

# DISCRETE-TIME SIGNAL PROCESSING



Alan V. Oppenheim · Ronald W. Schaffer

PRENTICE HALL SIGNAL PROCESSING SERIES  
ALAN V. OPPENHEIM, SERIES EDITOR



Oppenheim  
Schafer



# DISCRETE-TIME SIGNAL PROCESSING

QA402  
.O 615  
1989

# **Discrete-Time Signal Processing**

## PRENTICE HALL SIGNAL PROCESSING SERIES

*Alan V. Oppenheim, Editor*

ANDREWS AND HUNT *Digital Image Restoration*  
BRIGHAM *The Fast Fourier Transform*  
BRIGHAM *The Fast Fourier Transform and Its Applications*  
BURDIC *Underwater Acoustic System Analysis*  
CASTLEMAN *Digital Image Processing*  
COWAN AND GRANT *Adaptive Filters*  
CROCHIERE AND RABINER *Multirate Digital Signal Processing*  
DUDGEON AND MERSEREAU *Multidimensional Digital Signal Processing*  
HAMMING *Digital Filters, 3/E*  
HAYKIN, ED. *Array Signal Processing*  
JAYANT AND NOLL *Digital Coding of Waveforms*  
KAY *Modern Spectral Estimation*  
KINO *Acoustic Waves: Devices, Imaging, and Analog Signal Processing*  
LEA, ED. *Trends in Speech Recognition*  
LIM *Two-Dimensional Signal and Image Processing*  
LIM, ED. *Speech Enhancement*  
LIM AND OPPENHEIM, EDS. *Advanced Topics in Signal Processing*  
MARPLE *Digital Spectral Analysis with Applications*  
MCCLELLAN AND RADER *Number Theory in Digital Signal Processing*  
MENDEL *Lessons in Digital Estimation Theory*  
OPPENHEIM, ED. *Applications of Digital Signal Processing*  
OPPENHEIM, WILLSKY, WITH YOUNG *Signals and Systems*  
OPPENHEIM AND SCHAFER *Digital Signal Processing*  
OPPENHEIM AND SCHAFER *Discrete-Time Signal Processing*  
QUACKENBUSH ET AL. *Objective Measures of Speech Quality*  
RABINER AND GOLD *Theory and Applications of Digital Signal Processing*  
RABINER AND SCHAFER *Digital Processing of Speech Signals*  
ROBINSON AND TREITEL *Geophysical Signal Analysis*  
STEARNS AND DAVID *Signal Processing Algorithms*  
TRIBOLET *Seismic Applications of Homomorphic Signal Processing*  
WIDROW AND STEARNS *Adaptive Signal Processing*

# **Discrete-Time Signal Processing**

**Alan V. Oppenheim  
Ronald W. Schafer**



PRENTICE HALL, Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Oppenheim, Alan V.

Discrete-time signal processing / Alan V. Oppenheim, Ronald W. Schafer.

p. cm.—(Prentice Hall signal processing series)

Bibliography: p.

Includes index.

ISBN 0-13-216292-X

1. Signal processing—Mathematics. 2. Discrete-time systems.

I. Schafer, Ronald W. II. Title. III. Series.

TK5102.5.02452 1989

621.38'043—dc 19

88-25562

CIP

Editorial/production supervision: Barbara G. Flanagan

Interior design: Roger Brower

Cover design: Vivian Berman

Manufacturing buyer: Mary Noonan



© 1989 Alan V. Oppenheim, Ronald W. Schafer

Published by Prentice-Hall, Inc.

