Digital Signal Processing: An Overview of Basic Principles

DIGITAL SIGNAL PROCESSING

An Overview of Basic Principles

Jack Cartinhour
Oklahoma State University

Prentice Hall
Upper Saddle River, New Jersey   Columbus, Ohio

Library of Congress Cataloging-in-Publication Data

Cartinhour, Jack.
Digital signal processing: An overview of basic principles / Jack Cartinhour.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-769266-8
1. Signal processing—Digital techniques. 2. Discrete time systems. I. Title.
TK5102.9.C36 2000 98-53218
621.382'2—dc21 CIP

Cover photo: H. Armstrong Roberts
Publisher: Charles E. Stewart, Jr.
Production Editor: Alexandrina Benedicto Wolf
Production Coordination: Custom Editorial Productions, Inc.
Cover Design Coordinator: Karrie Converse-Jones
Cover Designer: Curt Besser
Production Manager: Deidra M. Schwartz
Marketing Manager: Ben Leonard

This book was set in Times by Custom Editorial Productions, Inc., and was printed and bound by R.R. Donnelley & Sons Company. The cover was printed by Phoenix Color Corp.

© 2000 by Prentice-Hall, Inc.
Pearson Education
Upper Saddle River, New Jersey 07458

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 6 5 4 3 2 1

ISBN: 0-13-769266-8

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall of Canada, Inc., *Toronto*
Prentice-Hall Hispanoamericana, S. A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall (Singapore) Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

PREFACE

This book was written to present the basic principles of digital signal processing (DSP) to seniors in electrical engineering technology programs. Seniors in the Department of Electrical Engineering Technology (Division of Engineering Technology) at Oklahoma State University formed the initial audience when a shorter and unpublished version of this book was first used in 1994.

Having taught DSP to electrical engineering students for seven years, I recognize that presenting this material to engineering technology students presents a different sort of challenge. Although the approach in this book is not as rigorous as the approach taken in DSP books for engineering students, it is written at a level that an engineering student would find useful.

This book emphasizes the theoretical aspects of digital signal processing. Although there is no detailed discussion of the specialized microprocessors that are popularly known as DSP chips, real time implementation issues are considered. Most of the emphasis is on the theory of *discrete time signal processing*, which is basically DSP theory minus the consideration of various finite precision effects (e.g., A/D conversion quantization noise and errors introduced by finite precision processor arithmetic). These finite precision effects are important and are dealt with at various sections in the book.

This book explains how the coefficients of the discrete time system difference equation are selected to implement the desired digital filter. It is worth noting that compared with a continuous time system, a discrete time system is much more flexible with respect to changing conditions and/or needs, since changing the system response is only a matter of changing some numbers in a program. This is one of the main reasons why DSP is such a powerful tool.

In studying circuit theory and control systems, electrical engineering technology students are exposed to much of the theory needed to understand *continuous time* (as opposed to *discrete time*) signal processing theory. I assume that the student has been exposed to linear constant coefficient differential equations, the Laplace transform, the Fourier series, and the steady-state response of a linear circuit to a sinusoidal input—and that more advanced topics such as convolution, system impulse response, and the Fourier transform have not been introduced to the student in previous courses. I believe that these advanced topics must be introduced before beginning a study of discrete time signal processing, as they are crucial to the understanding of DSP. The discrete time versions of these advanced topics are better understood if the analogous continuous time concepts are understood. Also, the important relationship between continuous time and discrete time systems is best understood in the Fourier transform domain. Therefore, the first three chapters of this book are devoted to an overview of continuous time system theory, which includes a review of some familiar concepts as well as the introduction of advanced ones.

Digital signal processing requires a way of looking at things that is perhaps more theoretical or mathematically intense than electrical engineering technology students are used to. For example, students can learn much about a continuous time filter circuit without ever considering the differential equation that describes it. The differential equation is one theoretical description of the filter and can be used to gain insight into the behavior of the filter, but this equation is not directly used to implement the filter. The bottom line is that the continuous time filter can be thought of first and foremost as a *circuit*, and its response can be tinkered with by changing things such as resistor and capacitor values. On the other hand, a discrete time system must be thought of first and foremost as a *mathematical algorithm*, since the *difference equation* that describes the system is directly used to implement the system by means of a computer program. To implement this algorithm in real time, a complicated circuit involving a microprocessor or specialized DSP chip, A/D and D/A converters, and so on, must be built; however, one does not tinker with the hardware as such to change the system response. The response of a discrete time system depends on the computer program that implements the system difference equation; the coefficients of this equation determine whether the system is, for example, a digital lowpass filter or a digital highpass filter, and what the cutoff frequency is.

