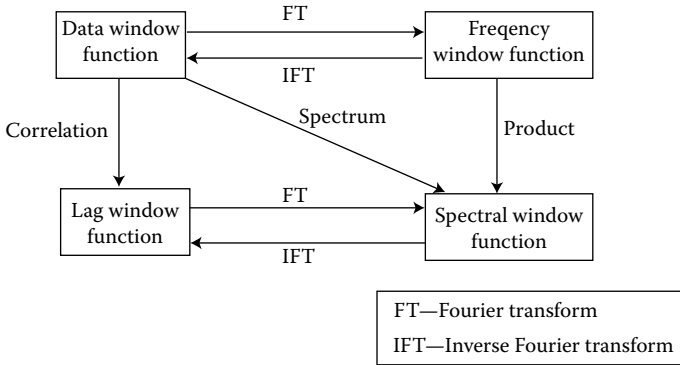
### 5.1 Different Classes of Windows

As we have seen in the earlier chapters, window functions are weighting functions associated with the spectral analysis of a time series. They act as either intrinsic parameters or externally applied kernels of the spectral estimation techniques, thereby improving the statistical characteristics of the spectrum.

We can classify windows into four different categories that are incorporated in time series analysis as well as in power spectrum estimation techniques. The four different classes are detailed below:

- i. *Data windows*: As the name indicates, this is a time function that is directly applied to the data or a time series. It may be a set of weighting coefficients or a smoothing time function, depending on the format of the data.
- ii. *Frequency window*: This is a weighting function applied in the frequency-domain on the Fourier transform of the data. This amounts to the convolution (or multiplication) of the data windows and the





**FIGURE 5.1**  
Interplay between different forms of window functions.

Fourier transform of the data. Taking inverse Fourier transform of a frequency window yields its time-domain counterpart.

- iii. *Lag window*: This is a sequence with which the covariance function of an observed process is weighted. The rectangular lag window is given by

$$f(n) = \begin{cases} 1, & -M \leq n \leq M \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

where  $(2M + 1)$  is the window length. All the lag windows here are defined for the interval  $[-M, M]$ , and are symmetrical about the origin.

- iv. *Spectral window*: This is obtained by taking the Fourier transform of the lag window. This can be considered as a filter through which the true spectrum of a process is estimated.

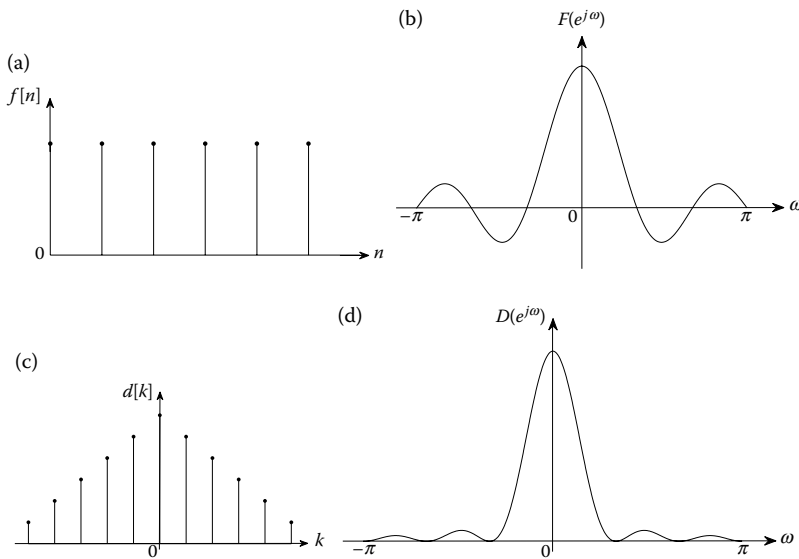
Figure 5.1 represents the inter-relationships between the various forms of windows described above. Lag windows are generally applied to the covariance estimate of an input process. This in turn reduces the variance of the resulting spectral estimate. Data windowing (or data smoothing) is employed to minimize the frequency leakage effects, which is similar to the Gibbs phenomenon occurring in approximation theory.

**Example**

We illustrate the four different categories of windows by means of some examples.

*Data window*: As an example, a rectangular data window is used.

$$f[n] = \begin{cases} 1, & n = 0, 1, \dots, (N - 1) \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$



**FIGURE 5.2** Interplay between various window functions. (a) Data window. (b) Frequency window. (c) Lag window. (d) Spectral window.

For the data window represented in Equation 5.2 and shown in Figure 5.2(a), the corresponding, lag, frequency and spectral windows are obtained as follows:

*Lag window:* Using the relationships given in Figure 5.1, we can generate the corresponding lag window as the autocorrelation of the data window. Thus, for a rectangular window, it generates a corresponding triangular lag window as shown in Figure 5.2(c). The amplitude of this triangular window is normalized to unity. This gives

$$d[k] = \begin{cases} 1 - \frac{|k|}{N}, & k = 0, \pm 1, \dots, \pm(N - 1) \\ 0, & \text{otherwise.} \end{cases} \quad (5.3)$$

*Frequency window:* This is obtained as the Fourier transform of the rectangular data window, which is shown in Figure 5.2(b), and is given by the following expression:

$$|F(e^{j\omega})| = \left| \frac{\sin(N\omega/2)}{\sin(\omega/2)} \right|. \quad (5.4)$$

*Spectral window:* It is the Fourier transform of the above lag window (see Equation 5.3). This is shown in Figure 5.2(d) and is equal to

$$|D(e^{j\omega})| = \frac{1}{N} \left| \frac{\sin(N\omega/2)}{\sin(\omega/2)} \right|^2. \quad (5.5)$$

## 5.2 Discrete-Time Windows

In this section, we will catalog some popular windows as well as some not-so-well-known windows, in the discrete-time-domain. Most of the windows are the sampled versions of the continuous-time windows described in Section 3.3. We will provide the discrete form of each window and also its important characteristics. All the windows are presented as even functions (symmetric about the origin), while their lengths  $(N + 1)$  are odd.

### 5.2.1 Rectangular (Box Car) Window

This is the simplest window since the truncation of the input data amounts to applying the rectangular window. It is otherwise called the Dirichlet window. It is defined as

$$f[n] = 1, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.6)$$

### 5.2.2 Triangular (Bartlett) Window

An  $(N + 1)$ -length triangular window (including zeros at the edges) is obtained by linearly convolving two rectangular windows of lengths  $\frac{N}{2}$  each. Therefore, the transform of this window is obviously the square of the Dirichlet kernel. It is the simplest among those windows that exhibit a nonnegative Fourier transform due to its self-convolution property. Its functional form is given as follows:

$$f[n] = 1 - \frac{|2n|}{N}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.7)$$

### 5.2.3 $\cos^\alpha x$ Window Family

In this family of windows, changing the value of the parameter  $\alpha$  generates different windows.

$$f[n] = \cos^\alpha \left( \frac{\pi n}{N} \right), \quad 0 \leq |n| \leq \frac{N}{2}, \quad (5.8)$$

where  $\alpha$  takes on integer values.

### 5.2.4 Hann Window

The Hann window is actually a special case of the  $\cos^\alpha x$  window, with  $\alpha = 2$ , that is,

$$f[n] = 0.5 + 0.5 \cos \left( \frac{2\pi n}{N} \right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.9)$$

### 5.2.5 Truncated Taylor Family of Windows

This is a generalized window that is expressed as

$$f[n] = \left(\frac{1+k}{2}\right) + \left(\frac{1-k}{2}\right) \cos\left(\frac{2\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.10)$$

Here  $k$  can take the values in the range  $0 \leq k \leq 1$ . We note that the Hann window is a special case of the truncated Taylor family with  $k = 0$ .

### 5.2.6 Hamming Window

This window is again a special case of the truncated Taylor family with  $k = 0.08$ .

$$f[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.11)$$

In the case of the Hamming window, the value of  $k$  is selected such that the peak side-lobe level (PSLL) is minimized.

### 5.2.7 Sum-Cosine Window

This is an optimized form of the  $\cos^3 x$  window.

$$f[n] = (1 - 2B) \cos\left(\frac{\pi n}{N}\right) + 2B \cos\left(\frac{3\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.12)$$

It is to be noted that we discuss the figures of merit of various windows, including the sum-cosine window, in Section 5.3. Further, in Table 5.3, we summarize the results with  $B = 0.1$  and  $0.103$ , for the sum-cosine window. The reason for obtaining two optimum values of  $B$  for the sum-cosine window was already given in Section 3.3.8. The values of the PSLL are, respectively, 52 and 54 dB, for the  $B = 0.1$  and  $0.103$  [1].

### 5.2.8 Raised-Cosine Window Family

In this class of windows, changing the value of the parameter  $D$  results in windows with slightly different main-lobe widths that lie in between the Hann (or Hamming) and sum-cosine windows.

$$f[n] = \frac{1-2D}{2} \left(1 + \cos\left(\frac{2\pi n}{N}\right)\right) + 2D \cos\left(\frac{4\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.13)$$

If  $D = 0$ , then  $f[n]$  gives the Hann window. As was done for the previous window, in Section 5.3 and Table 5.3, we summarize the results for

$D = 0.0113, 0.0138, 0.0155,$  and  $0.0165$  [1]. The rationale behind four different optimum values of  $D$  for the raised-cosine family of windows was provided in Section 3.3.10 (for its continuous-time counterpart). The PSLL varies between  $-42$  and  $-51$  dB, respectively.

### 5.2.9 Blackman Window

This belongs to the more general class of the Blackman–Harris window family (see Section 5.2.12) consisting of only two cosine terms and a constant.

$$f[n] = 0.42 + 0.5 \cos\left(\frac{2\pi n}{N}\right) + 0.08 \cos\left(\frac{4\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.14)$$

### 5.2.10 Optimized Blackman Window

The coefficients of the Blackman window given in Equation 5.14 were found to be not optimal. The optimized form of the Blackman window is given by

$$f[n] = 0.412 + 0.5 \cos\left(\frac{2\pi n}{N}\right) + 0.088 \cos\left(\frac{4\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.15)$$

While optimizing the PSLL, the second coefficient is kept to be the same (0.5) as the original Blackman window [1]. This is because the coefficient 0.5 has a special significance while implementing this window in the frequency-domain. We shall provide more details on this topic in chapter 6 (see Section 6.4).

### 5.2.11 Tukey Window

This window can be generated by convolving a cosine lobe of length  $(\frac{\alpha N}{2})$  with a rectangular window of length  $(1 - \frac{\alpha}{2})N$ . The Tukey window is also referred to as a *tapered cosine window*. Its time-domain expression is given by

$$f[n] = \begin{cases} 1, & 0 \leq |n| \leq (1 - \alpha)\frac{N}{2} \\ 0.5 \left[ 1 + \cos\left(\frac{\pi \left(n - (1 - \alpha)\frac{N}{2}\right)}{\alpha N/2}\right) \right], & (1 - \alpha)\frac{N}{2} \leq |n| \leq \frac{N}{2}. \end{cases} \quad (5.16)$$

If  $\alpha = 0$ , then the Tukey window reduces to a rectangular window and when  $\alpha = 1$ , it takes the form of the Hann window.

**5.2.12 Blackman–Harris Window**

There are several variations of the Blackman–Harris window depending on the coefficients chosen. Four examples are listed in Table 5.1 given below [2,3]. The general expression is given as

$$f[n] = a_0 + a_1 \cos\left(\frac{2\pi n}{N}\right) + a_2 \cos\left(\frac{4\pi n}{N}\right) + a_3 \cos\left(\frac{6\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \tag{5.17}$$

It has been found that the minimum three-term window can give a side-lobe level of  $-70.83$  dB, whereas the minimum four-term window can achieve a side-lobe level as low as  $-92.01$  dB for large values of window length.

**5.2.13 Nuttall Window Family**

The Nuttall window family has better side-lobe structure when compared to the Blackman–Harris windows for large values of window lengths. The window definition is similar to the sum-cosine expression given in Equation 5.18 with coefficients given in Table 5.2 [3]. The minimum three-term Nuttall window can achieve a PSL as low as  $-71.48$  dB and the four-term can reach upto  $-98.17$  dB, maintaining the same rate of fall-off of side-lobe level (RFSLL) of the Blackman–Harris window ( $-6$  dB/octave), for large values of window length.

**TABLE 5.1**  
Blackman–Harris Window Family

	Minimum Three-Term ( $-70.83$ dB)	Three-Term ( $-62.05$ dB)	Minimum Four-Term ( $-92.01$ dB)	Four-Term ( $-74$ dB)
$a_0$	0.42323	0.44959	0.35875	0.40217
$a_1$	0.49755	0.49364	0.48829	0.49703
$a_2$	0.07922	0.05677	0.14128	0.09892
$a_3$	–	–	0.01168	0.00188

**TABLE 5.2**  
Nuttall Window Family

Nuttall Window	$a_0$	$a_1$	$a_2$	$a_3$
Three-term with continuous third derivative	0.375	0.5	0.125	–
Three-term with continuous first derivative	0.40897	0.5	0.09103	–
Four-term with continuous fifth derivative	0.3125	0.46875	0.1875	0.03125
Four-term with continuous third derivative	0.338946	0.481973	0.161054	0.018027
Four-term with continuous first derivative	0.355768	0.487396	0.144232	0.012604
Minimum three-term	0.4243801	0.4973406	0.0782793	–
Minimum four-term	0.3635819	0.4891775	0.1365995	0.0106411

5.2.14 Flat-Top Window

Flat-top window got its name from the *maximally flat structure* of the main lobe. Here, we discuss only the third-order and fifth-order flat-top windows. They can be defined as follows:

$$f[n] = a_0 + a_1 \cos\left(\frac{2\pi n}{N}\right) + a_2 \cos\left(\frac{4\pi n}{N}\right) + a_3 \cos\left(\frac{6\pi n}{N}\right) + a_4 \cos\left(\frac{8\pi n}{N}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.18)$$

where  $a_0 = 0.2811, a_1 = 0.5209,$  and  $a_2 = 0.198$  for the third-order flat-top window, and  $a_0 = 0.21557895, a_1 = 0.41663158, a_2 = 0.277263158, a_3 = 0.083578947,$  and  $a_4 = 0.006947368$  for the fifth-order flat-top window [4].

The time-domain and frequency-domain plots for the fifth-order flat-top window are given in Figures 5.3(a) and (b), respectively. From Figures 5.3(a), we can see that the flat-top window exhibits *negative* weights. This is an exception from the windows discussed earlier. The maximally flat main-lobe structure can be obtained by equating derivatives of  $F(e^{j\omega})$  at  $\omega = 0$  to zero [4].

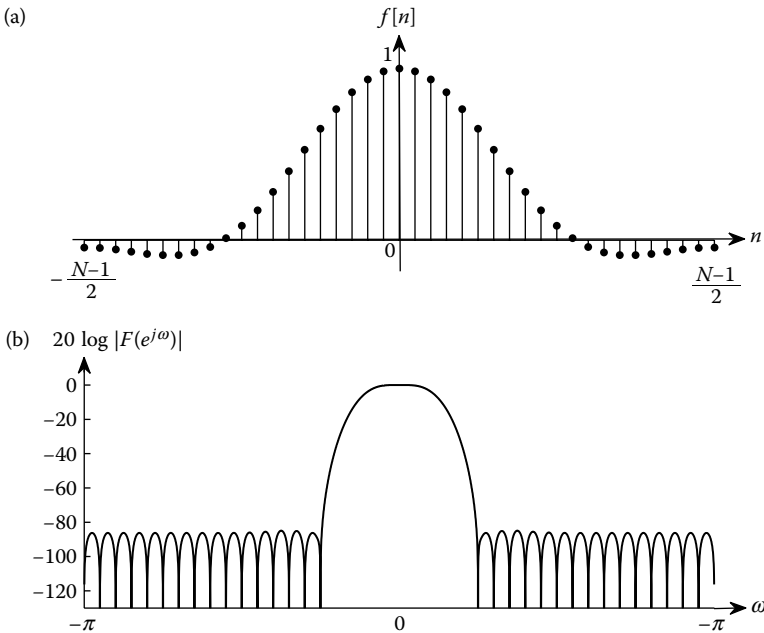


FIGURE 5.3

(a) Fifth-order flat-top window weights for  $N = 41$ . (b) Magnitude response of  $f[n]$ .

Due to the flat and wide main-lobe structure, the scalloping loss (discussed in Section 5.3) is very small for this window.

### 5.2.15 Parabolic Window

This window is described as follows:

$$f[n] = 1 - \left(\frac{2n}{N}\right)^2, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.19)$$

It exhibits a discontinuous first derivative at the edges and therefore its side lobes fall off at  $-12$  dB/octave.

### 5.2.16 Riemann Window

This is defined as the central lobe of the sinc kernel.

$$f[n] = \frac{\sin\left(\frac{2\pi n}{N}\right)}{\frac{2\pi n}{N}}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.20)$$

It is similar to the  $\cos \alpha_x$  window discussed in Section 5.2.3, when  $\alpha = 1$  value is substituted.

### 5.2.17 Poisson Window

This is actually a family of windows with  $\alpha$  being the variable parameter. It has the general form

$$f[n] = e^{-\alpha \frac{|n|}{N/2}}, \quad 0 \leq |n| \leq \frac{N}{2} \quad (5.21)$$

The usual  $\alpha$  values are 2, 3, and 4. It has a discontinuity at the boundaries and therefore the transform falls off at a rate of  $-6$  dB/octave only.

### 5.2.18 Gaussian Window

We know that the frequency response of a Gaussian time function is also a Gaussian. Since the time span of a Gaussian function is infinity, it must be truncated at the ends to use it as a window function. This window has the following form:

$$f[n] = e^{-\frac{1}{2} \left[ \alpha \frac{n}{N/2} \right]^2}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.22)$$

It should be noted that as we increase the value of  $\alpha$ , the width of the window decreases, and this will in turn reduce the severity of the discontinuity at the



edges. However, this will increase the main-lobe width, and consequently reduce the side-lobe levels, in the transform domain. The commonly used values of  $\alpha$  are 2.5, 3, and 3.5.

### 5.2.19 Cauchy Window

This window is also a family with the variable parameter  $\alpha$ . It is given by

$$f[n] = \frac{1}{1 + \left[\alpha \frac{n}{N/2}\right]^2}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.23)$$

The usual values of  $\alpha$  used in practice are 3, 4, and 5. The Fourier transform of the Cauchy window turns out to be a two-sided exponential function.

### 5.2.20 Hann–Poisson Window

This window can be constructed as the product of the Hann and the Poisson windows (which were introduced in Sections 5.2.4 and 5.2.17).

$$f[n] = 0.5 \left[ 1 + \cos \left( \pi \frac{n}{N/2} \right) \right] e^{(-\alpha \frac{|n|}{N/2})}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.24)$$

The usual values of  $\alpha$  are 0.5, 1, and 2. The RFSLL in the Hann–Poisson window is  $-18$  dB/octave, which is the same as the Hann window. This window is actually similar to the Poisson window family. This set of windows exhibit a very large main-lobe width.

### 5.2.21 Papoulis (Bohman) Window

This window is a result of the convolution of the half-duration cosine-lobe with itself.

$$f[n] = \left( 1 - \frac{|n|}{N/2} \right) \cos \left( \frac{\pi n}{N/2} \right) + \frac{1}{\pi} \sin \left( \frac{\pi |n|}{N/2} \right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.25)$$

Naturally, the Fourier transform of this window is the square of the cosine lobe's Fourier transform. The Papoulis window can alternatively be expressed as the product of the triangular window with a single cycle of a cosine window of the same period. The second term appearing in the above window function acts as a *correction term*. This is added so that the first derivative is set to zero at the boundaries. The third derivative of the Papoulis window is discontinuous and therefore the RFSLL is  $\frac{1}{\omega^4}$  or  $-24$  dB/octave.

### 5.2.22 Jackson (Parzen) Window

The Jackson window is obtained by convolving two triangular windows of one-half extent and is given by

$$f[n] = \begin{cases} 1 - 6 \left[ \frac{n}{N/2} \right]^2 \left[ 1 - \frac{|n|}{N/2} \right], & 0 \leq |n| \leq \frac{N}{4} \\ 2 \left[ 1 - \frac{|n|}{N/2} \right]^3, & \frac{N}{4} \leq |n| \leq \frac{N}{2}. \end{cases} \quad (5.26)$$

The Fourier transform of this window is the square of the Fourier transform of the triangular window. Therefore, the RFSLL of the side lobes in the case of the Parzen window will be  $\frac{1}{\omega^4}$ . The transform of this window is non-negative, just like the triangular window, because of its self-convolution construction.

### 5.2.23 Dolph–Chebyshev Window

In signal processing applications, we always look for a window that exhibits a narrow bandwidth for a known finite duration. A similar problem is faced by the *antenna design* community. In the antenna context, the problem is to illuminate an antenna of finite aperture to obtain a narrow main-lobe beam, while restricting the side-lobe levels. The closed-form solution to the minimum main-lobe width for a given side-lobe level is offered by the Dolph–Chebyshev window. The DFT of this window is given by

$$F[k] = \frac{\cos \left[ N \cos^{-1} \left[ \beta \cos \left( \frac{\pi k}{N} \right) \right] \right]}{\cosh \left[ N \cosh^{-1}(\beta) \right]}, \quad 0 \leq k \leq N - 1 \quad (5.27)$$

$$\text{where } \beta = \cosh[\cosh^{-1}(10^\alpha)/N] \quad (5.28)$$

$$\text{and } \cos^{-1}(X) = \begin{cases} \frac{\pi}{2} - \tan^{-1}[X/\sqrt{1-X^2}], & |X| \leq 1 \\ \ln[X + \sqrt{X^2-1}], & |X| \geq 1 \end{cases}.$$

To obtain the corresponding window samples  $f[n]$  in the time-domain, we simply perform an IDFT on the Dolph–Chebyshev window samples  $F[k]$  and scale the amplitude, such that the peak value is unity, like in the case of other window time samples.

### 5.2.24 Modified Zeroth-Order Kaiser–Bessel Window Family

The modified zeroth-order Kaiser–Bessel window is defined as

$$f[n] = \frac{I_0 \left[ \alpha \sqrt{1 - \left( \frac{2n}{N} \right)^2} \right]}{I_0[\alpha]}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.29)$$

where  $I_0(x)$  is the modified Bessel function of zeroth-order and first kind, defined as

$$I_0(x) = \sum_{m=0}^{\infty} \frac{1}{(m!)^2} \left(\frac{x}{2}\right)^{2m}. \quad (5.30)$$

Kaiser had discovered a simple approximation to the prolate-spheroidal wave function of order zero, which is difficult to compute. Since the prolate-spheroidal wave function maximizes the energy in a band of frequencies, say  $W$ , such a function is said to be optimum. As the zeroth-order Kaiser–Bessel family approximates this property of the zeroth-order prolate-spheroidal wave function, it is also called the *near-optimum* window family.

### 5.2.25 Modified First-Order Kaiser–Bessel Window Family

$$f[n] = \frac{I_1 \left[ \gamma \sqrt{1 - \left(\frac{2n}{N}\right)^2} \right]}{I_1[\gamma] \sqrt{1 - \left(\frac{2n}{N}\right)^2}}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (5.31)$$

In the above expression,  $I_1(x)$  represents the modified first-order Bessel function of the first kind. It is defined as

$$I_1(x) = \sum_{m=0}^{\infty} \frac{1}{m!(m+1)!} \left(\frac{x}{2}\right)^{2m+1}. \quad (5.32)$$

As in the case of the modified zeroth-order Kaiser–Bessel family, the side-lobe level can be varied by choosing different values of  $\gamma$ . The modified first-order Bessel family has the slight advantage of smaller first side-lobe compared to either the zeroth-order Bessel windows or the prolate-spheroidal wave functions, but on the other hand, the RFSLL is a bit slower.

### 5.2.26 Saramäki Window Family

This is a new class of window functions that are found to be a closer approximation to the prolate-spheroidal wave functions than the Kaiser–Bessel window family. These windows are derived using frequency transformation that maps the frequency response of a rectangular window to another frequency response having wider main-lobe width [5].

The Saramäki window function is defined as

$$\hat{f}[n] = \hat{f}_0[n] + 2 \sum_{k=1}^{N/2} \hat{f}_k[n], \quad (5.33)$$

where  $\hat{f}_0 = \delta[n]$  and  $\hat{f}'_k$ s can be calculated using the recursive relation as follows:

$$\hat{f}_1[n] = \begin{cases} \gamma - 1, & n = 0 \\ \gamma/2, & |n| = 1 \\ 0, & \text{otherwise} \end{cases} \quad (5.34)$$

$$\hat{f}_k[n] = \begin{cases} 2(\gamma - 1)\hat{f}_{k-1}[n] - \hat{f}_{k-2}[n] + \gamma[\hat{f}_{k-1}[n-1] + \hat{f}_{k-1}[n+1]], & -k \leq n \leq k \\ 0, & \text{otherwise.} \end{cases} \quad (5.35)$$

If we compute  $\hat{f}[n]$ , we find a window function for which  $\hat{f}[0]$  is not equal to 1. We therefore scale this function using the relation

$$f[n] = \begin{cases} \hat{f}[n]/\hat{f}[0], & 0 \leq |n| \leq \frac{N}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (5.36)$$

### 5.2.27 Ultraspherical Window

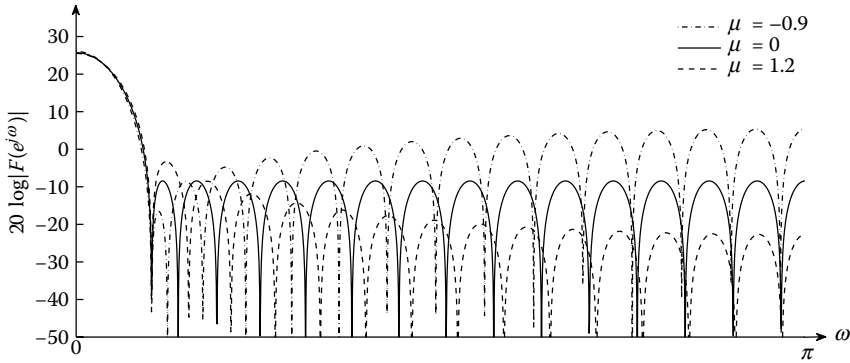
The ultraspherical window is one of the newly introduced windows, which has wide applications in signal processing. We note that the Kaiser–Bessel window, the Dolph–Chebyshev window, and the Saramäki window discussed earlier have a variable parameter in addition to the window length. This is to control the PSL of the window. Apart from these two parameters, the ultraspherical window has a third parameter that controls the RFSLL. The ultraspherical window coefficients in the range of  $-M$  to  $M$ , for a window of length  $N = 2M + 1$  can be calculated as follows [6]:

$$f[n] = \frac{\mu x_\mu^{2M}}{M + |n|} \binom{\mu + M + |n| - 1}{M + |n| - 1} \sum_{m=0}^{M-|n|} \binom{\mu + M - |n| - 1}{M - |n| - m} \times \binom{M + |n|}{m} (1 - x_\mu^{-2})^m. \quad (5.37)$$

The binomial coefficients can be calculated as

$$\binom{\alpha}{0} = 1, \quad \binom{\alpha}{p} = \frac{\alpha(\alpha - 1)\dots(\alpha - p + 1)}{p!}, \quad p \geq 1.$$

The parameter  $x_\mu$  given in the window definition (see Equation 5.37) can be used to adjust the width of the main lobe and  $\mu$  controls the RFSLL. Figure 5.4 shows the ultraspherical window magnitude response for  $\mu = 1.2, 0$ , and  $-0.9$  with fixed  $x_\mu$ . From this figure, we can see that



**FIGURE 5.4** Ultraspherical window magnitude response for  $x_\mu = 1.1$ ,  $N = 31$ , and  $\mu$  taking the values 0, 1.2, and  $-0.9$ .

- i. For positive values of  $\mu$ , the side lobes decrease as we go from 0 to  $\pi$ .
- ii. For negative values of  $\mu$ , the side lobes increases.
- iii. For  $\mu = 0$ , we get the Dolph–Chebyshev window, where all the side lobes are equal.
- iv. For  $\mu = 1$ , the Saramäki window is also a special case of ultraspherical window [6]. The coefficients calculated from Equation 5.37 are not normalized. The normalized ultraspherical window can be obtained by dividing  $f[n]$  by  $f[0]$  for odd  $N$ , or by  $f[0.5]$  for even  $N$ .

### 5.2.28 Odd and Even-Length Windows

The windows defined in this section can be obtained from the continuous-time windows (introduced in Chapter 3) by taking samples at

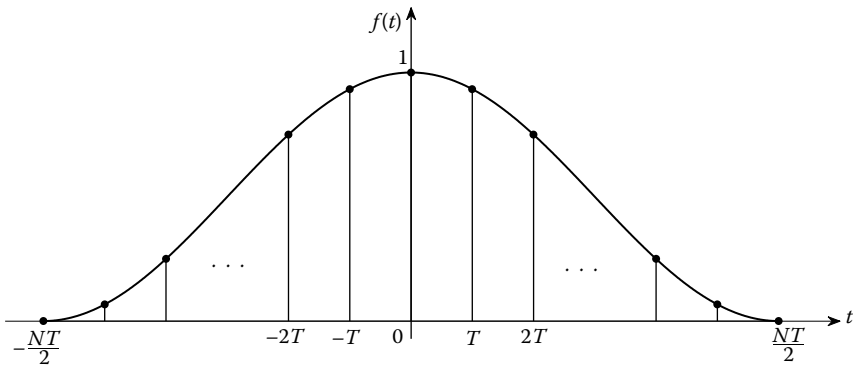
$$t = nT = \frac{2n\tau}{N}, \quad |n| \leq \frac{N}{2},$$

where  $\tau$  is the one-sided duration of the window and  $N$  is an even number (but length is odd). This is depicted in Figure 5.5 for the Hann window. These are *odd length* windows and therefore the point of symmetry is always an integer. However, in practice, we may come across many applications that uses radix-2 FFT and thus require even-length windows.

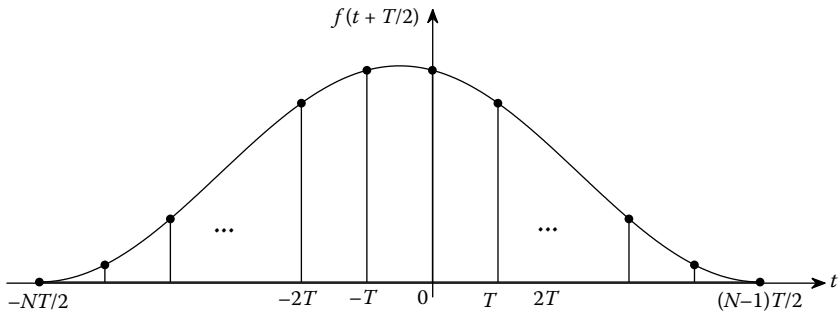
*Even-length* windows can be obtained by shifting the continuous-time windows by  $\frac{T}{2}$  and sampling the shifted window at

$$t = nT = \frac{2n\tau}{N}, \quad -\frac{N}{2} \leq n \leq \left(\frac{N}{2} - 1\right) \tag{5.38}$$

as shown in Figure 5.6. Equivalently, we can replace  $n$  by  $(n + \frac{1}{2})$  in all the discrete-time window definitions for odd lengths and compute the coefficients, with  $n$  taking the values  $-\frac{N}{2} \leq n \leq (\frac{N}{2} - 1)$ .



**FIGURE 5.5**  
Odd-length window.



**FIGURE 5.6**  
Even-length window.

**Note:** Windows such as rectangular, Hann, Blackman, and so on have zeros at the edges. These zeros do not contribute to the output. They are included only to maintain consistency in the discrete-time and continuous-time window definitions. In the hardware implementation, we generally take only nonzero coefficients to reduce unnecessary computations.

---

### 5.3 Figures of Merit

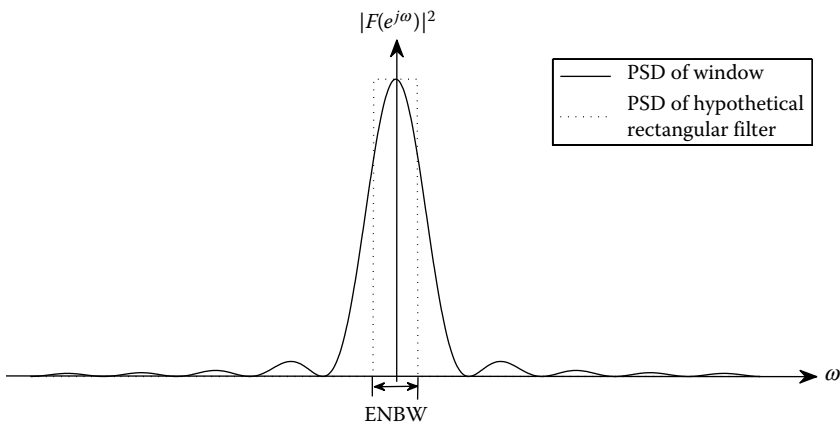
The most important application of window functions is in spectral analysis. They are sometimes used in the design of linear-phase FIR filters. In both these applications, the reduction of spectral leakage is one of the vital factors in choosing a window from among several window functions. In this section, various parameters of windows are defined that pertain to the time-domain

as well as to the frequency-domain [1,2]. All these parameters will help us in selecting an appropriate window function for a particular application. The computed parameters for all the window functions are presented in Tables 5.3, 5.4 and 5.5 at the end of this section.

1. *Equivalent noise bandwidth (B)*: Owing to spectral leakage, impulse functions are replaced by broader bandwidth sampling functions. As the side lobes of these functions contribute to noise, the window must be designed so as to reduce these side lobes. The Equivalent noise bandwidth (ENBW) measures the noise performance of the window function. The magnitude, say at a frequency  $\omega_0$ , is computed by taking the weighted sum of the contributions from all the frequencies. Therefore, the value at  $\omega_0$  is biased by its own frequency, broadband noise, and other interfering harmonics.

Consider the case of a signal having a single tone at  $\omega_0$  and the broadband white noise spread over the entire spectrum. The accumulated noise at this frequency can be minimized by using windows having lower bandwidths. The parameter ENBW is defined as the bandwidth of a hypothetical (ideal) rectangular filter that has the same root mean square (RMS) value of noise signal as that of the window of interest (shown in Figure 5.7) [7]. Here, the peak powers of the rectangular filter and that of the window under consideration must be the same. Let  $N_o$  be the noise power spectral density (PSD). Then, the noise power  $P_n$ , accumulated by the window, can be defined as

$$P_n = \frac{N_o}{2\pi} \int_{-\pi}^{\pi} |F(e^{j\omega})|^2 d\omega. \quad (5.39)$$



**FIGURE 5.7**  
Equivalent noise bandwidth.

From Parseval's theorem, this can also be computed in the time-domain using the expression

$$P_n = N_o \sum_{n=0}^{N-1} |f[n]|^2. \quad (5.40)$$

The noise power collected by the rectangular filter is

$$\begin{aligned} P_n &= N_o F^2(0) B \\ &= N_o \left[ \sum_{n=0}^{N-1} |f[n]| \right]^2 B \end{aligned} \quad (5.41)$$

Therefore, the ENBW measure is given by  $B = \frac{\sum_{n=0}^{N-1} |f[n]|^2}{\left[ \sum_{n=0}^{N-1} f[n] \right]^2}$ .

$$(5.42)$$

The ENBW is easy to compute and this definition can be used to compare the side-lobe behavior of different windows. The smaller the value of ENBW, the better the performance of the window, in the presence of broadband noise. We note that the ENBW of the rectangular window is  $\frac{1}{N}$  bins. Using the *Schwartz inequality*, it is possible to prove that no other window can have a smaller ENBW than  $\frac{1}{N}$  bins. The parameter ENBW (normalized to  $\frac{1}{N}$ ) for all windows considered in this chapter is presented in Tables 5.3 through 5.5. Owing to the normalization factor, the value of ENBW for a rectangular window is unity.

2. *Coherent gain (G)*: When a window is applied on a signal, it tends to taper the signal to zero near the boundaries or edges of the observation interval. This happens because of the effect of applying the data window, which smoothly approaches zero at the edges. The signal power gets reduced due to this tapering. The magnitudes at the frequency bins will no longer be equal to its true values. Coherent gain (G) gives a known scaling factor to get the absolute values at these frequencies. This is also known as the DC gain of the window. It is defined as

$$G = \sum_{n=0}^{N-1} f[n].$$

For a rectangular window, the DC gain is  $N$ , the number of terms in the window. However, for any other window, the DC gain will be reduced, since it tapers down to zero at the edges. This reduction in the DC gain is important since it accounts for a definite scaling of the



**TABLE 5.3**  
Performance Comparison of Windows

Window	Half-Power				6 dB		PSLL	L (dB)	Scalloping loss	WCPL	Overlap Correlation (0.5)	Overlap Correlation (0.75)	RFSLL (dB/Octave)	
	AW	ENBW	Q	BW	ΔBW	G								Bandwidth
Rectangular	1.000	1.000	1.000	0.443	1.000	1.000	0.603	-13.252	0.000	3.914	3.914	50.000	75.000	-6
Tukey ( $\beta = 0.25$ )	1.143	1.102	0.844	0.503	1.136	0.875	0.686	-13.601	0.422	2.958	3.380	44.444	74.074	-18
Tukey ( $\beta = 0.50$ )	1.334	1.222	0.688	0.574	1.297	0.750	0.786	-15.121	0.872	2.235	3.106	36.364	72.727	-18
Riesz	1.431	1.201	0.533	0.578	1.304	0.666	0.795	-21.262	0.795	2.218	3.013	34.335	76.435	-12
$\cos x$	1.500	1.234	0.500	0.594	1.342	0.636	0.819	-22.966	0.914	2.092	3.007	31.798	75.517	-12
Tukey ( $\beta = 0.75$ )	1.600	1.360	0.531	0.652	1.471	0.625	0.897	-19.394	1.335	1.727	3.062	25.056	70.476	-18
Riemann	1.639	1.300	0.451	0.625	1.411	0.589	0.865	-26.371	1.138	1.887	3.024	27.334	73.383	-12
Poisson ( $\alpha = 2$ )	1.818	1.314	0.246	0.606	1.368	0.433	0.849	-19.168	1.186	2.028	3.215	27.525	69.813	-6
Triangular	2.000	1.334	0.334	0.638	1.439	0.500	0.885	-26.448	1.252	1.819	3.071	24.952	71.845	-12
Hamming	2.000	1.363	0.397	0.651	1.469	0.540	0.906	-42.811	1.344	1.749	3.093	23.377	70.689	-6
Truncated Taylor	2.000	1.375	0.395	0.656	1.482	0.536	0.914	-41.797	1.382	1.718	3.100	22.743	70.261	-6
$\cos^2 x$ (Hann)	2.000	1.500	0.375	0.720	1.624	0.500	0.999	-31.467	1.761	1.421	3.182	16.667	65.916	-18
Dolph-Chebyshev ( $\beta = 2.5$ )	2.148	1.392	0.379	0.668	1.507	0.528	0.942	-50.000	1.418	1.720	3.138	22.282	69.579	0
Raised-cosine ( $D = 0.0113$ )	2.207	1.501	0.358	0.716	1.616	0.489	0.999	-41.929	1.764	1.441	3.205	16.690	65.487	-6
Raised-cosine ( $D = 0.0138$ )	2.298	1.502	0.355	0.715	1.614	0.486	0.999	-47.131	1.766	1.446	3.212	16.702	65.362	-6
Raised-cosine ( $D = 0.0155$ )	2.393	1.502	0.353	0.715	1.614	0.485	0.999	-52.204	1.767	1.449	3.216	16.712	65.270	-6
Raised-cosine ( $D = 0.0165$ )	2.485	1.502	0.351	0.714	1.613	0.484	0.999	-51.262	1.768	1.451	3.219	16.718	65.213	-6
Sum-cosine ( $\beta = 0.1$ )	2.500	1.560	0.340	0.743	1.677	0.467	1.038	-51.997	1.932	1.340	3.271	14.345	63.119	-18
Sum-cosine ( $\beta = 0.103$ )	2.500	1.578	0.336	0.751	1.696	0.462	1.050	-54.164	1.981	1.310	3.291	13.670	62.430	-18
$\cos^3 x$	2.500	1.735	0.313	0.828	1.870	0.424	1.154	-39.295	2.393	1.073	3.466	8.488	56.681	-24
Dolph-Chebyshev ( $\beta = 3$ )	2.510	1.513	0.344	0.734	1.657	0.473	1.024	-60.000	1.772	1.437	3.209	16.299	64.857	0

Hann-Poisson ( $\alpha = 0.5$ )	2.586	1.609	0.302	0.766	1.730	0.433	1.070	-35.184	2.066	1.257	3.323	12.551	61.304	-18
Dolph-Chebyshev ( $\beta = 3.5$ )	2.872	1.616	0.317	0.776	1.751	0.438	1.102	-70.000	2.190	1.191	3.379	11.814	59.295	0
Nuttall (minimum three-term)	2.980	1.704	0.307	0.810	1.841	0.424	1.130	-68.709	2.314	1.090	3.404	9.694	54.823	-6
Flat-top (third order)	2.980	2.965	0.234	1.470	3.341	0.281	1.780	-43.754	4.72	0.003	4.718	7.907	21.510	-6
Blackman-Harris (three-term -62.05 dB)	3.000	1.611	0.326	0.765	1.727	0.450	1.071	-60.077	2.070	1.265	3.335	12.578	61.019	-6
Blackman-Harris (three-term -70.83 dB)	3.000	1.709	0.306	0.811	1.830	0.423	1.136	-68.917	2.326	1.127	3.453	9.555	57.226	-6
Blackman	3.000	1.727	0.305	0.821	1.853	0.420	1.148	-58.111	2.372	1.097	3.469	8.963	56.667	-18
Bohman	3.000	1.786	0.293	0.850	1.919	0.405	1.188	-45.997	2.518	1.021	3.540	7.384	54.529	-24
$\cos^4 x$	3.000	1.944	0.273	0.925	2.088	0.375	1.293	-46.741	2.888	0.862	3.750	4.286	48.555	-30
Gaussian ( $\alpha = 2.5$ )	3.202	1.394	0.354	0.686	1.549	0.495	0.960	-42.157	1.601	1.677	3.278	19.934	67.714	-6
Dolph-Chebyshev ( $\beta = 4$ )	3.234	1.726	0.296	0.838	1.892	0.409	1.174	-80.000	2.386	1.105	3.482	8.143	56.667	0
Blackman-Harris (four-term -74.39 dB)	3.246	1.791	0.290	0.851	1.921	0.402	1.190	-73.486	2.531	1.020	3.551	7.358	54.216	-6
Gaussian ( $\alpha = 3$ )	3.482	1.702	0.295	0.801	1.809	0.417	1.128	-55.897	2.309	1.161	3.470	10.618	57.489	-6
Cauchy ( $\alpha = 3$ )	3.684	1.489	0.258	0.671	1.515	0.416	0.954	-31.019	1.730	1.667	3.397	20.021	61.579	-6
Poisson ( $\alpha = 3$ )	3.830	1.660	0.167	0.726	1.640	0.317	1.042	-24.753	2.202	1.433	3.635	14.916	54.680	-6
Cauchy ( $\alpha = 4$ )	3.936	1.776	0.195	0.759	1.713	0.331	1.106	-35.049	2.495	1.361	3.826	13.024	48.758	-6
Nuttall (minimum four-term)	3.970	1.976	0.261	0.930	2.114	0.364	1.310	-92.089	2.958	0.817	3.775	4.176	44.144	-6
De La Valle-Poussin	4.000	1.917	0.270	0.909	2.051	0.375	1.274	-53.044	2.827	0.896	3.723	4.967	49.296	-24
Blackman-Harris (four-term -92 dB)	4.000	2.004	0.258	0.948	2.141	0.359	1.331	-92.003	3.020	0.824	3.844	3.760	45.999	-6
Flat-top (fifth order)	4.960	3.771	0.175	1.860	4.227	0.2166	2.291	-86.523	5.762	0.008	5.772	1.526	1.526	-6
Gaussian ( $\alpha = 3.5$ )	5.226	1.908	0.253	0.928	2.095	0.358	1.311	-69.711	2.959	0.869	3.828	4.614	46.494	-6
Cauchy ( $\alpha = 5$ )	5.822	2.075	0.157	0.844	1.906	0.275	1.271	-30.402	3.171	1.121	4.276	8.915	38.314	-6
Poisson ( $\alpha = 4$ )	7.814	2.082	0.126	0.874	1.974	0.246	1.292	-31.363	3.184	1.019	4.203	7.279	40.184	-6

**TABLE 5.4**  
Performance Comparison of Modified Zeroth-Order Kaiser–Bessel Family

$\alpha$	Half-Power				6 dB Bandwidth				Scalloping Loss	WCPL	Overlap Correlation (0.5)	Overlap Correlation (0.75)
	$\Delta W$	ENBW	Q	BW	$\Delta BW$	G	Bandwidth	PSLL				
5	1.882	1.359	0.403	0.652	1.468	0.545	0.906	-36.624	1.333	3.066	23.596	71.015
$\pi\sqrt{3}$	2.002	1.407	0.386	0.674	1.518	0.524	0.938	-39.709	1.484	3.105	21.053	69.149
6	2.156	1.467	0.367	0.702	1.581	0.5	0.978	-43.711	1.664	3.161	18.2	66.821
$\pi\sqrt{3.84}$	2.2	1.483	0.362	0.71	1.599	0.494	0.988	-44.849	1.712	3.178	17.47	66.179
6.5	2.3	1.519	0.352	0.726	1.635	0.481	1.012	-47.383	1.815	3.215	15.961	64.783
$\pi\sqrt{4.76}$	2.402	1.555	0.342	0.742	1.671	0.469	1.036	-50.031	1.916	3.255	14.54	63.37
7	2.444	1.569	0.339	0.748	1.685	0.465	1.046	-51.132	1.957	3.272	13.991	62.796
$\pi\sqrt{5.25}$	2.502	1.589	0.334	0.758	1.707	0.458	1.058	-52.637	2.01	3.295	13.277	62.022
7.5	2.59	1.618	0.327	0.772	1.739	0.449	1.078	-54.942	2.09	3.33	12.258	60.861
8	2.738	1.666	0.316	0.794	1.788	0.436	1.11	-58.788	2.216	3.388	10.737	58.981
8.5	2.888	1.712	0.307	0.814	1.833	0.423	1.14	-62.641	2.336	3.447	9.402	57.155

**TABLE 5.5**  
Performance Comparison of Modified First-Order Kaiser–Bessel Family

$\gamma$	$\Delta W$	ENBW	Q	Half-Power BW	$\Delta BW$	G	6 dB Bandwidth	PSLL	L (dB)	Scalloping Loss	WCPL	Overlap Correlation (0.5)	Overlap Correlation (0.75)
6.1297	1.938	1.369	0.396	0.656	1.477	0.538	0.912	-40.384	1.363	1.713	3.075	23.089	70.626
6.5658	2.074	1.414	0.38	0.678	1.527	0.518	0.942	-43.523	1.505	1.609	3.114	20.72	68.835
7.104	2.262	1.47	0.363	0.702	1.581	0.497	0.98	-47.76	1.674	1.494	3.168	18.078	66.63
7.5767	2.448	1.518	0.349	0.724	1.631	0.479	1.012	-51.885	1.814	1.405	3.219	16.007	64.718
8.0452	2.704	1.565	0.337	0.746	1.68	0.464	1.042	-55.492	1.946	1.325	3.271	14.172	62.857
8.5131	2.864	1.611	0.326	0.768	1.73	0.45	1.072	-59.248	2.072	1.253	3.326	12.537	61.041
8.9836	3.036	1.657	0.316	0.788	1.775	0.437	1.102	-63.146	2.192	1.189	3.381	11.075	59.259
9.4586	3.22	1.701	0.307	0.808	1.82	0.425	1.132	-66.862	2.307	1.129	3.437	9.767	57.509

amplitudes of the signal spectrum. *Coherent power gain* is the square of the sum of the window terms, given by

$$G^2 = \left( \sum_{n=0}^{N-1} f[n] \right)^2.$$

The values of  $G$  for different windows are also presented in Tables 5.3 through 5.5. Note that the coherent gain is normalized by dividing with  $N$ . Therefore, the coherent gain ( $G$ ) of a rectangular window is unity and for all other windows, it is less than one.

3. *Degradation loss ( $L$ )*: This is also known as the processing gain, and it indicates the degradation of the signal-to-noise ratio (SNR) due to windowing. It depends on the shape of the window (since that in turn determines the ENBW). This loss  $L$  is the ratio of the output SNR of the signal after windowing  $[(\text{SNR})_o]$  to the input SNR of the original sequence  $[(\text{SNR})_i]$ :

$$L = \frac{(\text{SNR})_o}{(\text{SNR})_i}.$$

The use of window functions effectively results in the reduction of signal power and the accumulation of noise from the neighboring frequencies. The degradation loss ( $L$ ) can alternatively be defined as

$$L = \frac{\text{Coherent power gain}}{B}.$$

Coherent power gain is a measure of reduction in signal power, whereas  $B$  (which denotes the ENBW) measures the noise from other frequency components. If there is only one coherent component, then the coherent power gain will be unity. In such a case

$$L = \frac{1}{B} = \frac{\left( \sum_{n=0}^{N-1} f[n] \right)^2}{\sum_{n=0}^{N-1} f^2[n]}. \quad (5.43)$$

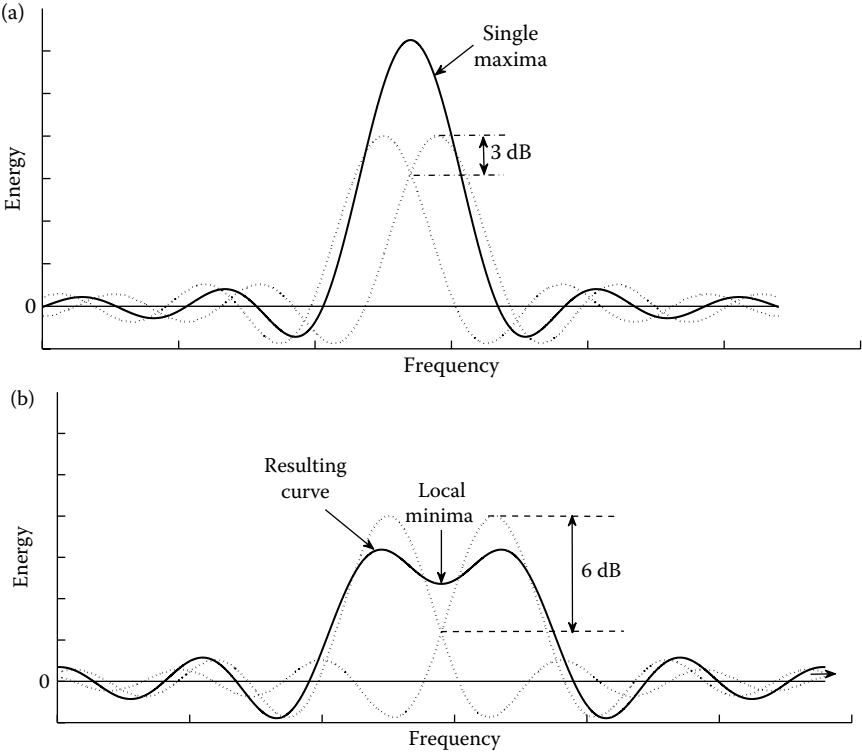
The degradation loss is computed for all the windows, by assuming only one coherent component, and this is presented in Tables 5.3 through 5.5.

4. *3 dB and 6 dB bandwidths*: These are important metrics that determine the minimum separation required between two frequency components of equal amplitudes, such that they can be resolved. In other words, resolution means that there should be a local minimum between the two peaks. The rectangular window has excellent resolution characteristics for signals of comparable strengths. In the case

of DFT, the neighboring components are weighted by the window and added coherently including the side lobes. As a rule of thumb for resolvability, the gain at the crossover of the kernel must be less than 0.5 [7]. This implies that 6 dB bandwidth is the deciding factor of spectral resolution rather than the 3 dB bandwidth. The comparison between 6-dB and 3-dB bandwidths is given in Figure 5.8. However, the difference between the 3 dB bandwidth and noise power bandwidth is a good indicator to study the performance of the windows. The 3 dB bandwidth, 6 dB bandwidth, and  $\Delta BW$  (normalized with respect to the rectangular window's 3 dB bandwidth) are all included in Tables 5.3 through 5.5. The parameter  $\Delta BW$  is defined as

$$\Delta BW = \frac{\text{3 dB bandwidth of window}}{\text{3 dB bandwidth of rectangular window}}$$

An example of the 6-dB and 3-dB bandwidths of the Hamming window is provided in Figure 5.8.

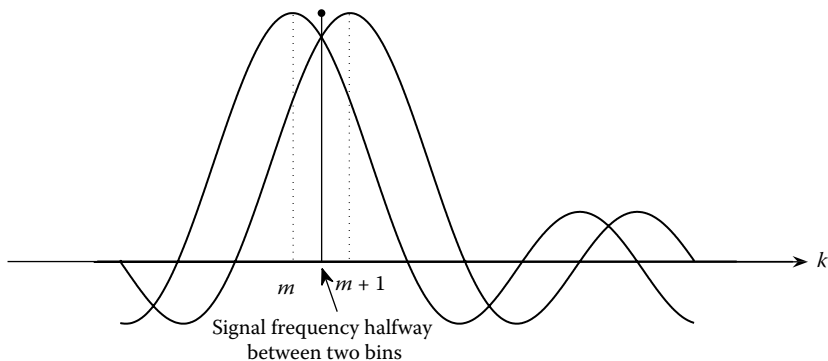


**FIGURE 5.8** Minimum resolution bandwidth. (a) Nonresolvable peaks. (b) Resolvable peaks.

5. *Main-lobe width ( $\Delta W$ )*: To represent fine structures in the spectrum, it is essential that the kernel should have a very narrow main-lobe width. This parameter ( $\Delta W$ ) determines the extent to which the visibility of a weak component is affected by the presence of a nearby strong coherent component. We note here that the rectangular window exhibits the smallest main-lobe width and therefore gives excellent frequency resolution capability. Hence, this parameter is also normalized with respect to the rectangular window's main-lobe width. The parameter  $\Delta W$  for all the windows are presented in Tables 5.3 through 5.5. Owing to the scaling factor,  $\Delta W$  for a rectangular window is 1.0.
6. *Scalloping loss or picket-fence effect*: In the DFT spectrum, all the frequency bins are equally spaced by the reciprocal of the length of the data. If the frequency of the signal falls exactly in between two bins, then its energy is distributed between these two bins, resulting in the distortion of the spectrum. This type of distortion is called the *scalloping loss* or *picket-fence effect*. This parameter is obviously maximum if it falls exactly halfway between any two frequency bins as shown in Figure 5.9. Scalloping loss can be defined as the ratio of the coherent gain ( $G$ ) for a signal frequency component located exactly midway between two DFT bins to the coherent gain of a signal frequency component located exactly at a DFT bin. It is defined as [2].

$$\text{Scalloping loss} = \frac{\left| \sum_{n=0}^{N-1} f[n] e^{-j \frac{2\pi n}{N}} \right|}{\sum_{n=0}^{N-1} f[n]} = \frac{|F(\frac{\omega_s}{2N})|}{F(0)}. \quad (5.44)$$

This can be reduced either by selecting a proper value of the length of finite extent data or by increasing the length of the observation



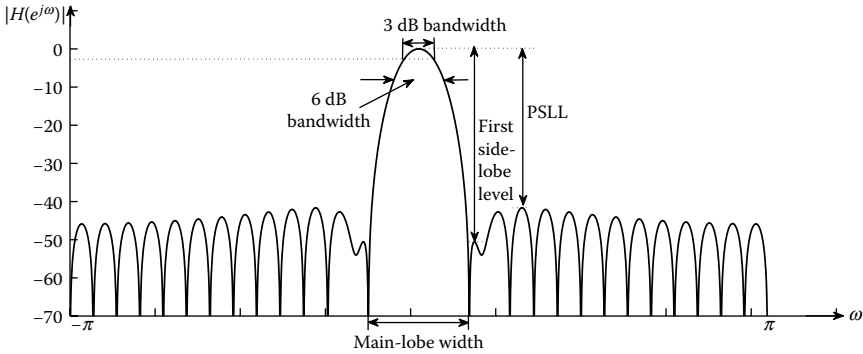
**FIGURE 5.9**

Frequency of signal halfway between two bins.

interval. Yet another solution involves arranging the harmonic frequencies to be more closely spaced and coincident with the signal frequencies. This may be achieved by zero-padding (adding zeros) the true data. However, these additional zeros will not contribute to any new information.

7. *Worst-case processing loss (WCPL)*: The WCPL is the sum of maximum scalloping loss due to the worst-case frequency allocation and the degradation loss due to the window. Therefore, this is a measure of the reduction of output SNR resulting from the combination of the window function and the worst-case frequency locations. This is essentially a measure of the worst-case reduction of SNR, and for any good window this must be less than 3.8 dB. Therefore, from Table 5.3, it can be seen that the windows such as rectangular, Cauchy ( $\alpha \geq 4$ ), Gaussian ( $\alpha = 3.5$ ), Poisson ( $\alpha = 4$ ), and Blackman–Harris (four-term,  $-92$  dB) are to be avoided. It can be observed from Table 5.3 that the WCPL value for these cases ranges from 3 to 4.3. It is interesting to note that the WCPL is always less than 3.8 for all the other windows considered here, including the modified zeroth-order Kaiser–Bessel window family and the modified first-order Kaiser–Bessel window family. The performance of the parameters of these two Kaiser–Bessel window families is compared in Tables 5.4 and 5.5.
8. *Overlap correlation*: When a window is applied to a signal, the data sequence gets tapered to zero at the edges, which will obviously lead to loss of information. Short-duration events occurring at the ends of the observation window will be neglected, if the nonoverlapped window is considered. This can be avoided by processing the sequence with overlapping windows. The amount of overlap required varies from 20% in the case of the rectangular window to 76% in the Blackman–Harris window case. However, in general, 50% or 75% overlap is necessary to retain most of the information in the signal. We obtain more number of segments for averaging in the overlapped case, which reduces the variance of the signal's spectrum. This also has the effect of the reduction of noise power to some extent.
9. *Peak side-lobe level (PSLL) and first side-lobe level (FSLL)*: We have seen in Chapter 2 that the noncoherent component causes spectral leakage. This leakage might sometimes dominate nearby weak harmonics. The parameters that helps in the selection of a window to detect these weak harmonics are (i) the RFSLL, (ii) the PSLL, and (iii) the first side-lobe level (FSLL). It is to be noted that for some windows, the PSLL need not necessarily be the first side-lobe level. For example, the Hamming window has a first side-lobe level  $-44$  dB while its PSLL is about  $-42$  dB. Therefore, the PSLL is higher than the FSLL (see Figure 5.10). Therefore, the Hamming window is a preferred choice in applications where the immediate side-lobe rejection is of

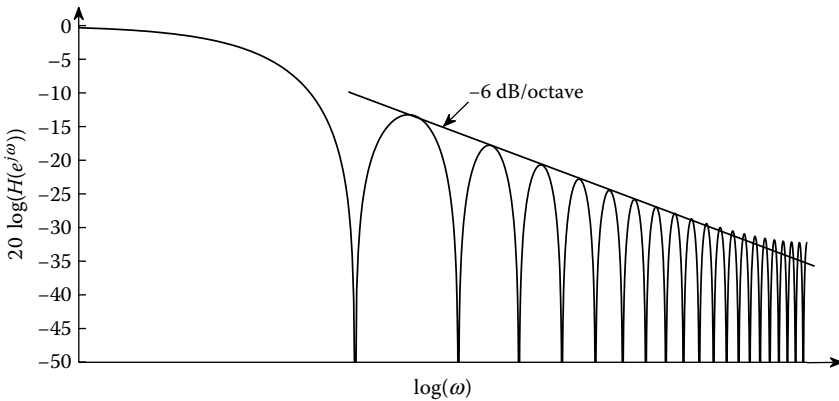




**FIGURE 5.10**  
DTFT of Hamming window.

major concern. The PSLL for all the windows mentioned are listed in Tables 5.3 through 5.5.

10. *Rate of fall-off of side-lobe level (RFSLL):* The RFSLL is yet another parameter with which we must be concerned with. The unit of RFSLL is in terms of decibels per octave (dB/oct). The graphical computation of RFSLL for the rectangular window is illustrated in Figure 5.11. This figure shows that for higher frequencies, the side lobe goes down by about  $-6$  dB/octave. If we require attenuation at higher frequencies, then we should select a window that has a more rapid RFSLL. For example, the Hann window is better suited in such an application than a Hamming window, since the RFSLL in this case is  $-18$  dB/octave.



**FIGURE 5.11**  
RFSLL for rectangular window.

11. *Variance compensation factor (Q)*: The variance compensation factor,  $Q$ , is defined as follows:

$$Q = \frac{1}{N} \sum_{n=0}^{N-1} f^2[n]. \quad (5.45)$$

The attenuation caused due to data smoothing can be compensated by dividing the spectral estimates from the DFT by the value of the compensating factor,  $Q$ . The value of  $Q$  in the case of the rectangular window is unity and is always less in the case of other windows.

All the above-mentioned parameters are also computed for the variable window function families, namely, the modified zeroth-order Kaiser–Bessel and the modified first-order Kaiser–Bessel families, and are presented in Tables 5.4 and 5.5, respectively. The parameters given in Table 5.3 through 5.5 are calculated for a window length of  $N = 57$ . Owing to aliasing in the frequency-domain, these parameters vary slightly with  $N$ .

---

## 5.4 Time–Bandwidth Product

The time–bandwidth product is usually defined for deterministic and finite energy signals. Although a signal  $f[n]$  cannot be both time-limited and band-limited simultaneously, its characteristics can be defined by using an equivalent time width and bandwidth. These definitions hold for signals that are real, nonnegative, and symmetric, and which exhibit maximum value at  $n = 0$  [8]. Therefore, this is applicable for almost all the window functions. The equivalent time width  $N_e$  is defined as

$$N_e = \frac{\sum_{n=-\infty}^{\infty} f[n]}{f[0]}. \quad (5.46)$$

The time width is defined as the width of the rectangular signal with height  $f[0]$  and area equal to that of the signal  $f[n]$ . Similarly, the equivalent bandwidth  $B_e$  can be defined in the frequency-domain as

$$B_e = \frac{\frac{1}{2\pi} \int_{-\pi}^{\pi} F(e^{j\omega}) d\omega}{F(0)}. \quad (5.47)$$

From the definition of the DTFT and IDTFT expressions, we obtain the following relationships:

$$f[0] = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(e^{j\omega}) d\omega, \quad (5.48)$$

$$F(0) = \sum_{n=-\infty}^{\infty} f[n]. \quad (5.49)$$

By substituting the expressions for  $f[0]$  and  $F(0)$  in Equations 5.46 and 5.47 and after canceling the common terms in the product of  $N_e$  and  $B_e$ , we obtain

$$N_e B_e = 1. \quad (5.50)$$

From the equivalent time–bandwidth product, it is clear that the spectral resolution is approximately the reciprocal of the observation interval. Hence, the length of the finite extent data ( $N$ ) can be selected from Equation 5.50 to meet the required spectral resolution as well as variance.

## 5.5 Applications of Windows

In present-day applications, we rely more on DSPs rather than their analog counterparts. We now consider two important applications of windows, namely, filter design and spectral analysis.

### 5.5.1 FIR Filter Design Using Windows

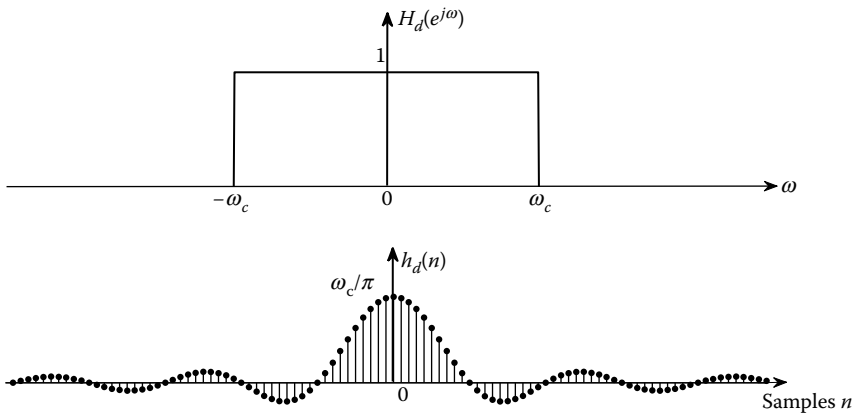
For implementing a filter in a DSP, we often prefer to use linear-phase FIR filters instead of infinite impulse response (IIR) filters [9,10]. However, the discontinuities in the frequency response of ideal filters result in infinite length impulse response. The desired frequency response of an ideal lowpass filter is given by

$$H_d(e^{j\omega}) = \begin{cases} 1, & 0 \leq |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| < \pi \end{cases} \quad (5.51)$$

and its impulse response is given by

$$h_d[n] = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty \leq n \leq \infty. \quad (5.52)$$

Figure 5.12 shows the ideal lowpass filter frequency response and its impulse response. The method of FIR filter design from the ideal IIR filter involves



**FIGURE 5.12**  
Ideal lowpass filter and its impulse response.

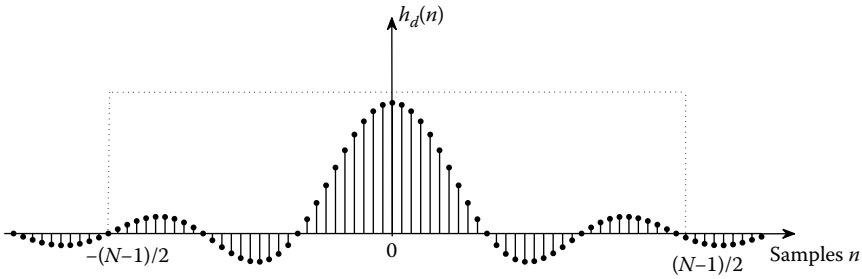
an approximation of the ideal filter response by a practically realizable filter. There are many techniques available for the design of a linear-phase FIR filter. All these techniques are based on some approximation criterion or a measure of goodness for the response of a designed filter, in comparison to the ideal desired response. The most popular design uses direct optimization techniques [11,12]. However, the simplest method for FIR filter design is called the impulse response truncation (IRT) method or the *windowing* method. The major drawback of this method in contrast to the *optimization* method is that in IRT it is impossible to design a filter of minimal length that meets the frequency response requirements in the passband and stopband. This method is based on the truncation of the Fourier series of the input sequence. Every desired filter response  $H_d(e^{j\omega})$  of a nonrecursive filter is a periodic function in  $\omega$  with period  $2\pi$ . Therefore,  $H_d(e^{j\omega})$  can be represented as a Fourier series as follows:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n]e^{-j\omega n}. \tag{5.53}$$

The coefficients of this Fourier series can be recognized as being equal to the impulse response of a digital filter given by

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega. \tag{5.54}$$

In general, the function  $h_d[n]$  designed is of infinite length and is noncausal (see Figure 5.12). The simplest method to design an FIR filter is to truncate the Fourier series for a desired length of the filter (as shown in Figure 5.13) and then apply a right shift of  $(\frac{N-1}{2})$  samples to make it causal, as depicted in



**FIGURE 5.13**  
Truncation of impulse response.

Figure 5.14. Hence, we can obtain the impulse response of the desired filter as

$$h[n] = \begin{cases} h_d[n], & 0 \leq n \leq N - 1 \\ 0, & \text{otherwise.} \end{cases} \quad (5.55)$$

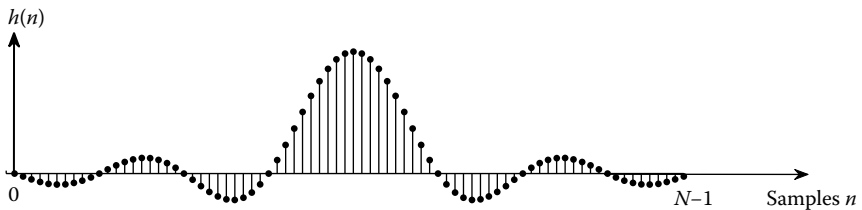
If we carefully examine Equation 5.55, we can conclude that this is nothing but a rectangular windowing operation. Hence, the impulse response of the filter can be rewritten as

$$h[n] = h_d[n]w[n], \quad (5.56)$$

where  $w[n]$  is a rectangular window defined as

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N - 1 \\ 0, & \text{otherwise.} \end{cases} \quad (5.57)$$

To understand the characteristics of the frequency response of the designed filter, it would be easier if we investigate the effect of this windowing operation in the frequency-domain. In the frequency-domain, this operation is represented by a convolution operation of  $H_d(e^{j\omega})$  and the frequency response



**FIGURE 5.14**  
Shifted impulse response.

of the rectangular window  $W(e^{j\omega})$  as follows:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta. \quad (5.58)$$

This convolution operation is depicted pictorially in Figure 5.15.

Let us now take an example of the FIR filter design using the windowing approach. The frequency response and the impulse response of an ideal lowpass filter were given in Figure 5.12. The magnitude response of the rectangular window is provided in Figure 5.16. The rectangular window has a narrower main lobe and larger side lobes when compared to the other common windows.

The frequency response of the resulting lowpass filter is shown in Figure 5.17. Here, we can see that the frequency response of the filter is not flat since it has ripples in the passband and the stopband. Besides, the transition between the passband and the stopband is not sharp. There is a large transition band in between. These features of the filter are due to the direct truncation of the IIR. When we convolve  $H_d(e^{j\omega})$  with the window response, the main lobe of the window results in a transition band between the passband and the stopband. The side-lobes of the window give rise to the ripples in the passband and stopband. Hence, the transition width of an FIR lowpass filter designed using the windowing method is proportional to the main-lobe width of the window and the ripples are proportional to the side-lobe levels of the window.

To compare the performance of the FIR lowpass filter designed using different windows, we show the effects of the Hann and Hamming windows. The Hann window and its Fourier transform are shown in Figure 5.18. The Hann window and the Hamming window have larger main-lobe widths and smaller side-lobe levels when compared to the rectangular window.

Hence, the FIR lowpass filter designed using the Hann window and the Hamming window have a larger transition bandwidth and smaller ripples. Figures 5.19 and 5.20 show the frequency responses of FIR lowpass filters

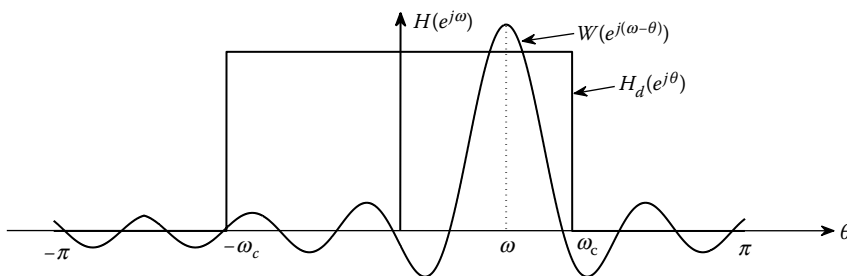
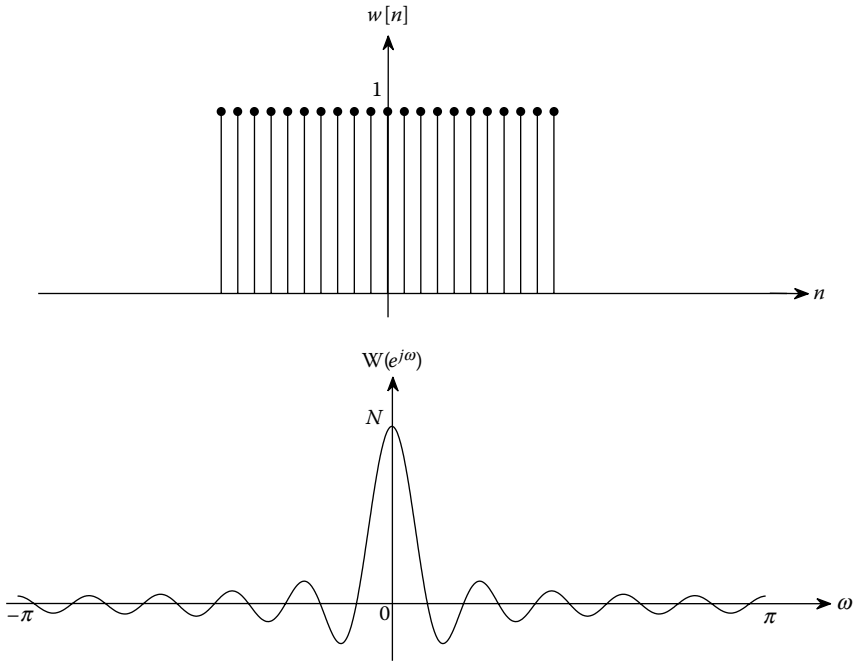


FIGURE 5.15

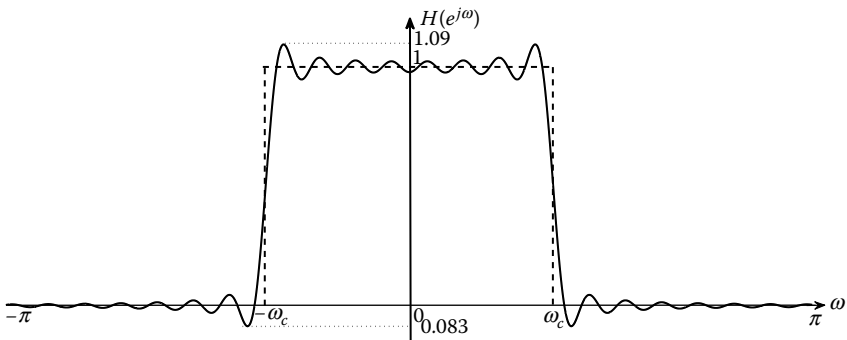
Convolution of the desired frequency response and the rectangular window response.



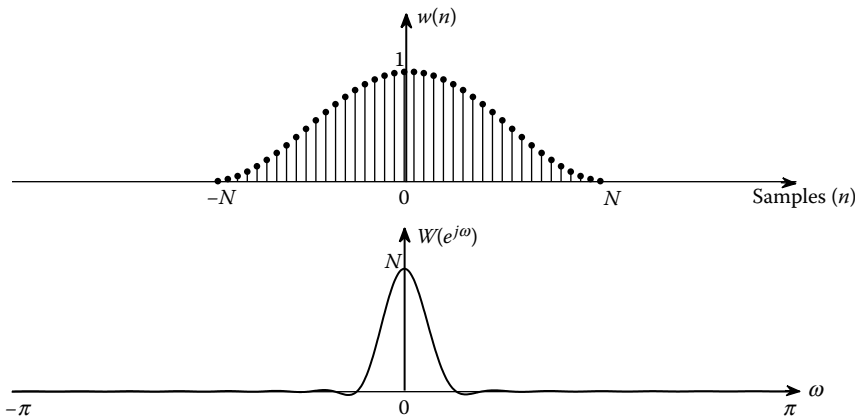
**FIGURE 5.16**  
Rectangular window in time-domain and frequency-domain.

designed using these two windows. The transition bandwidth of both are almost equal, but the ripples due to the Hamming window are lower, as expected.

The log magnitude frequency response of FIR lowpass filter using rectangular, Hann, and Hamming windows are shown in Figures 5.21 through 5.23.



**FIGURE 5.17**  
Response of FIR lowpass filter using rectangular window.



**FIGURE 5.18**  
Hann window in time-domain and in frequency-domain.

### 5.5.2 Spectral Analysis

The PSD of a random signal can be estimated by using two classical techniques based on Fourier analysis, called the *periodogram* and the *correlogram* [8,13,14]. In the periodogram method, the PSD is calculated from

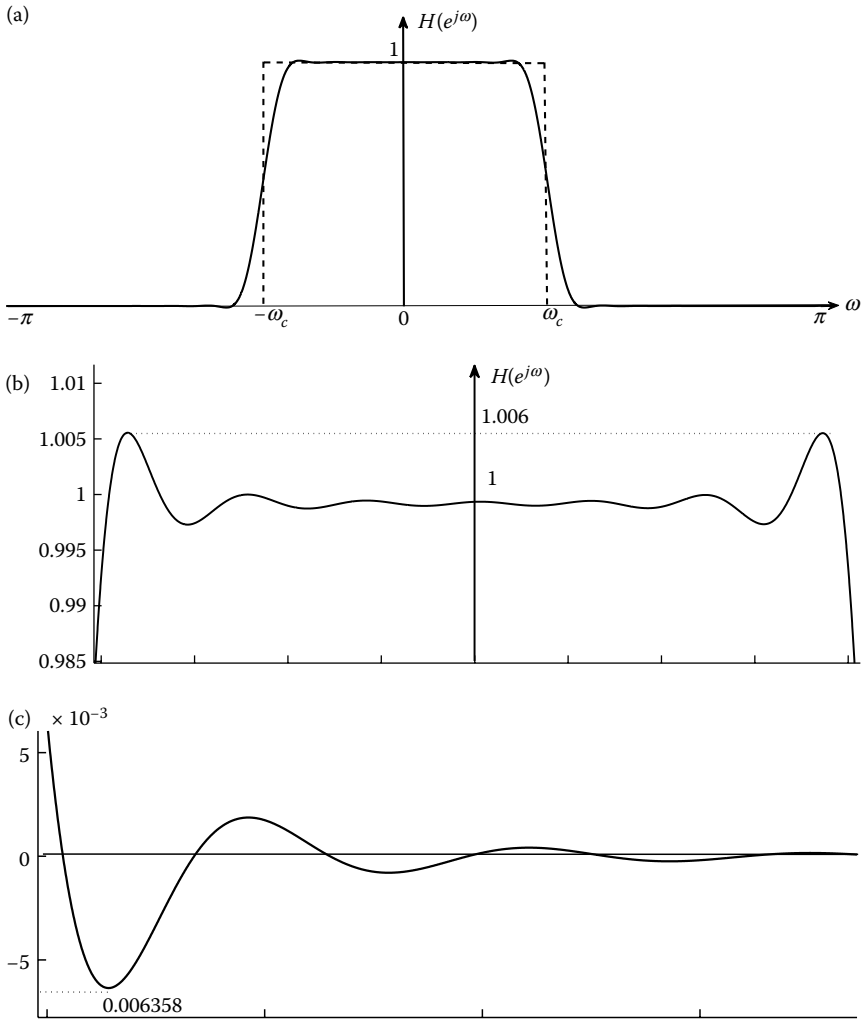
$$P_{xx}(f) = \lim_{M \rightarrow \infty} \xi \left[ \frac{1}{2M + 1} \left| \sum_{n=-M}^M x[n]e^{-j2\pi fn} \right|^2 \right]. \tag{5.59}$$

As  $M \rightarrow \infty$ , the periodogram approaches to its original PSD. However, in practical applications, we have access to only one set of samples for the estimation of PSD. Therefore, we can assume that the random signal is ergodic, that is, the time series properties are assumed to remain the same. Now, the PSD of the periodogram can be computed as

$$P_{PER}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n]e^{-j2\pi fn} \right|^2. \tag{5.60}$$

Although the bias in the estimated PSD tends to zero as the length of the finite extent data increases, the variance of the estimate remains unchanged because of the ergodicity assumption. In fact, the variance here will be very high and comparable to the mean itself.

