Adaptive Filtering
Algorithms and Practical Implementation

Third Edition
Note to Instructors

For the instructors this book has a solution manual for the problems written by Dr. L. W. P. Biscainho available from the publisher. Also available, upon request to the author, is a set of master transparencies as well as the MATLAB® codes for all the algorithms described in the text.

1MATLAB is a registered trademark of The MathWorks, Inc.
To: My Parents,

Mariza,

Paula,

and Luiza.
The field of Digital Signal Processing has developed so fast in the last three decades that it can be found in the graduate and undergraduate programs of most universities. This development is related to the increasingly available technologies for implementing digital signal processing algorithms. The tremendous growth of development in the digital signal processing area has turned some of its specialized areas into fields themselves. If accurate information of the signals to be processed is available, the designer can easily choose the most appropriate algorithm to process the signal. When dealing with signals whose statistical properties are unknown, fixed algorithms do not process these signals efficiently. The solution is to use an adaptive filter that automatically changes its characteristics by optimizing the internal parameters. The adaptive filtering algorithms are essential in many statistical signal processing applications.

Although the field of adaptive signal processing has been subject of research for over four decades, it was in the eighties that a major growth occurred in research and applications. Two main reasons can be credited to this growth, the availability of implementation tools and the appearance of early textbooks exposing the subject in an organized manner. Still today it is possible to observe many research developments in the area of adaptive filtering, particularly addressing specific applications. In fact, the theory of linear adaptive filtering has reached a maturity that justifies a text treating the various methods in a unified way, emphasizing the algorithms suitable for practical implementation. This text concentrates on studying on-line algorithms, those whose adaptation occurs whenever a new sample of each environment signal is available. The so-called block algorithms, those whose adaptation occurs when a new block of data is available, are also included using the subband filtering framework. Usually, block algorithms require different implementation resources than the on-line algorithms. This edition also includes basic introductions to nonlinear adaptive filtering and blind signal processing as natural extensions of the algorithms treated in the earlier chapters. The understanding of the introductory material presented is fundamental for further studies in these fields which are described in more detail in some specialized texts.

The idea of writing this book started while teaching the adaptive signal processing course at the graduate school of the Federal University of Rio de Janeiro (UFRJ). The request of the students to cover as many algorithms as possible made me think how to organize this subject such that not much time is lost in adapting notations and derivations related to different algorithms. Another common question was which algorithms really work in a finite-precision implementation. These issues led me to conclude that a new text on this subject could be written with these objectives in mind. Also, considering that most graduate and undergraduate programs include a single adaptive filtering course, this book should not be lengthy. Another objective to seek is to provide an easy access to the working algorithms for the practitioner.
It was not until I spent a sabbatical year and a half at University of Victoria, Canada, that this project actually started. In the leisure hours, I slowly started this project. Parts of the early chapters of this book were used in short courses on adaptive signal processing taught at different institutions, namely: Helsinki University of Technology, Espoo, Finland; University Menendez Pelayo in Seville, Spain; and at the Victoria Micronet Center, University of Victoria, Canada. The remaining parts of the book were written based on notes of the graduate course in adaptive signal processing taught at COPPE (the graduate engineering school of UFRJ).

The philosophy of the presentation is to expose the material with a solid theoretical foundation, while avoiding straightforward derivations and repetition. The idea is to keep the text with a manageable size, without sacrificing clarity and without omitting important subjects. Another objective is to bring the reader up to the point where implementation can be tried and research can begin. A number of references are included at the end of the chapters in order to aid the reader to proceed on learning the subject.

It is assumed the reader has previous background on the basic principles of digital signal processing and stochastic processes, including: discrete-time Fourier- and $Z$-transforms, finite impulse response (FIR) and infinite impulse response (IIR) digital filter realizations, multirate systems, random variables and processes, first- and second-order statistics, moments, and filtering of random signals. Assuming that the reader has this background, I believe the book is self-contained.

