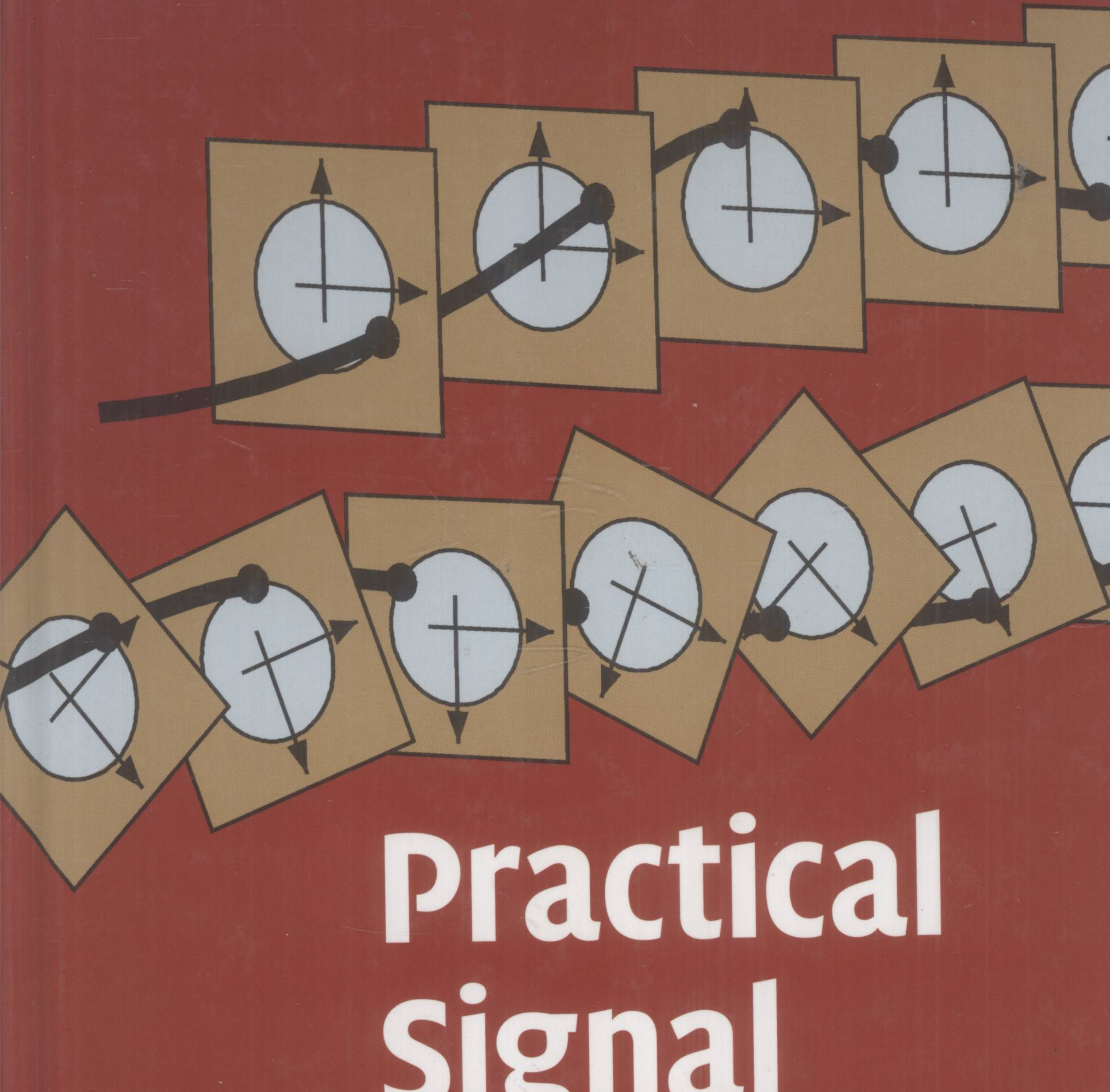

Practical Signal Processing

MARK OWEN

CAMBRIDGE

TN911.72
O97

PRACTICAL SIGNAL PROCESSING

MARK OWEN

E2008000047

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521854788

© Cambridge University Press 2007

This publication is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 2007

Printed in the United Kingdom at the University Press, Cambridge

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

ISBN 978-0-521-85478-8 hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external
or third-party internet websites referred to in this publication, and does not guarantee that any
content on such websites is, or will remain, accurate or appropriate.

PRACTICAL SIGNAL PROCESSING

The principles of signal processing are fundamental to the operation of everyday devices such as digital cameras, mobile telephones and digital audio players. This book introduces the basic theory of digital signal processing, placing a strong emphasis on the use of techniques in real-world applications. The author uses intuitive arguments rather than mathematical ones wherever possible, reinforced by practical examples and diagrams.

