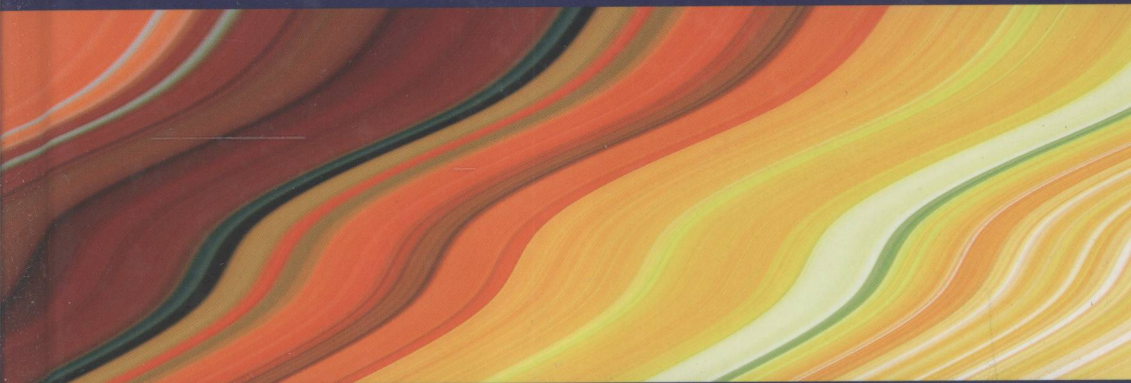


**DSP**

DIGITAL SIGNAL AND IMAGE PROCESSING SERIES



**Digital Signal  
Processing  
Using MATLAB**

**Edited by André Quinquis**

**STE**

 **WILEY**

TN911.72  
D575

# Digital Signal Processing using MATLAB<sup>®</sup>

Edited by  
André Quinquis



ISTE



E2008000832

 WILEY

First published in France in 2007 by Hermes Science/Lavoisier entitled "Le traitement du signal sous Matlab®: pratique et applications", 2<sup>nd</sup> edition

First published in Great Britain and the United States in 2008 by ISTE Ltd and John Wiley & Sons, Inc.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd  
6 Fitzroy Square  
London W1T 5DX  
UK

John Wiley & Sons, Inc.  
111 River Street  
Hoboken, NJ 07030  
USA

[www.iste.co.uk](http://www.iste.co.uk)

[www.wiley.com](http://www.wiley.com)

© ISTE Ltd, 2008

© LAVOISIER, 2007

The rights of André Quinquis to be identified as the author of this work have been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

---

Library of Congress Cataloging-in-Publication Data

Quinquis, André.  
[Traitement du signal sous MATLAB. English]  
Digital signal processing using MATLAB / André Quinquis.  
p. cm.  
Includes bibliographical references and index.  
ISBN 978-1-84821-011-0  
1. Signal processing--Digital techniques. I. Title.  
TK5102.9.Q853 2008  
621.382'2--dc22

2007043209

---

British Library Cataloguing-in-Publication Data

A CIP record for this book is available from the British Library  
ISBN: 978-1-84821-011-0

---

Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.



# Digital Signal Processing using MATLAB®

## Preface

# Why and How this Book was Written

Sometimes it is easier to say what a book is not than what it exactly represents. It may be also better to resume the authors' motivations than to explain the book content itself.

From this point of view, our book is certainly not a traditional course, although it recalls many theoretical signal processing concepts. Indeed, we emphasize a limited number of important ideas instead of making a detailed description of the involved concepts. Intuitive manners have been used to link these concepts to physical aspects. Hence, we hope that reading this book will be much more exciting than studying a traditional signal processing course.

This book is also not a physics course, although a major purpose of most proposed exercises is to link abstract signal processing concepts to real-life problems. These connections are illustrated in a simple and comprehensive manner through MATLAB<sup>®</sup> simulations.

The main topics of this book cover the usual program of an undergraduate signal processing course. It is especially written for language and computer science students, but also for a much larger scientific community who may wish to have a comprehensive signal processing overview. Students will certainly find here what they are looking for, while others will probably find new and interesting knowledge.

This book is also intended to illustrate our pedagogical approach, which is based on three major reasons:

1. Students need to know how the teaching provided can be useful for them; it is their customer attitude.

2. Students have good potential for doing independent work; their interest and curiosity should be continuously stimulated by:

- using a diversified pedagogical approach that combines the two sides of a complete presentation methodology: from components to the system and vice versa;
- encouraging them to take advantage of their creativity through interactive educational tools; they should be allowed to make changes and even contribute to their development.