Chapter 1 introduces the basic concepts of adaptive filtering and sets a general framework that all the methods presented in the following chapters fall under. A brief introduction to the typical applications of adaptive filtering are also presented.

In Chapter 2, the basic concepts of discrete-time stochastic processes are reviewed with special emphasis to the results that are useful to analyze the behavior of adaptive filtering algorithms. In addition, the Wiener filter is presented, establishing the optimum linear filter that can be sought in stationary environments. Appendix A briefly describes the concepts of complex differentiation mainly applied to the Wiener solution. The case of linearly constrained Wiener filter is also discussed, motivated by its wide use in antenna array processing. The transformation of the constrained minimization problem into an unconstrained one is also presented. The concept of mean-square error surface is then introduced, another useful tool to analyze adaptive filters. The classical Newton and steepest-descent algorithms are briefly introduced. Since the use of these algorithms would require a complete knowledge of the stochastic environment, the adaptive filtering algorithms introduced in the following chapters come into play. Practical applications of the adaptive filtering algorithms are revisited in more detail at the end of Chapter 2 where some examples with closed form solutions are included in order to allow the correct interpretation of what is expected from each application.

Chapter 3 presents and analyses of the least-mean-square (LMS) algorithm in some depth. Several aspects are discussed, such as convergence behavior in stationary and nonstationary environments. This chapter also includes a number of theoretical as well as simulation examples to illustrate how the LMS algorithm performs in different setups. Appendix B addresses the quantization effects on the LMS algorithm when implemented in fixed- and floating-point arithmetics.
Chapter 4 deals with some algorithms that are in a sense related to the LMS algorithm. In particular, the algorithms introduced are the quantized-error algorithms, the LMS-Newton algorithm, the normalized LMS algorithm, the transform-domain LMS algorithm, and the affine projection algorithm. Some properties of these algorithms are also discussed in Chapter 4, with special emphasis to the analysis of the fine projection algorithm.

Chapter 5 introduces the conventional recursive least-squares (RLS) algorithm. This algorithm minimizes a deterministic objective function, differing in this sense from most LMS-based algorithms. Following the same pattern of presentation of Chapter 3, several aspects of the conventional RLS algorithm are discussed, such as convergence behavior in stationary and nonstationary environments, along with a number of simulation results. Appendix C, deals with stability issues and quantization effects related to the RLS algorithm when implemented in fixed- and floating-point arithmetics. The results presented, except for the quantization effects, are also valid for the RLS algorithms presented in Chapters 7, 8, and 9. As as complement to Chapter 5, Appendix D presents the discrete-time Kalman filter formulation which despite being considered an extension of the Wiener filter has some relation with the RLS algorithm.

Chapter 6 discusses some techniques to reduce the overall computational complexity of adaptive filtering algorithms. The chapter first introduces the so called set-membership algorithms that update only when the output estimation error is higher than the prescribed upper bound. However, since set-membership algorithms require frequent updates during the early iterations in stationary environments, we introduce the concept of partial update to reduce the computational complexity in order to deal with situations where the available computational resources are not sufficient. This chapter presents several forms of set-membership algorithms related to the affine projection algorithms and their special cases. Chapter 6 also includes some simulation examples addressing standard as well as application oriented problems, where the algorithms of this and previous chapters are compared in some detail.

In Chapter 7, a family of fast RLS algorithms based on the FIR lattice realization is introduced. These algorithms represent interesting alternatives to the computationally complex conventional RLS algorithm. In particular, the unnormalized, the normalized and the error-feedback algorithms are presented.

Chapter 8 deals with the fast transversal RLS algorithms, which are very attractive due to their low computational complexity. However, these algorithms are known to face stability problems in practical implementation. As a consequence, special attention is given to the stabilized fast transversal RLS algorithm.

Chapter 9 is devoted to a family of RLS algorithms based on the QR decomposition. The conventional and a fast version of the QR-based algorithms are presented in this chapter.