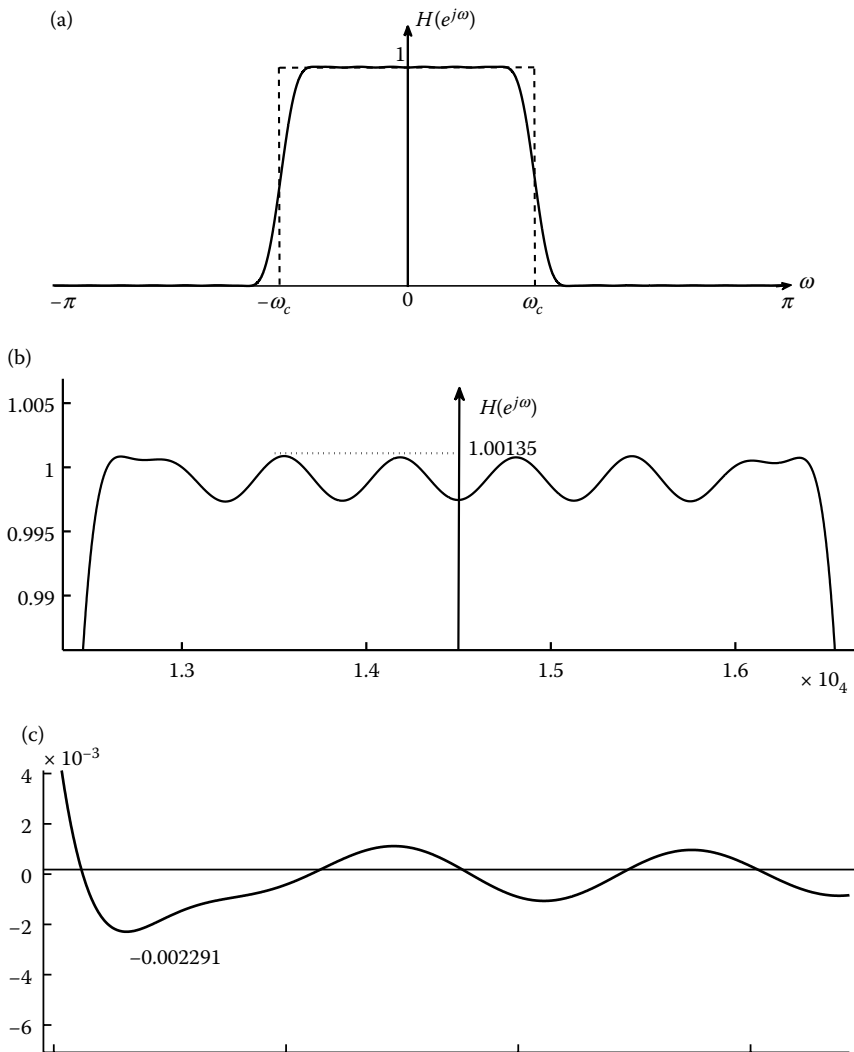




**FIGURE 5.19** Response of FIR lowpass filter (a) using Hann window. (b) Zoomed plot of passband. (c) Zoomed plot of stopband.

The variance of the signal can be reduced to some extent using the *Welch* method of periodogram. Here, the entire available data is divided into  $L$  segments, of  $N$  length each. The periodogram is computed using Equation 5.60 for each segment and finally averaged using the following expression:

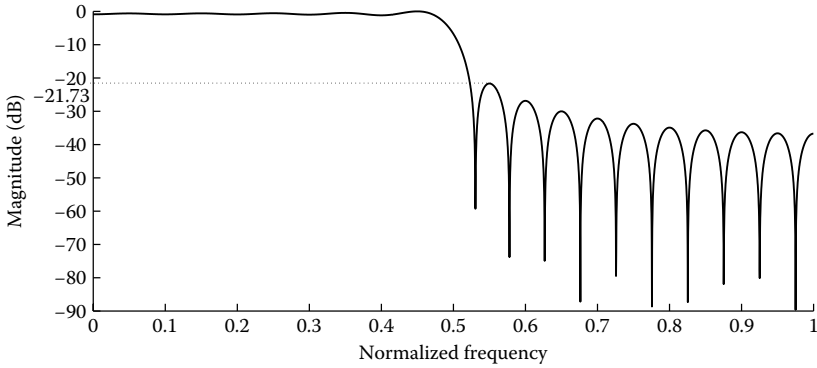
$$P_{avg}(f) = \frac{1}{L} \left| \sum_{m=0}^{L-1} P_{PER}^{(m)}(f) \right|, \tag{5.61}$$



**FIGURE 5.20** Response of FIR lowpass filter (a) using Hamming window. (b) Zoomed plot of passband. (c) Zoomed plot of stopband.

to obtain a better variance. Here,  $P_{PER}^{(m)}(f)$  is the periodogram of the  $m$ th segment. The variance in this case is reduced by  $\frac{1}{L}$ .

The sharp truncation of the segments leads to spurious peaks in the periodogram, thus increasing the bias. By changing the number of segments, either the bias or the variance can be controlled, but not both simultaneously. To reduce the bias in the estimate, each segment is multiplied with a data window. Data windows taper the data near the edges of the segments slowly



**FIGURE 5.21**  
Log magnitude response due to rectangular window.

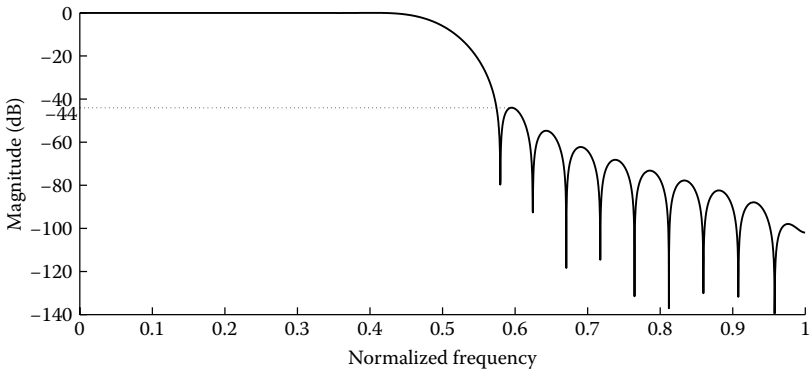
to zero, which reduces the bias. In general, overlapped segments are used to obtain more segments for averaging, thereby reducing the variance.

From the *Wiener–Khinchin theorem*, the PSD can also be estimated using the autocorrelation function as follows:

$$P_{COR}(f) = \sum_{k=-(N-1)}^{N-1} r_{xx}[k]e^{-j2\pi fk}, \tag{5.62}$$

where  $r_{xx}$  is the autocorrelation function, computed as

$$r_{xx}[k] = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} x^*[n]x[n+k], & k = 0, 1, \dots, N-1 \\ r_{xx}^*[k], & k = -(N-1), -(N-2), \dots, -1. \end{cases} \tag{5.63}$$



**FIGURE 5.22**  
Log magnitude response due to Hann window.

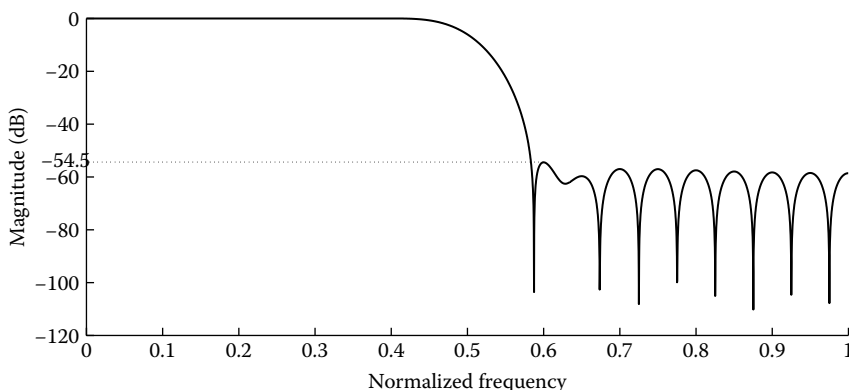


FIGURE 5.23

Log magnitude response due to Hamming window.

The problems of high bias and variance exist even in this method. To alleviate these effects, the data is first segmented and the autocorrelation is computed for individual sections. Here, both the bias and variance problems arise due to the autocorrelation estimator. As the lag increases, the number of product terms used for averaging will decrease, thereby increasing the variance. For  $k = N - 1$ , only one product term ( $x^*[0]x[N - 1]/N$ ) is used for averaging. Hence, the autocorrelation at these lags must be given less weightage. This can be done by multiplying the autocorrelation function with a lag window ( $w[k]$ ) and the expression for this is given by

$$P_{\text{COR}}(f) = \sum_{k=-(N-1)}^{N-1} w[k]r_{xx}[k]e^{-j2\pi fk}. \quad (5.64)$$

This method of estimating the PSD is called the *Blackman–Tukey* spectral estimation [15]. The lag window is always defined to be symmetrical about zero.

### 5.5.3 Window Selection for Spectral Analysis

In this section, we provide some guidelines that will enable the user to select appropriate windows for spectral analysis. In spectral analysis, the side lobes cause smearing or spreading of energy, while the main lobe is responsible for appropriate smoothing effects. Since the energy of a spectral window is constant, if the side lobes are to be reduced, the main-lobe width has to be increased and vice versa. The side-lobe level and the main-lobe width cannot be reduced simultaneously for a fixed window length. Thus, the shape of the spectral window should be selected appropriately to meet the desired specifications. A prior knowledge of the PSD is required for better estimation. Let

us consider different applications and study the effect of different windows (rectangular, Hann, and Hamming windows) for each of the following cases.

*Case 1: Two closely spaced frequency components of almost equal strengths*—If there are two frequency components that are closely spaced with equal strengths, then the appropriate window function to resolve both the frequency components is the rectangular window. This is due to the fact that the rectangular window has the smallest main-lobe width; hence, the smoothing effect will be minimum and both the spectral peaks can be seen distinctly. In the case of the Hann and Hamming windows, owing to wider main lobes, the smoothing effects will be quite significant and the two spectral peaks may merge into a single peak. For this application, the Hamming window gives a better result than the Hann window because of a smaller first side-lobe level than the PSL. This advantage will become more clear in Case 2. For example, let us take a signal

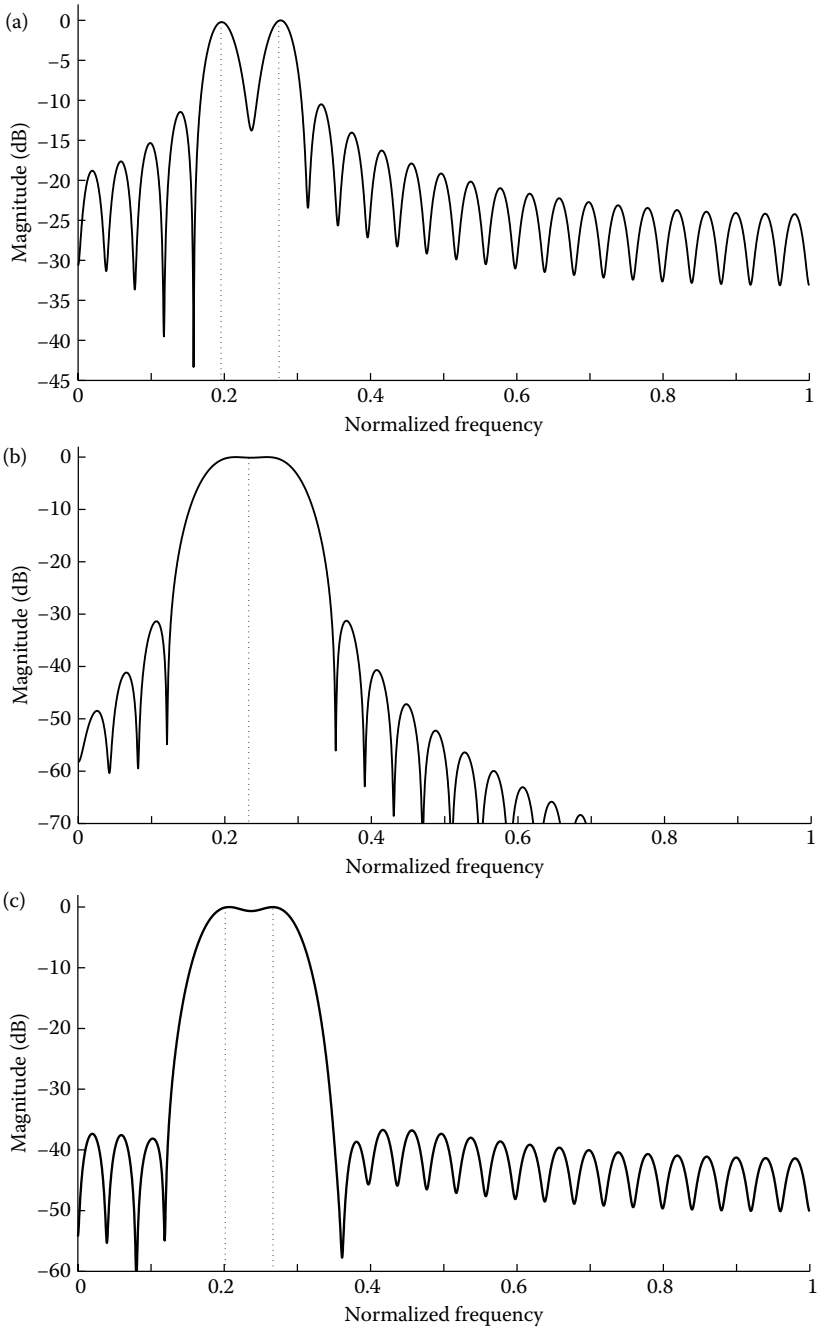
$$x[n] = \cos(2\pi n5/50) + \cos(2\pi n6.81/50), \quad n = 0, 1, 2, \dots, 49.$$

The spectral response of this signal using the rectangular, Hann, and Hamming windows is shown in Figure 5.24. Here, we can see that the rectangular window resolves both the spectral components distinctly, whereas the Hann and Hamming windows show it as a single component. However, in the Hamming case, we can still see a small separation at the top of the peak.

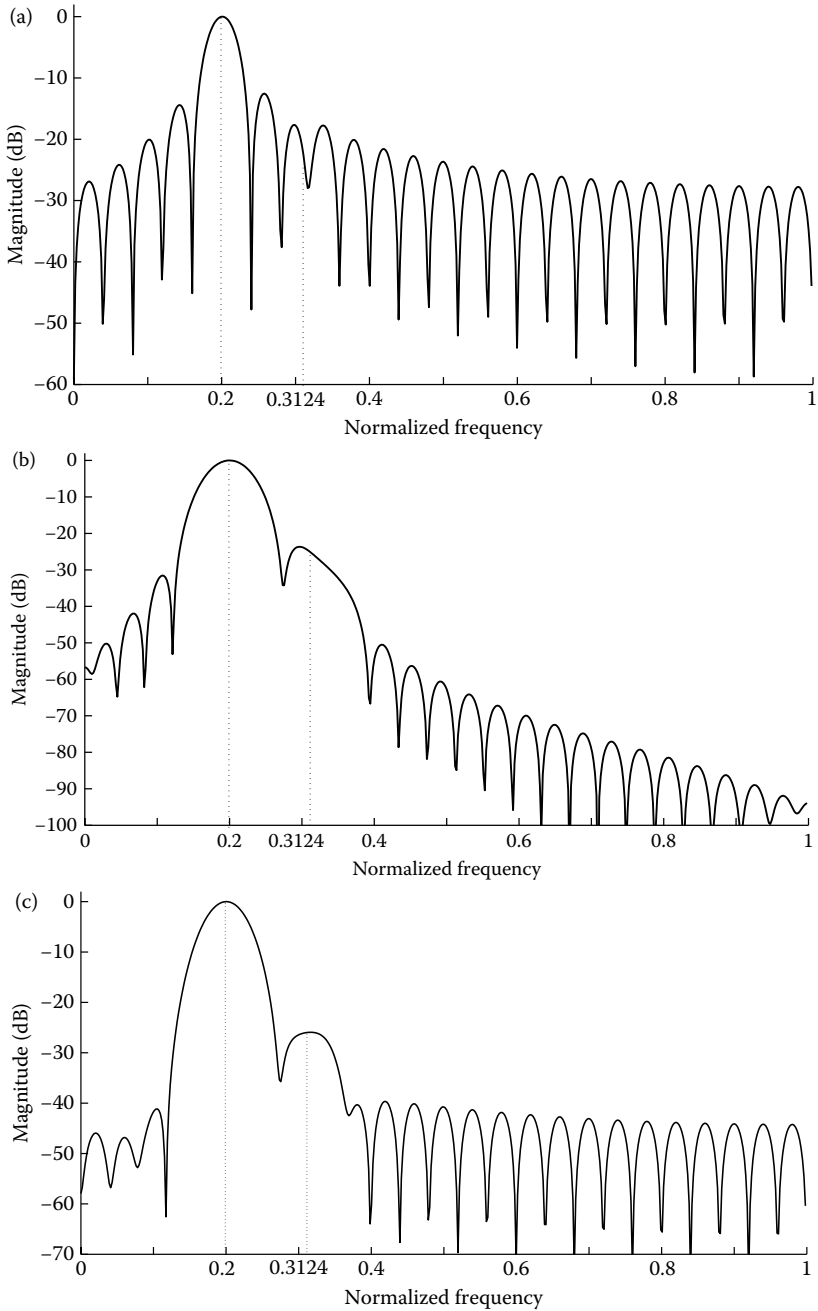
*Case 2: Two closely spaced frequency components with unequal strengths*—In this case, the PSL of the window also plays an important role. If one of the components is very weak in magnitude, then it gets submerged in the side lobes of the strong component. This happens in the case of a rectangular window, since its side lobe levels are much higher. Hence, the rectangular window may not even detect the weak spectral component. Thus, the Hamming window is preferred in this application. Even though both the Hamming and Hann windows have the same main-lobe widths, the fact that the first side-lobe level of the Hamming window is smaller than the PSL will prove to be an advantage. Here, the high side-lobe roll-off rate of the Hann window will not have much effect in resolving the weaker spectral component. For instance, let us consider the signal

$$x[n] = \cos(2\pi n5/50) + 0.05 \cos(2\pi n7.81/50), \quad n = 0, 1, 2, \dots, 49.$$

The spectral response of this signal using the rectangular, Hann, and Hamming windows is shown in Figure 5.25. Here, we can clearly observe that the rectangular window cannot detect the presence of the weak spectral component, whereas the Hamming window detects it clearly. The Hann window can



**FIGURE 5.24** Response of two closely spaced equal-strength signal components. (a) Using rectangular window. (b) Using Hann window. (c) Using Hamming window.

**FIGURE 5.25**

Magnitude response of signal with two closely spaced, unequal-strength components. (a) Using rectangular window. (b) Using Hann window. (c) Using Hamming window.

also detect it but the spectral estimate is very poor when compared to the Hamming window. However, as the separation between the spectral components increases, the situation changes, which is discussed in the next case.

*Case 3: Two far-away frequency components with unequal strengths*—In the presence of a very weak spectral component that is far away from the stronger spectral component, the Hann window turns out to be most appropriate. In such an application, the high side-lobe roll-off rate of the Hann window will be advantageous. As discussed before, the rectangular window will submerge the weak spectral component in the side lobes of the stronger component, since the sidelobe level is very high (and its falloff rate is slower). In the case of the Hamming window, although the side-lobe levels are lower than the Hann window, it takes longer time for the side lobes to fall off as its rate is slower. The Hann window has high RFSLL, which makes the side lobes to fall faster, even though the PSL is slightly higher than the Hamming window. For example, let us consider a signal

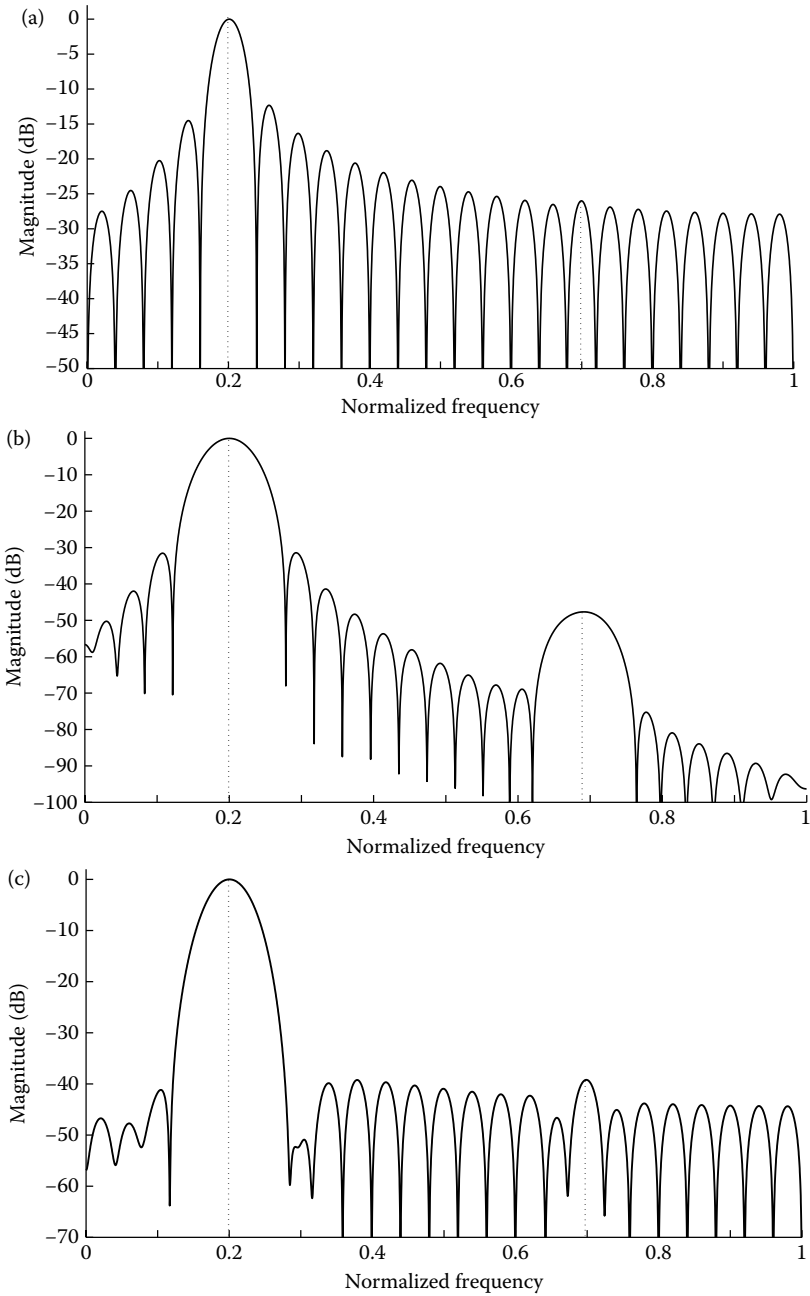
$$x[n] = \cos(2\pi n5/50) + 0.005 \cos(2\pi n17.26/50), \quad n = 0, 1, 2, \dots, 49.$$

The spectral response of this signal using the rectangular, Hann, and Hamming windows is shown in Figure 5.26. Here, we can clearly see that only the Hann window is able to detect the weak spectral component clearly. In short, if the immediate side-lobe rejection is important in an application, then using the Hamming window is the best option. Instead, if the far-off side-lobe rejection is the desired criterion, then the Hann window is the most preferred choice.

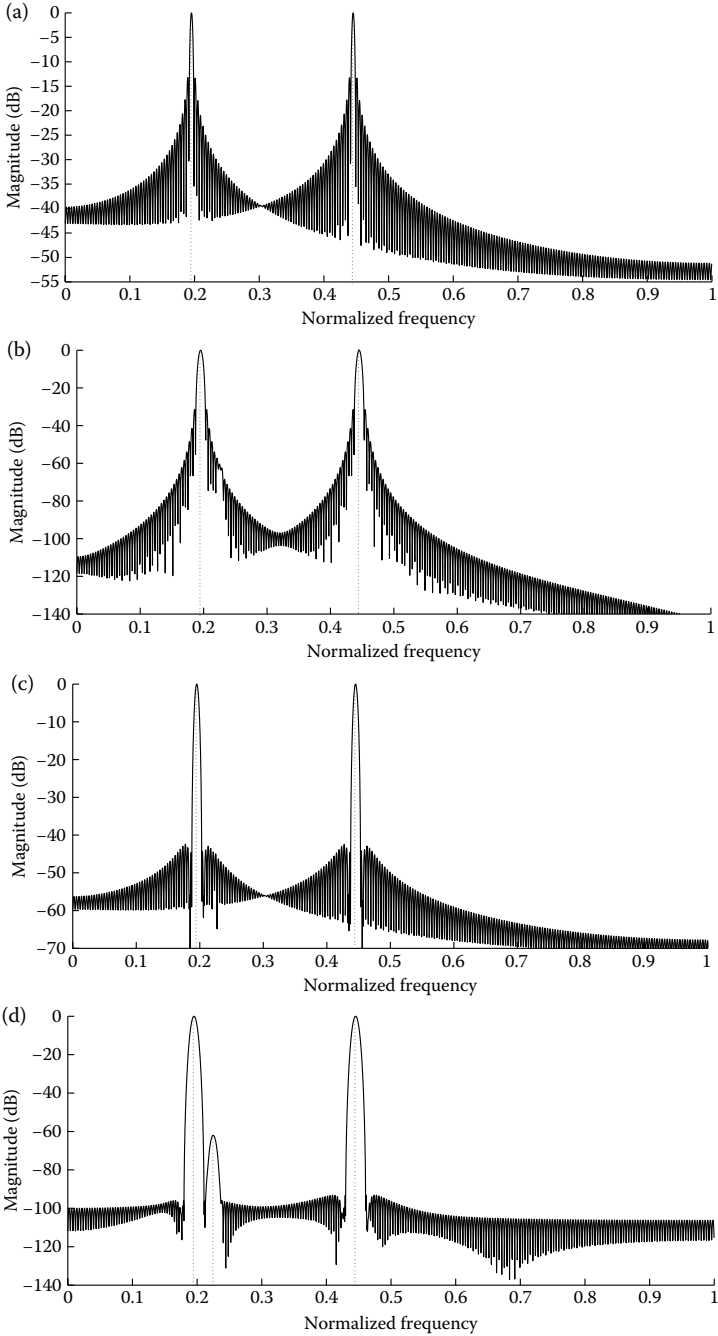
*Case 4: Weak component in the presence of moderate signal component, both close and distant in frequency*—The spectral response for case 4 is shown in Figure 5.27. We can observe that there are frequency components both nearby and distant; hence, a window with equiripple side lobes around the main lobe is required to keep the bias small. In this case, the side-lobe falloff rate is not important because of the presence of the nearby component. Hence, for this application, the Dolph–Chebyshev window is preferred because of its equiripple characteristics. Figure 5.27 clearly shows that only the Dolph–Chebyshev window can resolve the weak component in the presence of a moderately near and a far-away component.

In conclusion, we use window functions, other than rectangular, to obtain a compromise between a narrow main lobe (for high resolution) and low side lobes (for low spectral leakage). High resolution provides accurate estimates of the frequency of a sinusoid and results in the separation of two sinusoids that are closely spaced in frequency. Low spectral leakage improves the detectability of a weak sinusoid in the presence of a strong sinusoid that is not bin-centered. A detailed procedure for the FIR filter design using the window method is presented in Chapter 7. All the issues discussed with respect to spectral analysis will be detailed in Chapter 8.



**FIGURE 5.26**

Magnitude response of signal having two far-off, unequal-strength components. (a) Using rectangular window. (b) Using Hann window. (c) Using Hamming window.



**FIGURE 5.27** Response of weak component in the presence of moderate component (both close and distant in frequency). (a) Using rectangular window. (b) Using Hann window. (c) Using Hamming window. (d) Using Dolph-Chebyshev window.

---

## References

1. K.M.M. Prabhu, *Data Windows in Digital Signal Processing*, PhD Thesis, IIT Madras, India, October 1980.
2. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *Proceedings of IEEE*, pp. 51–83, 1978.
3. A.H. Nuttall, Some windows with very good sidelobe behaviour, *IEEE. Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 84–91, February 1981.
4. G. D'Antona and A. Ferrero, *Digital Signal Processing for Measurement Systems*, Springer Media, Inc., New York, NY, 2006, pp. 70–72.
5. T. Saramäki, A class of window functions with nearly mini-mumsidelobe energy for designing FIR filters, *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS '89)*, vol. 1, pp. 359–362, Portland, OR, May 1989.
6. S.W.A. Bergen and A. Antonio, Design of ultraspherical window functions with prescribed spectral characteristics, *EURASIP Journal on Applied Signal Processing*, 2053–2065, January 2004.
7. D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss, *Digital Signal Processing: A System Design Approach*, Wiley, 1988.
8. P. Stoica and R. Moses, *Spectral Analysis of Signals*, Pearson Education Inc., 2005.
9. A. Antoniou, *Digital Filters: Analysis and Design*, McGraw-Hill, 1979.
10. R.G. Lyons, *Understanding Digital Signal Processing*, Pearson Education Inc., 3rd Edn., 2011.
11. E.Ya. Remez, *General Computational Methods of Tchebycheff Approximation*, Kiev (Atomic Energy Commission Translation 4491), pp. 1–85, 1957.
12. T.W. Parks and J.H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, 1972.
13. S.M. Kay, *Modern Spectral Estimation: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
14. S.L. Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
15. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectra*, Dover Publications, New York, NY, 1958.

# 6

---

## *Time- and Frequency-Domain Implementations of Windows*

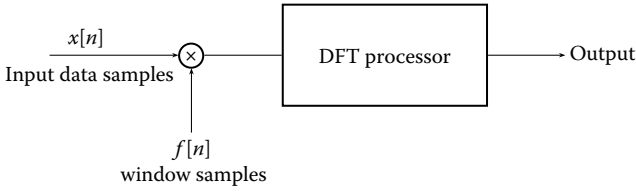
---

This chapter presents the implementation details of windows in the time and frequency-domains. While computing the discrete Fourier transform (DFT), the inevitable truncation of the input time sequence causes all the frequency components of the input signal to interfere with one another [1–4]. This phenomenon is often called frequency *leakage* or *smearing*, as it leads to the spreading of energy. Owing to this leakage, the computed spectrum differs from the true spectrum. To reduce this unwanted effect, the input data is usually multiplied by a suitable window function before performing the DFT [5–9]. For a special class of windows, windowing can alternatively be implemented in the frequency-domain [7–10]. We consider the implementation of many windows which belong to that class of windows such as Hann, Hamming, Blackman, raised-cosine, and so on, both in the time and in the frequency-domains. Computer simulation studies have been carried out to determine the error performance of these implementations in both the domains [10,11]. Efficient hardware structures for windowing in the frequency-domain are also presented [7,8].

---

### 6.1 Time-Domain Implementation

In the time-domain scheme, a block of  $N$  data samples,  $x[n]$ , is multiplied by  $N$  window samples,  $f[n]$ , before performing the DFT. The fast Fourier transform (FFT) is an efficient tool to compute the DFT. The special-purpose hardware that implements the FFT algorithm is called the FFT processor. The time-domain windowing technique is illustrated in Figure 6.1. The realization given in Figure 6.1 requires  $(N + 1)/2$  stored samples (since the window is symmetric) of the window with an odd length  $N$ , or  $(\frac{N}{2} + 1)$  samples with an even length  $N$ , and  $N$  number of multiplications [9]. Since there is no single efficient window that could be used in spectral estimation for all types of data, samples of several windows may be stored in a read-only memory (ROM) or in a programmable ROM. A special class of *programmable windowing* schemes



**FIGURE 6.1**  
Time-domain scheme.

that can be used to implement several windows in the frequency-domain is outlined in the next section.

## 6.2 A Programmable Windowing Technique

In this section, we describe the implementation of a special class of windows [9]. Its discrete version of the time function can be represented in the form:

$$f[n] = a - 2b \cos\left(\frac{2\pi n}{N}\right) + 2c \cos\left(\frac{4\pi n}{N}\right) - 2d \cos\left(\frac{6\pi n}{N}\right),$$

$$n = 0, 1, \dots, (N - 1) \quad (6.1)$$

If the sample  $f(N/2)$  has to be unity, we must have  $(a + 2b + 2c + 2d) = 1$ . The values of these constants  $a$ ,  $b$ ,  $c$ , and  $d$  can be chosen depending on the type of window preferred: for example,

- i.  $a = 0.5$ ,  $b = 0.25$ ,  $c = 0$ , and  $d = 0$  gives the Hann window;
- ii.  $a = 0.54$ ,  $b = 0.23$ ,  $c = 0$ , and  $d = 0$  yields the Hamming window and
- iii.  $a = 0.42$ ,  $b = 0.25$ ,  $c = 0.04$ , and  $d = 0$  gives the Blackman window.

We now consider a finite-valued sequence,  $x[n] = \{x[0], x[1], x[2], \dots, x[N - 1]\}$ . The DFT of this sequence  $x[n]$  is given by

$$A[r] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi rn/N), \quad r = 0, 1, \dots, (N - 1). \quad (6.2)$$

The DFT coefficients,  $A[r]$ , represent the unsmoothed spectrum of  $x[n]$ . Multiplying the data with a window,  $f[n]$ , and computing the DFT yields the smoothed spectrum,  $F[r]$ , which is given by

$$F[r] = \sum_{n=0}^{N-1} \{x[n]f[n]\} \exp(-j2\pi rn/N), \quad r = 0, 1, \dots, (N - 1). \quad (6.3)$$

Substituting Equation 6.1 in Equation 6.3, we get

$$F[r] = \sum_{n=0}^{N-1} x[n] \left[ a - 2b \cos\left(\frac{2\pi n}{N}\right) + 2c \cos\left(\frac{4\pi n}{N}\right) - 2d \cos\left(\frac{6\pi n}{N}\right) \right] \times \exp(-j2\pi rn/N), \quad r = 0, 1, \dots, (N-1). \quad (6.4)$$

By expressing the cosine terms in their exponential forms as

$$\begin{aligned} \cos\left(\frac{2\pi n}{N}\right) &= \frac{\exp(-j2\pi n/N) + \exp(j2\pi n/N)}{2}, \\ \cos\left(\frac{4\pi n}{N}\right) &= \frac{\exp(-j4\pi n/N) + \exp(j4\pi n/N)}{2}, \quad \text{and} \\ \cos\left(\frac{6\pi n}{N}\right) &= \frac{\exp(-j6\pi n/N) + \exp(j6\pi n/N)}{2}, \end{aligned} \quad (6.5)$$

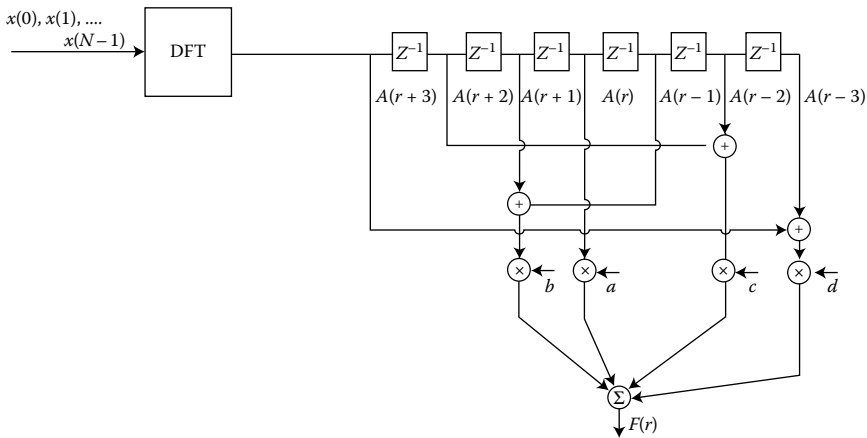
and substituting these terms in Equation 6.4 and collecting the appropriate exponential terms, we obtain the following expression:

$$\begin{aligned} F[r] &= a \sum_{n=0}^{N-1} x[n] \exp(-j2\pi rn/N) \\ &\quad - b \sum_{n=0}^{N-1} x[n] \{ \exp(-j2\pi n(r+1)/N) + \exp(-j2\pi n(r-1)/N) \} \\ &\quad + c \sum_{n=0}^{N-1} x[n] \{ \exp(-j2\pi n(r+2)/N) + \exp(-j2\pi n(r-2)/N) \} \\ &\quad - d \sum_{n=0}^{N-1} x[n] \{ \exp(-j2\pi n(r+3)/N) + \exp(-j2\pi n(r-3)/N) \}, \\ &\quad r = 0, 1, \dots, (N-1). \end{aligned} \quad (6.6)$$

By using the definition of the DFT as given in Equation 6.2, we can reduce Equation 6.6 as follows:

$$\begin{aligned} F[r] &= aA[r] - b\{A[r+1] + A[r-1]\} + c\{A[r+2] + A[r-2]\} \\ &\quad - d\{A[r+3] + A[r-3]\}. \end{aligned} \quad (6.7)$$

Owing to the periodicity property of the DFT,  $A[-1] = A[N-1]$ ,  $A[-2] = A[N-2]$ ,  $A[N] = A[0]$ , and in general  $A[\pm r] = A[N \pm r]$ . Thus, Equation 6.7 gives  $F[r]$  for all values of  $r$ , where  $r = 0, 1, \dots, (N-1)$ . We can now say that



**FIGURE 6.2**  
Frequency-domain implementation of windows.

$F[r]$  gives a smoothed spectrum. An efficient scheme for implementing different windows based on the operation given above is shown in Figure 6.2 [9].

The number of multiplications required for the scheme in Figure 6.2 is equal to  $4N$  instead of  $N$  for the block diagram of Figure 6.1. Although this seems to be a serious drawback for the frequency-domain implementation (FDI), it is however not the case. In the circuit of Figure 6.1, one needs a general-purpose multiplier where both inputs have to be represented by a relatively large number of bits. As will be shown in this chapter, for the windows defined in Equation 6.1, the multiplications by  $a, b, c,$  and  $d$  can be implemented by a very small number of shift and add/subtract operations.

### 6.3 Computational Error in Time and Frequency-Domains

In this section, the error performance of the implementation schemes of windows are computed as explained below [9]. In the following discussion, the time-domain implementation (TDI) is carried out as shown in Figure 6.1 and the FDI is realized by the scheme shown in Figure 6.2.

Throughout the computations, the input data samples are quantized to 12 bits and the FFT is computed with fixed-point arithmetic, by choosing the word length to be 12 bits. It may be noted that white Gaussian data is used as the input. The quantized data is multiplied by the unquantized samples of the window and then the FFT is performed. This resulting sequence is taken as the reference (for both the domains). In the TDI, the window samples are represented by  $L$  bits. The quantized data samples are multiplied by the quantized window samples and the FFT of the product is computed. This

result is subtracted from the above reference and the difference is recorded as the error. This is repeated several times ( $\beta$ ) and the variance of the error is computed from these trials, for various values of the number of bits  $L$  and data lengths ( $N$ ).

In case of the FDI, the FFT is performed on the input Gaussian data samples. The resulting DFT coefficients are multiplied by the coefficients  $a$ ,  $b$ ,  $c$ , and  $d$  (depending on the type of window employed) according to Equation 6.7. These coefficients are represented by  $L$  bits, where  $L$  is varied from 7 to 12 bits. In each case, the result obtained from Equation 6.7 is subtracted from the above-mentioned reference and the difference is taken as the error. This procedure is repeated over  $\beta$  number of trials and from the resulting errors, the variance is computed [9].

The signal-to-computational error ratio (SCER) is defined as the ratio of the input signal variance to the output error variance. This ratio is computed in dB, for various word lengths, and they are tabulated in Tables 6.1 through 6.4 for the different windows, namely, Hann, Hamming, Blackman, and raised-cosine windows, respectively. It must be noted that the higher the SCER, the more accurate is the performance.

As seen from Table 6.1, the FDI of the Hann window is far superior to its time-domain counterpart, for all the coefficient word lengths considered [9,10]. This is because the coefficients (0.5 and 0.25) of the Hann window can be represented exactly, whenever the word length is greater than or equal to two bits.

In the case of the Hamming window (Table 6.2), both the TDI and the FDI seem to provide mixed SCER performance for different word lengths and window coefficient lengths [9]. From the results of Table 6.3, which correspond to the Blackman window, it can be observed that the time-domain version yields better SCER for word lengths of 8 bits or less [9,10]. However, for the raised-cosine family, with  $D = 0.0113$  (Table 6.4), the implementation in the time-domain apparently provides better SCER when the word length

**TABLE 6.1**

SCER (in dB) Performance of Hann Window

	$N$	16	32	64	128	256	512
$L$	( $\beta$ )	(500)	(275)	(125)	(50)	(25)	(10)
7	TDI	37.86	35.14	33.21	33.41	30.95	30.50
8	TDI	42.29	40.69	37.00	35.82	33.22	32.21
9	TDI	46.40	44.74	40.15	37.76	34.44	33.44
10	TDI	49.74	45.53	41.45	38.07	35.45	33.95
11	TDI	52.35	45.72	41.36	39.38	35.45	34.15
12	TDI	51.62	46.17	42.58	39.91	35.72	34.32
	FDI	57.95	51.61	46.38	43.54	39.86	38.22

Note:  $L$ : window coefficient length,  $\beta$ : number of FFTs averaged.



**TABLE 6.2**

SCER (in dB) Performance of Hamming Window

$L$	$N$ ( $\beta$ )	16 (500)	32 (275)	64 (125)	128 (50)	256 (25)	512 (10)
7	TDI	35.22	34.07	34.35	33.44	30.95	30.64
	FDI	30.42	30.73	30.10	29.45	28.85	28.23
8	TDI	40.26	38.85	37.57	36.35	33.38	32.31
	FDI	41.44	41.07	39.27	37.96	36.06	34.88
9	TDI	45.58	42.91	40.10	38.20	34.64	33.89
	FDI	41.44	41.07	39.27	37.96	36.06	34.88
10	TDI	49.09	45.87	40.87	39.51	35.19	34.09
	FDI	46.70	45.13	42.45	40.46	37.81	36.42
11	TDI	49.45	47.41	41.52	39.54	35.31	34.15
	FDI	50.45	47.36	43.87	41.68	38.60	37.14
12	TDI	50.97	48.18	41.85	39.73	35.69	34.25
	FDI	54.80	49.86	45.38	42.87	39.42	37.85

Note:  $L$ : window coefficient length,  $\beta$ : number of FFTs averaged.

is 9 bits or less [9]. We must note that the above-mentioned results are only approximations [9,10].

From these experiments, it is seen that in most of the cases, the difference in SCER comes down with the increase in data size ( $N$ ). This happens because, with increase in  $N$ , the number of arithmetic operations increases and the round-off errors contributed by these operations will supercede the

**TABLE 6.3**

SCER (in dB) Performance of Blackman Window

$L$	$N$ ( $\beta$ )	16 (500)	32 (275)	64 (125)	128 (50)	256 (25)	512 (10)
7	TDI	36.90	33.70	31.78	32.08	30.09	29.94
	FDI	30.54	30.40	28.92	28.68	28.50	27.86
8	TDI	43.17	38.44	35.57	35.03	32.61	31.73
	FDI	39.38	38.00	36.25	36.06	34.37	33.68
9	TDI	45.95	40.83	37.62	36.80	33.54	32.71
	FDI	46.37	44.89	41.94	40.14	37.69	36.35
10	TDI	47.87	43.79	39.92	38.95	34.04	33.40
	FDI	52.37	49.86	45.88	42.54	39.56	37.80
11	TDI	50.33	44.39	40.91	38.14	34.48	33.46
	FDI	52.37	49.86	45.88	42.54	39.56	37.80
12	TDI	52.93	45.58	40.83	38.65	34.60	33.48
	FDI	55.21	51.39	46.34	43.07	39.71	38.03

Note:  $L$ : window coefficient length,  $\beta$ : number of FFTs averaged.

**TABLE 6.4**SCER (in dB) Performance of Raised-Cosine Family ( $D = 0.0113$ )

$L$	$N$ ( $\beta$ )	16 (500)	32 (275)	64 (125)	128 (50)	256 (25)	512 (10)
7	TDI	39.40	36.33	33.05	32.23	30.81	30.15
	FDI	29.21	29.36	29.49	28.01	28.16	27.10
8	TDI	41.28	39.70	36.29	35.50	32.63	32.10
	FDI	38.03	38.47	37.35	35.37	34.38	33.02
9	TDI	43.97	39.19	43.50	37.32	34.02	33.08
	FDI	40.83	39.71	40.28	37.40	36.34	34.72
10	TDI	48.55	43.59	40.03	38.45	34.93	34.09
	FDI	48.86	47.65	44.70	41.55	39.05	37.23
11	TDI	51.00	45.24	40.86	38.75	35.28	33.97
	FDI	52.38	49.81	45.64	42.45	39.28	37.60
12	TDI	51.44	45.05	41.98	39.46	35.31	34.38
	FDI	53.81	49.57	45.21	42.63	39.26	37.71

Note:  $L$ : window coefficient length,  $\beta$ : number of FFTs averaged.

quantization errors. In the following section, we shall describe an approach that improves the efficiency of the schemes in Figures 6.1 and 6.2.

## 6.4 Canonic Signed Digit Windowing

We illustrate that the block diagrams of Section 6.2 become very efficient if we use the canonic signed digits (CSD) technique. The CSD technique code is a *ternary code* where we use 0, +1, and -1, rather than 0 and 1 in a binary window [8]. This is particularly beneficial since we can construct efficient windows by reducing the number of additions using the simple geometrical progression

$$\sum_{n=N_1}^{N_2} 2^{-n} = 2^{-N_1+1} - 2^{-N_2}, \quad N_1, N_2 \in \mathcal{Z}^+, N_2 > N_1. \quad (6.8)$$

Here,  $\mathcal{Z}^+$  is the positive integer set.

As an example, consider the equation

$$2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} = 0.248046875. \quad (6.9)$$

We note that Equation 6.9 needs six adders to compute the sum. The same result can be obtained using just one adder, as given below:

$$2^{-2} + (-2^{-9}) = 0.248046875. \quad (6.10)$$

This is possible using Equation 6.8, since the summation terms are in series. In an implementation, the complexity of an adder is the same as the complexity of a subtractor.

Certain simplifications can be done to obtain a redundant form, even if the sequence is not in series, such as

$$2^{-2} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} = 0.466796875. \quad (6.11)$$

Again, Equation 6.11 has six adders in series, except for the missing  $2^{-5}$ . This can be computed either using a set of two different summations

$$2^{-2} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} = [2^{-1} - 2^{-4}] + [2^{-5} - 2^{-9}], \quad (6.12)$$

or by adding and subtracting the missing term as follows:

$$\begin{aligned} & 2^{-2} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} \\ &= 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} - 2^{-5}. \\ &= 2^{-1} - 2^{-9} - 2^{-5}. \end{aligned}$$

Here, the number of adders have been reduced to two when compared to three in Equation 6.12. These concepts are demonstrated later.

As discussed earlier, the discrete version of a generalized data window, involving cosine terms, can be represented by ( $N = \text{even}$ ):

$$f[n] = \sum_{k=0}^K (-1)^k a_k \cos\left(\frac{2\pi nk}{N}\right), \quad n = 0, 1, \dots, (N-1), \quad (6.13)$$

where  $\{a_k\}_{k=0}^K$  represents real constants. The negative sign results from the shift in the origin of the window. We note that the function is centered around  $N/2$ . We recall that a four-term window can be represented by Equation 6.1, which is reproduced below [9] for clarity:

$$\begin{aligned} f[n] &= a - 2b \cos\left(\frac{2\pi n}{N}\right) + 2c \cos\left(\frac{4\pi n}{N}\right) - 2d \cos\left(\frac{6\pi n}{N}\right), \\ n &= 0, 1, \dots, (N-1). \end{aligned}$$

We also recall that  $A[r]$  is the DFT of a data sequence  $x[n]$ ,  $n = 0, 1, \dots, (N-1)$  (see Section 6.2 for details). If  $f[n]$  is used to smooth the spectrum  $A[r]$ , the resulting smoothed spectrum is given by Equation 6.7, which is

$$\begin{aligned} F[r] &= aA[r] - b\{A[r+1] + A[r-1]\} + c\{A[r+2] + A[r-2]\} \\ &\quad - d\{A[r+3] + A[r-3]\}, \end{aligned}$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are real constants. By expressing these constants in terms of the negative powers of two, the multiplications are carried out by right shifts. Further reduction in hardware complexity is obtained by expressing  $F[r]$  in Equation 6.7 as

$$F[r] = 2^{-\mu} F'[r], \quad (6.14)$$

where  $2^{-\mu}$  represents the common factor of all constants used in Equation 6.7 [8,9]. As will be shown, the schemes proposed in this section do not require any multiplication, except by  $2^{-n}$ , which amounts to only scaling or shifting by  $n$  bits to the right. Also, in the scheme proposed, we need not recompute the DFT again if we decide to change the window function. Furthermore, these schemes do not require storage for coefficients.

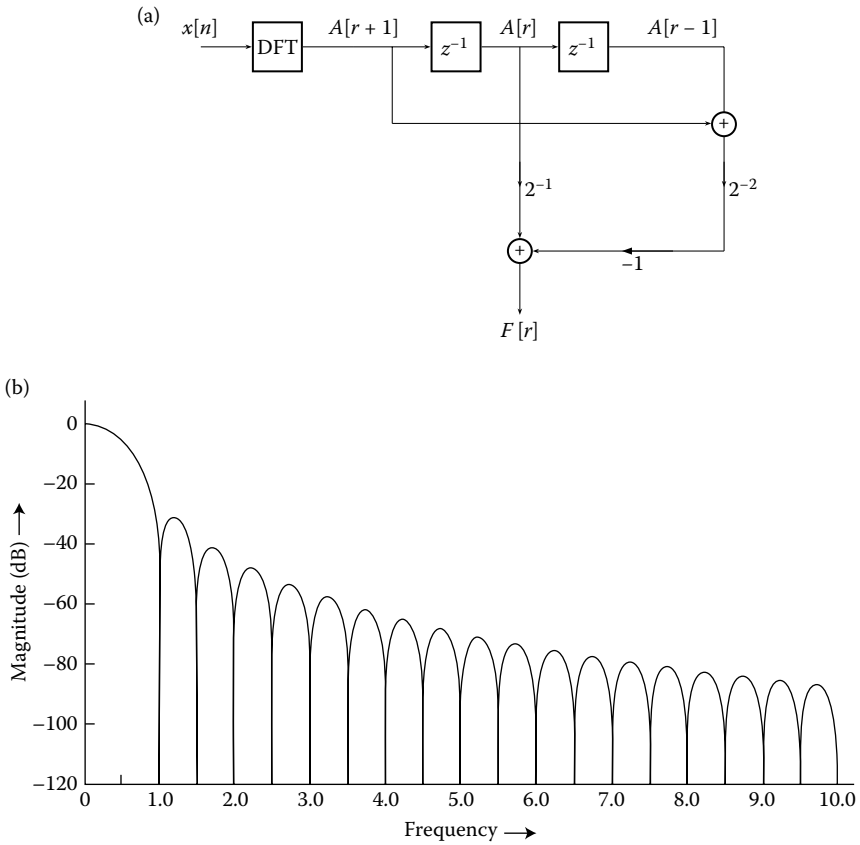
We shall now describe some of the typical CSD window implementations. These CSD windows have been derived by approximating the coefficients of the data windows in terms of the binary fractions and optimizing them with respect to their side-lobe performance. Hence, most of the CSD windows proposed perform at least equal or sometimes even better than the other known windows. The CSD window structures for a number of windows are presented in the following pages. Continuous-time frequency response plots have also been included to understand its frequency-domain behavior. In the frequency response plot, the  $y$ -axis shows the normalized magnitude of the Fourier transform (in dB), while the  $x$ -axis gives the normalized frequency.

### 6.4.1 Window 1

The first set of weights that we consider corresponds to the well-known **Hann window**, for which there are only two nonzero coefficients  $a_0 = 0.5$  and  $a_1 = 0.5$  (see Equation 6.13) [9]. Therefore, the coefficients for the FDI are  $a = 0.5$  and  $b = 0.25$  (Equation 6.4), which can be expressed in the binary form as  $2^{-1}$  and  $2^{-2}$ , respectively. The peak side-lobe level (PSLL) which is also the FSL for this window, is  $-31.47$  dB, but its asymptotic decay rate is 18 dB/octave. The structure for implementing this window in the frequency-domain is shown in Figure 6.3(a) and its frequency response is shown in Figure 6.3(b). The normalized half main-lobe width (NHMLW) of this CSD window is unity. Figure 6.3(b) shows the normalized values of frequency ( $x$ -axis) and the  $y$ -axis gives normalized values of the Fourier transform magnitude (expressed in dB).

### 6.4.2 Window 2

The second weighting function to be considered is the **Hamming window**, whose coefficients for FDI are  $a = 0.546875 = 2^{-1} + 2^{-5} + 2^{-6}$  and  $b = 0.2265625 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-7} = 2^{-2} - 2^{-5} + 2^{-7}$ , which is again a two-term CSD window. The structure for implementing this window is shown

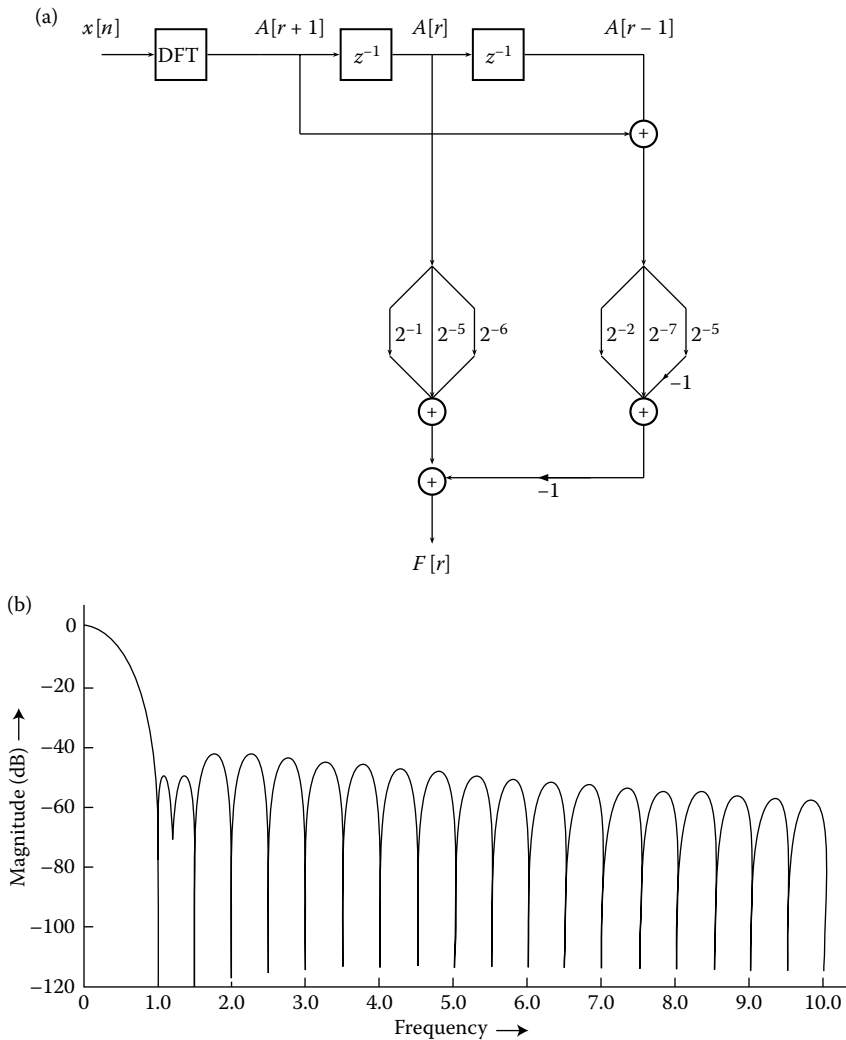


**FIGURE 6.3** Illustration of window 1. (a) Binary window structure. (b) Frequency response plot.

in Figure 6.4(a) and the frequency response of this CSD window is plotted in Figure 6.4(b). The peak side-lobe level (PSLL) and the first side-lobe level (FSL) are  $-40.84$  dB and  $-48.23$  dB, respectively. The NHMLW of this window is unity. However, its asymptotic decay rate of the side-lobe envelope is only 6 dB/octave.

### 6.4.3 Window 3

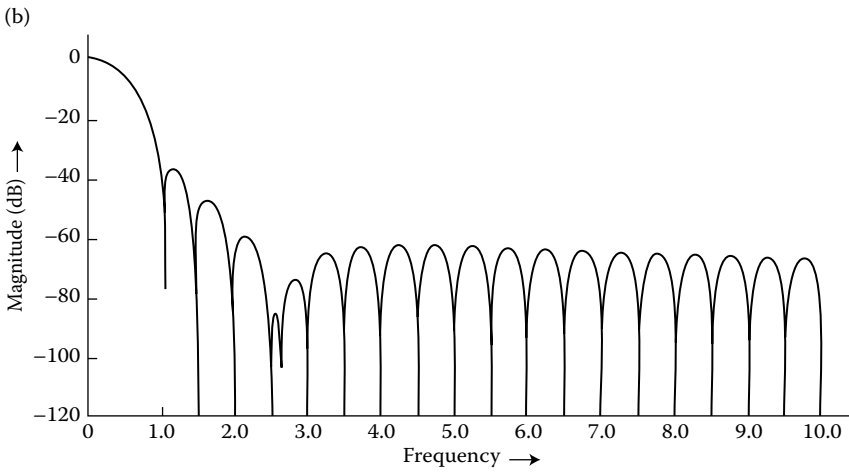
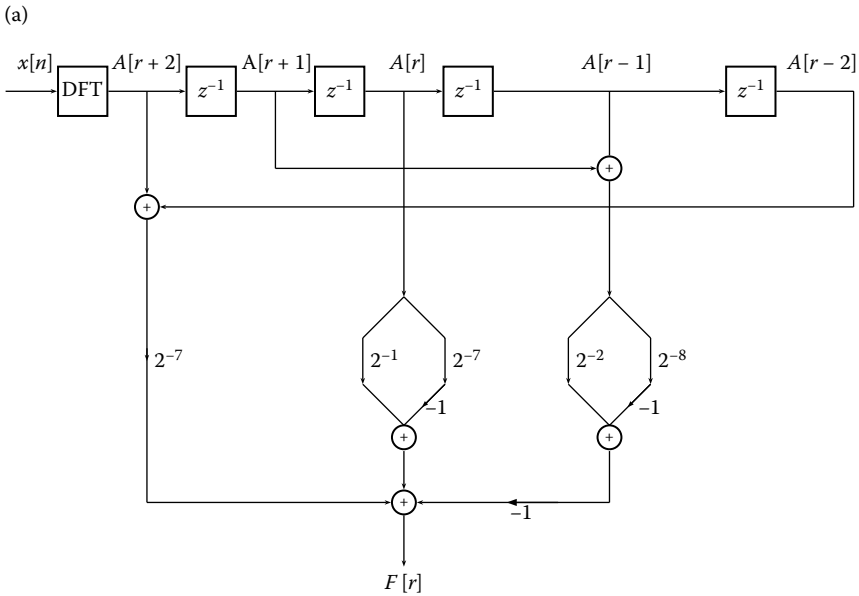
The next set of CSD windows proposed are shown in Figures 6.5(a) and 6.6(a), whose main-lobe widths fall in between the windows discussed above and the Blackman window.



**FIGURE 6.4** Illustration of window 2. (a) Binary window structure. (b) Frequency response plot.

The set of binary coefficients chosen are

$$\begin{aligned}
 a &= 0.4921875 = 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} = 2^{-1} - 2^{-7}, \\
 b &= 0.24609375 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} = 2^{-2} - 2^{-8}, \quad \text{and} \\
 c &= 0.0078125 = 2^{-7}.
 \end{aligned}
 \tag{6.15}$$

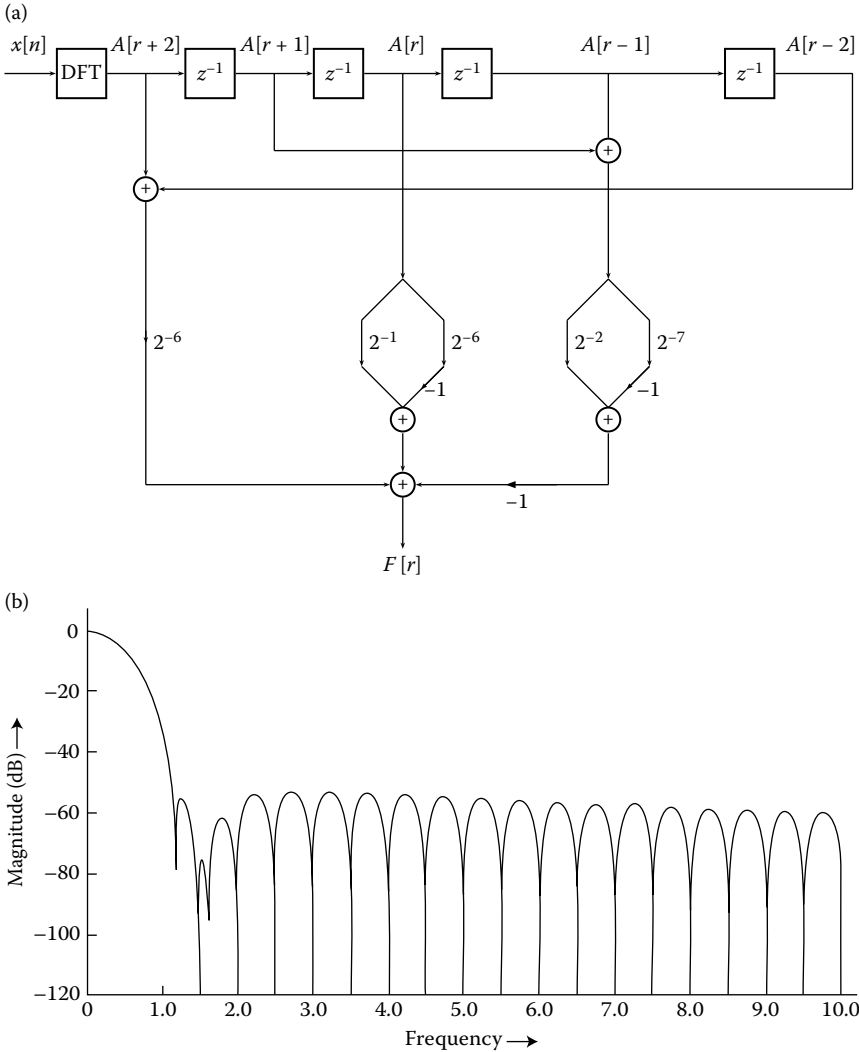


**FIGURE 6.5**

Illustration of window 3 (for  $a = 0.4921875$ ,  $b = 0.24609375$ ,  $c = 0.0078125$ ). (a) Binary window structure. (b) Frequency response plot

We also consider another set of slightly modified coefficients for the three-term CSD window as follows:

$$\begin{aligned}
 a &= 0.484375 = 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} = 2^{-1} - 2^{-6}, \\
 b &= 0.2421875 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} = 2^{-2} - 2^{-7}, \quad \text{and} \\
 c &= 0.015625 = 2^{-6}.
 \end{aligned}
 \tag{6.16}$$



**FIGURE 6.6** Illustration of Window 3 (for  $a = 0.484375$ ,  $b = 0.2421875$ ,  $c = 0.015625$ ). (a) Binary window structure. (b) Frequency response plot.

The frequency response plots of these windows, the coefficients of which are given in Equations 6.15 and 6.16, are given in Figures 6.5(b) and 6.6(b), respectively. The NHMLWs of these windows are 1.06 and 1.20, respectively, and the asymptotic decay rate of the side-lobe envelope is only 6 dB/octave in each case. For the first set of coefficients (Equation 6.15), the FSLL (which also happens to be the maximum side-lobe level) is  $-37.41$  dB. For the second



set of coefficients given by Equation 6.16, the first and maximum side lobes are  $-54.82$  and  $-52.69$  dB, respectively.

#### 6.4.4 Window 4

We now describe an equivalent of the **Blackman window**, which is another three-term CSD window. This window has an increased main-lobe width of 1.5. Its corresponding asymptotic decay rate is 30 dB/octave, which is considered to be excellent. The binary coefficients in this case are

$$a = 0.375 = 2^{-2} + 2^{-3}; \quad b = 0.25 = 2^{-2}; \quad \text{and} \quad c = 0.0625 = 2^{-4}.$$

The first (as well as the maximum side-lobe level) is only  $-46.74$  dB. The implementation scheme is shown in Figure 6.7(a) and the corresponding Fourier transform is given in Figure 6.7(b).

#### 6.4.5 Window 5

The second set of CSD windows in the category of Blackman windows with NHMLW of 1.5 has the following set of coefficients:

$$\begin{aligned} a &= 0.42578125 = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-8} = 2^{-1} - 2^{-4} - 2^{-6} + 2^{-8}, \\ b &= 0.248046875 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} = 2^{-2} - 2^{-9}, \quad \text{and} \\ c &= 0.0390625 = 2^{-5} + 2^{-7}. \end{aligned}$$

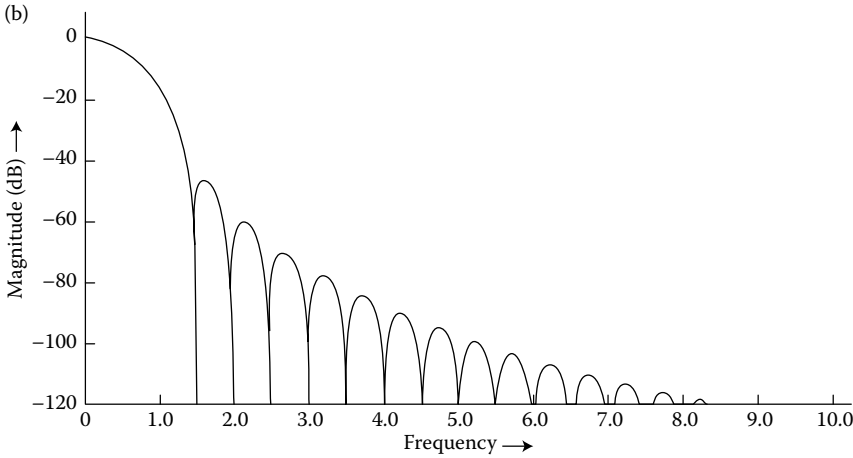
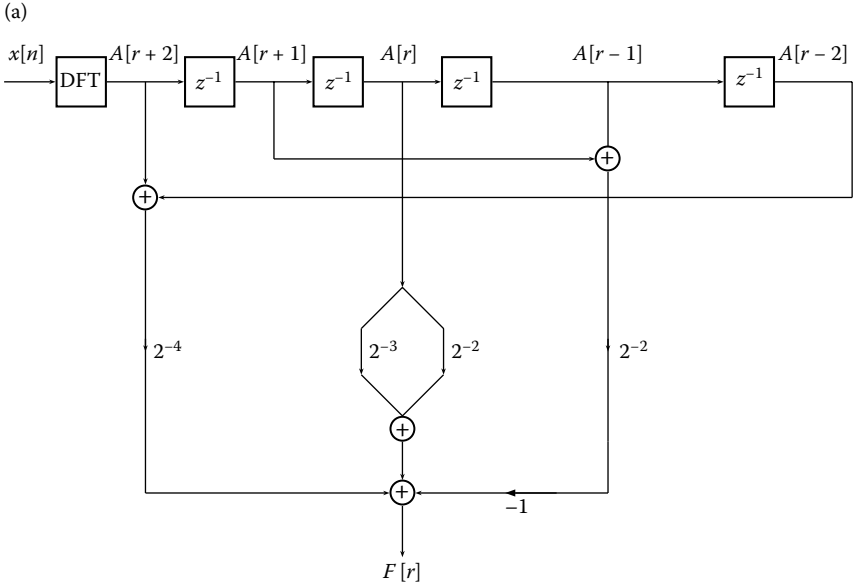
The scheme for the FDI of this window is shown in Figure 6.8(a) and its DTFT in Figure 6.8(b). This window yields an FSL of  $-64.73$  dB, which is also the same as the maximum side-lobe level, giving us a 6.5 dB improvement over the original Blackman window. However, the side-lobe decay rate of the proposed window is only 6 dB/octave.

#### 6.4.6 Window 6

This CSD window, which again falls in the category of a three-term window, has NHMLW of 1.42 and its coefficients are:

$$\begin{aligned} a &= 0.453125 = 2^{-2} + 2^{-3} + 2^{-4} + 2^{-6} = 2^{-1} - 2^{-4} + 2^{-6}, \\ b &= 0.24609375 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} = 2^{-2} - 2^{-8}, \quad \text{and} \\ c &= 0.02734375 = 2^{-6} + 2^{-7} + 2^{-8} = 2^{-5} - 2^{-8}. \end{aligned}$$

The FSL and the maximum side-lobe level are  $-67.60$  and  $-59.86$  dB, respectively. Figure 6.9(a) shows the FDI of this window and Figure 6.9(b) displays its spectrum.

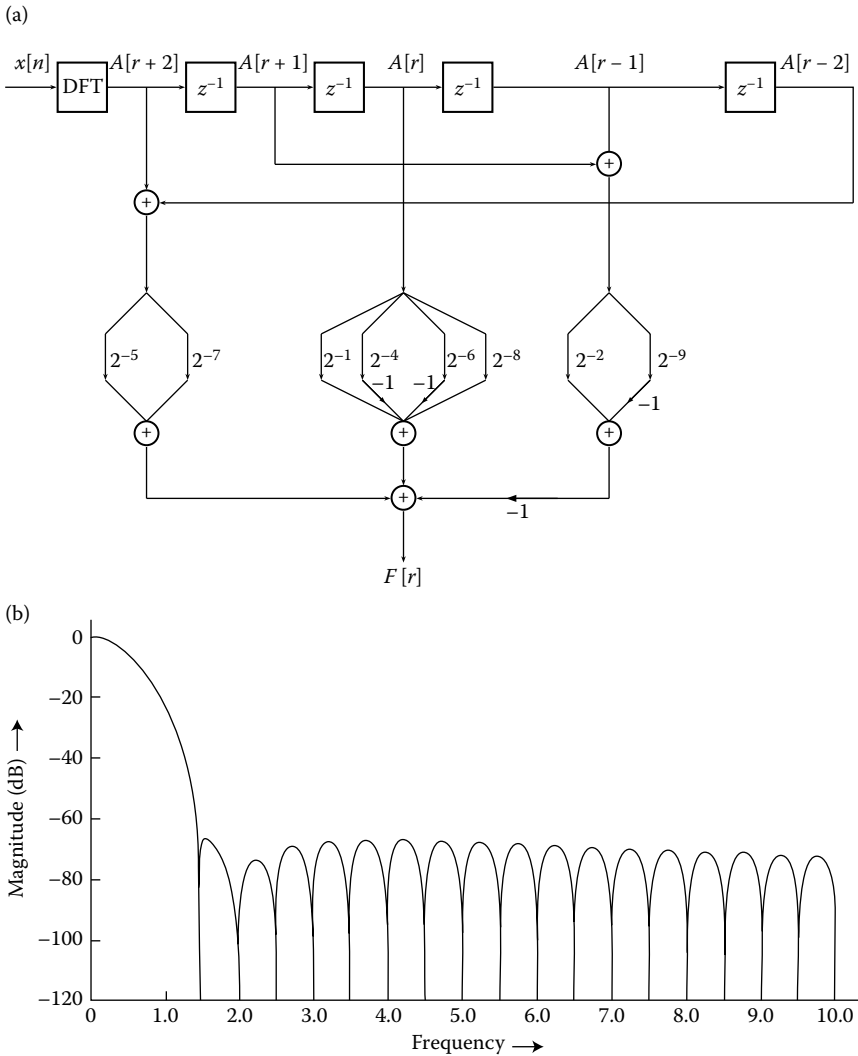


**FIGURE 6.7** Illustration of Window 4. (a) Binary window structure. (b) Frequency response plot.

**6.4.7 Window 7**

The next window function that we consider has the following set of coefficients:

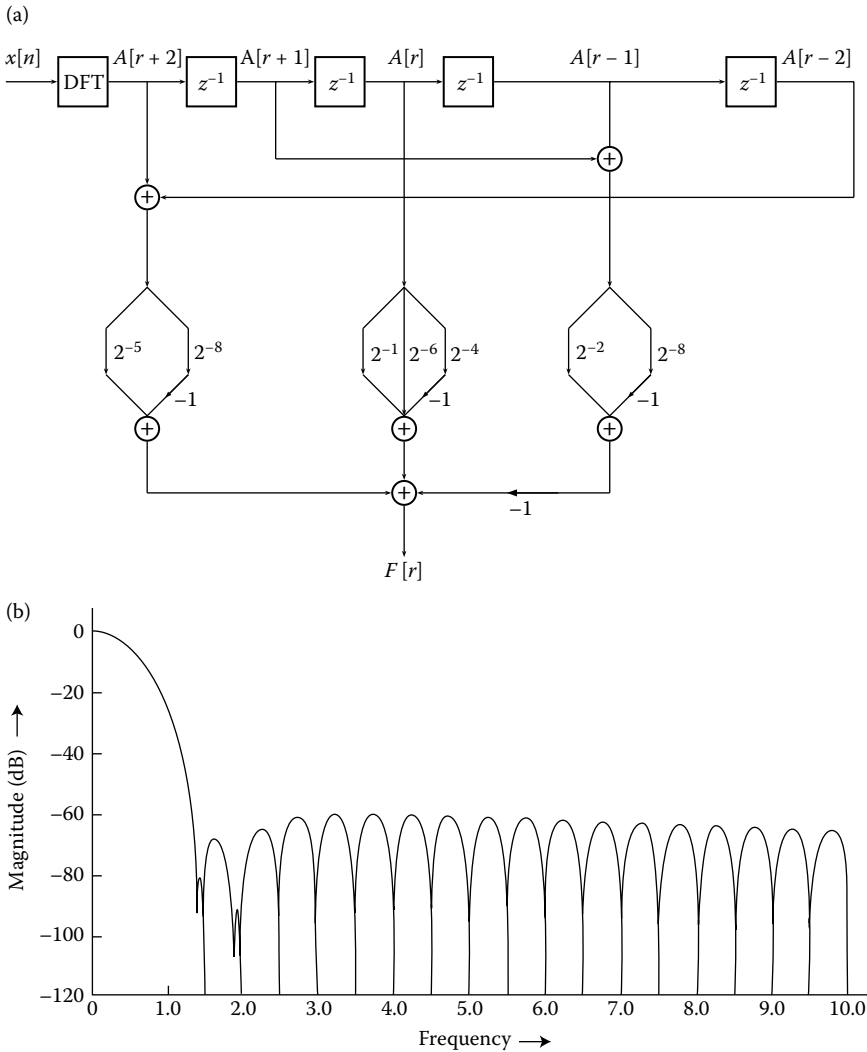
$$\begin{aligned}
 a &= 0.44921875 = 2^{-2} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} = 2^{-1} - 2^{-4} + 2^{-7} + 2^{-8}, \\
 b &= 0.24609375 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} = 2^{-2} - 2^{-8}, \quad \text{and} \\
 c &= 0.029296875 = 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} = 2^{-5} - 2^{-9}.
 \end{aligned}$$



**FIGURE 6.8** Illustration of window 5. (a) Binary window structure. (b) Frequency response plot.

This three-term CSD window offers an excellent FSL of  $-93.50$  dB, but the decay rate of the Fourier transform of the window is only 6 dB/octave. The structure of this window is shown in Figure 6.10(a) and its frequency-domain plot is given in Figure 6.10(b).

