GLOBAL EDITION

DSP First

SECOND EDITION

James H. McClellan
Ronald W. Schafer
Mark A. Yoder

PEARSON

DSP First

Second Edition

Global Edition

James H. McClellan
Georgia Institute of Technology

Ronald W. Schafer
Stanford University

Mark A. Yoder
Rose-Hulman Institute of Technology

PEARSON

Boston • Columbus • Indianapolis • New York • San Francisco
Hoboken • Amsterdam • Cape Town • Dubai • London • Madrid • Milan • Munich
Paris • Montreal • Toronto • Delhi • Mexico City • São Paulo
Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Vice President and Editorial Director, ECS: Marcia J. Horton
Aquisitions Editor: Julie Bai
Editorial Assistant: Michelle Bayman
Senior Acquisitions Editor, Global Editions: Sandhya Ghoshal
VP of Marketing: Christy Lesko
Director of Field Marketing: Tim Galligan
Product Marketing Manager: Bram van Kempen
Field Marketing Manager: Demetrius Hall
Marketing Assistant: Jon Bryant
Senior Managing Editor: Scott Disanno
Program Manager: Joanne Manning
Project Editor, Global Editions: K.K. Neelakantan
Senior Manufacturing Controller, Global Editions: Trudy Kimber
Media Production Manager, Global Editions: Vikram Kumar
Global HE Director of Vendor Sourcing and Procurement: Diane Hynes
Operations Specialist: Maura Zaldivar-Garcia
Creative Director: Blair Brown
Senior Art Director: Janet Slowik
Cover Designer: Lumina Datamatics
Manager, Rights and Permissions: Rachel Youdelman
Full-Service Project Management: Pavithra Jayapaul, Jouve India

Image credits:
Cover photo: fotographic1980/Shutterstock.com
Image on page 22: James McClellan, Ronald Schafer, Mark Yoder

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsonglobaleditions.com

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

ISBN 10: 1292113863

ISBN 13: 9781292113869

Typeset in 10/12 Times LT Std Roman by Jouve India

Printed and bound in Malaysia.

Contents

3 Spectrum Representation 69

6 Frequency Response of FIR Filters 214

7 Discrete-Time Fourier Transform 259

10 IIR Filters 414

B Programming in MATLAB 511

C Fourier Series 527

D Laboratory Projects 571

Index 574

This page intentionally left blank

Preface

This book, entitled simply *DSP First*, is the second edition of the text *DSP First: A Multimedia Approach* (1998) which was packaged with a CD-ROM that provided many resources to extend the boundaries of a traditional textbook. In 2003, a second book entitled *Signal Processing First* was produced with a broader set of topics that included four new chapters on continuous-time signal processing and the Fourier transform, as well as updated versions of the first eight chapters of *DSP First*. New material was produced for the CD-ROM bundled with the 2003 textbook, and all the supporting resources have now moved to a website for easier access.

These three books and the Companion Website are the result of more than 20 years of work grounded on the premise that digital signal processing (DSP) is an ideal starting point for the study of both electrical engineering and computer engineering. In the summer of 1993, two of us (JHMc and RWS) began to develop a one-quarter course that was to become the required introductory course for Georgia Tech computer engineering (CmpE) students. We argued that the subject of digital signal processing had everything we wanted in a first course for computer engineers: it introduced the students to the use of mathematics as a language for thinking about and solving engineering problems; it laid useful groundwork for subsequent courses; it made a strong connection to digital computation as a means for implementing systems; and it provided the tools to discuss interesting applications that would motivate beginning engineers to do the hard work of connecting mathematics and computation to problem solving. Nothing has happened in the past 22 years to change our minds on this point. Indeed, our teaching experience with more than 6,000 students at Georgia Tech has only strengthened our conviction that digital signal processing, distilled to its essence, is an ideal introductory subject for *both* electrical and computer engineering students.[1] In fact, we have become firmly convinced that a course on DSP at the level of this text should be required of every engineering and computer science student.

From the beginning, we believed that "hands-on" experience with real signals was crucial, so we expended considerable effort on developing additional material for laboratory exercises and projects based on MATLAB. In the laboratory assignments, students can experience the effects of signal processing operations that they have implemented on sound and image signals. For example, they can synthesize music from sinusoids, but they can also see that those same sinusoids are the basis for the wireless systems that they use routinely to access the Internet. These experiences, available on the Companion Website, will augment and reinforce the mathematical concepts that form the basis of DSP.

In addition to the 25 detailed lab assignments, the Companion Website includes many resources that extend the printed textbook with material such as demonstrations and animations used in classes, and hundreds of solved homework problems. The impetus for having this website came from Mark Yoder who, in 1995, while on sabbatical leave at Georgia Tech from Rose-Hulman, had the idea to put all of this material into a form that other teachers (and students) could access easily. Interactive MATLAB demonstrations have been created for demonstrating specific topics such as convolution and frequency response, and most of these are now used as the basis for some of the laboratory exercises. As teachers, all this material has changed the way we present ideas, because it expands the ways to visualize a concept "beyond the equations." Over the years, the collection of resources on our website has continued to grow. In the future, we will explore new ideas for presenting the concepts of DSP, and hope to move beyond the printed page to an e-Text version that would truly integrate the narrative of the book with the visualizations of the companion website.