Chapter 10 addresses the subject of adaptive filters using IIR digital filter realizations. The chapter includes a discussion on how to compute the gradient and how to derive the adaptive algorithms. The cascade, the parallel, and the lattice realizations are presented as interesting alternatives to the direct-form realization for the IIR adaptive filter. The characteristics of the mean-square error surface are also discussed in this chapter, for the IIR adaptive filtering case. Algorithms based on alternative error formulations, such as the equation error and Steiglitz-McBride methods are also introduced.
Chapter 11 deals with nonlinear adaptive filtering which consists of utilizing a nonlinear structure for the adaptive filter. The motivation is to use nonlinear adaptive filtering structures to better model some nonlinear phenomena commonly found in communications applications, such as nonlinear characteristics of power amplifier at transmitters. In particular, we introduce the Volterra series LMS and RLS algorithms, and the adaptive algorithms based on bilinear filters. Also, a brief introduction is given to some nonlinear adaptive filtering algorithms based on the concepts of neural networks, namely, the multilayer perceptron and the radial basis function algorithms. Some examples of DFE equalization are included in this chapter.

Chapter 12 deals with adaptive filtering in subbands mainly to address the applications where the required adaptive filter order is high, as for example in acoustic echo cancellation where the unknown system (echo) model has long impulse response. In subband adaptive filtering, some signals are split in frequency subbands via an analysis filter bank. Chapter 12 provides a brief review of multirate systems, and presents the basic structures for adaptive filtering in subbands. The concept of delayless subband adaptive filtering is also addressed, where the adaptive filter coefficients are updated in subbands and mapped to an equivalent fullband filter. The chapter also includes a discussion on the relation between subband and block adaptive filtering (also known as frequency-domain adaptive filters) algorithms.

Chapter 13 describes some adaptive filtering algorithms suitable for situations where no reference signal is available which are known as blind adaptive filtering algorithms. In particular, this chapter introduces some blind algorithms utilizing high-order statistics implicitly for the single-input single-output (SISO) equalization applications. In order to address some drawbacks of the SISO equalization systems, we discuss some algorithms using second-order statistics for the single-input multi-output (SIMO) equalization. The SIMO algorithms are naturally applicable in cases of oversampled received signal and multiple receive antennas. This chapter also discusses some issues related to blind signal processing not directly detailed here.

I decided to use some standard examples to present a number of simulation results, in order to test and compare different algorithms. This way, frequent repetition was avoided while allowing the reader to easily compare the performance of the algorithms. Most of the end of chapters problems are simulation oriented, however, some theoretical ones are included to complement the text.

The second edition differed from the first one mainly by the inclusion of chapters on nonlinear and subband adaptive filtering. Many other smaller changes were performed throughout the remaining chapters. In this edition, we introduced a number of derivations and explanations requested by students and suggested by colleagues. In addition, two new chapters on data-selective algorithms and blind adaptive filtering are included along with a large number of new examples and problems. Major changes take place in the first five chapters in order to make the technical details more accessible and to improve the ability of the reader in deciding where and how to use the concepts. The analysis of the fine projection algorithm is now presented in detail due to its growing practical importance. Several practical and theoretical examples are included aiming at comparing the families of algorithms introduced in the book.

In a trimester course, I usually cover Chapters 1 to 6 sometimes skipping parts of Chapter 2 and the analyses of quantization effects in Appendices B and C. In the remaining time, I try to cover as much as possible of the remaining chapters, usually consulting the audience to what they would
prefer to study. This book can also be used for self-study where the reader can examine Chapters 1 to 6, and those not involved with specialized implementations can skip the Appendices B and C, without loss of continuity. The remaining chapters can be followed separately, except for Chapter 8 that requires reading Chapter 7. Chapters 7, 8, and 9 deal with alternative and fast implementations of RLS algorithms and the following chapters do not use their results.
ACKNOWLEDGMENTS

The support of the Department of Electronics and Computer Engineering of the Polytechnic School (undergraduate school of engineering) of UFRJ and of the Program of Electrical Engineering of COPPE have been fundamental to complete this work.