A Division of Simon & Schuster

Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7

ISBN 0-13-216292-X

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

# Contents

---

**Preface**      **xi**

**1 Introduction**      **1**

**2 Discrete-Time Signals and Systems**      **8**

- 2.0 Introduction 8
- 2.1 Discrete-Time Signals: Sequences 9
- 2.2 Discrete-Time Systems 17
- 2.3 Linear Time-Invariant Systems 21
- 2.4 Properties of Linear Time-Invariant Systems 27
- 2.5 Linear Constant-Coefficient Difference Equations 33
- 2.6 Frequency-Domain Representation of Discrete-Time  
Signals and Systems 39
- 2.7 Representation of Sequences by Fourier Transforms 45
- 2.8 Symmetry Properties of the Fourier Transform 52
- 2.9 Fourier Transform Theorems 56
- 2.10 Discrete-Time Random Signals 63
- 2.11 Summary 67  
Problems 68

**3 Sampling of Continuous-Time Signals**      **80**

- 3.0 Introduction 80
- 3.1 Periodic Sampling 80
- 3.2 Frequency-Domain Representation of Sampling 82
- 3.3 Reconstruction of a Bandlimited Signal from Its Samples 87
- 3.4 Discrete-Time Processing of Continuous-Time Signals 91
- 3.5 Continuous-Time Processing of Discrete-Time Signals 99
- 3.6 Changing the Sampling Rate Using Discrete-Time  
Processing 101
- 3.7 Practical Considerations 112
- 3.8 Summary 130  
Problems 131

**4 The z-Transform**      **149**

- 4.0 Introduction 149
- 4.1 The z-Transform 149
- 4.2 Properties of the Region of Convergence for the  
z-Transform 160

|     |                                                         |     |
|-----|---------------------------------------------------------|-----|
| 4.3 | The Inverse $z$ -Transform                              | 165 |
| 4.4 | $z$ -Transform Properties                               | 172 |
| 4.5 | The Inverse $z$ -Transform Using Contour<br>Integration | 181 |
| 4.6 | The Complex Convolution Theorem                         | 184 |
| 4.7 | Parseval's Relation                                     | 186 |
| 4.8 | The Unilateral $z$ -Transform                           | 188 |
| 4.9 | Summary                                                 | 191 |
|     | Problems                                                | 192 |

## **5 Transform Analysis of Linear Time-Invariant Systems 202**

|     |                                                                                                   |     |
|-----|---------------------------------------------------------------------------------------------------|-----|
| 5.0 | Introduction                                                                                      | 202 |
| 5.1 | The Frequency Response of LTI Systems                                                             | 203 |
| 5.2 | System Functions for Systems Characterized by Linear<br>Constant-Coefficient Difference Equations | 206 |
| 5.3 | Frequency Response for Rational System Functions                                                  | 213 |
| 5.4 | Relationship Between Magnitude and Phase                                                          | 230 |
| 5.5 | Allpass Systems                                                                                   | 234 |
| 5.6 | Minimum-Phase Systems                                                                             | 240 |
| 5.7 | Linear Systems with Generalized Linear Phase                                                      | 250 |
| 5.8 | Summary                                                                                           | 270 |
|     | Problems                                                                                          | 270 |

## **6 Structures for Discrete-Time Systems 290**

|      |                                                                                         |     |
|------|-----------------------------------------------------------------------------------------|-----|
| 6.0  | Introduction                                                                            | 290 |
| 6.1  | Block Diagram Representation of Linear<br>Constant-Coefficient Difference Equations     | 291 |
| 6.2  | Signal Flow Graph Representation of Linear<br>Constant-Coefficient Difference Equations | 297 |
| 6.3  | Basic Structures for IIR Systems                                                        | 300 |
| 6.4  | Transposed Forms                                                                        | 309 |
| 6.5  | Basic Network Structures for FIR Systems                                                | 313 |
| 6.6  | Lattice Structures                                                                      | 317 |
| 6.7  | Overview of Finite-Precision Numerical Effects                                          | 328 |
| 6.8  | The Effects of Coefficient Quantization                                                 | 335 |
| 6.9  | Effects of Roundoff Noise in Digital Filters                                            | 351 |
| 6.10 | Zero-Input Limit Cycles in Fixed-Point Realizations<br>of IIR Digital Filters           | 373 |
| 6.11 | Summary                                                                                 | 378 |
|      | Problems                                                                                | 379 |