The distinguishing feature of this text (and its progenitors) is that it presents signal processing at a level consistent with an introductory ECE course, i.e., the sophomore level (second year) in a typical U.S. university. The list of topics in the book is not surprising given its emphasis on discrete-time signal processing, but since we want a course that is broadly accessible to sophomores, we feel that we must combine signal processing concepts with some introductory ideas. Part of the reason for this is that in many electrical engineering curriculums, signals and systems and DSP typically have been treated as junior- and senior-level courses, for which a traditional background of linear circuits and linear systems is assumed. Our approach, on the other hand, makes the subject much more accessible to students in other majors such as computer science

[1] In our development of these ideas, two books by Professor Ken Steiglitz of Princeton University had a major impact on our thinking: *An Introduction to Discrete Systems*, John Wiley & Sons, 1972, and *A Digital Signal Processing Primer: With Applications to Computer Music*, Addison-Wesley Publishing Company, 1996. Steiglitz's 1972 book was well ahead of its time, since DSP had few practical applications, and even simple simulations on then-available batch processing computers required significant programming effort. However, by 1993 when we began our work, easy-to-use software environments such as MATLAB were widely available for implementing DSP computations on powerful personal computers.

and other engineering fields. This point is increasingly important because non-specialists need to use DSP techniques routinely in many areas of science and technology.

Content of the New Edition. This new edition has an organization similar to the first edition of *DSP First*. A look at the table of contents shows that the book begins very simply (Chapter 2) with a detailed discussion of continuous-time sinusoidal signals and their representation by complex exponentials. This is a topic traditionally introduced in a linear circuits course, but including it here makes it immediately accessible for the rest of this book, especially for students who come from other backgrounds. If students have already studied linear circuits, this chapter can be skipped, or rapidly covered. We then proceed to introduce the spectrum concept (Chapter 3) by considering sums of sinusoidal signals, culminating with a brief introduction to Fourier series. Although Chapter 3 of the first edition covered the same basic ideas, this chapter has some new material.[2]

Next we make the transition to discrete-time signals by considering sampled sinusoidal signals (Chapter 4). We have found that it is not necessary to invoke the continuous-time Fourier transform to make the important issues in sampling clear. All that is needed is the simple trigonometric identity $\cos(\theta + 2\pi) = \cos(\theta)$. In fact, in Chapters 2–4 (with the exception of Fourier Series), we have only needed to rely on the simple mathematics of sine and cosine functions. The basic linear system concepts are then introduced with running average systems and other simple FIR filters (Chapter 5). Impulse sequences are introduced which leads to the impulse response characterizing a filter. Convolution is treated as a numerical operation in the first pass at this idea. The key concept of frequency response is derived and interpreted for FIR filters (Chapter 6). Sinusoids are the primary signals of interest, and we emphasize the magnitude and phase change experienced by a sinusoid when filtered by a linear time-invariant system.

At this point we depart significantly from the first edition by introducing (Chapter 7) the concept of discrete-time Fourier transform (DTFT), which arises naturally from the frequency response of a discrete-time system. The concept of the inverse DTFT completes the description of an invertible transform and also enables us to describe ideal filters. It is then natural to move from the DTFT to the discrete Fourier transform (DFT), which is simply a sampled version of the DTFT and thus computable through fast algorithms that are readily available (Chapter 8). Chapters 7 and 8 are completely new. They are a response to frequent requests from teachers who want to expose their students to the powerful concept of the Fourier transform, and we have found that sophomores are fully capable of understanding these concepts and putting them to use. These two chapters bring many of the ideas of practical spectrum analysis into focus with the goal of providing the knowledge to successfully employ the powerful spectrum analysis tools readily available in software environments such as MATLAB.

Finally, the last two chapters return to the flow of the first edition. We introduce z-transforms (Chapter 9) and IIR systems (Chapter 10). At this stage, a student who has faithfully read the text, worked homework problems, and done the laboratory assignments will be rewarded with the ability to understand applications involving the

[2]Furthermore, for instructors who prefer to dive deeper into Fourier analysis of periodic signals, Appendix C on Fourier series is essentially another entire chapter on that topic.

sampling theorem, discrete-time filtering, and spectrum analysis. Furthermore, they are well prepared to move on to courses in linear analog circuits, continuous-time signals and systems, and control systems. All of these courses can build on the foundation established through the study of this text.