The reader (or instructor) who is anxious to get to the heart of the book—DSP itself—can go directly from Chapter 3 to Chapter 5. However, the material in Chapter 4 on the frequency response and transfer functions of Butterworth and Chebyshev filters should be covered before Chapter 10.

The heart and soul of this book are contained in Chapters 5 through 10. Chapter 5 introduces the concept of a linear, shift-invariant (LSI) discrete time system and the difference equation that describes it. Important topics such as the discrete time system impulse response, discrete time convolution, the Z transform, and the discrete time system transfer function are also presented. Instructors may reduce the amount of time devoted to certain Z transform topics, such as finding the inverse Z transform using partial fraction expansion, which is covered in Section 5–6, but is not needed in the chapters that follow. Chapter 6 covers the discrete time Fourier transform (DTFT) and frequency domain analysis of discrete time signals and systems and discusses the relationship between continuous time and discrete time signal processing. Chapter 7 presents the details of the relationship between the Fourier transform and the DTFT, which is initially presented on an "it can be shown" basis in Chapter 6. The sampling theorem is also presented in Chapter 7, along with a discussion of ideal versus real-world digital-to-analog conversion. In Chapter 8, the student is introduced to the discrete Fourier transform (DFT) and the fast Fourier transform (FFT). Chapter 9 discusses the design of finite impulse response (FIR) filters using the Kaiser window method and the Parks-McClellan algorithm. Chapter 10 discusses the design of infinite impulse response (IIR) filters using the bilinear transformation.

Chapter 11 presents a topic that best illustrates the power of DSP—adaptive FIR filters based on the LMS algorithm. Chapter 12 begins by considering random signals and their associated power spectral density functions, and then shows how *oversampling* can be used to improve the signal-to-noise ratio at the output of an A/D converter. Also covered

此为试读，需要完整PDF请访问：www.ertongbook.com

understood in the Fourier transform domain. Therefore, the first three chapters of this book are devoted to an overview of continuous time system theory, which includes a review of some familiar concepts as well as the introduction of advanced ones.

Digital signal processing requires a way of looking at things that is perhaps more theoretical or mathematically intense than electrical engineering technology students are used to. For example, students can learn much about a continuous time filter circuit without ever considering the differential equation that describes it. The differential equation is one theoretical description of the filter and can be used to gain insight into the behavior of the filter, but this equation is not directly used to implement the filter. The bottom line is that the continuous time filter can be thought of first and foremost as a *circuit*, and its response can be tinkered with by changing things such as resistor and capacitor values. On the other hand, a discrete time system must be thought of first and foremost as a *mathematical algorithm*, since the *difference equation* that describes the system is directly used to implement the system by means of a computer program. To implement this algorithm in real time, a complicated circuit involving a microprocessor or specialized DSP chip, A/D and D/A converters, and so on, must be built; however, one does not tinker with the hardware as such to change the system response. The response of a discrete time system depends on the computer program that implements the system difference equation; the coefficients of this equation determine whether the system is, for example, a digital lowpass filter or a digital highpass filter, and what the cutoff frequency is.

The reader (or instructor) who is anxious to get to the heart of the book—DSP itself—can go directly from Chapter 3 to Chapter 5. However, the material in Chapter 4 on the frequency response and transfer functions of Butterworth and Chebyshev filters should be covered before Chapter 10.

The heart and soul of this book are contained in Chapters 5 through 10. Chapter 5 introduces the concept of a linear, shift-invariant (LSI) discrete time system and the difference equation that describes it. Important topics such as the discrete time system impulse response, discrete time convolution, the Z transform, and the discrete time system transfer function are also presented. Instructors may reduce the amount of time devoted to certain Z transform topics, such as finding the inverse Z transform using partial fraction expansion, which is covered in Section 5–6, but is not needed in the chapters that follow. Chapter 6 covers the discrete time Fourier transform (DTFT) and frequency domain analysis of discrete time signals and systems and discusses the relationship between continuous time and discrete time signal processing. Chapter 7 presents the details of the relationship between the Fourier transform and the DTFT, which is initially presented on an "it can be shown" basis in Chapter 6. The sampling theorem is also presented in Chapter 7, along with a discussion of ideal versus real-world digital-to-analog conversion. In Chapter 8, the student is introduced to the discrete Fourier transform (DFT) and the fast Fourier transform (FFT). Chapter 9 discusses the design of finite impulse response (FIR) filters using the Kaiser window method and the Parks-McClellan algorithm. Chapter 10 discusses the design of infinite impulse response (IIR) filters using the bilinear transformation.