I was lucky enough to have contact with a number of creative professors and researchers that by taking their time to discuss technical matters with me, raised many interesting questions and provided me with enthusiasm to write the first, second, and third editions of this book. In that sense, I would like to thank Prof. Pan Agathoklis, University of Victoria; Prof. R. C. de Lamare, University of York; Prof. M. Gerken, University of São Paulo; Prof. A. Hjørungnes, UniK-University of Oslo; Prof. T. I. Laakso, Helsinki Uuiversity of Technology; Prof. J. P. Leblanc, Luleå University of Technology; Prof. W. S. Lu, University of Victoria; Dr. H. S. Malvar, Microsoft Research; Prof. V. H. Nascimento, University of São Paulo; Prof. J. M. T. Romano, State University of Campinas; Prof. E. Sanchez Sinencio, Texas A&M University; Prof. Trac D. Tran, John Hopkins University.

My M.Sc. supervisor, my friend and colleague, Prof. L. P. Calôba has been a source of inspiration and encouragement not only for this work but for my entire career. Prof. A. Antoniou, my Ph.D. supervisor, has also been an invaluable friend and advisor, I learned a lot by writing papers with him. I was very fortunate to have these guys as Professors.

The good students that attend engineering at UFRJ are for sure another source of inspiration. In particular, I have been lucky to attract good and dedicated graduate students, who have participated in the research related to adaptive filtering. Some of them are: Dr. R. G. Alves, Prof. J. A. Apolinário, Jr., Prof. L. W. P. Biscainho, Prof. M. L. R. Campos, Prof. J. E. Cousseau, T. N. Ferreira, M. V. S. Lima, T. C. Macedo, Jr., W. A. Martins, Prof. R. Merched, Prof. S. L. Netto, G. O. Pinto, C. B. Ribeiro, A. D. Santana, Jr., Dr. M. G. Siqueira, Dr. S. Subramanian (Anna University), Prof. F. G. V. Resende Jr., M. R. Vassali, Dr. S. Werner (Helsinki University of Technology). Most of them took time from their M.Sc. and Ph.D. work to read parts of the manuscript and provided me with invaluable suggestions. Some parts of this book have been influenced by my interactions with these and other former students.

I am particularly grateful to Profs. L. W. P. Biscainho, M. L. R. Campos and J. E. Cousseau, for their support in producing some of the examples of the book. Profs. L. W. P. Biscainho, M. L. R. Campos, and S. L. Netto also read every inch of the manuscript and provided numerous suggestions for improvements.
I am most grateful to Profs. E. A. B. da Silva, UFRJ, and R. Merched, UFRJ, for their critical inputs on parts of the manuscript. Prof. E. A. B. da Silva seems to be always around in difficult times to lay a helping hand.

Indeed the friendly and harmonious work environment of the LPS, the Signal Processing Laboratory of UFRJ, has been an enormous source of inspiration and challenge. From its manager Michelle to the Professors, undergraduate and graduate students, and staff, I always find support that goes beyond the professional obligation. Jane made many of the drawings with care, I really appreciate it.

I am also thankful to Prof. I. Hartimo, Helsinki University of Technology; Prof. J. L. Huertas, University of Seville; Prof. A. Antoniou, University of Victoria; Prof. J. E. Cousseau, Universidad Nacional del Sur; Prof. Y.-F. Huang, University of Notre Dame; Prof. A. Hjørungnes, UniK-University of Oslo, for giving me the opportunity to teach at the institutions they work for.

In recent years, I have been working as consultant to INdT (NOKIA Institute of Technology) where its President G. Feitoza and their researchers have team up with me in challenging endeavors. They are always posing me with problems, not necessarily technical, which widen my way of thinking.