3. Students have to improve and validate their knowledge through written work; writing is still the best way to focus someone's concentration.

The role of simulations is becoming more and more important in the framework of a scientific education because it is an effective way to understand many physical phenomena, some of them less known or mastered, and to take into account their complexity. Simulations may be thus very useful for:

- *understanding* working principles and deriving behavior laws;
- *learning* about processing methods and systems running using algorithms to reproduce them off-line;
- *evaluating* the performance and robustness of various algorithms and estimating the influence of different parameters.

Simulations in signal processing education enable students to learn faster and facilitate the comprehension of the involved physical principles. From a teaching point of view, simulation tools lead to lower costs and time efficiency.

This book is based on a signal processing course, which has been successfully given for many years in several universities. According to our experience, signal theory abstract concepts and signal processing practical potentialities can be linked only through tutorial classes and simulation projects. In this framework, simulations appear to be the necessary complement for the classical tripod theory – modeling – experimentation.

This book brings together into a clear and concise presentation the main signal processing methods and the related results, using MATLAB software as a simulation tool. Why MATLAB? Because it is:

- simple to learn and to use;
- powerful and flexible;
- accurate, robust and fast;
- widespread in both academic and industrial environments;
- continuously updated by professionals.

The word “signal” stands for a physical entity, most often of an electrical nature, like that observed at a microphone output. It is submitted to various transformations when it goes through a system. Thus, in a communication chain, the signal is subject to some changes (distortion, attenuation, etc.), which can make it unrecognizable. The aim is to understand this evolution in order to properly recover the initial message.

In other words, a signal is a physical support of information. It may carry the orders in a control and command equipment or multimedia (speech and image) over a network. It is generally very weak and it has to be handled with much caution in order to reach the signal processing final goal, i.e. information extraction and understanding.

Signal processing is widely used in many industrial applications such as: telecommunications, audio and speech signal processing, radar, sonar, non-destructive control, vibrations, biomedicine, imagery, etc. Standard signal processing functions include signal analysis, improvement, synthesis, compression, detection, classification, etc., which depend on and interact with each other in an integrated information processing chain.

The digital signal processing methods provide noteworthy capabilities: accurate system design, excellent equipment reproducibility, high stability of their exploitation characteristics and an outstanding supervision facility.

The digital signal processing boom is related to the development of fast algorithms to calculate the discrete Fourier transform. Indeed, this is the equivalent of the Fourier transform in the discrete domain and so it is a basic tool to study discrete systems. However, related concepts are generally considered highly theoretical and accessible to scientific researchers rather than to most engineers. This book aims to overcome this difficulty by putting the most useful results of this domain within the understanding of the engineer.

Chapter 1 briefly describes essential concepts of MATLAB software, which is an interactive software tailored for digital signal processing. Language rules, elementary operations as well as basic functions are presented. Chapter 2 illustrates the generation of 1D or 2D (image) digital signals as data vectors and matrices respectively.

Finding the solution of a signal processing problem involves several distinct phases. The first phase is the modeling: the designer chooses a representation model for an observed data. When it can be done very accurately the signals are said to be deterministic. A powerful tool for analyzing them is provided by the Fourier transform, also called frequency representation, which is presented in Chapter 5. Its

equivalent in the discrete domain is represented by the z-transform, which is developed in Chapter 6.

There are many other processes, which give different and apparently unpredictable results, although they are observed using identical experimental conditions. They are known as random processes, such as the receiver's thermal noise. The wide sense stationary random processes, which form a particularly interesting class of these signals, are presented in Chapter 3. Some useful statistical tools for testing different hypothesis about their parameters behavior are provided in Chapter 4.

From a very general point of view, digital signal processing covers all the operations, arithmetical calculations and number handling performed on the signal to be processed, defined by a number series, in order to obtain a transformed number series representing the processed signal. Very different functions can be carried out in this way, such as classical spectral analysis (Chapter 10), time-frequency analysis (Chapters 11 and 12), linear filtering (Chapters 7 and 8), detection and estimation (Chapter 9), and feature extraction for information classification or compression (Chapters 13 and 14).

Theoretical developments have been reduced to the necessary elements for a good understanding and an appropriate application of provided results. A lot of MATLAB programs, solved examples and proposed exercises make it possible to directly approach many practical applications. The reader interested in some more complementary information will find this in the references cited at the end of this book.

Finally, I would like to acknowledge all the members of my team, Emanuel Radoi, Cornel Ioana, Ali Mansour and H el ene Thomas, for their contributions to this book.

