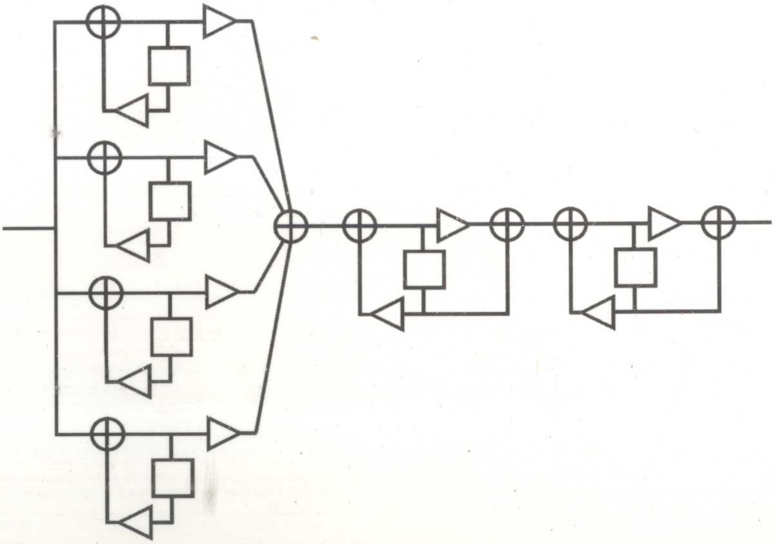
电子工程系列丛书（影印版）

INTRODUCTION TO Signal Processing

信号处理导论



Sophocles J. Orfanidis

清华大学出版社 . PRENTICE HALL
http://www.tup.tsinghua.edu.cn

INTRODUCTION TO
Signal Processing

信号处理导论

Sophocles J. Orfanidis
Rutgers University

清 华 大 学 出 版 社
Prentice Hall International, Inc.

(京)新登字 158 号

Introduction to signal processing/Sophocles J. Orfanidis

北京市版权局著作权合同登记号: 01-98-2058

图书在版编目(CIP)数据

信号处理导论:英文/奥法尼德斯(Orfanidis, S.J.)著. -影印版. -北京:清华大学出版社,1998.9
(电子工程系列丛书)
ISBN 7-302-03059-6

Ⅰ.信… Ⅱ.奥… Ⅲ.数字信号-信号处理-英文 Ⅳ.TN911.72

中国版本图书馆 CIP 数据核字(98)第 21434 号

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京发行所
开 本: 850×1168 1/32 印张: 25.5
版 次: 1999 年 1 月第 1 版 2001 年 3 月第 3 次印刷
书 号: ISBN 7-302-03059-6/TN·99
印 数: 7501~10500
定 价: 39.00 元

出 版 前 言

清华大学出版社与 Prentice Hall 出版公司合作推出的“大学计算机教育丛书(影印版)”和“ATM 与 B-ISDN 技术丛书(影印版)”受到了广大读者的欢迎。很多读者通过电话、信函、电子邮件对我们的工作以积极评价,并提出了不少极好的建议,令我们感动和鼓舞。我们除了继续努力完善上述两套丛书以外,还将努力拓宽影印图书的专业范围,以更好地满足读者的需要。

电子工程是信息科学的基础,高等学校新的教学要求指出,计算机专业和电子学专业的学生应相互学习渗透到彼此的专业领域,拓宽知识面,以适应信息技术飞速发展的时代。培养通晓相关专业领域知识的人才,成为面向新世纪的理工科教育的迫切要求。为此,我们挑选了与信息科学、电子学有关的国外优秀著作,组成电子工程系列丛书(影印版),奉献给国内读者。我们希望这套新的丛书能为国内的大专院校师生和科研单位的工作人员提供新的知识和营养,也衷心期待着读者对我们一如既往的支持。

清华大学出版社
Prentice Hall 出版公司
1998.12

Preface

This book provides an applications-oriented introduction to digital signal processing written primarily for electrical engineering undergraduates. Practicing engineers and graduate students may also find it useful as a first text on the subject.

Digital signal processing is everywhere. Today's college students hear "DSP" all the time in their everyday life—from their CD players, to their electronic music synthesizers, to the sound cards in their PCs. They hear all about "DSP chips", "oversampling digital filters", "1-bit A/D and D/A converters", "wavetable sound synthesis", "audio effects processors", "all-digital audio studios". By the time they reach their junior year, they are already very eager to learn more about DSP.