We note that for all the windows considered in the three-term category (except for window 4), the asymptotic decay rate of the side-lobe envelope



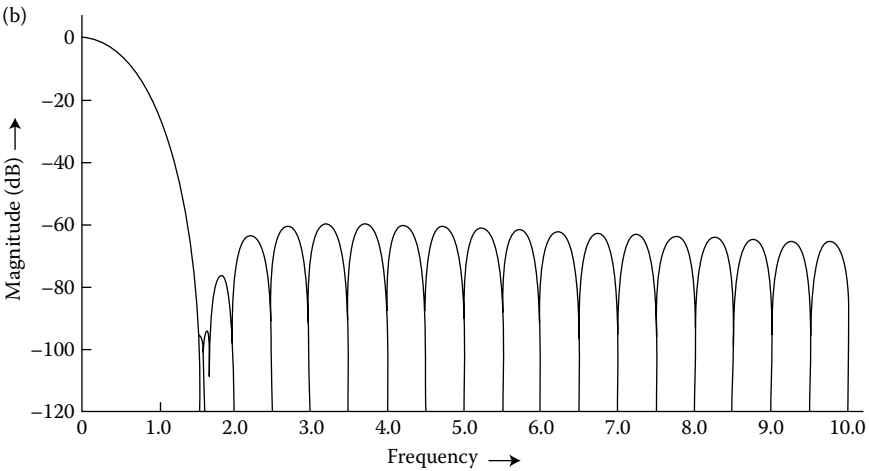
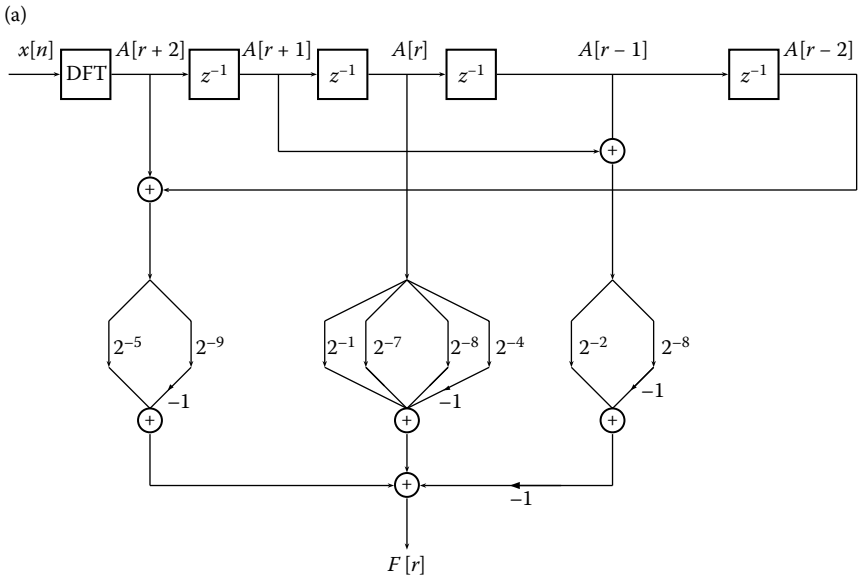
**FIGURE 6.9** Illustration of window 6. (a) Binary window structure. (b) Frequency response plot.

is only 6 dB/octave. However, we present some more windows that offer a much better side-lobe decay rate.

**6.4.8 Window 8**

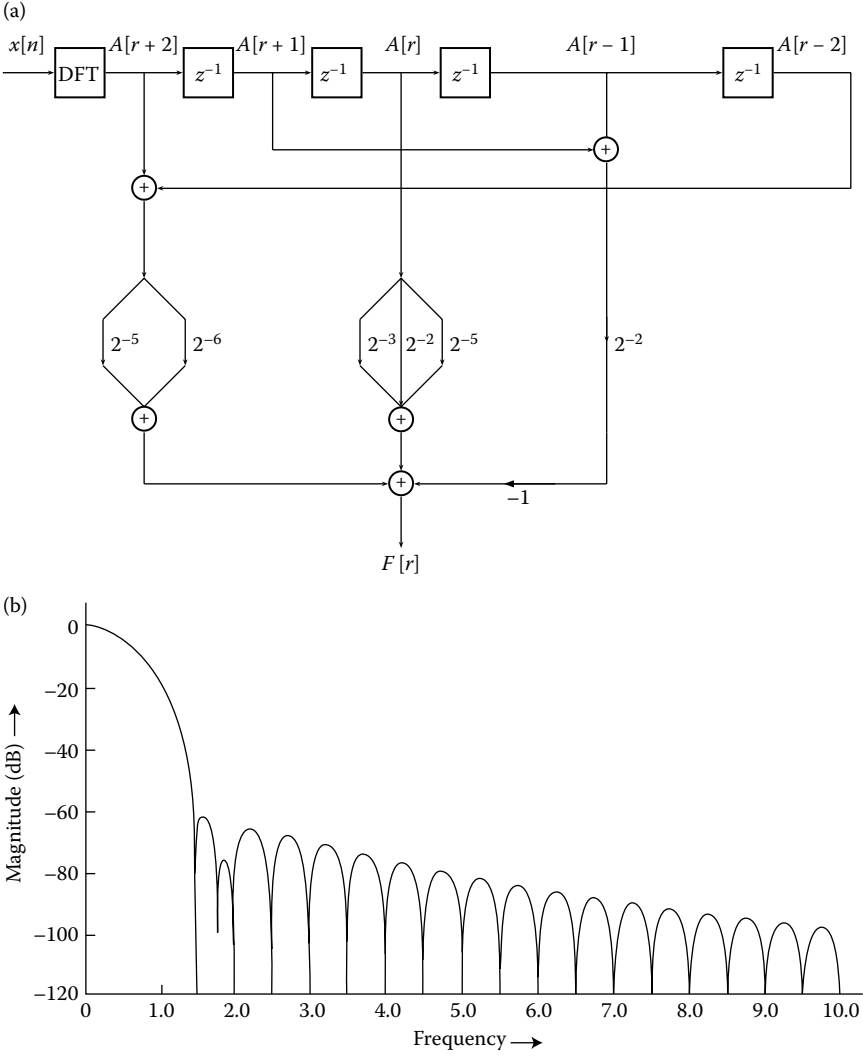
The first set of coefficients in this category of windows is

$$a = 0.40625 = 2^{-2} + 2^{-3} + 2^{-5}; b = 0.25 = 2^{-2}; \text{ and } c = 0.046875 = 2^{-5} + 2^{-6}.$$



**FIGURE 6.10** Illustration of window 7. (a) Binary window structure. (b) Frequency response plot.

The structure of this window is depicted in Figure 6.11(a), with its frequency response in Figure 6.11(b). This window has a side-lobe fall-off rate of 18 dB/octave, which is the same as the original continuous-time Blackman window (introduced in Chapter 3). However, the first (and maximum) side-lobe level in this case is  $-61.30$  dB, which is in fact better than the original Blackman window by about 3 dB.



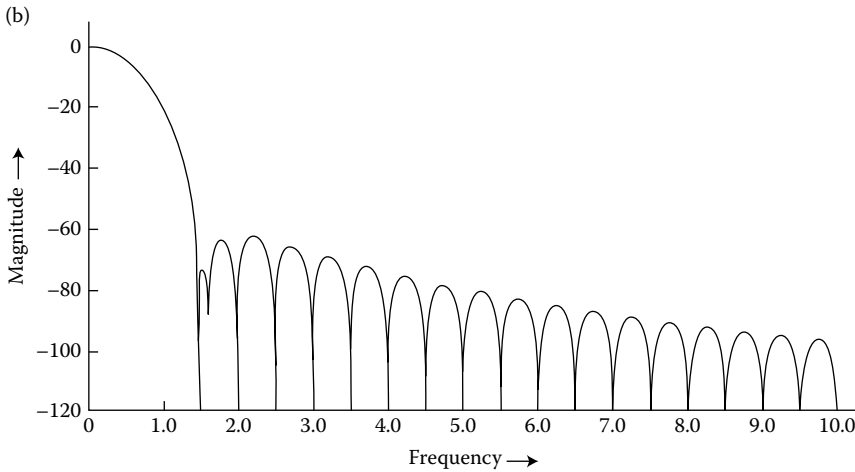
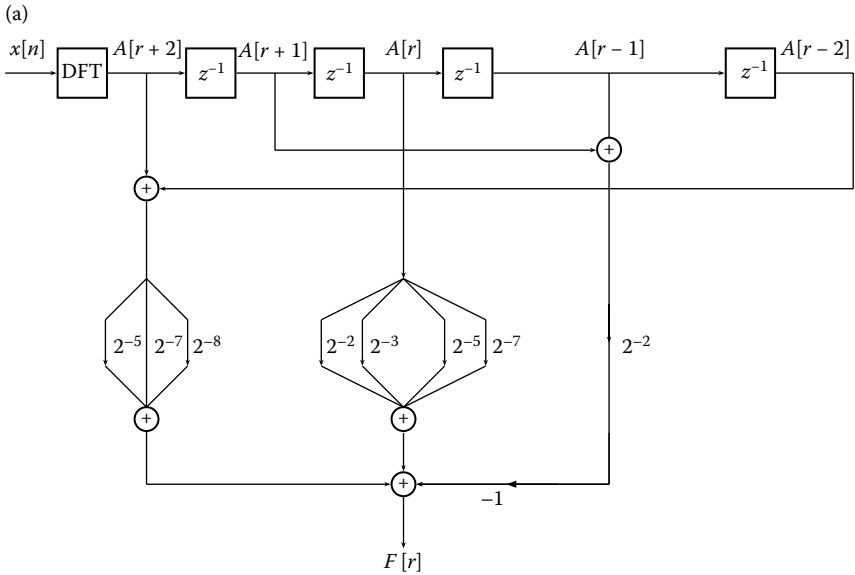
**FIGURE 6.11** Illustration of window 8. (a) Binary window structure. (b) Frequency response plot.

**6.4.9 Window 9**

In the next three-term CSD window, the coefficients are set as follows:

$$a = 0.4140625 = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7}, \quad b = 0.25 = 2^{-2},$$

$$c = 0.04296875 = 2^{-5} + 2^{-7} + 2^{-8}.$$



**FIGURE 6.12** Illustration of window 9. (a) Binary window structure. (b) Frequency response plot.

The structure is shown in Figure 6.12(a) and its response in Figure 6.12(b). This window offers FSL of  $-72.22$  dB, while its peak side-lobe level is  $-61.65$  dB. The asymptotic decay rate of this window is  $18$  dB/octave. Therefore, this window offers excellent *immediate side-lobe rejection* as well as *far-off side-lobe rejection*. The performance of this window is much better than the continuous-time Blackman window (presented in an earlier chapter).

#### 6.4.10 Window 10

The last set of binary coefficients to be considered in the three-term window category is the one with the following constants:

$$a = 0.41015625 = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-8}, \quad b = 0.25 = 2^{-2}, \quad \text{and}$$

$$c = 0.044921875 = 2^{-5} + 2^{-7} + 2^{-8} + 2^{-9} = 2^{-6} - 2^{-9} + 2^{-5}.$$

Its FSLL is  $-65.68$  dB, while the PSL is  $-63.52$  dB. The FDI is shown in Figure 6.13(a) and its frequency response in Figure 6.13(b). However, the side-lobes falloff rate is 18 dB/octave like in the previous two cases.

The third category of CSD windows is the four-term windows, which have excellent side-lobe behavior, but at the expense of increased main-lobe widths. We have obtained four sets of four-term CSD windows that are considered to be optimum with respect to the side-lobe levels and their asymptotic decay rates. These sets of windows are considered next.

#### 6.4.11 Window 11

The first set of binary coefficients are

$$a = 0.3125 = 2^{-2} + 2^{-4}; \quad b = 0.234375 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} = 2^{-2} - 2^{-6},$$

$$c = 0.09375 = 2^{-4} + 2^{-5}, \quad \text{and} \quad d = 0.015625 = 2^{-6}.$$

This window has an asymptotic decay rate of 42 dB/octave. The frequency response plot depicted in Figure 6.14(b) shows its maximum side lobe to be  $-60.96$  dB and its FDI structure is given in Figure 6.14(a).

#### 6.4.12 Window 12

The optimized binary coefficients in this four-term window are

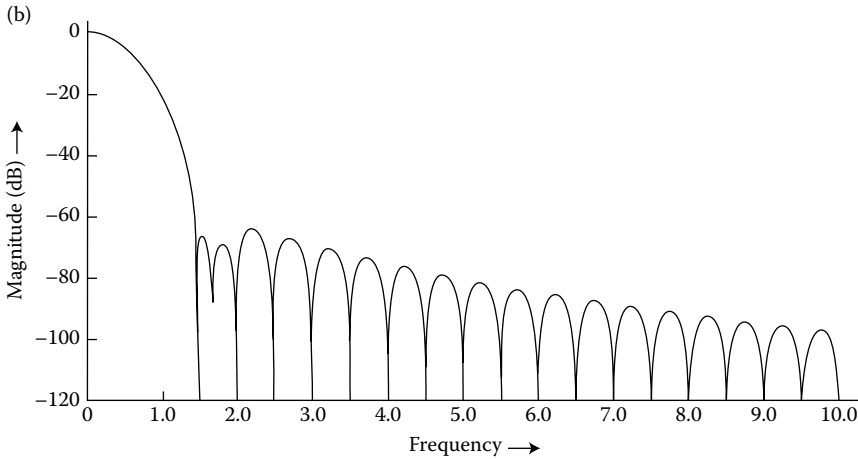
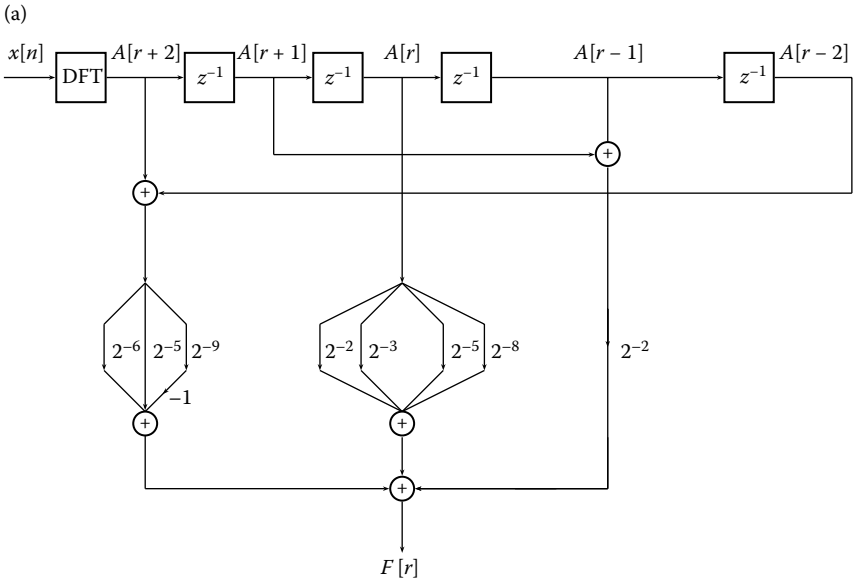
$$a = 0.3515625 = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-7};$$

$$b = 0.2421875 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} = 2^{-2} - 2^{-7}$$

$$c = 0.07421875 = 2^{-4} + 2^{-7} + 2^{-8}, \quad \text{and} \quad d = 0.0078125 = 2^{-7}.$$

In this case, the FSLL and the maximum side-lobe levels are the same, that is,  $-71.63$  dB, but its asymptotic decay rate is 18 dB/octave. This CSD window is depicted in Figure 6.15(a) and its frequency response plot in Figure 6.15(b).



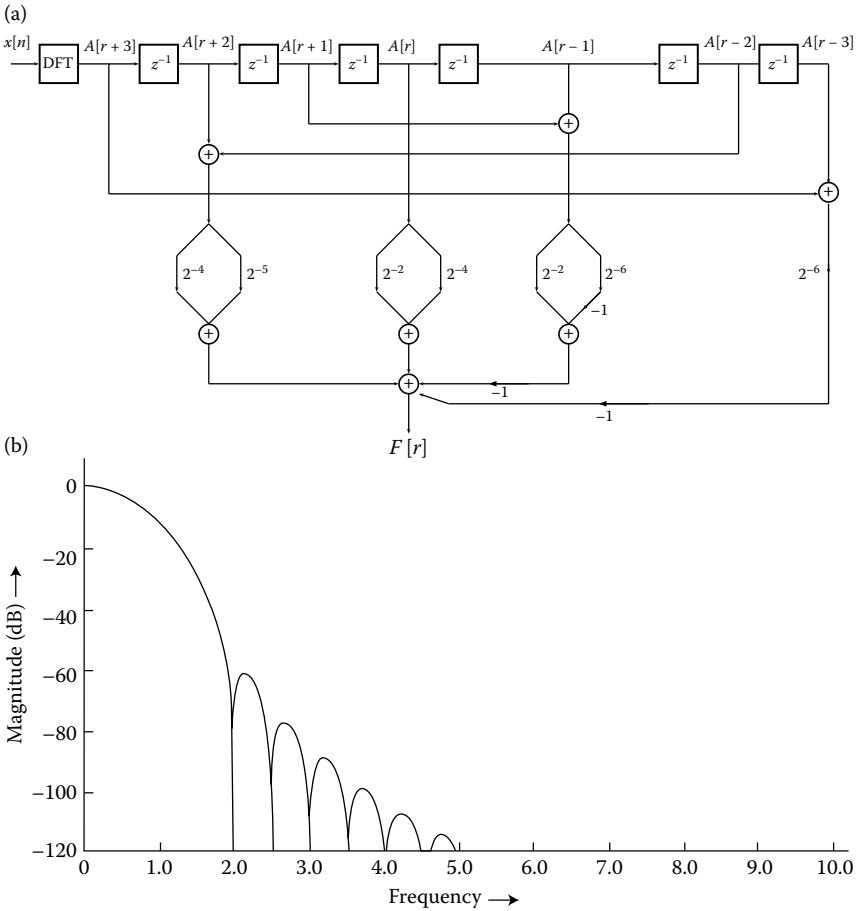


**FIGURE 6.13** Illustration of window 10. (a) Binary window structure. (b) Frequency response plot.

**6.4.13 Window 13**

This CSD window has four coefficients that are given by

$$\begin{aligned}
 a &= 0.35546875 = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-8} = 2^{-2} + 2^{-3} - 2^{-6} - 2^{-8}, \\
 b &= 0.244140625 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-9} = 2^{-2} - 2^{-7} + 2^{-9}, \\
 c &= 0.072265625 = 2^{-4} + 2^{-7} + 2^{-9}, \text{ and } d = 0.005859375 = 2^{-8} + 2^{-9}.
 \end{aligned}$$



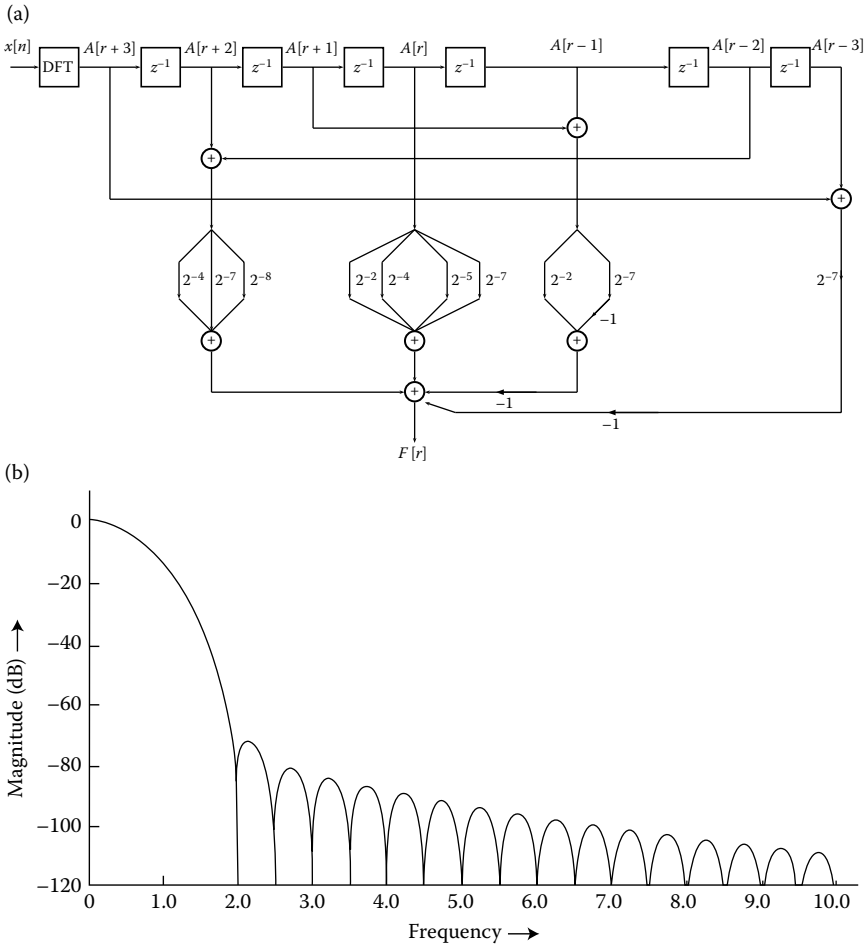
**FIGURE 6.14** Illustration of window 11. (a) Binary window structure. (b) Frequency response plot.

This results in a frequency response as indicated in Figure 6.16(b). The FSLI of this window is  $-75.42$  dB. However, this window has a wide NHMLW of 1.92, while the asymptotic decay rate is still 18 dB/octave. The discrete-time structure of this window is given in Figure 6.16(a).

**6.4.14 Window 14**

The last CSD window we discuss in this chapter has the set of optimized binary coefficients as follows:

$$a = 0.359375 = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-6} = 2^{-2} + 2^{-3} - 2^{-6},$$

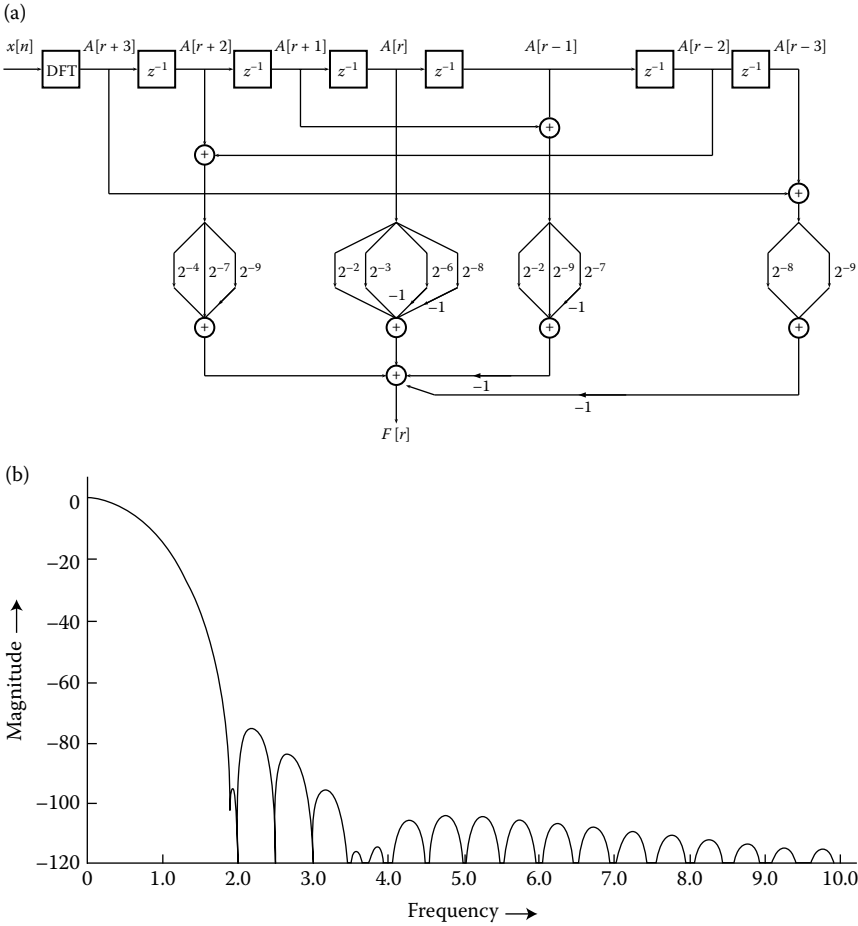


**FIGURE 6.15** Illustration of window 12. (a) Binary window structure. (b) Frequency response plot.

$$b = 0.244140625 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-9} = 2^{-2} - 2^{-7} + 2^{-9},$$

$$c = 0.0703125 = 2^{-4} + 2^{-7}, \text{ and } d = 0.005859375 = 2^{-8} + 2^{-9}.$$

From Figure 6.17(b), it is seen that this window gives excellent side-lobe attenuation, which is nearly  $-88$  dB. The structure for implementing this window is shown in Figure 6.17(a). Its asymptotic decay rate of the side-lobe envelope is still 18 dB/octave. However, the NHMLW of this window is 2.0, which is rather large.



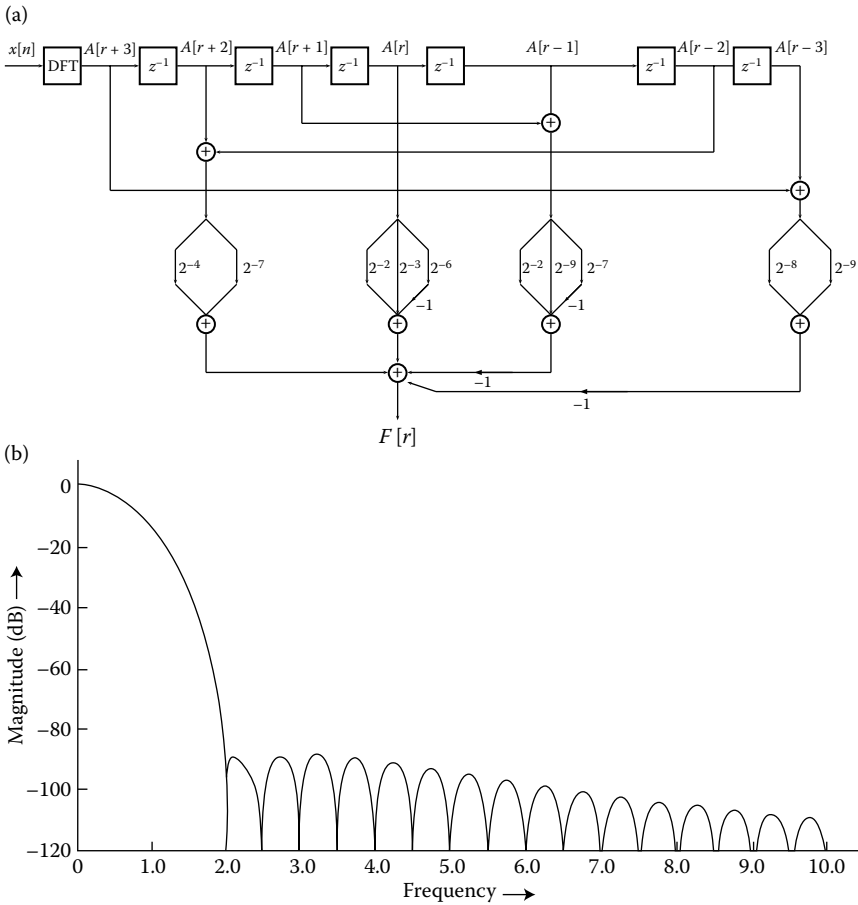
**FIGURE 6.16** Illustration of window 13. (a) Binary window structure. (b) Frequency response plot.

### 6.5 Modified Zeroth-Order Kaiser-Bessel Window Family

As discussed in an earlier chapter, the near-optimum modified zeroth-order Bessel window invented by Kaiser has a continuous-time Fourier transform pair given by

$$f(t) = \frac{I_0[\alpha\sqrt{1 - (t/\tau)^2}]}{I_0(\alpha)}, \quad |t| \leq \tau \quad \text{and}$$

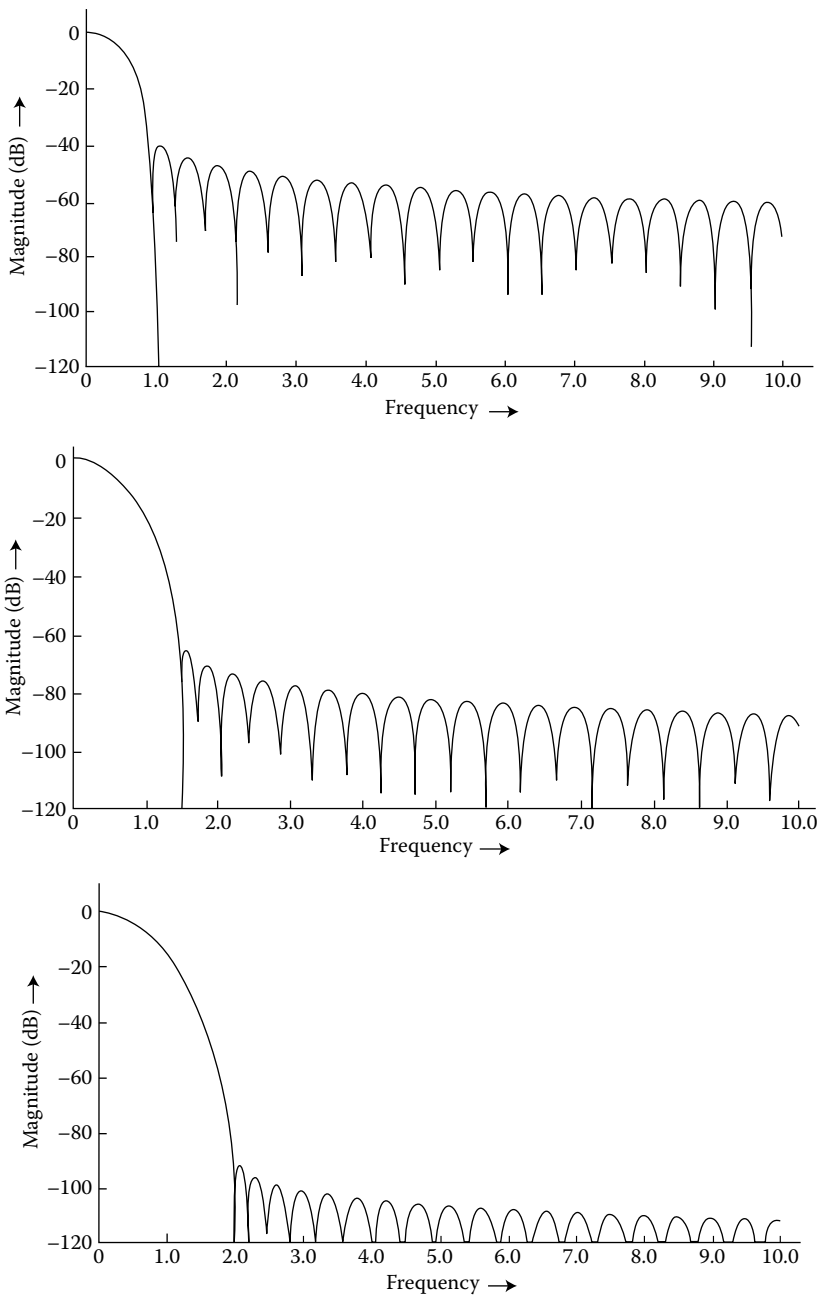
$$F(j\Omega) = \frac{\sinh[\alpha\sqrt{1 - (\Omega/\alpha)^2}]}{(\sinh \alpha)\sqrt{1 - (\Omega/\alpha)^2}}, \quad -\infty \leq \Omega \leq \infty.$$



**FIGURE 6.17** Illustration of Window 14. (a) Binary window structure. (b) Frequency response plot.

where  $\alpha$  is the parameter controlling the main-lobe width and consequently the side-lobe level. We have chosen the values of  $\alpha$  to be  $\pi\sqrt{3}$ ,  $\pi\sqrt{8}$ , and  $\pi\sqrt{15}$ , such that the normalized half main-lobe widths are 1.0, 1.5, and 2.0, respectively.

The normalized magnitude plots of the Fourier transform of these windows are shown in Figure 6.18. The asymptotic decay rate of the side-lobe envelope for the modified zeroth-order Kaiser–Bessel family is only 6 dB/octave. The performance of each CSD window proposed can be easily compared with respect to the near-optimum Kaiser–Bessel windows having nearly the same main-lobe widths. It should, however, be noted that the sum-cosine windows discussed in Section 5.2.7 and presented in refs. [9,11] cannot be implemented using the CSD structure.

**FIGURE 6.18**

Frequency response of zeroth-order Kaiser-Bessel window family for different values of  $\alpha$ . (a) For  $\alpha = \pi\sqrt{3}$ . (b) For  $\alpha = \pi\sqrt{8}$ . (c) For  $\alpha = \pi\sqrt{15}$ .

### 6.6 Summary

In this chapter, we have presented a number of windows with excellent side-lobe behavior for implementation in the frequency-domain. Their PSLL ranges from  $-31.47$  dB to  $-87.69$  dB, while the asymptotic decay rate of the side-lobe envelope varies from 6 dB/octave to 42 dB/octave. The variation in the FSLL is from  $-31.47$  dB to  $-93.50$  dB. Table 6.5 presents all the parameters of the CSD windows, considered in Section 6.4 as well as the Kaiser–Bessel windows of Section 6.5. The distinct advantage of using the proposed CSD windows is the ease with which they can be implemented in the frequency-domain. Conventional time-domain windowing requires  $(N/2 + 1)$  stored values of the window samples for even  $N$  and  $(N + 1)/2$  for odd  $N$ , with  $N$  multiplications. In the scheme suggested, these are replaced by shifts and add operations. Yet another advantage of frequency-domain windowing is that more than one window may be applied to the same spectrum without the need of computing the DFT more than once. To choose appropriate windows

**TABLE 6.5**  
Performance of Binary Windows

Window	Normalized Half	First	Peak	Asymptotic Decay
	Main-Lobe Width	Side-Lobe Level (dB)	Side-Lobe Level (dB)	Rate of SLL (dB/Octave)
Window 1	1.0	-31.47	-31.47	18
Window 2	1.0	-48.23	-40.84	6
Window 3	1.06	-37.41	-37.41	6
	1.20	-54.82	-52.69	6
Window 4	1.50	-46.74	-46.74	30
Window 5	1.50	-64.73	-64.73	6
Window 6	1.42	-67.60	-59.86	6
Window 7	1.50	-93.50	-59.42	6
Window 8	1.50	-61.30	-61.30	18
Window 9	1.50	-72.22	-61.65	18
Window 10	1.50	-65.68	-63.52	18
Window 11	2.0	-60.96	-60.96	42
Window 12	2.0	-71.63	-71.63	18
Window 13	1.92	-75.42	-75.42	18
Window 14	2.0	-88.42	-87.69	18
<i>Kaiser–Bessel with <math>\alpha</math></i>				
$\pi\sqrt{3}$	1.0	-39.79	-39.79	6
$\pi\sqrt{8}$	1.5	-65.47	-65.47	6
$\pi\sqrt{15}$	2.0	-91.25	-91.25	6

depending on the application of interest, efficient and economical structures are suggested for frequency-domain windowing. To illustrate the frequency-domain behavior, we have also included the plots of the continuous-time Fourier transforms of the windows.

---

## References

1. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectra*, Dover Publications, New York, NY, 1958.
2. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proceedings*, vol. 66, no. 1, pp. 51–83, January 1978.
3. A.H. Nuttall, Some windows with very good side-lobe behavior, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, no. 1, February 1981.
4. J.F. Kaiser, Digital filters, In *System Analysis by Digital Computer*, Chapter 7, F.F. Kuo and J.F. Kaiser, Eds., Wiley, New York, NY, 1966.
5. Y.C. Lim and S.R. Parker, FIR filter design over a discrete powers-of-two coefficient space, *IEEE Transactions Acoustics, Speech and Signal Processing*, vol. ASSP-31, pp. 583–591, June 1983.
6. K.M.M. Prabhu, V.U. Reddy, and J.P. Agrawal, Synthesis of windows from a raised-cosine pulse, *Electronics Letters*, vol. 11, pp. 161–163, April 1975.
7. K.M.M. Prabhu, Optimum binary windows for discrete fourier transform, Conference Record, IEEE International Conference on ASSP, Tampa, Florida, pp. 756–759, March 1985.
8. K.M.M. Prabhu and H. Renganathan, Optimum binary windows for discrete fourier transform, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp. 216–220, February 1986.
9. K.M.M. Prabhu, *Data Windows in Digital Signal Processing*, PhD Thesis, IIT Madras, India, October 1980.
10. K.M.M. Prabhu and J.P. Agrawal, Time and frequency-domain implementations of data windows, *Archiv Für Elektronik und Übertragungstechnik*, vol. 32, pp. 224–226, May 1979.
11. V.U. Reddy and K.M.M. Prabhu, *Sum-cosine window*, *Electronics Letters*, vol. 10, pp. 438–439, October 1974.





# 7

---

## *FIR Filter Design Using Windows*

---

### 7.1 Ideal Filters

Filtering refers to the time or frequency-domain processing of a signal, which is performed to enhance the required features or to remove unwanted frequency components. Ideal filters allow a band of frequencies and reject all other frequencies. Depending on their frequency-domain characteristics, filters can be classified as

- i. Lowpass
- ii. Highpass
- iii. Bandstop
- iv. Bandpass

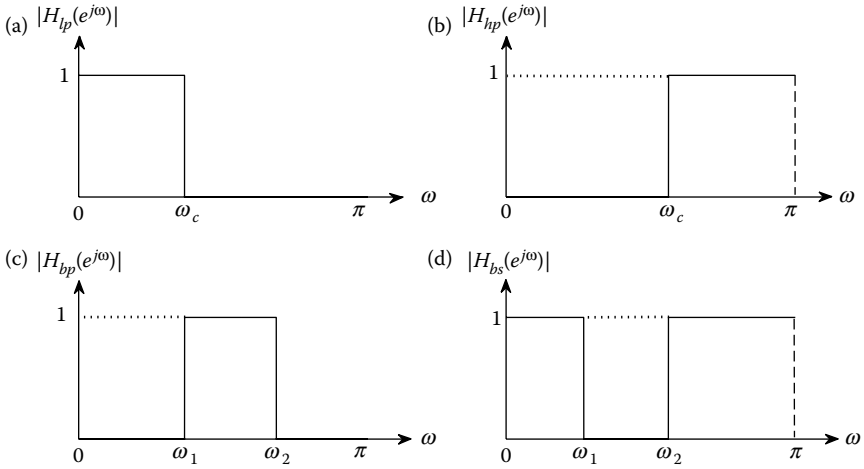
The magnitude responses of these ideal filters are shown in Figure 7.1 (for positive frequencies only). For real filters, the magnitude response is symmetric around  $\omega = 0$ . In this section, we define the magnitude response of these ideal filters and compute their impulse responses, assuming that the phase responses are equal to zero (zero-phase filters).

#### 7.1.1 Lowpass Filter

The frequency response of a zero-phase ideal lowpass filter (LPF) can be defined as

$$H_{lp}(e^{j\omega}) = \begin{cases} 1, & 0 \leq |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi, \end{cases} \quad (7.1)$$

where  $\omega_c$  is the cut-off frequency of this LPF. It allows only low-frequency components that are less than the cut-off frequency  $\omega_c$  to pass and to fully reject all the frequencies above  $\omega_c$ . The impulse response of the ideal LPF



**FIGURE 7.1** Magnitude response of ideal (a) lowpass, (b) highpass, (c) bandpass, and (d) bandstop filters.

$h_{lp}[n]$  can be obtained by taking the inverse DTFT of  $H_{lp}(e^{j\omega})$  as given below:

$$\begin{aligned}
 h_{lp}[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{lp}(e^{j\omega}) e^{j\omega n} d\omega \\
 &= \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{1}{2\pi} \left[ \frac{e^{j\omega n}}{jn} \right]_{-\omega_c}^{\omega_c} \\
 &= \frac{1}{2\pi} \left( \frac{e^{j\omega_c n}}{jn} - \frac{e^{-j\omega_c n}}{jn} \right) \\
 &= \begin{cases} \frac{\sin(\omega_c n)}{\pi n}, & -\infty < n < \infty, n \neq 0 \\ \frac{\omega_c}{\pi}, & n = 0. \end{cases} \tag{7.2}
 \end{aligned}$$

### 7.1.2 Highpass Filter

The frequency response of the ideal highpass filter (HPF) can be defined as

$$H_{hp}(e^{j\omega}) = \begin{cases} 0, & 0 \leq |\omega| \leq \omega_c \\ 1, & \omega_c < |\omega| \leq \pi. \end{cases} \tag{7.3}$$

The HPF rejects the frequencies less than the cut-off frequency  $\omega_c$  and allows all the frequencies between  $\omega_c$  and  $\pi$  to pass. From Figure 7.1, we can easily

relate the attenuation characteristics of the zero-phase highpass and the zero-phase LPFs as

$$H_{hp}(e^{j\omega}) = 1 - H_{lp}(e^{j\omega}). \quad (7.4)$$

From this relation, we can now obtain the impulse response of the HPF by taking the inverse DTFT on both sides of Equation 7.4.

$$h_{hp}[n] = \delta[n] - h_{lp}[n] = \begin{cases} -\frac{\sin(\omega_c n)}{(\pi n)}, & -\infty < n < \infty, n \neq 0. \\ 1 - \frac{\omega_c}{\pi}, & n = 0. \end{cases} \quad (7.5)$$

### 7.1.3 Bandpass Filter

The bandpass filter (BPF) passes frequencies only in the interval  $(\omega_1, \omega_2)$ .

$$H_{bp}(e^{j\omega}) = \begin{cases} 1, & \omega_1 \leq \omega \leq \omega_2 \\ 0, & \text{elsewhere.} \end{cases} \quad (7.6)$$

The impulse response of a BPF can be described as the difference between the impulse responses of an LPF with a cut-off frequency  $\omega_2$  and a second lowpass with a cut-off  $\omega_1$ . The corresponding impulse response can thus be given by

$$h_{bp}[n] = \begin{cases} \frac{\sin(\omega_2 n)}{\pi n} - \frac{\sin(\omega_1 n)}{\pi n}, & -\infty < n < \infty, n \neq 0 \\ \frac{\omega_2 - \omega_1}{\pi}, & n = 0. \end{cases} \quad (7.7)$$

### 7.1.4 Bandstop Filter

In this case, the filter blocks (or stops) the frequency components in the interval  $(\omega_1, \omega_2)$ .

$$H_{bs}(e^{j\omega}) = \begin{cases} 0, & \omega_1 \leq \omega \leq \omega_2 \\ 1, & \text{elsewhere.} \end{cases} \quad (7.8)$$

The impulse response of a bandstop filter (BSF) can be described as the difference between the impulse responses of an all-pass filter and the BPF in Figure 7.1(c). Using Equation 7.7, we obtain

$$h_{bs}[n] = \delta[n] - h_{bp}[n] = \begin{cases} \frac{\sin(\omega_1 n)}{\pi n} - \frac{\sin(\omega_2 n)}{\pi n}, & -\infty < n < \infty, n \neq 0 \\ 1 - \frac{\omega_2 - \omega_1}{\pi}, & n = 0. \end{cases} \quad (7.9)$$

## 7.2 Linear Time Invariant Systems

To define FIR and IIR systems, let us first introduce linear constant coefficient difference equations (LCCDEs) in the context of discrete-time systems. The LCCDE describes an important subclass of discrete-time systems called linear time invariant (LTI) systems, or also called as linear shift invariant (LSI) systems, whose input  $x[n]$  and output  $y[n]$  (see Figure 7.2) satisfy an  $N$ th-order LCCDE defined in the following form:

$$\sum_{k=0}^N a_k y[n - k] = \sum_{r=0}^M b_r x[n - r]. \quad (7.10)$$

We assume real LTI systems for which the coefficients of the LCCDE are real. Since the system discussed before is causal, we can rearrange Equation 7.10 in the following way, such that we can compute the present sample  $y[n]$  in terms of (i) the past output samples  $y[n - k]$  and (ii) the present and the past input samples  $x[n]$  and  $x[n - k]$ , respectively. This formulation gives us the following form:

$$a_0 y[n] + \sum_{k=1}^N a_k y[n - k] = \sum_{r=0}^M b_r x[n - r]. \quad (7.11)$$

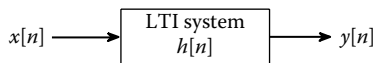
We can modify Equation 7.11 by normalizing the coefficient of  $y[n]$  to be unity (i.e.,  $a_0 = 1$ ). Also, the coefficients of the delayed output samples,  $y[n - k]$  are modified into negative coefficients  $-a_k$  so that after they are moved to right side of Equation 7.11, they become positive. The rearranged LCCDE is of the form

$$y[n] - \sum_{k=1}^N a_k y[n - k] = \sum_{r=0}^M b_r x[n - r]. \quad (7.12)$$

Therefore

$$y[n] = \sum_{k=1}^N a_k y[n - k] + \sum_{r=0}^M b_r x[n - r]. \quad (7.13)$$

The system function or the transfer function  $H(z)$ , which turns out to be the  $z$ -transform of  $h[n]$ , can be computed by taking the  $z$ -transform of the LCCDE;



**FIGURE 7.2**  
An LTI system.

from Equation 7.12, we obtain

$$Y(z) - \sum_{k=1}^N a_k z^{-k} Y(z) = \sum_{r=0}^M b_r z^{-r} X(z). \tag{7.14}$$

Rearranging and using the definition of the system function

$$H(z) = \frac{Y(z)}{X(z)}, \tag{7.15}$$

we then obtain

$$Y(z) \left[ 1 - \sum_{k=1}^N a_k z^{-k} \right] = X(z) \sum_{r=0}^M b_r z^{-r} \tag{7.16}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^M b_r z^{-r}}{1 - \sum_{k=1}^N a_k z^{-k}}, \tag{7.17}$$

or equivalently, we can also factor the numerator and denominator polynomials of  $H(z)$  as [1]

$$H(z) = b_0 \frac{\prod_{r=1}^{M_1} (1 - c_r z^{-1}) \prod_{r=1}^{M_2} (1 - f_r z^{-1})(1 - f_r^* z^{-1})}{\prod_{k=1}^{N_1} (1 - d_k z^{-1}) \prod_{k=1}^{N_2} (1 - g_k z^{-1})(1 - g_k^* z^{-1})}, \tag{7.18}$$

where  $M = M_1 + 2M_2$  and  $N = N_1 + 2N_2$ . In this expression,  $c_r$  and  $d_k$  are real zeros and real poles, respectively. The zeros  $f_r, f_r^*$  are complex conjugate pairs of zeros and  $g_k, g_k^*$  are complex conjugate pairs of poles. If we observe the right-hand side of Equation 7.13, the first sum corresponds to feedback terms and the second sum corresponds to feed-forward terms. Digital filters without feedback (i.e., all coefficients  $a_k$  are equal to zero) are called FIR filters. Equivalently, FIR filters can also be defined as digital filters having zeros and no poles except at the origin  $z = 0$ . If the input of such a filter is an isolated impulse, the output is nonzero only for  $(M + 1)$  samples, where  $M$  represents the order of the filter. In a similar way, digital filters with feedback terms also included are called IIR digital filters. The reason is that the nonzero outputs act as forcing terms when fed back to the input, thereby generating the possibility of infinite ringing.

Digital filters are characterized by their coefficients  $a_k$  and  $b_r$ . However, the rate at which the samples are given as inputs (i.e., sampling rate) is not vital. Only the time number  $n$  of the incoming samples of the signal is important. Therefore, digital filter coefficients are a function of only normalized frequencies (normalized with respect to the sampling frequency). This unique property of digital filters make them suitable for applications such as *zooming* or *multirate filtering* (i.e., changing the sampling rate at will), where the same digital filters are used, but at different sampling rates.

---

### 7.3 FIR Filters

The general equation relating the input and output of an LTI system is again reproduced below:

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{r=0}^M b_r x[n-r]. \quad (7.19)$$

If we set all  $a_k$  coefficients to zero, then Equation 7.19 reduces to

$$y[n] = \sum_{r=0}^M b_r x[n-r]. \quad (7.20)$$

Now, if we compare Equation 7.20 with the convolution sum expression for an LTI system, (reproduced from Chapter 1) which is given by

$$y[n] = \sum_{r=0}^M x[r]h[n-r] = \sum_{r=0}^M h[r]x[n-r] \quad (7.21)$$

$$= h[n] * x[n], \quad (7.22)$$

then Equation 7.20 is in the form of a convolution sum. By setting  $x[n] = \delta[n]$  in Equation 7.20, we see that the impulse response is

$$h[n] = \sum_{r=0}^M b_r \delta(n-r) \quad (7.23)$$

or

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq M \\ 0, & \text{elsewhere.} \end{cases} \quad (7.24)$$

The impulse response given in Equation 7.24 is obviously of finite duration. Nevertheless, the output of any FIR system can be computed nonrecursively using Equation 7.20, wherein the coefficients are the values of the impulse response. In Equation 7.24,  $M$  represents the order of the filter. A direct implementation of an FIR filter would require  $M$  delayed samples of the input signal  $x[n]$ , to produce a single output  $y[n]$ . Following the analog delay-line terminology,  $M$  is also known as the number of taps.

### 7.3.1 Advantages of FIR Filters

FIR filters are nonrecursive, that is, they have no feedback terms. Therefore, the outputs are a function of a finite number of previous input signals. The major advantages of FIR filters are

1. FIR filters are easy to understand, easy to design and to implement, and amenable to being made adaptive. The simple implementation of adaptive FIR filters requires the filters to change their coefficients in real time to accommodate changes in external conditions. For example, the equalization filters in modems change their characteristics in response to transmission-line degradations.
2. Since FIR filters do not contain feedback terms, they have no poles in their transfer function. Therefore, FIR filters are guaranteed to be stable (unconditionally). The guaranteed FIR stability is vital for adaptive filter design.
3. FIR filters can be designed to have a perfectly linear phase, which implies that such filters will have a constant time or group delay with respect to the input signal. Linear phase is guaranteed as long as the FIR coefficients are symmetrical (or antisymmetrical) with respect to the center point of the impulse response. This is important in applications such as speech processing, sonar, and radar, where the knowledge of the time delay is necessary.
4. FIR filters have low sensitivity to coefficient accuracy. This feature allows FIR filter implementation with small word lengths. A typical range of FIR coefficient accuracy is 12–16 bits, while IIR filters require 16–24 bits per coefficient.
5. Using the symmetry or antisymmetry property of the linear-phase FIR filters, the number of multiplications per output sample roughly gets reduced by a factor of two.
6. Yet another advantage of FIR filters (over IIR filters) is the flexibility they offer in the desired frequency response. FIR filter frequency response magnitudes can be easily designed to approximate any specified function of frequency with a sufficient number of coefficients.



## 7.4 IIR Filters

IIR systems are described by Equations 7.13 and 7.17, wherein at least one of the feedback coefficients  $a_k$  is nonzero. The order of the system depends on the number of feedback terms and, therefore, from Equations 7.13 and 7.17, we can say that they represent an  $N$ th-order system.

### 7.4.1 Properties of IIR Filters

1. Unlike FIR filters, IIR filters can become unstable due to the presence of poles in the structure. Besides that, finite word length effects can also make them unstable.
2. No IIR filters does have a perfect linear-phase characteristic. Nevertheless, one can design IIR filters with very good phase linearity.
3. Since IIR filters are basically recursive, they tend to be more sensitive to round-off noise. Such noise can actually introduce spurious oscillations known as limit cycles.
4. IIR filters must be implemented more carefully than FIR. Delay-free loops can cause instability.
5. However, the major advantage of IIR filters is that for the same frequency characteristic,  $H(e^{j\omega})$ , they require lesser coefficients than FIR filters. This leads to fewer operations, thus being able to achieve higher throughput.
6. IIR filters require the smallest storage requirement, since they need a least number of coefficients for achieving specified characteristics. For example, an IIR highpass filter typically requires only one-third of the coefficients of an equivalent FIR filter.

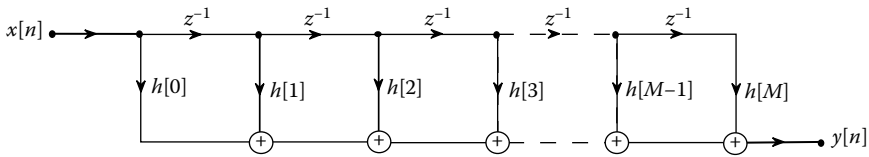
## 7.5 Structure of an FIR Filter

We will reproduce here the difference equation for an FIR filter as follows:

$$y[n] = \sum_{k=0}^M b_k x[n - k]. \quad (7.25)$$

This can be recognized as the discrete-time convolution of  $x[n]$  with the impulse response  $h[n]$  given as follows:

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq M \\ 0, & \text{elsewhere.} \end{cases} \quad (7.26)$$



**FIGURE 7.3**  
Realization of FIR filter.

The signal flow graph representation of one of the possible structures is shown in Figure 7.3. This structure is commonly referred to as tapped-delay line structure because of the presence of a chain of delay elements, represented by  $z^{-1}$ . It is also known as a *transversal* filter. The signal at each tap along the chain is multiplied by the coefficients of the impulse response, and the resulting products are summed to form the output  $y[n]$ . Often, the comparison between FIR and IIR or even better the various algorithms within each category may not be always clear-cut. Quite often, the available hardware, software, and *brainware* (know-how) may override some of the considerations while making a decision concerning whether to use an IIR or an FIR filter.

### 7.5.1 Filter Specifications

Ideal filters have a zero transition bandwidth, a constant passband, and a stopband with an infinite attenuation. In practice, we cannot achieve these specifications and we must allow some amount of tolerance. The permissible tolerances should be specified before designing the filter. For an LPF, we often have the following frequency response specifications:

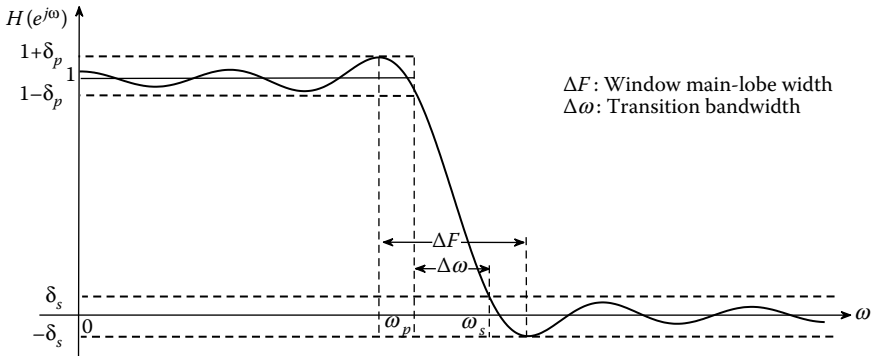
- $\delta_p$ : Peak passband deviation
- $\delta_s$ : Stopband deviation
- $\omega_p$ : Passband cut-off frequency
- $\omega_s$ : Stopband cut-off frequency
- $N$ : Filter length

All these specifications are depicted in Figure 7.4. In general,  $\delta_p$  and  $\delta_s$  on a linear scale are very small and it is often convenient to express them in dB. Therefore, they are expressed in terms of  $A_p$  and  $A_s$  (defined in Equations 7.27 and 7.28), respectively, as follows:

$$A_p = 20 \log_{10} \frac{1 + \delta_p}{1 - \delta_p} \tag{7.27}$$

and

$$A_s = -20 \log_{10} \delta_s \tag{7.28}$$



**FIGURE 7.4**  
FIR filter specifications.

Here  $\delta_p$  and  $\delta_s$  can be different for designing FIR filters. However, FIR filters designed using impulse response truncation (IRT) have the same passband and stopband ripples, that is,  $\delta = \delta_s = \delta_p$ . This is a serious drawback of the IRT technique. Very often, we have frequency response requirements on a linear scale where the ripples in the passband are allowed to be much larger than in the stopband. Also, we have equiripple requirements in both the bands. The IRT technique leads to filters where most ripples in both the bands are smaller than needed. These two requirements lead to a filter length that is larger than that found with digital optimization techniques. Here  $\omega_p$  and  $\omega_s$  are the passband and stopband cut-off frequencies, respectively, normalized with respect to the sampling frequency  $f_{\text{samp}}$ .

$$\omega_p = \frac{2\pi f_p}{f_{\text{samp}}} \quad \text{and} \quad \omega_s = \frac{2\pi f_s}{f_{\text{samp}}}.$$

The difference between passband and stopband cut-off frequencies,  $\Delta\omega$ , is approximately equal to the main-lobe width,  $\Delta F$ , of the window. This approximation is vital in calculating the time span of the window to obtain the required transition bandwidth.

---

## 7.6 FIR Filter Design

Any of the following three methods can be used to design FIR filters:

1. Impulse response truncation (windowing method)
2. Frequency sampling
3. Optimal method

Although frequency sampling and optimal methods give much better results for the given filter specifications, IRT (otherwise termed as windowing method) is considered as the simplest approach to design FIR filters. Before proceeding to the actual filter design procedure, we must first introduce the four types of linear-phase FIR filters.

### 7.6.1 Linear-Phase Filters

The frequency response of an ideal LPF with linear phase can be defined as

$$H_{lp}(e^{j\omega}) = \begin{cases} e^{-j\omega\alpha}, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi. \end{cases} \quad (7.29)$$

The magnitude response of this filter is

$$|H_{lp}(e^{j\omega})| = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi, \end{cases} \quad (7.30)$$

and the phase response is

$$\arg H_{lp}(e^{j\omega}) = -\alpha\omega. \quad (7.31)$$

The magnitude response and the phase response of the ideal LPF, with linear phase, are given in Figure 7.5. The impulse response corresponding to this ideal filter was derived earlier for  $\alpha = 0$ . The phase factor, however, only gives a shift over  $\alpha$  in the time-domain as follows:

$$h_{lp}[n] = \frac{\sin \omega_c(n - \alpha)}{\pi(n - \alpha)}, \quad -\infty < n < \infty. \quad (7.32)$$

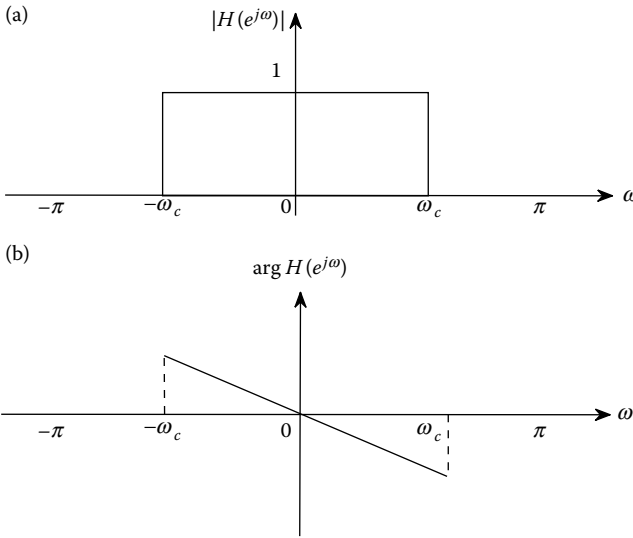
An additional phase of  $-\alpha\omega$  relates to the time shift of  $\alpha$  in the time-domain. Group delay ( $d_g(\omega)$ ) is one of the parameters used to measure the linearity of phase. It is defined as

$$d_g(\omega) = -\frac{d}{d\omega} \arg H(e^{j\omega}). \quad (7.33)$$

In this example, the group delay turns out to be

$$d_g(\omega) = -\frac{d}{d\omega}(-\alpha\omega) = \alpha. \quad (7.34)$$

Observe that the group delay is linearly independent of  $\omega$ . This will result in the frequency components of the signal delayed by the same amount, thereby



**FIGURE 7.5** Ideal LPF with linear phase: (a) magnitude response and (b) phase response.

avoiding the phase (or delay) distortion [2]. Also, note that the group delay for linear phase filters is nothing but the point of symmetry of impulse response.

Although there is no phase (or delay) distortion in either zero delay or constant group delay systems, these filters are nonrealizable due to their infinite extent. To get the finite duration causal linear-phase filter, we consider only the coefficients ranging from 0 to  $N - 1$ . Note that the truncated impulse response with zero delay is no more symmetric, resulting in the nonlinear-phase response. Therefore, we can always anticipate a small group delay  $\alpha$  as given in Equation 7.32. The impulse response of all the basic filters are shifted by  $\alpha$  and are given in Table 7.1 [3]. In the filter design procedures, these impulse responses are preferred rather than the one with zero delay. The number of coefficients that should be considered to design a causal filter depends on the specifications. Once the filter length is calculated, we can find the value of  $\alpha$  using the relation

$$\alpha = \frac{N - 1}{2}. \tag{7.35}$$

The coefficients of the FIR filter are therefore given by

$$h[n] = h_d[n]f[n] \tag{7.36}$$

$$= \frac{\sin \omega_c(n - \alpha)}{\pi(n - \alpha)}f[n]. \tag{7.37}$$

**TABLE 7.1**

Impulse Response of Linear-Phase Filters

Filter	Impulse Response ( $h[n], n \neq \alpha$ )	$h[\alpha]$
Lowpass	$\frac{\sin(\omega_c(n - \alpha))}{\pi(n - \alpha)}$	$\frac{\omega_c}{\pi}$
Highpass	$-\frac{\sin(\omega_c(n - \alpha))}{\pi(n - \alpha)}$	$1 - \frac{\omega_c}{\pi}$
Bandpass	$\frac{\sin(\omega_2(n - \alpha))}{\pi(n - \alpha)} - \frac{\sin(\omega_1(n - \alpha))}{\pi(n - \alpha)}$	$\frac{\omega_2 - \omega_1}{\pi}$
Bandstop	$\frac{\sin(\omega_1(n - \alpha))}{\pi(n - \alpha)} - \frac{\sin(\omega_2(n - \alpha))}{\pi(n - \alpha)}$	$1 - \frac{\omega_2 - \omega_1}{\pi}$

In Equation 7.37,  $f[n]$  represents the samples of the discrete-time data window that extends from 0 to  $N - 1$  and  $h_d[n]$  represents the desired impulse response. In this case,  $f[n]$  is a rectangular window. The frequency response of the filter, which is the periodic convolution of the DTFTs of  $h_d[n]$  and  $f[n]$ , is given by

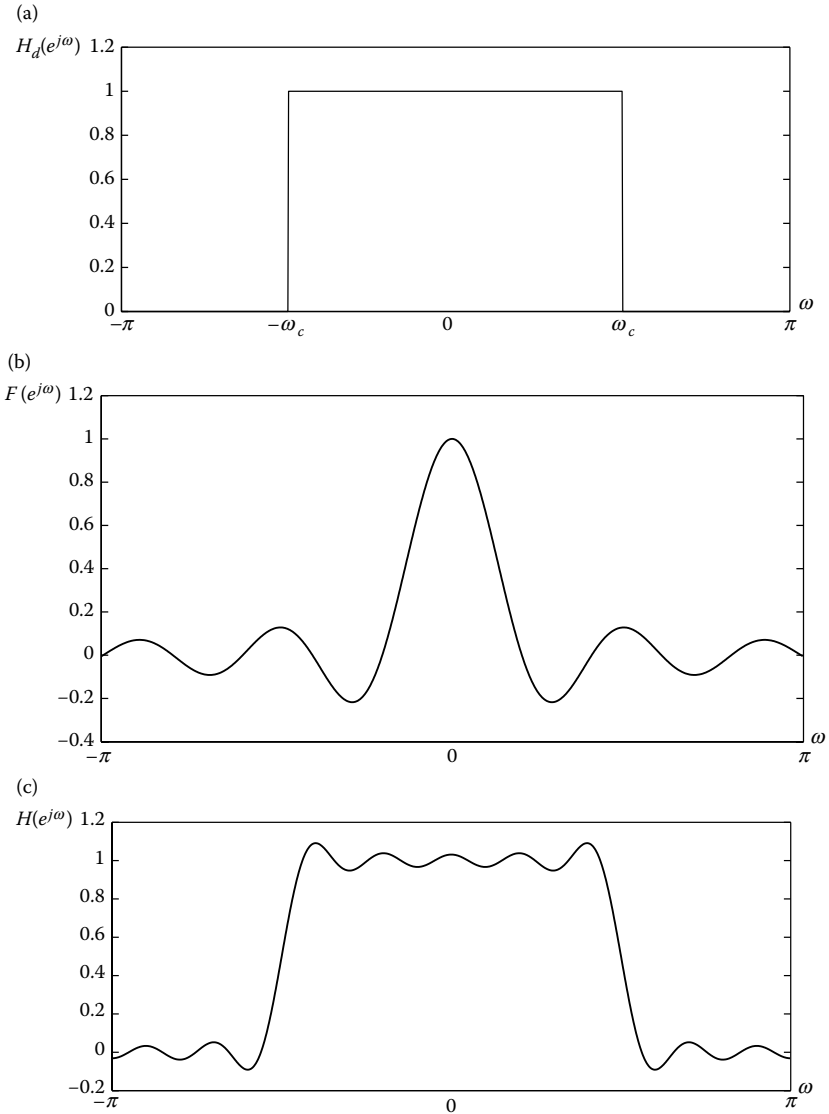
$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta})F(e^{j(\omega-\theta)})d\theta. \tag{7.38}$$

This can be written in short hand notation as

$$H(e^{j\omega}) = \frac{1}{2\pi} H_d(e^{j\omega}) * F(e^{j\omega}). \tag{7.39}$$

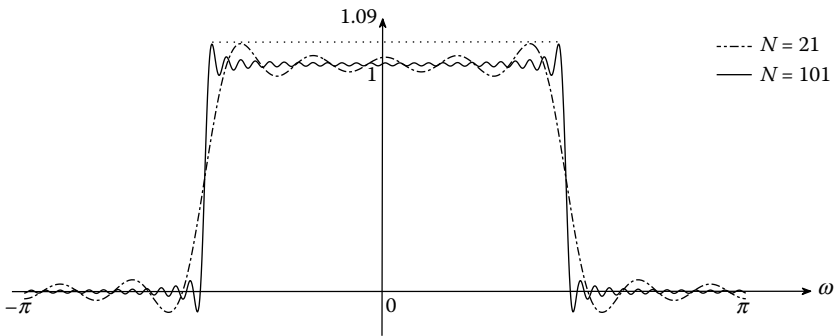
Here,  $H(e^{j\omega})$  represents the periodic convolution of the desired ideal frequency response with the Fourier transform of the window. Figure 7.6(a) shows the ideal frequency response of a LPF with cut-off frequency  $\omega_c$ . To obtain the frequency response of the FIR filter, the desired frequency response  $H_d(e^{j\omega})$  is convolved with the frequency response of the window  $F(e^{j\omega})$  (shown in Figure 7.6(b)). The resulting response  $H(e^{j\omega})$  is depicted in Figure 7.6(c).

From Figure 7.6(c), we can observe that the passband and stopband ripples are no longer constant and the transition width is not equal to zero. The side lobes present in the frequency window are responsible for the ripples in the stopband and passband, whereas the main-lobe width of the frequency response of the window is responsible for the nonzero transition width. In general, the larger the main-lobe width of the frequency window, the more the smearing of the filter response. In fact, the transition bandwidth is approximately equal to the main-lobe width ( $\frac{4\pi}{N}$  in the case of the rectangular window) as shown in Figure 7.4. To obtain a sharp transition band, we have to increase the value of  $N$  (i.e., number of coefficients in the filter response). However, higher values of  $N$  require more computations. Therefore, we are always left with a trade-off between smearing of the desired response and the computational complexity. Another important point to note is that the minimum attenuation remains constant irrespective of the filter length chosen for a



**FIGURE 7.6** Truncation of impulse response using the rectangular window. (a) Desired frequency response. (b) Frequency response of rectangular window. (c) Frequency response of truncated signal.

given window. This minimum attenuation is termed as *Gibbs number*. Gibbs number for the rectangular window is 0.0895 [4]. This is illustrated in Figure 7.7. Observe that the transition bandwidth of the filter designed with  $N = 101$  has a steeper transition band than the one designed with  $N = 21$ . However, the maximum ripple remains the same in both the cases.



**FIGURE 7.7**  
 FIR filters designed with direct truncation with filter lengths  $N = 21$  and  $N = 101$ .

Although a rectangular window gives the smallest transition width for a given value of  $N$ , its side lobes are much larger. By tapering the window at both ends, we can reduce the side-lobe levels, but at the expense of increased main-lobe widths, thereby increasing the transition bandwidth.

The peak side-lobe level (PSLL) of the window, the transition bandwidth, and the minimum stopband attenuation of the filter designed with some common windows are given in Table 7.2 [5]. From Table 7.2, we can see that there are two values for the side-lobe level, namely, PSLL and minimum stopband attenuation. We have seen that in the case of the rectangular window, it was mentioned in an earlier chapter that it has a PSLL of  $-13$  dB. For the design of the filter as described here, this results in a first stopband side lobe of  $-21$  dB.

**TABLE 7.2**  
 Filter Characteristics Using Different Windows

Window	Peak Side-Lobe Level	Transition Width	Minimum Stopband Attenuation
Rectangular	$-13$	$4\pi/N$	$-21$
Bartlett	$-26$	$8\pi/N$	$-25$
Riesz	$-21$	$5.72\pi/N$	$-31$
Riemann	$-27$	$6.56\pi/N$	$-39$
Hann	$-31$	$8\pi/N$	$-44$
Bohman	$-46$	$12\pi/N$	$-52$
Hamming	$-41$	$8\pi/N$	$-53$
de la Vallé-Poussin	$-53$	$16\pi/N$	$-57$
Blackman	$-57$	$12\pi/N$	$-74$
Exact Blackman	$-51$	$12\pi/N$	$-85$
Blackman–Harris (minimum three-term)	$-71$	$12\pi/N$	$-86$
Blackman–Harris (minimum four-term)	$-92$	$16\pi/N$	$-109$



If the cut-off frequency goes to zero, this first side-lobe level of the resulting frequency response gradually tends to  $-13$  dB. This is because  $H_d(e^{j\omega})$  in this case approaches an impulse function. Similar values for the minimum stopband attenuation for other windows are presented in Table 7.2.

### 7.6.2 Types of FIR Filters

Let  $h[n]$  be the causal finite length impulse response of the filter. Then, the frequency response of the filter is given by

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n]e^{-j\omega n}. \quad (7.40)$$

This can be expressed in terms of the amplitude response  $A(e^{j\omega})$  and the phase response  $\theta(\omega)$  as follows:

$$H(e^{j\omega}) = A(e^{j\omega})e^{j\theta(\omega)}. \quad (7.41)$$

The amplitude response  $A(e^{j\omega})$  can take both positive and negative real values. Phase response  $\theta(\omega)$  should be piece-wise linear to avoid group delay distortions, that is,

$$\theta(\omega) = \alpha\omega - \beta, \quad (7.42)$$

where  $\alpha$  and  $\beta$  are arbitrary constants. Equation 7.40 can be rewritten by expressing  $e^{-j\omega n}$  as the sum of sine and cosine terms to give

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n] \cos(\omega n) + j \sum_{n=0}^{N-1} h[n] \sin(\omega n). \quad (7.43)$$

The phase response can also be computed from Equation 7.43 using the following expression:

$$\theta(\omega) = \tan^{-1} \left[ \frac{\sum_{n=0}^{N-1} h[n] \sin(\omega n)}{\sum_{n=0}^{N-1} h[n] \cos(\omega n)} \right]. \quad (7.44)$$

Equating the phase response that we obtained in Equations 7.42 and 7.44, we get

$$(\alpha\omega - \beta) = \tan^{-1} \left[ \frac{\sum_{n=0}^{N-1} h[n] \sin(\omega n)}{\sum_{n=0}^{N-1} h[n] \cos(\omega n)} \right]. \quad (7.45)$$

Taking tan on both sides of Equation 7.45, we obtain

$$\tan(\alpha\omega - \beta) = \frac{\sin(\alpha\omega - \beta)}{\cos(\alpha\omega - \beta)} = \frac{\sum_{n=0}^{N-1} h[n] \sin(\omega n)}{\sum_{n=0}^{N-1} h[n] \cos(\omega n)}. \quad (7.46)$$

After cross-multiplying and simplifying the above equation, we obtain

$$\sum_{n=0}^{N-1} h[n] \sin(\omega(\alpha - n) + \beta) = 0. \quad (7.47)$$

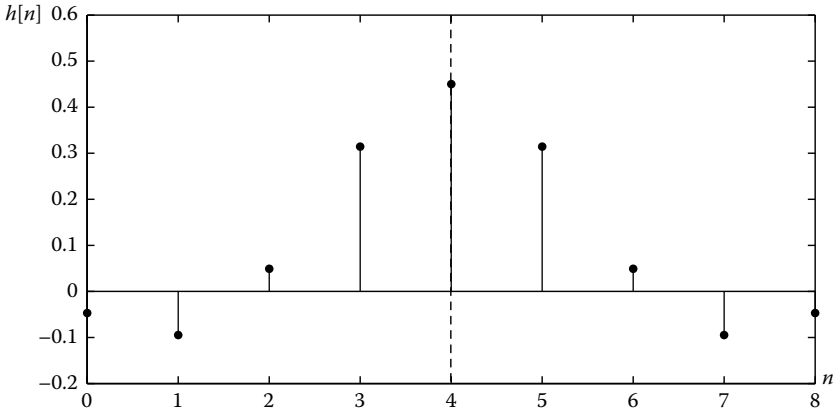
The solutions of this equation are given by [5,6]

1.  $\beta = 0, \alpha = (N - 1)/2$  and  $h[n] = h[N - 1 - n], \quad 0 \leq n \leq N - 1.$
2.  $\beta = \pm \frac{\pi}{2}, \alpha = (N - 1)/2$  and  $h[n] = -h[N - 1 - n], \quad 0 \leq n \leq N - 1.$

Note that here  $\alpha$  is the total group delay and can be either an integer or (integer +  $\frac{1}{2}$ ), depending on the value of  $N$ . Then, depending on whether  $h[n]$  is even symmetric or odd symmetric, and  $N$  being odd or even, four types of linear-phase FIR filters can be defined as follows:

1. *Type 1:* Even symmetric impulse response with odd filter length
2. *Type 2:* Even symmetric impulse response with even filter length
3. *Type 3:* Odd symmetric impulse response with odd filter length
4. *Type 4:* Odd symmetric impulse response with even filter length

The group delay is an integer for types 1 and 3, whereas it is (integer +  $\frac{1}{2}$ ) for types 2 and 4. This additional half group delay obtained for type 2 and type 4 FIR filters actually turns out to be useful in some applications. The impulse response of the LPF for these four FIR filter types are shown in Figures 7.8 through 7.11. It should be noted that in the case of antisymmetric impulse response with  $N$  odd (type 3),  $h[n]$  is always zero at the center point.

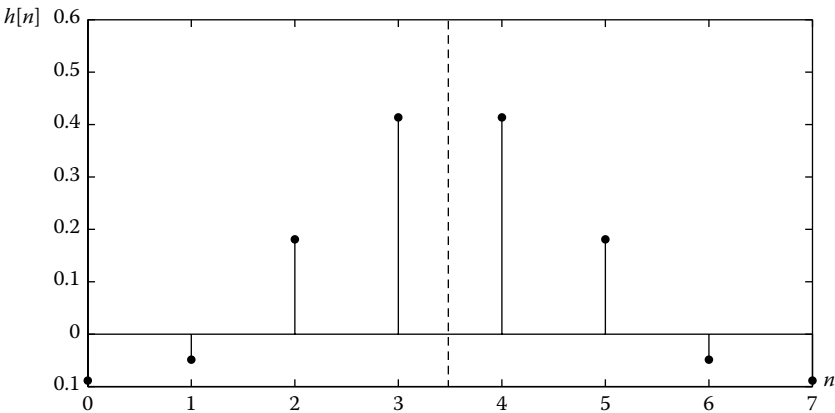


**FIGURE 7.8**  
Type 1 FIR impulse response.

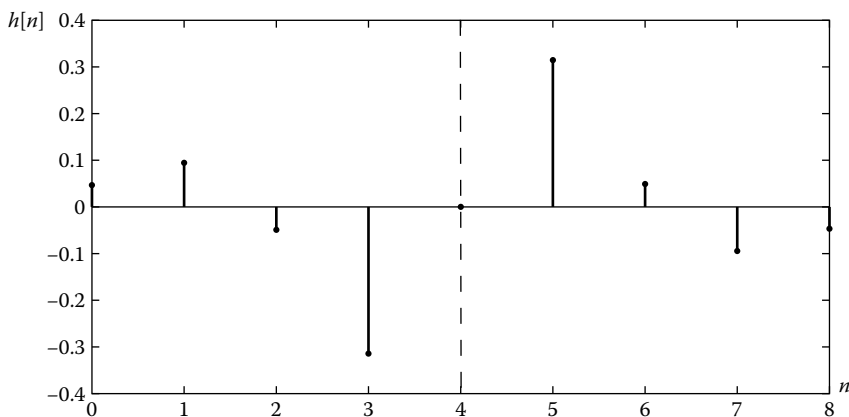
### 7.6.3 Frequency Response of Type 1 FIR Filter

In this section, we will actually derive the expression for the frequency response of type 1 FIR filter. Since the impulse response is symmetric and the filter length is odd, the frequency response of type 1 FIR filter can be written as follows:

$$\begin{aligned}
 H(e^{j\omega}) &= \sum_{n=0}^{N-1} h[n]e^{-j\omega n} \\
 &= \sum_{n=0}^{(N-3)/2} h[n]e^{-j\omega n} + h\left[\frac{N-1}{2}\right]e^{-j\omega(N-1)/2} + \sum_{n=(N+1)/2}^{N-1} h[n]e^{-j\omega n}. \quad (7.48)
 \end{aligned}$$



**FIGURE 7.9**  
Type 2 FIR impulse response.



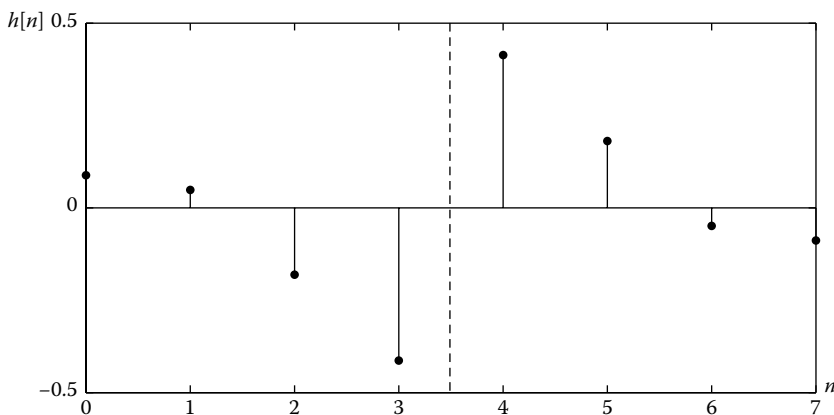
**FIGURE 7.10**  
Type 3 FIR impulse response.