Andr e QUINQUIS



# Table of Contents

<b>Preface</b> . . . . .	ix
<b>Chapter 1. Introduction</b> . . . . .	1
1.1. Brief introduction to MATLAB . . . . .	1
1.1.1. MATLAB software presentation . . . . .	1
1.1.2. Important MATLAB commands and functions . . . . .	3
1.1.3. Operating modes and programming with MATLAB . . . . .	8
1.1.4. Example of work session with MATLAB . . . . .	10
1.1.5. MATLAB language . . . . .	13
1.2. Solved exercises . . . . .	13
<b>Chapter 2. Discrete-Time Signals</b> . . . . .	23
2.1. Theoretical background. . . . .	23
2.1.1. Mathematical model of 1D and 2D discrete-time signals . . . . .	23
2.1.2. Basic 1D and 2D discrete-time signals . . . . .	25
2.1.3. Periodic 1D and 2D discrete-time signal representation using the discrete-time Fourier series . . . . .	26
2.1.4. Representation of non-periodic 1D and 2D discrete-time signals by discrete-time Fourier transform . . . . .	27
2.1.5. Analytic signals . . . . .	27
2.2. Solved exercises . . . . .	29
2.3. Exercises . . . . .	51
<b>Chapter 3. Discrete-Time Random Signals</b> . . . . .	55
3.1. Theoretical background. . . . .	55
3.1.1. Introduction . . . . .	55
3.1.2. Real random variables . . . . .	56
3.1.3. Random processes . . . . .	60

3.2. Solved exercises . . . . .	64
3.3. Exercises . . . . .	80
<b>Chapter 4. Statistical Tests and High Order Moments . . . . .</b>	<b>83</b>
4.1. Theoretical background. . . . .	83
4.1.1. Moments . . . . .	84
4.1.2. Cumulants . . . . .	84
4.1.3. Cumulant properties . . . . .	85
4.1.4. Chi-square (Chi <sup>2</sup> ) tests. . . . .	86
4.1.5. Normality test using the Henry line . . . . .	86
4.2. Solved exercises . . . . .	88
4.3. Exercises . . . . .	99
<b>Chapter 5. Discrete Fourier Transform of Discrete-Time Signals . . . . .</b>	<b>103</b>
5.1. Theoretical background. . . . .	103
5.1.1. Discrete Fourier transform of 1D digital signals. . . . .	104
5.1.2. DFT of 2D digital signals . . . . .	105
5.1.3. Z-transform of 1D digital signals . . . . .	106
5.1.4. Z-transform of 2D digital signals . . . . .	106
5.1.5. Methods and algorithms for the DFT calculation . . . . .	106
5.2. Solved exercises . . . . .	109
5.3. Exercises . . . . .	134
<b>Chapter 6. Linear and Invariant Discrete-Time Systems. . . . .</b>	<b>137</b>
6.1. Theoretical background. . . . .	137
6.1.1. LTI response calculation. . . . .	137
6.1.2. LTI response to basic signals . . . . .	139
6.2. Solved exercises . . . . .	141
6.3. Exercises . . . . .	169
<b>Chapter 7. Infinite Impulse Response Filters . . . . .</b>	<b>173</b>
7.1. Theoretical background. . . . .	173
7.1.1. Transfer function and filter specifications for infinite impulse response (IIR) filters. . . . .	173
7.1.2. Design methods for IIR filters . . . . .	174
7.1.3. Frequency transformations . . . . .	180
7.2. Solved exercises . . . . .	182
7.3. Exercises . . . . .	194

<b>Chapter 8. Finite Impulse Response Filters</b> . . . . .	197
8.1. Theoretical background. . . . .	197
8.1.1. Transfer function and properties of FIR filters. . . . .	197
8.1.2. Design methods. . . . .	199
8.1.3. General conclusion about digital filter design . . . . .	203
8.2. Solved exercises . . . . .	204
8.3. Exercises . . . . .	213
<b>Chapter 9. Detection and Estimation</b> . . . . .	215
9.1. Theoretical background. . . . .	215
9.1.1. Matched filtering: optimal detection of a known noisy signal. . . . .	215
9.1.2. Linear optimal estimates. . . . .	216
9.1.3. Least squares (LS) method . . . . .	221
9.1.4. LS method with forgetting factor . . . . .	222
9.2. Solved exercises . . . . .	223
9.3. Exercises . . . . .	239
<b>Chapter 10. Power Spectral Density Estimation</b> . . . . .	241
10.1. Theoretical background. . . . .	241
10.1.1. Estimate properties . . . . .	241
10.1.2. Power spectral density estimation . . . . .	242
10.1.3. Parametric spectral analysis . . . . .	245
10.1.4. Super-resolution spectral analysis methods. . . . .	250
10.1.5. Other spectral analysis methods . . . . .	256
10.2. Solved exercises . . . . .	257
10.3. Exercises. . . . .	277
<b>Chapter 11. Time-Frequency Analysis</b> . . . . .	279
11.1. Theoretical background. . . . .	279
11.1.1. Fourier transform shortcomings: interpretation difficulties . . . . .	279
11.1.2. Spectrogram . . . . .	280
11.1.3. Time-scale analysis – wavelet transform . . . . .	281
11.1.4. Wigner-Ville distribution . . . . .	284
11.1.5. Smoothed WVD (SWVD). . . . .	287
11.2. Solved exercises . . . . .	288
11.3. Exercises. . . . .	304
<b>Chapter 12. Parametrical Time-Frequency Methods</b> . . . . .	307
12.1. Theoretical background. . . . .	307
12.1.1. Fractional Fourier transform. . . . .	307