Summary of What's New in This Edition

- New material on the Discrete-Time Fourier Transform (DTFT) has been developed and is presented in Chapter 7. The presentation makes an easy transition from the frequency response concept to begin the study of the general idea of a Fourier transform.
- New material on ideal filters and digital filter design is presented in Chapter 7 as a very useful application of the DTFT. The window method for FIR filter design is presented in detail.
- New material on the Discrete Fourier Transform (DFT) has been developed and is presented in Chapter 8. The presentation adopts the point of view that the DFT is a sampled version of the DTFT, and also develops the relationship of the DFT to the discrete Fourier series (DFS).
- New material on spectrum analysis and the spectrogram has been developed for the last sections of Chapter 8. This provides a solid foundation for understanding time-frequency analysis of signals as is commonly done with the FFT algorithm, as well as the role of windowing in frequency resolution.
- Chapters 7 and 8 are derived from Chapter 9 in the first edition and Chapter 13 in *Signal Processing First*. The new chapters are a significant rewrite to make this material accessible at the introductory level. The benefit is that students can learn the ideas of practical spectrum analysis which can then be reinforced with a lab experience where actual signals are processed with the tools available in MATLAB.
- The presentation of the spectrum in Chapter 3 has been expanded to include a formal discussion of properties of the spectrum (e.g., time-delay, frequency shifting). This sets the stage for later discussions of the DTFT and DFT.
- The material on Fourier Series which was part of Chapter 3 has been expanded, but most of it is now placed in Appendix C. Chapter 3 contains a sufficient description of the Fourier series to present the spectrum of one periodic signal, the full wave rectified sine. Appendix C provides an in-depth presentation for instructors who choose to emphasize the topic. Details of other periodic signals (square wave, triangular wave, and half-wave rectified sine) are given along with a derivation of Parseval's theorem and a heuristic discussion of convergence. Properties of the Fourier Series are also developed.
- Extensive changes have been made to the end-of-chapter problems. There are a total of 241 problems in the book: 83 are new, 86 are different from the first edition by virtue of changing the details, and 72 are the same as in the first edition.

- The Companion Website contains new material for labs, MATLAB visualizations, and solved homework problems. The Companion Website may be found at www.pearsonglobaleditions.com/McClellan.

At Georgia Tech, our sophomore-level, 3 credit course covers most of the content of Chapters 2–10 in a format involving two one-hour lectures, one 1.5 hour recitation, and one 1.5 hour laboratory period per week. As mentioned previously, we place considerable emphasis on the lab because we believe that it is essential for motivating our students to learn the mathematics of signal processing, and because it introduces our students to the use of powerful software in engineering analysis and design. At Rose-Hulman, we use *DSP First* in a freshman-level, 10-week course that covers Chapters 1–6, 9, and 10. The Rose format is 3 one-hour lectures per week and one three-hour lab. The students use MATLAB throughout the course. The entire content of the present text was used by RWS for a 10-week, four credit course at Stanford University. Since this course followed quarter-long courses in continuous-time signals and systems and linear circuits, it was possible to skip Chapters 2 and 3 and move immediately into a focus on discrete-time signals and systems using the remaining chapters. One credit was devoted to a weekly lab assignment which was done individually without a regularly scheduled laboratory period.

These examples from our own teaching experience show that the text and its associated supporting materials can be used in many different ways depending on instructor preference and number of course hours. As can be seen from the previous discussion, the second edition of *DSP First* is not a conventional signals and systems book. One difference is the inclusion of a significant amount of material on sinusoids and complex phasor representations. In a traditional electrical engineering curriculum, these basic notions are covered under the umbrella of linear circuits taken before studying signals and systems. Indeed, our choice of title for this book and the first edition is designed to emphasize this departure from tradition. An important point is that teaching signal processing first also opens up new approaches to teaching linear circuits, since there is much to build upon that will allow redirected emphasis in the circuits course.

A second difference from conventional signals and systems texts is that *DSP First* emphasizes topics that rely on "frequency domain" concepts. This means that, in an electrical engineering curriculum, topics like Laplace transforms, state space, and feedback control, would have to be covered in later courses such as linear circuits or an upper-level course on control systems. Although our text has clearly been shaped by a specific point of view, this does not mean that it and the associated website can be used in only one way. Indeed, as our own experience shows, by appropriate selection of topics, our text can be used for either a one-quarter or one-semester signals and systems course that emphasizes communications and signal processing applications from the frequency domain point of view. For most electrical engineering curricula, the control-oriented topics would be covered in another course.

In other disciplines such as computer science and computer engineering, *DSP First* emphasizes those topics that are most relevant to computing for signal analysis. This is also likely to be true in other engineering fields where data acquisition and frequency domain analysis play an important role in modern engineering analysis and design.