Replacing  $n$  with  $(N - 1 - m)$  in the third term, we obtain

$$H(e^{j\omega}) = \sum_{n=0}^{(N-3)/2} h[n]e^{-j\omega n} + h\left[\frac{N-1}{2}\right]e^{-j\omega(N-1)/2} + \sum_{m=0}^{(N-3)/2} h[N-1-m]e^{-j\omega(N-1-m)}. \tag{7.49}$$

Since  $h[n] = h[N - 1 - n]$ , we get

$$H(e^{j\omega}) = \sum_{m=0}^{(N-3)/2} h[m] \left( e^{j\omega(\frac{N-1}{2}-m)} + e^{j\omega(\frac{N-1}{2}-m)} \right) + h\left[\frac{N-1}{2}\right] e^{-j\omega(N-1)/2}. \tag{7.50}$$



**FIGURE 7.11**  
Type 4 FIR impulse response.

Replacing  $m$  with  $[(N - 1)/2 - n]$  in the above equation and on simplifying, we obtain

$$H(e^{j\omega}) = e^{-j\omega(N-1)/2} \left[ \sum_{n=0}^{(N-3)/2} 2h \left[ \frac{N-1}{2} - n \right] \cos(\omega n) + h \left[ \frac{N-1}{2} \right] \right] \tag{7.51}$$

or, equivalently

$$H(e^{j\omega}) = e^{j\omega(N-1)/2} \left[ \sum_{n=0}^{(N-1)/2} a[n] \cos(\omega n) \right] \tag{7.52}$$

where

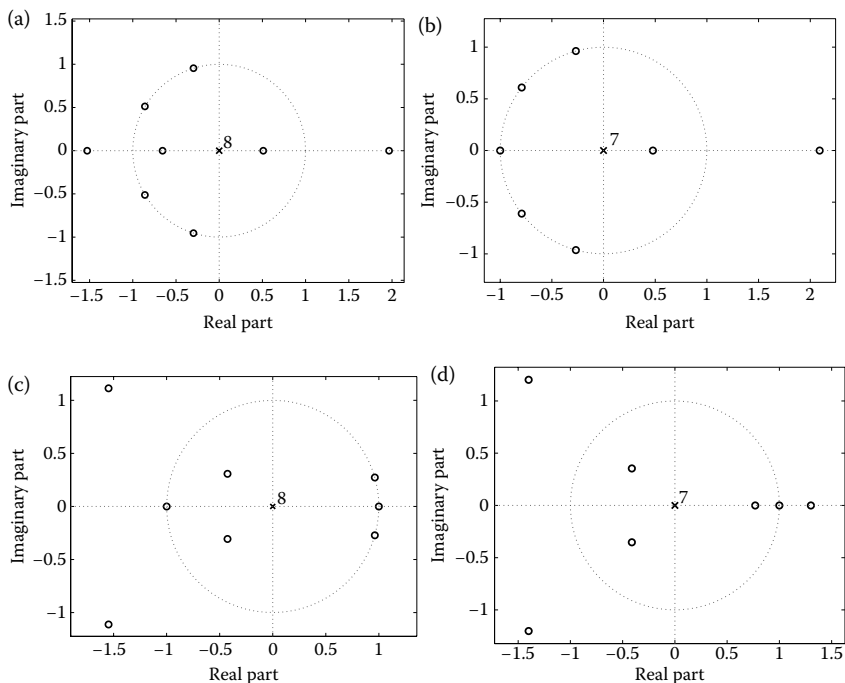
$$a[n] = \begin{cases} 2h \left[ \frac{N-1}{2} - n \right], & n = 1, 2, \dots, (N-1)/2 \\ h \left[ \frac{N-1}{2} \right], & n = 0. \end{cases} \tag{7.53}$$

Similarly, the frequency responses of the other three FIR filter types can be derived. The filter types, along with the corresponding frequency responses, are listed in Table 7.3.  $b[n]$  given for frequency responses of type 2 and type 4 filters in the table is defined as follows:

$$b[n] = 2h[N/2 - n]. \tag{7.54}$$

**TABLE 7.3**  
Linear-Phase FIR Filter Types

Filter Type	$N$	Symmetry	Frequency Response $H(e^{j\omega})$	Constrained Zeros	Filters That Can Be Designed
1	Odd	Even	$e^{j\omega(N-1)/2} \sum_{n=0}^{(N-1)/2} a[n] \cos(\omega n)$	—	Any filter
2	Even	Even	$e^{j\omega(N-1)/2} \sum_{n=1}^{N/2} b[n] \cos \left( \omega \left( n - \frac{1}{2} \right) \right)$	$\omega = \pi$	LPF, BPF
3	Odd	Odd	$je^{-j\omega(N-1)/2} \sum_{n=0}^{(N-1)/2} a[n] \sin(\omega n)$	$\omega = 0, \pi$	BPF, differentiator, Hilbert transformer
4	Even	Odd	$je^{-j\omega(N-1)/2} \sum_{n=1}^{N/2} b[n] \sin \left( \omega \left( n - \frac{1}{2} \right) \right)$	$\omega = 0$	HPF, BPF, differentiator, Hilbert transformer



**FIGURE 7.12** Pole-zero plots of the four FIR filter types. (a) Type 1. (b) Type 2. (c) Type 3. (d) Type 4.

The frequency response of type 2 FIR filter is zero at  $\omega = \pi$ , irrespective of the filter coefficients. Hence, there is a constrained zero at  $\omega = \pi$ , which implies that we cannot design filters with nonzero attenuation at  $\omega = \pi$ . For instance, we cannot design highpass or bandstop filters using type 2 FIR filter. Similarly, we find that type 3 FIR filter has constrained zeros at  $\omega = 0$  and  $\omega = \pi$ , whereas type 4 FIR filter has constrained zero at  $\omega = 0$ . Therefore, type 3 FIR filter can be used for designing BPFs. Type 4 is useful for designing highpass, bandpass, differentiators, and Hilbert transformers (see Section 7.9). Note that type 1 FIR filter is the most versatile one as it has no constrained zeros. Therefore, it can be used for designing any type of filter. The typical pole-zero plots of the four FIR filter types are given in Figure 7.12.

### 7.6.4 Design Procedure for Filters

1. First, we design an ideal filter with the given specifications, by assuming the cut-off frequency ( $\omega_c$ ) to be the mean of passband and

stopband cut-off frequencies, that is,

$$\omega_c = \frac{(\omega_p + \omega_s)}{2}.$$

2. Determine the impulse response  $h_d[n]$  of the ideal filter from Table 7.1.
3. From Table 7.2, we can select the window that has the smallest main-lobe width and can satisfy the minimum stopband attenuation requirement. Note that the stopband and passband ripples are equally affected by the side lobes of the window. Therefore, the maximum passband attenuation (or the minimum stopband attenuation) is selected to be

$$20 \log_{10}[\min\{\delta_p, \delta_s\}]. \quad (7.55)$$

4. Find the number of filter coefficients ( $N$ ) using the relation between transition bandwidth and  $N$  given in Table 7.2.
5. Truncate the impulse response using the selected window to obtain the symmetric filter coefficients.

Note that the value  $N$  computed in step 4 is only an approximate value. Therefore, if the desired filter specifications are not met, then we repeat the procedure with a different value of  $N$ .

### Example 1

Design an HPF using the IRT method with the following specifications:

$$\omega_p = 0.65\pi \text{ rad/sample} \quad \omega_s = 0.55\pi \text{ rad/sample} \quad A = 54 \text{ dB}$$

From the windows listed in Table 7.2, the Hamming window satisfies the minimum attenuation criterion and it has the narrowest main-lobe width (for the given attenuation), when compared to other windows. Therefore, we choose the Hamming window to design this filter with the above specifications.

Transition bandwidth of the filter ( $\Delta\omega$ ) =  $(\omega_p - \omega_s) = 0.1\pi$

The main-lobe width of the Hamming window is  $\frac{8\pi}{N}$ .

As stated earlier, the transition bandwidth is approximately equal to the main-lobe width of the window. Therefore, we can calculate the filter length ( $N$ ) from this approximation.

$$\begin{aligned} \frac{8\pi}{N} &\approx 0.1\pi \\ \implies N &\approx 80. \end{aligned}$$

Since we cannot use type 2 FIR filter for designing HPF, we select  $N = 81$  and we use a type 1 linear-phase filter. Now, we can obtain the filter coefficients by truncating the impulse response with the Hamming window

$f[n]$  of length  $N = 81$ .

$$h_d[n] = \delta[n] - \frac{\sin(0.6\pi n)}{\pi n}, \quad -\infty < n < \infty$$

$$h[n] = h_d[n - 40]f[n], \quad 0 \leq n \leq 80.$$

The magnitude response of the filter designed with the Hamming window is given in Figure 7.13. To understand the effect of windows on filter attenuation characteristics, let us design the filter with the same specifications using the Hann and Blackman windows as well.

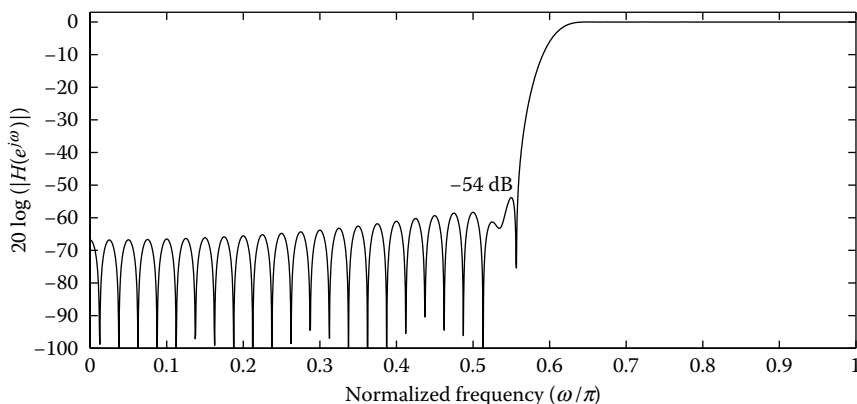
Since the main-lobe width of the Hann window is the same as that of the Hamming window, for a given transition width, the filter length remains the same. The frequency response of the filter truncated with the Hann window is given in Figure 7.14. Observe that the minimum stopband attenuation is only  $-44$  dB. Even if we decrease the transition width by taking a higher filter order, we cannot meet the required minimum stopband attenuation. Therefore, we can use the Hann window for the filter design only if the stopband attenuation is less than  $-44$  dB (see Table 7.2).

Now, let us design the filter using the Blackman window. The main-lobe width of the Blackman window is  $\frac{12\pi}{N}$ . Therefore, the filter length  $N$  is given by

$$\frac{12\pi}{N} \approx 0.1\pi$$

$$\implies N \approx 120.$$

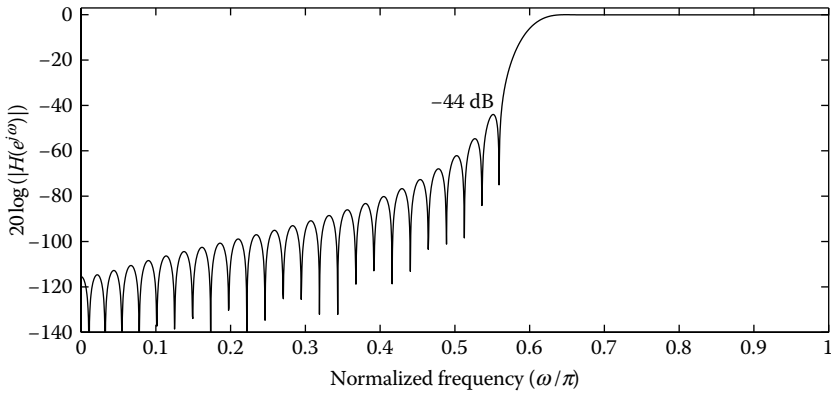
The filter length is increased by one to design the filter with type 1 (odd length) FIR filter. Filter coefficients can be obtained by truncating the



**FIGURE 7.13**

Magnitude response of the filter designed with the Hamming window.



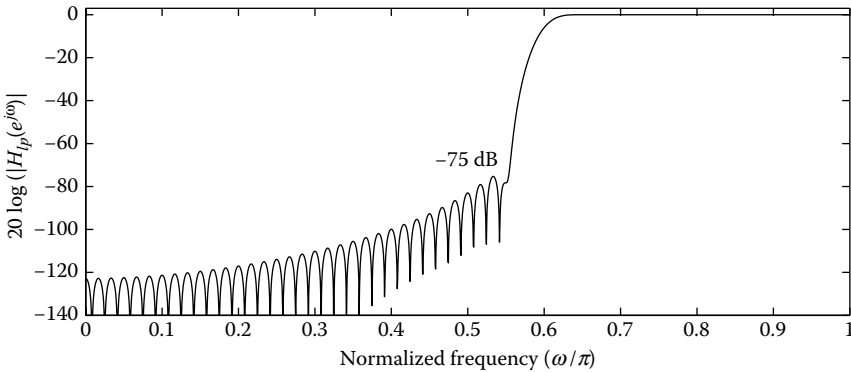


**FIGURE 7.14**  
Magnitude response of the filter designed with the Hann window.

impulse response as follows:

$$h[n] = h_i[n - 60]f[n], \quad 0 \leq n \leq 120 \tag{7.56}$$

where the window function  $f[n]$  used is the Blackman window of length  $N = 121$ . From the magnitude response plot of the filter designed using the Blackman window (Figure 7.15), we can observe that the minimum side-lobe attenuation is  $-75$  dB, that is, the designed filter has outperformed the required specifications. However, we make an observation at this juncture that to obtain the same transition bandwidth, the Blackman window approach requires a larger filter length than the Hann or the Hamming window.



**FIGURE 7.15**  
Magnitude response of the filter designed with the Blackman window.

## 7.7 Kaiser–Bessel Windows for FIR Filter Design

From the above example, we have seen that the filters designed using the window functions given in Table 7.2 have fixed values of  $\delta$ . Therefore, in many cases, these windows may not satisfy (or even outperform) the required specifications. Kaiser–Bessel windows (both  $I_0$ –sinh and  $I_1$ –cosh families) do not suffer from the above limitations. These windows have adjustable shape parameters ( $\alpha$  and  $\gamma$ ) that allow us to choose the appropriate window to obtain any desired value of ripple  $\delta$  or attenuation. Such windows are called variable windows. The filter design procedure using the Kaiser–Bessel zeroth-order ( $I_0$ –sinh) and first-order ( $I_1$ –cosh) windows are discussed in this section.

### 7.7.1 Filter Design Using Kaiser–Bessel Zeroth-Order ( $I_0$ –Sinh) Window

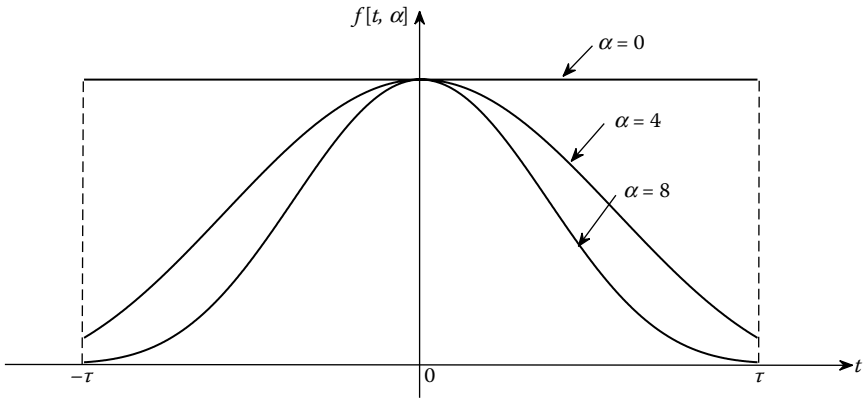
The  $I_0$ –sinh window is unique in the above class since it has a near-optimum performance, in the sense that it has maximum energy in the main-lobe of its Fourier transform as well as a relatively simple implementation. This window depends on two parameters: the shape parameter  $\alpha$  and its length  $N$ . For an odd length  $N = 2M + 1$ , the  $I_0$ –sinh window is defined, for  $n = 0, 1, \dots, (N - 1)$ , as follows:

$$f[n] = \frac{I_0 \left[ \alpha \sqrt{1 - (n - M)^2 / M^2} \right]}{I_0(\alpha)}, \quad 0 \leq n \leq 2M \quad (7.57)$$

where  $I_0(x)$  is the modified Bessel function of the first kind and order zero. The numerator of Equation 7.57 can be rewritten in the following form. This form is more convenient for the purpose of numerical evaluation:

$$f[n] = \frac{I_0 \left[ \alpha \sqrt{n(2M - n)} / M \right]}{I_0(\alpha)}. \quad (7.58)$$

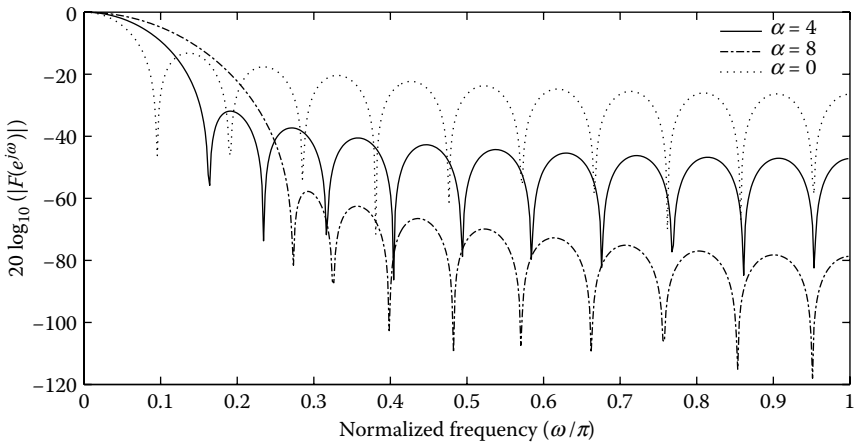
The  $I_0$ –sinh window is symmetric about its center point, that is,  $n = M$ , and has the value  $f[M] = 1$ . At the edges of the window, that is, at  $n = 0$  and  $n = (N - 1)$ , it has the value  $\frac{1}{I_0(\alpha)}$ , since  $I_0(0) = 1$ . Note that the case  $\alpha = 0$  in Equation 7.58 reduces to the rectangular window. Figure 7.16 shows the time function of the  $I_0$ –sinh window for  $\alpha = 0, 4$ , and 8. Figure 7.17 displays the magnitude of the frequency response of the  $I_0$ –sinh window for  $\alpha = 0, 4$ , and 8, with  $N$  being constant. Figure 7.18 shows the magnitude response of the  $I_0$ –sinh window for  $N = 11, 21$ , and 41, with  $\alpha$  held constant, that is,  $\alpha = 8$ . It can be clearly observed from Figure 7.17 that by increasing  $\alpha$ , the side-lobe level decreases, at the expense of increased main-lobe width. Similarly, from Figure 7.18, we can see that the resolution is increased by varying the filter length from  $N = 11$  to 41, but the side-lobe levels remain the same in all



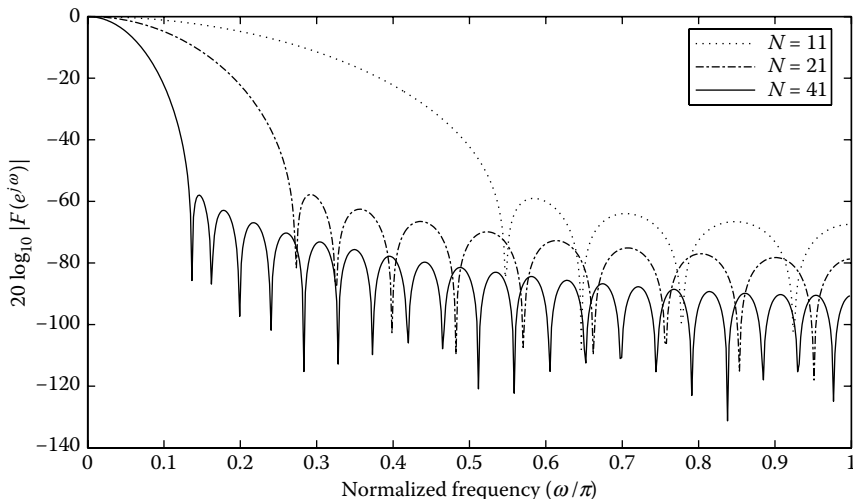
**FIGURE 7.16**  
Kaiser window with  $\alpha = 0, 4, \text{ and } 8$ .

the cases. Therefore, we note that the desired trade-off between the side-lobe amplitude and the main-lobe width can be achieved by varying  $N$  and  $\alpha$ . Also note that if a window in the time-domain is tapered more, the side lobes of the Fourier transform becomes smaller, but the main-lobe width gets broader.

Kaiser had proposed a pair of equations that allow the filter designer to predict the values  $N$  and  $\alpha$  required to meet certain given frequency-selective filter specifications [7,8]. The window parameters ( $N, \alpha$ ) can be computed in terms of the filter specifications, namely the ripple  $\delta$  and the transition width  $\Delta\omega$ . The design equations developed by Kaiser [7,8] are given in the remaining section.



**FIGURE 7.17**  
Magnitude Responses of  $I_0$ -sinh window with  $\alpha = 0, 4, \text{ and } 8$ , and  $N = 21$ .



**FIGURE 7.18** Magnitude responses of  $I_0$ -sinh window with parameters  $N = 11, 21,$  and  $41,$  and  $\alpha = 8.$

We define

$$\begin{aligned} \Delta\omega &= (\omega_s - \omega_p) \quad \text{and} \\ \delta &= \delta_s = \delta_p. \end{aligned} \tag{7.59}$$

Therefore, the resulting filter will have passband and stopband ripples equal to  $\delta.$  This value of  $\delta$  is usually expressed in dB as follows:

$$A = -20 \log_{10} \delta \quad \text{or} \quad \delta = 10^{-A/20}. \tag{7.60}$$

The shape parameter  $\alpha$  of the  $I_0$ -sinh window was determined empirically by Kaiser and can be calculated from Equation 7.61 as follows:

$$\alpha = \begin{cases} 0.1102(A - 8.7), & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & 21 \leq A \leq 50 \\ 0, & A < 21. \end{cases} \tag{7.61}$$

Recall that when  $\alpha = 0,$  the  $I_0$ -sinh window becomes a rectangular window for which  $A = 21.$  Note that  $A$  represents the ripple in dB given by Equation 7.60. The filter order  $N$  can also be calculated from  $A.$  We know that the filter order is related to the transition width. Kaiser has also found that to achieve the prescribed values of  $A$  and  $\Delta\omega,$   $N$  must satisfy the relation as follows:

$$N = \frac{(A - 7.95)}{2.285 \Delta\omega}. \tag{7.62}$$

Equation 7.62 predicts  $N$  to be within  $\pm 2$  over a wide range of values of  $\Delta\omega$  and  $A$ . Therefore, by using the closed-form expressions, the  $I_0$ -sinh window method avoids the trial-and-error approach.

In most of the practical applications, the value of the attenuation  $A$  is always greater than 50 dB, and therefore we generally use the following formulae for calculating  $\alpha$  and  $M$ :

$$\alpha = 0.1102(A - 8.7), \quad N = \frac{(A - 7.95)}{2.285 \Delta\omega}. \quad (7.63)$$

For designing an LPF using the  $I_0$ -sinh window, we summarize the procedure as follows:

1. Find the passband ripple ( $\delta_p$ ) and the stopband ripple ( $\delta_s$ ) from the actual passband attenuation ( $A_p$ ) and the minimum stopband attenuation ( $A_s$ ) using the following formulae:

$$\delta_p = \frac{10^{0.05A_p} - 1}{10^{0.05A_p} + 1} \quad \text{and} \quad \delta_s = 10^{-0.05A_s}. \quad (7.64)$$

2. Determine the stopband attenuation ( $A$ ) using the following equation:

$$A = -20 \log_{10} \min\{\delta_p, \delta_s\}. \quad (7.65)$$

3. Calculate the shape parameter ( $\alpha$ ) of the  $I_0$ -sinh window and filter length ( $N$ ) from Equations 7.61 and 7.62, respectively.
4. Plug in the values of  $N$  and  $\alpha$  in Equation 7.57 to obtain  $I_0$ -sinh window coefficients.
5. Truncate the ideal impulse response  $h_d[n]$  using the  $I_0$ -sinh window coefficients (obtained in step 4) to get the required filter coefficients.

### Example 2

Design an LPF using the  $I_0$ -sinh window with the following filter specifications:

$$\begin{aligned} \omega_p &= 0.2\pi \text{ rad/sample} & A_p &= 0.1 \text{ dB} \\ \omega_s &= 0.3\pi \text{ rad/sample} & A_s &= 43 \text{ dB} \end{aligned}$$

We can calculate  $\delta_p$  and  $\delta_s$  using Equations 7.27 and 7.28, respectively, as follows:

$$\delta_p = \frac{10^{0.05A_p} - 1}{10^{0.05A_p} + 1} = 0.0058$$

$$\delta_s = 10^{-0.05A_s} = 0.0071.$$

The passband ripple is less than the stopband ripple. Therefore, choose

$$\delta = \min\{\delta_p, \delta_s\} = \delta_p = 0.0058.$$

Minimum stopband attenuation:  $A = -20 \log \delta = 44.797$  dB.

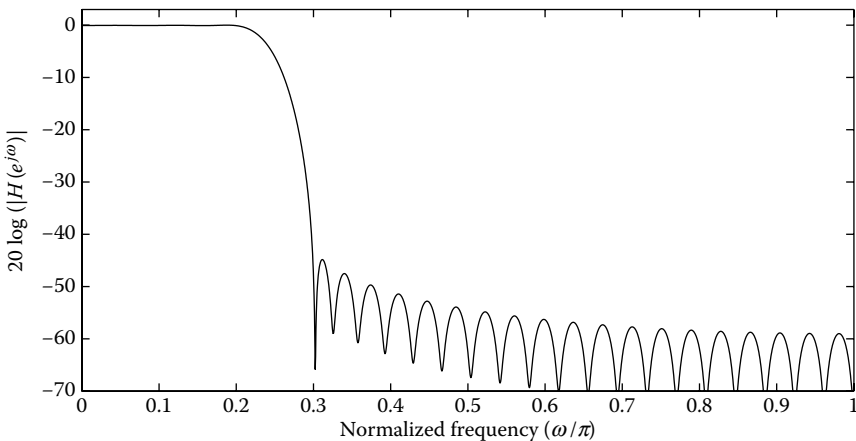
The filter length  $N$  can be calculated using Equation 7.62 as follows:

$$N = \frac{(44.797 - 7.95)}{2.285(0.1\pi)} \approx 52.$$

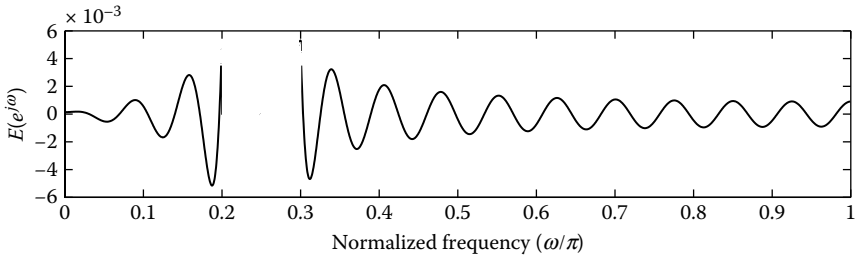
Substituting the value of  $A$  in Equation 7.61 yields

$$\alpha = 0.5842(44.797 - 21)^{0.4} + 0.07886(44.797 - 21) = 3.9524.$$

Compute the filter coefficients as stated in steps 4 and 5 in the design procedure. The frequency response of the designed FIR filter is shown in Figure 7.19. To observe the ripples in the passband, the error between the desired frequency response and the frequency response of the designed FIR filter is also provided in Figure 7.20. It is clear from this plot that the maximum passband and stopband ripples are equal in magnitude but less than 0.0058.



**FIGURE 7.19** Magnitude response of the filter designed with the Kaiser–Bessel window ( $N = 59$ ,  $\alpha = 3.9524$ ).



**FIGURE 7.20**  
Error plot.

**Example 3**

Design a BPF using the  $I_0$ -sinh window with the following specifications:

$$\omega_{s1} = 0.35\pi \text{ rad/sample} \quad \omega_{p1} = 0.44\pi \text{ rad/sample}$$

$$\omega_{p2} = 0.72\pi \text{ rad/sample} \quad \omega_{s2} = 0.83\pi \text{ rad/sample}$$

$$A_p = 0.1 \text{ dB}, \quad A_s = 56 \text{ dB}.$$

The transition widths are  $\Delta\omega_1 = \omega_{p1} - \omega_{s1} = 0.09\pi$  and  $\Delta\omega_2 = \omega_{s2} - \omega_{p2} = 0.11\pi$ .

Since the transition width of the filter depends on the main-lobe width, we cannot have different transition widths in a window-based filter design. Therefore, we always choose the minimum transition bandwidth for designing multiband filters. In this example, we choose  $\Delta\omega = 0.09\pi$ .

Since the transition width is changed, the cut-off frequency is no longer the mean of the passband and the stopband frequencies. The new cut-off frequencies can be computed as follows:

$$\omega_{c1} = \omega_{p1} - \Delta\omega/2 = 0.395\pi$$

$$\omega_{c2} = \omega_{p2} + \Delta\omega/2 = 0.765\pi.$$

The ripples  $\delta_p$  and  $\delta_s$  can be calculated using Equations 7.27 and 7.28 as follows:

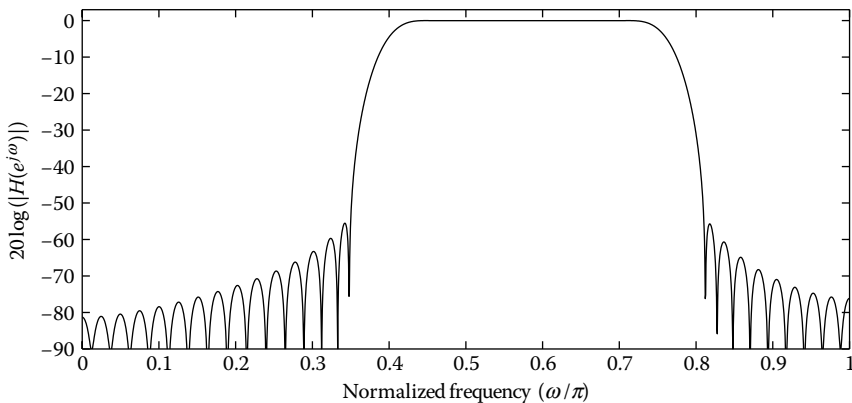
$$\delta_p = \frac{10^{0.05A_p} - 1}{10^{0.05A_p} + 1} = 0.0058$$

and

$$\delta_s = 10^{-0.05A_s} = 0.00158.$$

Minimum stopband attenuation  $A = -20 \log \delta = 56 \text{ dB}$ .  
Filter length  $N$  can be calculated as

$$N = \frac{(56 - 7.95)}{2.285(0.09\pi)} = 75.$$



**FIGURE 7.21** Magnitude response of the bandpass filter designed with the  $I_0$ -sinh window.

The shape parameter can be obtained from Equation 7.61 as

$$\alpha = 0.1102(56 - 8.7) = 5.2125.$$

The magnitude response of the designed filter is given in Figure 7.21. From the figure, we can see that this filter satisfies all the given specifications. From these two examples, we can conclude that the  $I_0$ -sinh window can be used to design any filter with arbitrary minimum side-lobe attenuation and transition width, by selecting appropriate values of  $\alpha$  and  $N$ .

### 7.7.2 Filter Design Using Kaiser-Bessel First-Order ( $I_1$ -Cosh) Window

The time-domain expression for the  $I_1$ -cosh window [9] is

$$f[n] = \frac{I_1 \left[ \gamma \sqrt{1 - \left( \frac{n-N/2}{N/2} \right)^2} \right]}{I_1[\gamma] \sqrt{1 - \left( \frac{n-N/2}{N/2} \right)^2}}, \quad 0 \leq n \leq N \tag{7.66}$$

where  $I_1[\gamma]$  is the modified Bessel function of the first kind and first order, and  $\gamma$  is the shape parameter of the window. Similar to the Kaiser-Bessel zeroth-order window, by adjusting the values of  $N$  and  $\gamma$ , we can get the desired main-lobe width and minimum stopband attenuation in the case of the Kaiser-Bessel first-order window [9].



The first three steps in the filter design procedure using the  $I_1$ -cosh window are similar to the design procedure outlined for the  $I_0$ -sinh windows in Section 7.7.1. Once we get the attenuation  $A$  from step 3, we can calculate  $\gamma$  and the filter length using the relations given below:

$$\gamma = \begin{cases} 0.1095(A - 0.9703), & A > 37.5 \\ 0.5106(A - 20)^{0.7262}, & 20 \leq A \leq 37.5 \\ 0, & A < 20 \end{cases} \quad (7.67)$$

and

$$N = \frac{A - 6.9539}{2.285\Delta\omega}. \quad (7.68)$$

#### Example 4

Now, let us design an LPF using the  $I_1$ -cosh window with the same specifications as in Example 2.

$$\begin{aligned} \omega_p &= 0.2\pi \text{ rad/sample} & A_p &= 0.1 \text{ dB} \\ \omega_s &= 0.3\pi \text{ rad/sample} & A_s &= 43 \text{ dB} \end{aligned}$$

The passband ripple is smaller than the stopband ripple in this example. So we design the filter that meets the passband attenuation. Attenuation  $A$  can be obtained as

$$A = -20 \log \delta = 44.797 \text{ dB.}$$

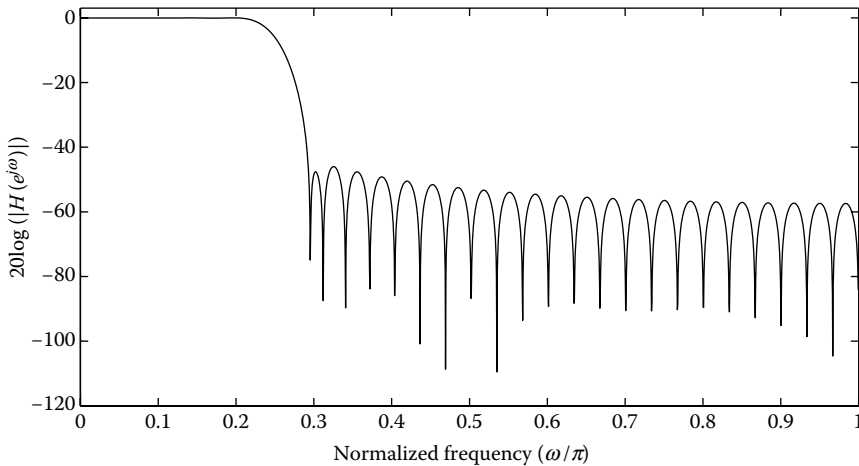
For this value of attenuation and the required transition bandwidth ( $0.1\pi$ ), we can compute the filter order  $N$  and shape parameter  $\gamma$  using Equations 7.68 and 7.67, respectively, as

$$N = \frac{44.797 - 6.9539}{2.285(0.1\pi)} = 53$$

and

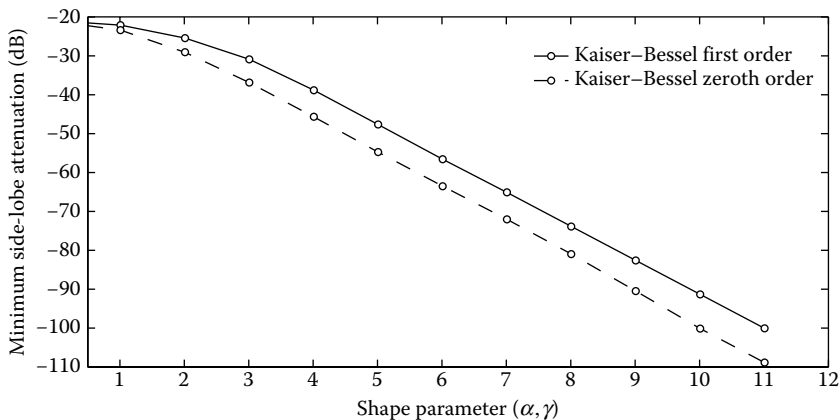
$$\gamma = 0.1095(44.797 - 0.9703) = 4.799.$$

Since the filter length is even, we design the LPF using the type 2 filter. From these values of  $N$  and  $\gamma$ , we can get the  $I_1$ -cosh window coefficients and use it to truncate the impulse response of the ideal LPF. The magnitude response of this filter is given in Figure 7.22 to verify whether it meets the required specifications. It is to be observed that the filter length



**FIGURE 7.22**  
 Magnitude response of lowpass filter designed using the  $I_1$ -cosh window.

required to design a filter for the given specifications is almost the same for both the  $I_1$ -cosh and the  $I_0$ -sinh windows. In almost all the cases, the filter designed with the  $I_0$ -sinh window outperforms the one designed using the  $I_1$ -cosh window. For the purpose of comparison of the zeroth and the first-order Kaiser–Bessel windows, the minimum side-lobe attenuation (Gibbs number) that can be attained for a given shape parameter is given in Figure 7.23.



**FIGURE 7.23**  
 Comparison of  $I_1$ -cosh and  $I_0$ -sinh filters.

### 7.8 Design of Differentiator by Impulse Response Truncation

In signal processing applications, we usually come across many situations where we need to differentiate a time signal to find the rate of change of a signal. The differentiator can be represented using the difference equation, which can be easily implemented by a general-purpose digital computer or a special-purpose digital hardware. The procedure for finding the differentiator coefficients for the hardware implementation is given in this section.

We now proceed with the frequency response of a discrete-time differentiator

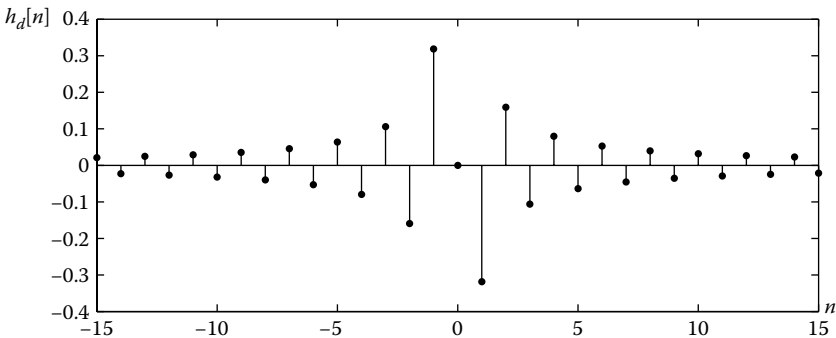
$$H_d(e^{j\omega}) = j\omega, \quad -\pi \leq \omega \leq \pi. \tag{7.69}$$

The impulse response of the differentiator was derived in Chapter 1 (see Section 1.3.2), which is reproduced below:

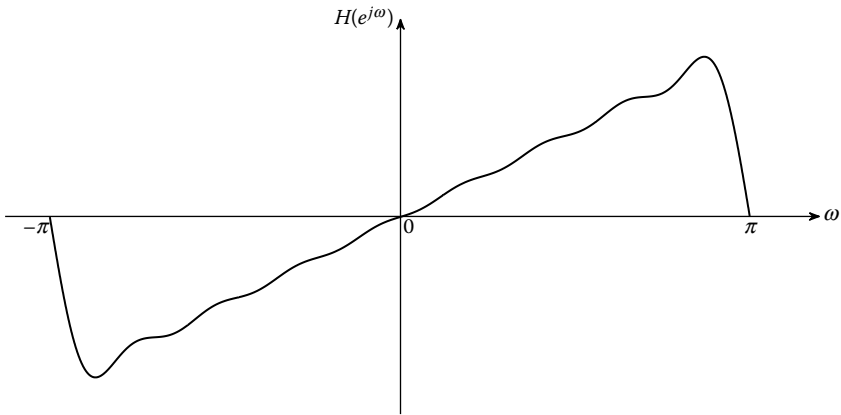
$$h_d[n] = \frac{\cos \pi n}{\pi n} - \frac{\sin \pi n}{\pi n^2}. \tag{7.70}$$

Figure 7.24 shows the impulse response of the differentiator. Clearly, the impulse response is anti-symmetric. We can see from Table 7.3 that the filters having antisymmetric impulse response can be designed with type 3 or type 4 FIR filters.

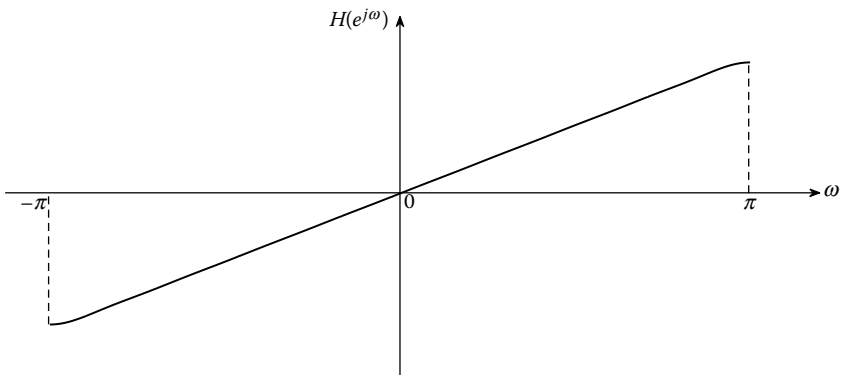
The design procedure of a differentiator is similar to the design of other filters. The impulse response is truncated by an appropriate window function to make it realizable. The differentiators designed with type 3 and type 4 FIR filters, using the rectangular window, are shown in Figures 7.25 and 7.26, respectively. From these figures, it is clear that type 4 gives a closer approximation to the ideal differentiator. This is because type 3 filter has a



**FIGURE 7.24**  
Impulse response of differentiator.



**FIGURE 7.25**  
Frequency response of a differentiator designed with type 3 FIR filter.



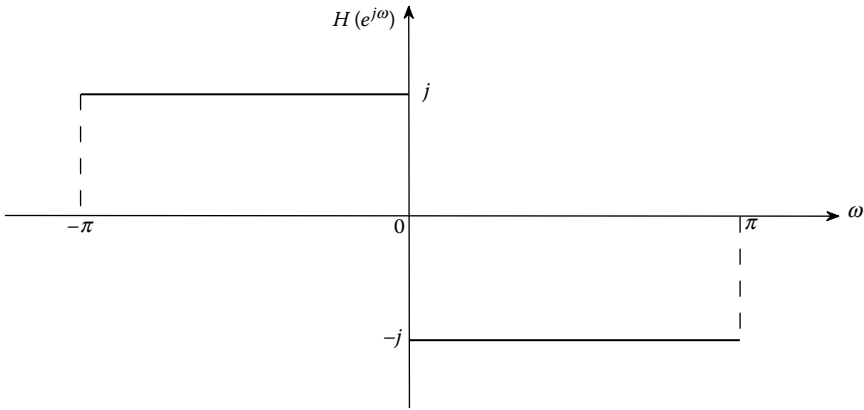
**FIGURE 7.26**  
Frequency response of differentiator designed with type 4 FIR filter.

constrained zero at  $\omega = \pi$ , which is responsible for bringing down the impulse response to zero at both the edges.

We also note that we get a better approximation of the differentiator using type 4 at the expense of (integer  $+\frac{1}{2}$ ) group delay. This half-delay can be compensated at other parts of the system.

### 7.9 Design of Hilbert Transformer Using Impulse Response Truncation

Quite often, a Hilbert transformer is used in communication systems to eliminate either the negative or the positive frequency components from the



**FIGURE 7.27**  
Frequency response of ideal Hilbert transform.

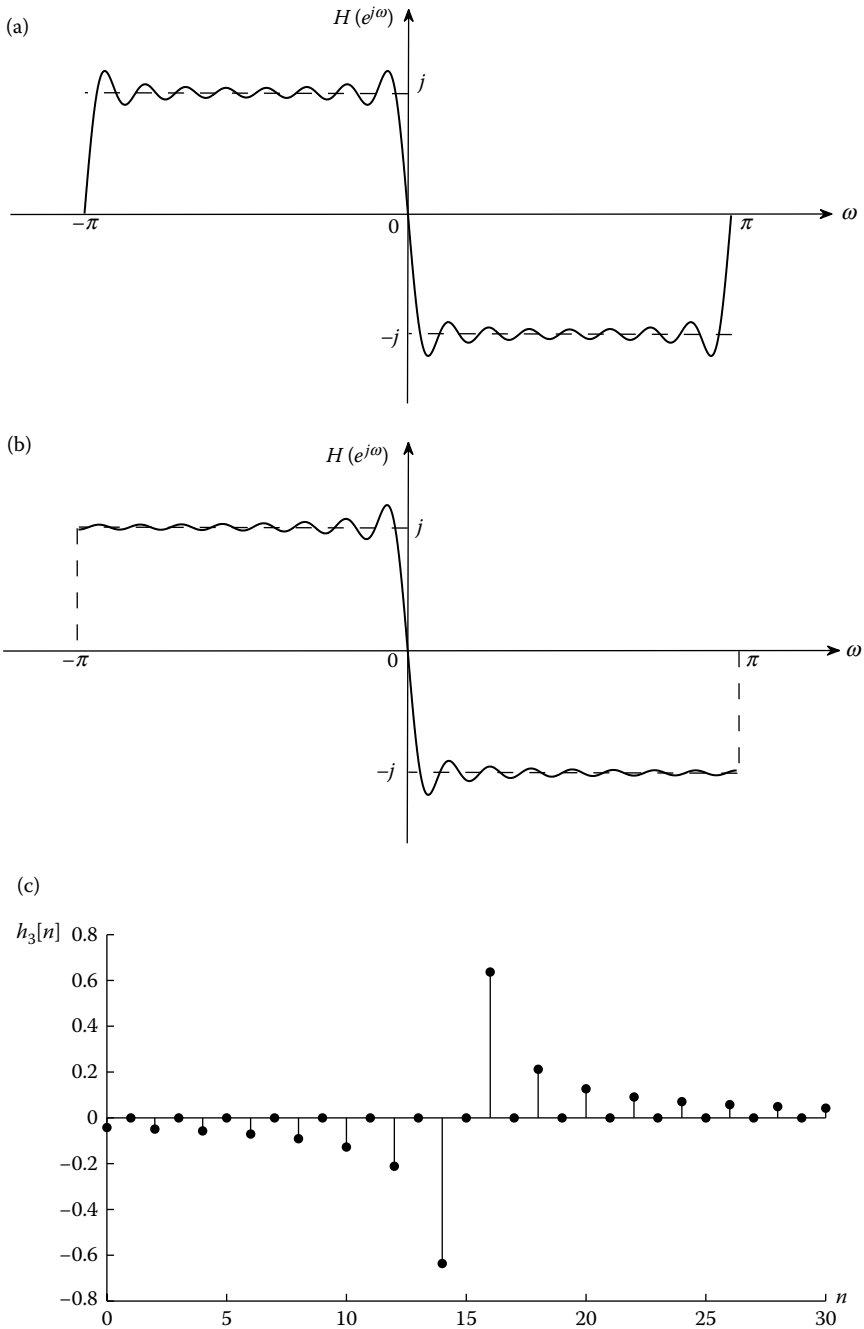
real signals.\* It can also be used as an ideal 90° phase shifter. It only alters the phase response of the signal by keeping the magnitude response unchanged. The frequency response of the Hilbert transformer (depicted in Figure 7.27) can be defined as follows:

$$H(e^{j\omega}) = \begin{cases} j, & \omega < 0 \\ 0, & \omega = 0 \\ -j, & \omega > 0. \end{cases} \tag{7.71}$$

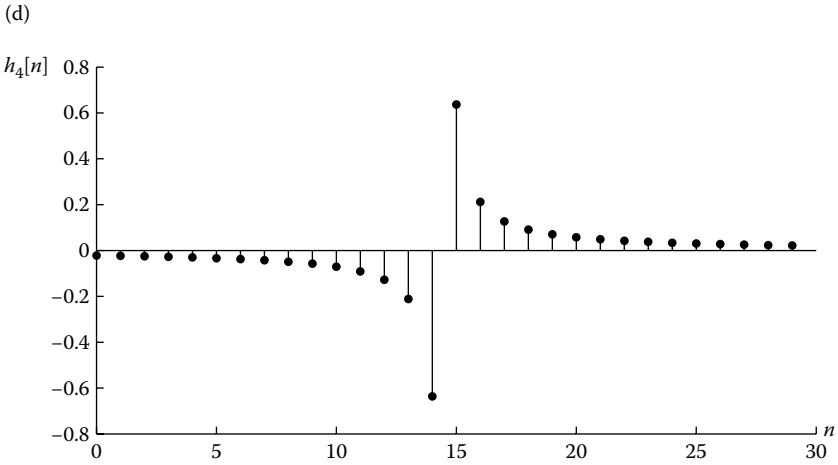
The impulse response of the Hilbert transformer  $h[n]$  can now be obtained by taking the inverse DTFT of  $H(e^{j\omega})$ , as described below:

$$\begin{aligned} h[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^0 j e^{j\omega n} d\omega + \frac{1}{2\pi} \int_0^{\pi} -j e^{j\omega n} d\omega \\ &= \frac{1}{2\pi} \left[ \frac{j e^{j\omega n}}{jn} \right]_{-\pi}^0 - \frac{1}{2\pi} \left[ \frac{j e^{j\omega n}}{jn} \right]_0^{\pi} \\ &= \frac{1}{2\pi} \left[ \frac{2}{n} - \frac{2 \cos(\pi n)}{n} \right] = \frac{1 - \cos(\pi n)}{\pi n} \\ &= \begin{cases} \frac{2}{\pi n}, & n = \text{odd} \\ 0, & n = \text{even.} \end{cases} \end{aligned} \tag{7.72}$$

\* Signals having only positive or negative frequencies are called analytic signals.



**FIGURE 7.28** Frequency response of the Hilbert transformer designed using type 3 and type 4. (a) Type 3 frequency response. (b) Type 4 frequency response. (c) Type 3 impulse response.



**FIGURE 7.28** (Continued). Frequency response of the Hilbert transformer designed using type 3 and type 4. (d) Type 4 impulse response.

Observe that the impulse response is antisymmetric in nature, leaving us with only type 3 or type 4 FIR filters to design a Hilbert transformer. The design procedure is straightforward. Once we obtain the impulse response, it is truncated to a suitable length using a window. Figures 7.28(a) and (b) display the frequency response plots of the Hilbert transformer designed with type 3 and type 4 filters, with lengths  $N = 31$  and  $N = 30$ , respectively. The impulse response given in Equation 7.72 is for odd lengths. Refer to Section 5.2.28 to find the impulse response of the Hilbert transformer for even lengths. In Figures 7.28(c) and (d), the corresponding impulse responses are shown. Similar to the case of a differentiator design discussed earlier, type 4 seems to be a better approximation in the case of the Hilbert transformer, when compared to type 3. However, in practice, type 3 filter requires lesser number of computations and can generate the output at twice the speed, when compared to type 4 filter. This is evident once we observe the impulse responses of the type 3 and type 4 Hilbert transformer. Type 3 impulse response has zeros for all even values of  $n$  (see Figure 7.28(a)), whereas to design a Hilbert transformer with type 4 impulse response, it is sampled at  $(\text{integer} + \frac{1}{2})$  values making all the coefficients nonzero, as shown in Figure 7.28(b). The number of multiplications required reduces to  $(N/2)$  in type 3 design, making it a better choice from the implementation point of view.



**References**

1. A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd Edn., Upper Saddle River, NJ, 2010.

2. L.C. Ludeman, *Fundamentals of Digital Signal Processing*, John Wiley and Sons, New York, NY, 1986.
3. E.C. Ifeachor and B.W. Jervis, *Digital Signal Processing: A Practical Approach*, Prentice-Hall, 2nd Edn., Harlow, England, 2001.
4. R.J. Webster, On qualifying windows for FIR filter design, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 31, pp. 237–240, February 1983.
5. A. Antonio, *Digital Filters: Analysis and Design*, McGraw Hill, 1st Edn., New York, 1979.
6. L.R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, NJ, 1975.
7. J.F. Kaiser, Nonrecursive digital filter design using the  $I_0$ -sinh window function, *IEEE International Symposium on Circuits and Systems*, pp. 20–23, April 1974.
8. S.J. Orfanidis, *Introduction to signal processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1996.
9. K.M.M. Prabhu and K. Bhoopathy Bagan, FIR digital filter design with variable parameter window families, *Journal of the Institution of Electronic and Radio Engineers*, Vol. 58, pp. 295–298, December 1988.





# 8

## *Application of Windows in Spectral Analysis*

Every signal obtained from nature can be considered as a realization of a random process, but in practice, we will only have finite number of samples of this process. Therefore, to characterize this process, we need to estimate its statistical parameters from the available data samples. Power spectral estimation is a process of estimating the different frequency components contained in a signal. Spectral estimation methods can be broadly classified into two categories: nonparametric estimators and parametric estimators. Since window functions are only applicable to nonparametric methods, we will only concentrate on this category of spectral analysis.

The performance of any power spectral estimator is evaluated based on several goodness measures that are outlined below:

- *Bias*: It is defined as the difference between the mean or expected value  $\varepsilon[\hat{x}]$  of the estimates and its true mean:

$$\text{bias}(\hat{X}(\omega)) = \varepsilon\{\hat{X}(\omega) - X(\omega)\}. \quad (8.1)$$

If the bias is zero, then it is called an unbiased estimator.

- *Variance*: It denotes the spread of the power spectral density (PSD) about its mean value. It is expressed as

$$\text{var}(\hat{X}(\omega)) = \varepsilon\{(\hat{X}(\omega) - \varepsilon\{\hat{X}(\omega)\})^2\}. \quad (8.2)$$

A good estimator should have a small variance, in addition to having a small bias.

- *Mean square error (MSE)*: It is a measure that combines the bias and variance associated with the estimator. It is defined as

$$\begin{aligned} \text{MSE}(\hat{X}(\omega)) &= \varepsilon\{(\hat{X}(\omega) - X(\omega))^2\} \\ &= \varepsilon\{(\hat{X}(\omega) - \varepsilon\{\hat{X}(\omega)\})^2\} + (\varepsilon\{\hat{X}(\omega) - X(\omega)\})^2 \\ &= \text{var}(\hat{X}(\omega)) + (\text{bias}(\hat{X}(\omega)))^2. \end{aligned} \quad (8.3)$$

- *Consistency*: If the bias and variance both tend to zero as the number of observations becomes large, then the estimator is said to be consistent.
- *Resolution*: This corresponds to the ability of an estimator to provide fine details of the random process. For example, suppose the PSD of the random process has two peaks with the same amplitude at frequencies  $\omega_1$  and  $\omega_2$ . Then, the resolution of the estimator is measured by the minimum separation between  $\omega_1$  and  $\omega_2$  for which the estimator still reproduces two distinct peaks at those frequencies.

Spectral estimation aims at finding the PSD or spectral information of a signal on an average basis, such that the estimate has less bias and variance, apart from having the required frequency resolution. In general, a signal can be obtained as an output sequence of a system where the input can be an impulse train (as in the case of voiced speech) or white noise input for cases such as seismic signals (reflections from the layers of the earth). Signals are generally associated with noise, which inevitably introduces variance. Any attempt to achieve more frequency resolution than a limit also increases the variance as it picks up undesired noise. It is very difficult to achieve the twin goals of reducing both bias and variance.

---

## 8.1 Nonparametric Methods

The nonparametric methods are based on Fourier transform techniques to find the PSD estimate [1]. There are two basic PSD estimators:

- Periodogram methods*: They are based on the direct transformation of the data, followed by averaging.
- Correlogram methods*: These methods first formulate the correlation estimates from the given data.

### 8.1.1 Periodogram PSD Estimator

The introduction to the periodogram method was given in Chapter 5. Here, we will discuss the concept in detail. The power spectral density (PSD) is defined as in [2] by

$$P_{xx}(e^{j\omega}) = \lim_{N \rightarrow \infty} \varepsilon \left\{ \frac{1}{2N+1} \left| \sum_{n=-N}^N x[n] e^{-j\omega n} \right|^2 \right\}. \quad (8.4)$$

where  $2N + 1$  is the length of the signal window. We can define a periodogram in two ways. Taking the Fourier transform of the autocorrelation estimate results in an estimate of the PSD, which is known as the periodogram:

$$P_{PER}(e^{j\omega}) = \sum_{k=-(N-1)}^{N-1} r_{xx}[k]e^{-jk\omega}. \quad (8.5)$$

The autocorrelation estimate  $r_{xx}[k]$  is the autocorrelation of the rectangular windowed version of the sequence  $x[n]$  of length  $N$ . Let us define this windowed sequence as  $x_R[n]$ . Then  $r_{xx}[k]$  is defined as

$$r_{xx}[k] = \frac{1}{N} x_R[k] * x_R^*[-k]. \quad (8.6)$$

Taking the Fourier transform of Equation 8.6, we get

$$\begin{aligned} P_{PER}(e^{j\omega}) &= \frac{1}{N} X_R(e^{j\omega}) X_R^*(e^{j\omega}) = \frac{1}{N} |X_R(e^{j\omega})|^2 \\ &= \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \right|^2, \end{aligned} \quad (8.7)$$

where  $X_R(e^{j\omega})$  is the DFT of  $x_R[n]$ , which is in turn equal to the  $N$  point DFT of  $x[n]$ , thus giving an alternate expression for periodogram. Now let us examine the value of the estimate at a particular frequency  $\omega_0$ . Then, the periodogram can be expressed as

$$P_{PER}(e^{j\omega_0}) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\omega_0 n} \right|^2. \quad (8.8)$$

Therefore, Equation 8.8 represents the power of the frequency component  $\omega_0$  in that signal. This is just an inner product of two data sequences. Hence, Equation 8.8 can be expressed as a convolution operation with a filter which has an impulse response  $h[n]$ . Let the impulse response of the filter be

$$h[n] = \begin{cases} \frac{1}{N} e^{j\omega_0 n}, & \text{for } n = -(N-1), -(N-2), \dots, -1, 0 \\ 0, & \text{otherwise.} \end{cases} \quad (8.9)$$

The sequence  $h[n]$  is time-reversed and hence, after convolution, becomes the inner product expression. This operation can be expressed as

$$P_{PER}(e^{j\omega_0}) = N \left| \sum_{k=0}^{N-1} h[n-k] x[k] \right|^2. \quad (8.10)$$

The DTFT of  $h[n]$  is given as

$$\begin{aligned}
 H(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n} = \sum_{n=-(N-1)}^0 \frac{1}{N} e^{j\omega_0 n} e^{-j\omega n} \\
 &= \frac{e^{j(\omega-\omega_0)(N-1)}}{N} \left[ \frac{1 - e^{-j(\omega-\omega_0)N}}{1 - e^{-j(\omega-\omega_0)}} \right] \\
 &= \frac{1}{N} e^{j(\omega-\omega_0)(N-1)/2} \frac{\sin((\omega - \omega_0)N/2)}{\sin((\omega - \omega_0)/2)}. \tag{8.11}
 \end{aligned}$$

The scale factor  $N$  is necessary to account for the filter bandwidth. To obtain the spectral density, the power is divided by the length of the data, which gives the PSD estimate. By examining Equation 8.11, we infer that it is the frequency response of a shifted rectangular window. Hence, the convolution operation is just a set of bandpass filtering operations to give the amount of power in that particular frequency band. Therefore, we can say that the periodogram method generates the PSD estimate of the given signal.

*Bias of periodogram:* The expected value of the periodogram can now be calculated as in Ref. [3]:

$$\begin{aligned}
 \varepsilon\{P_{PER}(e^{j\omega})\} &= \varepsilon \left\{ \sum_{k=-(N-1)}^{N-1} r_{xx}[k]e^{-j\omega k} \right\} \\
 &= \sum_{k=-(N-1)}^{N-1} \varepsilon\{r_{xx}[k]\}e^{-j\omega k}, \tag{8.12}
 \end{aligned}$$

where

$$\begin{aligned}
 \varepsilon\{r_{xx}[k]\} &= \frac{1}{N} \sum_{n=0}^{N-1-k} \varepsilon\{x[n+k]x^*[n]\} \\
 &= \left( \frac{N-k}{N} \right) r_x[k]. \tag{8.13}
 \end{aligned}$$

In Equation 8.13,  $r_x[k]$  represents the unbiased estimate of the true autocorrelation of the signal  $x[n]$ . Therefore, the autocorrelation estimate of the rectangular windowed signal is weighted with a Bartlett (triangular) window. Thus, Equation 8.13 can be written as

$$\varepsilon\{r_{xx}[k]\} = f_B[k]r_x[k], \tag{8.14}$$

where

$$f_B[k] = \begin{cases} \frac{N-|k|}{N}, & |k| \leq N-1 \\ 0, & \text{otherwise.} \end{cases} \tag{8.15}$$

Equation 8.15 represents a triangular or Bartlett window in the time-domain. Substituting Equation 8.14 into Equation 8.12, we obtain

$$\varepsilon\{P_{PER}(e^{j\omega})\} = \sum_{k=-\infty}^{\infty} r_{xx}[k]f_B[k]e^{-j\omega k}. \quad (8.16)$$

Equation 8.16 can be interpreted as a convolution operation, and using the properties of the Fourier transform we can represent it as

$$\varepsilon\{r_{xx}[k]\} = \frac{1}{2\pi}P_{xx}(e^{j\omega}) * F_B(e^{j\omega}), \quad (8.17)$$

where  $F_B(e^{j\omega})$  is the Fourier transform of the Bartlett window, represented as

$$F_B(e^{j\omega}) = \frac{1}{N} \left[ \frac{\sin(N\omega/2)}{\sin(\omega/2)} \right]^2. \quad (8.18)$$

Now, as  $N \rightarrow \infty$ , the sinc-squared pulse converges toward a Dirac delta function in the frequency-domain. Hence, the expected value of the periodogram estimate approaches the true PSD, and thereby the bias tends to zero, as given below:

$$\lim_{N \rightarrow \infty} \varepsilon\{P_{PER}(e^{j\omega})\} = P_{xx}(e^{j\omega}). \quad (8.19)$$

*Variance of periodogram:* The variance of the periodogram method [3] can be calculated as follows:

$$\begin{aligned} P_{PER}(e^{j\omega}) &= \frac{1}{N} \left| \sum_{k=0}^{N-1} x[k]e^{-j\omega k} \right|^2 \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x[k]x^*[l]e^{-j(k-l)\omega}. \end{aligned} \quad (8.20)$$

The variance can be determined from the covariance expression given by

$$\text{cov}\{P_{PER}(e^{j\omega_1})P_{PER}(e^{j\omega_2})\} = \varepsilon\{P_{PER}(e^{j\omega_1})P_{PER}(e^{j\omega_2})\} - \varepsilon\{P_{PER}(e^{j\omega_1})\}\varepsilon\{P_{PER}(e^{j\omega_2})\}. \quad (8.21)$$

The second moment of the periodogram can be computed as

$$\varepsilon\{P_{PER}(e^{j\omega_1})P_{PER}(e^{j\omega_2})\} = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \varepsilon\{x[k]x^*[l]x[m]x^*[n]\}e^{-j(k-l)\omega_1}e^{-j(m-n)\omega_2}. \quad (8.22)$$

Now, assuming  $x[n]$  as a Gaussian random process, we can make use of the *moment factorizing* theorem, which is given by

$$\varepsilon\{x[k]x^*[l]x[m]x^*[n]\} = \varepsilon\{x[k]x^*[l]\}\varepsilon\{x[m]x^*[n]\} + \varepsilon\{x[k]x^*[n]\}\varepsilon\{x[m]x^*[l]\}. \quad (8.23)$$

Thus, using Equation 8.23, we can simplify Equation 8.22 as

$$\varepsilon\{P_{PER}(e^{j\omega_1})P_{PER}(e^{j\omega_2})\} = \sigma_x^4 \left[ 1 + \left( \frac{\sin(N(\omega_1 - \omega_2)/2)}{N \sin((\omega_1 - \omega_2)/2)} \right)^2 \right], \quad (8.24)$$

where  $\sigma_x^2$  is the variance of  $x[n]$ , when assumed that  $x[n]$  is a Gaussian random process. For a Gaussian signal  $x[n]$ , the variance is  $\varepsilon\{P_{PER}(e^{j\omega})\} = \sigma_x^2$ . Therefore, the covariance of the periodogram is given by

$$\text{cov}\{P_{PER}(e^{j\omega_1})P_{PER}(e^{j\omega_2})\} = \sigma_x^4 \left[ \frac{\sin(N(\omega_1 - \omega_2)/2)}{N \sin((\omega_1 - \omega_2)/2)} \right]^2. \quad (8.25)$$

Finally, the variance of the periodogram estimate is obtained from Equation 8.25 by setting  $\omega_1 = \omega_2$ .

$$\text{var}\{P_{PER}(e^{j\omega})\} = \sigma_x^4. \quad (8.26)$$

However, as  $N \rightarrow \infty$ , the variance does not go to zero. For the Gaussian process

$$\begin{aligned} \lim_{N \rightarrow \infty} \varepsilon\{P_{PER}(e^{j\omega})\} &= P_{xx}(e^{j\omega}) = \sigma_x^2 \\ \implies \text{var}\{P_{PER}(e^{j\omega})\} &= P_{xx}^2(e^{j\omega}). \end{aligned} \quad (8.27)$$

*Resolution of periodogram estimate:* For a fixed value of  $N$ , there is a limit on the proximity of the two sinusoids such that they can be resolved as two distinct peaks. This is usually given by the 6 dB bandwidth of the window. Hence, for periodogram PSD estimate, the resolution is the 6 dB bandwidth of the Bartlett window [3]:

$$\text{Res}\{P_{PER}(e^{j\omega})\} = 0.89 \left( \frac{2\pi}{N} \right). \quad (8.28)$$

The technique described above was the original unmodified periodogram PSD estimate [2]. However, this method produces statistically inconsistent PSD estimates, that is, as the length of the data increases ( $N \rightarrow \infty$ ), the mean converges to the true PSD, but the variance does not tend to zero. This problem arises because of the fact that the expectation operation was ignored in the PSD computation. To overcome this, it is necessary to average the PSD of many outputs to cause the variance to decrease.

In many cases of practical interest, the data consist of sinusoidal inputs or white noise-like inputs. In such a case, the application of an appropriate window is essential. The effect of different windows on spectral estimation will be discussed in the following sections. Presently, let us discuss in detail the effect of the length of the sequence on the bias and variance of the periodogram. Let us consider a signal having three sinusoidal components: two of which are closely spaced ones and the third component is distant from them, described as follows:

$$x[n] = \sin[0.15625\pi n] + 0.01 \sin[0.21875\pi n] + 0.00316 \sin[0.46875\pi n]. \quad (8.29)$$

To the signal  $x[n]$ , white noise with a variance of 0.002 is added.

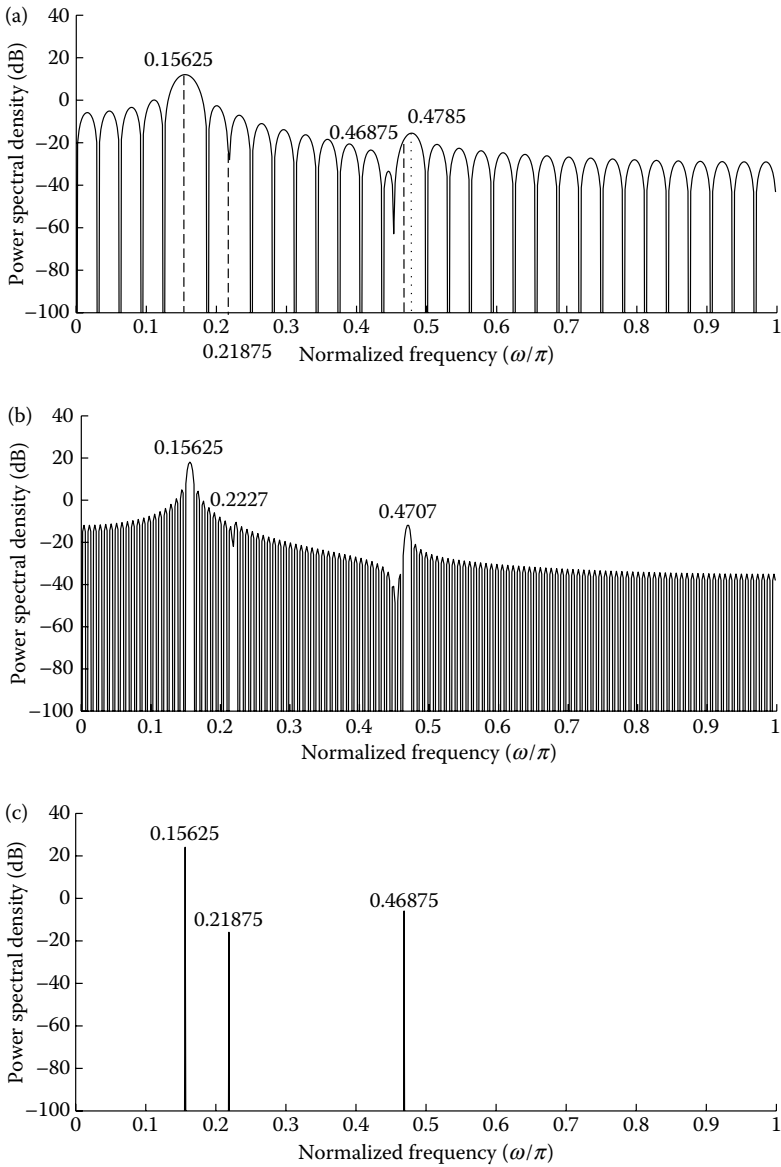
From Figure 8.1, we can say that the bias is reduced as the value of window length  $N$  is increased, giving us an accurate spectral estimate, but the variance does not reduce. Figures 8.2 and 8.3 show the superimposed figures of 50 iterations of the PSD estimate using the periodogram method. Here, we can see that the width of the dark region (in Figures 8.2 and 8.3) remains constant with an increase in  $N$ . This dark region corresponds to the spread of the periodogram PSD estimate curves. Hence, we can conclude that the periodogram is not a consistent estimation of the PSD.

### 8.1.2 Modified Periodogram PSD Estimator

The periodogram estimate of a random process that is windowed with any general window (other than the rectangular window) is called modified periodogram. This method uses a window to smooth a single periodogram to obtain a better spectral estimate [2]. The operation of windowing in the time-domain is equivalent to convolution in the frequency-domain. We note that all the windows presented in the previous chapters have a lowpass filtering effect. Thus, it will result a smoothing of the periodogram, in based on the type of window selected [4]. This smoothing effect of the window reduces the variance of the PSD estimate. At the same time, it improves the resolution of the PSD estimate. This is because even in the absence of a window in the periodogram PSD estimator, a rectangular window was present implicitly. The rectangular window has a more narrow main-lobe width, but has higher side-lobe levels, which causes the main lobes of the weaker signal components to be masked by the side lobes of a stronger signal component. In this case, we can use an effective window function,  $f[n]$ , which has lower side-lobe levels, such as Hann, Hamming, Blackman, or Kaiser window. The effect of different windows on a periodogram PSD estimator is shown in Figures 8.4 and 8.5. The modified periodogram estimate is given by

$$P_M(e^{j\omega}) = \frac{1}{NU} \left| \sum_{n=-\infty}^{\infty} x[n]f[n]e^{-j\omega n} \right|^2, \quad (8.30)$$

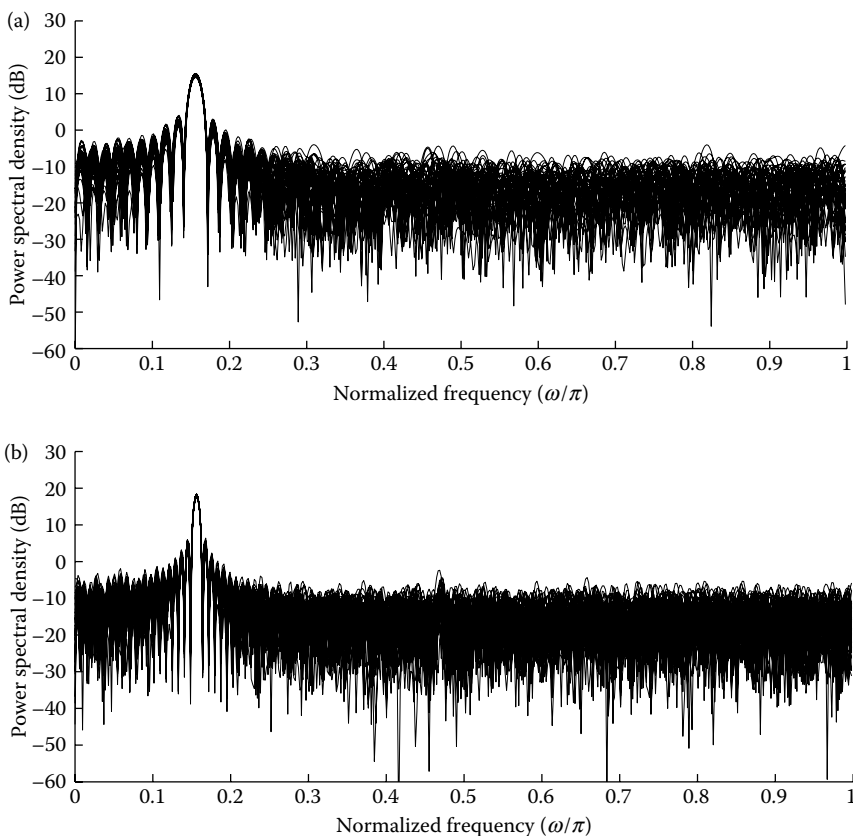




**FIGURE 8.1** Periodogram estimate for increasing  $N$  (no noise added). (a)  $N = 64$ . (b)  $N = 256$ . (c)  $N = 1024$ .

where  $N$  is the length of the window and  $U$  (which is defined in Equation 8.31) is a constant that makes the modified periodogram asymptotically unbiased:

$$U = \frac{1}{N} \sum_{n=0}^{N-1} |f[n]|^2. \tag{8.31}$$



**FIGURE 8.2** Periodogram for several values of  $N$  (with additive white noise). (a)  $N = 128$ . (b)  $N = 256$ .

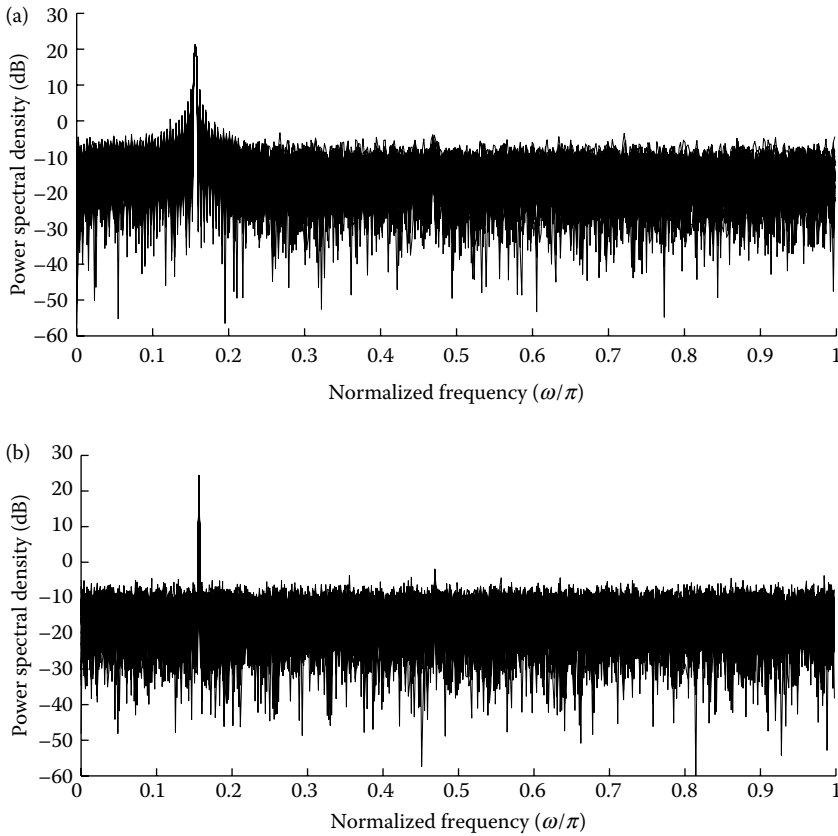
*Bias of modified periodogram:* Using the procedure for calculating the bias of the periodogram estimate, we can get the expected value of the modified periodogram as

$$\varepsilon\{P_M(e^{j\omega})\} = \frac{1}{2\pi NU} P_{xx}(e^{j\omega}) * |F(e^{j\omega})|^2. \tag{8.32}$$

As  $N \rightarrow \infty$ , the term  $\frac{|F(e^{j\omega})|}{NU}$  in Equation 8.32 will converge to an impulse function. This will result in  $\varepsilon\{P_M(e^{j\omega})\}$  approaching close to  $P_{xx}(e^{j\omega})$ , which in turn causes the bias to tend to zero.