Chapter 11 presents a topic that best illustrates the power of DSP—adaptive FIR filters based on the LMS algorithm. Chapter 12 begins by considering random signals and their associated power spectral density functions, and then shows how *oversampling* can be used to improve the signal-to-noise ratio at the output of an A/D converter. Also covered

BRIEF CONTENTS

CONTENTS

此为试读，需要完整PDF请访问：www.ertongbook.com

APPENDICES

INTRODUCTION

When engineers talk about digital signal processing (DSP), they are essentially discussing *discrete time* signal processing, as opposed to *continuous time* (analog) signal processing. (Discrete time signal processing theory is basically DSP theory minus the explicit consideration of various nonideal "finite precision" effects due to A/D conversion quantization error, finite precision computer arithmetic, etc.) The purpose of this brief introduction is to describe the difference between continuous time and discrete time signal processing and to explain some of the advantages of DSP. It is assumed that the reader is already familiar with continuous time signal processing, having previously studied a variety of circuits that carry out this kind of processing (for example, an active lowpass filter circuit).

Continuous time signal processing is represented by the system shown in Figure I–1. The input signal, $x_a(t)$, is a continuous function of time (t), usually a time-varying voltage at the output of a transducer of some sort (e.g., a microphone). The continuous time system is a circuit of some kind, such as an active lowpass filter circuit (built with opamps [operational amplifiers], resistors, and capacitors). The output of the system, $y_a(t)$, is also a continuous time signal.

The continuous time signal processing system of Figure I–1 should be compared with the digital signal processing system shown in Figure I–2. For the purposes of this discussion, let's assume that the two systems have been designed to achieve essentially the same purpose in that $x_a(t)$ is the same in both cases and the desired result with respect to $y_a(t)$ is also the same. In the DSP system, the continuous time input signal is *sampled* by an analog-to-digital (A/D) converter once every T seconds. At the output of the A/D converter, a new *number* appears once every T seconds. Therefore, a sequence of numbers $x(n)$ (where n is the sample number) appears at the output of the A/D converter. Assuming an *ideal* A/D converter, this sequence is related to the original continuous time signal as follows:

$$x(n) = x_a(nT) \qquad \textbf{(I–1)}$$

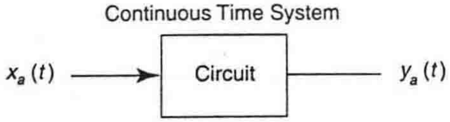


FIGURE I–1
Continuous time system.

The sequence of numbers, $x(n)$, is called a *discrete time signal.* This discrete time signal is processed by a mathematical algorithm implemented by a computer program. The algorithm generates an output sequence of numbers, denoted as $y(n)$ in Figure I–2. The output sequence $y(n)$ is processed by a digital-to-analog (D/A) converter to create an output continuous time signal $y_a(t)$.

Strictly speaking, the subject of "discrete time signal processing" pertains to the subsystem in Figure I–2 that is enclosed by the dashed line. A discrete time input signal $x(n)$ is processed by a discrete time system (which is the algorithm implemented by the computer) to create a discrete time output signal $y(n)$. The overall DSP system of Figure I–2 has a continuous time input and output and processes the continuous time signal using discrete time methodology.

If the input sequence $x(n)$ is immediately processed by the computer, such that each time an input number is fetched from the A/D converter an output number is calculated and fed to the D/A converter, then the DSP system of Figure I–2 is operating in "real time." In some applications, it is permissible to operate in "non-real time," as depicted in Figure I–3. In the non-real time, or off-line processing scenario, the sequence of numbers $x(n)$ is stored in a computer file. This file is later accessed by the computer program, which generates the output sequence $y(n)$ without being forced to do so at the same speed that real-time processing would require. The output sequence is written to another computer file, which can later be accessed by a D/A converter or perhaps be subjected to additional processing. It should be emphasized here that the basic principles and theory underlying discrete time signal processing are the same regardless of whether the actual processing rate is real time or non-real time.