The first part of the book covers sampling, quantisation, the Fourier transform, filters, Bayesian methods and numerical considerations. These ideas are then developed in the second part, illustrating how they are used in audio, image, and video processing and compression, and in communications. The book concludes with methods for the efficient implementation of algorithms in hardware and software. Throughout, links between various signal processing techniques are stressed and real-world examples showing the advantages and disadvantages of the different approaches are presented, enabling the reader to choose the best solution to a given problem.

With over 200 illustrations and over 130 exercises (including solutions), this book will appeal to practitioners working in any branch of signal processing, as well as to undergraduate students of electrical and computer engineering.

MARK OWEN received his Ph.D. in Speech Recognition from Cambridge University in 1992, after which he has worked in industry on digital signal processing applications in video, audio and radar. He is currently a freelance consultant in this, and related, fields.

Preface

This is a book you can read in the park, on the beach, at the bus stop – or even in the bath.

The book is in two parts. The first part takes you step-by-step through the fundamental ideas of digital signal processing, while the second part shows how these ideas are used in a wide range of practical situations. My aim is that by the end of the book you will understand many of the signal processing algorithms and techniques that are essential to everyday devices such as digital cameras, modems, digital set-top boxes, mobile telephones and digital audio players. I have used examples drawn from the operation of such devices to help explain points in the text.

You do not need to know any calculus to understand any of the ideas discussed. A basic understanding of trigonometry and of arithmetic on complex numbers is necessary, however; and a very basic knowledge of the principles of electronic circuits is helpful, but by no means essential.

If you are a student, I hope that the approach this book takes will give you a more concrete and more intuitive grasp of the principles of digital signal processing than a purer mathematical treatment would. If you are a practising engineer or programmer with a particular problem to solve, I hope that the book helps you understand the problem and decide on the right way to tackle it. And if you are just interested in the subject for its own sake, I hope you enjoy the book.

There are exercises at the end of each chapter. Some of them you can probably do in your head; for some you might need pencil and paper; and for some you will need to write a short program. Some are slightly more ambitious programming projects. A few ask you to criticise inappropriate solutions to signal processing problems suggested by a hypothetical friend who has clearly not read this book: if you have any friends in this position you can remedy the situation by buying them a copy. Please try the easier exercises and at least think about how you would go about the harder ones; and please don't take your computer into the bath.

试读结束，需要全本PDF请购买 www.ertongbook.com

Acknowledgements

I am grateful to Philip Meyler and staff at Cambridge University Press for their work in editing and producing this book; to Laurence Nicolson, Alex Scott, Alex Selby, Roger Sewell, Mark Wainwright and Jason Wong, who all shaped the book for the better in various ways; and to Laura Doherty and Martin Oldfield, who read early versions of the manuscript and provided many helpful comments.

Contents

Part I

Foundations

试读结束，需要全本PDF请购买 www.ertongbook.com

1
Introduction

1.1 What is a signal?

A *signal* is a varying quantity whose value can be measured and which conveys information.

For example, we can consider temperature to be a signal. It can vary over time, we can measure it using a thermometer, and it conveys information: knowing the temperature outside will inform our decision as to which clothes to wear.

In a digital signal processing system we represent a signal as a sequence of numbers either on a computer or in digital hardware. For example, we could store the temperature at various times of the day as a sequence of numbers in an array on a computer: each number might be a temperature reading in Celsius.

Digital signal processing involves transforming one signal into another signal, represented digitally throughout. The transformation is achieved using simple operations on the numbers representing the signal. For example, we might want to know the average temperature over a day: we could calculate this by adding up the elements in the array of temperature data and dividing the total by the size of the array.

1.2 Domain and range of a signal

Temperature is a function of a single real-valued variable, time: see Figure 1.1. We say that the *domain* of the signal is one-dimensional. Some signals are functions of more than one variable. For example, a black-and-white photograph can be regarded as a signal: the brightness u of a point on the photograph is a function of two variables, the x and y coordinates of the point on the photograph: see Figure 1.2. In this case the domain of the signal is two-dimensional.

In a black-and-white photograph, the brightness of a point on the photograph can be represented as a single real number, and so we call it a real-valued signal, and say that the *range* of the signal is one-dimensional. In a colour photograph,

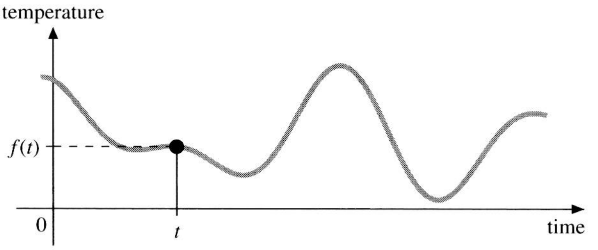

Figure 1.1 A signal with one-dimensional range and one-dimensional domain.