*Variance of modified periodogram:* The variance of the modified periodogram does not change much, since it is just a periodogram of a windowed sequence. Hence

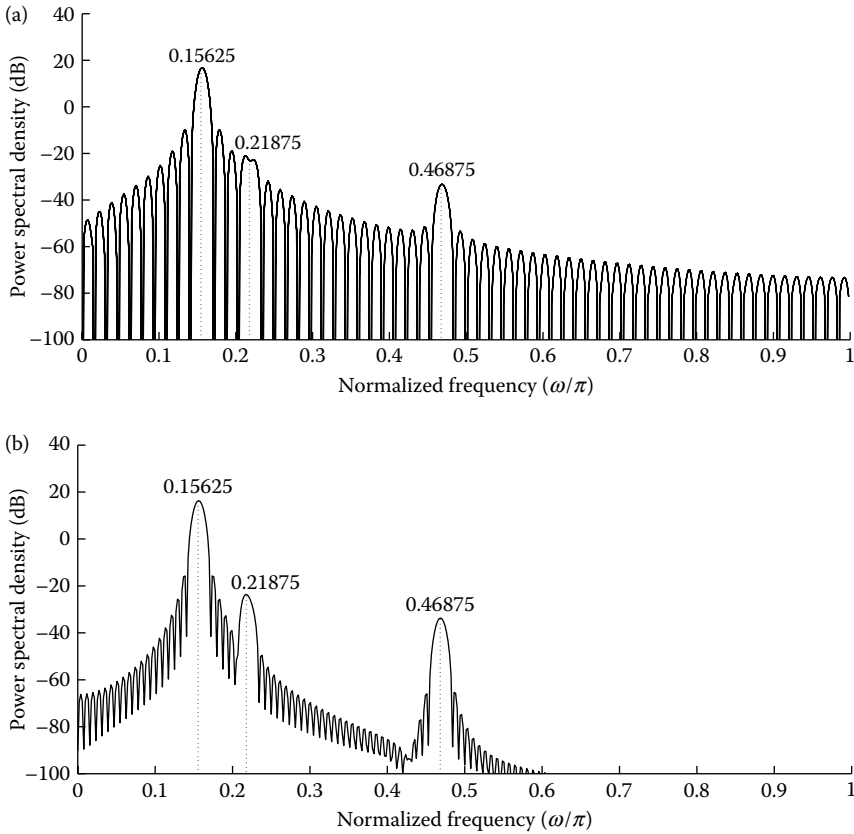
$$\text{var}\{P_M(e^{j\omega})\} \approx P_{xx}^2(e^{j\omega}). \tag{8.33}$$

**FIGURE 8.3**

Periodogram for several values of  $N$  (with additive white noise). (a)  $N = 512$ . (b)  $N = 1024$ .

*Resolution of modified periodogram:* The spectral resolution of the periodogram is the 6 dB bandwidth of the lag window, which will be applied onto the autocorrelation function (ACF). The resolution can be defined as the 3 dB bandwidth of the data window since it transforms into the 6 dB bandwidth of the lag window. The 3 dB as well as the 6 dB bandwidths of all common windows were already presented in Chapter 5 (see Table 5.3).

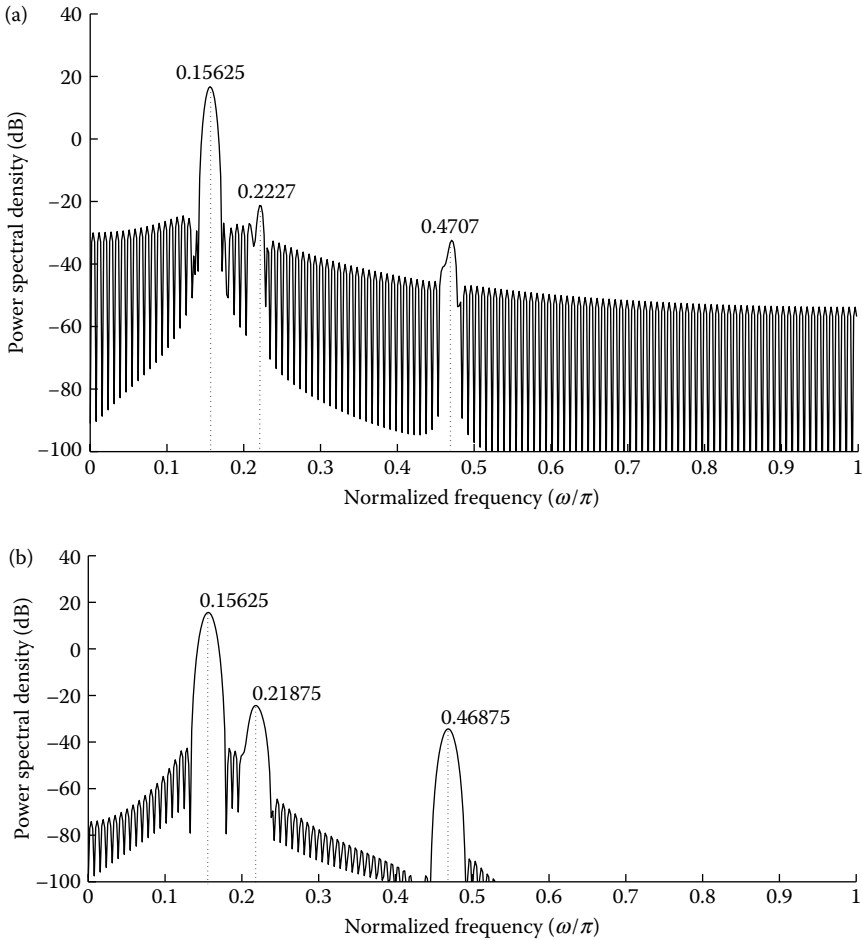
Figures 8.4 and 8.5 display the modified periodogram PSD estimates using different windows. The signal contains three sinusoidal components: one strong signal component, next a weak signal component, and then a third much weaker component that is farther away from the other two components, as given in Equation 8.29. Here, the length of the sequence  $N$  is fixed as 256. In Figure 8.2(b), we found that the periodogram with  $N = 256$  was unable to resolve the nearby weak signal components. Also, the estimation of the far-off weaker signal component was not accurate. However, with the aid of windowing, these components can be resolved better. In Figure 8.4(a),

**FIGURE 8.4**

Modified periodogram estimate. (a) Using Bartlett window. (b) Using Hann window.

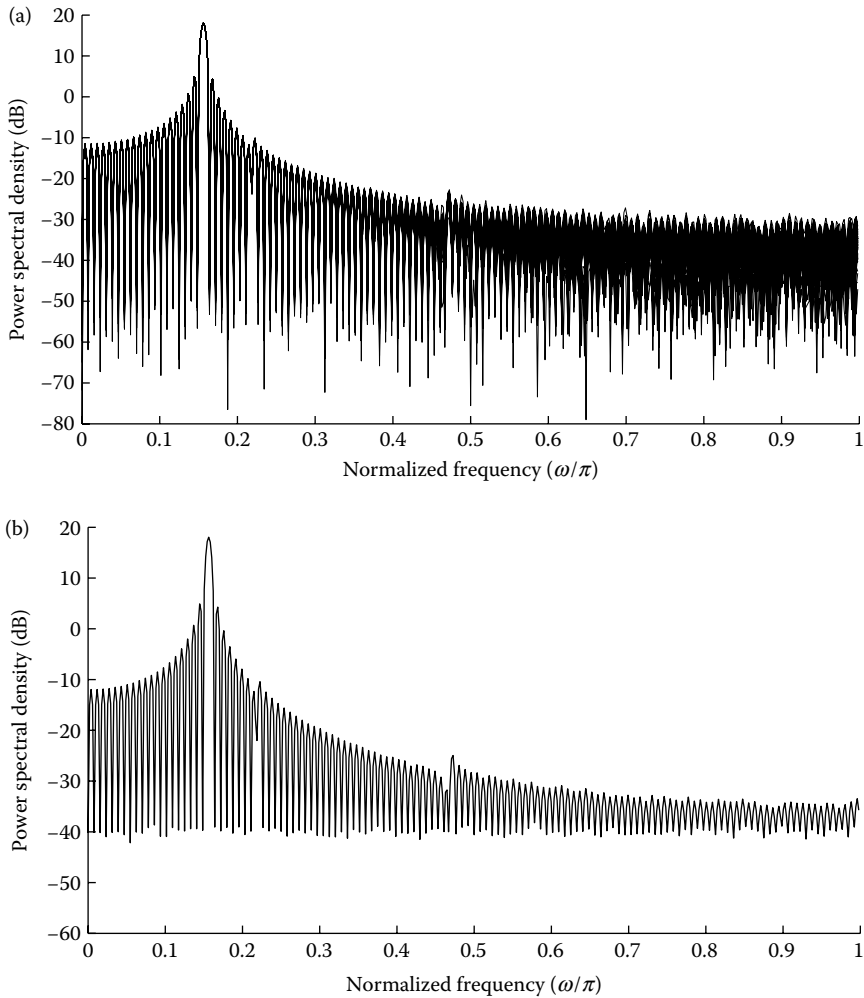
the Bartlett window was unable to resolve the nearby weak signal component, but surprisingly, it could resolve the far-off weaker component better. This is due to the faster decay rate of the side lobes of the Bartlett window. In the case of Hann and Hamming windows, both the weaker components could be resolved properly for this example (see Figures 8.4(b) and 8.5(a)). The Blackman window also resolves all the spectral components in this case (see Figure 8.5(b)), but due to the larger main-lobe width, the spectral resolution is less than that obtained using Hann and Hamming windows. The application of Hann, Hamming and Blackman windows in spectral estimation were already discussed in Chapter 5.

The modified periodogram is still not a consistent estimator, since the variance of the estimator does not go to zero as  $N \rightarrow \infty$ . Thus, the advantage of windowing is to provide a trade-off between resolution and the spectral masking provided by the side lobes of the window functions.

**FIGURE 8.5**

Modified periodogram estimate (continued). (a) Using Hamming window. (b) Using Blackman window.

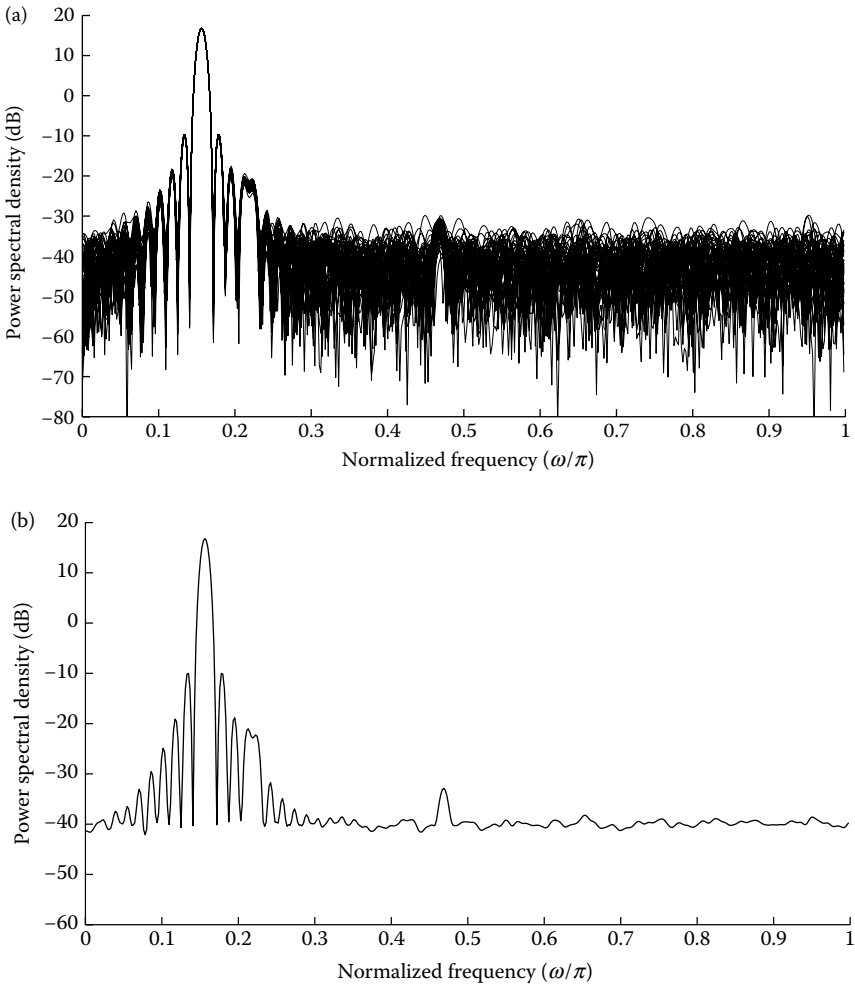
Figures 8.6 through 8.10 show the effect of noise on the modified periodogram estimator. In all the examples illustrated in these figures, we have considered white noise with a variance of 0.01, which is  $-40$  dB in magnitude. In Figures 8.6 through 8.10, each has two subplots: one showing the overlaid plots of 50 iterations of the modified periodogram method on the signal (with random white noise added) and the second subplot showing the average of all the 50 plots. From these plots, we can conclude that if the noise level is close to the PSLL of the window used, then the effect of noise on spectral estimation will be greater. Thus, all the windows that have PSLL less than this value will give poorer estimates. Hence, in this example, the performance of rectangular, Bartlett, and Hann windows (in the presence of white noise) is poor when compared to the other windows, since their PSLL is close to  $-40$  dB.

**FIGURE 8.6**

Effect of noise on modified periodogram using rectangular window. (a) Overlaid plots. (b) Averaged plots.

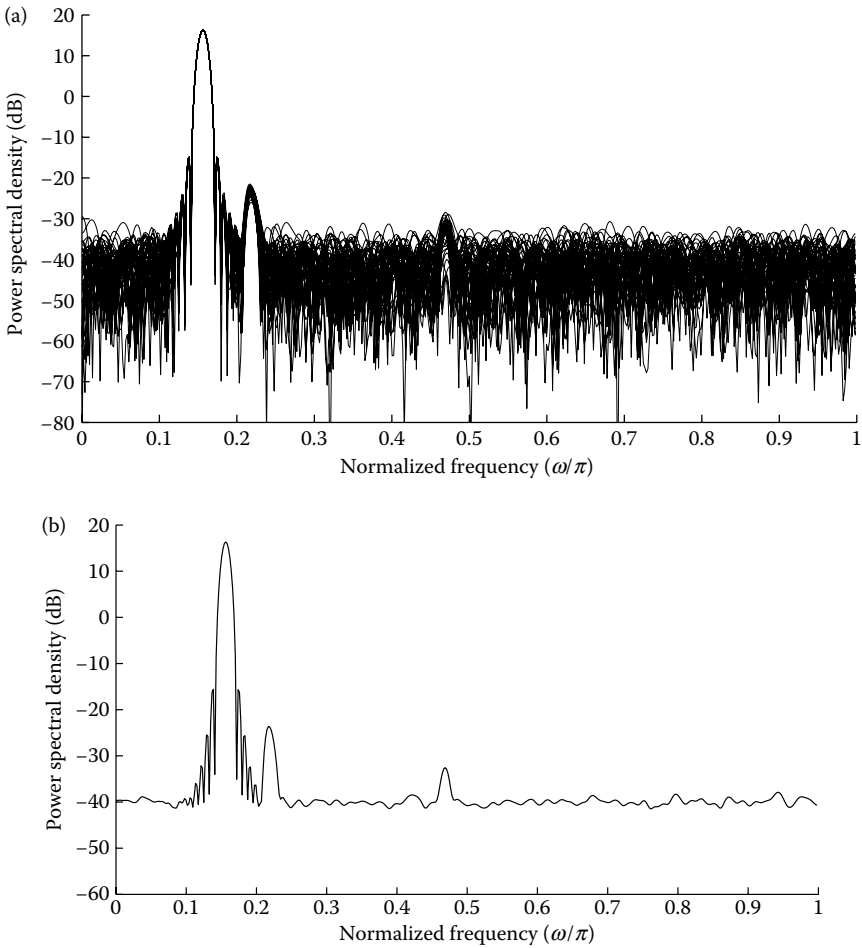
### 8.1.3 Spectral Analysis Using Kaiser–Bessel Window

In the previous sub-section, we have used only the common fixed parameter windows. For such windows, we are unable to fix the required resolution and frequency leakage simultaneously. When we try to improve the resolution using a window with a narrow main lobe, we will have more spectral leakage due to the high side-lobe levels. Hence, we often prefer *variable parameter* windows, where both these parameters can be controlled simultaneously [5]. One among them is the Kaiser–Bessel window with a variable parameter  $\alpha$ .



**FIGURE 8.7** Effect of noise on modified periodogram using Bartlett window. (a) Overlaid plots. (b) Averaged plots.

In the recent times, the Kaiser window has become popular for FIR filter design. In the case of FIR filter design problem, a convolution of the frequency response of the window with a “brick-wall” ideal filter response is important. However, in spectral analysis, the frequency response of the window is more vital. Hence, the formula used for calculating the Kaiser window parameters (as used for FIR filter design) are not appropriate for the spectral analysis applications. Kaiser and Schafer [6] have further developed simple design formulae that facilitate the usage of the Kaiser window for spectral analysis. These equations are used in computing accurate values for the window length



**FIGURE 8.8** Effect of noise on modified periodogram using Hann window. (a) Overlaid plots. (b) Averaged plots.

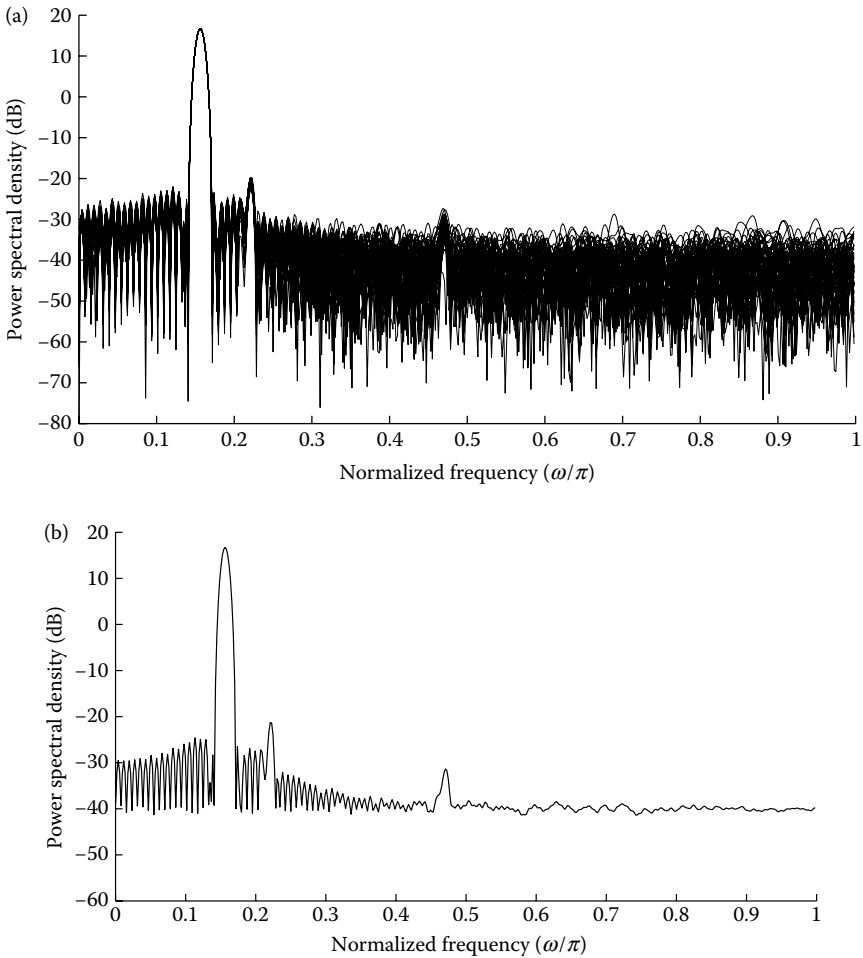
$N$  and the shape factor  $\alpha$ , for a given side-lobe level  $R$  in dB, as well as the frequency resolution  $\Delta\omega$ . The expression for the length ( $N$ ) is given as

$$(N - 1) = c \left( \frac{2\pi}{\Delta\omega} \right), \tag{8.34}$$

where  $c$  is a factor that depends on the type of window. For the Kaiser window,  $c$  is given by

$$c = \frac{6(R + 12)}{155}. \tag{8.35}$$

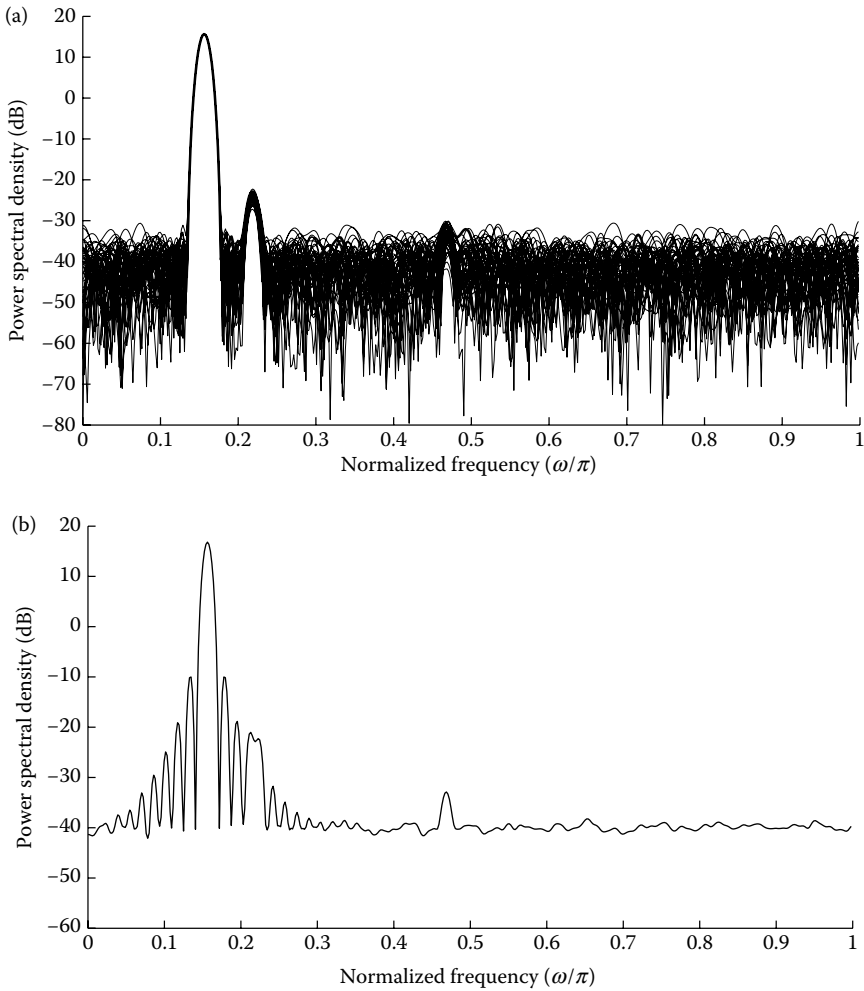




**FIGURE 8.9** Effect of noise on modified periodogram using Hamming window. (a) Overlaid plots. (b) Averaged plots.

The window shape parameter  $\alpha$  can be obtained in terms of  $R$  as follows:

$$\alpha = \begin{cases} 0, & R \leq 13.26 \\ 0.76609(R - 13.26)^{0.4} + 0.09834(R - 13.26), & 13.26 < R \leq 60 \\ 0.12438(R + 6.3), & 60 < R \leq 120. \end{cases} \quad (8.36)$$



**FIGURE 8.10** Effect of noise on modified periodogram using Blackman window. (a) Overlaid plots. (b) Averaged plots.

Incidentally, when  $\alpha = 0$ , the Kaiser–Bessel window becomes rectangular and 13.26 dB represents the attenuation of the first side-lobe level of the rectangular window. Once the window length  $N$  and  $\alpha$  are computed, we can generate the desired Kaiser window in the time-domain using the expression

$$f[n] = \frac{I_0(\alpha\sqrt{1 - (n - M)^2/M^2})}{I_0(\alpha)}, \quad n = 0, 1, \dots, N - 1 \quad (8.37)$$

where  $M = (N - 1)/2$ . The given data can then be windowed by  $f[n]$ . Let us now demonstrate the procedure by considering a signal

$$x[n] = \sin[0.4\pi n] + 0.00316 \sin[0.5\pi n] + 0.8 \sin[0.6\pi n]. \quad (8.38)$$

Since the magnitude of the second sinusoidal component is approximately about  $-50$  dB, we have to use a window with side-lobe levels less than  $-50$  dB. However, to be on the safe side, we can choose  $R = 70$  dB, and the resolution can be computed by the formula

$$\begin{aligned} \Delta\omega &= \frac{\omega_2 - \omega_1}{3} \\ &= \frac{0.6\pi - 0.5\pi}{3} = 0.0333\pi. \end{aligned}$$

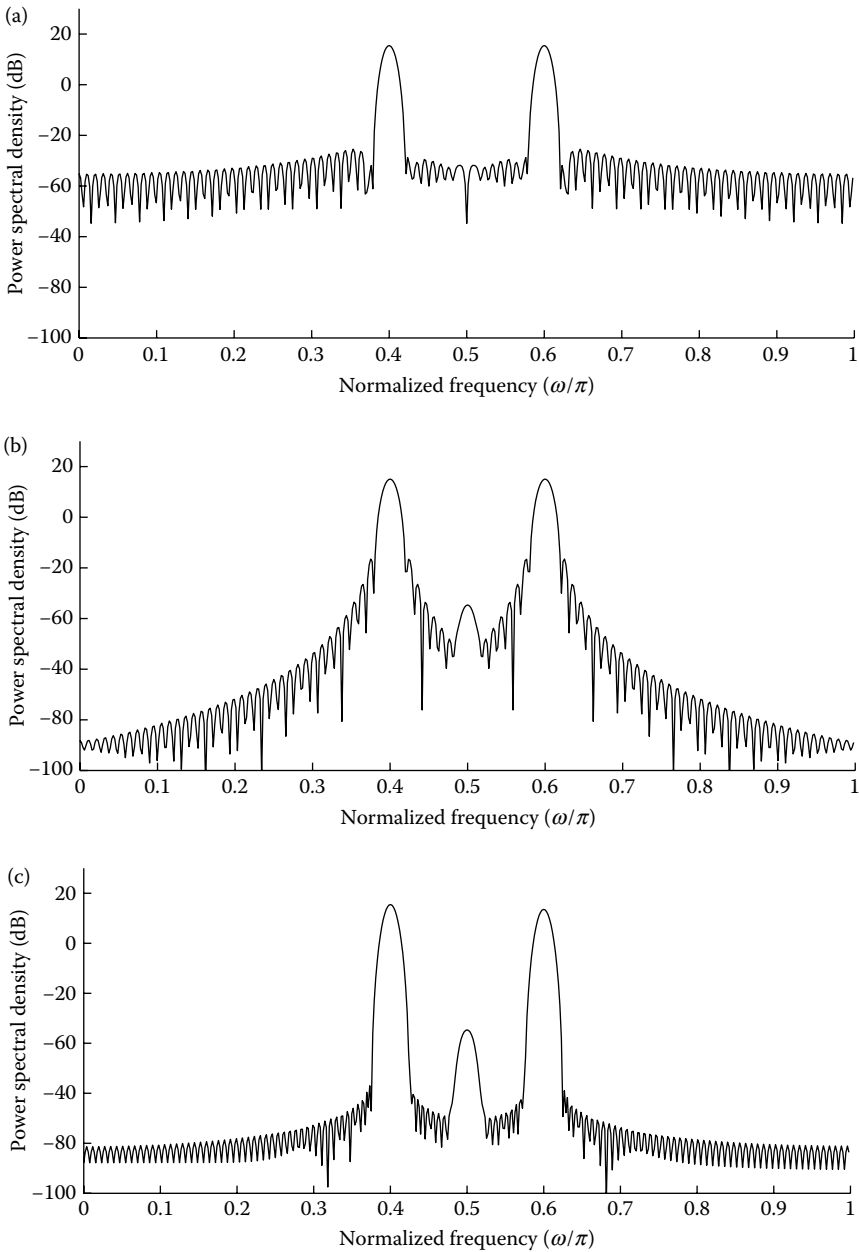
Now, we can compute the values of  $N$ ,  $c$ , and  $\alpha$  using the expression given in Equations 8.34 through 8.36, respectively, as

$$\begin{aligned} \alpha &= 0.12438(70 + 6.3) = 9.490, \\ c &= \frac{6(70 + 12)}{155} = 3.174, \\ \text{and } N &= 1 + 3.174 \times \frac{2\pi}{\Delta\omega} = 191.44. \end{aligned}$$

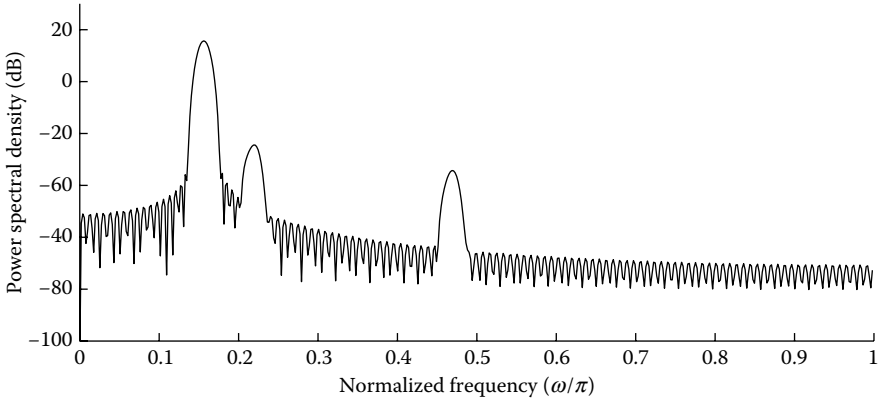
From Figures 8.11 and 8.12, we can clearly observe that the Kaiser–Bessel window is one of the most suitable windows for the spectral analysis of the above signal. We can also see that the Hamming window is not even able to resolve the weak signal component. The Hann window is able to detect the weak component, but the peak is smaller in magnitude and hence, under noisy conditions, the peak may not be detected and this window may also not perform well. Hence, we can confirm that for spectral analysis requiring high resolution and lower side-lobe levels, the Kaiser window is a better choice.

Now, let us consider the signal defined in Equation 8.29 (in the previous sub-section). The modified periodogram estimate using the Kaiser window is given below. The weakest signal component has a magnitude  $-50$  dB. Now, let us calculate the required values of  $N$ ,  $c$ , and  $\alpha$  for this signal. To make a comparison with other windows, let us take  $R = 50$ . Then, we find from Equations 8.34 to Equation 8.36 that

$$\Delta\omega = \frac{(0.21875\pi - 0.15625\pi)}{3} = \frac{\pi}{48'} \quad (8.39)$$



**FIGURE 8.11** Spectral analysis using Hamming, Hann, and Kaiser-Bessel windows. (a) Using Hamming window. (b) Using Hann window. (c) Using Kaiser-Bessel window with  $\alpha = 9.49$ .



**FIGURE 8.12**  
Spectral analysis using Kaiser-Bessel window (second example).

$$\alpha = 0.76609(50 - 13.26)^{0.4} + 0.09834(50 - 13.26) = 6.85, \quad (8.40)$$

$$c = \frac{6(50 + 12)}{155} = 2.4, \quad (8.41)$$

$$\text{and } N = 1 + 2.4 \times \frac{2\pi}{\Delta\omega} = 231. \quad (8.42)$$

By comparing the plot given in Figure 8.12 with those of Figures 8.4 and 8.5, we can clearly see that the Kaiser window is a better choice for spectral analysis purposes because of the fact that the side-lobe attenuation as well as the resolution required can be simultaneously achieved, for a wide range of signals, by varying the parameter  $\alpha$ . Similar to the FIR filter design problem, the periodogram using the Kaiser-Bessel window requires lower lengths of the data to provide better spectral estimates when compared to the other windows.

### 8.1.4 Bartlett Periodogram

The main drawback of the periodogram PSD estimate is its high variance since we have neglected the expectation operation. Therefore, we create a pseudo-ensemble by dividing the input signal into non-overlapping segments of length  $L$  [1]. Then, we individually compute the periodogram estimate for each of these segments and finally average the periodogram estimates with respect to the frequency points as follows:

$$P_B(e^{j\omega}) = \frac{1}{P} \sum_{i=0}^{P-1} P_{PER}^{(i)}(e^{j\omega}), \quad (8.43)$$

where  $P$  is the number of segments and  $P_{PER}^{(i)}$  represents the individual segment's periodogram estimate given by

$$P_{PER}^{(i)}(e^{j\omega}) = \frac{1}{L} \left| \sum_{n=0}^{L-1} x_i[n] e^{-j\omega n} \right|^2. \quad (8.44)$$

*Bias of Bartlett periodogram:* The bias of the Bartlett periodogram is computed as follows:

$$\varepsilon\{P_B(e^{j\omega})\} = \frac{1}{P} \sum_{p=0}^{P-1} \varepsilon\{P_{PER}^{(i)}(e^{j\omega})\} = \varepsilon\{P_{PER}(e^{j\omega})\}. \quad (8.45)$$

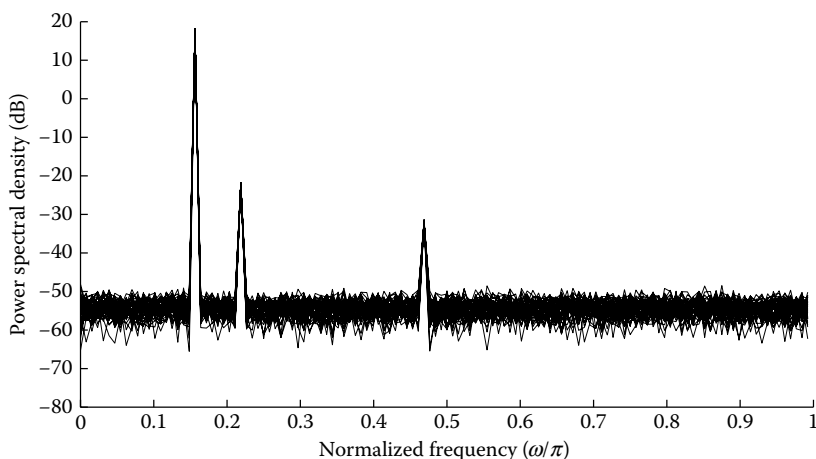
This is because all the segments have identical and independent periodograms.

*Variance of Bartlett periodogram:* Its variance is given by

$$\text{var}\{P_B(e^{j\omega})\} \propto \frac{P_{xx}^2(e^{j\omega})}{P}. \quad (8.46)$$

Thus, the variance of the spectral estimate is reduced as  $P$  is increased. However, the decrease in variance may be much less than desired, if the segments are not statistically independent.

By comparing the Bartlett periodogram estimate given in Figure 8.13 with the periodogram estimate given in Figure 8.3(b), we can observe that the variance is reduced considerably. This reduction in variance is due to averaging of the periodograms of  $P$  sub-sequences. In this case, the length of window  $N = 1024$  and  $P = 4$  was used, but the reduction in variance is less than a factor of four because of the correlation existing between the sub-sequences. Thus,



**FIGURE 8.13**

Bartlett periodogram for  $N = 1024$ .

as  $N \rightarrow \infty$ , with both  $P \rightarrow \infty$  and  $L \rightarrow \infty$ , the bias as well as the variance of the Bartlett periodogram approaches zero. Hence, it can be considered as a consistent estimator.

In Figure 8.14, Bartlett periodograms with  $P = 4$  and  $P = 16$ , with  $N = 2048$  are shown. Larger length of data has been used to make the difference more clear. In the case of  $P = 4$ , we can see that the variance of the periodogram for 50 iterations (with noise added) produces more variance than in the case when  $P = 16$ . The variance accounts for the darker regions of the plot due to the overlaying of different plots. For the  $P = 16$  case, the variance is comparatively reduced, because 16 different periodograms are averaged out. However, the reduction in variance is not as good as that given by Equation 8.46. In this method, the reduction in variance is achieved at the expense of loss in resolution. This is because, as the value of  $P$  increases, the segment length reduces; hence, the individual periodograms will have low resolution.

*Resolution of Bartlett periodogram:* The expression for the resolution of the Bartlett method is similar to the periodogram estimate, except that the length of the data is changed with respect to the length of the segment [3]. Therefore, the resolution of the Bartlett method is poorer, that is,

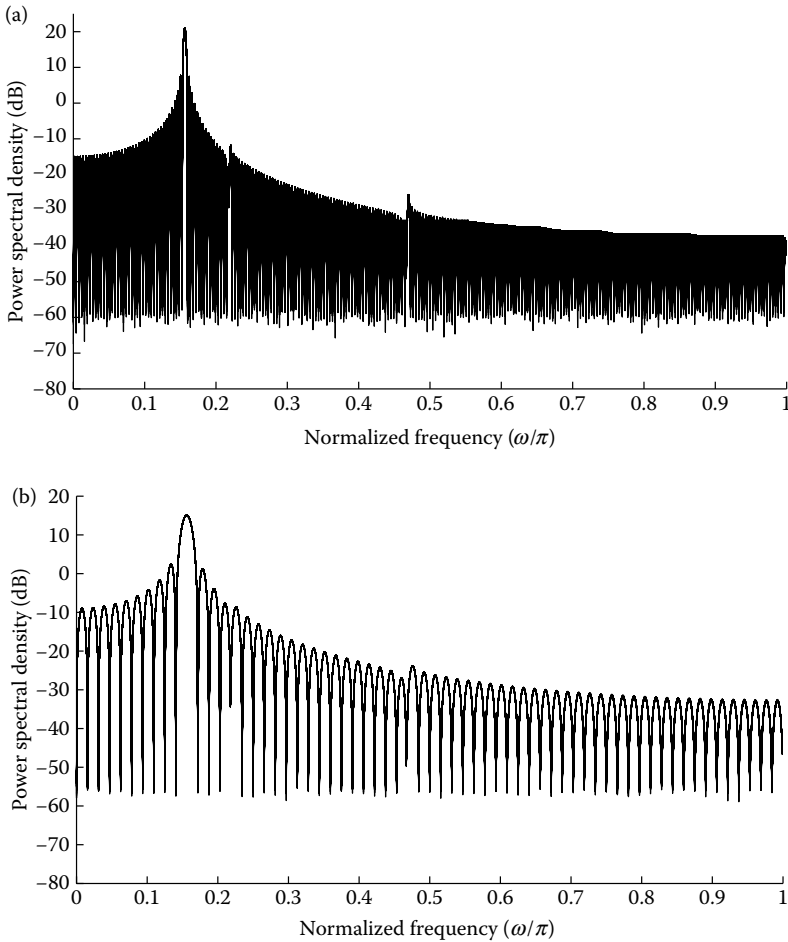
$$\text{Res}\{P_B(e^{j\omega})\} = 0.89 \left( \frac{2\pi}{L} \right) = 0.89P \left( \frac{2\pi}{N} \right), \quad (8.47)$$

where  $L = \frac{N}{P}$ . In Figure 8.15, Bartlett periodograms with  $P = 4$  and  $P = 16$  (with  $N = 1024$ ) are shown. From these plots, we can clearly see that the resolution gets reduced dramatically when averaging is done over  $P$  periodograms. The disadvantage of this method is that the spectral resolution is reduced due to the fact that we divide the input signal into segments. Hence, a trade-off should be maintained between the number of segments  $P$  and the number of data samples within a segment  $N$  as  $LP \leq N$ , the length of the input sequence.

### 8.1.5 Welch Periodogram Method

This is a modified version of the Bartlett periodogram in which a data window is applied on each of the segments before computing the periodogram. The segments are allowed to overlap, and hence the number of segments over which averaging is done can be increased, thereby giving a better estimate with reduced variance. The effect of applying a data window is to suppress the effects of side lobes. However, this results in smearing of peaks, which inevitably reduces the resolution [1]. The expression for the Welch periodogram estimate is given by

$$P_W(e^{j\omega}) = \frac{1}{PLU} \sum_{i=0}^{P-1} \left| \sum_{n=0}^{L-1} f[n]x[n+iD]e^{-j\omega n} \right|^2, \quad (8.48)$$



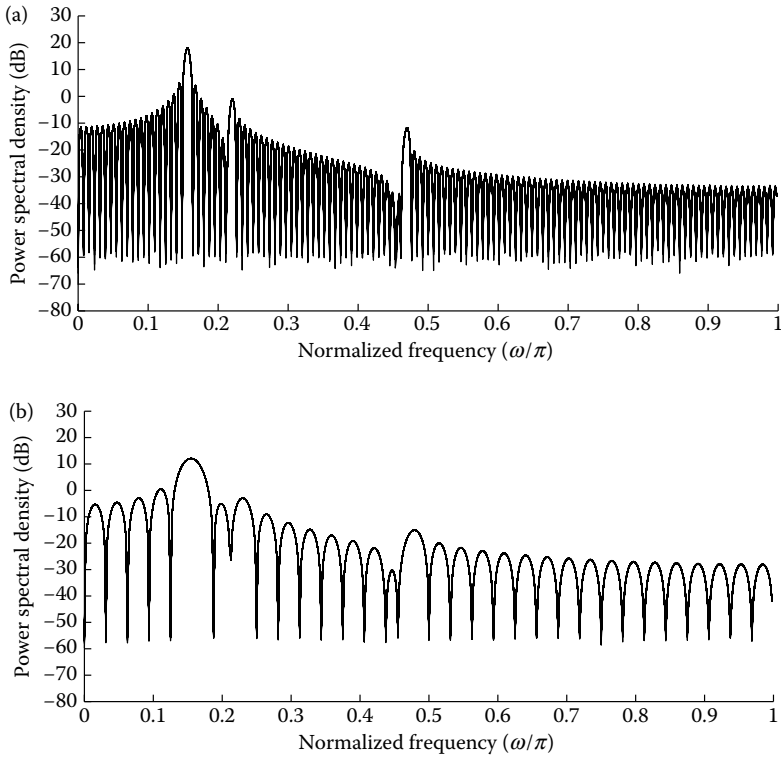
**FIGURE 8.14** Effect of  $P$  on variance of Bartlett periodogram (for  $N = 2048$ ). (a)  $P = 4$ . (b)  $P = 16$ .

where  $P$  is the number of sub-sequences of length  $L$  and  $D$  is the offset between data points in  $x[n]$ , which goes to a particular point in two adjacent subsequences. Hence,  $(L - D)$  represents the number of overlapping segments. We recall that  $U$  is a normalization constant defined in Equation 8.31 (see modified periodogram discussed in Section 8.1.2).

*Bias of Welch periodogram:* The Welch periodogram can be represented in terms of the modified periodogram as

$$P_W(e^{j\omega}) = \frac{1}{P} \sum_{i=0}^{P-1} P_M^{(i)}(e^{j\omega}), \tag{8.49}$$





**FIGURE 8.15** Effect of different values of  $P$  on resolution of Bartlett periodogram ( $N = 1024$ ). (a)  $P = 4$ . (b)  $P = 16$ .

where  $P_M^{(i)}(e^{j\omega})$  is the modified periodogram of each sub-sequence. The expected value of the Welch estimate is given as

$$\begin{aligned} \varepsilon\{P_W(e^{j\omega})\} &= \varepsilon\{P_M(e^{j\omega})\} \\ &= \frac{1}{2\pi LU} P_{xx}(e^{j\omega}) * |F(e^{j\omega})|^2, \end{aligned} \tag{8.50}$$

where  $F(e^{j\omega})$  is the Fourier transform of the  $L$ -point data window. The Welch periodogram estimator is an asymptotically unbiased estimator, similar to the case of the modified periodogram.

*Variance of Welch periodogram:* The variance is difficult to compute because the overlapping introduces a correlation between different subsequences. Hence, similar to the Bartlett periodogram, the variance of the Welch periodogram is also inversely proportional to the number of sub-sequences, as

shown below:

$$\text{var}\{P_W(e^{j\omega})\} \propto \frac{P_{xx}^2(e^{j\omega})}{P}. \quad (8.51)$$

For an overlap of 50%, the variance of the Welch periodogram estimate is

$$\text{var}\{P_W(e^{j\omega})\} \approx \frac{9}{16} \frac{L}{N} P_{xx}^2(e^{j\omega}). \quad (8.52)$$

Thus, as  $N \rightarrow \infty$ , the variance of the Welch periodogram estimate tends to zero. Hence, it can be called a consistent PSD estimator.

*Resolution of Welch periodogram:* The resolution of  $P_W(e^{j\omega})$  is determined by the spectral resolution of each sub-sequence of  $x[n]$ . Now, similar to the modified periodogram, the resolution of the Welch method is also window-dependent. However, the resolution of the Welch method will be poorer because the sub-sequences have smaller lengths.

From Figures 8.16 and 8.17, we can observe that as the overlap is increased, and the variance of the Welch estimator using the Hann window is reduced. This reduction in variance arises because a larger number of segmental periodograms are available for averaging (than without overlap) as in the case of Bartlett method. However, we can see that as the overlap increases beyond 70%, there is not much reduction in the variance, since the segments become more correlated. Hence, we commonly use an overlap of 50–75%, as increasing the overlap beyond this does not improve the variance.

Figures 8.18 through 8.20 display the effect of applying different window functions for the Welch method. The introduction of windows in the Welch periodogram method has the same advantage as given in the case of the modified periodogram. In these figures, an overlap of 50% is used with  $N = 1024$ . We can clearly see that the resolution of the Welch periodogram depends on the data window used.

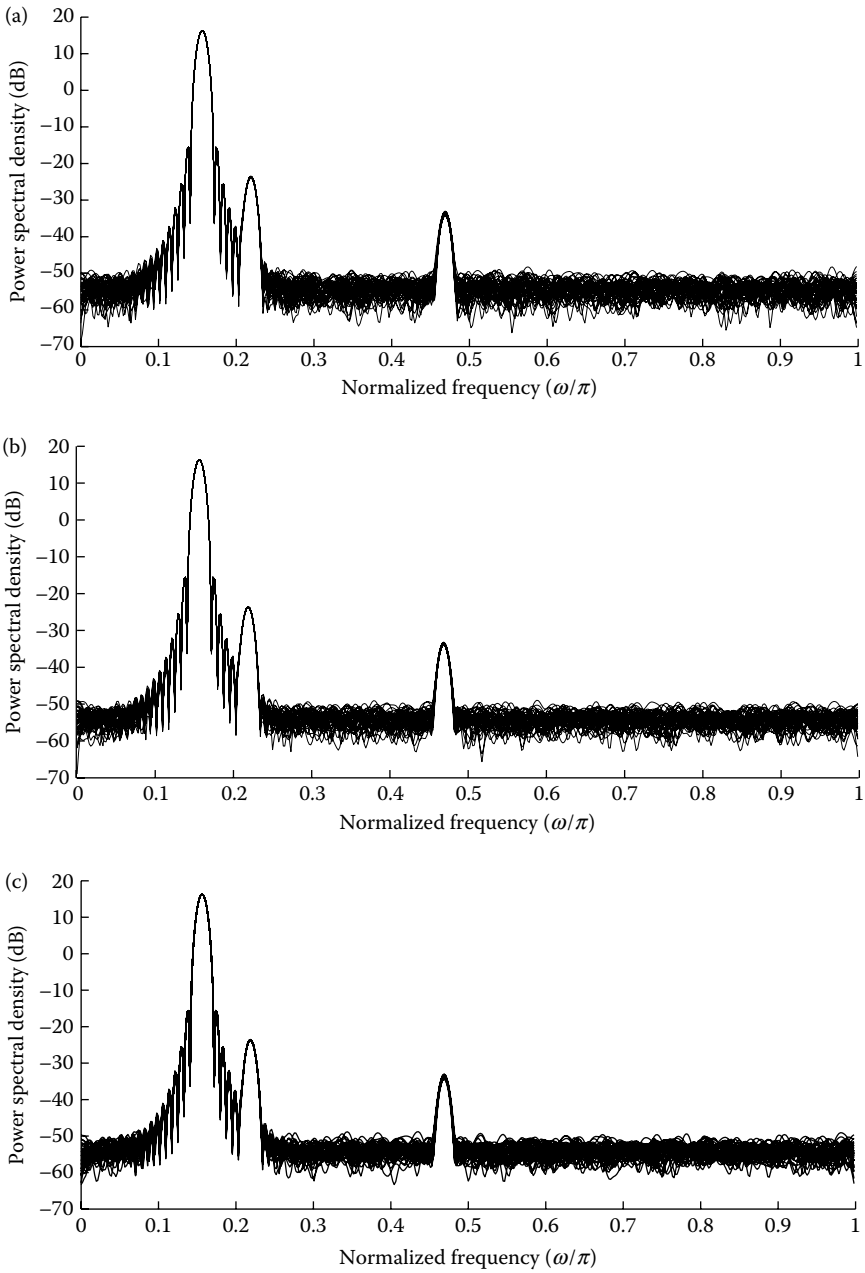
### 8.1.6 Blackman–Tukey Method

Another method to compute PSD is through the correlogram method, where we compute a PSD estimate as the Fourier transform of the ACF as shown below:

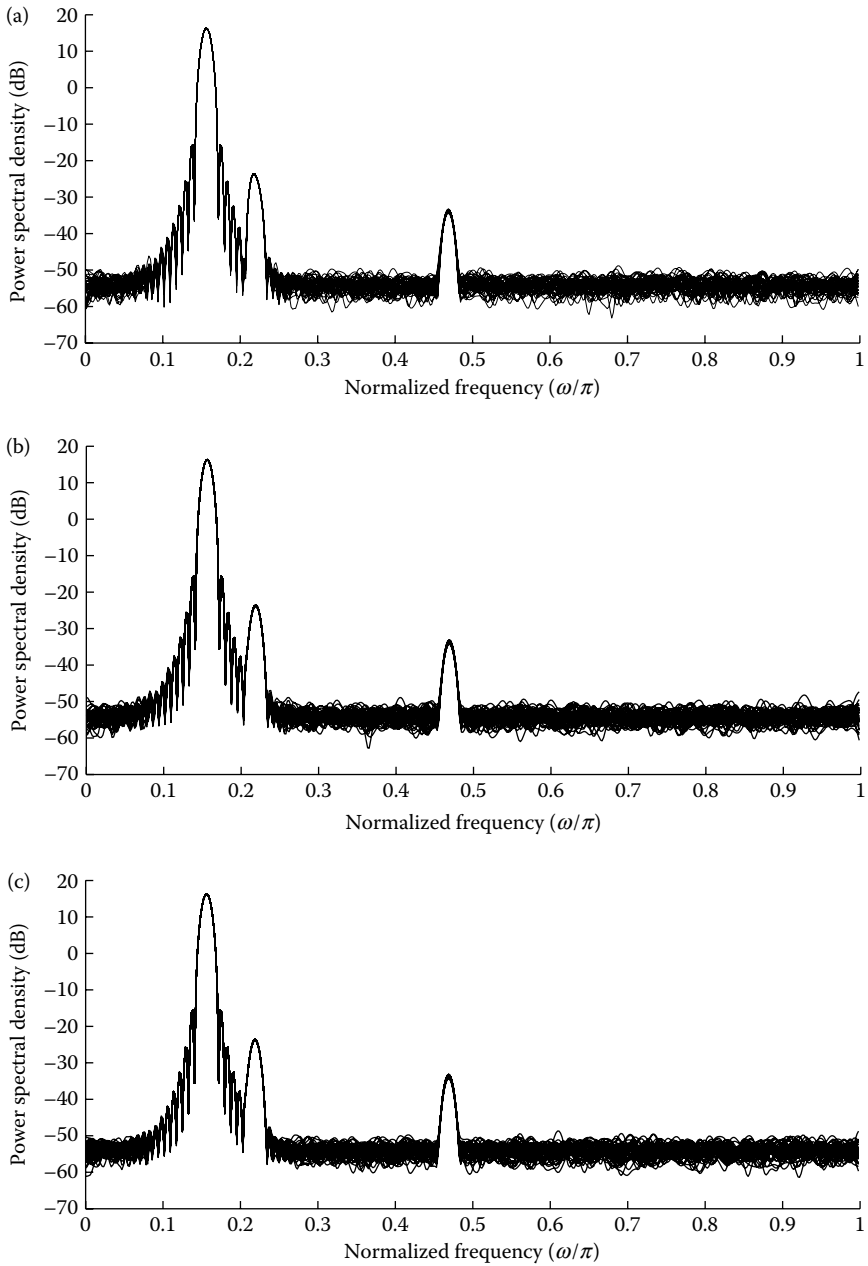
$$P_{PER}(\omega) = \sum_{-(N-1)}^{(N-1)} r_{xx}[k] e^{-j\omega k}, \quad (8.53)$$

where

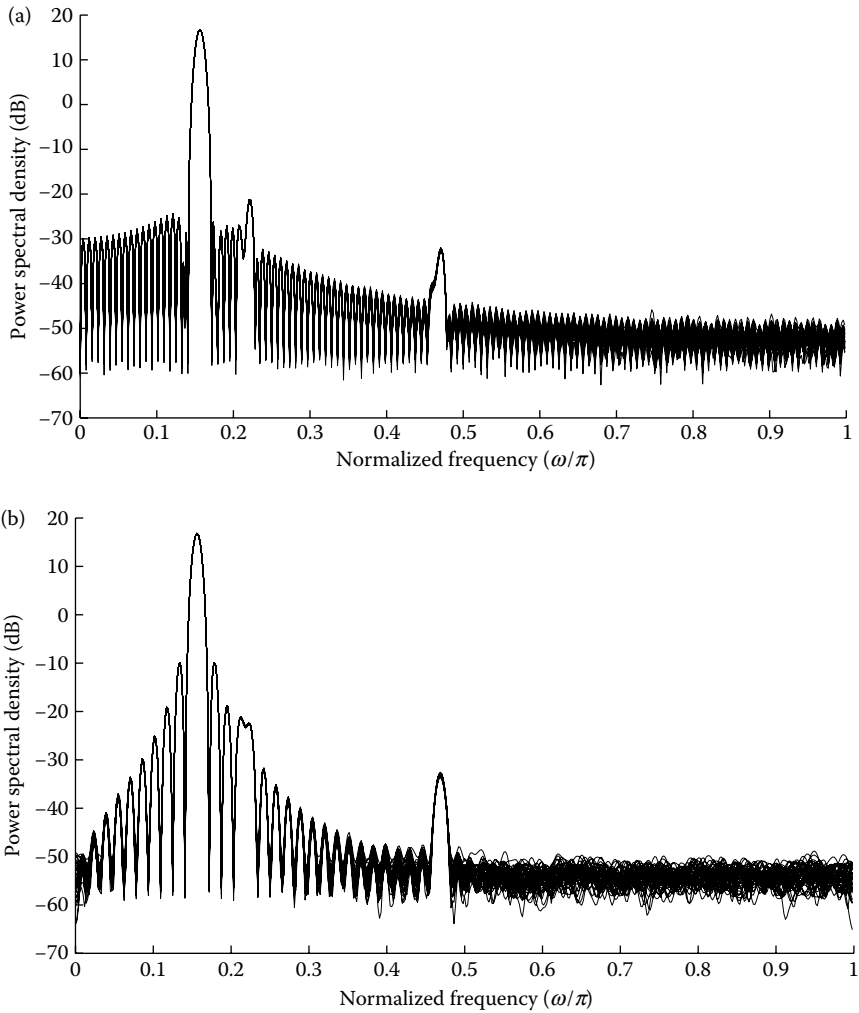
$$r_{xx}[k] = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} x^*[n] x[n+k], & \text{for } k = 0, 1, \dots, (N-1). \\ r_{xx}^*[-k], & \text{for } k = -(N-1), \dots, -1. \end{cases} \quad (8.54)$$



**FIGURE 8.16** Welch periodogram with different overlapping (with Hann window). (a) Overlap of 10%. (b) Overlap of 20%. (c) Overlap of 40%.



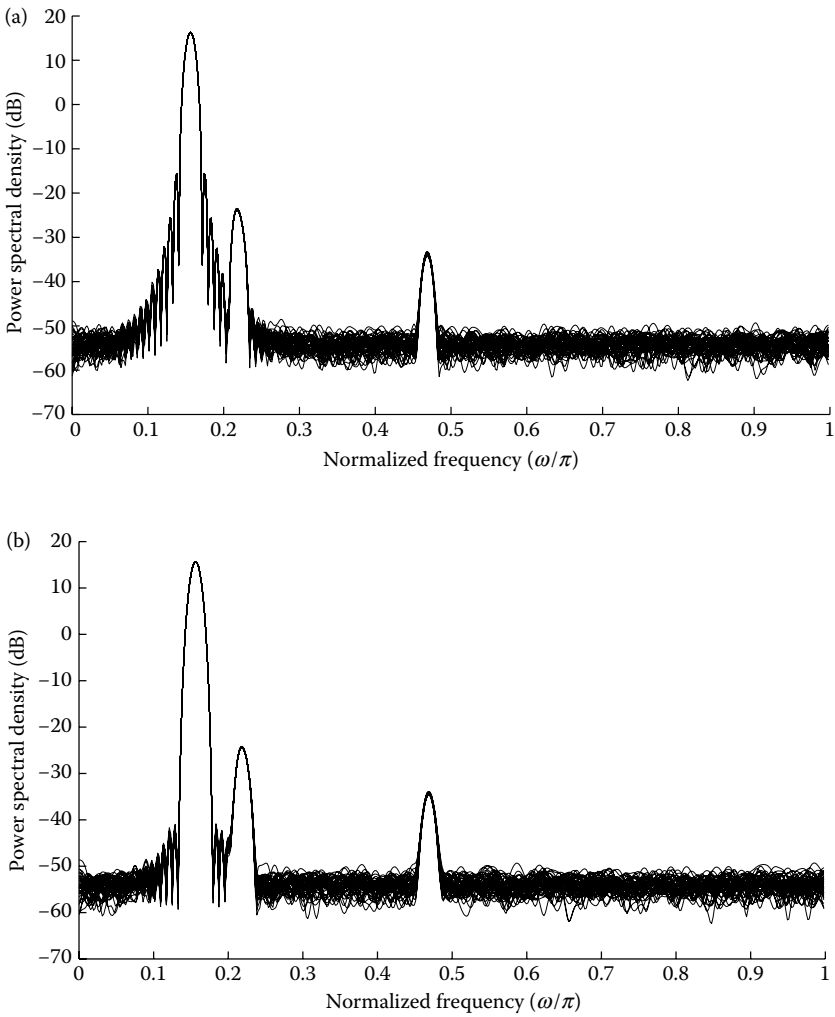
**FIGURE 8.17** Welch periodogram with different overlapping (with Hann window). (a) Overlap of 50%. (b) Overlap of 70%. (c) Overlap of 90%.



**FIGURE 8.18** Welch periodogram with different windows. (a) Hamming window. (b) Bartlett window.

Poor estimates of the ACF leads to inferior performance of the periodogram method. The ACF estimate with smaller lags can be estimated more accurately than the ones with lags close to  $N$ . This is because a smaller number of terms are used in the summation. Therefore, the large variance of the periodogram is due to the high weightage given to the flawed autocorrelation estimate used in its evaluation.

Blackman and Tukey [7] have proposed a weighting scheme, such that the autocorrelations with greater lags are associated with lower weights. The

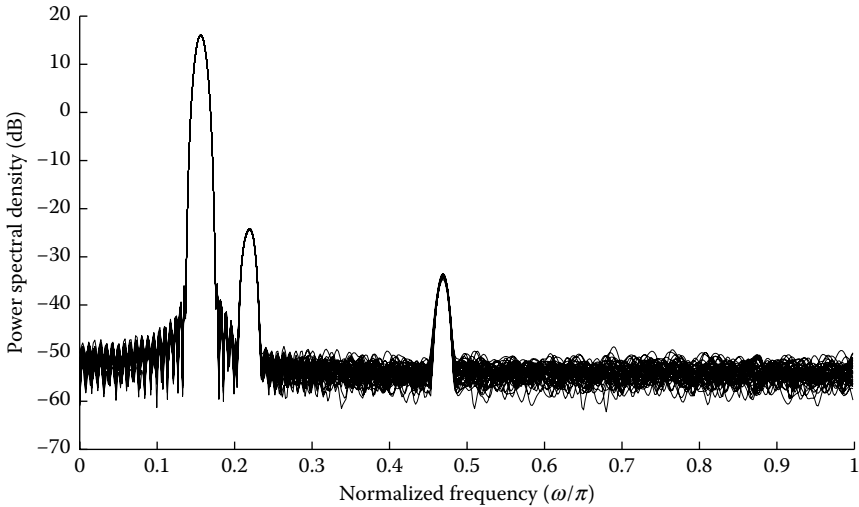


**FIGURE 8.19** Welch periodogram with different windows. (a) Hann window. (b) Blackman window.

expression for this estimator is given by

$$P_{BT}(\omega) = \sum_{-(N-1)}^{(N-1)} f[k]r_{xx}[k]e^{-j\omega k}, \tag{8.55}$$

where  $f[k]$  is a window function that is non-negative, symmetric, and non-increasing with respect to  $|k|$ , that is,



**FIGURE 8.20** Welch periodogram with Kaiser–Bessel window with  $\alpha = 7$ .

1.  $0 \leq f[k] \leq f[0] = 1$
2.  $f[-k] = f[k]$
3.  $f[0] = 0$ , for  $|k| > M$ , where  $M \leq N - 1$

This scheme is called the Blackman–Tukey spectral estimator. This happens to be equivalent to the periodogram method, if

$$f[k] = 1, \quad \text{for } |k| \leq M = N - 1. \tag{8.56}$$

The weighting of ACF will reduce the variance of the spectral estimator but this is at the expense of increasing the bias. There are several lag windows available in the literature [8] but only a certain class of windows is used for computing the PSD using the Blackman–Tukey method. This arises from the fact that when we represent Equation 8.55 in the frequency-domain, it becomes a convolution operation of the lag window with  $P_{PER}$ , which can be expressed as

$$P_{BT} = \int_{-\pi}^{\pi} F(\omega - \xi)P_{PER}(\xi)d\xi. \tag{8.57}$$

Thus, owing to the convolution operation, we can use the spectral window function, which has the property

$$F(e^{j\omega}) \geq 0, \quad -\pi \leq \omega \leq \pi. \tag{8.58}$$

This constraint is set to avoid a negative spectral estimate due to the application of the lag window. Hence, only certain window functions such as the Bartlett, Parzen (Jackson), Bohman windows are used.

*Bias of Blackman–Tukey method:* In the Blackman–Tukey method, we apply a lag window to the ACF. Hence, the expectation of  $P_{BT}(e^{j\omega})$  is obtained as

$$\begin{aligned}\varepsilon\{P_{BT}(e^{j\omega})\} &= \varepsilon \left\{ \sum_{k=-M}^M f_{BT}[k] r_{xx}[k] e^{-j\omega k} \right\} \\ &= \sum_{k=-M}^M f_{BT}[k] \left( \frac{N-|k|}{N} \right) r_x[k] e^{-j\omega k},\end{aligned}\quad (8.59)$$

where  $f_{BT}[k]$  denotes the lag window and  $f_B[k] = \frac{N-|k|}{N}$  is the Bartlett window. The length of the lag window is  $2M$ . Now, applying the modulation property of the Fourier transform in Equation 8.59, we obtain

$$\varepsilon\{P_{BT}(e^{j\omega})\} = \frac{1}{2\pi} F_{BT}(e^{j\omega}) * F_B(e^{j\omega}) * P_{xx}(e^{j\omega}).\quad (8.60)$$

Owing to the presence of the explicit window term  $F_B(e^{j\omega})$ , we will get the effect of applying both the lag window and the implicit Bartlett window on the autocorrelation sequence. As  $N \rightarrow \infty$ , the expression  $F_{BT}(e^{j\omega}) * F_B(e^{j\omega})$  tends to an impulse function. Therefore

$$\lim_{N \rightarrow \infty} \varepsilon\{P_{BT}(e^{j\omega})\} = P_{xx}(e^{j\omega}).\quad (8.61)$$

Thus, the Blackman–Tukey method is asymptotically unbiased.

*Variance of Blackman–Tukey method:* The variance of the Blackman–Tukey method is given by

$$\text{var}\{P_{BT}(e^{j\omega})\} \approx \frac{P_{xx}^2(e^{j\omega})}{N} \sum_{k=-M}^M f_{BT}[k], \quad N \gg M.\quad (8.62)$$

Thus, the higher the value of  $M$ , the higher the variance.

*Resolution of Blackman–Tukey method:* The spectral estimate of the  $P_{BT}(e^{j\omega})$  is dependent on the window function used. Since the length of the lag window is  $2M$ , the spectral resolution is given by

$$\text{Res}\{P_{BT}(e^{j\omega})\} = b \frac{2\pi}{2M},\quad (8.63)$$

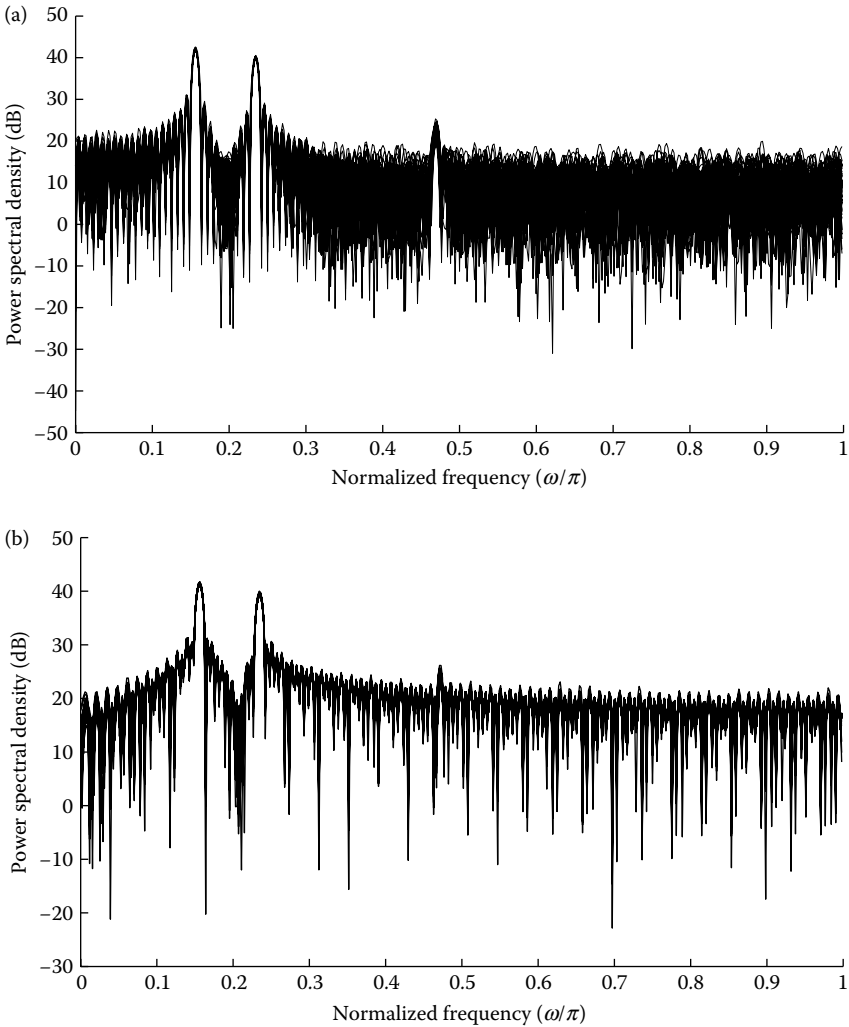
where  $b$  is a parameter that depends on the 6 dB bandwidth of the window. Hence, as the value of  $M$  increases, the resolution gets better.



Figures 8.21 and 8.22 show the Blackman–Tukey PSD estimate, each with a Bartlett lag window of different lengths. A typical composite signal consisting of three sinusoidal components can be represented as

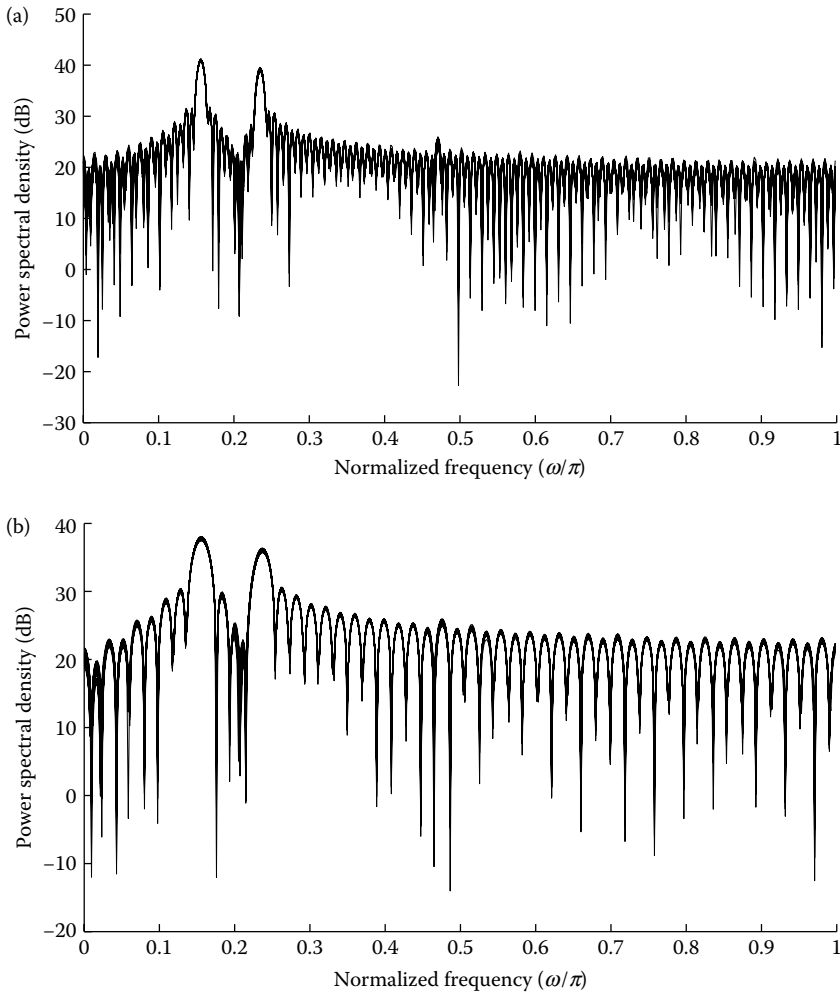
$$x[n] = \sin[0.15625\pi n] + 0.8 \sin[0.25\pi n] + 0.2 \sin[0.46875\pi n], \quad (8.64)$$

White noise with a variance of 0.2 is added to the above signal.



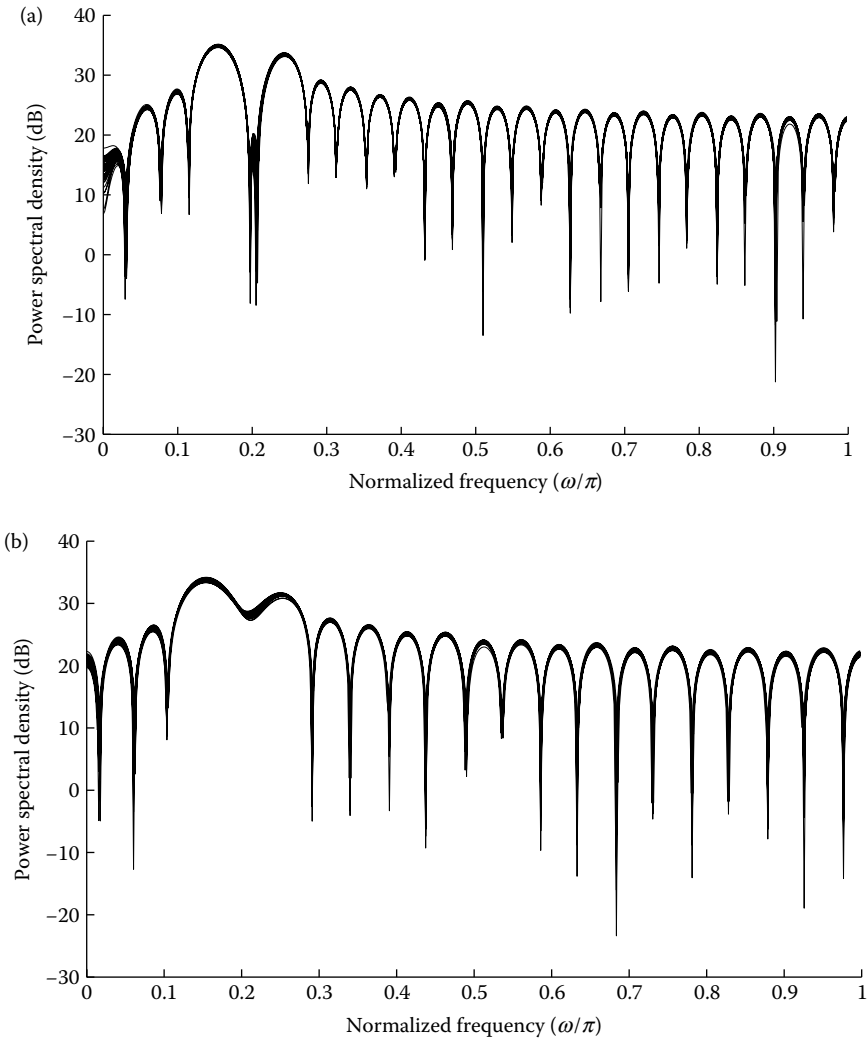
**FIGURE 8.21**

Effect of varying  $M$  on variance and resolution of Blackman–Tukey method. (a)  $M = N$ . (b)  $M = 0.6N$ .

**FIGURE 8.22**

Effect of varying  $M$  on variance and resolution of Blackman–Tukey method (continued). (a)  $M = 0.5N$ . (b)  $M = 0.2N$ .

In Figure 8.21(a), we have used  $M = N$ . Hence, it includes the complete autocorrelation sequence, thereby resulting in a higher resolution of the spectral estimate. However, this resolution is obtained at the expense of higher variance, which is apparent from the plot. Now, as the value of  $M$  is reduced, the variance of the Blackman–Tukey estimate also gets reduced (see Figure 8.21(b)). The plots given in Figure 8.21 have very high variance, whereas the result shown in Figure 8.22 has a reduced variance (without much effect on resolution). Hence, the preferred value of  $M$  is less than  $N/2$ . In Figure 8.23, the plots have a much lower variance, but the resolution of the estimate is



**FIGURE 8.23** Effect of varying  $M$  on variance and resolution of Blackman–Tukey method (continued). (a)  $M = 0.1N$ . (b)  $M = 0.08N$ .

poorer. Therefore, the recommended value of  $M$  can be taken to be greater than  $N/5$ .

**8.1.7 Daniel Periodogram**

This is an approach to obtain a smooth periodogram estimate by averaging over the adjacent spectral frequencies [1]. Hence, we can define the expression

for averaging over  $P$  points on either side of the frequency axis,  $\omega_i$ , as

$$P_D(e^{j\omega_i}) = \frac{1}{2P+1} \sum_{n=i-P}^{i+P} P_{xx}(e^{j\omega_n}). \quad (8.65)$$

This approach is a special case of the Blackman–Tukey method, where the window used is rectangular. We notice that this operation is just lowpass filtering.

### 8.1.8 Application of the FFT to the Computation of a Periodogram

The computation of a periodogram using a digital computer can be simplified by the use of an FFT algorithm. In case of FFT computation, we sample the frequency range  $[-\pi, \pi]$  into  $N$  equidistant discrete-frequency points spaced at  $\frac{2\pi}{N}$ . This was already described in Chapter 1. An interesting fact is that here the frequency at the  $k$ th sample point is  $\frac{2\pi}{N}k$ . We find that the frequencies can have finer spacing if  $N$  is increased (since the spacing is  $\frac{2\pi}{N}$ ). This can be done by padding  $(N_1 - N)$  zeros to the right of  $x[n]$ , thereby increasing the length to  $N_1$ . However, we can see from practical results that this modification only improves the computation of periodogram but not the actual resolution (see Chapter 1).