There was a time not very many years ago when DSP was an esoteric subject studied at the graduate school level. There have been significant applications for DSP for many years, applications that span a number of different disciplines (for example, seismic exploration for oil and gas, speech and image processing, sonar and radar signal processing,

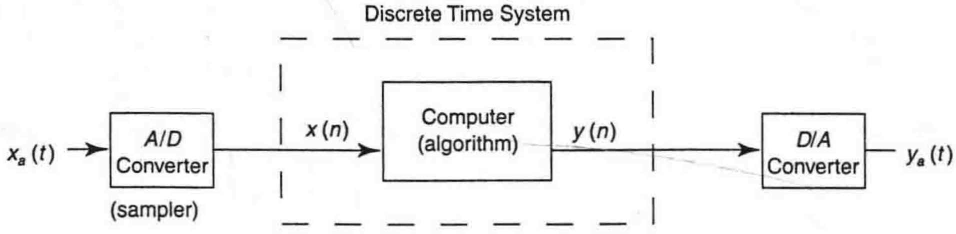


FIGURE I–2
Digital signal processing system. The subsystem within the dashed line is a discrete time system.

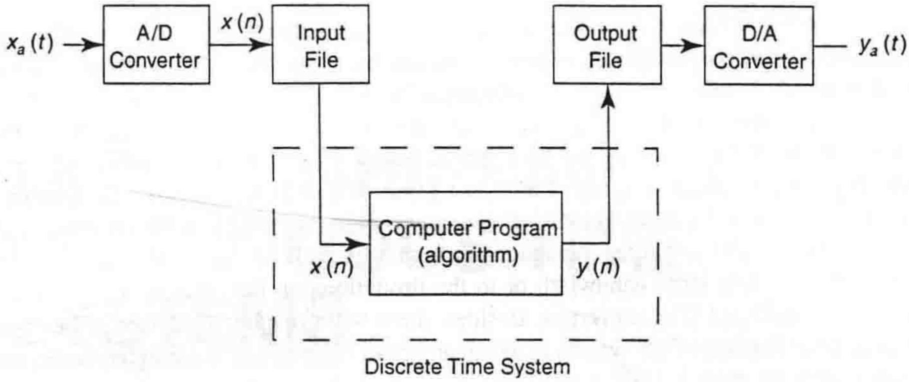


FIGURE I–3
Non-real time (off-line) DSP.

etc.). In earlier times (say 20 years ago), most DSP was carried out in non-real time. The theory was well developed, but the available digital computers and other supporting hardware (e.g., A/D converters) could not operate fast enough to make real-time DSP practical. In more recent years, rapid advances in integrated circuit technology have made real time DSP a realistic option for an increasing number of applications. DSP is now a buzzword even in the world of consumer electronics.

Let us once again compare the systems of Figure I–1 and Figure I–2 and assume that both systems have the same input signal and achieve essentially the same result with respect to the output signal. The DSP system certainly looks more complicated than the continuous time system; it is fair to ask what advantages are gained from such complexity. The main answer is that the DSP system has tremendous flexibility relative to the continuous time system. Suppose the purpose of these systems is to filter the input signal but that the exact filter requirements will vary according to changing conditions or needs. For example, suppose each of these systems represents a bandpass filter, but the bandwidth and center frequency may need to be changed by the user of the system. To change the continuous time system, various circuit components (resistors, capacitors, etc.) must be changed. To change the discrete time system, the algorithm must be changed; this is just a matter of changing some numbers in the program! In other words, the DSP system of Figure I–2 is programmable and can can be changed on the fly. In fact, there are *adaptive* filter algorithms that actually realize self-adjusting filters, such as a narrow bandstop (notch) filter that tracks an interfering sinusoid characterized by a changing frequency (this is discussed in Chapter 11). In addition to this flexibility, the DSP system can realize certain types of filters that are either impossible or impractical to implement with a continuous time system; an example is an extremely sharp lowpass or bandpass filter having an exact "linear phase" characteristic in the passband.