The earlier support of Catherine Chang, Prof. J. E. Cousseau, Prof. F. G. V. Resende Jr., and Dr. S. Sunder for solving my problems with the text editor is also deeply appreciated.

The financial supports of the Brazilian research councils CNPJq, CAPES, and FAPERJ were fundamental for the completion of this book.

The friendship and trust of my editor Alex Greene, from Springer, have been crucial to make this third edition a reality.

My parents provided me with the moral and educational support needed to pursue any project, including this one. My mother's patience, love and understanding seem to be endless.

My brother Fernando always says yes, what else do I want? He also awarded me with my nephews Fernandinho and Daniel.

My family deserves special thanks. My daughters Paula and Luiza have been extremely understanding, and always forgive daddy for being busy. They are wonderful young ladies. My wife Mariza deserves my deepest gratitude for her endless love, support, and friendship. She always does her best to provide me with the conditions to develop this and other projects.

Prof. Paulo S. R. Diniz

Niterói, Brazil
# CONTENTS

## PREFACE

ix

## 1 INTRODUCTION TO ADAPTIVE FILTERING

1.1 Introduction 1

1.2 Adaptive Signal Processing 2

1.3 Introduction to Adaptive Algorithms 4

1.4 Applications 7

1.5 References 11

## 2 FUNDAMENTALS OF ADAPTIVE FILTERING

2.1 Introduction 13

2.2 Signal Representation 14

2.2.1 Deterministic Signals 14

2.2.2 Random Signals 15

2.2.3 Ergodicity 21

2.3 The Correlation Matrix 23

2.4 Wiener Filter 34

2.5 Linearly Constrained Wiener Filter 39

2.5.1 The Generalized Sidelobe Canceller 43

2.6 Mean-Square Error Surface 44

2.7 Bias and Consistency 47

2.8 Newton Algorithm 48

2.9 Steepest-Descent Algorithm 49

2.10 Applications Revisited 54

2.10.1 System Identification 54

2.10.2 Signal Enhancement 55

2.10.3 Signal Prediction 56

2.10.4 Channel Equalization 57

2.10.5 Digital Communication System 65

2.11 Concluding Remarks 67

2.12 References 68

2.13 Problems 70
### 4.7 Simulation Examples

- 4.7.1 Signal Enhancement Simulation
- 4.7.2 Signal Prediction Simulation

### 4.8 Concluding Remarks

### 4.9 References

### 4.10 Problems

### 5 CONVENTIONAL RLS ADAPTIVE FILTER

- 5.1 Introduction
- 5.2 The Recursive Least-Squares Algorithm
- 5.3 Properties of the Least-Squares Solution
  - 5.3.1 Orthogonality Principle
  - 5.3.2 Relation Between Least-Squares and Wiener Solutions
  - 5.3.3 Influence of the Deterministic Autocorrelation Initialization
  - 5.3.4 Steady-State Behavior of the Coefficient Vector
  - 5.3.5 Coefficient-Error-Vector Covariance Matrix
  - 5.3.6 Behavior of the Error Signal
  - 5.3.7 Excess Mean-Square Error and Misadjustment
- 5.4 Behavior in Nonstationary Environments
- 5.5 Complex RLS Algorithm
- 5.6 Simulation Examples
- 5.7 Concluding Remarks
- 5.8 References
- 5.9 Problems