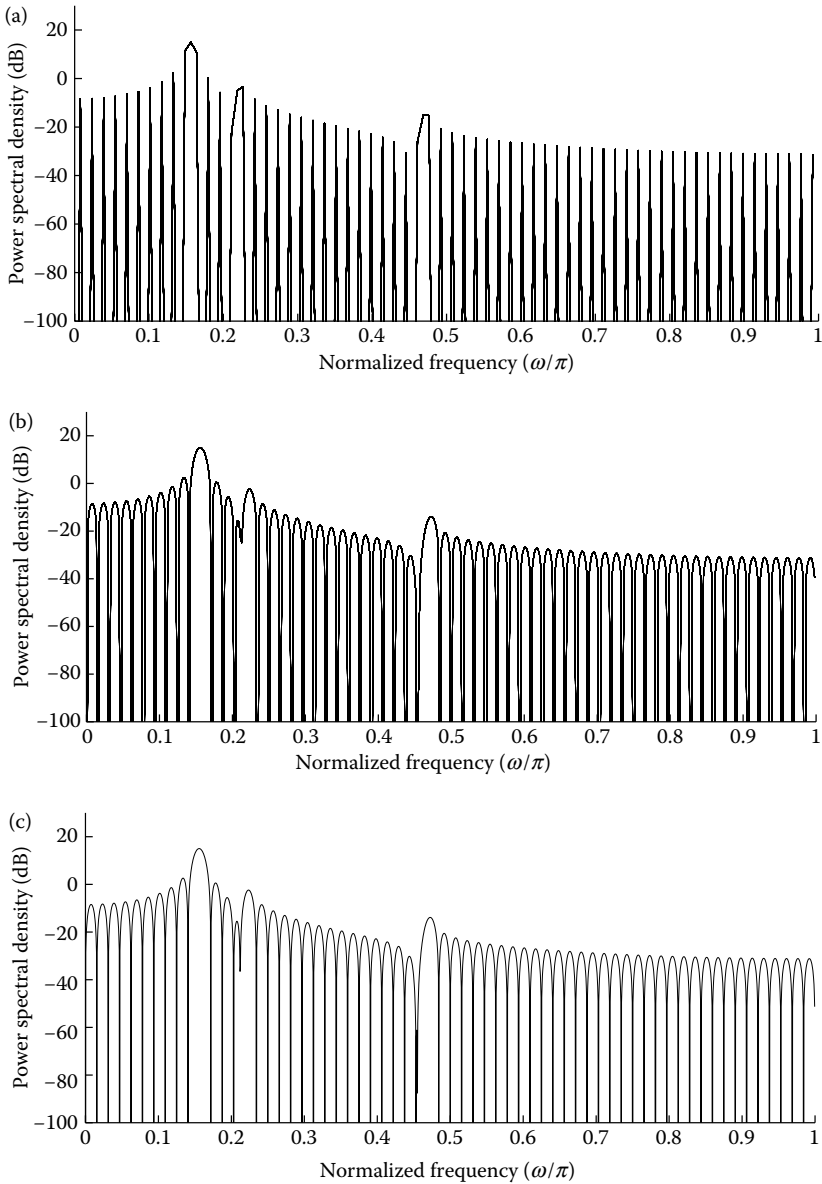
In Figure 8.24, we have shown the effect of increasing the number of FFT points around the unit circle. From these plots, we can conclude that the increase in frequency sampling points by padding with zeros does not increase the resolution; instead, it only improves the visual representation of the periodogram. The finer details are revealed better in Figure 8.24(c) than in Figure 8.24(a).

### 8.1.9 Short-Time Fourier Transform

We can define the time-dependent or short-time Fourier transform (STFT) [9] as

$$X_n(e^{j\omega}) = \sum_{k=-\infty}^{\infty} f[n-k]x[k]e^{-j\omega k}. \quad (8.66)$$

The STFT can be interpreted in two different ways: either as the Fourier transform of a windowed sequence or as a linear filtering operation. From Equation 8.66, it is clear that STFT is the Fourier transform of a windowed sequence  $x[k]f[n-k]$ , where  $f[n]$  represents a window function. Using this approach, we can obtain a localization in time of the signal as well. This method is used in the case of nonstationary signals, such as speech signals.



**FIGURE 8.24** Effect of increasing the number of FFT points. (a)  $N_1 = 256$ . (b)  $N_1 = 1024$ . (c)  $N_1 = 4096$ .

In this approach, we localize the signal in time by applying a window. The window length has to be chosen in such a way that the signal within that period can be assumed to be stationary.

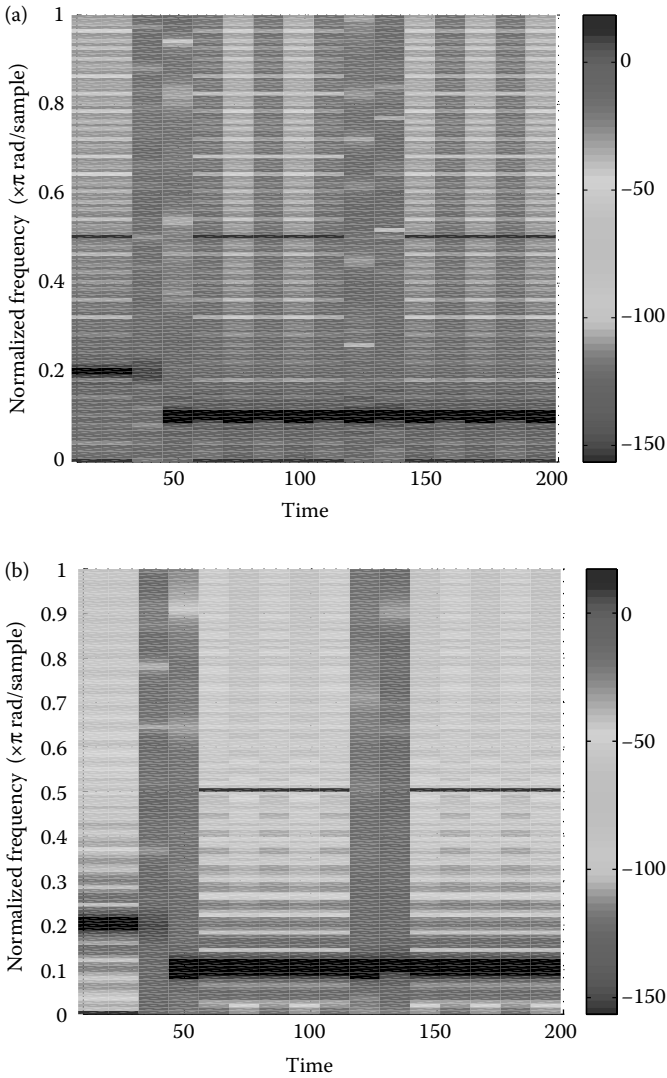
The spectrogram is a graphical representation of the magnitude response of the STFT. Figures 8.25 through 8.27 show the spectrograms using some common windows such as rectangular, Hamming, Hann, triangular, and Blackman. The DTFTs of these windows, their main-lobe widths, side-lobe attenuation levels, and other properties have already been discussed in Chapters 3 and 5. All these characteristics can be clearly verified from the spectrogram plots. In these figures, the magnitude of the STFT is shown as a color gradient. The spectrogram is computed for a nonstationary signal having three signal components, which occur for only a short duration. The signal consists of two short-duration sinusoidal components separated by white noise in time.

*Using rectangular window:* For this window, the main-lobe width is very narrow, but the side-lobe attenuation is quite low (around  $-13$  dB) and remains constant irrespective of the length of the window. Only the main-lobe width becomes much sharper when the window length increases. However, the high side-lobe levels will lead to spectral leakage. Since this window has high spectral resolution, it can be used in problems where we need to resolve very closely spaced spectral peaks with high amplitudes. One such application is *formant extraction* in a speech signal. In the spectrogram shown in Figure 8.25(a), we can see that due to spectral leakage, the energy is spread over the entire spectrum, but with narrow lines at the actual frequencies (due to the high resolution).

*Using Hamming window:* In this case, the main-lobe width is twice that of the rectangular window, but the side-lobe attenuation is more than  $-40$  dB. Therefore, the spectral resolution obtained due to this window is poorer when compared to the rectangular window case, but it has less spectral leakage. Another additional feature of this window is that it provides a smoother spectrum when compared to a spiky spectrum exhibited by the rectangular window. Hence, it is mainly used in speech applications as a trade-off between resolution and spectral leakage. In the spectrogram, we can see that the frequency lines have concentrated energy and the spectral leakage is less.

*Using Hann window:* The spectral resolution that can be obtained by the Hann window is almost the same as the Hamming window case, but the side-lobe attenuation is much higher. We can also observe from the spectrogram that the spectral leakage is quite less than that of the Hamming window (Figure 8.26(a)). Since the Hann window is a very smooth filter, it undesirably distorts the time-domain signal waveform (more than the Hamming window case) because of its shape. However, the Hann window is preferred over the Hamming window because of its high side-lobe fall-off rate.

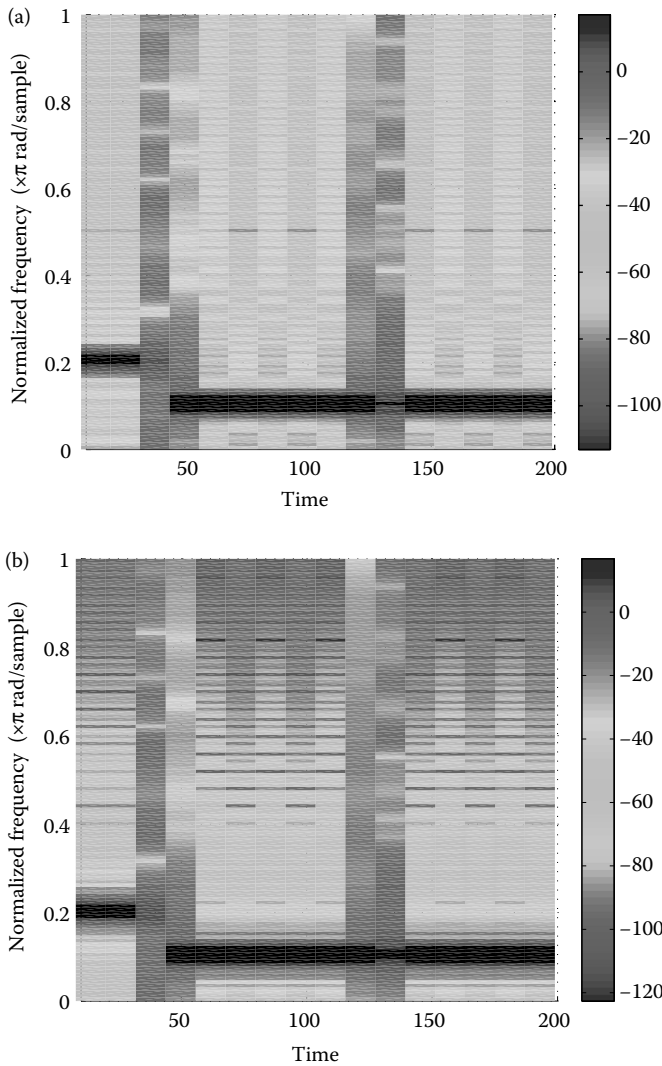
*Using Blackman window:* For this window, the spectral bandwidth is very large and the side-lobe attenuation is very high. Hence, we can see from the spectrogram that the spectral leakage is very low compared to the above windows, but the spectral resolution is poorer (Figure 8.27(a)). This window



**FIGURE 8.25** Spectrograms. (a) Using rectangular window. (b) Using Bartlett window.

is not preferred for spectral estimation applications, since it smoothens out the spectral peaks.

*Using Kaiser–Bessel window (with  $\alpha = 7$ ):* From the spectrogram plots, it is clear that a Kaiser–Bessel window (Figure 8.27(b)), with an appropriate value of  $\alpha$ , is optimum for spectral estimation. The Kaiser–Bessel window is a variable-parameter window and we can choose  $\alpha$  depending on the



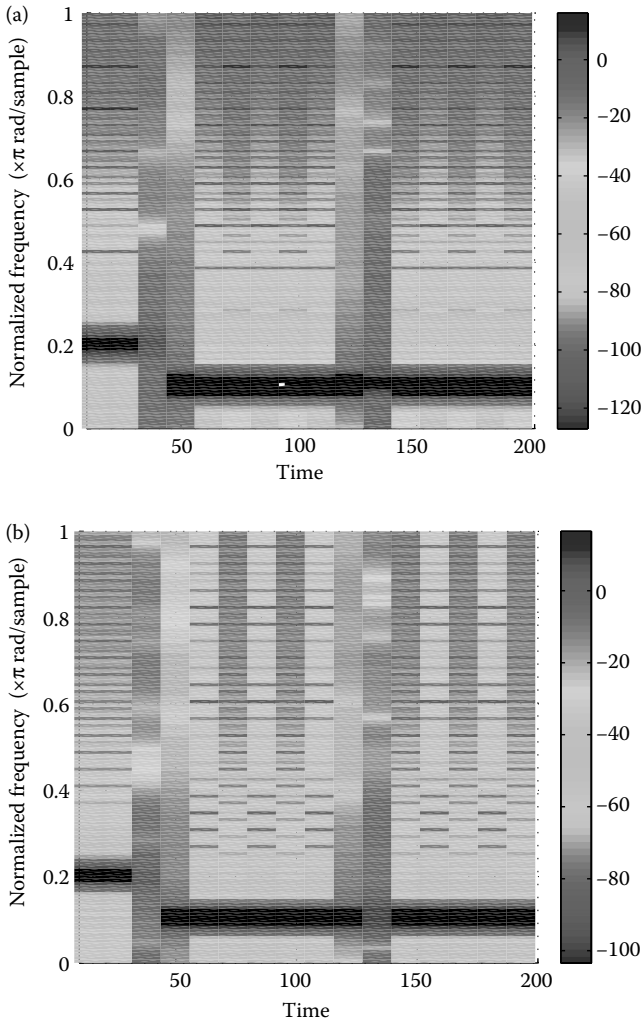
**FIGURE 8.26** Spectrograms. (a) Using Hamming window. (b) Using Hann window.

requirements. This is, of course, an advantage when compared to all the other standard fixed windows.

### 8.1.10 Conclusions

By comparing all the methods that make use of windows for spectral estimation, it is not easy to converge on a single window that is the best in



**FIGURE 8.27**

Spectrograms. (a) Using Blackman window. (b) Using Kaiser–Bessel window with  $\alpha = 7$ .

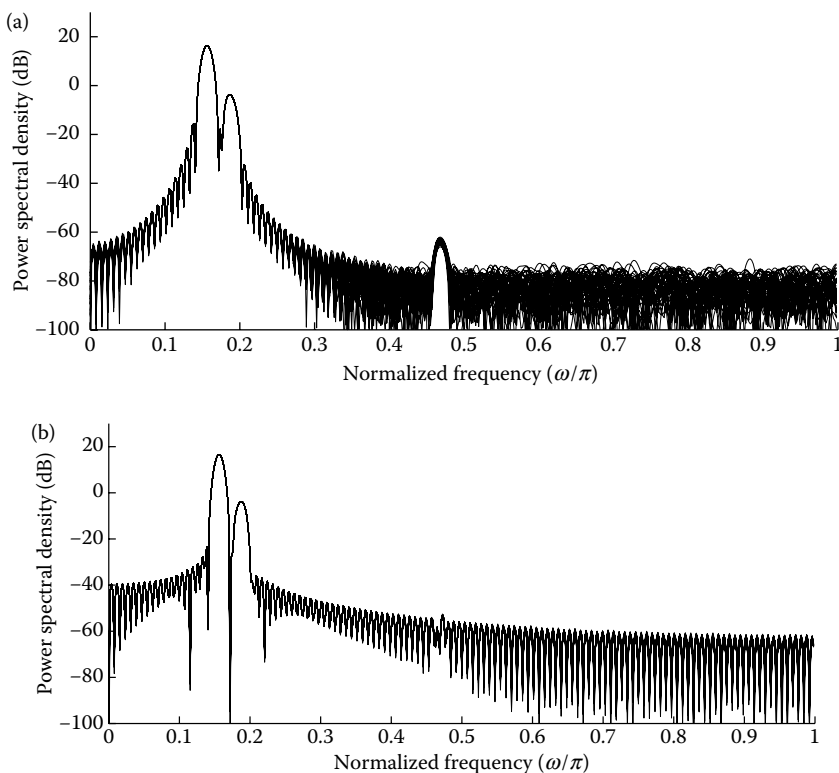
all applications. However, we can say that one of the appropriate fixed-parameter windows for most applications is the Hann window due to its higher side-lobe roll-off rate. The Kaiser–Bessel window has the advantage of being a variable-parameter window which makes it amenable for good performance in most applications. Hence, for the same window length, we can achieve lower PSLR using Kaiser–Bessel window, which will enable us to detect weaker signal components even better than the Hann window. Under certain conditions, the Hann window turns out to be superior to the Kaiser–Bessel window. One such case is illustrated in the following example.

Consider a signal

$$x[n] = \sin[0.15625\pi n] + 0.1 \sin[0.1875\pi n] + 0.0001 \sin[0.46875\pi n] + \eta, \quad (8.67)$$

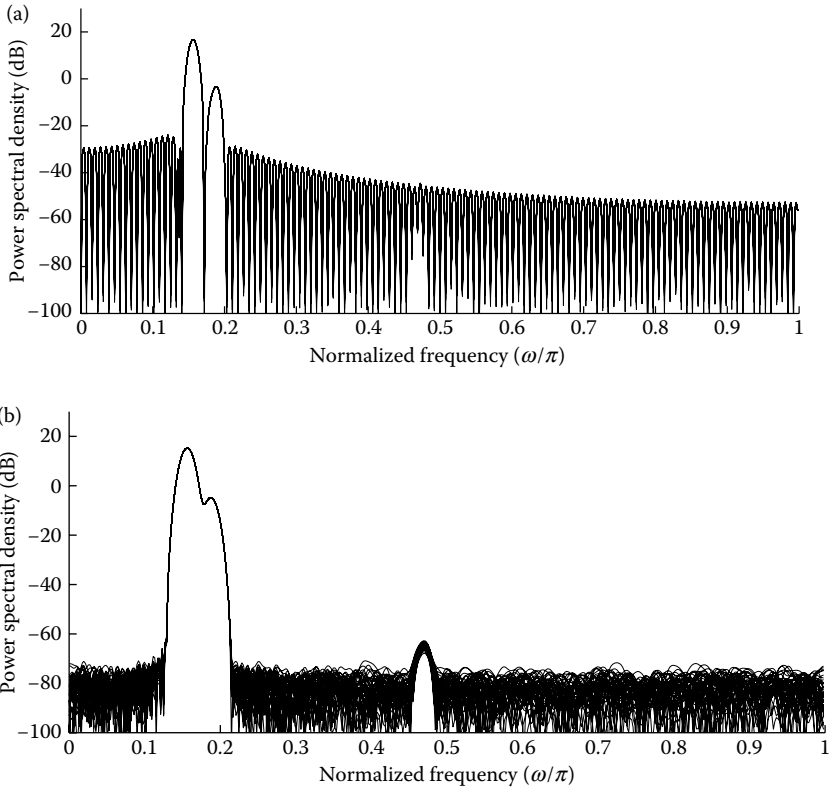
where  $\eta$  is additive white noise with a variance of 0.00009.

From Figures 8.28 through 8.30, we can observe that the Hann window performs better than all the other commonly used windows. By using a Kaiser–Bessel window with  $\alpha = \pi\sqrt{3}$ , we obtain the same main-lobe width as that of the Hann or Hamming window. Under this constraint and for a fixed window length  $N = 256$ , only the Hann window is able to resolve all the spectral components perfectly. In the case of Kaiser and Hamming windows, they can still resolve the closely spaced frequency components (since their main-lobe widths are smaller), but both fail to detect the far away weak signal component. This failure is due to the high side-lobe level as well as

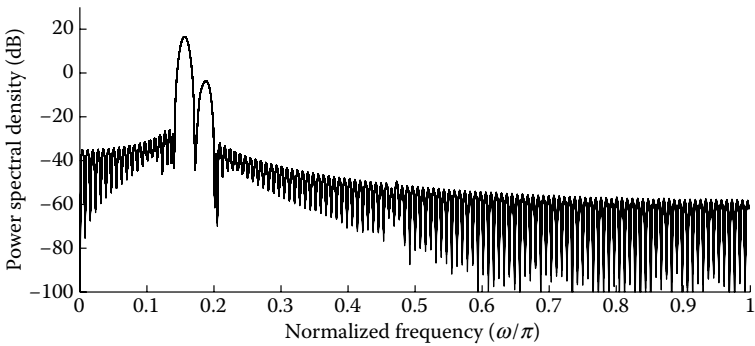


**FIGURE 8.28**

Comparison of performance of spectral estimator using different windows. (a) Hann window. (b) Zeroth-order Kaiser window (with  $\alpha = \pi\sqrt{3}$ ).



**FIGURE 8.29** Comparison of performance of spectral estimator using different windows. (a) Hamming window. (b) Zeroth-order Kaiser-Bessel window (with  $\alpha = 10.73$ ).



**FIGURE 8.30** Performance of spectral estimator using first-order Kaiser-Bessel window.

low side-lobe roll-off rate of Hamming and Kaiser–Bessel windows. However, in the case of the Kaiser–Bessel window, we can change the parameter  $\alpha$  such that the required side-lobe level can detect the weaker signal component. This is shown in Figure 8.29(b), where it is clear that as the value of  $\alpha$  is increased, the side-lobe level reduces at the expense of increased main-lobe width, which in turn results in merging of the nearby signal components. This reduces the resolution of the spectral estimation method. Figure 8.30 presents the spectrum of the signal given in Equation 8.67 using the first-order Kaiser–Bessel window (with  $\gamma = 2\pi$ ). This window has the same main-lobe width as that of the Hann and Hamming windows. Under such circumstances, this window also fails to detect all the spectral components distinctly. The superior performance of the Hann window is due to the fact that the side-lobe roll-off rate is  $-18$  dB/octave, when compared to  $-6$  dB/octave for the Kaiser–Bessel window. Therefore, in most of the spectral analysis applications, the Hann window is preferred especially when the signal contains a weak signal component, which is farther away from the other components.

---

## References

1. S.L. Marple Jr., *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
2. S.M. Kay, *Modern Spectral Estimation: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
3. M.H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons Inc., New York, NY, 1996.
4. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proceedings*, vol. 66, no. 1, pp. 51–83, January 1978.
5. S.J. Orfanidis, *Introduction to Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1996.
6. J.F. Kaiser and R.W. Schafer, On the use of the  $I_0$ -sinh window for spectral analysis, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, no. 1, pp. 105–107, February 1980.
7. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectra from the Point of View of Communication Engineering*, Dover Publications, Inc., New York, 1958.
8. P. Stoica and R. Moses, *Spectral Analysis of Signals*, PHI Learning Private Limited, New Delhi, 2011.
9. A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, 3rd Edn., Prentice-Hall, Upper Saddle River, NJ, 2010.



# 9

---

## *Applications of Windows*

---

---

### 9.1 Windows in High Range Resolution Radars

The term “radar” stands for radio detection and ranging. As the name suggests, the primary function of most of the radars is to find the range of certain target objects. When two or more such targets are very close, it becomes difficult to identify them as individual targets. Radar that overcomes such a difficulty is said to have good range resolution capability. In radar applications, it is desirable to have high range resolution (HRR) [1], while maintaining adequate average transmitted power (ATP). This is accomplished by a technique called pulse compression, as a part of which, either the frequency modulation (FM) or the phase modulation (PM), is employed. FM can in turn have variants that use one of the following waveforms: linear FM (LFM) waveform, frequency-modulated continuous waveform, and stepped frequency waveform (SFW). One undesirable effect of pulse compression is that side lobes appear at the output. This problem can be solved by making use of window functions. It is known from the earlier chapters that a proper choice of a window can considerably reduce the side-lobe effect. LFM pulse compression can be implemented either by correlation processing (mainly used for narrow-band and some medium-band applications) or by stretch processing (used for wideband applications). The use of SFW is known to produce HRR target profiles. We will now proceed to see how exactly windowing is used in (i) obtaining HRR target profiles and (ii) stretch processing. Furthermore, we shall illustrate the effect of different windows on pulse compression using computer simulations for various scenarios.

#### 9.1.1 HRR Target Profiling

Consider a case where a series of  $N$  narrow-band pulses are transmitted, such that the frequency is stepped up or down by a fixed-value  $\delta f$  from pulse to pulse. Each group of  $N$  pulses can be referred to as a burst. Let  $\tau'$  and  $T$  denote the pulse width and pulse repetition interval (PRI), respectively.

Each pulse may employ some form of modulation, say LFM or PM. Let the center frequency of the  $n$ th pulse of a burst be given by

$$f_n = f_0 + n(\delta f), \quad n \in [0, N - 1]. \quad (9.1)$$

Then, the corresponding transmitted waveform is given by

$$S_n(t) = \begin{cases} C_n \cos(2\pi f_n t + \theta_n), & t \in [nT, nT + \tau'] \\ 0, & \text{otherwise} \end{cases}$$

where  $C_n$  is a constant and  $\theta_n$  is the relative phase. A target located at range  $R_0$  and at time  $t = 0$ , with a radial velocity  $v$  has a round-trip delay of  $\tau(t)$ , which is given by

$$\tau(t) = \frac{R_0 - vt}{0.5c}, \quad (9.2)$$

where  $c$  is the velocity of light ( $3 \times 10^8$  m/s). Then, the received signal from such a target is given by

$$S_{rn}(t) = C'_n \cos(2\pi f_n(t - \tau(t)) + \theta_n), \quad t \in [nT + \tau(t), nT + \tau(t) + \tau'], \quad (9.3)$$

where  $C'_n$  is a constant.

Before we can analyze the received signal, it has to be first down-converted to base-band. By mixing the received signals with the waveform  $y_n = C \cos(2\pi f_n t + \theta_n)$ , we obtain

$$S_{rn} y_n = \frac{C'_n C}{2} [\cos(2\pi f_n(2t - \tau) + 2\theta_n) + \cos(2\pi f_n \tau)]. \quad (9.4)$$

Upon lowpass filtering, it will result in the following in-phase component:

$$X_I = \frac{C'_n C}{2} \cos\left(2\pi f_n \left(\frac{2R_0 - 2vt}{c}\right)\right). \quad (9.5)$$

The quadrature component can be obtained by considering the product of  $S_{rn}$  with  $y_n$  phase-shifted by  $90^\circ$ , that is,  $-C \sin(2\pi f_n t + \theta_n)$  which results in

$$-\frac{C'_n C}{2} [\sin(2\pi f_n(2t - \tau) + 2\theta_n) + \sin(2\pi f_n \tau)]. \quad (9.6)$$

After applying a lowpass filter, we obtain the quadrature component as

$$X_Q = -\frac{C'_n C}{2} \sin(2\pi f_n \tau). \quad (9.7)$$

The samples of the target's reflection due to a single burst is given by

$$X_n = 0.5C'_n C(X_I + jX_Q), \quad (9.8)$$

whose IDFT is defined as

$$H_l = \frac{1}{N} \sum_{n=0}^{N-1} 0.5C'_n C e^{j2\pi \left( \frac{ln}{N} - 2\frac{ln}{c} (R_0 - vt) \right)}, \quad 0 \leq l < N - 1. \quad (9.9)$$

This provides information about the range profile. Normalizing  $H_l$ , by taking  $0.5C'_n C = 1$  and substituting  $n(\delta f_n)$  for  $f_n$ , we obtain

$$H_l = \sum_{n=0}^{N-1} e^{j\left(\frac{2\pi n}{N}\right)\left(l - \frac{2NR_0(\delta f_n)}{c}\right)}. \quad (9.10)$$

Denoting  $\eta = \left(l - \frac{2NR_0(\delta f_n)}{c}\right)$  in the previous equation, we arrive at the following geometric series:

$$H_l = \sum_{n=0}^{N-1} e^{j\left(\frac{2\pi n\eta}{N}\right)}, \quad (9.11)$$

with the first term being unity and having a common ratio of  $e^{j\left(\frac{2\pi\eta}{N}\right)}$ . Therefore

$$H_l = \frac{1 - e^{j2\pi\eta}}{1 - e^{j\frac{2\pi\eta}{N}}} = \frac{\sin \pi\eta}{\sin \frac{\pi\eta}{N}} e^{j\pi\eta\left(1 - \frac{1}{N}\right)}. \quad (9.12)$$

From this, it can easily be seen that the synthesized range profile,  $\left| \frac{\sin \pi\eta}{\sin \frac{\pi\eta}{N}} \right|$ , has high side-lobe amplitudes, which can be reduced by performing the windowing operation, prior to taking the IDFT.

### 9.1.2 Simulation Results

Simulations have been carried out to plot the range profiles of targets for two specific cases, where the relative distances between the three targets is taken as follows:

Case 1: (908, 910, 910.2) m (two of the targets very close)

Case 2: (908, 910, 912) m (targets reasonably apart)

Among the various window functions discussed in ref. [2], we now use the following: rectangular, triangular, Hamming, Hann, Blackman, and Nuttall window functions, to study the relative performance. The Nuttall window function [3] that is in-built in MATLAB is in fact the minimum four-term



Blackman–Harris window, which is given by

$$f[n] = a_0 + a_1 \cos\left(\frac{2\pi n}{N}\right) + a_2 \cos\left(\frac{4\pi n}{N}\right) + a_3 \cos\left(\frac{6\pi n}{N}\right), \quad 0 \leq n \leq N \quad (9.13)$$

where  $(a_0, a_1, a_2, a_3) = (0.3635819, -0.4891775, 0.1365995, -0.0106411)$ .

The simulation results are shown in Figures 9.1(a) through (f) for Case 1 and in Figures 9.2(a) through (f) for Case 2. The subplots shown in Figure 9.1 are supposed to show the three peaks corresponding to the three targets. However, we note in the subplots (other than the rectangular case in Figure 9.1(a)) that the small amplitude corresponding to the third target makes it very difficult to distinguish it from the peak due to the presence of target 2. This can be attributed to the following two reasons:

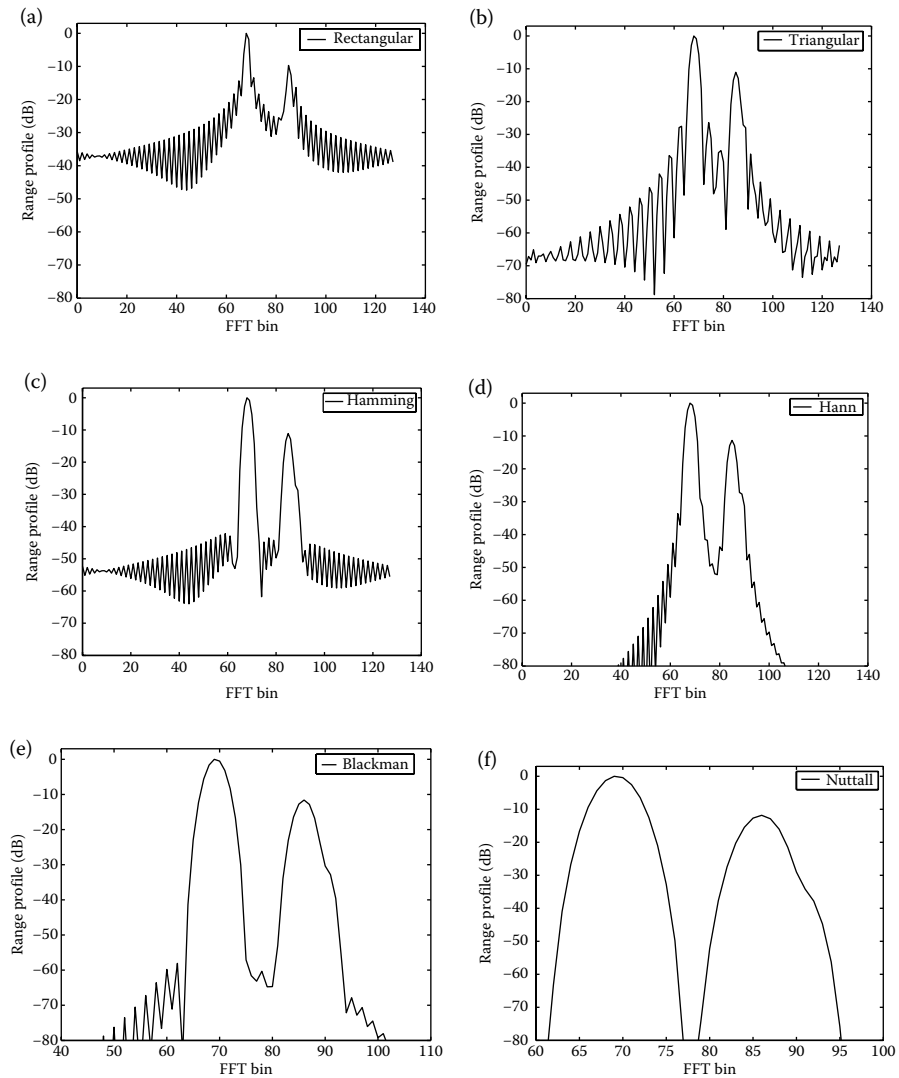
1. The radar cross section (RCS) for the three targets are taken as  $(100, 10, 1) \text{ m}^2$ , and hence the amplitudes of the peaks appear to be decreasing from the left to the right in each subplot of Figure 9.1.
2. The side lobes associated with the second target suppress the peak of the third target. This can be clearly observed in Figure 9.1.

In Figure 9.2 (corresponding to Case 2), where the targets are reasonably apart, all the windows considered are capable of resolving the three targets. However, we observe from Figures 9.2(a) through (d) that the Hann window (Figure 9.2(d)) has some advantage over the rectangular, triangular, and Hamming windows in such a scenario, since the side-lobe roll-off rate is relatively higher in the case of the Hann window.

---

## 9.2 Effect of Range Side Lobe Reduction on SNR

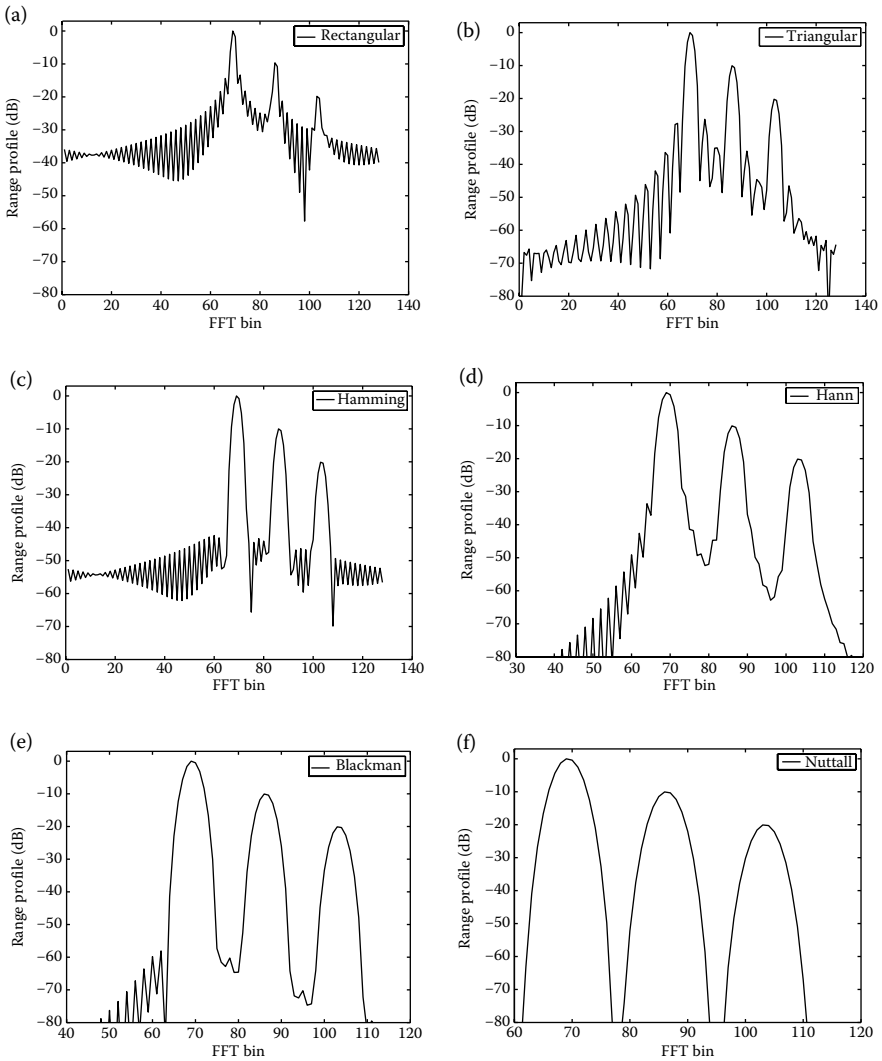
Pulse compression is required to achieve the twin goals of HRR [1] and the maintenance of adequate ATP. LFM waveform is used often in several applications that rely on pulse compression. It is well known that the matched filter waveform associated with the LFM pulse compression signal essentially has the *sinc* shape, with range side lobes extending on either side of the compressed pulse. These Doppler side lobes may be partially controlled by varying the amplitudes of the pulses upon transmission and/or reception. However, this reduces the SNR as well as range resolution, under peak power limitations. In this section, we provide the general expressions for the loss factor for all the three cases. The numerical results are presented for physically realizable weighting functions that possess desirable characteristics.



**FIGURE 9.1** Range profiles for various windows when the relative distance of three targets is taken as [908, 910, 910.2] m. (Targets 2 and 3 are very close.) (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

### 9.2.1 Introduction

The matched filters for a linear FM pulse compression radar are designed to yield a sharp pulse output. The peak amplitude of the pulse depends upon the target cross section. In a multiple-target environment,  $N$  targets of different cross sections must be observed by the pulse compression radar. The



**FIGURE 9.2**

Range profiles for various windows when the relative distance of three targets is taken as [908, 910, 912] m. (Targets are reasonably apart.) (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

matched filter generates  $N$  sharp pulses; the peak amplitude of the  $n$ th pulse being proportional to the cross section of the  $n$ th target. However, owing to the finite-duration nature of the input data, each sharp pulse (main lobe) is surrounded by secondary side lobes (minor lobes). These are also called range side lobes in this context, which can be fairly high in amplitude.

Consider, for example, a uniform weighting function  $f[n]$  of unit height and length  $N$ . The magnitude function of the DTFT of  $f[n]$  is  $(\sin(0.5\omega N))/(\sin(0.5\omega))$ , from which it can be shown that the first (and the largest) of the range side lobes is 13.2 dB below the peak of the compressed pulse. It has a decay rate of 6 dB/octave (or equivalently 20 dB/decade), which is rather slow. Hence, when a radar is processing signals from several targets of different cross sections, the main lobes of smaller targets can be masked by the side lobes of a stronger target. This obviously makes the detection of smaller targets difficult.

The situation could be improved by appropriately weighting either the transmitted or the received waveform. However, there is a trade-off in terms of loss in SNR and range resolution. A better control of the side lobes may be obtained by time-weighting the transmitted waveform, as well as weighting upon reception [4]. However, weighting at both sides will lead to greater loss in SNR when compared to only receiver weighting, in obtaining the same range side lobes [3]. Urkowitz et al. [4] have extended the results of Ref. [5] to a sequence of pulses, in which the pulse height (amplitude of the pulse) is weighted, again with peak height limitation. The purpose of this section is to apply the formulae given by Urkowitz to several realizable weighting functions [4,5], which provide excellent characteristics. A general expression is derived for the loss factor under peak power limitations. The loss factor formulae are provided for the following three cases:

*Case A:* Full weighting upon reception, with uniform weighting on transmission

*Case B:* Square root of the nominal weighting on transmission and reception

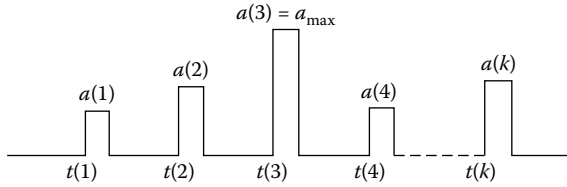
*Case C:* Full weighting on both transmission and reception

The loss factor is defined as the ratio of the maximum achievable SNR to the actual SNR. We have experimented with many weighting functions that are simple to implement. The general characteristics of the weighting functions pertaining to range side lobe reduction are also tabulated.

### 9.2.2 Loss Factor

To obtain the general expression for the loss factor (LF) [4], it is necessary to get the actual SNR as well as the maximum possible SNR. The maximum SNR is obtained when the weighting is uniform (or no weighting), both upon transmission and reception. The loss factor is computed as detailed below:

Consider a sequence of  $N$  transmitted pulses of amplitudes  $a[k]$ ,  $k = 1, 2, \dots, N$ , occurring at the time instances  $t[k]$  (Figure 9.3). One of the pulses will have the maximum amplitude such that  $a[k] = a_{\max}$ , for some  $k$ . The pulse sequence might undergo pulse compression. The sequence of the compressed



**FIGURE 9.3**  
Typical sequence of pulses.

pulse is given by

$$a [1] a [p] , a [2] a [p] , \dots , a [N] a [p] . \tag{9.14}$$

Some amount of noise, having a variance of  $\sigma_0^2$ , may also get added to the sequence above. This process is illustrated in Figure 9.4. These pulses will be again weighted by the reference sequence  $r[k]$ . With  $n[k]$  representing the noise, the final output of this scheme will be of the form:

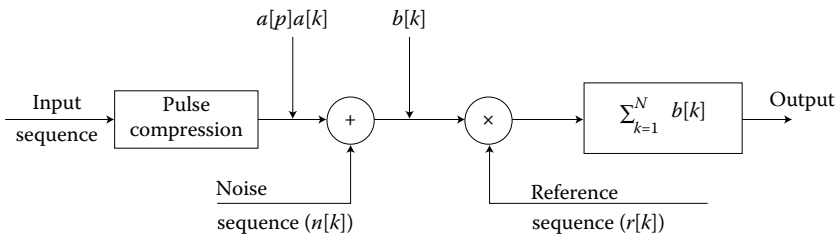
$$\sum_{k=1}^N b[k] = \sum_{k=1}^N r[k](a[k]a[p] + n[k]). \tag{9.15}$$

The output signal component is

$$\text{Signal} = a[p] \sum_{k=1}^N a[k]r[k] \tag{9.16}$$

and the output noise variance, since all are statistically independent, is

$$\text{Noise} = \sigma_0^2 \sum_{k=1}^N r^2[k]. \tag{9.17}$$



**FIGURE 9.4**  
Weighting and summing of received pulse sequence. (Redrawn from C.L. Temes, Side lobe suppression in a range-channel pulse-compression radars, *IRE Transactions on Military Electronics*, vol. MIL-6, pp. 162–169 April, 1962.)

The SNR ratio is defined by the ratio of square of Equations 9.16 and 9.17, which is given by

$$\text{SNR} = \frac{a^2[p]}{\sigma_0^2} \frac{\left(\sum_{k=1}^N a[k]r[k]\right)^2}{\sum_{k=1}^N r^2[k]}. \quad (9.18)$$

The maximum SNR is obtained by substituting  $a[k] = a_{\max}$  and  $r[k] = 1$ , that is

$$(\text{SNR})_{\max} = Na_{\max}^2 \frac{a^2[p]}{\sigma_0^2}, \quad (9.19)$$

and the LF is obtained by taking the ratio of Equation 9.19 to Equation 9.18 and is obtained as

$$\text{LF} = \frac{Na_{\max}^2 \sum_{k=1}^N r^2[k]}{\left(\sum_{k=1}^N a[k]r[k]\right)^2}. \quad (9.20)$$

Note that LF is always  $>1$ . The loss factor is calculated from Equation 9.20 for the following three cases.

*Case A:* Full nominal weights upon reception and uniform weighting on transmission, that is

$$a[k] = 1 \quad \text{and} \quad r[k] = f[k], \quad (9.21)$$

where  $f[k]$  represents nominal weights. The LF is given by

$$\text{LF}_1 = \frac{N \sum_{k=1}^N f^2[k]}{\left(\sum_{k=1}^N f[k]\right)^2}. \quad (9.22)$$

*Case B:* Square root of the nominal weights upon transmission and reception, that is

$$a[k] = a_{\max} \sqrt{f[k]/f_{\max}} \quad \text{and} \quad r[k] = \sqrt{f[k]}. \quad (9.23)$$

Then, we obtain the corresponding loss factor as

$$\text{LF}_2 = \frac{N \sum_{k=1}^N f_{\max}}{\sum_{k=1}^N f[k]}. \quad (9.24)$$

Here,  $f_{\max}$  represents the maximum value of  $f[k]$ .

Case C: Full nominal weights on both transmission and reception. In this case

$$a[k] = a_{\max}f[k]/f_{\max} \quad \text{and} \quad r[k] = f[k]. \quad (9.25)$$

In such a situation, we obtain the loss factor  $LF_3$  as

$$LF_3 = \frac{Nf_{\max}^2}{\sum_{k=1}^N f^2[k]}. \quad (9.26)$$

It has been shown in Ref. [4] that  $LF_1 \leq LF_2 \leq LF_3$ . which can easily be verified from Tables 9.1 and 9.2. It is interesting to note that  $LF_1, LF_2,$  and  $LF_3$

**TABLE 9.1**  
Loss Factor for Different Weighting Functions

Function Number	Weighting Constants	Number of Pulses	LF <sub>1</sub> (dB)	LF <sub>2</sub> (dB)	LF <sub>3</sub> (dB)
1	A = 0.5	4	1.7609	2.3226	2.8843
	B = 0.5	8	1.7609	2.8418	3.9226
	C = 0.0	16	1.7609	2.9684	4.1758
	D = 0.0	32	1.7609	2.9998	4.2388
2	A = 0.54	4	1.3444	2.0476	2.7507
	B = 0.46	8	1.3444	2.5213	3.6981
	C = 0.0	16	1.3444	2.6375	3.9306
	D = 0.0	32	1.3444	2.6664	3.9885
3	A = 0.53836	4	1.3597	2.0584	2.7570
	B = 0.46164	8	1.3597	2.5339	3.7081
	C = 0.0	16	1.3597	2.6506	3.9414
	D = 0.0	32	1.3597	2.6796	3.9995
4	A = 0.42	4	2.3264	2.6524	2.9784
	B = 0.5	8	2.3723	3.4919	4.6115
	C = 0.08	16	2.3723	3.6988	5.0253
	D = 0.0	32	2.3723	3.7503	5.1284
5	A = 0.375	4	2.7621	2.8843	3.0065
	B = 0.5	8	2.8880	3.9226	4.9573
	C = 0.125	16	2.8880	4.1758	5.4637
	D = 0.0	32	2.8880	4.2388	5.5895
6	A = 0.44959	4	2.0487	2.4954	2.9420
	B = 0.49364	8	2.0703	3.2298	4.3893
	C = 0.05677	16	2.0703	3.4115	4.7526
	D = 0.0	32	2.0703	3.4567	4.8432

*continued*

**TABLE 9.1 (continued)**

Loss Factor for Different Weighting Functions

Function Number	Weighting Constants	Number of Pulses	LF <sub>1</sub> (dB)	LF <sub>2</sub> (dB)	LF <sub>3</sub> (dB)
7	A = 0.4243801	4	2.2704	2.6214	2.9724
	B = 0.497396	8	2.3140	3.4501	4.5862
	C = 0.0782793	16	2.3140	3.6545	4.9951
	D = 0.0	32	2.3140	3.7055	5.0970
8	A = 0.355768	4	2.7658	2.8862	3.0067
	B = 0.487396	8	3.0562	4.0924	5.1287
	C = 0.144232	16	3.0562	4.3896	5.7231
	D = 0.012604	32	3.0562	4.4637	5.8712
9	A = 0.3635819	4	2.7095	2.8571	3.0047
	B = 0.4891775	8	2.9581	4.0138	5.0695
	C = 0.1365995	16	2.9581	4.2992	5.6403
	D = 0.0106411	32	2.9581	4.3703	5.7825

**TABLE 9.2**

Loss Factor of Some Additional Weighting Functions

Function Number	Weighting Constants	Number of Pulses	LF <sub>1</sub> (dB)	LF <sub>2</sub> (dB)	LF <sub>3</sub> (dB)
1	A = 0.42323	4	2.2815	2.6275	2.9736
	B = 0.49755	8	2.3262	3.4605	4.5948
	C = 0.07922	16	2.3262	3.6660	5.0057
	D = 0.0	32	2.3262	3.7172	5.1081
2	A = 0.35875	4	2.8958	2.9527	3.0095
	B = 0.48829	8	2.9704	3.9933	5.0162
	C = 0.14128	16	2.9704	4.2411	5.5118
	D = 0.01168	32	2.9704	4.3023	5.6342
3	A = 0.40217	4	2.4784	2.7351	2.9918
	B = 0.49703	8	2.5309	3.6270	4.7231
	C = 0.09392	16	2.5309	3.8453	5.1598
	D = 0.00188	32	2.5309	3.8996	5.2684
4	A = 0.40897	4	2.4238	2.7056	2.9874
	B = 0.5	8	2.4849	3.5925	4.7000
	C = 0.09103	16	2.4849	3.8107	5.1364
	D = 0.0	32	2.4849	3.8650	5.2450
5	A = 0.338946	4	3.1974	3.1029	3.0084
	B = 0.481973	8	3.2742	4.1985	5.1228
	C = 0.161054	16	3.2742	4.4551	5.6361
	D = 0.018027	32	3.2742	4.5183	5.7623



are in geometric progressions, with  $LF_2$  being the geometric mean of the other two, that is  $LF_1$  and  $LF_3$ .

### 9.2.3 Weighting Function

The generalized weighting function is given by

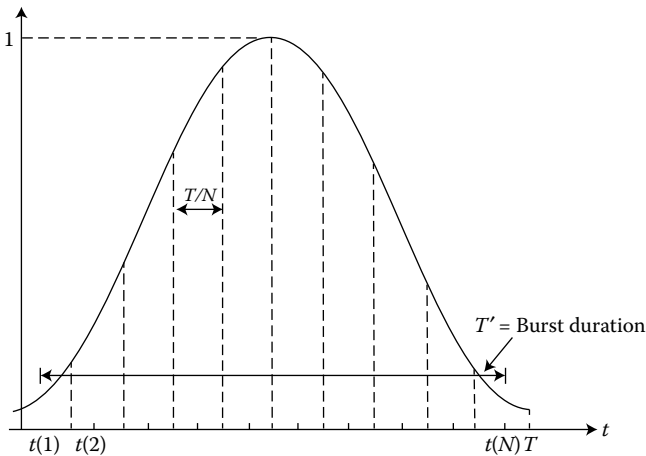
$$f(t) = A - B \cos\left(\frac{2\pi t}{T}\right) + C \cos\left(\frac{4\pi t}{T}\right) - D \cos\left(\frac{6\pi t}{T}\right), \quad 0 \leq t \leq T. \quad (9.27)$$

The minus sign results due to the shift in the origin of the weighting function. A typical weighting function is shown in Figure 9.5. Here,  $T$  represents the duration of the weighting function and  $T'$  is the burst duration. The sampling instances are represented by  $t(1), t(2), \dots, t(N)$ . The  $k$ th sampling time, for uniform spacing, is given by

$$t[k] = \frac{(2k - 1)T}{2N}, \quad k = 1, 2, \dots, N. \quad (9.28)$$

Using Equation 9.28 in Equation 9.27, we get

$$f[k] = f[t[k]] = A - B \cos\left[\frac{\pi(2k - 1)}{N}\right] + C \cos\left[\frac{2\pi(2k - 1)}{N}\right] - D \cos\left[\frac{3\pi(2k - 1)}{N}\right], \quad k = 1, 2, \dots, N. \quad (9.29)$$



**FIGURE 9.5**  
A typical weighting function.

For regular spacing and when  $N$  is even,  $f_{\max}$  is given by

$$f_{\max} = A - B \cos \left[ \frac{\pi(N+1)}{N} \right] + C \cos \left[ \frac{2\pi(N+1)}{N} \right] - D \cos \left[ \frac{3\pi(N+1)}{N} \right]. \quad (9.30)$$

The loss factors for all the three cases considered are calculated for different weighting constants,  $A, B, C$ , and  $D$ , and are given in Tables 9.1 and 9.2. The loss factors are expressed in dB, that is,  $10 \log_{10}(\text{LF})$ . These values are tabulated for different numbers of pulses as well. As an illustration to compute various entries of Table 9.1, we consider the Hamming window of size  $N = 4$ . Then, Equation 9.29 can be written as

$$f[k] = 0.54 - 0.46 \cos \frac{\pi(2k-1)}{N}, \quad k = 1, 2, 3, 4. \quad (9.31)$$

$$f[1] = f[4] = 0.2147; \quad f[2] = f[3] = 0.8653 = f_{\max}.$$

Using Equation 9.22,

$$\text{LF}_1 = 4 \left( \frac{1.5896}{4.6656} \right) = 1.3628. \quad (9.32)$$

This is equivalent to 1.3444 dB. From Equation 9.24, we get

$$\text{LF}_2 = 4 \left( \frac{0.8653}{2.1600} \right) = 1.6024, \quad (9.33)$$

which when expressed in dB corresponds to 2.0476 dB. These values are tabulated in Table 9.1, under function no. 2. The other entries of Tables 9.1 and 9.2 can be computed on similar lines. The various combinations of the quadlet ( $A, B, C, D$ ) have been taken from Refs. [2,3]. For example, function nos. 1, 2, and 4 correspond to the Hann, Hamming, and Blackman windows, respectively. Apart from LF, the other factors to be considered in the selection of weighting function in the range side lobe reduction are

1. Main-lobe broadening factor given by the ratio

$$\frac{\text{Half-power bandwidth of weighting function}}{\text{Half-power bandwidth of uniform weighting}}$$

2. First side-lobe and peak side-lobe levels.

**TABLE 9.3**  
Weighting Function Data

S. No.	Main-Lobe Broadening Factor	First Side Lobe (dB)	Peak Side Lobe (dB)	Integration Loss (dB)	Side-Lobe Decay Rate (dB/octave)
1	1.63	-31.47	-31.47	-1.76	18
2	1.48	-44.05	42.69	-1.34	6
3	1.48	-43.19	-43.19	-1.36	6
4	1.86	-58.12	-58.12	-2.37	18
5	2.12	-46.75	-46.75	-2.89	30
6	1.73	-74.52	-62.05	-2.07	6
7	1.83	-71.48	-71.48	-2.31	6
8	2.17	-93.36	-93.32	-3.06	18
9	2.12	-98.34	-98.14	-2.96	6

3. The integration loss defined as

$$10 \log_{10} \left[ \frac{\frac{1}{2T} (\int_0^{2T} f(t) dt)^2}{\int_0^{2T} f^2(t) dt} \right]. \tag{9.34}$$

4. The decay rate of the side lobes, which can be expressed in dB/octave. These are given in Table 9.3 for all the cases considered in Table 9.1.

**9.2.4 Results and Discussions**

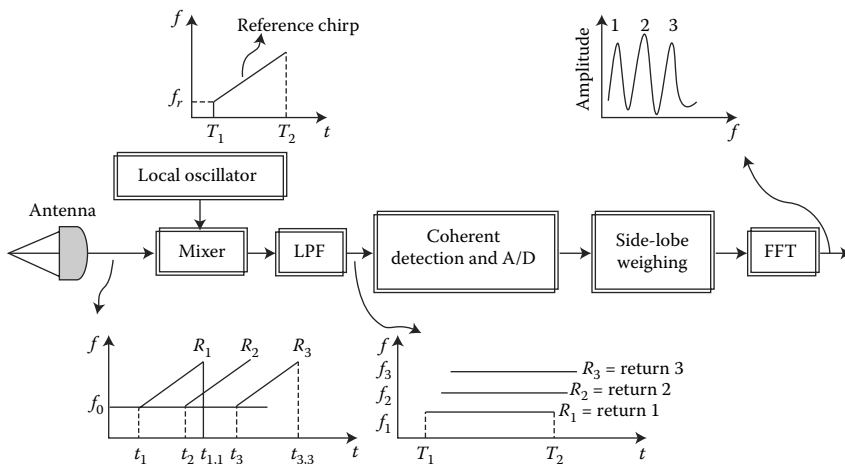
A larger loss factor indicates a smaller SNR. The three cases considered here are in the order of increasing loss functions. The main-lobe-broadening factor indicates the loss in frequency resolution due to the effect of weighting functions. Hence, a weighting function having better side-lobe rejection and minimum main-lobe-broadening factor will be desirable. The integration loss is another factor that indicates the loss in SNR incurred due to weighting. The side-lobe decay rate is also yet another powerful indicator that must be considered. It is desirable that the side-lobes decay at a faster rate, to preserve the dynamic range between the two targets widely spaced in frequency. In conclusion, a weighting function having low loss factor (for a particular case), low side-lobe level, smaller main-lobe broadening factor, and faster decay rate of the side-lobe envelope is required for range side-lobe reduction in a linear FM pulse compression radar. Though Hamming window is considered as a reasonable choice on an average, the choice of a window for a particular application has to be decided on a case-to-case basis, keeping in view the

window properties. The Kaiser and ultraspherical windows have the additional advantage that there is a flexibility to vary the different parameters and get closer to the desired characteristics.

### 9.3 Window Functions in Stretch Processing

In radar applications that employ extremely high-bandwidth LFM signals, a technique called stretch processing is used at the receiving end [1]. This facilitates proper decision making (such as the presence/absence of targets, their count, etc.) based on which subsequent action can be taken. The term “proper decision making” here refers to the ability to correctly detect the presence of targets, along with other desired parameters (say, range, radial velocity, etc.). As seen from the block diagram given in Figure 9.6, the received signal is mixed with a reference signal (which is a replica of the transmitted signal) and is passed through a lowpass filter.

This results in constant tones corresponding to the positions of the targets. Let the reference signal be an LFM waveform and the starting frequency of the chirp signal used be  $f_r$ . The instantaneous frequency after lowpass filtering will be proportional to the target range. Hence, if a peak at frequency  $f_1$  implies the presence of a target at range  $R$ , then the presence of a peak at frequency  $f_2$  indicates that a target at a range  $\left(\frac{Rf_2}{f_1}\right)$  is present. In case the radar receives echoes from some targets that are very close (in time or



**FIGURE 9.6** Block diagram of stretch processing. (Redrawn from B.R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, CRC Press, 2nd edn., 2005.)

range), then identifying them as separate targets depends on the resolution capability of the system. The task of identifying relatively close objects is often difficult.

Let the normalized transmitted signal be of the form:

$$s(t) = \cos(2\pi(f_r t + 0.5\mu t^2)), \quad t \in [0, \tau], \quad (9.35)$$

where  $f_r$  = starting frequency of the chirp signal,  $\mu = \frac{B}{\tau}$  is the LFM coefficient,  $\tau$  is the chirp duration, and  $B$  is the chirp bandwidth. Assuming the range of the  $n$ th scatterer be  $R_n$ , the output of the lowpass filter will be

$$s_o(t) = \sum_{n=1}^N a_n \cos\left(4\pi\left(\frac{2R_n B t}{c\tau'} + \frac{f_o R_n}{c} - \frac{R_n^2 B}{c^2 \tau'}\right)\right), \quad (9.36)$$

where  $a_n$  is proportional to the  $n$ th target's cross section, antenna gain, and range attenuation.  $\tau'$  is relatively larger when compared to  $\frac{2R_n}{c}$ . To study the effect of various windows on detection, simulations are carried out to implement stretch processing using the following specifications [1]:

- i. Number of scatterers = 3
- ii. Uncompressed pulse width,  $\tau = 0.01$  s
- iii.  $f_o = 5.6$  GHz
- iv. Chirp bandwidth = 1 GHz
- v. Range receive window,  $R = 30$  m
- vi. Vector of scatterer's range

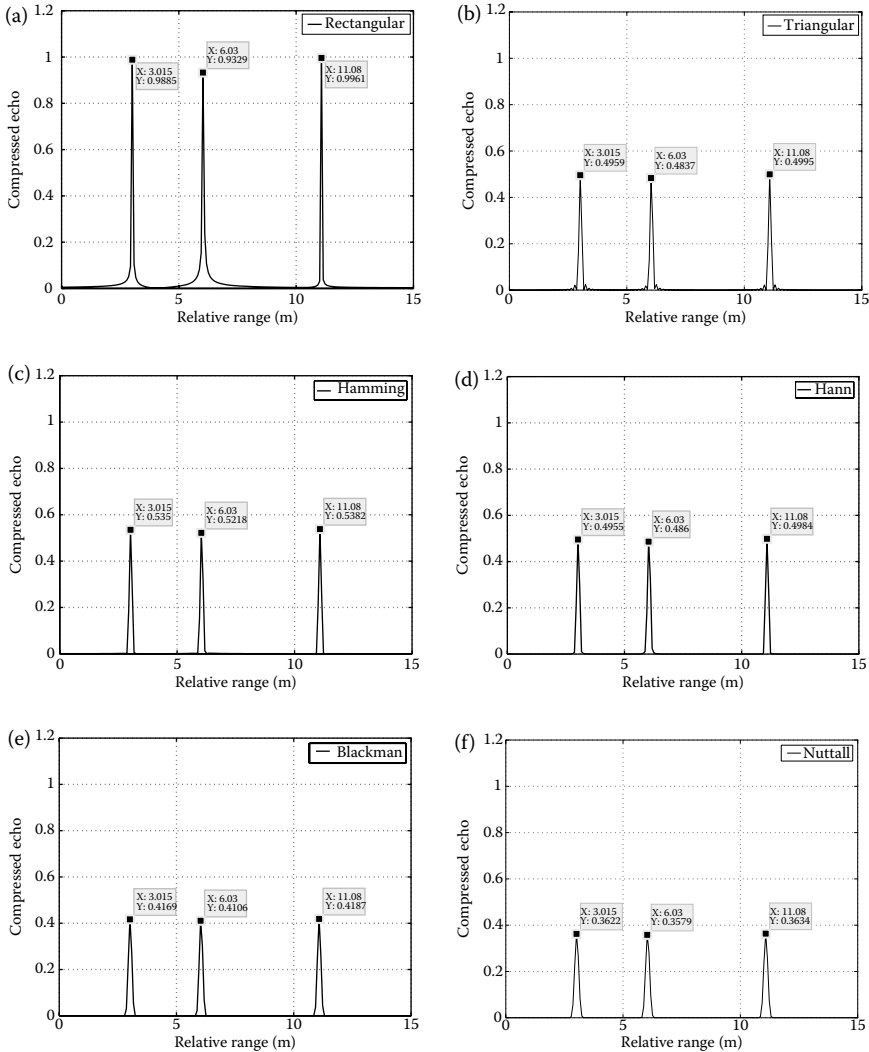
*Case 1:* Targets spaced at (3, 6, 11) m (reasonably apart), RCS = (1, 1, 1) m<sup>2</sup> (Figure 9.7)

*Case 2:* Targets spacing is (3, 11, 11.12) m (two of them are close, i.e.,  $11.12 - 11 < \frac{R_r}{2}$ ), RCS = (1, 1, 1) m<sup>2</sup> (Figure 9.8)

*Case 3:* Targets spaced reasonably apart at (3, 6, 11) m, RCS = (1, 1, 2) m<sup>2</sup> (Figure 9.9)

*Case 4:* Targets spaced at (3, 11, 11.12) m (two of them are close) RCS = (1, 1, 2) m<sup>2</sup> (Figure 9.10)

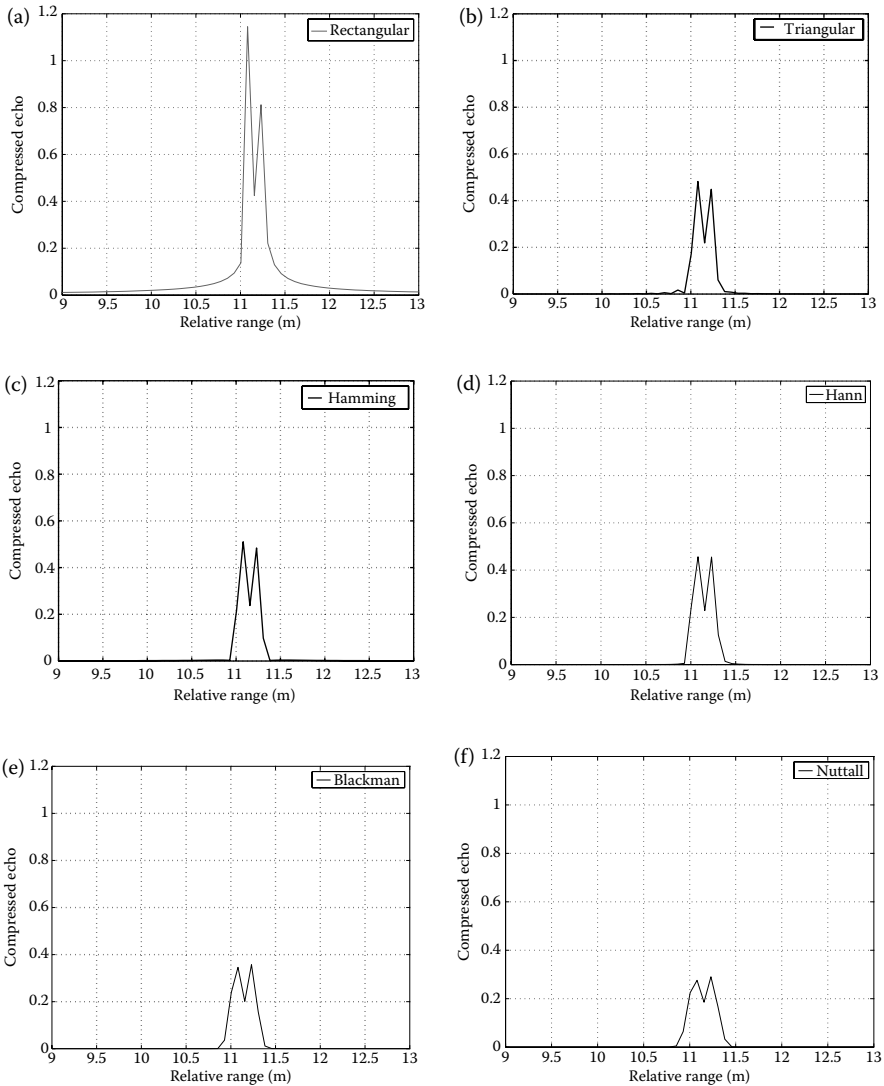
The corresponding results are shown in Figures 9.7 through 9.10, wherein subfigures (a) through (f) correspond to different window functions. From Figure 9.7, it can be seen that the peak locations obtained (3.015, 6.03, 11.08) m differ from the actual values (3, 6, 11) m. A similar shift can also be inferred



**FIGURE 9.7**

Compressed echo signals with relative ranges of three targets being (3, 6, 11) m (targets are reasonably apart) and their RCS given by (1, 1, 1) m<sup>2</sup>. (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

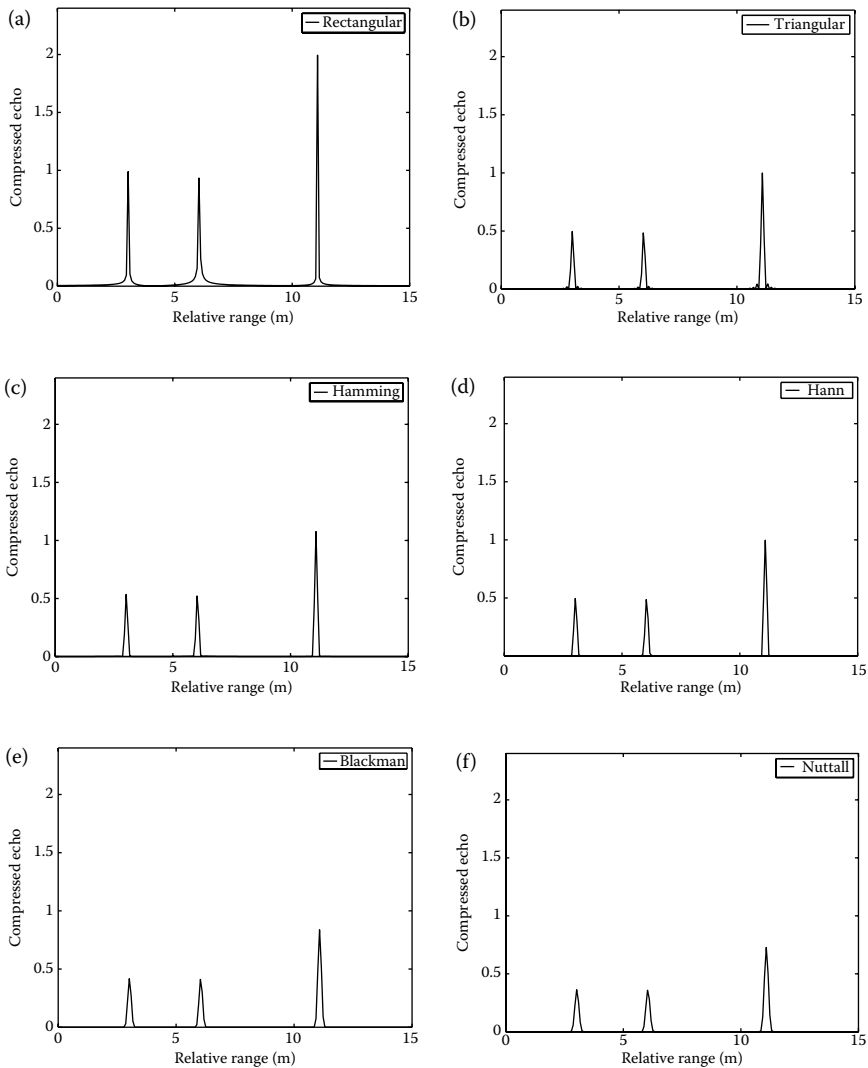
from Figure 9.10. One of the reasons for such a shift can be attributed to the finite precision arithmetic used in the computations. When the relative range tuple is (3, 6, 11) m, such that all the targets are well apart, then all the window functions considered (rectangular, triangular, Hamming, Hann, Blackman, and Nuttall) are capable of identifying each target distinctly (Figures 9.7 and 9.9). However, when the two targets are closer and are of equal strengths,



**FIGURE 9.8**

Compressed echo signals with relative ranges of three targets being (3, 11, 11.12) m (two of them very close) and their RCS given by (1, 1, 1) m<sup>2</sup>. (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

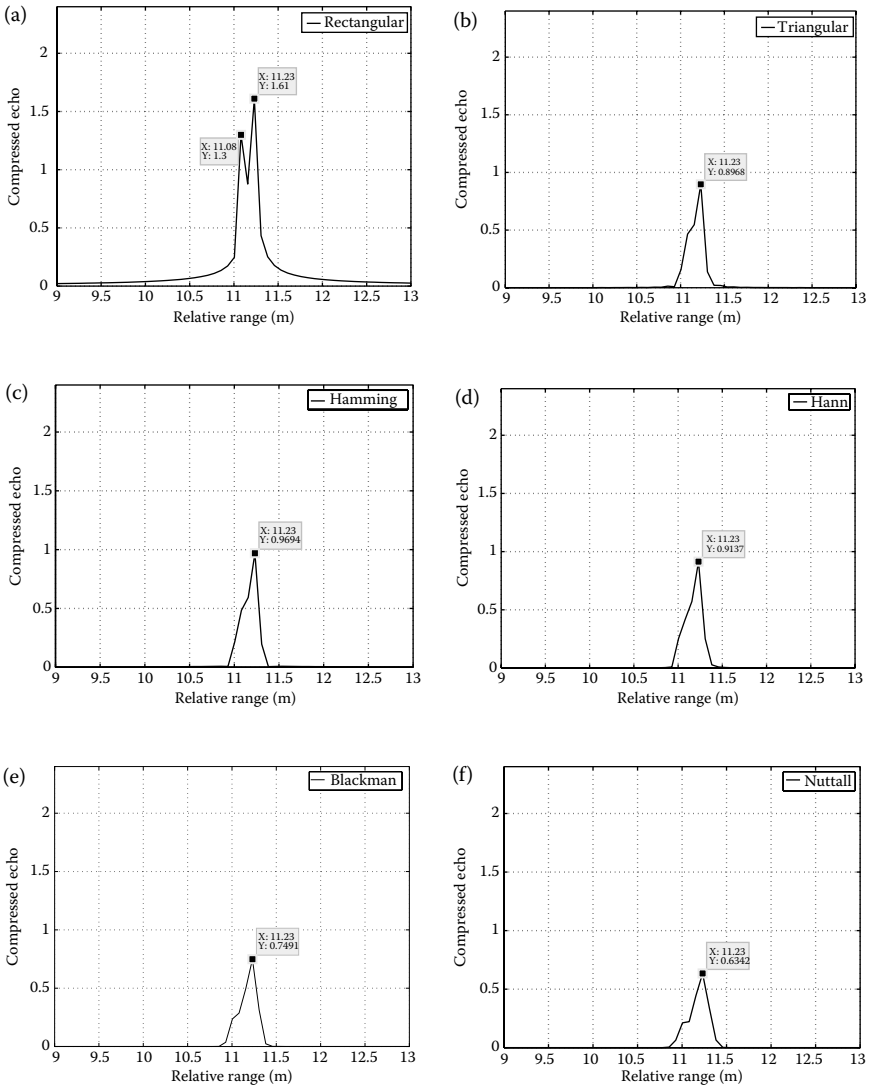
then, as can be seen in Figure 9.8(a), one of the peaks (the right-most in this case) is considerably smaller than the other peak. This can be attributed to the stronger side lobe of the rectangular window. Such a phenomenon cannot be observed in Figure 9.8(b) through (f). From Figures 9.8(c) and (d), Hamming window is preferred over the Hann window, as the difference (*peak* – *valley*)

**FIGURE 9.9**

Compressed echo signals with relative ranges of three targets being (3, 6, 11) m (reasonably apart) and their RCS given by (1, 1, 2)  $m^2$ . (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

is more in the case of the former than the latter. It is interesting to note that if closely spaced targets of unequal strengths are considered (Figure 9.10), then for the rectangular case, the two peaks are distinctly seen (due to the narrow main-lobe width), while the two peaks seem to merge into a single peak in the case of other window functions (which have relatively wider main-lobe widths).





**FIGURE 9.10** Compressed echo signals (as obtained by using various windows) with relative ranges of three targets being (3, 11, 11.12) m (two of them very close) and the corresponding RCS is given by the tuple (1, 1, 2) m<sup>2</sup>. (a) Rectangular. (b) Triangular. (c) Hamming. (d) Hann. (e) Blackman. (f) Nuttall.

### 9.4 Application of Window Functions in Biomedical Signal Processing

Organisms have many complex systems, which are in turn made up of several subsystems that carry out different physiological processes. Biomedical

signals are observations of these physiological processes, gene to protein sequences, neural to cardiac rhythms, and tissues and organ images. They convey the information about the corresponding biological system. Some of the commonly studied biomedical signals are action potential (AP) of cells, electroneurogram (ENG), electromyogram (EMG), electrocardiogram (ECG), electroencephalogram (EEG), event-related potential (ERP), electrogastrogram (EGG), phonocardiogram (PCG), vibromyogram (VMG), vibroarthrogram (VAG), electro-oculogram (EOG), electroretinogram (ERG), and so on. Table 9.4 lists the frequency ranges and dynamic ranges of a few biomedical signals [6].

In biomedical applications, mere acquisition of the signals is not sufficient; it is also required to process it to extract the relevant information. Biomedical signals get corrupted by several sources of errors such as the addition of noise due to imprecision of instruments or interference from power lines. Besides, the measured signal may also contain the interference from other systems, which may also be a complex function of the required and unnecessary signals. For example, in measuring the ECG of a fetus, the signal acquired is correlated with the mother's ECG. There are more complex processes that corrupt the relevant signal for diagnosis. Therefore, accurate processing of these signals is necessary for an optimal estimation of the signal and its parameters for proper diagnosis.

### 9.4.1 Biomedical Signal Processing

By now it is quite obvious that the processing of biomedical signals is extremely important. Even though these signals differ only in application, the processing has to be done with utmost care because making errors in

**TABLE 9.4**

List of Biomedical Signals, Their Frequency Ranges, and Their Dynamic Ranges

Signals	Frequency Range	Dynamic Range	Comments
AP	100 Hz–2 kHz	10 $\mu$ V–100 mV	Cell membrane potential
ENG	100 Hz–1 kHz	5 $\mu$ V–10 mV	Nerve bundle potential
ERG	0.2 Hz–200 Hz	0.5 $\mu$ V–1 mV	Evoked flash potential
EOG	0 Hz–100 Hz	10 $\mu$ V–5 mV	Corneal retinal potential
EEG (surface)	0.5 Hz–100 Hz	2 $\mu$ V–100 $\mu$ V	Scalp potential
EMG (single fiber)	500 Hz–10 kHz	1 $\mu$ V–10 $\mu$ V	AP from single muscle fiber
Surface EMG (skeletal)	2 Hz–500 Hz	50 $\mu$ V–5 mV	AP from skeletal muscle fiber
Surface EMG (smooth)	0.01 Hz–1 Hz	50 $\mu$ V–5 mV	AP from smooth muscle fiber
ECG	0.05 Hz–100 Hz	1 $\mu$ V–10 mV	AP from heart potential
High-frequency ECG	100 Hz–1 kHz	100 $\mu$ V–2 mV	Notches and slus superimposed on the ECG

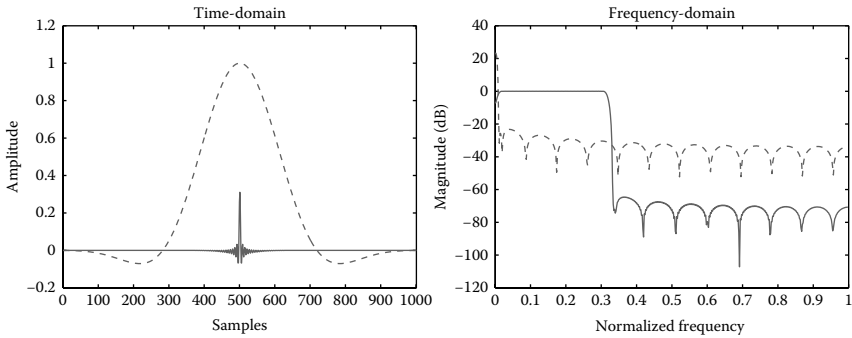
normal signals may lead to a minor damage (cost or product), but making errors in biomedical signal processing may lead to irreparable losses (may cost a life in the worst circumstances). The success of the application depends on the origin and knowledge of the signal. It requires special treatment and demands a clear understanding of the biomedical signal characteristics.