This text and its Companion Website represents an untold amount of work by the three authors, numerous colleagues, and many students. Fortunately, we have been able to motivate a number of extremely talented students to contribute MATLAB demos to this project. There are so many that to name them all would be impractical. We simply thank them all for their valuable contributions to our project. Greg Krudysz who authored several of the demos has now taken over the primary role of developing new demos and visualizations with GUIs and updating the existing ones. Since the beginning in 1993, many professors have participated in the sophomore course ECE-2025 (and now ECE-2026) at Georgia Tech as lecturers and recitation instructors. Once again, naming all the recitation instructors would be impractical, but their comments and feedback have given life to the course as it evolved during the past 12 years. For example, Pamela Bhatti developed a laboratory on simulating the filter bank of a Cochlear Implant hearing system. Recently, the lecturing and administration of the course has been shared by Russ Mersereau, Fred Juang, Chin Lee, Elliot Moore, Mark Clements, Chris Rozell, G. K. Chang, David Taylor, David Anderson, John Barry, Doug Williams, and Aaron Lanterman. We are indebted to them for their many suggestions that have made a positive impact on this second edition, especially the new material on the DFT and DTFT. We are also indebted to Wayne Padgett and Bruce Black, who have taught ECE-380 at Rose-Hulman and have contributed many good ideas, and we appreciate the work of Ed Doering who created a whole new set of labs for ECE-180, the new freshman-level DSP First. These labs start with traditional audio processing and end with video object tracking. A new first for freshman.

We also want to acknowledge the contributions of Tom Robbins (formerly at Pearson Prentice-Hall) who was an early supporter of our efforts to bring DSP to the fore in ECE education. Tom bought into our concept of *DSP First* from the beginning, and he encouraged us during the initial project, as well as the 2003 book. More recently, Andrew Gilfillan and Julie Bai have been the editors who helped make this second edition a reality.

Finally, we want to recognize the understanding and support of our wives (Carolyn McClellan, Dorothy Schafer, and Sarah Yoder). Carolyn's photo of the cat Kilby appears in Chapter 1. They have patiently supported us as this multi-year project continued to consume energy and time that might have been spent with them.

JHMc
RWS
MAY

Pearson would like to thank and acknowledge Ankita Pramanik, Indian Institute of Engineering Science and Technology, Shibpur, and Sanjay H.S., M.S. Ramaiah Institute of Technology, for contributing to the Global Edition, and Chih-Wei Liu, National Chiao Tung University, Siddhartha Bhattacharya, RCC Institute of Information Technology, and Neeraj Varshney, Indian Institute of Technology Kanpur, for reviewing the Global Edition.

CHAPTER

1

Introduction

This is a book about signals and systems. In this age of multimedia gaming computers, audio and video entertainment systems, and smartphones, it is almost certain that you, the reader of this text, have formed some impression of the meaning of the terms ***signal*** and ***system***, and you probably use the terms often in daily conversation.

It is likely that your usage and understanding of the terms are correct within some rather broad definitions. For example, you may think of a signal as "something" that carries information. Usually, that something is a pattern of variations of a physical quantity that can be manipulated, stored, or transmitted by physical processes. Examples include speech signals, audio signals, video or image signals, biomedical signals, radar signals, and seismic signals, to name just a few. An important point is that signals can take many equivalent forms or *representations.* For example, a speech signal is produced as an acoustic signal, but it can be converted to an electrical signal by a microphone, and then to a string of numbers as in digital audio recording.

The term ***system*** may be somewhat more ambiguous and subject to interpretation. For example, we often use "system" to refer to a large organization that administers or implements some process, such as the "Social Security system" or the "airline transportation system." However, we are interested in a much narrower definition that is very closely linked to signals. More specifically, a system, for our purposes, is something that can manipulate, change, record, or transmit signals. For example, a DVD recording stores or represents a movie or a music signal as a sequence of numbers. A DVD player is a system for converting the numbers stored on the disc (i.e., the numerical representation of the signal) to a video and/or acoustic signal. In general, systems ***operate*** on signals to produce new signals or new signal representations.

Our goal in this text is to develop a framework wherein it is possible to make precise statements about both signals and systems. Specifically, we want to show that mathematics is an appropriate language for describing and understanding signals and systems. We also want to show that the representation of signals and systems by mathematical equations allows us to understand how signals and systems interact and how we can design and implement systems that achieve a prescribed purpose.

1-1 Mathematical Representation of Signals

Signals are patterns of variations that represent or encode information. Many signals are naturally thought of as a pattern of variations in time. A familiar example is a speech signal, which initially arises as a pattern of changing air pressure in the vocal tract. This pattern, of course, evolves with time, creating what we often call a ***time waveform***. Figure 1-1 shows a plot of a recorded speech waveform. In this plot, the vertical axis represents microphone voltage (proportional to air pressure), and the horizontal axis represents time. Notice that there are four plots in the figure corresponding to four contiguous time segments of the speech waveform. The second plot is a continuation of the first, and so on, with each graph corresponding to a time interval of 50 milliseconds (ms).