## **7 Filter Design Techniques 403**

|     |                                                                      |     |
|-----|----------------------------------------------------------------------|-----|
| 7.0 | Introduction                                                         | 403 |
| 7.1 | Design of Discrete-Time IIR Filters from Continuous-<br>Time Filters | 406 |
| 7.2 | Frequency Transformations of Lowpass IIR Filters                     | 430 |
| 7.3 | Computer-Aided Design of Discrete-Time IIR Filters                   | 438 |
| 7.4 | Design of FIR Filters by Windowing                                   | 444 |

- 7.5 Examples of FIR Filter Design by the Kaiser Window Method 458
- 7.6 Optimum Approximations of FIR Filters 464
- 7.7 Examples of FIR Equiripple Approximation 481
- 7.8 Comments on IIR and FIR Digital Filters 488
- 7.9 Summary 489
- Problems 490

## **8 The Discrete Fourier Transform 514**

- 8.0 Introduction 514
- 8.1 Representation of Periodic Sequences: The Discrete Fourier Series 515
- 8.2 Properties of the Discrete Fourier Series 520
- 8.3 Summary of Properties of the DFS Representation of Periodic Sequences 525
- 8.4 The Fourier Transform of Periodic Signals 526
- 8.5 Sampling the Fourier Transform 527
- 8.6 Fourier Representation of Finite-Duration Sequences: The Discrete Fourier Transform 530
- 8.7 Properties of the Discrete Fourier Transform 535
- 8.8 Summary of Properties of the Discrete Fourier Transform 547
- 8.9 Linear Convolution Using the Discrete Fourier Transform 548
- 8.10 Summary 560
- Problems 561

## **9 Computation of the Discrete Fourier Transform 581**

- 9.0 Introduction 581
- 9.1 Efficient Computation of the Discrete Fourier Transform 582
- 9.2 The Goertzel Algorithm 585
- 9.3 Decimation-in-Time FFT Algorithms 587
- 9.4 Decimation-in-Frequency FFT Algorithms 599
- 9.5 Implementation of FFT Algorithms 605
- 9.6 FFT Algorithms for Composite  $N$  610
- 9.7 Implementation of the DFT Using Convolution 622
- 9.8 Effects of Finite Register Length in Discrete Fourier Transform Computations 628
- 9.9 Summary 641
- Problems 642

## **10 Discrete Hilbert Transforms 662**

- 10.0 Introduction 662
- 10.1 Real and Imaginary Part Sufficiency of the Fourier Transform for Causal Sequences 664
- 10.2 Sufficiency Theorems for Finite-Length Sequences 670
- 10.3 Relationships Between Magnitude and Phase 674
- 10.4 Hilbert Transform Relations for Complex Sequences 676

|      |          |     |
|------|----------|-----|
| 10.5 | Summary  | 689 |
|      | Problems | 689 |

## **11 Fourier Analysis of Signals Using the Discrete Fourier Transform 695**

|      |                                                                                     |     |
|------|-------------------------------------------------------------------------------------|-----|
| 11.0 | Introduction                                                                        | 695 |
| 11.1 | Fourier Analysis of Signals Using the DFT                                           | 696 |
| 11.2 | DFT Analysis of Sinusoidal Signals                                                  | 699 |
| 11.3 | The Time-Dependent Fourier Transform                                                | 713 |
| 11.4 | Block Convolution Using the Time-Dependent Fourier Transform                        | 721 |
| 11.5 | Fourier Analysis of Nonstationary Signals                                           | 723 |
| 11.6 | Fourier Analysis of Stationary Random Signals: The Periodogram                      | 730 |
| 11.7 | Spectrum Analysis of Random Signals Using Estimates of the Autocorrelation Sequence | 742 |
| 11.8 | Summary                                                                             | 755 |
|      | Problems                                                                            | 756 |