- 12.1.2. Phase polynomial analysis concept. . . . . 309
- 12.1.3. Time-frequency representations based on warping operators . . . 314
- 12.2. Solved exercises . . . . . 317
- 12.3. Exercises. . . . . 338
  
- Chapter 13. Supervised Statistical Classification. . . . . 343**
  - 13.1. Theoretical background . . . . . 343
    - 13.1.1. Introduction . . . . . 343
    - 13.1.2. Data analysis methods . . . . . 344
    - 13.1.3. Supervised classifiers. . . . . 348
  - 13.2. Solved exercises . . . . . 362
  - 13.3. Exercises. . . . . 379
  
- Chapter 14. Data Compression . . . . . 383**
  - 14.1. Theoretical background . . . . . 383
    - 14.1.1. Transform-based compression methods . . . . . 384
    - 14.1.2. Parametric (predictive) model-based compression methods . . . . 385
    - 14.1.3. Wavelet packet-based compression methods . . . . . 386
    - 14.1.4. Vector quantization-based compression methods . . . . . 387
    - 14.1.5. Neural network-based compression methods. . . . . 388
  - 14.2. Solved exercises . . . . . 390
  - 14.3. Exercises. . . . . 403
  
- References . . . . . 405**
  
- List of Authors . . . . . 407**
  
- Index . . . . . 409**

# Chapter 1

## Introduction

### **1.1. Brief introduction to MATLAB**

#### **1.1.1. *MATLAB software presentation***

MATLAB (MATrix LABoratory) is an interactive software, developed by Math Works Inc. and intended especially for digital signal processing. It is particularly effective when the data format is vector or matrix.

MATLAB integrates digital calculus, data visualization and open environment programming. MATLAB exists under both Windows and UNIX. Many demonstrations are available using the command `demo`.

This digital simulation software enables a fast and simple visualization of the obtained results.

MATLAB was primarily written in FORTRAN and C. However, MATLAB knows to interpret commands, while a compilation of the source code is required by FORTRAN and C.

MATLAB is especially designed for digital signal processing and for complex digital system modeling and simulation. It is also suitable for processing data series, images or multidimensional data fields.

MATLAB software general structure is provided in Figure 1.1.