Biomedical signals are stochastic in nature and hence they cannot be predicted. With the acquired signal (discrete-time), processing has to be done to extract its information. There are several techniques that are directly used in biomedical signal processing. These include sampling, frequency-domain analysis, windowing, short-time Fourier transform (STFT), spectral estimation, signal enhancement, optimal filtering, adaptive filtering, segmentation of nonstationary signals, and so on. Of these techniques, we have considered only the applications that use window functions. The calculation of various functions that are used as major tools such as the power spectral density, auto-correlation, and cross-correlation requires the knowledge of the signal from  $-\infty$  to  $+\infty$ . This is, of course, impractical because the signal is not available for long durations. Therefore, we do not use the signal itself but a windowed signal. For computing STFT, windowing is used for breaking down long-duration signals into signals of shorter duration, thereby the characteristics of the signals during these intervals can be assumed to be stationary.

#### **9.4.2 FIR Filtering of Biomedical Signals**

During acquisition, the biomedical signals get corrupted by different types of artifacts and interferences such as power line interference, electrode contact noise, motion artifacts, muscle contraction, and so on. For a meaningful analysis of these signals, steps have to be taken to filter out all these noise sources. Here, we consider the effect of FIR filters designed from various window functions under additive white Gaussian noise (AWGN) conditions.

We consider an example of an ECG signal that is taken from the MIT-BIH Arrhythmia Database [7] and the signal used here is the modified limb lead II (MLII), obtained by placing the electrodes on the chest. The ECG signal is sampled at a rate of 360 Hz and then corrupted by white Gaussian noise at different SNR levels. To denoise the ECG signal, FIR bandpass filters with different window functions at the sampling frequency of 360 Hz have been designed. The filter passband was set at 3–55 Hz, the transition bandwidths are 0.1–3 Hz on the lower side and 56–58.9 Hz on the upper side, with a pass-band attenuation of 0.01 dB and a stop-band attenuation of 80 dB [8]. The length of all the designed FIR filters is 1001. The flat-top window and its frequency response are shown in Figure 9.11. Figure 9.11 also shows the FIR filter coefficients and the frequency response of the filter designed using the flat-top window. Figures 9.12 through 9.16 show the noisy ECG signal and the bandpass-filtered ECG signal using various window functions. Table 9.5 shows the SNR of the noisy ECG signal and the FIR-filtered ECG signal using various windows. The first row in Table 9.5 represents the SNR of the noisy

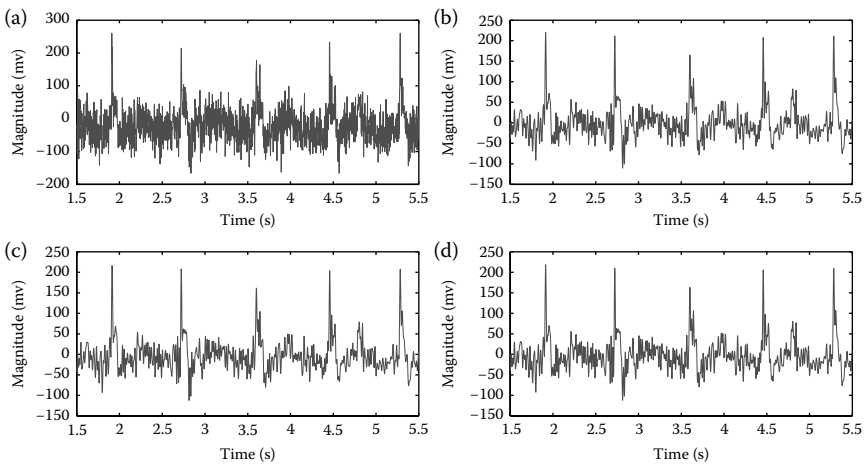


**FIGURE 9.11** Time-domain and frequency-domain representations of flat-top window (dashed line) and bandpass filter designed using flat-top window (solid line).

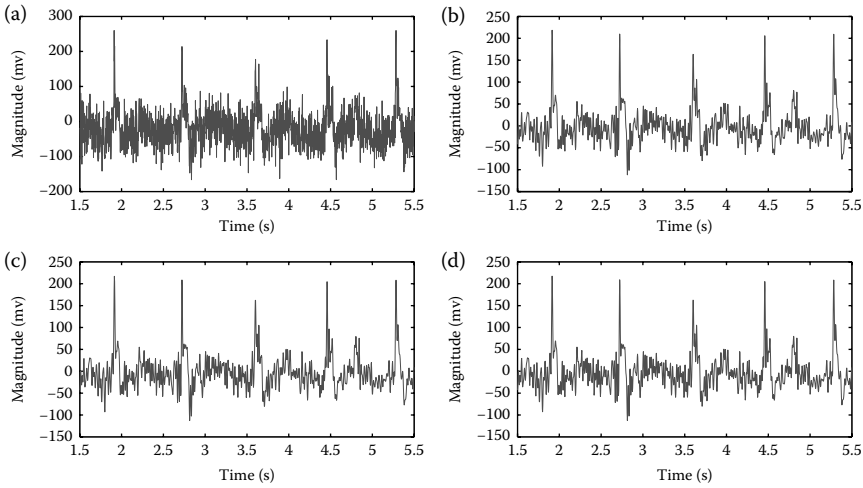
ECG signals subjected to different noise levels. The remaining rows represent the corresponding SNR after bandpass FIR filtering using the various window functions.

To conclude, from the results shown in Table 9.5, we can observe that at high SNRs, the FIR filter designed using flat-top window performs better than other FIR filters. However, at low SNRs, all the FIR filters designed using different windows are almost identical in their performance.

An EEG signal example is taken from the CHB-MIT Scalp EEG Database [7], and this signal is sampled at 256 Hz. The EEG signal was subjected to additive white Gaussian noise at different SNR levels to obtain noisy EEG signals.

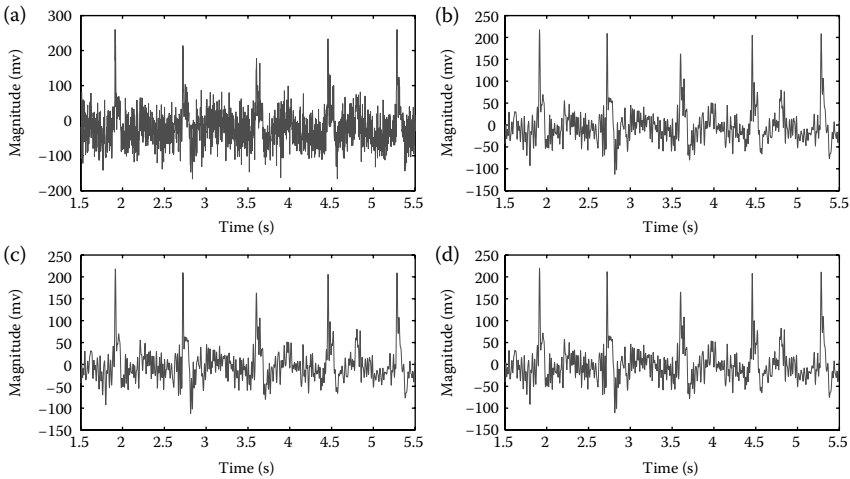


**FIGURE 9.12** (a) Noisy ECG signal and bandpass-filtered ECG signals using (b) rectangular, (c) triangular, and (d) Hann windows.

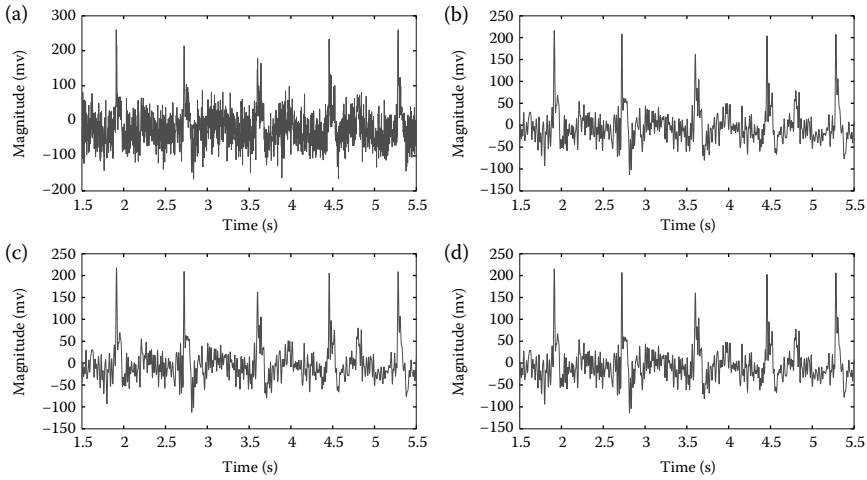


**FIGURE 9.13** (a) Noisy ECG signal and bandpass-filtered ECG signals using (b) Hamming, (c) Parzen, and (d) Tukey windows.

To denoise the EEG signal, FIR lowpass filters with different windows at a sampling frequency of 256 Hz have been designed. The filter cut-off was set at 40 Hz, the transition bandwidth is from 40 to 42.8 Hz with a passband attenuation of 0.01 dB and a stop-band attenuation of 80 dB. The length of all the FIR filters that have been designed is 1001. Table 9.6 shows the SNRs of the



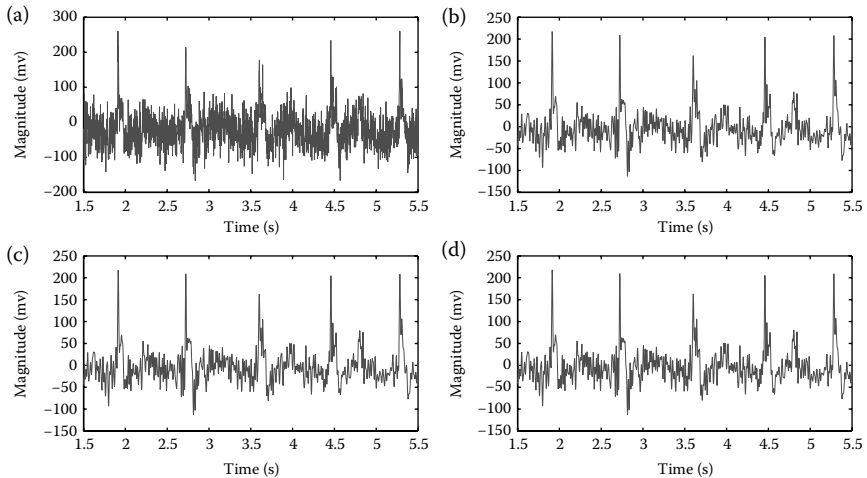
**FIGURE 9.14** (a) Noisy ECG signal and bandpass-filtered ECG signals using (b) Bohman, (c) Gaussian, and (d) Kaiser-Bessel windows.



**FIGURE 9.15** (a) Noisy ECG signal and bandpass-filtered ECG signals using (b) Bartlett, (c) Blackman, and (d) flat-top windows.

noisy EEG signal and the FIR-filtered EEG signal using various windows. The first row in Table 9.6 represents the SNR of the EEG signals that are subjected to different noise levels and subsequent rows represent the corresponding SNR after lowpass FIR filtering using various window functions.

From the results shown in Table 9.6, we can observe that the triangular window-based FIR filter performs slightly better than the other windows



**FIGURE 9.16** (a) Noisy ECG signal and bandpass-filtered ECG signals using (b) Blackman–Harris, (c) four-term Blackman–Harris, and (d) Bartlett–Hann windows.

**TABLE 9.5**

Signal-to-Noise Ratio (SNR) of Bandpass-Filtered ECG Signals Using Various Windows

Signal	SNR at Different Levels of Noise				
Noisy ECG	0.1268	0.0529	-1.0303	-2.526	-4.9923
Rectangular	-0.0254	-0.0487	-0.4027	-0.9913	-2.2384
Triangular	0.7308	0.7041	-0.4027	-0.3883	-1.7843
Hann	0.4027	0.3775	-0.0112	-0.6511	-1.9849
Hamming	0.3681	0.3430	-0.0427	-0.6783	-2.0051
Parzen	0.6524	0.6259	0.2157	-0.4554	-1.8408
Tukey	0.0815	0.0577	-0.3047	-0.9059	-2.1746
Bohman	0.5856	0.5594	0.1551	-0.5075	-1.8791
Gaussian	0.446	0.4206	0.0283	-0.6169	-1.9596
Kaiser-Bessel	-0.0113	-0.0347	-0.3897	-0.9800	-2.2299
Blackman	0.5477	0.5218	0.1207	-0.5372	-1.901
Flat-top	1.095	1.0661	0.6151	-0.1144	-1.5937
Blackman-Harris	0.6806	0.6539	0.2412	-0.4335	-1.8249
Four-term	0.6699	0.6433	0.2316	-0.4418	-1.831
Blackman-Harris					
Bartlett-Hann	0.4812	0.4556	0.0609	-0.5879	-1.9364

considered. However, the improvement in the performance is not significant. In both the cases (bandpass and lowpass) considered here, it is clearly evident that the use of window functions in the FIR filter design improves the performance.

### 9.4.3 Moving Average Filtering of Biomedical Signals

The measured biomedical signals are usually corrupted by random noise. Moving average (MA) filter can be used to denoise the corrupted biomedical signals before analyzing them. In MA filtering, the value at each sample is obtained as a weighted sum of neighboring samples. The number of neighboring samples used depends on the length of the window. The MA filter can be considered as an FIR type, lowpass filter that removes the high-frequency noise present in the signal. These are also called as smoothing filters, which increase the SNR. The length of the window roughly determines the cut-off frequency of this filter. A long smoothing window reduces the variance in the EEG amplitude estimate, but at the cost of increased bias. On the other hand, a short smoothing filter has low bias, but the variance is increased. Generally, weighted windows are used (i.e., nonrectangular). The windows we considered to average the signals are: rectangular, triangular, Hann, Hamming, Parzen, Tukey, Bohman, Gaussian, Kaiser-Bessel, Blackman, flat-top,

**TABLE 9.6**

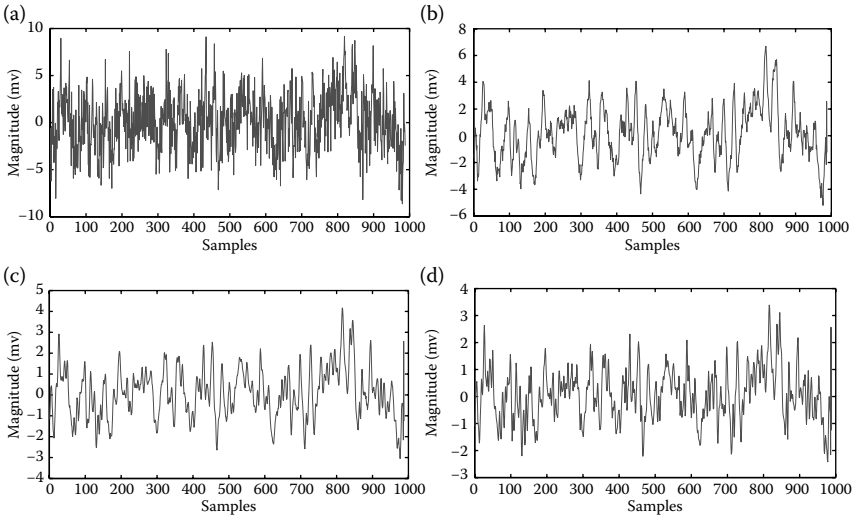
Signal-to-Noise Ratio (SNR) of Lowpass-Filtered EEG Signals Using Various Windows

Signal	SNR at Different Levels of Noise				
Noisy EEG	5.0524	2.5171	0.0024	-2.0545	-4.9546
Rectangular	9.0622	6.9292	4.6376	2.7053	-0.0948
Triangular	9.0916	6.9576	4.6651	2.7324	-0.0679
Hann	9.064	6.9309	4.6392	2.7067	-0.0933
Hamming	9.0639	6.9307	4.639	2.7066	-0.0935
Parzen	9.0665	6.933	4.6411	2.7086	-0.0916
Tukey	9.0621	6.9291	4.6376	2.7053	-0.0948
Bohman	9.0658	6.9324	4.6406	2.708	-0.0921
Gaussian	9.0645	6.9313	4.6396	2.7071	-0.093
Kaiser-Bessel	9.0623	6.9292	4.6377	2.7053	-0.0948
Blackman	9.0652	6.9319	4.6401	2.7076	-0.0925
Flat-top	9.071	6.937	4.6447	2.7119	-0.0884
Blackman-Harris	9.0665	6.9331	4.6412	2.7086	-0.0915
Four-term Blackman-Harris	9.0664	6.933	4.6411	2.7085	-0.0916
Bartlett-Hann	9.0707	6.9373	4.6454	2.7129	-0.0872

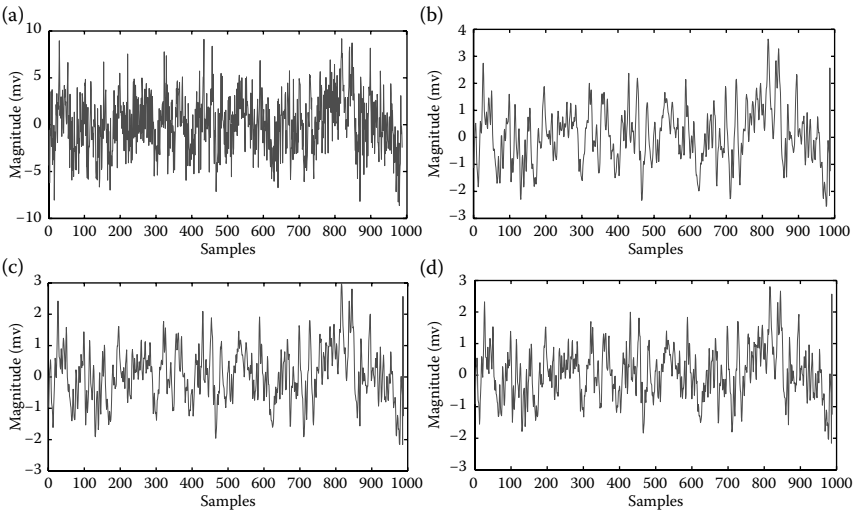
Blackman-Harris, four-term Blackman-Harris, and Bartlett-Hann. A window length of seven is used in this moving average filter. To compare the performance of the moving average filter using various windows, we use EEG signals to which synthetic AWGN has been added. Figures 9.17 through 9.21 show the noisy EEG signal and the moving average filtered EEG signal using various windows. Table 9.7 shows the SNR of the noisy EEG signal and the moving average filtered EEG signal using various windows. From the results shown in Table 9.7, we can observe that at high SNRs, rectangular and Kaiser-Bessel windows perform better than all the other windows. However, at low SNRs, triangular and Hamming windows perform better than other windows.

Another performance evaluation measure of moving average filter using different windows is carried out using EMG signal from muscles. Table 9.8 shows the SNRs of the noisy EMG signal and the moving average filtered EMG signal using various windows. The first row shows the SNRs of the noisy EMG signals (for various noise levels) and the remaining rows represent the corresponding SNRs after applying various windows for moving average filtering. The results shown in Table 9.8 indicate that at high SNRs, the rectangular and Kaiser-Bessel windows perform better than other windows. However, at low SNRs, the triangular and Hamming windows perform better than other windows. In both the cases (EEG and EMG) considered here, at high SNRs, the performance of the rectangular and Kaiser-Bessel windows does not differ significantly when compared to the triangular window. Therefore, using windows in moving average filter will improve its performance.

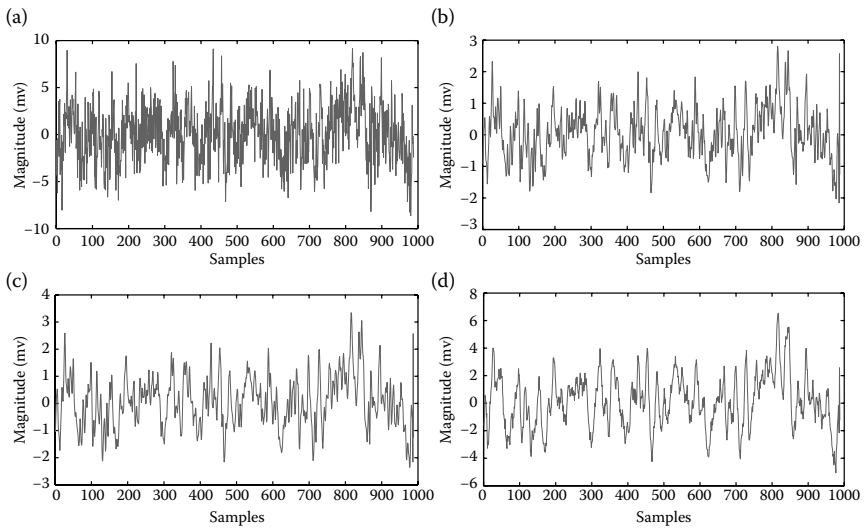




**FIGURE 9.17** (a) Noisy EEG signal and moving average-filtered EEG signals using (b) rectangular, (c) triangular, and (d) Hann windows.

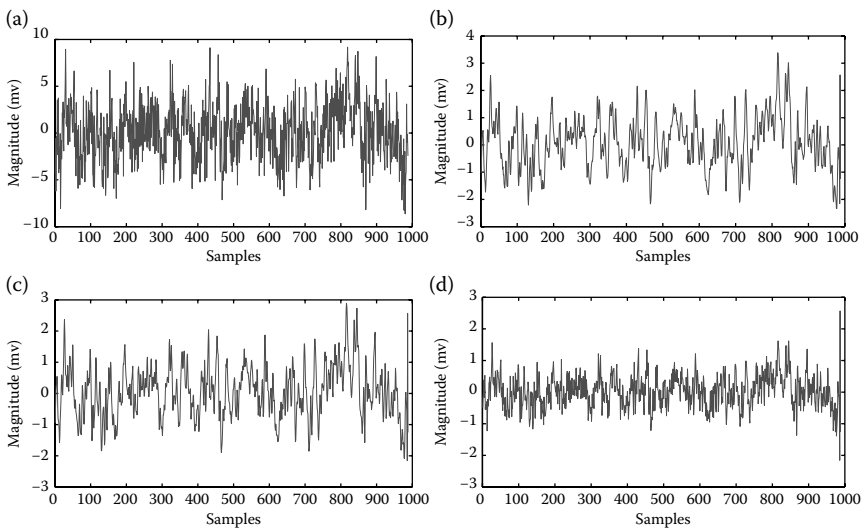


**FIGURE 9.18** (a) Noisy EEG signal and moving average filtered EEG signals using (b) Hamming, (c) Parzen, and (d) Tukey windows.



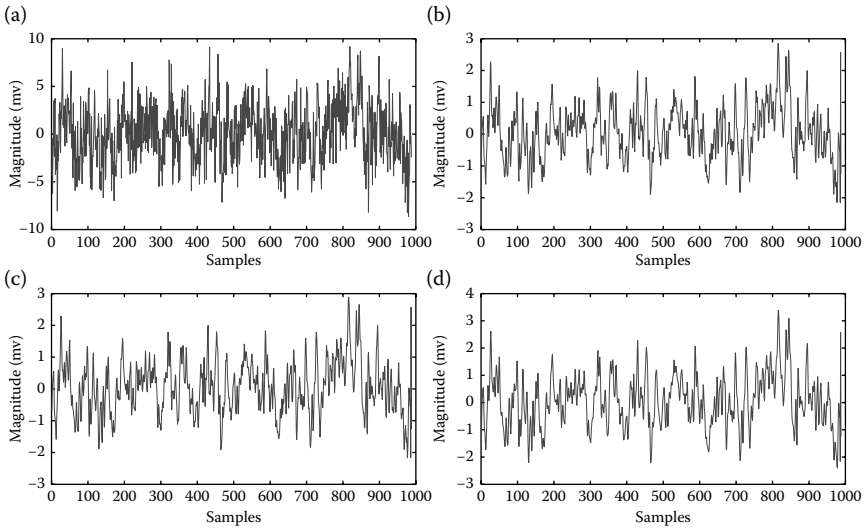
**FIGURE 9.19**

(a) Noisy EEG signal and moving average filtered EEG signals using (b) Bohman, (c) Gaussian, and (d) Kaiser–Bessel windows.



**FIGURE 9.20**

(a) Noisy EEG signal and moving average filtered EEG signals using (b) Bartlett, (c) Blackman, and (d) flat-top windows.



**FIGURE 9.21** (a) Noisy EEG signal and moving average filtered EEG signals using (b) Blackman–Harris, (c) four-term Blackman–Harris, and (d) Bartlett–Hann windows.

**TABLE 9.7**

Signal-to-Noise Ratio (SNR) of Moving Averaged Filtered EEG Signal Using Various Windows

Noisy EEG	1.2042	−0.0597	−2.8092	−5.257	−7.2061
Rectangular	2.3077	2.0621	1.2361	0.2011	−0.9327
Triangular	2.0752	1.9732	1.6358	1.1407	0.5328
Hann	1.5429	1.4672	1.2178	0.8347	0.3603
Hamming	1.7205	1.6372	1.363	0.9474	0.4339
Parzen	1.4005	1.3373	1.1312	0.8081	0.4027
Tukey	1.8919	1.7518	1.2804	0.6153	−0.1588
Bohman	1.3094	1.2512	1.0618	0.7625	0.3854
Gaussian	1.5851	1.5122	1.2734	0.9054	0.4466
Kaiser–Bessel	2.3446	2.1087	1.3141	0.3102	−0.7967
Bartlett	1.5808	1.5088	1.2718	0.9066	0.4501
Blackman	1.3479	1.2874	1.0902	0.7799	0.3897
Flat top	0.7223	0.689	0.5813	0.4045	0.1755
Blackman–Harris	1.1648	1.1063	0.9197	0.6246	0.2504
Four-term Blackman–Harris	1.1774	1.1181	0.9287	0.6298	0.2511
Bartlett–Hann	1.5525	1.4779	1.232	0.8537	0.3844

**TABLE 9.8**

Signal-to-Noise Ratio (SNR) of Moving Averaged Filtered EMG Signal Using Various Windows

Noisy EMG	1.4044	0.0538	-1.0107	-3.0366	-4.9888
Rectangular	3.3907	3.0354	2.7034	1.8937	0.9799
Triangular	2.9111	2.7791	2.6515	2.3051	1.9068
Hann	2.18	2.0883	1.997	1.7494	1.4602
Hamming	2.4238	2.3210	2.2194	1.9436	1.6233
Parzen	1.9474	1.8734	1.7989	1.5978	1.3626
Tukey	2.8459	2.6562	2.4726	1.9905	1.4335
Bohman	1.8121	1.745	1.6769	1.4942	1.28
Gaussian	2.2155	2.1276	2.0403	1.8029	1.527
Kaiser-Bessel	3.4395	3.0978	2.7781	1.9933	1.1065
Bartlett	2.2101	2.1226	2.0363	1.8006	1.5274
Blackman	1.8704	1.8002	1.7292	1.538	1.3141
Flat top	0.9583	0.9228	0.885	0.7916	0.6778
Blackman-Harris	1.6574	1.5912	1.5249	1.3451	1.1346
Four-term Blackman-Harris	1.6785	1.6112	1.5438	1.3611	1.1471
Bartlett-Hann	2.1878	2.0973	2.0074	1.7631	1.4783

#### 9.4.4 QRS Detection in ECG Based on STFT

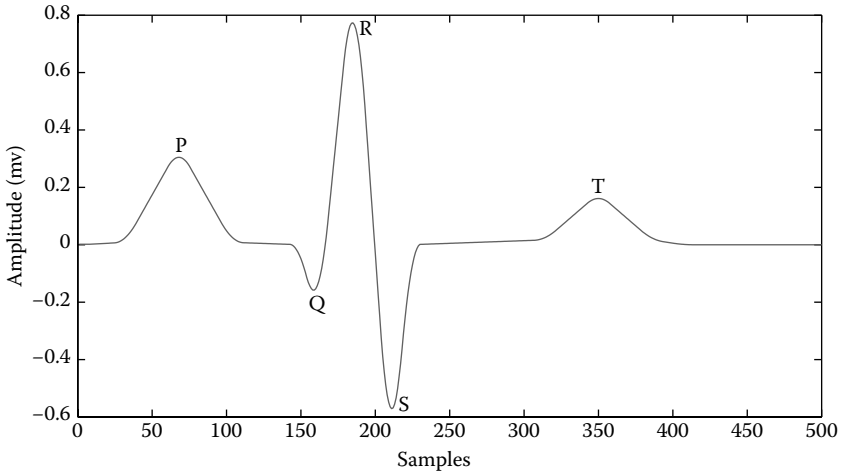
The STFT finds several applications in biomedical signal processing. In the case of an ECG signal, it is used to detect the location of the QRS complex. The QRS complex denotes the deflections on an ECG signal; it is a combination of Q wave, R wave and S wave (see Figure 9.22). The morphology of an ECG signal mainly consists of a P wave, a QRS complex, and a T wave for each cardiovascular cycle. Figure 9.22 shows a synthetic ECG signal for one cardiac cycle. The QRS complex locations are useful in determining the heart rate variability. Here, the STFT is employed to remove the unwanted information such as the P wave, the T wave, and the noise, and the STFT temporal information at 45 Hz is used to detect the QRS complex [9].

The STFT of an input signal  $x(t)$  is defined as

$$T(f, \tau) = \int_{-\infty}^{\infty} [x(t)f(t - \tau)]e^{-j2\pi ft} dt,$$

where  $f(t - \tau)$  denotes the shifted window function and  $x(t)$  is the input signal. The temporal information located at 45 Hz is obtained as described below (see also Ref. [9]):

$$E(\tau) = \log(|T(45, \tau)|^2).$$

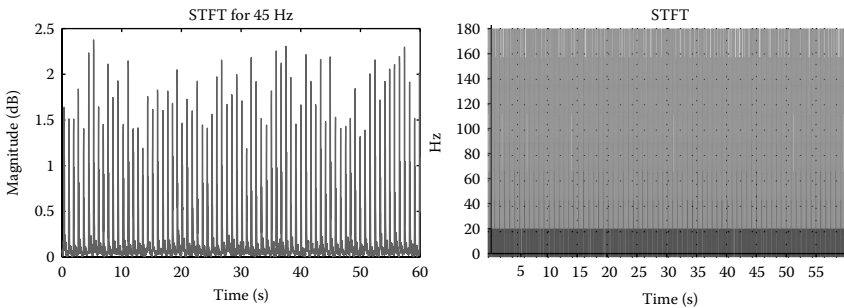


**FIGURE 9.22**  
Synthetic ECG signal.

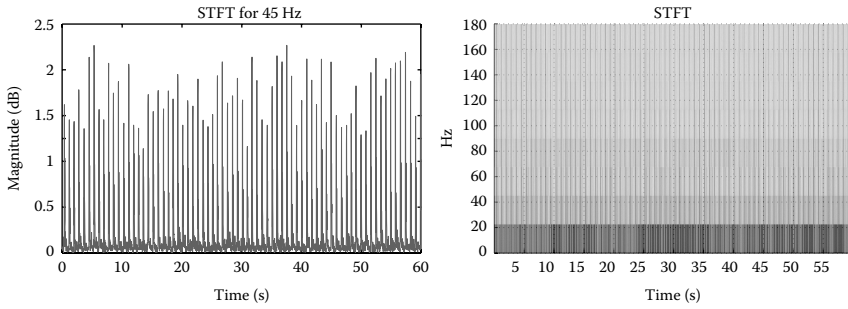
To detect the QRS complex, adaptive thresholding is done on the temporal information. Finally, the locations of QRS complexes have been evaluated from the position of local maxima.

Figures 9.23 through 9.28 show the STFT temporal information at 45 Hz and the complete STFT using various windows. From these figures, we can observe that the STFT temporal information at 45 Hz is different for these windows. Therefore, the performance of the QRS detection algorithm will also vary with the type of window used.

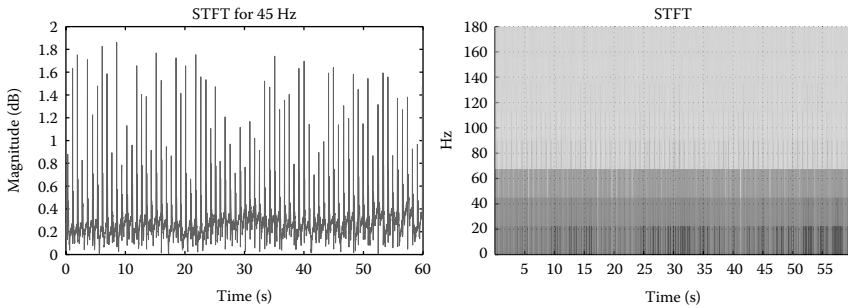
Figures 9.29 through 9.33 show a portion of the ECG signal and the corresponding locations of the QRS complex that have been detected using various window functions that are used to calculate STFT. By observing these figures, we conclude that the use of Hamming, Hann, and Bartlett window functions for computing the STFT will result in a much accurate detection of



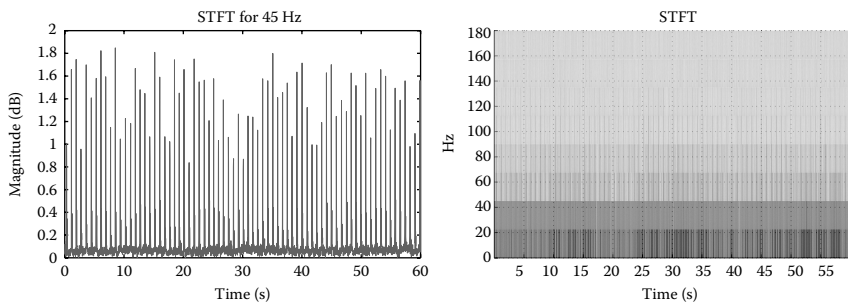
**FIGURE 9.23**  
STFT and its temporal information at 45 Hz using rectangular window.



**FIGURE 9.24** STFT and its temporal information at 45 Hz using Kaiser–Bessel window.

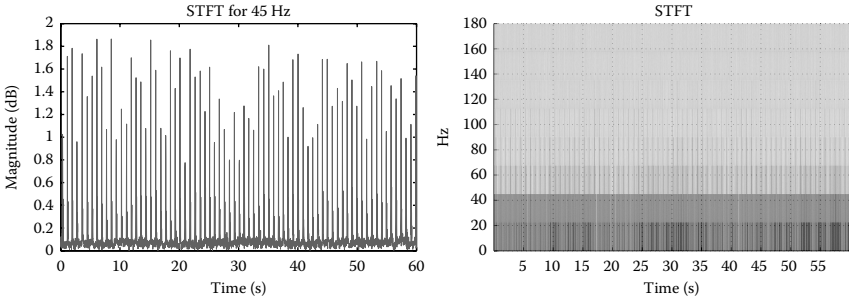


**FIGURE 9.25** STFT and its temporal information at 45 Hz using Blackman window.

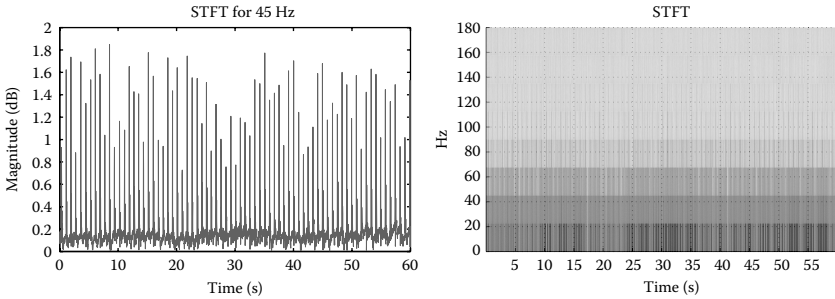


**FIGURE 9.26** STFT and its temporal information at 45 Hz using Hamming window.

the QRS complex than the other windows. When no window is used (i.e., in effect applying a rectangular window) in the STFT, the QRS locations are not detected correctly. This can be clearly seen in Figure 9.29(b). Therefore, the use of window functions plays a significant role in biomedical signal processing.



**FIGURE 9.27** STFT and its temporal information at 45 Hz using Hann window.



**FIGURE 9.28** STFT and STFT magnitude at 45 Hz using Gaussian window.

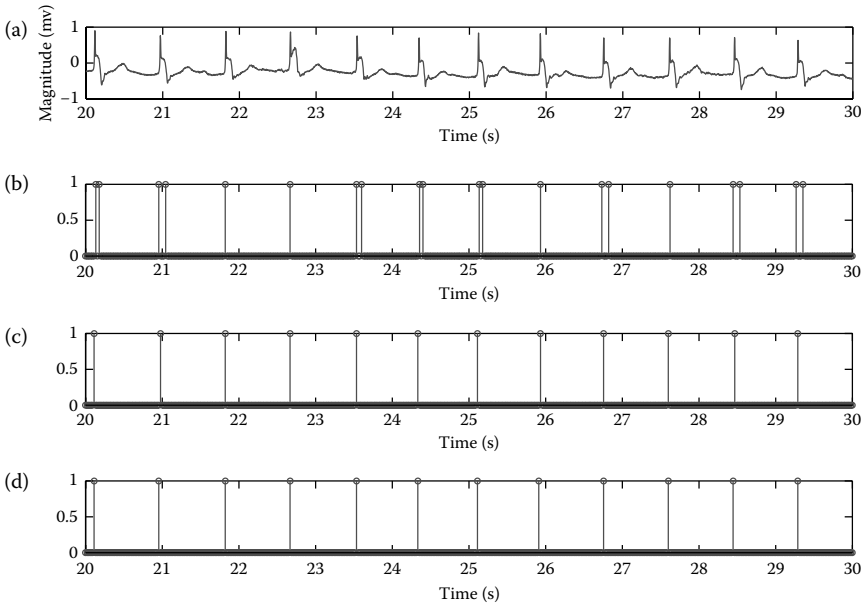
---

## 9.5 Audio Denoising Using the Time–Frequency Plane

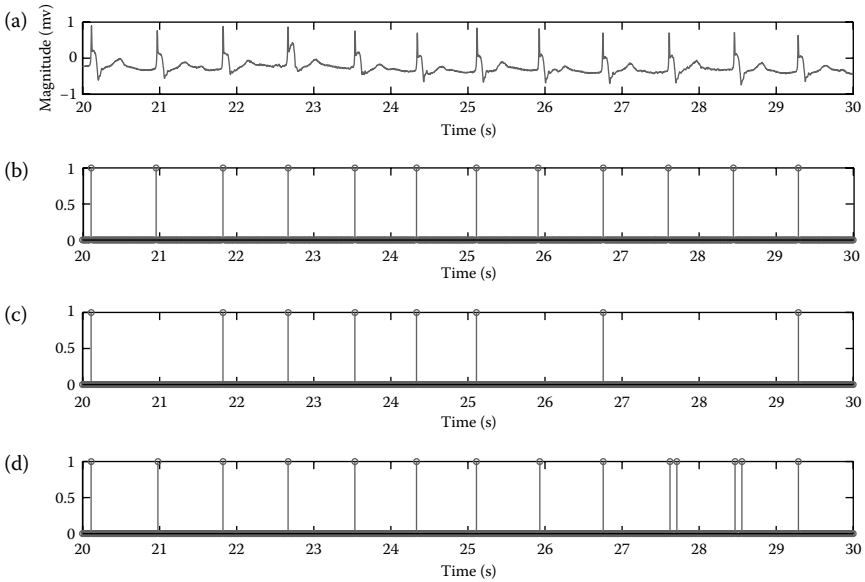
Music and other audio signals such as speech are often susceptible to background noise from audio equipments and surrounding environment. Several methods such as power subtraction and processing in wavelet and Fourier domains have been tried to address this problem. These methods, in addition to removing noise, create isolated time–frequency structures, which are known as “musical noise.”

### 9.5.1 Time–Frequency Plane

Signals are generally analyzed either in the time-domain (the zero crossing rate) or in the frequency-domain (Fourier analysis, subband energy, etc.). Joint time and frequency analysis gives critical information of signals whose frequency varies with time and is generally used in the analysis of human speech, multicomponent signals, and source separation among others. A time–frequency (TF) plane,  $X[l, k]$  is a two-dimensional plane obtained by

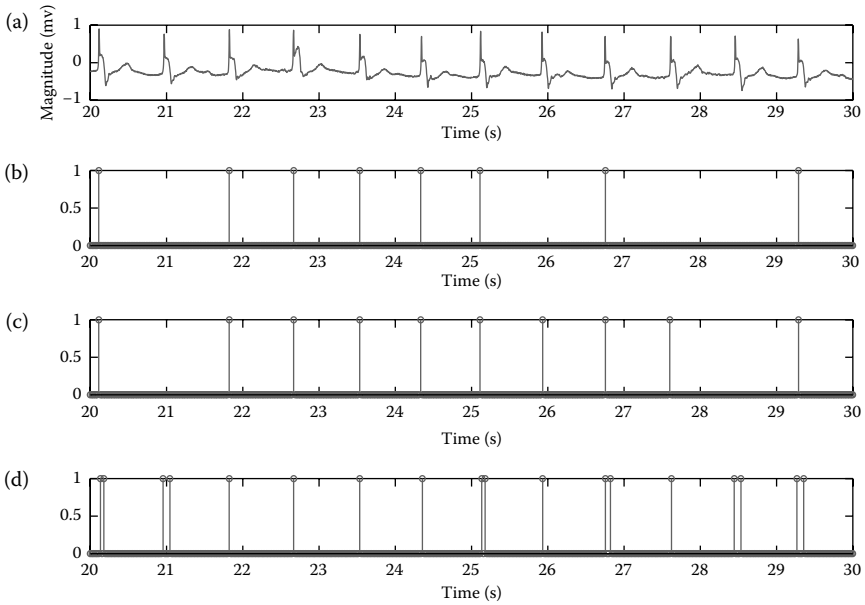


**FIGURE 9.29** (a) ECG signal and detected QRS locations using (b) rectangular, (c) triangular, and (d) Hann windows.

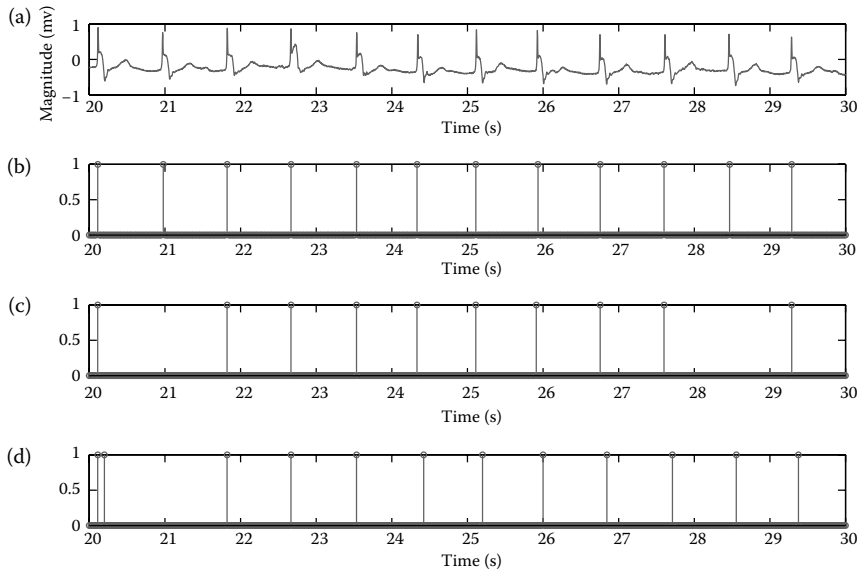


**FIGURE 9.30** (a) ECG signal and detected QRS locations using (b) Hamming, (c) Parzen, and (d) Tukey windows.

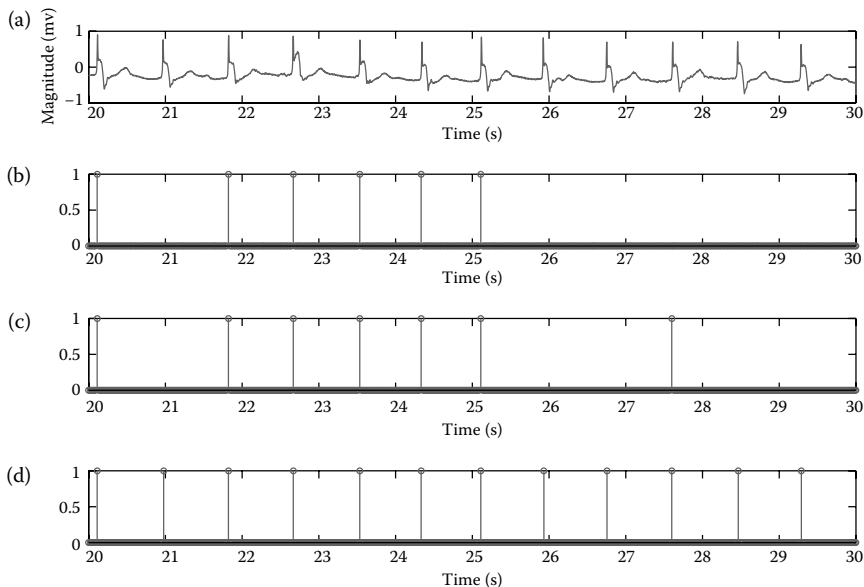




**FIGURE 9.31** (a) ECG signal and detected QRS locations using (b) Bohman, (c) Gaussian, and (d) Kaiser-Bessel windows.



**FIGURE 9.32** (a) ECG signal and detected QRS locations using (b) Bartlett, (c) Blackman, and (d) flat-top windows.

**FIGURE 9.33**

(a) ECG signal and detected QRS locations using (b) Blackman-Harris, (c) four-term Blackman-Harris, and (d) Bartlett-Hanning windows.

transforming a signal  $x[n]$  using a set of time-frequency atoms  $g_{l,k}$ , where  $l$  and  $k$ , respectively, are the time and frequency scale parameters given as follows:

$$X[l, k] = \langle x, g_{l,k} \rangle = \sum_{n=0}^{N-1} x[n] g_{l,k}^*[n].$$

The commonly used TF atoms are the Fourier, Gabor, or Gammatone atoms. The T-F plane obtained using the Fourier atom is called a spectrogram and the plane obtained using Gammatone atoms is called the cochleagram [10]. To reconstruct the signal back to  $x[n]$  from the TF plane, the TF atoms must be on a tight frame, which implies that there exists some  $A > 0$  such that

$$\|x\|^2 = \frac{1}{A} \sum_{l,k} |\langle x, g_{l,k} \rangle|^2.$$

Given one such  $A$ , the reconstruction of the signal is

$$x[n] = \frac{1}{A} \sum_{l,k} X[l, k] g_{l,k}[n].$$

If the atoms are orthogonal, then  $A = 1$  and the reconstruction is unique, else there are many possible reconstructions for  $x[n]$  and are given by

$$x[n] = \frac{1}{A} \sum_{l,k} C[l,k] g_{l,k}[n],$$

with the constraint that

$$\|x\|^2 \leq \frac{1}{A} \sum_{l,k} |C[l,k]|^2, \quad (9.37)$$

where  $C[l,k] = \langle x, g_{l,k} \rangle$ .

### 9.5.2 Audio Denoising Using Time–Frequency Plane

Ephraim and Mallah [12] suggested the removal of musical noise using non-diagonal time–frequency estimators. An improvement in the performance of the algorithm was observed using a block thresholding technique. This section details the diagonal estimator method leading to the power subtraction technique for noise reduction and concludes with the more recent block thresholding technique as detailed in Ref. [11].

Consider a noisy signal  $y[n]$  given by

$$y[n] = x[n] + e[n],$$

where  $x[n]$  is the true signal and  $e[n]$  is the noise component. The time–frequency transform decomposition of  $y[n]$  over the set of time–frequency atoms  $g_{l,k}[n]$  results in coefficients written as

$$Y[l,k] = \langle y, g_{l,k} \rangle = \sum_{n=0}^{N-1} y[n] g_{l,k}^*[n].$$

Assuming that  $g_{l,k}[n]$  defines a tight frame,  $y[n]$  can be reconstructed by

$$y[n] = \frac{1}{A} \sum_{l,k} Y[l,k] g_{l,k}[n].$$

The denoising algorithm reconstructs the signal by attenuating the individual TF units using a factor  $a[l,k]$ . The denoised signal is then given by

$$\hat{x}[n] = \frac{1}{A} \sum_{l,k} a[l,k] Y[l,k] g_{l,k}[n],$$

where  $\hat{x}[n]$  is the estimated denoised signal. We use  $\hat{X}[l, k] = a[l, k]Y[l, k]$  to denote its corresponding time–frequency coefficients.

Following from Equation 9.37, the quadratic estimation risk,  $r$ , associated while reconstructing the signal is given by

$$r = E\{||x - \hat{x}||^2\} \leq \frac{1}{A} \sum_{l,k} E\{|X[l, k] - \hat{X}[l, k]|^2\}, \quad (9.38)$$

where  $E\{\theta\}$  is the estimate of  $\theta$ . The upper bound on  $r$ , called the oracle risk,  $r_o$ , can be found by differentiating Equation 9.38 with respect to  $a$ . The risk  $r_o$  can be evaluated to occur at

$$a[l, k] = 1 - \frac{1}{\hat{\xi}[l, k] + 1}, \quad (9.39)$$

where  $\hat{\xi}[l, k] = X^2[l, k]/\sigma^2[l, k]$  is the *a priori* SNR, which unfortunately is an unknown quantity.

To overcome this, we use *diagonal estimators* of the SNR  $\xi[l, k]$  that are computed from the *a posteriori* SNR defined by  $\gamma[l, k] = |Y[l, k]|^2/\sigma^2[l, k]$ . The empirical Wiener estimator is then defined as

$$a[l, k] = \left(1 - \frac{1}{\hat{\xi}[l, k] + 1}\right)_+, \quad (9.40)$$

where  $(z)_+ = \max(z, 0)$  and  $\hat{\xi}[l, k] = \gamma[l, k] - 1$ .

A more generalized form of the Wiener estimator is

$$a[l, k] = \left(1 - \lambda \left[\frac{1}{\hat{\xi}[l, k] + 1}\right]^{\beta_1}\right)^{\beta_2}_+ \quad (9.41)$$

where  $\beta_1, \beta_2 \geq 0$ , and  $\lambda \geq 1$  is the over-subtraction factor. This is generally observed as the power subtraction method for noise reduction.

### 9.5.3 Block Thresholding

To minimize the musical noise, we divide the time–frequency plane into  $I$  blocks of  $B_i$ . The resulting estimator depends on the TF units in a neighborhood. The signal estimated  $\hat{x}$  from  $y$  using the block thresholding estimator is calculated by

$$\hat{x}[n] = \sum_{i=1}^I \sum_{(l,k) \in B_i} a_i Y[l, k] g_{l,k}[n].$$

To find  $a_i$ , we again minimize the risk  $r$

$$r = E\{|x - \hat{x}|^2\} \leq \frac{1}{A} \sum_{i=1}^I \sum_{(l,k) \in B_i} E\{|a_i Y[l,k] - X[l,k]|^2\}. \quad (9.42)$$

Differentiation of Equation 9.42 with respect to  $a_i$  and equating it to zero, we find  $r$  to have a maximum value at

$$a_i = 1 - \frac{1}{\xi_i + 1}, \quad (9.43)$$

where  $\xi_i = \bar{F}_i^2 / \bar{\sigma}_i^2$  is the average *a priori* SNR in  $B_i$ . Note that  $\bar{F}_i^2$  is obtained as

$$\bar{F}_i^2 = \frac{1}{B_i^\#} \sum_{(l,k) \in B_i} |F[l,k]|^2,$$

where  $B_i^\#$  is the number of TF units in the  $i$ th block.

We can estimate  $\xi_i$  using the *a posteriori* SNR by using the relation

$$\hat{\xi}_i = \frac{\bar{Y}_i^2}{\bar{\sigma}_i^2} - 1,$$

where

$$\bar{\sigma}_i^2 = \frac{1}{B_i^\#} \sum_{(l,k) \in B_i} |\sigma[l,k]|^2$$

and

$$\bar{Y}_i^2 = \frac{1}{B_i^\#} \sum_{(l,k) \in B_i} |Y[l,k]|^2.$$

If the noise is stationary, the noise variance does not depend upon time:  $\sigma^2[l,k] = \sigma^2[k]$ . Generalizing Equation 9.43,  $a_i$  can be written as

$$a_i = \left( 1 - \frac{\lambda}{\hat{\xi}_i + 1} \right)_+. \quad (9.44)$$

*Adaptive Block Thresholding:* Better denoising performance can be achieved by adaptively choosing the block size of block  $B_i$  by minimizing the risk defined in Equation 9.42. This approach requires the estimation of the attenuating factor  $a_i$  using the Stein unbiased risk estimate (SURE) [13].

### 9.5.4 Effect of Windows

In this section, we use Fourier atoms to obtain the TF plane. The windowed Fourier atoms can be written as  $g_{l,k}[n] = w[n - lu] \exp(i2\pi kn/K)$ , where  $w[n]$

is the window. The resulting TF plane, called the spectrogram, is given by

$$X[l, k] = \sum_{n=0}^{N-1} x[n]w[n - lu] \exp(i2\pi kn/K). \quad (9.45)$$

The signal  $x[n]$  chosen for this experiment is a musical signal having an SNR of  $-5$  dB and sampled at 11 KHz. The objective of this experiment is to improve the SNR of the signal using a block thresholding technique.

Since windowing in the time-domain results in convolution in the frequency-domain, disturbing the spectral characteristics of the signal, the choice of the window  $w[n]$  becomes very important. Neglecting the sinusoidal component in Equation 9.45, the windowed signal,  $x_w[n] = x[n]w[n]$ , in the frequency-domain is

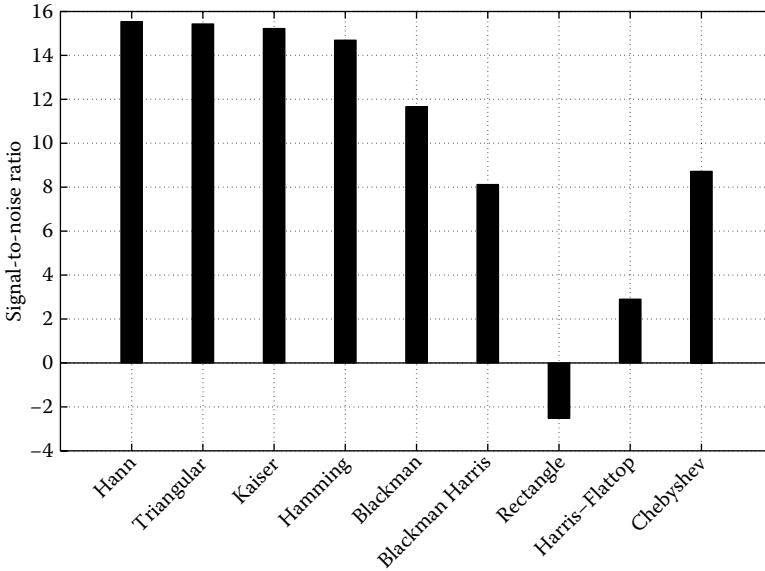
$$X_w(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{-j\omega'}) X(e^{-j(\omega-\omega')}) d\omega'.$$

For a proper representation of the signal,  $W(e^{j\omega})$  should be highly concentrated around  $\omega = 0$ . A rectangular window, for example, has a main-lobe width that is inversely proportional to the window length, with a substantial energy spread in the side lobe. Windows such as Hamming, Hann and triangular have a very high energy concentration in the main lobe when compared to the side lobes. The localization of energy around  $\omega = 0$  for Hann and other similar windows ensures minimum spectral leakage across the time–frequency units. This property of the Hann window gives an improved denoising performance when compared to other windows.

A summary of the results obtained by using different windows is displayed in the form of a bar chart (shown in Figure 9.34). As expected, the performance of the rectangular window is poor, due to spectral leakages. It results in a very low SNR of only  $-2.51$  dB, when compared to  $15.53$  dB using the Hann window. The best SNR result was obtained using the Hann window, with comparable results using Hamming, triangular, and Kaiser windows.

## 9.6 Effect of Windows on Linear Prediction of Speech

One of the breakthroughs of automatic speech recognition research is in speech coding. Human speech transmission through mobile networks, in its current form, owes its existence to speech-coding techniques, without which the cost of a phone call through a mobile network would be prohibitively high. To reduce the amount of data transmitted through a network, the raw human speech is encoded using algorithms such as linear predictive coding.



**FIGURE 9.34**  
Signal-to-noise ratio comparison of various windows.

### 9.6.1 Linear Prediction Coder

The most commonly used speech-coding algorithm is the lossy linear predictive coding (LPC). This uses an  $N$ th order autoregressive (AR) or all-pole model to represent speech frame using  $N$  coefficients. The LPC algorithm models the spectral envelope of the signal and gives lesser importance to details as shown in Figure 9.35. The reconstructed signal from LPC coefficients has a different temporal structure when compared to the original signal. However, it still retains the vital aspects of the speech data, such as pitch and spectral peaks, which are critical to understand the human speech.

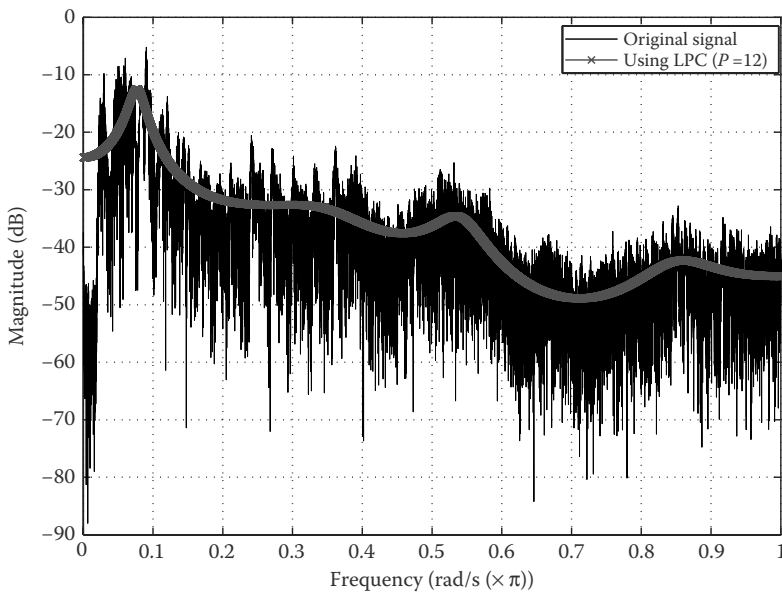
To compute the LPC coefficients [14], the digital signal,  $x[n]$ , is segmented using a window,  $w[n]$ . Each windowed frame is given by

$$y[n] = x[n]w[n].$$

The  $N$ th order linear predictor of  $y[n]$  is defined using  $N$  previous components as

$$\tilde{y}[n] = \sum_{k=1}^N \alpha_k y[n - k].$$

where  $\alpha_k$  is the weight associated with the  $k$ th previous sample. When viewed as a system, the linear predictor has a system function defined as

**FIGURE 9.35**

Magnitude response comparison of the original signal and its estimate using LPC coefficients.

$$P(z) = \frac{\tilde{Y}(z)}{Y(z)} = \sum_{k=1}^N \alpha_k z^{-k}.$$

The error obtained using the  $N$ th-order all-pole model is computed as

$$e[n] = y[n] - \tilde{y}[n] = y[n] - \sum_{k=1}^N \alpha_k y[n - k]. \quad (9.46)$$

Representing Equation 9.46 as a system, the error  $e[n]$  is obtained by passing the signal  $y[n]$  through a system  $A(z)$  defined by

$$A(z) = \frac{E(z)}{Y(z)} = \frac{Y(z) - \tilde{Y}(z)}{Y(z)} = 1 - P(z) = 1 - \sum_{k=1}^N \alpha_k z^{-k}. \quad (9.47)$$

We note that  $A(z)$  is called the LPC polynomial. The all-pole system,  $H(z)$ , is then denoted as

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^N \alpha_k z^{-k}}. \quad (9.48)$$



Here,  $H(z)$  represents the LPC model and the  $N$  values of  $\alpha_k$  are known as the LPC coefficients.

### *Advantages of using LPC*

A speech signal is generally sampled at 8000 samples per second. Each frame is represented by 10 all-pole spectrum parameters. Furthermore, we have three parameters: the pitch frequency, voicing versus nonvoicing decision, and gain [17], making a total of 13 parameters. Assuming 100 frames/s and  $N = 10$  gives a total of 1300 parameters, which have to be transmitted every second, when compared to 8000 samples of uncompressed speech. From this, the reduction in the data and bandwidth requirement associated with LPC is apparent.

### *Limitations of LPC*

Though LPC gives a concise representation of a speech signal, it is highly sensitive to noise. Small errors in LPC coefficients, which often occur while transmitting data through a noisy wireless channel, destroy speech properties and render the reconstructed speech signal illegible.

## 9.6.2 Line Spectral Frequencies

We shall now introduce line spectral frequencies (LSF). To make speech representation coefficients more robust to noise, we split each pole of the LPC transfer function into two separate poles as shown in Figure 9.36 for a segment of speech signal.

The LPC polynomial,  $A(z)$ , which generates the prediction error sequence, Equation 9.47 on expansion gives

$$A(z) = 1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} - \alpha_3 z^{-3} - \alpha_4 z^{-4} - \dots - \alpha_N z^{-N}. \quad (9.49)$$

The reciprocal polynomial  $\tilde{A}(z)$  is then formed by

$$\tilde{A}(z) = z^{-(N+1)} A(z^{-1}) = -\alpha_N z^{-1} - \dots - \alpha_2 z^{-N+1} - \alpha_1 z^{-N} + z^{-(N+1)}, \quad (9.50)$$

where the roots of  $\tilde{A}(z)$  are the inverse of the roots of  $A(z)$ .

The line spectral pairs  $P(z)$  and  $Q(z)$  corresponding to vocal tract models [14] are now defined by

$$\begin{aligned} P(z) &= A(z) + \tilde{A}(z) = A(z) + z^{-(p+1)} A(z^{-1}), \\ Q(z) &= A(z) - \tilde{A}(z) = A(z) - z^{-(p+1)} A(z^{-1}). \end{aligned}$$

Polynomials  $P(z)$  and  $Q(z)$  have their roots on the unit circle. These roots, ordered in ascending order of frequencies, are called the *line spectral frequencies*

(LSF) [16]. Polynomial  $A(z)$  can be reconstructed from  $P(z)$  and  $Q(z)$  by

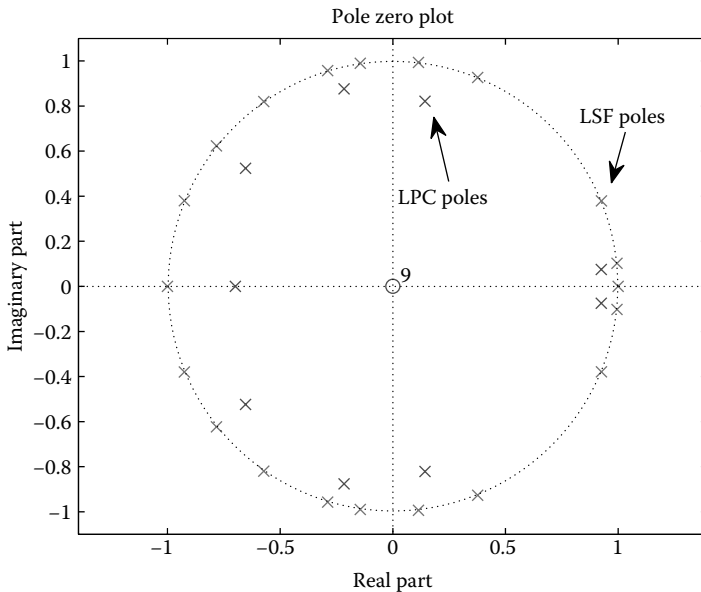
$$A(z) = \frac{P(z) + Q(z)}{2}.$$

The reconstructed  $A(z)$  is stable and is robust to errors.

### 9.6.3 LSF Variation due to Windows

A speech signal is generally broken into a set of frames using windows for further processing. Windows play an important role in the linear analysis of speech signal and several factors need to be taken into account before choosing the right window. Here, we analyze the effect of windows on LSF coefficients.

For speech analysis, an important aspect of a window is its length. A window of 30 ms duration is considered to be optimal in speech-processing applications [14]. This window length is optimum, since it is not so long as to lose the local statistical properties of the signal such as stationarity. It is also not so short that the autocorrelation values can no longer be estimated by averaging lagged values [15]. Since multiplication in the time-domain is convolution in the frequency-domain, the convolution smears the frequency features, depending on the width of the main lobe of the window frequency



**FIGURE 9.36**  
Pole zero plot of LPC ( $N = 9$ ) and LSF transfer functions.

**TABLE 9.9**  
Pedestal Heights of Different Windows

Window	Pedestal (%)
Rectangular	100
Hann	0
Modified Hann	0
Hamming	8
Modified Hamming	8
Ultraspherical ( $\alpha = 0.8$ )	1.35
Kaiser ( $\alpha = \pi\sqrt{3}$ )	2.47
Blackman	0
Blackman–Harris	0

response. Selecting the appropriate window, therefore, depends on choosing a window with the right main-lobe width. Another important aspect of windows is the pedestal height, which is the difference between the value of the first and the last sample of the window function. Pedestal height has an effect on the attenuation of the side lobes in the frequency response and significantly impacts the estimation of linear parameters. The pedestal heights for different windows are provided in Table 9.9.

LSF comparison using a Hamming window (with a pedestal height of 8%), a Hann window (zero pedestal height), and a rectangular window are shown in Figures 9.37(a) through (c). These coefficients were obtained for a speech segment “*This was easy for me*” spoken by a male that was sampled at 8000 Hz. Windows were advanced one sample at a time.

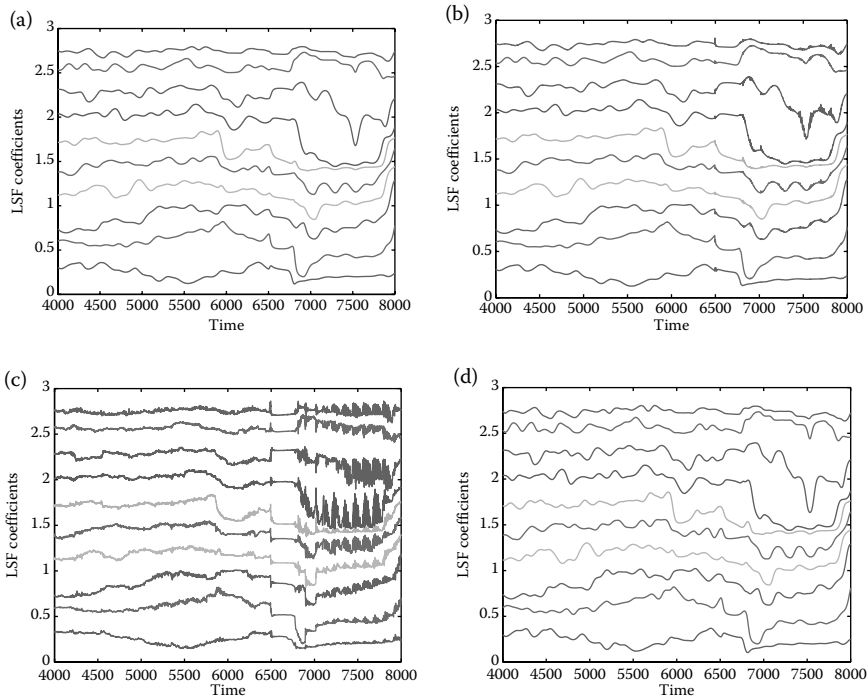
A modified Hann window [15] is given by

$$w[n] = \begin{cases} \frac{1 + \alpha}{2} - \frac{1 - \alpha}{2} \cos\left(\frac{\pi(2n + 1)}{N}\right), & 0 \leq n \leq N - 1 \\ 0, & \text{elsewhere} \end{cases} \quad (9.51)$$

We observe smooth variations as the Hann window (with  $\alpha = 0$ ) advances (Figure 9.37(a)), mainly due to the zero pedestal property of the Hann window.

A slightly more perturbed LSF variation can be observed while using the Hamming window. A Hamming window [15] can be obtained from Equation 9.51 using  $\alpha = 0.08$ . The glitches in the LSF variation (Figure 9.37(b)) occur due the small pedestal height of the Hamming window.

A rectangular window has a high pedestal that gives heavy oscillations in LSF estimation as shown in Figure 9.37(c). LSF variation as a function of time for the Blackman, flat-top, Kaiser, triangular, and ultraspherical windows are shown in Figures 9.37(d) through 9.38(d).

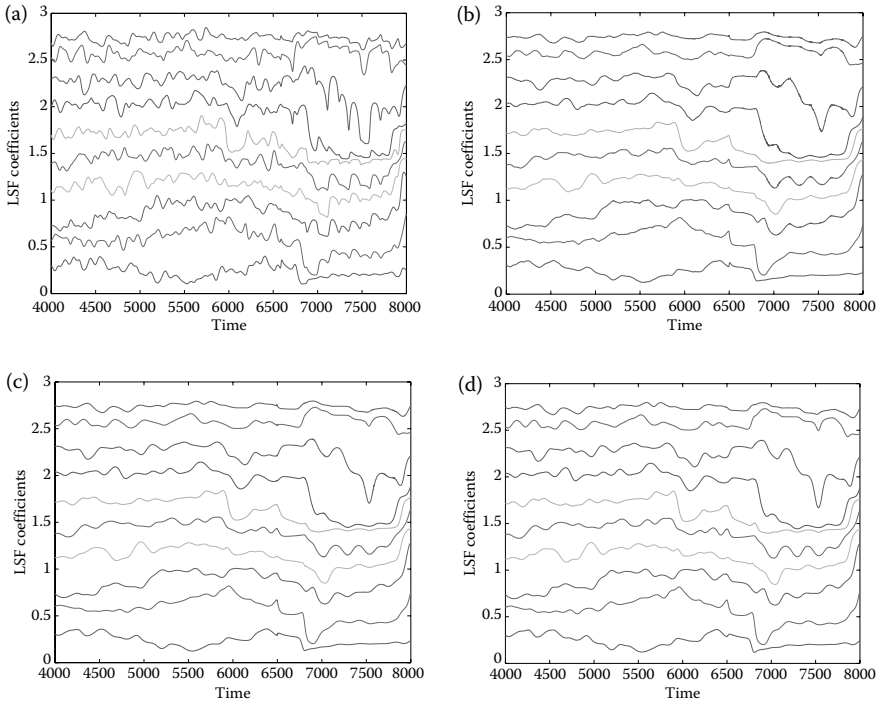
**FIGURE 9.37**

LSF coefficients using different windows. (a) Hann window. (b) Hamming window. (c) Rectangular window. (d) Blackman window.

---

## 9.7 Application of Windows in Image Processing

Since its origin in the 1950s, microwave imaging systems have been widely exploited in applications such as target identification, remote sensing, nondestructive testing, and the military. Two systems of this kind are the synthetic aperture radar (SAR) and the inverse synthetic aperture radar (ISAR). SAR is a high-resolution remote sensing technique carried on a moving platform such as a satellite or an airplane and it is intended for imaging remote terrains. SAR is applied mainly in surveillance, archaeology, mining, agriculture, ecology, and geophysics. Inverse SAR is quite similar, but it is a stationary radar system that captures images of moving targets like aircraft, ship, or tank. ISAR is generally used for the identification and classification of targets, especially in airborne maritime surveillance. In a common scenario, aerial targets are imaged via a ground-based radar, whereas ground- or sea-based targets are imaged with the help of an airborne radar. In this study, we focus on aircraft ISAR imaging via ground-based radar.



**FIGURE 9.38**

LSF coefficients using different windows. (a) Flat-top window. (b) Triangular window. (c) Kaiser window ( $\alpha = \pi\sqrt{3}$ ). (d) Ultraspherical window ( $\alpha = 0.8$ ).

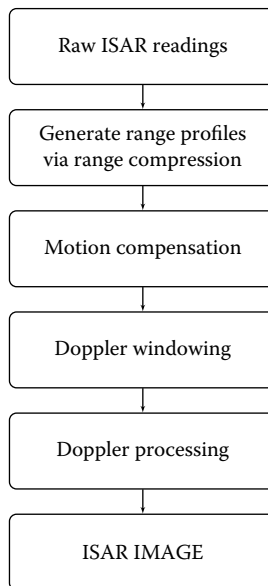
### 9.7.1 Windows for ISAR Images

The basic principle of ISAR imaging is to coherently collect the scattered large bandwidth echoes produced due to the rotation of the object, which brings about a change in the viewing angle to the radar. By processing the echo signals collected, information of the individual point scatterers on the target object and their relative range can be derived. Therefore, the radar image can be assumed to consist of many energy points called scattering centers. ISAR signal processing consists of the following steps [18]:

- i. Range compression deconvolves the echoed signal from the transmitted signal, thereby forming the range profile.
- ii. Motion compensation registers the moving targets with respect to the radar.
- iii. Next, the image is constructed by arranging the received signal samples in a polar grid of different viewing angles and Doppler frequencies in frequency spatial domain.

The radar image resolution is decided collectively in range and cross-range directions [19]. The range resolution is inversely proportional to the transmitted electromagnetic signal bandwidth and the cross-range resolution is inversely proportional to the radar antenna aperture size. Finally, the range and cross-range mapped ISAR image is formed by taking the inverse two-dimensional FFT of the sample signals interpolated on the grid. The ISAR technique requires the phase of the received signal to be in coherence. The ISAR processing steps involved in ISAR are shown sequentially in Figure 9.39.

Several algorithms for the ISAR image reconstruction have been proposed in Refs. [20–22]. The simplest of them is the range-Doppler (or range vs. cross-range) technique, which is applicable when the effective rotation vector does not vary with the integration time. The range is defined as the axis parallel to the direction of propagation from the radar to the target. Cross-range is the axis perpendicular to the range direction. If the target is moving or rotating at a constant speed relative to the radar, then the Doppler spectral analysis of the time history of range profiles provides information regarding the target's scattering centers. Thus, the Doppler frequency content and the relative position of scatterers can be determined. While generating range profiles, the location of target scatterers upto within one range cell is obtained. On the other hand, by cross-range processing, the targets residing even within the same range cell can also be separated. In this manner, the range versus cross-range map of the target's scattering centers is formed.



**FIGURE 9.39**  
ISAR processing steps.

While constructing ISAR images, certain parameters need to be controlled to obtain a good quality image. One such enhancement technique is windowing, which smoothens the point spread function (PSF) of the ISAR image [23]. The PSF is the impulse response of an ISAR imaging system to a point scatterer. In the Cartesian coordinate system, if  $B_x$  is the aspect bandwidth and  $B_y$  is the frequency bandwidth, then the PSF is given by

$$\text{PSF}(x, y) = \text{sinc}((B_x/\pi)x)\text{sinc}((B_y/\pi)y) \quad (9.52)$$

The physical meaning of PSF is illustrated in Figures 9.40(a) through (f). Here, seven point scatterers are shown in Figure 9.40(a), and the corresponding PSF in Figure 9.40(b). The resultant ISAR image is the convolution of the point scatterers with the 2-D PSF, shown in Figure 9.40(c). The image in the 2-D range versus cross-range planes is shown in Figure 9.40(e). When a suitable window is applied, the image gets smoothed out as in Figure 9.40(d) and its convolved plot is shown in Figure 9.40(f).