We don't mean to imply that DSP has made all continuous time filtering operations obsolete. As will be shown in Chapter 6, the usable bandwidth of a DSP system is equal to one-half of the sampling frequency. (If the A/D converter takes one sample every T seconds, the sampling frequency is $f_s = 1/T$ Hertz.) The sampling frequency must be greater than two times the bandwidth of the signal being sampled (again, for reasons that are explained in Chapter 6); this places a *lower bound* on the allowable sampling frequency. On the other hand, if the DSP system is to function in real time, the system must be able to fetch a sample from the A/D converter, execute the algorithm, and output a number to the D/A converter in T seconds or less; this puts an *upper bound* on the allowable sampling frequency. (There are also limits on how fast a given A/D or D/A converter can function.) Obviously, there can be cases in which the system requirements with respect to lower and upper bounds on the sampling frequency are mutually exclusive due to the input signal having an extremely large bandwidth or to the limitations on the speed of the proposed processor or A/D and D/A converters. Besides, there will probably always be some cases (at least in the lifetime of the author) in which an active filter circuit is a simpler, faster, and cheaper solution even if DSP could easily be used instead. Therefore, continuous time signal processing is still extremely important.

Figures I–2 and I–3 suggest that the desired output from a DSP system is always another continuous time signal, but this is not always the case. Sometimes the purpose of the DSP system is to *analyze* the input signal in some sense; in this case, the desired output is some form of information about the input signal. Figure I–4 shows the basic idea. A classic example is spectral analysis, for which the purpose of the system is to extract and display information about the frequency content, or *spectrum*, of the signal. In this case, the signal samples, $x(n)$, are processed by some kind of spectral analysis program, which is quite often based on the *Fast Fourier Transform* (FFT). (The FFT is covered in Chapter 8.)

Before going on to Chapter 1, read the Preface to this book if you have not already done so. Among other things, the Preface explains why Chapters 1, 2, 3, and 4 are devoted to continuous time signal processing instead of digital signal processing. There is good reason that the background information on continuous time signal and system theory must be covered before considering the DSP theory itself (which begins with Chapter 5).

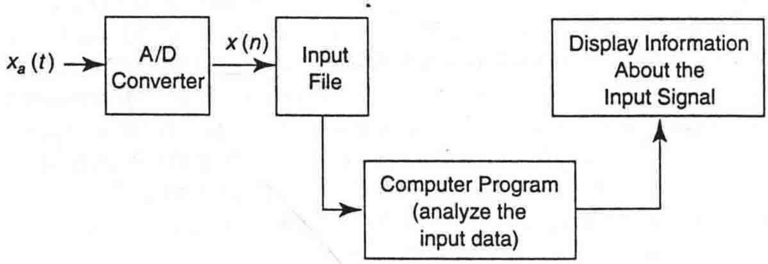


FIGURE I–4
Using DSP for signal analysis. A classic example is spectral analysis, in which the purpose is to determine the frequency content of the input signal.

1

LINEAR, SHIFT-INVARIANT CONTINUOUS TIME SYSTEMS

1–1 TIME DOMAIN DESCRIPTION

We begin by considering linear, shift-invariant (LSI) systems in a very general sense. Let $x(t)$ be the input signal to the system. (For example, $x(t)$ could be a time-varying voltage.) There is some operation, denoted $T[\ ]$, that creates an output signal $y(t)$:

$$y(t) = T\left[x(t)\right] \tag{1–1}$$

Figure 1–1 illustrates the basic idea.

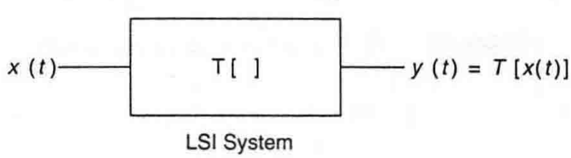
In this book it is assumed that the system under consideration is "at rest" before the input signal is applied. In other words, it is assumed that all of the initial conditions are zero. For example, if the system is a circuit having inductors and capacitors, the initial conditions are that all of the capacitors are discharged and all of the inductor currents are zero.

A system is said to be *linear* if superposition holds. That is, $T[\ ]$ represents a linear system if the following relationship holds:

$$T\left[a_1x_1(t) + a_2x_2(t)\right] = a_1T\left[x_1(t)\right] + a_2T\left[x_2(t)\right] \tag{1–2}$$

A circuit having only passive components (resistors, capacitors, and inductors) is an example of a linear system. A circuit having active devices (such as transistors) could be either linear or nonlinear. One distinguishing characteristic of a linear system is that if the input signal is an undistorted sinusoid, the output signal will also be an undistorted sinusoid at the

FIGURE 1–1
Linear, shift-invariant system.



此为试读，需要完整PDF请访问：www.ertongbook.com