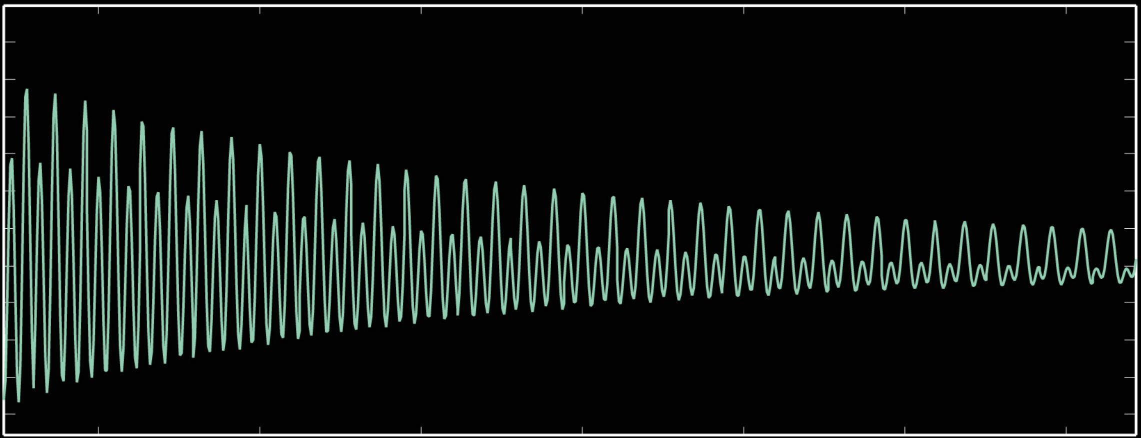
Structure and Interpretation of

Signals and Systems

Edward Ashford Lee
Pravin Varaiya

UC Berkeley



Second Edition
http://LeeVaraiya.org

Second Edition, Version 2.02

ISBN 978-0-578-07719-2

Please cite this book as:

E. A. Lee and P. Varaiya,
Structure and Interpretation of Signals and Systems,
Second Edition, LeeVaraiya.org, 2011.

First Edition was printed by:

Addison-Wesley, ISBN 0-201-74551-8, 2003, Pearson Education, Inc.

Contents

Preface

Signals convey information. Systems transform signals. This book introduces the mathematical models used to design and understand both. It is intended for students interested in developing a deep understanding of how to digitally create and manipulate signals to measure and control the physical world and to enhance human experience and communication.

The discipline known as "signals and systems" is rooted in the intellectual tradition of electrical engineering (EE). This tradition, however, has evolved in unexpected ways. EE has lost its tight coupling with the "electrical." So although many of the techniques introduced in this book were first developed to analyze circuits, today they are widely applied in information processing, system biology, mechanical engineering, finance, and many other disciplines.

This book approaches signals and systems from a computational point of view. A more traditional introduction to signals and systems would be biased towards the historic application, analysis and design of circuits. It would focus almost exclusively on linear time-invariant systems, and would develop continuous-time models first, with discrete-time models then treated as an advanced topic.

The approach in this book benefits students by showing from the start that the methods of signals and systems are applicable to software systems, and most interestingly, to systems that mix computers with physical devices and processes, including mechanical control systems, biological systems, chemical processes, transportation systems, and financial systems. Such systems have become pervasive, and profoundly affect our daily lives.

The shift away from circuits implies some changes in the way the methodology of signals and systems is presented. While it is still true that a voltage that varies over time is a signal, so is a packet sequence on a network. This text defines *signals* to cover both. While it is still true that an RLC circuit is a system, so is a computer program for decoding Internet audio. This text defines *systems* to cover both. While for some systems the state is still captured adequately by variables in a differential equation, for many it is now the values in registers and memory of a computer. This text defines *state* to cover both.

The fundamental limits also change. Although we still face thermal noise and the speed of light, we are likely to encounter other limits–such as complexity, computability, chaos, and, most commonly, limits imposed by other human constructions–before we get to these. The limitations imposed, for example, when transporting voice signals over the Internet, are not primarily physical limitations. They are instead limitations arising from the design and implementation of the Internet, and from the fact that transporting voice was never one of the original intentions of the design. Similarly, computer-based audio systems face latency and jitter imposed by an operating system designed to time share scarce computing resources among data processing tasks. This text focuses on composition of systems so that the limits imposed by one system on another can be understood.