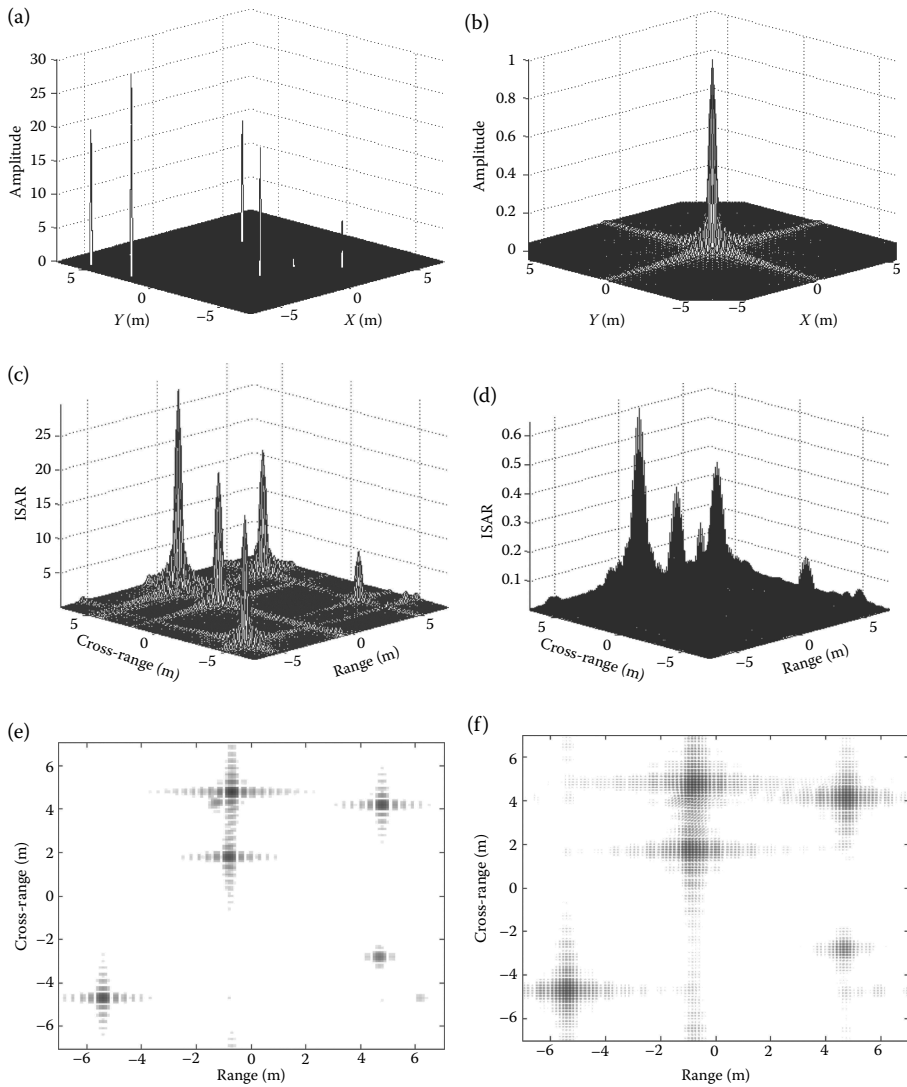
The tails in the PSF that represent the side lobes in the sinc functions (in range and cross-range directions) must be suppressed and this is accomplished by the process of windowing. Prior to windowing, we perform zero-padding on the ISAR readings, which boosts the image quality by interpolating the image data in the frequency-domain. Zero-padding in the time-domain increases the number of sample points (in between the actual samples) in the frequency-domain, and thus makes the reconstruction better by allowing continuous transition. For a 2-D ISAR image, the zero-padding is performed in both the directions. After zero-padding, various window functions are applied with suitable parameters. It is to be noted that though the windows make the ISAR image smoother, the resolution of the image gets poorer.

### 9.7.2 Experimental Analysis

The ISAR image readings are taken corresponding to ISAR aircraft image [23]. The Cartesian coordinate system ( $X$ - $Y$ ) is fixed on the object with range given along the  $Y$ -axis and cross-range along the  $X$ -axis. The center frequency is taken as ( $f_c$ ) and the speed of light is denoted by  $c$  with a value  $3 \times 10^8$  m/s. After range compression, 32 range profiles ( $M$ ) and 64 cross-range profiles ( $N$ ) are taken. The range frequency bandwidth is denoted by  $B_y$  and the cross-range frequency bandwidth by  $B_x$ . Using these, the range/frequency resolution ( $\Delta f$ ) and cross-range/aspect resolution ( $\Delta\phi$ ), which decide the resultant image quality, can be found by

$$\Delta f = c/2B_x \quad (9.53)$$

$$\text{and } \Delta\phi = (\pi/k_c)B_y, \quad (9.54)$$



**FIGURE 9.40** (a) Point scatterers. (b) Point spread function. (c) Convolved image. (d) Convolved windowed image. (e) 2-D ISAR image. (f) Windowed ISAR image.

where  $k_c$  is the wave number for  $f_c$  given by

$$k_c = 2\pi f_c / c. \tag{9.55}$$



Using these, the frequencies ( $f$ ) and the look-angles ( $\phi$ ) are selected as follows:

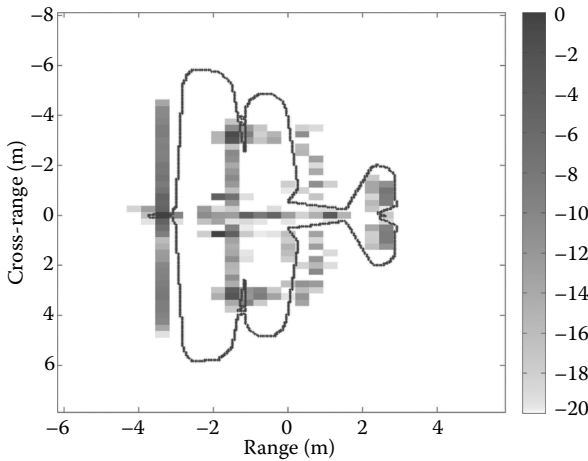
$$f = [(f_c - \Delta f M/2), (f_c - \Delta f (M/2 - 1)), \dots, f_c, \dots, (f_c + \Delta f (M/2 - 1))] \tag{9.56}$$

and  $\phi = [(\phi_c - \Delta \phi N/2), (\phi_c - \Delta \phi (N/2 - 1)), \dots, \phi_c, \dots, (\phi_c + \Delta \phi (N/2 - 1))]$  (9.57)

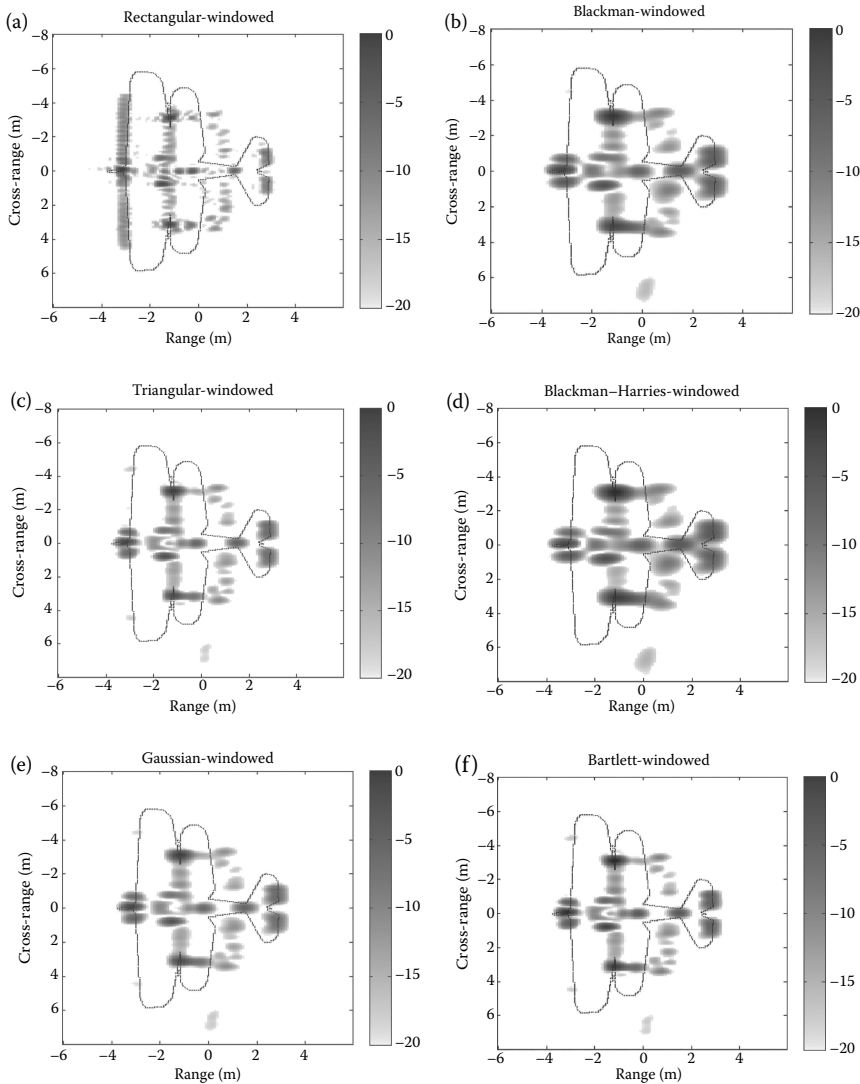
The Equations 9.56 and 9.57 are used for obtaining the backscattered electric field samples denoted by  $E(f, \phi)$ . Then, the two-dimensional inverse Fourier transform integral is applied as

$$\text{ISAR}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(f, \phi) e^{j2\pi(2f/c)x} e^{j2\pi(k_c \phi/\pi)y} d(2f/c) d(k_c \phi/\pi). \tag{9.58}$$

Since, in this study, the backscattered electric field samples are collected within a small frequency and aspect angle bandwidth, inverse fast Fourier transform (IFFT) can be easily applied. The final ISAR image thus formed is shown in Figure 9.41. The backscattered electric field samples for the three radar images are obtained from Ref. [23]. The ISAR images formed are represented in a logarithmic scale of around 20 dB dynamic range. It is observed in the ISAR images that the main backscattering centers are present in the wings, propellers, tires, nose and tail of the aircrafts. The ISAR images are now interpolated by zero-padding four times, which provides continuity in the values of the 2-D ISAR image. On the zero-padded ISAR images, we have tested many windows to observe their performance. On the three ISAR

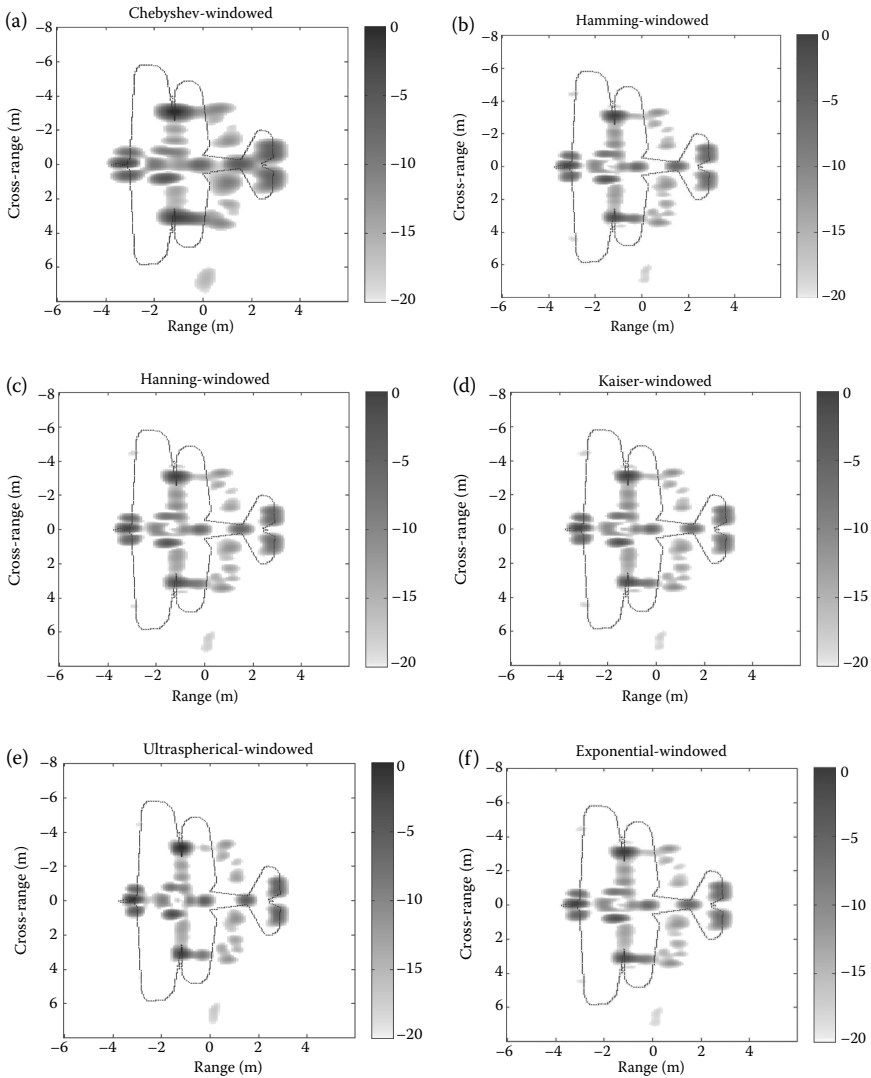


**FIGURE 9.41**  
ISAR image of an aircraft.

**FIGURE 9.42**

Results of windowing on the ISAR image. (a) Rectangular window. (b) Blackman window. (c) Triangular window. (d) Blackman–Harris window. (e) Gaussian window. (f) Bartlett window.

images formed, the windows used are rectangular, Blackman, triangular, Blackman–Harris, Gaussian, Bartlett, Dolph–Chebyshev, Hamming, Hann, Kaiser, ultraspherical, and exponential. The ISAR example in Figure 9.41 is zero-padded and then a window is applied. The results are shown in Figures 9.42 and 9.43.



**FIGURE 9.43** Results of windowing on the ISAR image (continued). (a) Dolph–Chebyshev window. (b) Hamming window. (c) Hann window. (d) Kaiser window ( $\alpha = 4.71$ ). (e) Ultraspherical window ( $x_0 = 1.007, \beta = 0.95$ ). (f) Exponential window ( $\alpha = 4.71$ ).

### 9.7.3 Results and Conclusions

From Figures 9.42 and 9.43, it can be seen that Hamming, Kaiser, exponential, and ultraspherical windows offer comparatively better performance for ISAR images, since they provide the desired smoothing and suppression of unwanted side lobes. We can observe from these figures that Blackman,

Blackman–Harris, and Dolph–Chebyshev windows provide a smudging effect on the ISAR images, thereby drastically decreasing the resolution. Hence, these windows are not preferred. The rectangular window has the least smoothing effect on the ISAR images and, therefore, can be ruled out. If we compare the rest of the windows, it turns out that the Hamming window provides an increase in the contrast ratio of the ISAR image.

Although the triangular and Bartlett windows provide more contrast to the ISAR images than the Hamming window, their smoothing performance is not satisfactory. Gaussian window decreases the resolution of ISAR images considerably (compared to the triangular and Bartlett windows), so using this window is not advisable.

For Kaiser windows, the range of  $\alpha$  that gives better performance is observed to be  $4 \leq \alpha \leq 5$ . The optimum can be chosen to be  $\alpha = 4.71$  ( $1.5\pi$ ). The same is true for the exponential window. The Kaiser window comparisons for different values of  $\alpha$  are shown in Figure 9.44. The parameters for the ultraspherical window can be flexibly chosen in accordance with the position and nature of the target. The optimum values chosen here are  $x_0 = 1.007$  and  $\alpha = 0.95$  as they provide the optimum results required in our application. Thus, the ultraspherical window provides a more compact ISAR image compared to all the other windows considered in this study.

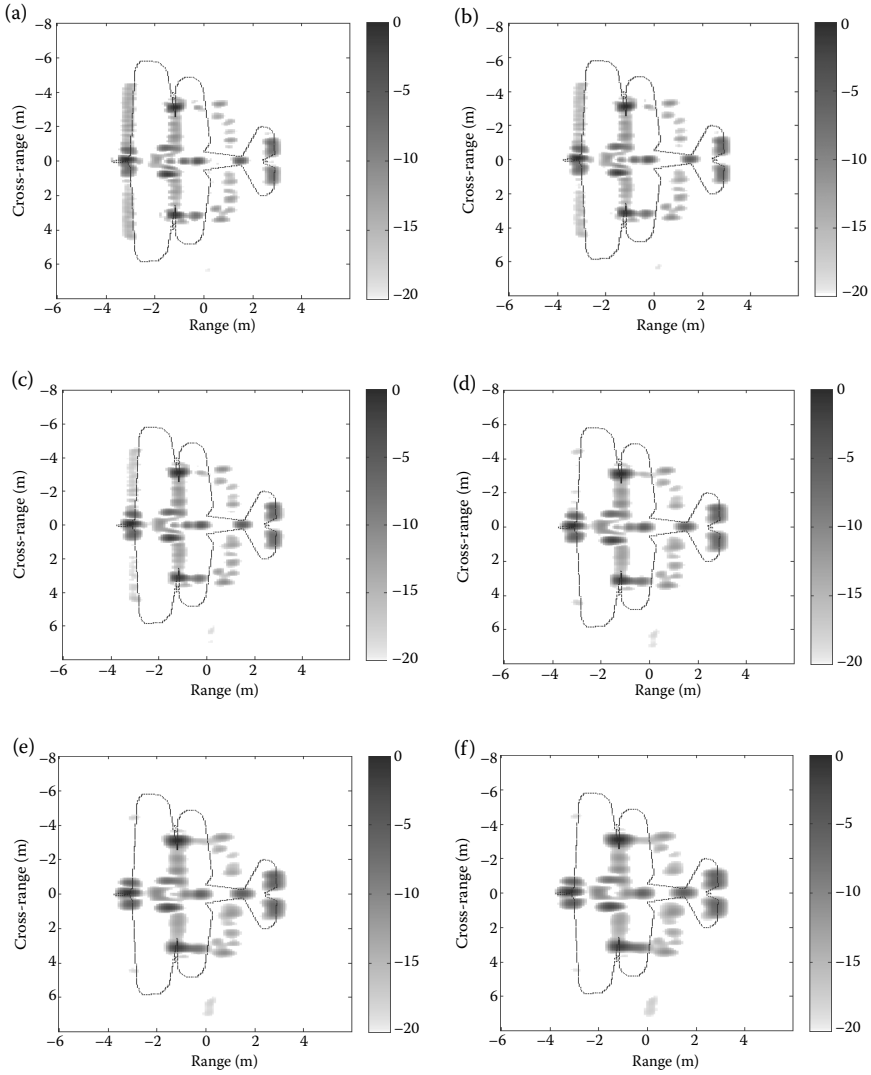
All the windows and their parameters are compared in terms of their smoothing effect on three 2-D ISAR images, taking into consideration the loss of resolution. The specific characteristics of each window on the ISAR images are observed and noted, thereby making it possible for the ISAR image interpreter to apply the desired window depending on the requirement.

---

## 9.8 Windows to Improve Contrast Ratio in Imaging Systems

Image processing has widespread applications in the fields of computer vision, medical imaging, microscope imaging, and radar imaging, wherein the image data can take various forms such as a video sequence, views from multiple cameras, or multidimensional data from a medical scanner. The quality of an image measures the perceived image's degradation. It is very much dependent on the imaging systems used, as they generate distortions or artifacts in the signal readings. Therefore, image quality assessment is quite relevant. One of the image quality metrics that is of utmost significance is the contrast ratio (CR) [24].

An imaging system's CR is defined as the difference between the whitest and the blackest pixel values, in terms of brightness or luminescence, and is usually expressed as a ratio of an integer to unity (integer:1). The CR measure allows an imaging system to detect low-contrast objects lying next to high-contrast objects, which is quite useful in medical imaging applications.



**FIGURE 9.44** Results of using Kaiser window for different values of  $\alpha$ . (a)  $\alpha = 2$ . (b)  $\alpha = 2.5$ . (c)  $\alpha = 3$ . (d)  $\alpha = 4$ . (e)  $\alpha = 5$ . (f)  $\alpha = 6$ .

Imaging systems [25], such as SAR/ISAR, computerized tomography, and charge coupled device-based x-rays construct images using inverse 2-D windowed DFTs on spatial frequency domain data. They are highly influenced by the windows used, as the characteristic of the window has a direct consequence on the contrast of the image.

The frequency response of a window,  $f(n)$ , is given by

$$F(e^{j\omega}) = \sum_{n=0}^{N-1} f(n)e^{-j\omega n} . \tag{9.59}$$

A vital characteristic of a window that is often overlooked is the main-lobe to side-lobe energy ratio (MSR). The MSR of a window is a very important image quality measure as it describes the worst-case energy leakage from the bright to the dark area of the image. Thus, it has a direct impact on the CR tolerance of the imaging systems. For describing the MSR, we consider the parameter of the window ( $\omega_m$ ), which corresponds to the frequency in the main-lobe region with an amplitude equal to the highest side lobe of the window. It is depicted in Figure 9.45 for the ease of understanding. The MSR is represented as

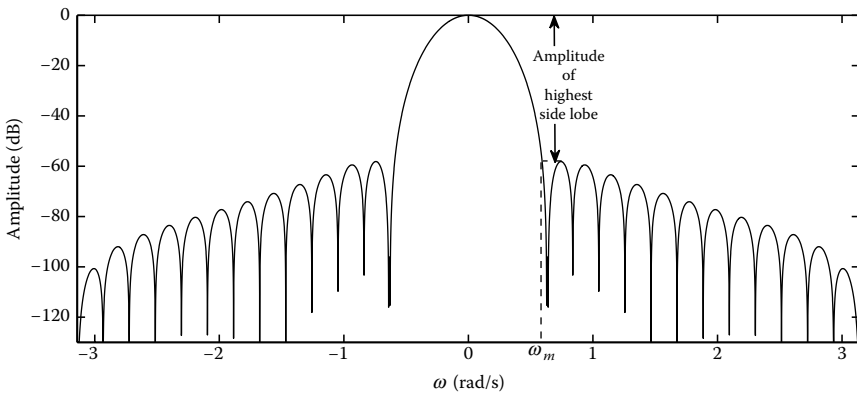
$$\text{MSR} = \frac{\int_0^{\omega_m} |F(e^{j\omega})|^2 d\omega}{\int_{\omega_m}^{\pi} |F(e^{j\omega})|^2 d\omega} . \tag{9.60}$$

The side-lobe energy is given by

$$E_s = 2 \int_{\omega_m}^{\pi} |F(e^{j\omega})|^2 d\omega . \tag{9.61}$$

The main-lobe energy is defined as

$$E_m = 2 \int_0^{\omega_m} |F(e^{j\omega})|^2 d\omega . \tag{9.62}$$



**FIGURE 9.45**  
Position of  $\omega_m$  in a Blackman window.

Hence, the total energy is

$$E_t = E_s + E_m . \tag{9.63}$$

Therefore, the contrast ratio can be defined as

$$CR = \frac{E_s + E_m}{E_s} = 1 + MSR. \tag{9.64}$$

Now, the design and analysis of the windows' parameters affecting the CR is performed by considering the following equation [26]:

$$CR = \frac{\mathbf{f}^T \mathbf{f}}{\mathbf{f}^T \mathbf{Q} \mathbf{f}'} \tag{9.65}$$

**TABLE 9.10**  
CR Values for Different Window Functions

Window Function	CR (in dB)
Rectangular	19.9059
Kaiser–Bessel, $\alpha = 2$	31.4298
Exponential, $\alpha = 2$	40.6247
Kaiser–Bessel, $\alpha = 3$	43.6460
Triangular	50.2682
Bartlett	50.1182
Exponential, $\alpha = 3$	52.1751
Kaiser–Bessel, $\alpha = 4$	57.7346
Exponential, $\alpha = 4$	64.3444
Hann	65.5163
Hamming	68.7951
Kaiser–Bessel, $\alpha = 5$	72.6151
Gaussian	76.6924
Exponential, $\alpha = 5$	77.3568
Kaiser–Bessel, $\alpha = 6$	88.0623
Exponential, $\alpha = 6$	91.2039
Kaiser–Bessel, $\alpha = 7$	103.8475
Exponential, $\alpha = 7$	105.6402
Exact Blackman	114.2239
Kaiser–Bessel, $\alpha = 8$	119.7239
Exponential, $\alpha = 8$	120.6123
Kaiser–Bessel, $\alpha = 9$	135.9583
Exponential, $\alpha = 9$	135.9805
Dolph–Chebyshev	170.0871
Blackman–Harris	178.1653

where  $F(e^{j\omega}) = \mathbf{f}^T \mathbf{v}$  such that  $\mathbf{f}^T = [f(0), f(1), f(2), \dots, f(N-1)]$  and  $\mathbf{v} = [1, e^{-j\omega}, e^{-j2\omega}, \dots, e^{-j(N-1)\omega}]^T$ .

$\mathbf{Q}$  is a real, symmetric, and a positive-definite Toeplitz matrix whose elements are given by

$$q(m, n) = \begin{cases} -\frac{\omega_m}{\pi} \operatorname{sinc}(\omega_s(m-n)), & \text{for } m \neq n \\ 1 - \frac{\omega_m}{\pi}, & \text{for } m = n \end{cases}. \quad (9.66)$$

In this manner, by choosing the appropriate window length ( $N$ ), different windows can be substituted in Equation 9.65 and analyzed for their CR value measurement. The ultraspherical window, which has three parameters, has an advantage over the other windows due to its adjustable side-lobe pattern. This can alter the energy contained in the side lobes, and consequently the CR measure.

### 9.8.1 Experimental Analysis

The contrast ratio of various windows are tabulated so that we can compare their performance. The analysis is performed by applying different windows in Equation 9.65 and comparing their CR performance. The value of  $N$  is 131 in all the cases, while  $\omega_m$  for each window is found independently. Table 9.10 contains all CR values in the ascending order. We seek a window that exhibits the maximum contrast ratio.

### 9.8.2 Results and Conclusions

From Table 9.10, it can be observed that the Blackman–Harris window gives the maximum contrast ratio, whereas the rectangular window provides the least. Among Kaiser–Bessel and exponential windows, for the same value of  $\alpha$ , the exponential window gives a better CR, with  $2 \leq \alpha \leq 9$ . Therefore, from the point of view of contrast ratio, the Blackman–Harris window is found to be the optimum window for imaging systems.

---

## References

1. B.R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, CRC Press, 2nd edn., Florida, USA, 2005.
2. F.J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proceedings*, vol. 66, pp. 51–83, January 1978.



3. H. Urkowitz, J.D. Geisler, and N.A. Ricciardi Jr., The effect of weighing upon signal-to-noise ratio in pulse bursts, *IRE Transactions on Aerospace and Electronic Systems*, vol. AES-9, pp. 486–494, July 1973.
4. C.L. Temes, Side lobe suppression in a range-channel pulse-compression radar, *IRE Transactions on Military Electronics*, vol. MIL-6, pp. 162–169, April 1962.
5. A.H. Nuttall, Some windows with very good side lobe behavior, *IEEE Transactions on Acoustic and Speech Signal Processing*, vol. ASSP-29, pp. 84–91, February 1981.
6. J.D. Bronzino, *The Biomedical Engineering Handbook*, CRC Press, 2nd Edn., Florida, USA, 1999.
7. A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, and H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, circulation electronic pages: <http://circ.ahajournals.org/cgi/content/full/101/23/e215> PMID: 1085218; doi: 10.1161/01.CIR.101.23.e215.
8. K.D. Chinchkhede, S.Y. Govind, S.R. Hirekhan, and D.R. Solanke, On the implementation of FIR filter with various windows for enhancement of ECG signal, *International Journal of Engineering Science and Technology*, vol. 3, pp. 2031–2040, 2011.
9. S. Inban and N. Uchaipichat, Development of QRS detection using short-time Fourier transform based technique, *IJCA, Special Issue on CASCT*, no. 1, pp. 7–10, 2010, published by Foundation of Computer Science.
10. E. de Boer and H.R. de Jongh, On cochlear encoding: Potentialities and limitations of the reverse-correlation technique, *The Journal of the Acoustical Society of America*, vol. 63, no. 1, pp. 115–135, January 1978.
11. G. Yu, S. Mallat, and E. Bacry, Audio denoising by time-frequency block thresholding, *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1830–1839, 2008.
12. Y. Ephraim and D. Malah, Speech enhancement using MMSE short time spectral amplitude estimator, *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. ASSP32, pp. 1109–1121, 1984.
13. C.M. Stein, Estimation of the mean of a multivariate normal distribution, *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, November 1981.
14. L.R. Rabiner and R.W. Schafer, *Theory and Applications of Digital Speech Processing*, Pearson, 1st Edn., New Jersey, USA, 2010.
15. P. Kabal, Time Windows for Linear Prediction of Speech, Technical Report, Department of Electrical & Computer Engineering, McGill University, 2009, Version 2a [online at <http://www-mmssp.ece.mcgill.ca/Documents/Reports/2009/KabalR2009b.pdf>].
16. P. Kabal and R.P. Ramachandran, The computation of line spectral frequencies using Chebyshev polynomials, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1419–1426, December 1986.
17. T.F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, Prentice Hall, New Jersey, USA, 3rd Edn., 2009.
18. B. Haywood, R. Kyprianou, C. Fantarella, and J. McCarthy, *ISARLAB—Inverse Synthetic Aperture Radar Simulation and Processing Tool*, Surveillance Systems Division, Electronics and Surveillance Research Laboratory, Canberra, Australia, June 1999.

19. D.R. Wehner, *High-Resolution Radar (2nd Edition)*, Artech House, Boston, Norwood, Massachusetts, 1994.
20. W.C. Carrara, R.S. Goodman, and R.M. Majewsky, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*, Artech House, Boston, Norwood, Massachusetts, 1995.
21. J. Li, R. Wu, and V.C. Chen, Robust autofocus algorithm for ISAR imaging of moving targets, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 1056–1069, July 2001.
22. F. Berizzi, E. Dalle Mese, and M. Martorella, Performance analysis of a contrast-based ISAR autofocusing algorithm, In *Proceedings of the 2002 IEEE Radar Conference*, Long Beach, CA, pp. 200–205, April 2002.
23. C. Ozdemir, *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*, Wiley, New Jersey, 2012.
24. G.G. Kuperman and T.D. Penrod, Evaluation of compressed synthetic aperture radar imagery, *IEEE National Aerospace and Electronics Conference*, vol. 1, pp. 319–326, May 1994.
25. S. William and A. Bergen, Design of ultraspherical window function with prescribed spectral characteristics, *EURASIP Journal on Applied Signal Processing*, vol. 13, pp. 2053–2065, 2004.
26. J.W. Adams, A new optimal window, *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1753–1769, August 1991.



---

# Index

---

## A

- Action potential (AP), 323
- Additive white Gaussian noise (AWGN), 324
  - EEG signal, 325
  - synthetic, 329
- Aliasing, 40, 59, 60, 63, 64; *see also* Reconstruction
- All-pole model, *see* Autoregressive model (AR model)
- Amplitude response, 234
  - of bandpass filters, 78
- Analysis formula, *see* Discrete-time Fourier transform (DTFT)
- Antenna design community, 155
- Apodization, 96
- AR model, *see* Autoregressive model (AR model)
- Audio denoising
  - SNR comparison, 344
  - using time–frequency plane, 336–340
  - windows effect, 342–343
- Autocorrelation function (ACF), 180, 268
- Autoregressive model (AR model), 344
- Average transmitted power (ATP), 303

## B

- Bandpass filter (BPF), 221
  - impulse response of, 221
  - type 3 FIR filter, 239
- Bandstop filter (BSF), 221
- Bandwidths, 3 dB and 6 dB, 166–167
- Bartlett periodogram; *see also* Welch periodogram method
  - bias of, 279
  - high variance, 278
  - resolution of, 280, 282
  - variance of, 279, 281
- Bartlett windows, 89, 148, 357
  - ISAR image, 355

- Jackson window, 155
  - LSF coefficients, 350
  - parameters, 93
  - self-convolution property, 91
  - window functions, 90
- Bias, 259
- of Bartlett periodogram, 279
  - of Blackman–Tukey method, 289
  - modified periodogram, 267
  - MSE, 259
  - of periodogram, 262
  - problems of, 181
  - on spectral amplitudes, 131
  - of Welch periodogram, 281
- Biomedical signal processing, 322; *see also* Stretch processing
- applications, 323
  - FIR filtering, 324–328
  - frequency and dynamic ranges, 323
  - MA filter, 328–333
  - QRS detection, 333–336
  - techniques, 324
- Blackman–Harris window, 93, 109
- Nuttall window family, 151
  - rectangular window to, 169
  - simulation results, 306
  - time-and frequency-domain plots, 110
  - variations of, 151
  - windowing on ISAR image, 355
- Blackman–Tukey method
- ACF estimate, 286
  - autocorrelation sequence, 291
  - bias of, 289
  - PSD estimate, 283
  - resolution of, 289, 290, 291, 292
  - sinusoidal components, 290
  - spectral estimation, 181
  - spectral estimator, 288
  - variance of, 289, 290, 291, 292
- Blackman window, 106–107
- Blackman–Harris window family, 150

- Blackman window, 106–107 (*Continued*)
  - CSD window, 202
  - Fourier transformation, 106
  - functions, 108
  - MSR, 359
  - parameters, 93
  - SCER performance, 194
- Block thresholding technique, 340
  - adaptive, 342
  - musical noise, 341
  - noise variance, 342
- Bohman window, *see* Papoulis window
- Box car window, *see* Rectangular window
- BPF, *see* Bandpass filter (BPF)
- BSF, *see* Bandstop filter (BSF)
- Burst, 303, 304, 314
  
- C
- Canonic signed digit windowing (CSD windowing), 195–196
  - binary windows performance, 216
  - implementations, 197
  - using smoothed spectrum, 196–197
  - technique code, 195
  - window 10, 209, 210
  - window 11, 209, 211
  - window 1, 197, 198
  - window 12, 209, 212
  - window 13, 210, 211, 213
  - window 14, 211, 212, 214
  - window 2, 197, 198, 199
  - window 3, 198, 199, 200, 201–202
  - window 4, 202, 203
  - window 5, 202, 204
  - window 6, 202, 205
  - window 7, 203, 204, 205, 206
  - window 8, 205, 206, 207
  - window 9, 207, 208
- Cauchy window, 154
- C/D converter, 63, 65
- Characteristics, window functions, 87–88
- Circular convolution, 39, 80–81
  - by linear convolution, 83
  - linear convolution aliased form, 82
- Circular shifting property, 38
- Cochleagram, 339
- Coherent gain, 131, 161, 166
- Coherent power gain, 166
- Computational error
  - arithmetic operations, 194
  - input data samples, 192
  - quantization errors, 195
  - quantized data samples, 192–193
  - SCER, 193, 194
  - in time and frequency-domains, 192
- Computational savings, DIT-FFT, 53–54
- Computation, window parameters, 132–139
  - denominator, 138
  - NHMLW *vs.* MSL, 139, 140
  - performance comparison, 134–135, 136, 137
  - rest of parameters, 138
  - using trigonometric identities, 133
  - truncated Taylor family, 133
  - variance compensation factor, 133
- Conjugation property, 38
- Consistency, 260
- Continuous-time Dolph–Chebyshev weighting functions, 114
- Continuous-time Fourier transform (CTFT), 1; *see also* Discrete-time Fourier transform (DTFT)
  - basic CTFT pairs, 17
  - complex exponential formula, 6–7
  - complex one-sided exponential signal, 9, 10
  - conjugation and conjugate symmetry property, 8
  - convolution property, 9
  - differentiation and integration property, 8
  - differentiation property application, 15–16
  - duality property application, 9, 16
  - exponential signal
    - determination, 10, 11
  - frequency-domain conversion, 7
  - linearity property, 7
  - modulation property, 9
  - Parseval’s theorem, 9
  - rectangular signal
    - determination, 12, 13
  - scaling property, time and frequency, 8

- shifted rectangular pulse signal, 13, 14
    - shifting property, time and frequency, 7–8
    - sinusoidal signals, 11–12, 13
    - time-domain signal, 14, 15
  - Continuous-time signal (CT signal), 1
    - engineering applications, 60
    - Fourier transform of, 64
    - frequency-domain, 21
    - Nyquist rate, 68
    - reconstruction of, 69
    - sampling, 59, 63
  - Contrast ratio (CR), 357
    - experimental analysis, 361
    - results, 361
    - values for window functions, 360
  - Convergence, 19, 32, 33
  - Convolution property, 9
    - circular, 39
    - DTFTs, 20
  - Correlogram methods, 177, 260, 283
  - $\text{Cos}(x)$  window, 91, 92, 93
  - $\text{Cos}^3x$  window, *see* Sum-cosine window
  - $\text{Cos}^4(x)$  window, 101, 102
  - Cosine-tapered window, *see* Tukey window
  - $\text{Cos}^\alpha x$  window family, 148
  - CR, *see* Contrast ratio (CR)
  - CSD windowing, *see* Canonic signed digit windowing (CSD windowing)
  - CTFT, *see* Continuous-time Fourier transform (CTFT)
  - CT signal, *see* Continuous-time signal (CT signal)
- D**
- Daniel periodogram, 292–293
  - Data smoothing, *see* Data windowing
  - Data window, 130, 145, 146, 147
    - application of, 131
    - coefficients of, 197
    - discrete-time, 231
    - Fourier transform of, 132
    - input observations, 130
    - optimal, 142
    - performance comparison, 141
    - rectangular, 70
    - use of, 84
  - Data windowing, 146
  - Decibels (dB), 124
  - Decibels per octave (dB/oct), 170
  - Decimation-in-frequency FFT (DIF-FFT), 54
    - 4-point FFT using, 55
    - even-and frequency sample, 55
    - flow graph, 56
    - in-place computation, 55
    - linear convolution, 57, 58
    - $m$ th Stage butterfly, 56
    - odd-numbered frequency samples, 55
  - Decimation-in-time FFT (DIT-FFT), 50–51
    - butterfly structures, 51, 53
    - computational savings, 53–54
    - decomposition, 52
    - elementary computation, 51, 52
    - in-place computation, 54
    - linear convolution, 57, 58
  - Degradation loss, 132, 166, 169
  - Denominator (DEN), 138
  - DFT, *see* Discrete Fourier transform (DFT)
  - DIF-FFT, *see* Decimation-in-frequency FFT (DIF-FFT)
  - Differentiation property, 8
    - application, 15–16, 26
    - in frequency-domain, 21
  - Digital filters, 224
    - coefficients, 224
    - design of FIR, 129
    - with feedback terms, 223
    - purpose of, 142
  - Digital signal processors (DSPs), 1, 36
  - Dirichlet conditions, 6
  - Discrete-time (DT), 1
    - data window, 231
    - discrete-time system, 4
    - Fourier transform, 17
    - LTI systems, 222
    - periodic signals, 18
    - sequences, 21
    - signals, 3
    - sinusoidal signal, 61

- Discrete-time Fourier transform (DTFT),
    - 1, 17
    - application of differentiation
      - property, 26
    - basic DTFT pairs, 30
    - discrete-time periodic signals, 18–19
    - finite-length exponential sequence, 27
    - Fourier series expansion, 22
    - frequency-domain representation, 18
    - of Hamming window, 170
    - Hann window, 28–29
    - impulse response, 27–28
    - properties, 19–22
    - rectangular pulse signal
      - determination, 23, 24, 25
    - sum of complex exponentials, 18
    - types of convergence, 19
  - Discrete-time windows, 145, 148; *see also* Window functions
    - Blackman–Harris window, 151
    - Blackman window, 150
    - Cauchy window, 154
    - $\text{Cos}^a x$  window family, 148
    - Dolph–Chebyshev window, 155
    - even-length windows, 158, 159
    - flat-top window, 152–153
    - Gaussian window, 153–154
    - Hamming window, 149
    - Hann window, 148
    - Hann–Poisson window, 154
    - Jackson window, 155
    - modified first-order Kaiser–Bessel function, 156
    - modified zeroth-order Kaiser–Bessel window, 155–156
    - Nuttall window family, 151
    - odd length windows, 158, 159
    - optimized Blackman window, 150
    - Papoulis window, 154
    - Parabolic window, 153
    - Poisson window, 153
    - raised-cosine window family, 149–150
    - rectangular window, 148
    - Riemann window, 153
    - Saramäki window family, 156–157
    - sum-cosine window, 149
    - triangular window, 148
    - truncated Taylor family, 149
    - Tukey window, 150
    - ultraspherical window, 157–158
  - Discrete Fourier transform (DFT), 1, 36, 129; *see also* Fast Fourier Transform (FFT)
    - circular convolution, 46, 47–48
    - comparison, 44–45, 46
    - filter bank, 76–77
    - frequency components, 189
    - interpolation formula, 37
    - linear convolution, 45, 49
    - magnitude response plots, 78
    - output response, 73
    - periodic cosine sequence, 44
    - pitfalls, 59
    - properties, 37–40, 49
    - rectangular pulse, 40–41, 43
    - relation between DTFT and 36–37
    - sampling and truncation, 59
    - symmetry properties, 50
    - two time-domain sequences, 42
    - zero-interpolated signal, 43, 44
    - zero leakage case, 76
  - Dispersion factor, 131, 132, 138
  - DIT-FFT, *see* Decimation-in-time FFT (DIT-FFT)
  - Dolph–Chebyshev window, 155
    - equiripple characteristics, 185
    - Fourier transform, 114
    - side-lobe level, 155
    - smudging effect, 357
    - time-and frequency-domain plots, 116
    - variable parameter, 157
    - weak component response, 187
    - windowing on ISAR image, 356
  - Doppler side lobes, 306
  - DSPs, *see* Digital signal processors (DSPs)
  - DT, *see* Discrete-time (DT)
  - DTFT, *see* Discrete-time Fourier transform (DTFT)
  - Duality property, 9
    - application, 16
    - DFT, 38
- E**
- Electro-oculogram (EOG), 323
  - Electrocardiogram (ECG), 323
  - Electroencephalogram (EEG), 323

- Electrogastrogram (EGG), 323
- Electromyogram (EMG), 323
- Electroneurogram (ENG), 323
- Electroretinogram (ERG), 323
- Equivalent bandwidth, 171
- Equivalent noise bandwidth (ENBW), 160–161
- Equivalent noise bandwidth, 131
- Equivalent time width, 171
- Even symmetric impulse response
  - with even filter length, 235
  - with odd filter length, 235
- Even-length windows, 158, 159
- Event-related potential (ERP), 323
  
- F**
- Fast Fourier Transform (FFT), 1, 50; *see also* Continuous-time Fourier transform (CTFT)
  - decimation-in-time FFT, 50–54
  - DFT, 49, 50
  - inverse DFT from FFT, 56, 57
  - linear convolution, 57, 58
  - processor, 189
- FDI, *see* Frequency-domain implementation (FDI)
- Filter length, 227
  - data, 78, 84
  - digital optimization techniques, 228
  - direct truncation with, 233
  - even, 235
  - odd, 235
- Filtering, 219
  - bandpass, 262
  - biomedical signals, 324
  - frequency-domain, 68
  - linear, 293
  - lowpass, 265
  - MA filter, 328–333
  - multirate, 224
  - optimal, 324
- Finite impulse response filter design, 178
  - advantages of, 225
  - amplitude response, 234
  - artifacts and interferences, 324
  - convolution operation, 175
  - digital filters, 142
  - discrete-time convolution, 226
  - filter cut-off, 326
  - filter specifications, 227–228
  - flat-top window, 325
  - frequency response, 236
  - Hamming window, 175, 179
  - Hann window, 175, 177, 178
  - HPF design, 240
  - ideal filters, 219–221, 239, 240
  - impulse response, 173, 174, 236, 237
  - IRT method, 173, 174
  - linear time invariant systems, 222–224
  - linear-phase filters, 172, 229–234, 238
  - log magnitude frequency response, 176, 180, 181
  - lowpass filter frequency response, 172–173, 175
  - LTI system, 224
  - magnitude response, 241, 242
  - methods, 228
  - noisy and bandpass-filtered ECG signals, 325, 326, 327
  - nonrecursive filter response, 173
  - number of taps, 225
  - pole-zero plots, 239
  - realization of, 227
  - rectangular window, 174, 175, 176
  - response using rectangular window, 176
  - shifted impulse response, 174
  - SNR, 328
  - time-domain and frequency-domain representations, 325
  - triangular window-based FIR filter, 327
  - using windowing approach, 175
- Finite-length sequence, 82
  - DFT properties, 49, 50
  - DTFT, 76
  - linearity property, 38
- First side-lobe level (FSL), 88, 170, 198
  - data window parameter, 93
  - Hamming window, 182
  - Hann window, 182
  - MSLL, 89
  - rectangular window, 275
  - weak harmonics, 169
- Fixed windows, 93, 126
- Flat-top window
  - fifth-order, 152



- Flat-top window (*Continued*)
    - frequency response, 324
    - LSF coefficients, 350
    - time-domain and frequency-domain, 325
  - FM, *see* Frequency modulation (FM)
  - Formant extraction, 295
  - Fourier atom, 339, 342
  - Fourier series, 6
    - coefficients, 12
    - continuous-time periodic signal, 11
    - convergent, 18
    - representation, 17
    - truncation of, 173
  - Fourier transforms (FTs), 1
    - convolution, 63
    - raised-cosine and rectangular windows, 103
    - rectangular and cosine windows, 98, 99
    - sum-cosine window, 100
  - Frequency leakage, 59, 189
    - data windowing, 146
    - smearing of energy, 73
  - Frequency leakage effect, 69; *see also* Picket-fence effect
    - CTFT, 70
    - example, 74–76
    - magnitude response and leakage, 71
    - maximum leakage case, 73, 74
    - rectangular data window, 70
    - spectral leakage, 74
    - zero leakage case, 71–72, 73
  - Frequency modulation (FM), 303
  - Frequency resolution, 84, 85
    - dispersion factor, 130
    - loss of, 71
    - main-lobe-broadening factor, 316
    - signals, 260
    - spectral estimates, 84
    - zeropadding, 84
  - Frequency sampling, 228, 293
  - Frequency scaling property, 8
  - Frequency shifting property, 7–8, 20
  - Frequency window, 145
    - inverse Fourier transform, 146
    - side lobes, 231
    - window functions, 147
  - Frequency-domain implementation (FDI), 192
    - arithmetic operations, 194
    - computational error in, 192
    - input data samples, 192
    - quantization errors, 195
    - quantized data samples, 192–193
    - SCER, 193, 194
  - FSL, *see* First side-lobe level (FSL)
  - FTs, *see* Fourier transforms (FTs)
- G**
- Gammatone atoms, 339
  - Gaussian process
    - functions of, 142
    - random process, 264
  - Gaussian window
    - ISAR image, 355, 357
    - main-lobe width, 154
    - STFT magnitude, 336
    - time function, 153
  - Gibbs number, 232, 251
- H**
- Half-power bandwidth
    - main-lobe broadening factor, 315
    - normalized, 132
    - 3 dB bandwidth, 131
  - Hamming window, 93, 95, 96, 122–123
    - DTFT of, 170
    - effect, 175
    - FIR lowpass filter response, 179
    - ISAR images, 357
    - log magnitude response, 181
    - magnitude response of signal, 186
    - PSLL, 149
    - SCER performance, 194
    - weak component response, 187
    - window 2, 197
  - Hann window, 123, 148
    - effect, 175
    - FIR lowpass filter response, 178
    - Fourier transform pair, 91
    - log magnitude response, 180
    - magnitude response of signal, 186
    - parameters, 93
    - SCER performance, 193

- in short-hand notation, 94
  - in time and frequency-domain, 177
  - transition bandwidth and ripples, 175
  - weak component response, 187
  - window functions, 92, 197
  - Hann–Poisson window, 154
  - Hertz (Hz), 61
  - High range resolution (HRR), 303
    - HRR radars, 303
    - HRR target profiling, 303–305
    - IDFT, 305
    - pulse of burst, 303
    - simulation results, 305–306
    - target's reflection, 304
    - windows in, 303
  - Highpass filter (HPF), 220; *see also*
    - Lowpass filter (LPF)
    - frequency response of, 220
    - impulse response, 221
  - Hilbert transformer design
    - filter types, 256
    - frequency response, 254, 255
    - using impulse response truncation, 253
  - HPF, *see* Highpass filter (HPF)
  - HRR, *see* High range resolution (HRR)
- I**
- Ideal filters
    - BPF, 221
    - BSF, 221
    - HPF, 220–221
    - LPF, 219–220
  - IDFT, *see* Inverse discrete Fourier transform (IDFT)
  - IDTFT, *see* Inverse discrete-time Fourier transform (IDTFT)
  - IFFT, *see* Inverse fast Fourier transform (IFFT)
  - IIR filters, *see* Infinite impulse response filters (IIR filters)
  - Image processing, 349
    - CR in, 357–361
    - experimental analysis, 352–356
    - results, 356–357
    - windows for ISAR images, 350–352
  - Impulse response
    - antisymmetric, 235
    - BSF, 221
    - coefficients of, 227
    - design of differentiator, 252, 253
    - design of Hilbert transformer, 253–256
    - discrete-time convolution, 226
    - FIR filters, 228
      - of HPF, 221
    - ideal discrete-time differentiator, 27
    - ideal Hilbert transformer, 27
    - ideal lowpass filter and, 173
    - infinite length, 172
    - of linear-phase filters, 231
    - LTI system, 5, 20
    - PSF, 352
    - shifted, 174
    - truncation of, 174
    - type 2 FIR, 236
    - type 3 FIR, 237
    - type 4 FIR, 237
  - Impulse response truncation method (IRT method), 173, 228
    - design of differentiator, 252
    - frequency response, 253
    - half-delay, 253
    - Hilbert transformer design, 253–256
  - In-place computation, 54, 55
  - Infinite impulse response filters (IIR filters), 172; *see also* Finite impulse response filters (FIR filters)
    - coefficient, 225
    - properties of, 226
  - Initial-value theorem, 34
  - Integration loss, 316
  - Integration property, 8
  - Interpolation formula, 83
    - DFT, 37, 84
    - sampling, 83
  - Inverse discrete Fourier transform (IDFT), 1, 36
    - block diagram for, 57
    - from FFT, 56
    - synthesis formula, 18
  - Inverse discrete-time Fourier transform (IDTFT), 1, 18, 27, 28
  - Inverse fast Fourier transform (IFFT), 354

- Inverse synthetic aperture radar (ISAR), 349
  - aircraft image, 354
  - image readings, 352
  - processing steps, 351
  - signal processing, 350
  - windowing results, 355, 356
  - windows for, 350
- IRT method, *see* Impulse response truncation method (IRT method)
- J**
- Jackson window, *see* Parzen window
- Just leakage, *see* Frequency leakage
- K**
- Kaiser–Bessel first-order window
  - comparison with Zeroth-order window, 251
  - filter design procedure, 250
  - time-domain expression, 249
- Kaiser–Bessel window
  - FIR filter design, 272
  - first-order window, 249–251
  - using Hamming window, 277
  - using Hann window, 277
  - spectral analysis using, 271, 278
  - time-domain, 275
  - weakest signal component, 276
  - window shape parameter, 274
- Kaiser–Bessel zeroth-order window
  - bandpass filter design, 249
  - BPF design, 248
  - error plot, 248
  - LPF design, 246, 247
  - magnitude responses, 244, 245, 247
  - pair of equations, 244
  - parameters, 243
  - shape parameter, 245
- Kaiser windows, 357, 358
- Kaiser’s modified first-order Bessel window function
  - family, 116, 117
  - parameters, 120
  - time-and frequency-domain plots, 118
  - window functions, 119
- Kaiser’s modified zeroth-order Bessel window function
  - family, 116
  - optimum window function, 115
  - parameters, 118
  - window functions, 117
- L**
- Lag window, 146
  - autocorrelation of data window, 147
  - Bartlett, 290
  - Blackman–Tukey method, 288
  - effect of, 289
  - rectangular, 146
  - 6 dB bandwidth, 268
  - symmetrical, 181
- LCCDE, *see* Linear constant coefficient difference equation (LCCDE)
- Leakage
  - in finite extent data, 71
  - frequency, 59, 189
  - low spectral, 185
  - maximum leakage case, 73–76
  - spectral, 129, 343
  - worst-case energy, 359
  - zero leakage case, 71–73
- LF, *see* Loss factor (LF)
- LFM, *see* Linear frequency modulation (LFM)
- Line spectral frequencies (LSFS), 346
  - coefficients, 349, 350
  - pedestal height, 348
  - polynomial, 347
  - speech signal, 347
  - vocal tract models, 346
- Linear constant coefficient difference equation (LCCDE), 34
  - discrete-time systems, 222
  - rational transforms, 33
- Linear convolution
  - using circular convolution, 39–40, 83
  - DFT and IDFT, 49, 81
  - using DIT-FFT and DIF-FFT, 57–58
  - multiplication, 45
  - result, 46
  - sequence samples, 48, 81
- Linear frequency modulation (LFM), 303
  - coefficient, 318

- high-bandwidth, 317
  - pulse compression, 303, 306
  - Linear prediction of speech, 343
    - LPC, 344–346
    - LSFS, 346–347
    - LSF variation, 347–349
  - Linear predictive coding (LPC), 344
    - advantages of, 346
    - coefficients, 346
    - limitations of, 346
    - magnitude response comparison, 345
    - pole zero plot of, 347
    - polynomial, 345
    - windowed frame, 344
  - Linear shift invariant systems (LSI systems), 222
  - Linear time invariant systems (LTI systems), 222
    - analysis of, 9
    - convolution sum expression, 224
    - digital filters, 224
    - FIR filters, 223
    - impulse response, 5, 20
    - input and output, 224
    - LCCDE, 222
    - orthogonal basis signals, 6
  - Linear-phase filters
    - filter characteristics, 233
    - filter length, 230
    - FIR filters, 233
    - frequency response, 229
    - ideal LPF with, 230
    - impulse response of, 231
    - periodic convolution, 231
    - truncation of impulse response, 232
    - types, 238
  - Linearity property, 19
    - DFT, 38
    - DTFT, 19, 29
    - finite-length sequences, 38
    - integration, 7
    - transformations, 4
    - z-transform, 36
  - Loss factor (LF), 309
    - expression for, 306, 309
    - output signal component, 310
    - sequence of pulses, 310
    - SNR ratio, 311
    - weighting functions, 312, 313, 316
  - Lowpass filter (LPF), 219; *see also*
    - Highpass filter (HPF)
    - cutoff frequency, 68
    - DTFT, 220
    - frequency response, 176, 219, 220
    - using Hamming window, 179
    - using Hann window, 178
    - ideal, 64
    - impulse response, 175
    - MA filter, 328
    - magnitude response of, 251
    - using rectangular window, 176
    - SNR, 329
    - windowing method, 175
  - LPC, *see* Linear predictive coding (LPC)
  - LPF, *see* Lowpass filter (LPF)
  - LSFS, *see* Line spectral frequencies (LSFS)
  - LSI systems, *see* Linear shift invariant systems (LSI systems)
  - LTI systems, *see* Linear time invariant systems (LTI systems)
- ## M
- MA filter, *see* Moving average filter (MA filter)
  - Magnitude response, 241
    - of bandpass filter, 249
    - with Blackman window, 242
    - cosine signal, 71
    - DTFT, 83
    - FT, 11
    - with Hamming window, 181, 241
    - with Hann window, 180, 242
    - of lowpass filter, 251
    - rectangular window, 175, 180
    - spectrogram, 295
    - time-shifting property, 20
    - ultraspherical window, 158
  - Main-lobe broadening factor, 315, 316
  - Main-lobe energy (MLE), 88, 93, 142, 359
    - Kaiser's window, 124
    - ratio of, 88, 123
    - sum-cosine window, 125
  - Main-lobe peak, 107
    - Fourier transform, 89
    - FSLL, 111

- Main-lobe peak (*Continued*)  
 maximum side-lobe amplitudes, 116  
 MSL, 88, 124
- Main-lobe to side-lobe energy ratio (MSR), 359
- Main-lobe width, 131, 168  
 Blackman window, 109  
 data window, 79  
 dispersion factor, 131  
 Dolph–Chebyshev window, 114  
 expense of, 94  
 FIR lowpass filter, 175  
 frequency response, 156  
 frequency window, 231  
 Hann window, 241  
 Kaiser’s modified zeroth-order Bessel family, 139  
 normalized, 102  
 PSL, 142  
 rectangular window, 182, 343  
 transition bandwidth, 240  
 variable, 105
- Maximum leakage case  
 frequency leakage, 73  
 leakage effect, 74–76  
 leakage problem, 74  
 spectral leakage, 74
- Maximum side-lobe level (MSLL), 132;  
*see also* Peak side-lobe level (PSL)  
 data windows parameters, 93  
 Kaiser’s modified zeroth-order Bessel family, 118  
 main-lobe peak, 88, 107  
 NHMLW, 139  
 parameters, 142  
 time-and Fourier-domain plots, 96
- Mean square error (MSE), 259
- Microwave imaging systems, 349
- MLE, *see* Main-lobe energy (MLE)
- Modified first-order Bessel family, 171  
 Kaiser–Bessel function, 156  
 parameters of Kaiser’s, 120, 125  
 performance comparison of, 137, 165
- Modified limb lead II (MLII), 324
- Modified periodogram, 265  
 using Bartlett window, 269, 272  
 bias of, 267  
 using Blackman window, 270, 275  
 using Hamming window, 270, 274  
 using Hann window, 269, 273  
 PSD estimator, 265  
 using rectangular window, 271  
 resolution of, 268  
 sinusoidal components, 268  
 variance of, 267
- Modified zeroth-order Bessel family, 142, 171  
 binary windows performance, 216  
 continuous-time Fourier transform pair, 213  
 frequency response, 215  
 Kaiser–Bessel window family, 155–156, 213  
 normalized magnitude plots, 214  
 performance comparison of, 136, 164
- Modulation property, 9  
 dual of convolution property, 9  
 of Fourier transform, 289  
 window sequence, 21
- Moment factorizing theorem, 264
- Moving average filter (MA filter), 328  
 EMG signal, 329  
 neighboring samples, 328  
 noisy and MA-filtered EEG signals, 330, 331, 332  
 SNR, 332
- MSE, *see* Mean square error (MSE)
- MSLL, *see* Maximum side-lobe level (MSLL)
- MSR, *see* Main-lobe to side-lobe energy ratio (MSR)
- Multirate filtering, *see* Zooming filtering
- Musical noise, 336, 340, 341
- N**
- Near-optimum Kaiser–Bessel window, 142, 214
- Near-optimum window family, 105, 156
- Near-optimum window function family, *see* Zeroth-order Bessel family
- NHMLW, *see* Normalized half main-lobe width (NHMLW)
- Nonparametric methods, 260  
 Bartlett Periodogram, 278–280  
 Daniel periodogram, 292–293  
 FFT computation, 293, 294

- Fourier transform techniques, 260
  - modified periodogram PSD estimator, 265–271
  - periodogram PSD estimator, 260–265
  - spectral analysis, 271–278
  - STFT, 293–297
  - Welch periodogram method, 280–283
  - Normalized half-power bandwidth, 132, 167
  - Normalized half main-lobe width (NHMLW), 88, 197
    - Bessel window, 125
    - Blackman windows, 202
    - data windows comparison, 140, 141
    - data windows parameters, 93
    - Fourier transform, 132
    - Kaiser's modified first-order Bessel family, 139
    - rectangular window, 89
  - Nuttall window family, 151
  - Nyquist frequency, 64
  - Nyquist rate, 64
  - Nyquist–Shannon sampling theorem, 64, 65
    - frequency-scaled version, 67
    - FT, 66
    - relationship, 65
    - summation and integration, 66
- O**
- Odd length windows, 158, 159
  - Odd symmetric impulse responses, 235
  - Optimal method, 228, 229
  - Optimized Blackman window, 107
    - coefficients, 150
    - Fourier transform, 108
    - using numerical techniques, 109
    - parameters, 93
  - Optimum window function, 115
  - Overlap correlation, 169
- P**
- Papoulis window, 93
    - discontinuous, 154
    - time-limited function of, 111
    - window functions, 112
  - Parabolic window, 93, 110–111
    - discontinuous first derivative, 153
    - window functions, 112
  - Parseval's theorem
    - CTFT, 9
    - DTFT, 21
    - time-domain, 161
    - Z-transform, 34
  - Parzen window, 93, 113, 114
  - Passband cut-off frequency, 227
  - Passband ripple, 250
  - PCG, *see* Phonocardiogram (PCG)
  - Peak passband deviation, 227
  - Peak side-lobe level (PSLL), 170, 197, 198
    - asymptotic decay rate, 197
    - decrease in, 142
    - filter characteristics, 233
    - Hamming window, 149
    - using Kaiser–Bessel window, 298
    - rectangular window, 73
    - weak harmonics, 169
    - zero frequency, 132
  - Pedestal height, 348
    - function of, 95
    - of windows, 348
  - Performance comparison
    - data windows, 140, 141
    - modified first-order Bessel family, 137
    - modified zeroth-order Bessel family, 136
    - windows, 134
  - Periodic cosine sequence, 44
  - Periodic sampling, 41, 60
  - Periodicity property, 19, 191
  - Periodogram methods, 177, 260, 267, 268;
    - see also* Modified periodogram
    - autocorrelation estimate, 261
    - bias of, 262
    - Gaussian random process, 264
    - peaks in, 179
    - PSD, 260
    - resolution of, 264
    - sinusoidal components, 265
    - spectral estimation, 266
    - variance of, 263
    - Welch* method of, 178
  - Phase modulation (PM), 303
  - Phase response
    - FIR filters, 234

- Phase response (*Continued*)  
 of FT, 10  
 of ideal LPF, 229, 230  
 zero-phase filters, 219
- Phonocardiogram (PCG), 323
- Picket-fence effect, 59, 168, 169; *see also*  
 Frequency leakage effect  
 finite-length data, 84  
 inability of DFT, 78  
 rippled curve, 79
- PM, *see* Phase modulation (PM)
- Point spread function (PSF), 352, 353
- Poisson window, 153, 154
- Power spectral density (PSD), 160  
 Blackman–Tukey spectral  
 estimation, 181  
 modified periodogram estimate, 270  
 of periodogram, 177, 260  
 variance, 259  
 Wiener–Khinchin theorem, 180, 181
- Power spectral estimator, 259
- PRI, *see* Pulse repetition interval (PRI)
- Processing gain, *see* Degradation loss
- Programmable windowing technique  
 DFT, 191  
 frequency-domain  
 implementation, 192  
 time function, 190
- Prolate-spheroidal wave function, 115  
 modified first-order Bessel family, 118  
 near-optimum windows, 139  
 zeroth-order, 116, 156
- PSD, *see* Power spectral density (PSD)
- PSF, *see* Point spread function (PSF)
- PSLL, *see* Peak side-lobe level (PSLL)
- Pulse compression, 303  
 goals of HRR, 306  
 LFM, 306  
 linear FM, 307  
 radar applications, 303  
 range side-lobe reduction, 316
- Pulse repetition interval (PRI), 303
- Pulse sequence  
 pulse compression, 309  
 weighting and summing of, 310
- Q**
- QRS wave detection  
 adaptive thresholding, 334  
 ECG signal, 334, 337, 338, 339  
 QRS locations, 337, 338, 339  
 STFT, 333
- R**
- Radar cross section (RCS), 306
- Radar image, 350  
 backscattered electric field  
 samples, 354  
 resolution, 351
- Radix-2 FFT algorithm, 51, 80
- Raised-cosine family, 105, 106  
 asymptotic attenuation, 104, 105  
 Fourier transforms, 103  
 inverse Fourier transform, 104  
 using numerical techniques, 104  
 parameters, 93  
 rectangular window, 103  
 SCER performance, 195  
 synthesized window functions, 102  
 window family, 149–150
- Raised-cosine window, *see* Hann  
 window
- Range side lobe reduction effect, 306  
 FM pulse compression radar,  
 307, 308  
 loss factor, 309–314  
 results, 316–317  
 on SNR, 306  
 uniform weighting function, 309  
 weighting function, 314–316
- Range-Doppler technique, 351
- Rate of fall-off of side-lobe level  
 (RFSLL), 93, 151, 170  
 decomposition, 120  
 Fourier transform, 118, 119  
 in Hann–Poisson window, 154  
 for rectangular window, 170  
 theorem, 119, 120–121  
 in time-domain, 121–123  
 window function decomposition, 120
- Rational transforms, 33
- RCS, *see* Radar cross section (RCS)
- Read-only memory (ROM), 189
- Reconstructed signal, 60  
 Fourier transform of, 69  
 frequency response, 66  
 from LPC coefficients, 344

- Reconstruction, 59
  - characteristics, 68
  - continuous-time signals, 67, 69
  - cutoff frequency, 69
  - impulse train-modulated signal, 69
  - integration and summation, 68
- Rectangular window
  - CTFT of, 89
  - data window, 70, 147
  - Dirichlet window, 148
  - in FIR filter design, 174, 175
  - half main-lobe width, 90
  - log magnitude response, 180
  - magnitude response of signal, 186
  - resolution characteristics, 166–167
  - RFSLL for, 170
  - 3 dB bandwidth, 167
  - in time and frequency-domain, 176
  - transition bandwidth and ripples, 175
  - weak component response, 187
- Region of convergence (ROC), 31–32
- Resolution, 129, 260
  - Bartlett periodogram, 280, 282
  - Blackman–Tukey method, 289, 292
  - frequency, 84
  - high resolution, 185
  - HRR, 303
  - ISAR images, 357
  - loss of frequency, 71, 142
  - periodogram estimate, 264
  - PSD estimate, 265
  - radar image, 351
  - random process, 260
  - rectangular window, 166, 167
  - spectrum estimate, 129, 167
  - speech applications, 295
- RFSH, *see* Rate of fall-off side-lobe level (RFSLL)
- RFSLL, *see* Rate of fall-off of side-lobe level (RFSLL)
- Riemann window, 153
- ROC, *see* Region of convergence (ROC)
- ROM, *see* Read-only memory (ROM)
- Root mean square (RMS), 160
- S**
- Sampling, 60
  - discrete-time signal, 60–61
  - linear relationship, 61
  - normalized frequency, 61, 62
  - Nyquist–shannon sampling theorem, 64, 65
  - relationship, 65–67
  - sampled data, 62
  - signal reconstructions, 62
  - WKS sampling theorem, 60, 63, 64
- SAR, *see* Synthetic aperture radar (SAR)
- Saramäki window family, 156–157
- Scaling factor, 8–9, 161
- Scalloping loss, *see* Picket-fence effect
- Scattering centers, 350
  - Doppler spectral analysis, 351
  - ISAR images, 354
- SCER, *see* Signal-to-computational error ratio (SCER)
- Schwartz inequality, 161
- Self-convolution property, 91, 148
- SFW, *see* Stepped frequency waveform (SFW)
- Short-time Fourier transform (STFT), 293, 324
  - using Bartlett window, 296
  - using Blackman window, 295, 296, 298
  - using Gaussian window, 336
  - using Hamming window, 295, 296
  - using Hann window, 295, 296, 299, 335
  - using Kaiser–Bessel window, 296, 297, 298
  - using rectangular window, 295, 296, 334
  - window length, 294
- Side lobes, 71
  - asymptotic attenuation of, 101
  - of Bartlett window, 269
  - decay rate of, 316
  - Dolph–Chebyshev window, 155
  - Doppler, 306
  - first side-lobe and peak, 315
  - Fourier transform, 87, 114
  - FSL, 89
  - in frequency window, 231
  - Hamming window, 182
  - modified first-order Bessel family, 118
  - MSLL, 88
  - peak magnitude, 99, 124
  - PSLL, 132
  - PSF, 352
  - range of, 95



- Side lobes (*Continued*)
  - rate of fall-off, 118, 125
  - rectangular window, 73, 79, 275
  - RFSLL, 87, 155
  - in spectral window, 129
  - weak spectral component, 185
- Side-lobe fall-off rate, 97
  - Blackman window, 206
  - Hann window, 295
  - NHMLW, 113
  - of Parzen window, 114
  - in time-domain, 121–123
- Signal processing
  - biomedical, 324
  - characteristics, 1
  - continuous-time signals, 2, 3
  - discrete-time signals, 3–4
  - FT, 1
  - linear convolution, 48
  - LTI systems, 5–6
  - system and properties, 4, 5
- Signal-to-computational error ratio (SCER), 193
  - Blackman window performance, 194
  - Hamming window performance, 194
  - Hann window performance, 193
  - raised-cosine family performance, 195
- Signal-to-Noise Ratio (SNR), 166, 328
  - of bandpass-filtered ECG signals, 328
  - of lowpass-Filtered EEG signals, 329
  - of MA filtered EEG signal, 332
  - of MA filtered EMG signal, 333
- Simulations
  - Blackman–Harris window, 306
  - range profiles of windows, 305, 307, 308
- Smearing, 189
  - DFT, 189
  - effect, 131
  - of energy, 73
  - filter response, 231
  - side lobes, 181
- Smoothing, 130
  - effects, 182
  - performance, 357
  - periodogram, 265
  - time function, 145
- Smoothing filters, *see* Moving average filter (MA filter)
- SNR, *see* Signal-to-Noise Ratio (SNR)
- Spectral analysis, 159–160; *see also* Finite impulse response filters (FIR filters)
  - autocorrelation function, 180
  - Blackman–Tukey spectral estimation, 181
  - correlogram, 177
  - equal strengths, frequency
    - components of, 182, 183
  - far-away frequency components with unequal strengths, 185, 186
  - periodogram, 177, 178, 178
  - PSD calculation, 177
  - spectral estimation, 260
  - unequal strengths, frequency
    - components of, 182, 184–185
  - weak component in signal
    - component, 185, 187
  - Welch* method, 178
  - Wiener–Khinchin theorem, 180, 181
  - window selection, 181–182
  - windows application in, 259
- Spectral estimator
  - Blackman–Tukey, 288
  - Hamming window, 300
  - Hann window, 299
  - Zeroth-order Kaiser–Bessel window, 300
  - Zeroth-order Kaiser window, 299
- Spectral leakage, 74
  - high side-lobe levels, 295
  - impulse functions, 160
  - low spectral leakage, 185
  - minimum, 343
  - noncoherent component, 169
  - reduction of, 159
  - weighting functions, 129
- Spectral response, 131
- Spectral window, 129
  - convolution operation, 288
  - data window, 147
  - energy of, 181
  - Fourier transform, 111, 146
  - side lobes, 129
- Spectrogram, 295, 339
  - using Blackman window, 298
  - using Hamming window, 297

- using Kaiser–Bessel window, 296
    - TF plane, 343
  - Speech analysis, 347
  - Square summability, 19
  - Stability, 129
    - guaranteed FIR, 225
    - IIR filters, 226
    - property, 5
  - Stein unbiased risk estimate (SURE), 342
  - Stepped frequency waveform (SFW), 303
  - STFT, *see* Short-time Fourier transform (STFT)
  - Stopband
    - cut-off frequency, 227
    - deviation, 227
    - FIR lowpass filter, 178
    - passband and, 175
    - passband ripple, 250
    - PSLL, 233
  - Stretch processing
    - block diagram of, 317
    - echo signals compression, 319, 320, 321, 322
    - specifications, 318
    - window functions in, 317
  - Summability, absolute, 19
  - Sum-cosine window, 96, 97–98
    - asymptotic attenuation, 101
    - Bessel window, 125
    - $\cos^3x$  window, 149
    - Fourier transforms, 98, 99, 101
    - linear combination, 98
    - normalized time and frequency parameters, 100
    - parameters, 93
    - side-lobe ripples, 99
    - synthesis problem, 99–100
    - window functions, 102
  - Summability, *see* Convergence
  - SURE, *see* Stein unbiased risk estimate (SURE)
  - Symmetry properties
    - DFT, 38–39, 50
    - DTFT, 22
    - FFT, 50
  - Synthesis formula, *see* Inverse DTFT (IDTFT)
  - Synthesized sum-cosine windows, 125
  - Synthetic aperture radar (SAR), 349, 358
- ## T
- Tapered cosine window, *see* Tukey window
  - Tapped-delay line structure, 227
  - Taylor functions, 94
  - TDI, *see* Time-domain implementation (TDI)
  - Ternary code, 195
  - TF plane, *see* Time–frequency plane (TF plane)
  - 3 dB bandwidth, *see* Half-power bandwidth
  - Time reversal property, 20, 38
  - Time scaling property, 8
  - Time shifting property, 7–8, 20
  - Time-domain function, 88
  - Time-domain implementation (TDI), 189, 190, 192
    - arithmetic operations, 194
    - computational error in, 192
    - input data samples, 192
    - quantization errors, 195
    - quantized data samples, 192–193
    - SCER, 193, 194
    - scheme, 190
    - windows, 192
  - Time–bandwidth product, 171–172
  - Time–frequency plane (TF plane), 336
    - block thresholding technique, 340
    - diagonal estimators of SNR, 341
    - Fourier atom, 339
    - reconstructions, 340
  - Total energy, 131
    - half-power bandwidth, 132
    - optimum window function, 115
    - parameters, 123
  - Transforms, 1
    - duality property, 9
    - rectangular and cosine windows, 98
    - sum-cosine window, 100
    - z-transform, 33
  - Transversal filter, 227
  - Triangular window, *see* Bartlett windows

- Truncated Taylor family, 94, 95
  - dispersion factor for, 138
  - Hamming window, 149
  - parameters, 93, 130–131
  - performance comparison, 134
  - of windows, 149
- Truncation, 71
  - DFT, 59
  - Fourier series, 173
  - impulse response, 174, 232
  - input time sequence, 189
  - rectangular window, 89
- Tukey window, 111, 150
  - ECG signal and QRS locations, 337
  - Fourier transform, 112
  - frequency-domain plots, 113
  - functions, 113
  - noisy ECG signal, 326
  - noisy EEG signal, 330
  - parameters, 93
- U**
- Ultraspherical window, 157, 158, 361
  - LSF variation, 348, 350
  - magnitude response, 158
  - parameters for, 357
- Unbiased estimator, 259, 282
- Uniform sampling, *see* Periodic sampling
- Uniform window, *see* Rectangular window
- V**
- Variable window, 118, 126, 243
- Variance, 259
  - Bartlett periodogram, 279
  - Blackman–Tukey method, 289
  - lag window, 146
  - LTI systems, 5–6
  - narrow spectral window, 142
  - SCER, 193
  - Welch periodogram, 282–283
- Variance compensation factor, 130–131
  - DFT, 171
  - expression for, 133
  - window selection, 139, 142
- Vibroarthogram (VAG), 323
- Vibromyogram (VMG), 323
- Vocal tract models, 346
- Von Hann window, *see* Hann window
- W**
- WCPL, *see* Worst-case processing loss (WCPL)
- Weighting function
  - data, 316
  - Dolph–Chebyshev window, 114
  - factors, 315
  - frequency window, 145
  - loss factor for, 312
  - uniform spacing, 314
  - windows, 129
- Welch method
  - periodogram, 178
  - resolution, 283
- Welch periodogram method
  - with Bartlett window, 286
  - bias of, 281
  - with Blackman window, 287
  - expression for, 280
  - with Hamming window, 286
  - with Hann window, 287
  - with Kaiser–Bessel window, 288
  - with overlapping, 284, 285
  - resolution of, 283
  - variance of, 282, 283
  - window dependent, 283
- Whittaker Kotelnikov Shannon sampling theorem (WKS sampling theorem), 60
  - C/D converter, 63
  - modulator output, 63
  - sampling frequency, 64
  - two-stage process, 63
- Wiener–Khinchin theorem, 180, 181
- Window functions, 87
  - bandwidths, 3 dB and 6 dB, 166–167
  - Blackman window, 106–107
  - Blackman–Harris window, 109, 110
  - characteristics, 87–88
  - coherent gain, 161, 166
  - comparison, 123–126
  - Cos( $x$ ) window, 91, 92
  - Cos<sup>3</sup>( $x$ ) window, 96, 97
  - Cos<sup>4</sup>( $x$ ) window, 101, 102
  - data windows, 145, 146, 147

- degradation loss, 166
  - Dolph–Chebyshev window, 114
  - ENBW, 160–161
  - frequency window, 145, 146, 147
  - Hamming window, 95, 96
  - Hann window, 91, 92, 93, 94
  - interplay between window functions, 146, 147
  - Kaiser’s modified first-order Bessel window function family, 116, 117, 118
  - Kaiser’s modified zeroth-order Bessel window function family, 115, 116
  - lag window, 146, 147
  - list of windows, 88
  - main-lobe width, 168
  - optimized Blackman window, 107, 108–109
  - overlap correlation, 169
  - Papoulis window, 111
  - parabolic window, 110–111
  - Parzen window, 113, 114
  - performance comparison of windows, 162–165
  - picket-fence effect, 168, 169
  - PSLL and FSLL, 169, 170
  - raised-cosine family, 102–103, 104, 105, 106
  - rectangular window, 89, 90
  - RFSLL, 118, 119, 121–123, 170
  - scalloping loss, 168
  - spectral analysis, 159–160
  - spectral window, 146, 147
  - sum-cosine window, 97–98, 99–100, 101
  - triangular window, 89–90, 91
  - truncated Taylor family, 94, 95
  - Tukey window, 111, 112, 113
  - variance compensation factor, 171
  - WCPL, 169
  - Window parameters, 130
    - coherent gain, 131
    - computation, 132–139
    - dispersion factor, 131
    - half-power bandwidth, 131–132
    - MLE content, 131
    - normalized half main-lobe width, 132
    - normalized half-power bandwidth, 132
    - PSLL or MSLL, 132
    - total energy, 131
    - variance compensation factor, 130–131
  - Window selection, 139–142
    - almost equal strengths, 182
    - for spectral analysis, 181–182
    - unequal strengths, 182, 185
    - weak component, 185
  - Window shape parameter, 274, 275
  - Windows, 129
    - classes, 145, 146, 147
    - CSD window structures, 197
    - Dolph–Chebyshev, 357
    - filter characteristics, 233
    - in HRR radars, 303–306
    - list, 88
    - parameters of data, 93
    - pedestal heights, 348
    - performance comparison, 134–135, 141
    - performance of binary, 216
    - picket-fence effects, 84
    - raised-cosine set, 102
    - synthesized sum-cosine, 125
  - Windowing method, *see* Impulse response truncation method (IRT method)
  - Windowing or modulation property, 21
  - WKS sampling theorem, *see* Whittaker Kotelnikov Shannon sampling theorem (WKS sampling theorem)
  - Worst-case processing loss (WCPL), 169
- Z**
- z-plane, 31–32
    - position, 33, 34
    - z-transform pairs, 33
  - Z-transform, 1
    - advantage, 31
    - basic z-transform pairs, 33
    - examples, 34–36
    - LCCDE, 33, 34
    - properties, 34
    - rational transforms, 33
    - ROC, 31–32

- Z-transform, 1 (*Continued*)
  - system function, 222–223
  - transform-domain analysis, 30–31
- Zero leakage case, 71–72
  - DFT output response, 73
  - frequency-domain, 73
  - geometric summation, 72
  - in time-domain, 352
- Zero-padding
  - filling sequence, 80
  - interpolation formula, 83, 84
- Zero-padding
  - ISAR images, 354
  - linear convolution implementation, 80–82
  - spectrum display, 82–83
- Zooming filtering, 224