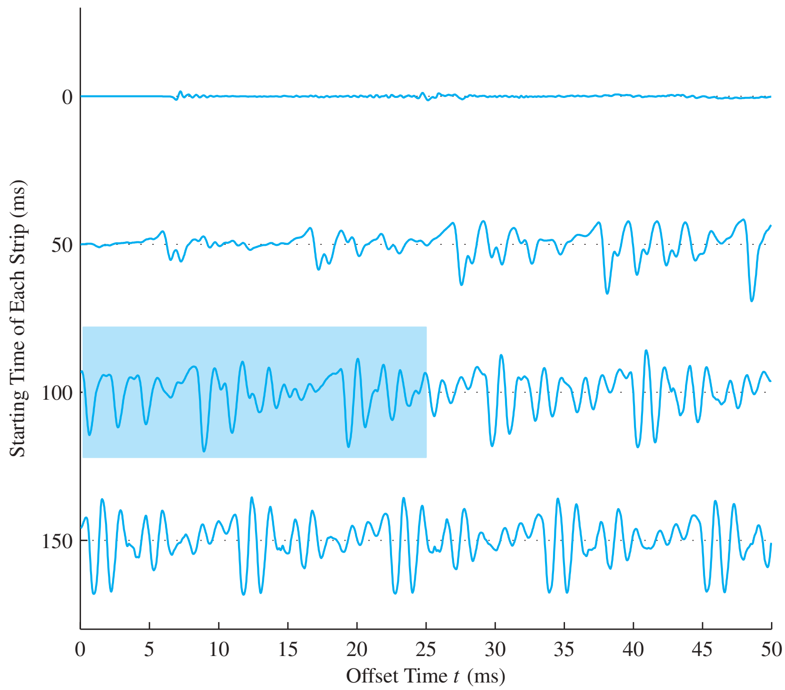

Figure 1-1 Strip plot of a speech signal where each row is a continuation of the row above. This signal $s(t)$ can be represented as a function of a single (time) variable. The shaded region is shown in more detail in Fig. 1-2.

The speech signal in Fig. 1-1 is an example of a one-dimensional ***continuous-time signal***. Such signals can be represented mathematically as a function of a single independent variable, which is normally called time and denoted t. Although in this particular case we cannot write a simple equation that describes the graph of Fig. 1-1 in terms of familiar mathematical functions, we can nevertheless associate a function $s(t)$ with the graph. Indeed, the graph itself can be taken as a definition of the function that assigns a number $s(t)$ to each instant of time (each value of t).

Many, if not most, signals originate as continuous-time signals. However, for reasons that will become increasingly obvious as we progress through this text, it is often desirable to obtain a discrete-time representation of a signal. This can be done by ***sampling*** a continuous-time signal at isolated, equally spaced points in time. The result is a sequence of numbers that can be represented as a function of an index variable that takes on only integer values. This can be represented mathematically as $s[n] = s(nT_s)$, where n is an integer (i.e., $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$), and T_s is the ***sampling period***. Note that our convention is to use parentheses ( ) to enclose the independent variable of a continuous-variable function such as $s(t)$, and square brackets [ ] to enclose the independent variable of a discrete-variable function, e.g., the sequence $s[n]$. Sampling is, of course, exactly what we do when we plot values of a function on graph paper or on a computer screen. We cannot evaluate the function at every possible value of a continuous variable, but only at a set of discrete points. Intuitively, we know that the closer the spacing in time of the points, the more the sequence retains the shape of the original continuous-variable function. Figure 1-2 shows an example of a short segment of a discrete-time signal that was derived by sampling the speech waveform of Fig. 1-1 with a sampling period of $T_s = 1/8$ ms. In this case, the dots show the sample values for the sequence $s[n]$.

While many signals can be thought of as evolving patterns in time, many other signals are not time-varying patterns. For example, an image formed by focusing light through a lens is a spatial pattern, and thus is appropriately represented mathematically as a function of two spatial variables. Such a signal would be considered, in general, as a function of two independent variables [i.e., a picture might be denoted $p(x, y)$]. A photograph is another example, such as the gray-scale image shown in Fig. 1-3. In this case, the value $p(x_0, y_0)$ represents the shade of gray at position (x_0, y_0) in the image.

Images such as that in Fig. 1-3 are generally considered to be two-dimensional continuous-variable signals, since we normally consider space to be a continuum.

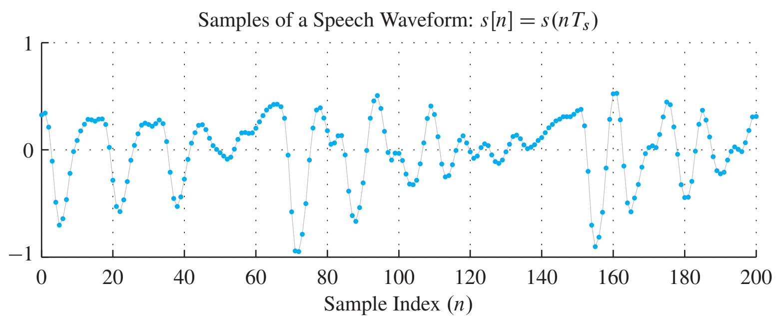

Figure 1-2 Discrete-time signal represented as a one-dimensional sequence which is a function of a discrete variable n. Signal samples are taken from the shaded region of Fig. 1-1. The continuous-time speech signal $s(t)$ is shown in gray.

Figure 1-3 Example of a signal that can be represented by a function of two spatial variables.

