System Analysis & Signal Processing

Philip Denbigh

with emphasis on the use of MATLAB®

Addison-Wesley

System analysis and signal processing

with emphasis on the use of MATLAB

PHILIP DENBIGH

University of Sussex

ADDISON-WESLEY

Harlow, England • Reading, Massachusetts • Menlow Park, California
New York • Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore
Tokyo • Madrid • San Juan • Milan • Mexico City • Seoul • Taipei

Addison Wesley Longman Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the World.

Typset in 10/12.5 Times by 25
Printed and bound in the United States of America

First printed 1998

ISBN 0-201-17860-5

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
Denbigh, Philip.
System analysis and signal processing : with emphasis on the use of MATLAB / Philip Denbigh.
p. cm.
Includes bibliographical references (p. 504) and index.
ISBN 0-201-17860-5 (alk. paper)
1. Signal processing—Computer simulation. 2. MATLAB. I. Title.
TK5102.9.D46 1998
621.382′2—dc21 97-40611
CIP

System analysis and signal processing

Preface

This is intended to be very much a *teaching* book, and the effort in writing it has been directed more at developing clear explanations of difficult concepts than at producing a definitive text. It covers a very broad syllabus on system analysis and continuous and discrete-time signal processing that is aimed at being suited to all years of an undergraduate programme, but also includes a small amount of more advanced material.

The book commences at a much more elementary level than would normally be associated with the later content of the book. The motive behind this is to ensure that the more difficult concepts are based upon a solid foundation. It is intended that the book should be substantially self-contained, showing how all the concepts build on each other – how the pieces of the jigsaw fit together. This means that a wide range of topics is covered and, in order to keep the book of reasonable length, the treatment is constrained to the essentials that are thought necessary for a good understanding. What has been left out is considered almost as important as what has been included. Details that some other texts include are deliberately omitted. This is for brevity, so as not to detract from those aspects which are thought to be more important, and so that the reader should not find the book heavy going or daunting. It is accepted that it may sometimes be useful to supplement the material with that from other texts. The bibliography contains a short list of books that have been selected for their readability and/or extended coverage of relevant material.

One special feature of the book is that an attempt is made to throw some physical insight into the Laplace transform, somewhat similar to that which is commonly and much more easily done with the Fourier transform. It is felt that the common practice of introducing the Laplace transform by a *definition*, without any *interpretation* of that definition, is unconvincing and too mathematical for most readers. It is hoped that the treatment given will be more acceptable.

A second major feature of the book is the extensive use in some chapters of the scientific software MATLAB. This is done, firstly because it produces impressive displays that help to explain the text, and secondly because it has become an invaluable and widely accepted software tool whose existence has affected signal processing design procedures. As such, an aquaintance with its power is considered to be an essential part of a signal processing syllabus. A particularly impressive feature of MATLAB is that student versions of it are available for the price of an average hardback textbook. These student versions do have limitations compared with the professional version of MATLAB, but are nevertheless very powerful and totally adequate for many applications. For use with Microsoft Windows 3.1 the relevant student edition is Version 4, which is available as a diskette plus manual (MATLAB 1995). This can also be used on Microsoft Windows 95 and Windows NT, but the updated student edition for these is Version 5, and this is available as a CD-ROM plus manual (MATLAB 1997). Versions 4 and 5 are also available for the appropriate Macintosh computers. All the MATLAB commands in this book are restricted to commands that are valid for both of these student editions, as well as being fully compatible with the professional versions of MATLAB which may be available to students through a copy on their university network.

Although primarily concerned with system analysis and signal processing, a secondary function of this book is to provide an introductory tutorial on the use of MATLAB. For this reason the code used is listed in the text rather than on an attached diskette. Sometimes MATLAB is used is solely for producing the figures, and it is felt inappropriate to include code in the main text. Since, however, this coding may be of interest to the reader learning MATLAB programming, a selected number of these figures are marked by an asterisk (for example Fig. 3.3*) and the code is listed in Appendix A.

The problems at the end of each chapter are few in number but have been carefully designed to make a significant contribution to the learning process. If the material is understood most of them can be solved quite quickly. A solutions manual is available to instructors.

It is important to end with some acknowledgements. I should like to express my gratitude to the anonymous reviewers of the original manuscript for making a variety of constructive suggestions, many of which were subsequently incorporated. Also, I should like to thank Ahmad Shamsoddini for passing on some of his knowledge on the less well-documented aspects of MATLAB and for his meticulous reading of my corrected proofs. As well as finding a number of remaining typos he pointed out and helped remedy a few unclear, and even erroneous, statements. However it is almost inevitable that some mistakes will remain and if readers find any, or can suggest improvements or areas that need clarification, I would be grateful to hear of them. My e-mail address is p.n.denbigh@sussex.ac.uk.