The mathematical basis for the discipline also changes with this new emphasis. The mathematical foundations of circuit analysis are calculus and differential equations. Although we still use calculus and differential equations, we frequently need discrete math, set theory, and mathematical logic. Whereas the mathematics of calculus and differential equations evolved to describe the physical world, the world we face as system designers often has nonphysical properties that are not such a good match for this mathematics. This text bases the entire study on a highly adaptable formalism rooted in elementary set theory.

Despite these fundamental changes in the medium with which we operate, the methodology of signals and systems remains robust and powerful. It is the methodology, not the medium, that defines the field.

The book is based on a course at Berkeley required of all majors in Electrical Engineering and Computer Sciences (EECS). The experience developing the course is reflected in certain distinguished features of this book. First, no background in electrical engineering or computer science is assumed. Readers should have some exposure to calculus, elementary set theory, series, first order linear differential equations, trigonometry, and elementary complex numbers. The appendices review set theory and complex numbers, so this background can be made up students.

Approach

This book is about mathematical modeling and analysis of signals and systems, applications of these methods, and the connection between mathematical models and computational realizations. We develop three themes. The first theme is the use of sets and functions as a universal language to describe diverse signals and systems. Signals—voice, images, bit sequences—are represented as functions with an appropriate domain and range. Systems are represented as functions whose domain and range are themselves sets of signals. Thus, for example, an Internet voice signal is represented as a function that maps voice-like signals into sequences of packets.

The second theme is that complex systems are constructed by connecting simpler subsystems in standard ways—cascade, parallel, feedback. The connections determine the behavior of the interconnected system from the behaviors of component subsystems. The connections place consistency requirements on the input and output signals of the systems being connected.

Our third theme is to relate the declarative view (mathematical, "what is") with the imperative view (procedural, "how to"). That is, we associate mathematical analysis of systems with realizations of these systems. This is the heart of engineering. When electrical engineering was entirely about circuits, this was relatively easy, because it was the physics of the circuits that was being described by the mathematics. Today we have to somehow associate the mathematical analysis with very different realizations of the systems, most especially software. We do this association through the study of state machines, and through the consideration of many real-world signals, which, unlike their mathematical abstractions, have little discernable declarative structure. Speech signals, for instance, are far more interesting than sinusoids, and yet many signals and systems textbooks talk only about sinusoids.

Content

We begin in Chapter 1 by describing signals as functions, focusing on characterizing the domain and the range for familiar signals that humans perceive, such as sound, images, video, trajectories of vehicles, as well as signals typically used by machines to store or manipulate information, such as sequences of words or bits.

Systems, also introduced in Chapter 1, are described as functions, but now the domain and the range are themselves sets of signals. Systems can be connected to form a more complex system, and the function describing these more complex systems is a composition of functions describing the component systems.

Chapter 2 focuses on how to define the functions that we use to model both signals and systems. It distinguishes declarative definitions (assertions of what a signal or system is) from imperative ones (descriptions of how a signal is produced or processed by a system).

The imperative approach is further developed in Chapter 3 using the notion of state, the state transition function, and the output function, all in the context of finite state machines. In Chapter 4, state machines are composed in various ways (cascade, parallel, and feedback) to make more interesting systems. Applications to feedback control illustrate the power of the state machine model.

In Chapter 5, time-based systems are studied, first with discrete-time systems (which have simpler mathematics), and then with continuous-time systems. We introduce the notion of a state machine and define linear time-invariant (LTI) systems as state machines with linear state transition and output functions and zero initial state. The input-output behavior of these systems is fully characterized by their impulse response.

Chapter 7 introduces frequency decomposition of signals, Chapter 8 introduces frequency response of LTI systems, and Chapter 9 brings the two together by discussing filtering. The approach is to present frequency domain concepts as a complementary toolset, different from that of state machines, and much more powerful when applicable. Frequency decomposition of signals is motivated first using psychoacoustics, and gradually developed until all four Fourier transforms (the Fourier series, the Fourier transform, the discrete-time Fourier transform, and the discrete Fourier transform) have been described. We linger on the first of these, the Fourier series, since it is conceptually the easiest, and then more quickly present the others as generalizations of the Fourier series. LTI systems yield best to frequency-domain analysis because of the property that complex exponentials are eigenfunctions (the output is a scaled version of the input). Consequently, they are fully characterized by their frequency response—the main reason that frequency domain methods are important in the analysis of filters and feedback control.

Chapter 10 covers classical Fourier transform material such as properties of the four Fourier transforms and transforms of basic signals. Chapter 11 applies frequency domain methods to a study of sampling and aliasing.