However, sampling can likewise be used to obtain a discrete-variable two-dimensional signal from a continuous-variable two-dimensional signal. In a digital camera, this sampling is done by recording light values which have been focused on a sensor array composed of millions of points, or mega-pixels. In a color camera, there would be three separate arrays for RGB: red, green, and blue. A two-dimensional gray-scale image like Fig. 1-3 would be represented by a two-dimensional discrete-variable sequence or an array of numbers, and would be denoted $p[m, n] = p(m\Delta_x, n\Delta_y)$, where both m and n would take on only integer values, and Δ_x and Δ_y are the horizontal and vertical sampling periods, respectively.

Two-dimensional functions are appropriate mathematical representations of still images that do not change with time; on the other hand, videos are time-varying images that would require a third independent variable for time, so a video signal would be denoted $v(x, y, t)$. In analog television, time is discrete (30 frames/s), each horizontal line (x) is continuous, but there are a finite number of horizontal lines, so y is discrete. In present day digital video, all three variables of the video signal $v(x, y, t)$ are discrete since the signal is a sequence of discrete images.

Our purpose in this section has been to introduce the idea that signals can be represented by mathematical functions. Although we will soon see that many familiar functions are quite valuable in the study of signals and systems, we have not even attempted to demonstrate that fact. Our sole concern is to make the connection between functions and signals, and, at this point, functions simply serve as abstract symbols for signals. Thus, for example, now we can refer to “the speech signal $s(t)$” or “the sampled image $p[m, n]$.” Although this may not seem highly significant, we will see in the next section that it is indeed a very important step toward our goal of using mathematics to describe signals and systems in a systematic way.

1-2 Mathematical Representation of Systems

As we have already suggested, a system is something that transforms signals into new signals or different signal representations. This is a rather vague definition, but it is useful as a starting point. To be more specific, we say that a one-dimensional continuous-time system takes an input signal $x(t)$ and produces a corresponding output signal $y(t)$. This can be represented mathematically by

$$y(t) = \mathcal{T}\{x(t)\} \tag{1.1}$$

which means that the input signal (waveform, image, etc.) is operated on by the system (symbolized by the operator $\mathcal{T}$) to produce the output $y(t)$. While this sounds very abstract at first, a simple example shows that this need not be mysterious. Consider a system such that the output signal is the square of the input signal. The mathematical description of this system is simply

$$y(t) = [x(t)]^2 \tag{1.2}$$

which says that at each time instant the value of the output is equal to the square of the input signal value at that same time. Such a system would logically be termed a "squarer system." Figure 1-4 shows the output signal of the squarer for the input of Fig. 1-1. As would be expected from the properties of the squaring operation, we see that the output signal is always nonnegative and the larger signal values are emphasized relative to the smaller signal values.

The squarer system defined by (1.2) is a simple example of a ***continuous-time system*** (i.e., a system whose input and output are continuous-time signals). Can we build a physical system that acts like the squarer system? The answer is yes; the system of (1.2) can be approximated through appropriate connections of electronic circuits. On the other hand, if the input and output of the system are both discrete-time signals (sequences of numbers) related by

$$y[n] = (x[n])^2 \tag{1.3}$$

then the system would be a ***discrete-time system***. The implementation of the discrete-time squarer system would be trivial given a digital computer; one simply multiplies each discrete signal value by itself.

In thinking and writing about systems, it is often useful to have a visual representation of the system. For this purpose, engineers use ***block diagrams*** to represent operations performed in an implementation of a system and to show the interrelations among the many signals that may exist in an implementation of a complex system. An example of the general form of a block diagram is shown in Fig. 1-5. What this diagram shows is simply that the signal $y(t)$ is obtained from the signal $x(t)$ by the operation $\mathcal{T}\{\ \}$.

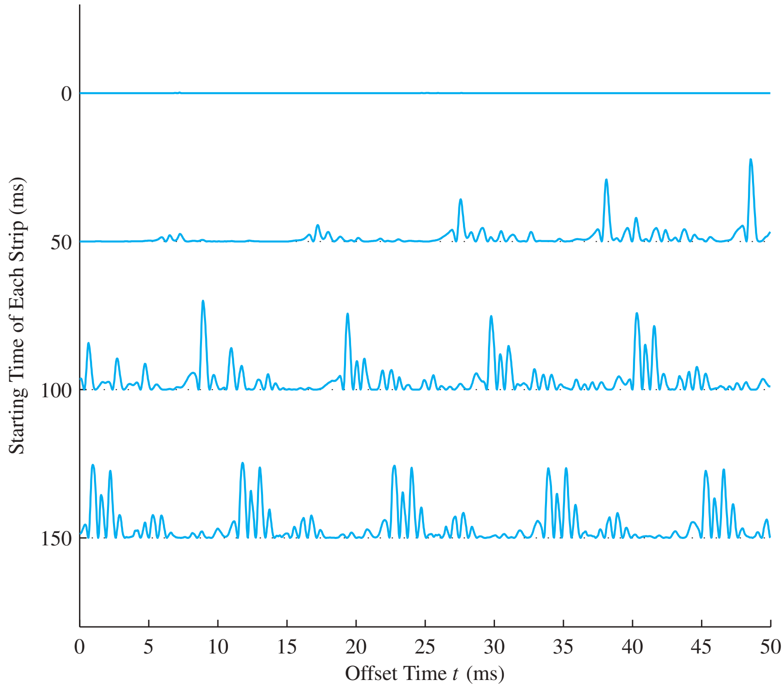

Figure 1-4 Output of a squarer system for the speech signal input of Fig. 1-1. The squarer system is defined by the equation $y(t) = [x(t)]^2$.