Approach

The learning of DSP can be made into a rewarding, interesting, and fun experience for the student by weaving into the material several applications, such as the above, that serve as vehicles for teaching the basic DSP concepts, while generating and maintaining student interest. This has been the guiding philosophy and objective in writing this text. As a result, the book's emphasis is more on signal processing than discrete-time system theory, although the basic principles of the latter are adequately covered.

The book teaches by example and takes a hands-on practical approach that emphasizes the *algorithmic, computational, and programming* aspects of DSP. It contains a large number of worked examples, computer simulations and applications, and several C and MATLAB functions for implementing various DSP operations. The practical slant of the book makes the concepts more concrete.

Use

The book may be used at the *junior or senior* level. It is based on a junior-level DSP course that I have taught at Rutgers since 1988. The assumed background is only a first course on linear systems. Sections marked with an asterisk (*) are more appropriate for a second or senior elective course on DSP. The rest can be covered at the junior level. The included computer experiments can form the basis of an accompanying DSP lab course, as is done at Rutgers.

A solutions manual, which also contains the results of the computer experiments, is available from the publisher. The C and MATLAB functions may be obtained via anonymous FTP from the Internet site `ece.rutgers.edu` in the directory `/pub/sjo` or by pointing a Web browser to the book's WWW home page at the URL `ftp://ece.rutgers.edu/pub/sjo/intro2sp.html`, or, `http://www.ece.rutgers.edu/~orfanidi/intro2sp.html`.

Contents and Highlights

Chapters 1 and 2 contain a discussion of the two key DSP concepts of *sampling and quantization*. The first part of Chapter 1 covers the basic issues of sampling, aliasing, and *analog reconstruction* at a level appropriate for juniors. The second part is more advanced and discusses the practical issues of choosing and defining specifications for *antialiasing prefilters and anti-image postfilters*.

Chapter 2 discusses the *quantization process* and some practical implementations of A/D and D/A converters, such as the conversion algorithm for bipolar two's complement successive approximation converters. The standard model of quantization noise is presented, as well as the techniques of *oversampling, noise shaping, and dithering*. The tradeoff between oversampling ratio and savings in bits is derived. This material is continued in Section 12.7 where the implementation and operation of delta-sigma noise shaping quantizers is considered.

Chapter 3 serves as a review of basic *discrete-time systems* concepts, such as linearity, time-invariance, impulse response, convolution, FIR and IIR filters, causality, and stability. It can be covered quickly as most of this material is assumed known from a prerequisite linear systems course.

Chapter 4 focuses on FIR filters and its purpose is to introduce two basic signal processing methods: *block-by-block* processing and *sample-by-sample* processing. In the block processing part, we discuss various approaches to convolution, transient and steady-state behavior of filters, and real-time processing on a block-by-block basis using the overlap-add method and its software implementation. This is further discussed in Section 9.9 using the FFT.

In the sample processing part, we introduce the basic building blocks of filters: adders, multipliers, and delays. We discuss *block diagrams* for FIR filters and their time-domain operation on a sample-by-sample basis. We put a lot of emphasis on the concept of *sample processing algorithm*, which is the repetitive series of computations that must be carried out on each input sample.

We discuss the concept of *circular buffers* and their use in implementing delays and FIR filters. We present a systematic treatment of the subject and carry it on to the remainder of the book. The use of circular delay-line buffers is old, dating back at least 25 years with its application to computer music. However, it has not been treated systematically in DSP texts. It has acquired a new relevance because all modern DSP chips use it to minimize the number of hardware instructions.

Chapter 5 covers the basics of z-transforms. We emphasize the z-domain view of causality, stability, and frequency spectrum. Much of this material may be known from an earlier linear system course.

Chapter 6 shows the equivalence of various ways of characterizing a linear filter and illustrates their use by example. It also discusses topics such as sinusoidal and steady-state responses, time constants of filters, simple pole/zero designs of first-and second-order filters as well as comb and notch filters. The issues of inverse filtering and causality are also considered.

Chapter 7 develops the standard *filter realizations* of canonical, direct, and cascade forms, and their implementation with *linear and circular* buffers. Quantization effects are briefly discussed.