Chapters 12, 13 and 14 extend frequency domain techniques to include the Z transform and the Laplace transform. Applications in signal processing and feedback control illustrate the concepts and the utility of the techniques. Mathematically, the Z transform and the Laplace transform are introduced as extensions of the discrete-time and continuous-time Fourier transforms to signals on which Fourier transforms do not work, specifically signals that are not absolutely summable or integrable. Practically, the concern is for systems that are not stable and for systems that consume unbounded amounts of energy. These chapters extend the intuition of previous chapters to cover such systems.

The unified modeling approach in this text is rich enough to describe a wide range of signals and systems, including those based on discrete events and those based on signals in time, both continuous and discrete. The complementary tools of state machines and frequency domain methods permit analysis and implementation of concrete signals and systems. Hybrid systems and modal models offer systematic ways to combine these complementary toolsets. The framework and the tools of this text provide a foundation on which to build later courses on digital systems, embedded systems, communications, signal processing, hybrid systems, and control.

Pedagogical features

This book has a number of highlights that make it well suited as a textbook for an introductory course.

1. "Probing Further" sidebars briefly introduce the reader to interesting extensions of the subject, to applications, and to more advanced material. They serve to indicate directions in which the subject can be explored.

2. "Basics" sidebars offer readers with less mathematical background some basic tools and methods.

3. Appendix A reviews basic set theory and helps establish the notation used throughout the book.

4. Appendix B reviews complex variables, making it unnecessary for students to have much background in this area.

5. Key equations are boxed to emphasize their importance. They can serve as the places to pause in a quick reading. In the index, the page numbers where key terms are defined are shown in bold.

6. The exercises at the end of each chapter are annotated with the letters E, T, or C to distinguish those exercises that are mechanical (E for excercise) from those requiring a plan of attack (T for thought) and those that generally have more than one reasonable answer (C for conceptualization).

Notation

The notation in this text is unusual when compared to standard texts on signals and systems. We explain our reasons for this as follows:

Domains and ranges. It is common in signals and systems texts to use the form of the argument of a function to define its domain. For example, $x(n)$ is a discrete-time signal, while $x(t)$ is a continuous-time signal; $X(j\omega)$ is the continuous-time Fourier transform and $X(e^{j\omega})$ is the discrete-time Fourier transform. This leads to apparent nonsense like $x(n) = x(nT)$ to define sampling, or to confusion like $X(j\omega) \neq X(e^{j\omega})$ even when $j\omega = e^{j\omega}$.

We treat the domain of a function as part of its definition. Thus a discrete-time, real-valued signal is a function $x : \mathbb{Z} \to \mathbb{R}$, which maps integers to real numbers. Its discrete-time Fourier transform (DTFT) is a function $X : \mathbb{R} \to \mathbb{C}$, which maps real numbers into complex numbers. The DTFT is found using a function whose domain and range are sets of functions,

$$DTFT : [\mathbb{Z} \to \mathbb{R}] \to [\mathbb{R} \to \mathbb{C}].$$

This function maps functions of the form $x : \mathbb{Z} \to \mathbb{R}$ into functions of the form $X : \mathbb{R} \to \mathbb{C}$. The notation $[\mathbb{Z} \to \mathbb{R}]$ means the set of all functions mapping integers into real numbers. Then we can unambiguously write $X = DTFT(x)$.

Functions as values. Most texts call the expression $x(t)$ a function. A better interpretation is that $x(t)$ is an element in the range of the function x. The difficulty with the former interpretation becomes obvious when talking about systems. Many texts pay lip service to the notion that a system is a function by introducing a notation like $y(t) = T(x(t))$. This makes it seem that T acts on the value $x(t)$ rather than on the entire function x.

Our notation includes set of functions, allowing systems to be defined as functions with such sets as the domain and range. Continuous-time convolution, for example, becomes

$$Convolution\colon [\mathbb{R} \to \mathbb{R}] \times [\mathbb{R} \to \mathbb{R}] \to [\mathbb{R} \to \mathbb{R}].$$

We then introduce the notation $*$ as a shorthand,

$$y = x * h = Convolution(x, h),$$

and define the convolution function by

$$\forall\, t \in \mathbb{R}, \quad y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau) y(t - \tau) d\tau.$$

Note the careful parenthesization. The more traditional notation, $y(t) = x(t) * h(t)$, would seem to imply that $y(t - T) = x(t - T) * h(t - T)$. But it is not so! Such notation undermines a student's confidence in algebra, since substitution of a value for t does not work!

A major advantage of our notation is that it easily extends beyond LTI systems to the sorts of systems that inevitably arise in any real world application, such as mixtures of discrete event and continuous-time systems.