### 6 DATA-SELECTIVE ADAPTIVE FILTERING

- 6.1 Introduction
- 6.2 Set-Membership Filtering
- 6.3 Set-Membership Normalized LMS Algorithm
- 6.4 Set-Membership Affine Projection Algorithm
  - 6.4.1 A Trivial Choice for Vector $\hat{\gamma}(k)$
  - 6.4.2 A Simple Vector $\hat{\gamma}(k)$
  - 6.4.3 Reducing the Complexity in the Simplified SM-AP Algorithm
- 6.5 Set-Membership BINormalized LMS Algorithms
  - 6.5.1 SM-BNLMS Algorithm 1
  - 6.5.2 SM-BNLMS Algorithm 2
- 6.6 Computational Complexity
- 6.7 Time-Varying $\hat{\gamma}$
- 6.8 Partial-Update Adaptive Filtering
  - 6.8.1 Set-Membership Partial-Update NLMS Algorithm
7 ADAPTIVE LATTICE-BASED RLS ALGORITHMS

7.1 Introduction 289

7.2 Recursive Least-Squares Prediction 290

7.2.1 Forward Prediction Problem 290

7.2.2 Backward Prediction Problem 293

7.3 Order-Updating Equations 295

7.3.1 A New Parameter $\delta(k, i)$ 295

7.3.2 Order Updating of $\xi_{b_{\min}}(k, i)$ and $w_b(k, i)$ 297

7.3.3 Order Updating of $\xi_{f_{\min}}(k, i)$ and $w_f(k, i)$ 298

7.3.4 Order Updating of Prediction Errors 298

7.4 Time-Updating Equations 300

7.4.1 Time Updating for Prediction Coefficients 300

7.4.2 Time Updating for $\delta(k, i)$ 302

7.4.3 Order Updating for $\gamma(k, i)$ 304

7.5 Joint-Process Estimation 307

7.6 Time Recursions of the Least-Squares Error 311

7.7 Normalized Lattice RLS Algorithm 313

7.7.1 Basic Order Recursions 313

7.7.2 Feedforward Filtering 315

7.8 Error-Feedback Lattice RLS Algorithm 318

7.8.1 Recursive Formulas for the Reflection Coefficients 318

7.9 Lattice RLS Algorithm Based on A Priori Errors 319

7.10 Quantization Effects 321

7.11 Concluding Remarks 327

7.12 References 328

7.13 Problems 329

8 FAST TRANSVERSAL RLS ALGORITHMS

8.1 Introduction 333

8.2 Recursive Least-Squares Prediction 334

8.2.1 Forward Prediction Relations 334

8.2.2 Backward Prediction Relations 335

8.3 Joint-Process Estimation 337
8.4 Stabilized Fast Transversal RLS Algorithm 339
8.5 Concluding Remarks 345
8.6 References 346
8.7 Problems 347

9 QR-DECOMPOSITION-BASED RLS FILTERS 351
9.1 Introduction 351
9.2 Triangularization Using QR-Decomposition 351
  9.2.1 Initialization Process 353
  9.2.2 Input Data Matrix Triangularization 353
  9.2.3 QR-Decomposition RLS Algorithm 360
9.3 Systolic Array Implementation 365
9.4 Some Implementation Issues 372
9.5 Fast QR-RLS Algorithm 373
  9.5.1 Backward Prediction Problem 376
  9.5.2 Forward Prediction Problem 378
9.6 Conclusions and Further Reading 384
9.7 References 387
9.8 Problems 389

10 ADAPTIVE IIR FILTERS 395
10.1 Introduction 395
10.2 Output-Error IIR Filters 396
10.3 General Derivative Implementation 400
10.4 Adaptive Algorithms 402
  10.4.1 Recursive Least-Squares Algorithm 402
  10.4.2 The Gauss-Newton Algorithm 404
  10.4.3 Gradient-Based Algorithm 407
10.5 Alternative Adaptive Filter Structures 407
  10.5.1 Cascade Form 407
  10.5.2 Lattice Structure 409
  10.5.3 Parallel Form 416
  10.5.4 Frequency-Domain Parallel Structure 417
10.6 Mean-Square Error Surface 426
10.7 Influence of the Filter Structure on the MSE Surface 433
10.8 Alternative Error Formulations 435
  10.8.1 Equation Error Formulation 435
  10.8.2 The Steiglitz-McBride Method 439
10.9 Conclusion 442
10.10 References 443
10.11 Problems 446
# 11 Nonlinear Adaptive Filtering