## **12 Cepstrum Analysis and Homomorphic Deconvolution 768**

|       |                                                        |     |
|-------|--------------------------------------------------------|-----|
| 12.0  | Introduction                                           | 768 |
| 12.1  | Definition of the Complex Cepstrum                     | 769 |
| 12.2  | Homomorphic Deconvolution                              | 771 |
| 12.3  | Properties of the Complex Logarithm                    | 775 |
| 12.4  | Alternative Expressions for the Complex Cepstrum       | 778 |
| 12.5  | The Complex Cepstrum of Exponential Sequences          | 779 |
| 12.6  | Minimum-Phase and Maximum-Phase Sequences              | 781 |
| 12.7  | Realizations of the Characteristic System $D_*[\cdot]$ | 787 |
| 12.8  | Examples of Homomorphic Filtering                      | 797 |
| 12.9  | Applications to Speech Processing                      | 815 |
| 12.10 | Summary                                                | 825 |
|       | Problems                                               | 826 |

## **Appendix A Random Signals 835**

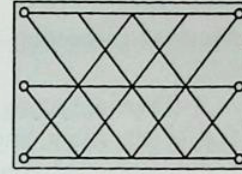
|     |                                                    |     |
|-----|----------------------------------------------------|-----|
| A.1 | Discrete-Time Random Processes                     | 835 |
| A.2 | Averages                                           | 837 |
| A.3 | Properties of Correlation and Covariance Sequences | 841 |
| A.4 | Transform Representations of Random Signals        | 843 |

## **Appendix B Continuous-Time Filters 845**

|     |                             |     |
|-----|-----------------------------|-----|
| B.1 | Butterworth Lowpass Filters | 845 |
| B.2 | Chebyshev Filters           | 847 |
| B.3 | Elliptic Filters            | 849 |

## **Bibliography 851**

## **Index 869**



# Computation of the Discrete Fourier Transform

---

## 9

---

### 9.0 INTRODUCTION

The discrete Fourier transform (DFT) plays an important role in the analysis, design, and implementation of discrete-time signal processing algorithms and systems. The basic properties of the Fourier transform and discrete Fourier transform, discussed in Chapters 2 and 8, make it particularly convenient to analyze and design systems in the Fourier domain. Equally important is the fact that efficient algorithms exist for explicitly computing the DFT. As a result, the DFT is an important component in many practical applications of discrete-time systems.

As discussed in Chapter 8, the DFT is identical to samples of the Fourier transform at equally spaced frequencies. Consequently, computation of the  $N$ -point DFT corresponds to the computation of  $N$  samples of the Fourier transform at  $N$  equally spaced frequencies,  $\omega_k = 2\pi k/N$ , i.e., at  $N$  points on the unit circle in the  $z$ -plane. In this chapter, we discuss several methods for computing values of the DFT. The major focus of this chapter is a particularly efficient class of algorithms for the digital computation of the  $N$ -point DFT. Collectively, these efficient algorithms are called *fast Fourier transform* (FFT) algorithms, and we discuss them in Sections 9.3–9.6. To achieve the highest efficiency, the FFT algorithms must compute all  $N$  values of the DFT. When we require values of the DFT over only a portion of the frequency range  $0 \leq \omega < 2\pi$ , other algorithms may be more efficient and flexible even though they are less efficient than the FFT algorithms for computation of all the values of the DFT. Examples of such algorithms are the Goertzel algorithm,

discussed in Section 9.2, and the chirp transform algorithm, discussed in Section 9.7.2.

There are many ways to measure complexity and efficiency of an implementation or algorithm, and a final assessment depends on both the available technology and the intended application. We will use the number of arithmetic multiplications and additions as a measure of computational complexity. This measure is simple to apply, and the number of multiplications and additions is directly related to computational speed when algorithms are implemented on general-purpose digital computers or special-purpose microprocessors. However, other measures are sometimes more appropriate. For example, in custom VLSI implementations, chip area is often the most important consideration, and chip area may not be directly related to the number of arithmetic operations.