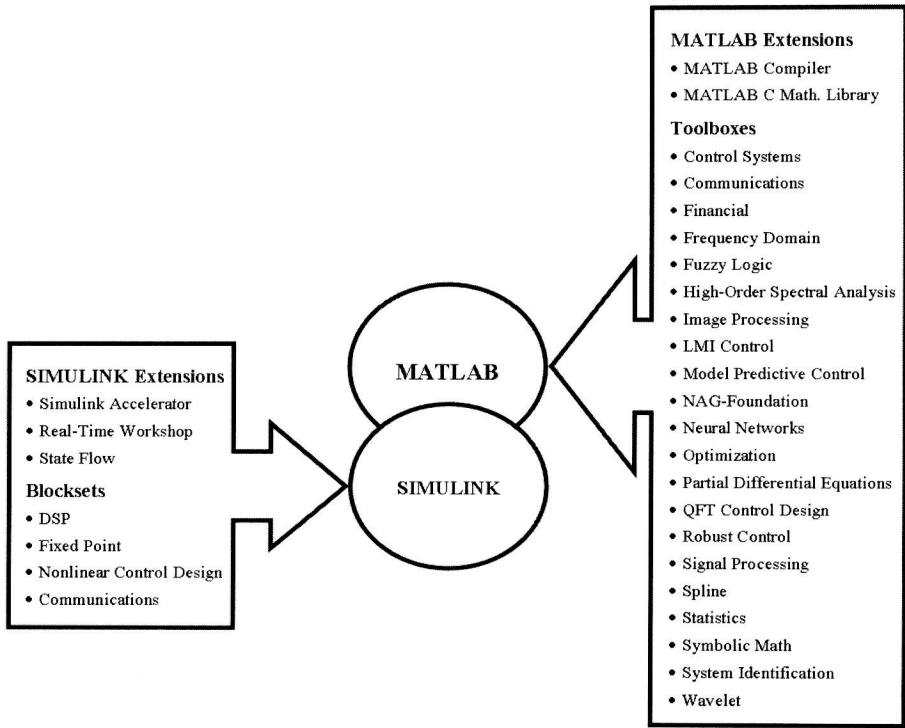


Figure 1.1. MATLAB software general structure

The toolboxes extend the basic MATLAB functions and perform specific tasks corresponding to different digital processing fields, such as image processing, optimization, statistics, system control and identification, neural networks, fuzzy systems, etc.

SIMULINK is an interactive software designed for modeling and simulating continuous-time or discrete-time dynamical systems or hybrid structures containing both analog and digital systems. It makes use of a mathematical equation set and provides a large variety of predefined or user-defined functional blocks.

MATLAB has been developed for several years, especially as a consequence of its use in the academic environment as an excellent education tool in mathematics, engineering and science. In addition, MATLAB has already proven its utility for scientific research and technological development.

In order to run MATLAB, type the command `matlab` with UNIX shell (if a MATLAB license under UNIX is available) or double click on the MATLAB icon if the operating system is Windows. To exit MATLAB, type `exit` or `quit`. If MATLAB is running under UNIX, you may have access to all UNIX commands using just before the symbol `!` (example: `!ls -l`).

### 1.1.2. Important MATLAB commands and functions

<code>who</code>	lists the variables in the current workspace
<code>whos</code>	the same as previous, but lists more information about each variable
<code>what</code>	lists MATLAB-specific files in directory
<code>size</code>	provides the size of a data array
<code>length</code>	provides the size of a data vector
<code>help</code>	displays help text in Command Window
<code>exit, quit</code>	exits from MATLAB

**Table 1.1.** *General commands*

<code>dir, chdir, delete, load, save, type</code>	similar to the corresponding DOS commands
<code>pack</code>	consolidates workspace memory

**Table 1.2.** *Commands related to the workspace*

<code>+, -, *, /, ^</code>	usual arithmetical operators
<code>.</code>	followed by an arithmetical operator for applying it to each array element
<code>'</code>	Hermitian operator
<code>.'</code>	transpose operator

**Table 1.3.** *Arithmetical operators*

#### 4 Digital Signal Processing using MATLAB

<, <=, >, >=	usual relational operators
==	equality operator
~=	inequality operator
&	element-wise logical AND
	element-wise logical OR
~	logical complement (NOT)

**Table 1.4.** *Relational and logical operators*

=	variable assignment operator
,	used to separate the arguments of a function or the elements of a data array
[]	used to build data arrays
()	used in arithmetical expressions
:	used for indexing variables
;	used at the end of a statement to cancel displaying any output
...	used to continue a command on the next line
%	used to enter a comment

**Table 1.5.** *Special characters*

ans	default name of a variable or a result
eps	spacing of floating point numbers
pi	value of $\pi = 3.14159\dots$
i, j	value of $\sqrt{-1}$
Inf	IEEE arithmetic representation for positive infinity (1/0)
NaN	IEEE arithmetic representation for Not-a-Number (0/0)
nargin	returns the number of function input arguments
nargout	returns the number of function output arguments

**Table 1.6.** *Special variables and constants*



abs	absolute value function
sqrt	square root function
real	real part of a complex variable
imag	imaginary part of a complex variable
angle	returns the phase angles, in radians, of a complex variable
conj	complex conjugate operator
sign	signum function
rem	returns the remainder after division
exp	exponential function
log	natural logarithm function
log10	base 10 logarithm function

**Table 1.7.** *Elementary mathematical functions*

sin, cos, tan, cot, sec	usual trigonometric functions
asin, acos, atan, acot, asec	inverse trigonometric functions
sinh, cosh, tanh, coth, sech	hyperbolic functions
asinh, acosh, atanh, acoth, asech	inverse hyperbolic functions

**Table 1.8.** *Trigonometric functions*

max	largest component
min	smallest component
mean	average or mean value
std	standard deviation
sum	sum of elements
cumsum	cumulative sum of elements
prod	product of elements
cumprod	cumulative product of elements

**Table 1.9.** *Data analysis functions*