DISCRETE-TIME SIGNAL PROCESSING

SECOND EDITION

DISCRETE-TIME SIGNAL PROCESSING

ALAN V. OPPENHEIM
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RONALD W. SCHAFER
GEORGIA INSTITUTE OF TECHNOLOGY

WITH

JOHN R. BUCK
UNIVERSITY OF MASSACHUSETTS DARTMOUTH

PRENTICE HALL
UPPER SADDLE RIVER, NEW JERSEY 07458

Oppenheim, Alan V.
Discrete-time signal processing / Alan V. Oppenheim, Ronald W. Schafer, with John R. Buck. — 2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-754920-2
1. Signal processing—Mathematics. 2. Discrete-time systems.
I. Schafer, Ronald W. II. Buck, John R. III. Title.
TK5102.9.067 1998
621.382′2—dc21 98-50398
CIP

Acquisitions editor: **Tom Robbins**
Production service: **Interactive Composition Corporation**
Editorial/production supervision: **Sharyn Vitrano**
Copy editor: **Brian Baker**
Cover design: **Vivian Berman**
Art director: **Amy Rosen**
Managing editor: **Eileen Clark**
Editor-in-Chief: **Marcia Horton**
Director of production and manufacturing: **David W. Riccardi**
Manufacturing buyer: **Pat Brown**
Editorial assistant: **Dan De Pasquale**

Published by Prentice-Hall, Inc.
Upper Saddle River, New Jersey 07458

Printed in the United States of America

10 9 8 7 6 5 4

ISBN 0-13-754920-2

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*

4

Sampling of Continuous-Time Signals

4.0 INTRODUCTION

Discrete-time signals can arise in many ways, but they most commonly occur as representations of sampled continuous-time signals. It is remarkable that under reasonable constraints, a continuous-time signal can be quite accurately represented by samples taken at discrete points in time. In this chapter we discuss the process of periodic sampling in some detail, including the phenomenon of aliasing, which occurs when the signal is not bandlimited or when the sampling rate is too low. Of particular importance is the fact that continuous-time signal processing can be implemented through a process of sampling, discrete-time processing, and the subsequent reconstruction of a continuous-time signal.

4.1 PERIODIC SAMPLING

Although other possibilities exist (see Steiglitz, 1965; Oppenheim and Johnson, 1972), the typical method of obtaining a discrete-time representation of a continuous-time signal is through periodic sampling, wherein a sequence of samples, $x[n]$, is obtained from a continuous-time signal $x_c(t)$ according to the relation

$$x[n] = x_c(nT), \qquad -\infty < n < \infty. \tag{4.1}$$

In Eq. (4.1), T is the *sampling period*, and its reciprocal, $f_s = 1/T$, is the *sampling*

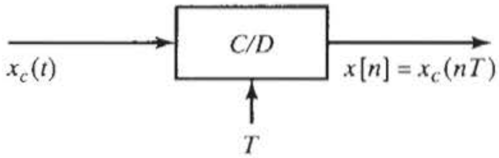

Figure 4.1 Block diagram representation of an ideal continuous-to-discrete-time (C/D) converter.

frequency, in samples per second. We also express the sampling frequency as $\Omega_s = 2\pi/T$ when we want to use frequencies in radians per second.

We refer to a system that implements the operation of Eq. (4.1) as an *ideal continuous-to-discrete-time (C/D) converter,* and we depict it in block diagram form as indicated in Figure 4.1. As an example of the relationship between $x_c(t)$ and $x[n]$, in Figure 2.2 we illustrated a continuous-time speech waveform and the corresponding sequence of samples.

In a practical setting, the operation of sampling is implemented by an analog-to-digital (A/D) converter. Such systems can be viewed as approximations to the ideal C/D converter. Important considerations in the implementation or choice of an A/D converter include quantization of the output samples, linearity of quantization steps, the need for sample-and-hold circuits, and limitations on the sampling rate. The effects of quantization are discussed in Sections 4.8.2 and 4.8.3. Other practical issues of A/D conversion are electronic circuit concerns that are outside the scope of this text.

The sampling operation is generally not invertible; i.e., given the output $x[n]$, it is not possible in general to reconstruct $x_c(t)$, the input to the sampler, since many continuous-time signals can produce the same output sequence of samples. The inherent ambiguity in sampling is a fundamental issue in signal processing. Fortunately, it is possible to remove the ambiguity by restricting the input signals that go into the sampler.

It is convenient to represent the sampling process mathematically in the two stages depicted in Figure 4.2(a). The stages consist of an impulse train modulator followed by conversion of the impulse train to a sequence. Figure 4.2(b) shows a continuous-time signal $x_c(t)$ and the results of impulse train sampling for two different sampling rates. Figure 4.2(c) depicts the corresponding output sequences. The essential difference between $x_s(t)$ and $x[n]$ is that $x_s(t)$ is, in a sense, a continuous-time signal (specifically, an impulse train) that is zero except at integer multiples of T. The sequence $x[n]$, on the other hand, is indexed on the integer variable n, which in effect introduces a time normalization; i.e., the sequence of numbers $x[n]$ contains no explicit information about the sampling rate. Furthermore, the samples of $x_c(t)$ are represented by finite numbers in $x[n]$ rather than as the areas of impulses, as with $x_s(t)$.

It is important to emphasize that Figure 4.2(a) is strictly a mathematical representation that is convenient for gaining insight into sampling in both the time domain and frequency domain. It is not a close representation of any physical circuits or systems designed to implement the sampling operation. Whether a piece of hardware can be construed to be an approximation to the block diagram of Figure 4.2(a) is a secondary issue at this point. We have introduced this representation of the sampling operation because it leads to a simple derivation of a key result and because the approach leads to a number of important insights that are difficult to obtain from a more formal derivation based on manipulation of Fourier transform formulas.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.