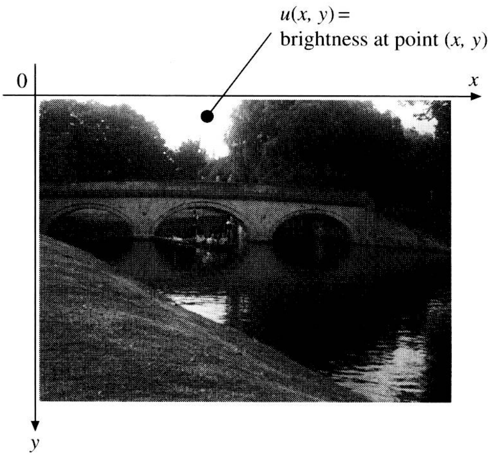

Figure 1.2 A signal with one-dimensional range and two-dimensional domain.

however, one real number will not do. The colour of a point can be expressed as three real numbers, separately giving the amount of red, green and blue that go to make up the colour. In this case, we would say that the range of the signal is three-dimensional.

Now let us consider a colour movie. The colour of a point on the screen with given x and y coordinates can be expressed as three real numbers for the amounts of red, green and blue as described above. However, the picture also changes with time, which adds an extra dimension to the domain of the signal. In total we therefore have three dimensions in the domain (x, y and time) and three in the range (red, green and blue).

1.3 Converting signals from one form to another

A device that converts a signal from one form to another is called a *transducer*. Often the signal on one side of the conversion will be an electrical one.

A loudspeaker is an example of a transducer, in this case converting an electrical signal into a varying air pressure to create a sound. The variation in air pressure is a real-valued function of time. A microphone is a transducer that converts in the opposite direction, from variations in air pressure to variations in an electrical signal.

1.4 Processing signals

Suppose we want to

- modify the amount of bass and treble in an audio signal
- analyse an image to determine what objects are present in it
- compute seasonally adjusted temperature values
- make a photograph sharper or increase its contrast
- measure the pitch of a musical instrument

All of these are examples of signal processing tasks. Later in this book we shall look at how we might go about these kinds of task.

Another common signal processing task is data compression, where we take advantage of special characteristics of a signal to reduce the resources required to store or transmit it. For example, recorded speech often contains long pauses. We can identify these pauses and delete them. To take another example, one frame of a movie is often very similar to the previous one. We can process the signal to find which parts of the picture are changing, and only record those.

Why digital?

Many signal processing tasks can be done using conventional analogue electronics. Our first example above, a tone control which modifies the amount of bass and treble in an audio signal, is particularly simple using analogue technology. If our signal is in the form of a varying voltage, the necessary circuit consists of just a couple of components costing a few pence in total!

Analogue processing systems suffer from several disadvantages, however. Our tone control circuit will be fine as long as we are not too demanding about its performance: if we manufacture several copies of the circuit, the chances are that its characteristics will vary, probably by as much as several per cent from unit to unit. If we needed less variation from unit to unit we could use better-quality components, but that would increase the cost. Alternatively, we could add an adjustment to the circuit so that we can accurately trim each unit to compensate for manufacturing variations, but that would make the assembly process more complicated and so more expensive. The characteristics of analogue systems also tend to drift slowly over time and with temperature.

Any processing which involves storing a lot of information – comparing successive frames of a movie, for example – will almost certainly be complicated and expensive when implemented using analogue circuitry.

Further disadvantages become apparent when we turn to integrated circuits (ICs). It is perfectly possible to make ICs with analogue circuits on them, but processing involving low frequencies, such as the bass part of an audio signal, usually requires physically large components which are difficult to make on an IC. There are also technical difficulties, as well as extra cost, associated with mixing analogue and digital circuitry on a single IC.

Many electronic devices already necessarily incorporate digital circuitry. For example, consider adding a tone control to a CD (compact disc) player which already uses a digital integrated circuit to decode the information stored on the disc. In this case, the cheapest option could easily be to implement the tone control digitally on the same IC: this would add very slightly to the size and hence the cost of the IC, but no extra components would be needed and assembly costs would remain the same.

Once the decision is made to implement the tone control digitally, it becomes very simple to add extra features, such as tone settings optimised for different types of music, all at very little extra cost compared with an analogue implementation.

There are, of course, disadvantages to doing things digitally. As we shall see in later chapters, there are some hazards to avoid when converting a signal between analogue and digital forms. If the processing you want to do is not demanding, and the signal is not already in digital form, the overhead of converting from analogue to digital and back can easily outweigh the advantages of digital processing.

1.5 Notation

Before proceeding further we will make a few remarks on the notation used in this book.

Complex numbers

Sometimes in this book we will be considering signals which take on complex values. They can be thought of as having a two-dimensional range, the two dimensions being the real and imaginary parts of the complex values. They arise more frequently in the intermediate steps in processing a signal than they do naturally as physical quantities (although of course you can think of any two-dimensional quantity as a complex number if you like).