In terms of multiplications and additions, the class of FFT algorithms can be orders of magnitude more efficient than competing algorithms. The efficiency of FFT algorithms is so high, in fact, that in many cases the most efficient procedure for implementing a convolution is to compute the transform of the sequences to be convolved, multiply their transforms, and then compute the inverse transform of the product of transforms. The details of this technique were discussed in Section 8.9. In seeming contradiction to this, a set of algorithms (mentioned briefly in Section 9.7) for evaluation of the DFT (or a more general set of samples of the Fourier transform) derive their efficiency from a reformulation of the Fourier transform in terms of a convolution and thereby implement the Fourier transform computation by using efficient procedures for evaluation of the associated convolution. This suggests the possibility of implementing a convolution by multiplication of DFTs, where the DFTs have been implemented by first expressing them as convolutions and then taking advantage of efficient procedures for implementing the associated convolutions. While this seems on the surface to be a basic contradiction, we will see in Section 9.7 that in certain cases it is an entirely reasonable approach and is not at all a contradiction.

## 9.1 EFFICIENT COMPUTATION OF THE DISCRETE FOURIER TRANSFORM

As defined in Chapter 8, the DFT of a finite-length sequence of length  $N$  is

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \dots, N-1, \quad (9.1)$$

where  $W_N = e^{-j(2\pi/N)}$ . The inverse discrete Fourier transform is given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad n = 0, 1, \dots, N-1. \quad (9.2)$$

In Eqs. (9.1) and (9.2), both  $x[n]$  and  $X[k]$  may be complex. Since the expressions of Eqs. (9.1) and (9.2) differ only in the sign of the exponent of  $W_N$  and in the scale factor

$1/N$ , a discussion of computational procedures for Eq. (9.1) applies with straightforward modifications to Eq. (9.2). (See Problem 9.1.)

To create a frame of reference for our discussion of computation of the DFT, let us first consider direct evaluation of the DFT equation in Eq. (9.1). Since  $x[n]$  may be complex,  $N$  complex multiplications and  $(N - 1)$  complex additions are required to compute each value of the DFT if we use Eq. (9.1) directly as a formula for computation. To compute all  $N$  values therefore requires a total of  $N^2$  complex multiplications and  $N(N - 1)$  complex additions. Expressing Eq. (9.1) in terms of operations on real numbers, we obtain

$$X[k] = \sum_{n=0}^{N-1} [(\mathcal{R}e\{x[n]\}\mathcal{R}e\{W_N^{kn}\} - \mathcal{I}m\{x[n]\}\mathcal{I}m\{W_N^{kn}\}) + j(\mathcal{R}e\{x[n]\}\mathcal{I}m\{W_N^{kn}\} + \mathcal{I}m\{x[n]\}\mathcal{R}e\{W_N^{kn}\})], \quad (9.3)$$

$$k = 0, 1, \dots, N - 1,$$

which shows that each complex multiplication requires four real multiplications and two real additions, and each complex addition requires two real additions. Therefore, for each value of  $k$ , the direct computation of  $X[k]$  requires  $4N$  real multiplications and  $(4N - 2)$  real additions.† Since  $X[k]$  must be computed for  $N$  different values of  $k$ , the direct computation of the discrete Fourier transform of a sequence  $x[n]$  requires  $4N^2$  real multiplications and  $N(4N - 2)$  real additions. Besides the multiplications and additions called for by Eq. (9.3), the digital computation of the DFT on a general-purpose digital computer or with special-purpose hardware also requires provision for storing and accessing the  $N$  complex input sequence values  $x[n]$  and values of the complex coefficients  $W_N^{kn}$ . Since the amount of computation, and thus the computation time, is approximately proportional to  $N^2$ , it is evident that the number of arithmetic operations required to compute the DFT by the direct method becomes very large for large values of  $N$ . For this reason, we are interested in computational procedures that reduce the number of multiplications and additions.

Most approaches to improving the efficiency of the computation of the DFT exploit the symmetry and periodicity properties of  $W_N^{kn}$ ; specifically,

1.  $W_N^{k[N-n]} = W_N^{-kn} = (W_N^{kn})^*$  complex conjugate symmetry;
2.  $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$  periodicity in  $n$  and  $k$ .