A specific example of a system was suggested earlier when we discussed the sampling relationship between continuous-time signals and discrete-time signals. A ***sampler*** is defined as a system whose input is a continuous-time signal $x(t)$ and whose output is the corresponding sequence of samples, defined by the equation

$$x[n] = x(nT_s) \tag{1.4}$$

which simply states that the sampler "takes an instantaneous snapshot" of the continuous-time input signal once every T_s s.[1] Thus, the operation of sampling fits our definition of a system, and it can be represented by the block diagram in Fig. 1-6. Often we will refer to the sampler system as an "ideal continuous-to-discrete converter" or ***ideal C-to-D converter***. In this case, as in the case of the squarer, the name that we give to the system is really just a description of what the system does.

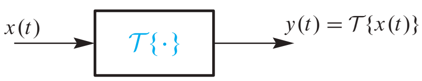

Figure 1-5 Block diagram representation of a continuous-time system.

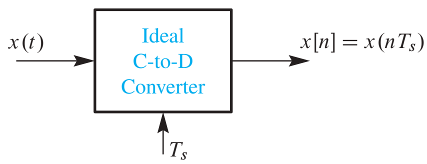

Figure 1-6 Block diagram representation of a sampler.

[1]The units of time in seconds are abbreviated as s.

1-3 Systems as Building Blocks

Block diagrams are useful for representing complex systems in terms of simpler systems, which are more easily understood. For example, Fig. 1-7 shows a block diagram representation of the process of recording and playback of music using MP3 compression. This block diagram breaks the operation down into four subsystems, each of which could be broken down further into smaller subsystems. The first operation is A-to-D (analog-to-digital) conversion to acquire the music waveform in digital form. The A-to-D system is a physical approximation to the ideal C-to-D converter defined in (1.4). An A-to-D converter produces finite-precision numbers as samples of the input signal (quantized to a limited number of bits), while the ideal C-to-D converter produces samples with infinite precision. For the high-accuracy A-to-D converters used in precision audio systems, the difference between an A-to-D converter and our idealized C-to-D converter is slight, but the distinction is very important—only finite-precision quantized sample values can be stored in digital memory of finite size.

Figure 1-7 shows that the output of the A-to-D converter is the input to a system that compresses the numbers $x[n]$ into a much smaller bit stream using the MP3 method. This is a complex process, but for our purposes it is sufficient to show it as a single operation. The output is a compressed digital representation that may be efficiently stored as data on a server or transmitted to a user. Once another user has the compressed data file, the MP3 compression must be reversed in order to listen to the audio signal. Since MP3 is a "lossy" compression scheme, the signal synthesized by the MP3 decoder is only an approximation to the original. The value of MP3 is that this approximation is audibly indistinguishable from the original because the MP3 encoding method exploits aspects of human hearing that render certain coding errors inaudible. Once the music waveform is reconstituted in digital form as $\hat{x}[n]$, the last block does the conversion of the signal from discrete-time form to continuous-time (acoustic) form using a system called a D-to-A (digital-to-analog) converter. This system takes finite-precision binary numbers in sequence and fills in a continuous-time function between the samples. The resulting continuous-time electrical signal could then be fed to other systems, such as amplifiers, loudspeakers, and headphones, for conversion to sound. In Chapter 4, we will discuss the ideal D-to-C converter, which is an idealization of the physical device called an D-to-A converter.

Systems like MP3 audio are all around us. For example, digital cameras use JPEG encoding to reduce digital image file sizes prior to storage, and JPEG decoding to view pictures. Most of the time we do not need to think about how such systems work, but this example illustrates the value of thinking about a complex system in a hierarchical form.

Figure 1-7 Simplified block diagram for MP3 audio compression and playback system.

In this way, we can first understand the individual parts, then the relationship among the parts, and finally the whole system. By looking at the MP3 audio system in this manner, we can discuss two things. First of all, the conversion from continuous-time to discrete-time and back to continuous-time can be considered separately from the other parts of the system. The effect of connecting these blocks to the system is then relatively easy to understand because they provide the input and output interface to real audio signals. Secondly, details of some parts can be hidden and left to experts who, for example, can develop more detailed breakdowns of the MP3 encoder and decoder subsystems. In fact, those systems involve many signal processing operations, and it is possible to specify their operations by connecting several canonical DSP blocks that we will study in this text.