Chapter 8 presents three DSP application areas. The first is on digital *waveform generation*, with particular emphasis on *wavetable generators*. The second is on *digital audio effects*, such as flanging, chorusing, reverberation, multitap delays, and dynamics processors, such as compressors, limiters, expanders, and gates. These areas were chosen for their appeal to undergraduates and because they provide concrete illustrations of the use of delays, circular buffers, and filtering concepts in the context of audio signal processing.

The third area is on *noise reduction/signal enhancement*, which is one of the most important applications of DSP and is of interest to practicing engineers and scientists who remove noise from data on a routine basis. Here, we develop the basic principles for designing noise reduction and signal enhancement filters both in the frequency and time domains. We discuss the design and circular buffer implementation of *notch and comb* filters for removing periodic interference, enhancing periodic signals, signal averaging, and separating the luminance and chrominance components in digital color TV systems. We also discuss *Savitzky-Golay* filters for data smoothing and differentiation.

Chapter 9 covers *DFT/FFT algorithms*. The first part emphasizes the issues of spectral analysis, frequency resolution, windowing, and leakage. The second part discusses the computational aspects of the DFT and some of its pitfalls, the difference between physical and computational frequency resolution, the FFT, and fast convolution.

Chapter 10 covers *FIR filter design* using the window method, with particular emphasis on the *Kaiser window*. We also discuss the use of the Kaiser window in spectral analysis.

Chapter 11 discusses *IIR filter design* using the bilinear transformation based on Butterworth and Chebyshev filters. By way of introducing the bilinear transformation, we show how to design practical second-order digital audio *parametric equalizer* filters having prescribed widths, center frequencies, and gains. We also discuss the design of periodic notch and comb filters with prescribed widths.

In the two filter design chapters, we have chosen to present only a few design methods that are simple enough for our intended level of presentation and effective enough to be of practical use.

Chapter 12 discusses *interpolation, decimation, oversampling DSP systems, sample rate converters, and delta-sigma quantizers*. We discuss the use of oversampling for alleviating the need for high quality analog prefilters and postfilters. We present several practical design examples of interpolation filters, including *polyphase and multistage* designs. We consider the design of sample rate converters and study the operation of oversampled delta-sigma quantizers by simulation. This material is too advanced for juniors but not seniors. All undergraduates, however, have a strong interest in it because of its use in digital audio systems such as CD and DAT players.

The Appendix has four parts: (a) a review section on *random signals*; (b) a discussion of random number generators, including uniform, Gaussian, low frequency, and $1/f$ noise generators; (c) C functions for performing the complex arithmetic in the DFT routines; (d) listings of MATLAB functions.

Paths

Several course paths are possible through the text depending on the desired level of presentation. For example, in the 14-week junior course at Rutgers we cover Sections 1.1–1.4, 2.1–2.4, Chapters 3–7, Sections 8.1–8.2, Chapter 9, and Sections 10.1–10.2 and 11.1–11.4. One may omit certain of these sections and/or add others depending on the available time and student interest and background. In a second DSP course at the senior year, one may add Sections 1.5–1.7, 2.5, 8.3, 11.5–11.6, and Chapter 12. In a graduate course, the entire text can be covered comfortably in one semester.

Acknowledgments

I am indebted to the many generations of students who tried earlier versions of the book and helped me refine it. In particular, I would like to thank Mr. Cem Saraydar for his thorough proofreading of the manuscript. I would like to thank my colleagues Drs. Zoran Gajic, Mark Kahrs, James Kaiser, Dino Lelic, Tom Marshall, Peter Meer, and Nader Moayeri for their feedback and encouragement. I am especially indebted to Dr. James Kaiser for enriching my classes over the past eight years with his inspiring yearly lectures on the Kaiser window. I would like to thank the book's reviewers Drs. A. V. Oppenheim, J. A. Fleming, Y-C. Jenq, W. B. Mikhael, S. J. Reeves, A. Sekey, and J. Weitzen, whose comments helped improve the book. And I would like to thank Rutgers for providing me with a sabbatical leave to finish up the project. I welcome any feedback from readers—it may be sent to orfanidi@ece.rutgers.edu.

Finally, I would like to thank my wife Monica and son John for their love, patience, encouragement, and support.