As an illustration, using the first property, i.e., the symmetry of the implicit cosine and sine functions, we can group terms in the summation in Eq. (9.3) for  $n$  and  $(N - n)$ . For example,

$$\begin{aligned} \mathcal{R}e\{x[n]\}\mathcal{R}e\{W_N^{kn}\} + \mathcal{R}e\{x[N-n]\}\mathcal{R}e\{W_N^{k[N-n]}\} \\ = (\mathcal{R}e\{x[n]\} + \mathcal{R}e\{x[N-n]\})\mathcal{R}e\{W_N^{kn}\} \end{aligned}$$

† Throughout the discussion, the figure for the number of computations is only approximate. Multiplication by  $W_N^0$ , for example, does not in fact require a multiplication. Nevertheless, the estimate of computational complexity obtained by including such multiplications is sufficiently accurate to permit comparisons between different classes of algorithms.

and

$$\begin{aligned} & - \mathcal{I}\{x[n]\} \mathcal{I}\{W_N^{kn}\} - \mathcal{I}\{x[N-n]\} \mathcal{I}\{W_N^{k[N-n]}\} \\ & = - (\mathcal{I}\{x[n]\} - \mathcal{I}\{x[N-n]\}) \mathcal{I}\{W_N^{kn}\} \end{aligned}$$

Similar groupings can be used for the other terms in Eq. (9.3). In this way, the number of multiplications can be reduced by approximately a factor of 2. We can also take advantage of the fact that for certain values of the product  $kn$ , the sine and cosine functions take on the value 1 or 0, thereby eliminating the need for multiplications. However, reductions of this type still leave us with an amount of computation that is proportional to  $N^2$ . Fortunately, the second property, the periodicity of the complex sequence  $W_N^{kn}$ , can be exploited in achieving significantly greater reductions of the computation.

Computational algorithms that exploit both the symmetry and the periodicity of the sequence  $W_N^{kn}$  were known long before the era of high-speed digital computation. At that time, any scheme that reduced manual computation by even a factor of 2 was welcomed. Heideman et al. (1984) have traced the origins of the FFT back to Gauss as early as 1805. Runge (1905) and later Danielson and Lanczos (1942) described algorithms for which computation was roughly proportional to  $N \log N$  rather than  $N^2$ .† However, the distinction was not of great importance for the small values of  $N$  that were feasible for hand computation. The possibility of greatly reduced computation was generally overlooked until about 1965, when Cooley and Tukey (1965) published an algorithm for the computation of the discrete Fourier transform that is applicable when  $N$  is a composite number, i.e.,  $N$  is the product of two or more integers. The publication of their paper touched off a flurry of activity in the application of the discrete Fourier transform to signal processing and resulted in the discovery of a number of highly efficient computational algorithms. Collectively, the entire set of such algorithms has come to be known as the *fast Fourier transform*, or the FFT.

FFT algorithms are based on the fundamental principle of decomposing the computation of the discrete Fourier transform of a sequence of length  $N$  into successively smaller discrete Fourier transforms. The manner in which this principle is implemented leads to a variety of different algorithms, all with comparable improvements in computational speed. In this chapter we are concerned with two basic classes of FFT algorithms. The first, called decimation-in-time, derives its name from the fact that in the process of arranging the computation into smaller transformations, the sequence  $x[n]$  (generally thought of as a time sequence) is decomposed into successively smaller subsequences. In the second general class of algorithms, the sequence of discrete Fourier transform coefficients  $X[k]$  is decomposed into smaller subsequences; hence its name, decimation-in-frequency.

In this chapter we consider a number of algorithms for computing the discrete Fourier transform. The algorithms vary in efficiency, but all of them require fewer multiplications and additions than direct evaluation of Eq. (9.3). We begin in the next section with a discussion of Goertzel's algorithm (Goertzel, 1958), which requires

† See Cooley et al. (1967) for a historical summary of results related to the FFT.

computation proportional to  $N^2$  but with a smaller constant of proportionality than the direct method. One of the principal advantages of Goertzel's algorithm is that it is not restricted to computation of the DFT but is in fact equally valid for the computation of any desired set of frequency samples.