In this book we will write j for the square root of -1. (Engineers often use i to stand for electrical current.) We will write the real part of a complex number z

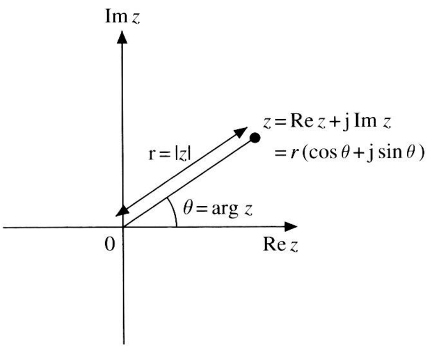

Figure 1.3 A point z in the complex plane can be represented in terms of its real and imaginary parts, or in terms of its magnitude and argument.

as $\operatorname{Re} z$ and the imaginary part as $\operatorname{Im} z$. The *magnitude* of z, $|z|$, is the distance between the point z and the origin of the complex plane; and the *argument* of z, $\arg z$, is the angle between the real axis and the line from the origin to z, measured anticlockwise.

Figure 1.3 shows that we can write a complex number z in terms of its real part $x = \operatorname{Re} z$ and imaginary part $y = \operatorname{Im} z$ as

$$z = x + \mathrm{j}y$$

or in terms of its magnitude $r = |z|$ and argument $\theta = \arg z$:

$$\begin{aligned} z &= r\cos\theta + \mathrm{j}r\sin\theta \\ &= r\mathrm{e}^{\mathrm{j}\theta}. \end{aligned}$$

We will frequently switch between these two representations of complex numbers.

One handy use of complex numbers is to represent angles. An angle θ can be represented by the complex number $z = \cos\theta + \mathrm{j}\sin\theta = \mathrm{e}^{\mathrm{j}\theta}$. The magnitude of z is 1, and so this is a point in the complex plane that lies on a circle of radius 1 centred on the origin (the *unit circle*). The argument of z is θ: see Figure 1.4. The advantage of this representation is that the angle 359° (very nearly a complete revolution) is represented by a point very near to the one that represents 0°, which simplifies calculations in some applications: Exercise 1.4 gives an example.

Block diagrams

We will often use block diagrams to help explain how digital signal processing systems are built up from simple modules. Each module is thought of as carrying out an operation on a sequence of incoming numbers one at a time, producing

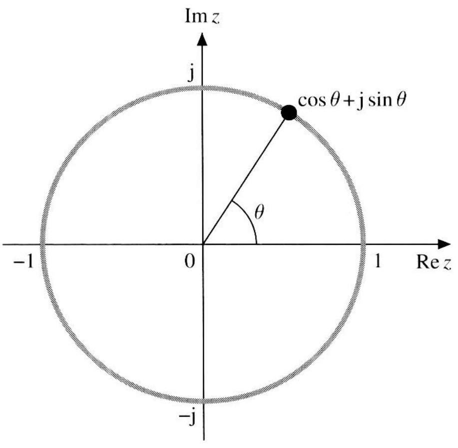

Figure 1.4 Representing an angle as a point on the unit circle in the complex plane.

a processed sequence of numbers at its output. Lines with arrows show the interconnections between modules.

The most basic modules are those that carry out the arithmetic operations of addition, subtraction, multiplication and division. The symbols we use for these modules are shown in Figure 1.5(a) to (d). The label on the input to the subtractor indicates that it subtracts its bottom input from its top one; likewise, the divider divides its bottom input into its top one.

Modules that carry out other operations are shown as boxes labelled with the relevant function. A box labelled with a 'D' delays its input by one time unit: see Figure 1.5(e).

These basic modules can be put together to construct higher-level building blocks such as filters, as we will describe in Chapter 5. Filters are often used as modules in more complicated signal-processing systems and so have their own

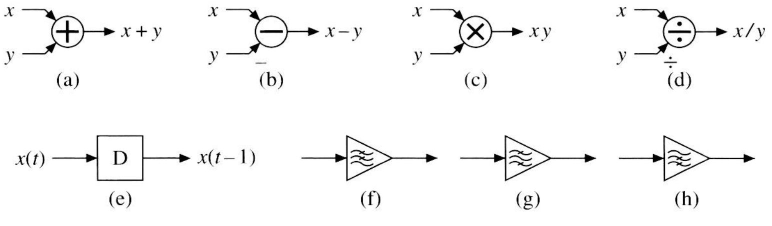

Figure 1.5 Block diagram symbols: (a) adder; (b) subtractor; (c) multiplier; (d) divider; (e) delay element; (f) low-pass filter; (g) band-pass filter; (h) high-pass filter.

试读结束，需要全本PDF请购买 www.ertongbook.com