Sophocles J. Orfanidis

Contents

1

Sampling and Reconstruction

1.1 Introduction

Digital processing of analog signals proceeds in three stages:

1. The analog signal is *digitized*, that is, it is *sampled* and each sample *quantized* to a finite number of bits. This process is called A/D conversion.

2. The digitized samples are processed by a *digital signal processor.*

3. The resulting output samples may be converted back into *analog* form by an analog reconstructor (D/A conversion).

A typical digital signal processing system is shown below.

The digital signal processor can be programmed to perform a variety of signal processing operations, such as filtering, spectrum estimation, and other DSP algorithms. Depending on the speed and computational requirements of the application, the digital signal processor may be realized by a general purpose computer, minicomputer, special purpose DSP chip, or other digital hardware dedicated to performing a particular signal processing task.

The design and implementation of DSP algorithms will be considered in the rest of this text. In the first two chapters we discuss the two key concepts of *sampling* and *quantization*, which are prerequisites to every DSP operation.

1.2 Review of Analog Signals

We begin by reviewing some pertinent topics from analog system theory. An *analog* signal is described by a function of time, say, $x(t)$. The *Fourier transform* $X(\Omega)$ of

$x(t)$ is the *frequency spectrum* of the signal:

$$\boxed{X(\Omega)=\int_{-\infty}^{\infty} x(t)\,e^{-j\Omega t}\,dt} \qquad (1.2.1)$$

where Ω is the radian frequency† in [radians/second]. The ordinary frequency f in [Hertz] or [cycles/sec] is related to Ω by

$$\boxed{\Omega = 2\pi f} \qquad (1.2.2)$$

The physical meaning of $X(\Omega)$ is brought out by the *inverse* Fourier transform, which expresses the arbitrary signal $x(t)$ as a linear superposition of *sinusoids* of different frequencies:

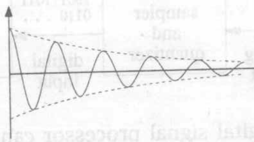
$$\boxed{x(t)=\int_{-\infty}^{\infty} X(\Omega)\,e^{j\Omega t}\,\frac{d\Omega}{2\pi}} \qquad (1.2.3)$$

The relative importance of each sinusoidal component is given by the quantity $X(\Omega)$. The *Laplace transform* is defined by

$$X(s)=\int_{-\infty}^{\infty} x(t)\,e^{-st}\,dt$$

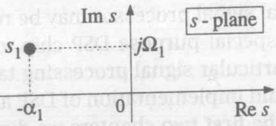
It reduces to the Fourier transform, Eq. (1.2.1), under the substitution $s = j\Omega$. The s-plane pole/zero properties of transforms provide additional insight into the nature of signals. For example, a typical exponentially decaying sinusoid of the form

$$x(t)= e^{-\alpha_1 t}e^{j\Omega_1 t}u(t)= e^{s_1 t}u(t)$$

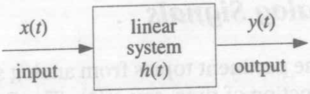


where $s_1 = -\alpha_1 + j\Omega_1$, has Laplace transform

$$X(s)=\frac{1}{s-s_1}$$



with a pole at $s = s_1$, which lies in the left-hand s-plane. Next, consider the response of a *linear system* to an input signal $x(t)$:



†We use the notation Ω to denote the physical frequency in units of [radians/sec], and reserve the notation ω to denote digital frequency in [radians/sample].

The system is characterized completely by the *impulse response* function $h(t)$. The output $y(t)$ is obtained in the time domain by *convolution*:

$$y(t) = \int_{-\infty}^{\infty} h(t - t')x(t')\,dt'$$

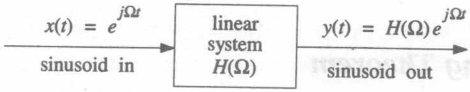
or, in the frequency domain by *multiplication*:

$$Y(\Omega) = H(\Omega)X(\Omega) \tag{1.2.4}$$

where $H(\Omega)$ is the *frequency response* of the system, defined as the Fourier transform of the impulse response $h(t)$:

$$H(\Omega) = \int_{-\infty}^{\infty} h(t)e^{-j\Omega t}\,dt \tag{1.2.5}$$

The *steady-state* sinusoidal response of the filter, defined as its response to sinusoidal inputs, is summarized below:



This figure illustrates the *filtering* action of linear filters, that is, a given frequency component Ω is attenuated (or, magnified) by an amount $H(\Omega)$ by the filter. More precisely, an input sinusoid of frequency Ω will reappear at the output modified in *magnitude* by a factor $|H(\Omega)|$ and shifted in *phase* by an amount $\arg H(\Omega)$:

$$x(t) = e^{j\Omega t} \quad \Rightarrow \quad y(t) = H(\Omega)e^{j\Omega t} = |H(\Omega)|e^{j\Omega t + j\arg H(\Omega)}$$

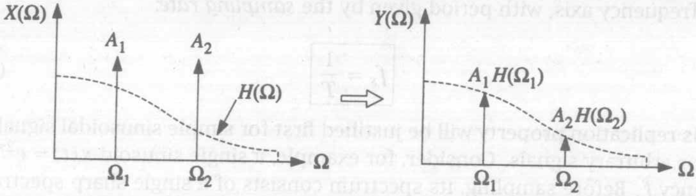
By linear superposition, if the input consists of the sum of two sinusoids of frequencies Ω_1 and Ω_2 and relative amplitudes A_1 and A_2,

$$x(t) = A_1 e^{j\Omega_1 t} + A_2 e^{j\Omega_2 t}$$

then, after filtering, the steady-state output will be

$$y(t) = A_1 H(\Omega_1)e^{j\Omega_1 t} + A_2 H(\Omega_2)e^{j\Omega_2 t}$$

Notice how the filter changes the *relative* amplitudes of the sinusoids, but not their frequencies. The filtering effect may also be seen in the frequency domain using Eq. (1.2.4), as shown below:



The input spectrum $X(\Omega)$ consists of two sharp spectral lines at frequencies Ω_1 and Ω_2, as can be seen by taking the Fourier transform of $x(t)$:

$$X(\Omega) = 2\pi A_1 \delta(\Omega - \Omega_1) + 2\pi A_2 \delta(\Omega - \Omega_2)$$

The corresponding output spectrum $Y(\Omega)$ is obtained from Eq. (1.2.4):

$$\begin{aligned} Y(\Omega) &= H(\Omega) X(\Omega) = H(\Omega)\big(2\pi A_1 \delta(\Omega - \Omega_1) + 2\pi A_2 \delta(\Omega - \Omega_2)\big) \\ &= 2\pi A_1 H(\Omega_1) \delta(\Omega - \Omega_1) + 2\pi A_2 H(\Omega_2) \delta(\Omega - \Omega_2) \end{aligned}$$

What makes the subject of linear filtering useful is that the designer has complete *control* over the shape of the frequency response $H(\Omega)$ of the filter. For example, if the sinusoidal component Ω_1 represents a desired signal and Ω_2 an unwanted interference, then a filter may be designed that lets Ω_1 pass through, while at the same time it filters out the Ω_2 component. Such a filter must have $H(\Omega_1) = 1$ and $H(\Omega_2) = 0$.

1.3 Sampling Theorem

Next, we study the sampling process, illustrated in Fig. 1.3.1, where the analog signal $x(t)$ is *periodically measured* every T seconds. Thus, time is discretized in units of the *sampling interval* T:

$$t = nT, \qquad n = 0, 1, 2, \dots$$

Considering the resulting stream of samples as an *analog* signal, we observe that the sampling process represents a very drastic chopping operation on the original signal $x(t)$, and therefore, it will introduce a lot of spurious *high-frequency* components into the frequency spectrum. Thus, for system design purposes, two questions must be answered:

1. What is the effect of sampling on the original frequency spectrum?
2. How should one choose the sampling interval T?

We will try to answer these questions intuitively, and then more formally using Fourier transforms. We will see that although the sampling process generates high frequency components, these components appear in a very regular fashion, that is, *every* frequency component of the original signal is *periodically replicated* over the entire frequency axis, with period given by the *sampling rate*:

$$\boxed{f_s = \frac{1}{T}} \tag{1.3.1}$$

This replication property will be justified first for simple sinusoidal signals and then for arbitrary signals. Consider, for example, a single sinusoid $x(t) = e^{2\pi jft}$ of frequency f. Before sampling, its spectrum consists of a single sharp spectral line