1-4 The Next Step

The MP3 audio coding system is a good example of a relatively complicated discrete-time system. Buried inside the blocks of Fig. 1-7 are many discrete-time subsystems and signals. While we do not promise to explain all the details of MP3 coders or any other complex system, we do hope to establish the foundations for the understanding of discrete- and continuous-time signals and systems so that this knowledge can be applied to understanding components of more complicated systems. In Chapter 2, we will start at a basic mathematical level and show how the well-known sine and cosine functions from trigonometry play a fundamental role in signal and system theory. Next, we show how complex numbers can simplify the algebra of trigonometric functions. Subsequent chapters introduce the concept of the frequency spectrum of a signal and the concept of filtering with a linear time-invariant system. By the end of the book, if you have diligently worked the problems, experienced the demonstrations, and done the laboratory exercises on the Companion Website (which are marked with icons), you will be rewarded with a solid understanding of many of the key concepts underlying much of modern signal processing technology.

NOTE

Companion Website has many labs, demonstrations and homework problems with solutions

CHAPTER

2

Sinusoids

We begin our discussion by introducing a general class of signals that are commonly called *cosine signals* or, equivalently, *sine signals*, which are also commonly referred to as cosine or sine *waves*, particularly when speaking about acoustic or electrical signals. Collectively, such signals are called ***sinusoidal signals*** or, more concisely, ***sinusoids***. Sinusoidal signals are the basic building blocks in the theory of signals and systems, and it is important to become familiar with their properties. The most general mathematical formula for a sinusoid is

$$x(t) = A\cos(\omega_0 t + \varphi) \tag{2.1}$$

where $\cos(\cdot)$ denotes the cosine function that is familiar from the study of trigonometry. When defining a continuous-time signal, we typically use a function whose independent variable is t, a continuous real variable that represents time. From (2.1) it follows that $x(t)$ is a mathematical function in which the angle (or argument) of the cosine function is, in turn, a function of the variable t. Since we normally think of time as increasing uniformly, the angle of the cosine function likewise increases in proportion to time. The parameters A, ω_0, and φ are fixed numbers for a particular cosine signal. Specifically, A is called the ***amplitude***, ω_0 the ***radian frequency***, and φ the ***phase*** of the cosine signal.

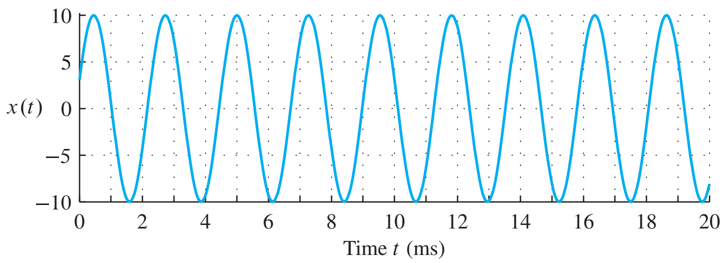

Figure 2-1 Sinusoidal signal generated from the formula: $x(t) = 10\cos(2\pi(440)t - 0.4\pi)$.

Figure 2-1 shows a plot of the continuous-time sinusoid

$$x(t) = 10\cos(2\pi(440)t - 0.4\pi)$$

where $A = 10$, $\omega_0 = 2\pi(440)$, and $\varphi = -0.4\pi$ in (2.1). Note that $x(t)$ oscillates between A and $-A$, and repeats the same pattern of oscillations every $1/440 = 0.00227$ s (approximately). This time interval is called the ***period*** of the sinusoid. We will show later in this chapter that most features of the sinusoidal waveform are directly dependent on the choice of the parameters A, ω_0, and φ.

2-1 Tuning-Fork Experiment

One of the reasons that cosine waves are so important is that many physical systems generate signals that can be modeled (i.e., represented mathematically) as sine or cosine functions versus time. Among the most prominent of these are signals that are audible to humans. The tones or notes produced by musical instruments are perceived as different pitches. Although it is an oversimplification to equate notes to sinusoids and pitch to frequency, the mathematics of sinusoids is an essential first step to understanding complicated sound signals and their perception by humans.

To provide some motivation for our study of sinusoids, we will begin by considering a very simple and familiar system for generating a sinusoidal signal. This system is a *tuning fork*, an example of which is shown in Fig. 2-2. When struck sharply, the tines of the tuning fork vibrate and emit a “pure” tone. This tone has a single frequency, which is usually stamped on the tuning fork. It is common to find “A–440” tuning forks, because 440 hertz (Hz) is the frequency of A above middle C on a musical scale, and is often used as the reference note for tuning a piano and other musical instruments. If you can obtain a tuning fork, perform the following experiment which is shown in a movie on the Companion Website.

DEMO
Tuning Fork

> Strike the tuning fork against your knee, and then hold it close to your ear. You should hear a distinct “hum” at the frequency designated for the tuning fork. The sound will persist for a rather long time if you have struck the tuning fork properly; however, it is easy to do this experiment incorrectly. If you hit the tuning fork sharply on a hard surface such as a table, you will hear a high pitched metallic “ting” sound. This is *not* the characteristic sound