Philip Denbigh
December 1997

Contents

Trademark notice

Chapter one

Getting started in MATLAB and an introduction to systems and signal processing

1.1 ● Preview

This introductory chapter begins by presenting some of the main commands and capabilities of the powerful and widely used scientific software sold as MATLAB®. It is a very condensed version of what is described more fully in the manuals that accompany the student editions of MATLAB, constraining itself to the very minimum that is needed for getting started with its use in this book.

The chapter continues by classifying various types of signal and defining the linear time-invariant systems that form the subject of the book. Some simple examples of analogue and digital processing are given and reasons are presented for why digital processing is increasingly the preferred option. Finally, the special importance of sinusoidal, impulse function and step function signals is justified.

1.2 Getting started in MATLAB

MATLAB has become an invaluable and widely accepted software tool whose existence has affected signal processing design procedures; as such, an acquaintance with its power has become an important part of a signal processing syllabus. This section is intended as a very brief tutorial on some of its most important features. A particularly useful feature of MATLAB is that student editions are available from scientific bookshops for no more than the price of a typical hardback textbook, and yet they retain many of the most important capabilities of the full version. The relevant student edition for use with Microsoft Windows 3.1 is Version 4 and is available as a diskette plus manual (MATLAB 1995). This can also be used on Microsoft Windows 95 and Windows NT, but the updated student edition for these is Version 5, and this is available as a CD-ROM plus manual (MATLAB 1997). Versions 4 and 5 are also available for the appropriate Macintosh computers. Although fully compatible with the professional versions of MATLAB which may well be available to students through a copy on their university network, *all the MATLAB commands in this book are restricted to commands that are included in both of these student editions.*

Many sections of this book use MATLAB to demonstrate signal processing, and at these times the code will often be included in the main text. This is partly to demonstrate the power of the software and partly to provide further guidance in its use. In other sections MATLAB is used solely for producing figures, and on these occasions the code is usually omitted. Sometimes however, when there is something new to be learnt about the use of MATLAB for visualization, the figure is marked with an asterisk and the code is included in Appendix A.

The purpose of this present section is to describe some of the most basic features of MATLAB code so that the listings included later in the book will not need too much explanation to be understood.

1.2.1 Variables

When a number or a sequence of numbers is given a name it is termed a variable. A variable name can contain up to 19 characters, excluding punctuation characters. The first character must be a letter, but the remainder can be letters, digits or underscores and MATLAB distinguishes between lower case and upper case letters. An example of allocating a specific value to a variable is given by the command

```
b = 7
```

This causes the variable *b* to have the value 7. After entering this command the following appears on the screen

```
b =
        7
```

If it is wished to suppress such a message one follows the command with a semicolon. For example

```
b=7;
```

results in no such message on the screen.

A calculation can be involved when defining a variable. For example, the command

```
my_fingers=2*5;
```

introduces a variable *my_fingers* that has the value 10.

A few variables are built into MATLAB, namely *pi*, *i*, *j*, *ans*, *eps*, *inf*, *NaN*, *realmin* and *realmax*. Only *pi*, *j* and *ans* are used in this text. The variable *pi* is π. The variable *j* (which is the same as *i*) is $\sqrt{-1}$. The variable *ans* is a default variable name used for results. If, for example, we use MATLAB as a simple calculator by entering

4*5

we see

```
ans =
        20
```

1.2.2 Sequences, arrays and vectors

The previous subsection considered the situation where the variable had a single numerical value (it can be termed a *scalar* variable). Very often the variable is a *sequence* of numbers and one example of a four-element sequence might be noon temperatures over a four-day period. A set of one or more sequences can be termed an *array* and an example would be the daily noon temperatures over a four-day period at three locations, say London, New York and Tokyo. A hypothetical set of such temperatures could be entered as a MATLAB variable by encasing the values in square brackets separated by spaces and with the different number sequences separated by semicolons. Thus the command

x = [12.3 13.6 14.5 16.2; 18.1 18.8 19.2 18.4; 20.9 21.7 19.0 19.4]

results in the display

```
x =

   12.3000   13.6000   14.5000   16.2000
   18.1000   18.8000   19.2000   18.4000
   20.9000   21.7000   19.0000   19.4000
```

where x is an array in which the top row represents the London data, the second row the New York data, and the third row the Tokyo data. This array can also be thought of as a *matrix* of numbers. In this example it is a 3×4 matrix (3 rows by 4 columns). Generalizing this, an $m \times n$ matrix A is an array of numbers given by

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

This matrix A contains m rows each of n elements. Alternatively it can be considered as containing n columns each of m elements.

When the sequence is a one-dimensional array of numbers it is commonly termed a *vector*. In the example of temperatures given by the x array variable above, each set of numbers representing the temperatures at a single location is a *row vector*. If on the other hand we consider the temperatures at all three locations, but on one specific day, we would have a *column vector*. This book will deal mainly with vectors, and particularly with row vectors.