In Sections 9.3–9.6 we present a detailed discussion of FFT algorithms for which computation is proportional to  $N \log N$ . This class of algorithms is considerably more efficient in terms of arithmetic operations than the Goertzel algorithm but is specifically oriented toward computation of *all* the values of the DFT. We do not attempt to be exhaustive in our coverage of this class of algorithms, but we illustrate the general principles common to all algorithms of this type by considering in detail only a few of the more commonly used schemes.

In Section 9.7 we discuss algorithms that rely on formulating the computation of the DFT in terms of a convolution. Finally, in Section 9.8 we consider the effects of arithmetic quantization in DFT computations.

## 9.2 THE GOERTZEL ALGORITHM

The Goertzel algorithm (Goertzel, 1958) is an example of how the periodicity of the sequence  $W_N^{kn}$  can be used to reduce computation. To derive the algorithm, we begin by noting that

$$W_N^{-kN} = e^{j(2\pi/N)Nk} = e^{j2\pi k} = 1. \quad (9.4)$$

This is, of course, a direct result of the periodicity of  $W_N^{kn}$ . Because of Eq. (9.4) we may multiply the right side of Eq. (9.1) by  $W_N^{-kN}$  without affecting the equation. Thus,

$$X[k] = W_N^{-kN} \sum_{r=0}^{N-1} x[r] W_N^{kr} = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)}. \quad (9.5)$$

To suggest the final result, let us define the sequence

$$y_k[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(n-r)} u[n-r]. \quad (9.6)$$

From Eqs. (9.5) and (9.6) and the fact that  $x[n] = 0$  for  $n < 0$  and  $n \geq N$ , it follows that

$$X[k] = y_k[n] \Big|_{n=N}. \quad (9.7)$$

Equation (9.6) can be interpreted as a discrete convolution of the finite-duration sequence  $x[n]$ ,  $0 \leq n \leq N-1$ , with the sequence  $W_N^{-kn} u[n]$ . Consequently,  $y_k[n]$  can be viewed as the response of a system with impulse response  $W_N^{-kn} u[n]$  to a finite-length input  $x[n]$ . In particular,  $X[k]$  is the value of the output when  $n = N$ .

A system with impulse response  $W_N^{-kn} u[n]$  is depicted in Fig. 9.1, where initial rest conditions are assumed. Since the general input  $x[n]$  and the coefficient  $W_N^{-k}$  are both complex, the computation of each new value of  $y_k[n]$  using the system of Fig. 9.1

# DISCRETE-TIME SIGNAL PROCESSING

Alan V. Oppenheim · Ronald W. Schaffer

Discrete-time signal processing continues to be a dynamic and rapidly growing field with a wide range of applications including speech and data communication, acoustics, radar, sonar, seismology, remote sensing, instrumentation, consumer electronics, and many others. *Discrete-Time Signal Processing*, by the authors of the classic text *Digital Signal Processing* (Prentice Hall, 1975), is a completely up-to-date, thorough, and coherent treatment of the fundamentals of this field. Considerable emphasis is placed on illustrative examples and intuitive interpretation. The authors include more than 400 carefully prepared problems to help the reader develop a thorough foundation in the use of this material.

## Contents:

1. Introduction
2. Discrete-Time Signals and Systems
3. Sampling of Continuous-Time Signals
4. The z-Transform
5. Transform Analysis of Linear Time-Invariant Systems
6. Structures for Discrete-Time Systems
7. Filter Design Techniques
8. The Discrete Fourier Transform
9. Computation of the Discrete Fourier Transform
10. Discrete Hilbert Transforms
11. Fourier Analysis of Signals Using the Discrete Fourier Transform
12. Cepstrum Analysis and Homomorphic Deconvolution
  - A. Random Signals
  - B. Analog Filters

PRENTICE HALL, Englewood Cliffs, N.J. 07632

ISBN 0-13-216292-X