Names of functions. We use long names for functions and variables when they have a concrete interpretation. Thus, instead of x we might use *Sound*. This follows a long-standing tradition in software, where readability is considerably improved by long names. By giving us a much richer set of names to use, this helps us avoid some of the preceding pitfalls. For example, to define sampling of an audio signal, we might write

$$SampledSound = Sampler_T(Sound).$$

It also helps bridge the gap between realizations of systems (which are often software) and their mathematical models. How to manage and understand this gap is a major theme of our approach.

How to use this book

At Berkeley, the first 11 chapters of this book are covered in a 15-week, one-semester course. Even though it leaves Laplace transforms, Z transforms, and feedback control systems to a follow-up course, it remains a fairly intense experience. Each week consists of three 50-minute lectures, a one-hour problem session, and one three-hour laboratory. The lectures and problem sessions are conducted by a faculty member while the laboratory is led by teaching assistants, who are usually graduate students, but are also often talented juniors or seniors.

We have developed laboratory components based on MATLAB and Simulink, and a separate set based on LabVIEW. In both cases, then lab content is closely coordinated with the lectures. The text does not offer a tutorial on LabVIEW, MATLAB, or Simulink, although the labs include enough material so that, combined with on-line help, they are sufficient. Some examples in the text and some exercises at the ends of the chapters depend on MathScript, the mathematical expression language used by both MATLAB and LabVIEW.

At Berkeley, this course is taken by all electrical engineering and computer science students, and is followed by a more traditional signals and systems course. That course covers the material in the last three chapters plus applications of frequency-domain methods to communications systems. The follow-up course is not taken by most computer science students. In a program that is more purely electrical and computer engineering than ours, a better approach might be to spend two quarters or two semesters on the material in this text, since the unity of notation and approach would be better than having two disjoint courses, the introductory one using a modern approach, and the follow-up course using a traditional one.

Acknowledgements

Many people have contributed to the content of this book. Dave Messerschmitt conceptualized the first version of the course on which the book is based, and later committed considerable departmental resources to the development of the course while he was chair of the EECS department at Berkeley. Randy Katz, Richard Newton, and Shankar Sastry continued to invest considerable resources in the course when they each took over as chair, and backed our efforts to establish the course as a cornerstone of our undergraduate

curriculum. This took considerable courage, since the conceptual approach of the course was largely unproven.

Tom Henzinger probably had more intellectual influence over the approach than any other individual. The view of state machines, of composition of systems, and of hybrid systems owe a great deal to Tom. Gerard Berry also contributed a great deal to our way of presenting synchronous composition.

We were impressed by the approach of Harold Abelson and and Gerald Jay Sussman, in *Structure and Interpretation of Computer Programs* (MIT Press, 1996), who confronted a similar transition in their discipline. The title of our book shows their influence. Jim McLellan, Ron Shafer, and Mark Yoder influenced this book through their pioneering departure from tradition in signals and systems, *DSP First—A Multimedia Approach* (Prentice-Hall, 1998). Ken Steiglitz greatly influenced the labs with his inspirational book, *A DSP Primer: With Applications to Digital Audio and Computer Music* (Addison-Wesley, 1996). Babak Ayazifar, with his visionary treatment of the course, has significantly influenced more recent versions of the book.

A number of people have been involved in the media applications, examples, the laboratory development, and the web content associated with the book. These include Brian Evans and Ferenc Kovac. We also owe gratitude for the superb technical support from Christopher Brooks. Jie Liu contributed sticky masses example to the hybrid systems chapter, and Yuhong Xiong contributed the technical stock trading example. Other examples and ideas were contributed by Steve Neuendorffer, Cory Sharp, and Tunc Simsek.

Over several years, students at Berkeley have taken the course that provided the impetus for this book. They used successive versions of the book and the Web content. Their varied response to the course helped us define the structure of the book and the level of discussion. The course is taught with the help of undergraduate teaching assistants. Their comments helped shape the laboratory material.

Parts of this book were reviewed by more than 30 faculty members around the country. Their criticisms helped us correct defects and inconsistencies in earlier versions. Of course, we alone are responsible for the opinions expressed in the book, and the errors that remain. We especially thank: Jack Kurzweil, San Jose State University; Lee Swindlehurst, Brigham Young University; Malur K. Sundareshan, University of Arizona; Stéphane Lafortune, University of Michigan; Ronald E. Nelson, Arkansas Tech University; Ravi Mazumdar, Purdue University; Ratnesh Kumar, University of Kentucky; Rahul Singh, San Diego State University; Paul Neudorfer, Seattle University; R. Mark Nelms,