11.1 Introduction 451
11.2 The Volterra Series Algorithm 452
   11.2.1 LMS Volterra Filter 454
   11.2.2 RLS Volterra Filter 457
11.3 Adaptive Bilinear Filters 464
11.4 Multilayer Perceptron Algorithm 469
11.5 Radial Basis Function Algorithm 473
11.6 Conclusion 480
11.7 References 482
11.8 Problems 484

# 12 Subband Adaptive Filters

12.1 Introduction 485
12.2 Multirate Systems 486
   12.2.1 Decimation and Interpolation 486
12.3 Filter Banks 488
   12.3.1 Two-Band Perfect Reconstruction Filter Banks 493
   12.3.2 Analysis of Two-Band Filter Banks 494
   12.3.3 Analysis of M-Band Filter Banks 494
   12.3.4 Hierarchical M-Band Filter Banks 495
   12.3.5 Cosine-Modulated Filter Banks 495
   12.3.6 Block Representation 497
12.4 Subband Adaptive Filters 497
   12.4.1 Subband Identification 501
   12.4.2 Two-Band Identification 502
   12.4.3 Closed-Loop Structure 502
12.5 Cross-Filters Elimination 508
   12.5.1 Fractional Delays 510
12.6 Delayless Subband Adaptive Filtering 515
   12.6.1 Computational Complexity 517
12.7 Frequency-Domain Adaptive Filtering 521
12.8 Conclusion 530
12.9 References 531
12.10 Problems 533

# 13 Blind Adaptive Filtering

13.1 Introduction 537
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D KALMAN FILTERS</strong></td>
<td>605</td>
</tr>
<tr>
<td>D.1 Introduction</td>
<td>605</td>
</tr>
<tr>
<td>D.2 State-Space Model</td>
<td>605</td>
</tr>
<tr>
<td>D.2.1 Simple Example</td>
<td>606</td>
</tr>
<tr>
<td>D.3 Kalman Filtering</td>
<td>608</td>
</tr>
<tr>
<td>D.4 Kalman Filter and RLS</td>
<td>614</td>
</tr>
<tr>
<td>D.5 References</td>
<td>615</td>
</tr>
</tbody>
</table>

**INDEX** 617
1.1 INTRODUCTION

In this section, we define the kind of signal processing systems that will be treated in this text.

In the last thirty years significant contributions have been made in the signal processing field. The advances in digital circuit design have been the key technological development that sparked a growing interest in the field of digital signal processing. The resulting digital signal processing systems are attractive due to their low cost, reliability, accuracy, small physical sizes, and flexibility.

One example of a digital signal processing system is called filter. Filtering is a signal processing operation whose objective is to process a signal in order to manipulate the information contained in the signal. In other words, a filter is a device that maps its input signal to another output signal facilitating the extraction of the desired information contained in the input signal. A digital filter is the one that processes discrete-time signals represented in digital format. For time-invariant filters the internal parameters and the structure of the filter are fixed, and if the filter is linear the output signal is a linear function of the input signal. Once prescribed specifications are given, the design of time-invariant linear filters entails three basic steps, namely: the approximation of the specifications by a rational transfer function, the choice of an appropriate structure defining the algorithm, and the choice of the form of implementation for the algorithm.

An adaptive filter is required when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters. Strictly speaking an adaptive filter is a nonlinear filter since its characteristics are dependent on the input signal and consequently the homogeneity and additivity conditions are not satisfied. However, if we freeze the filter parameters at a given instant of time, most adaptive filters considered in this text are linear in the sense that their output signals are linear functions of their input signals. The exceptions are the adaptive filters discussed in Chapter 11.

The adaptive filters are time-varying since their parameters are continually changing in order to meet a performance requirement. In this sense, we can interpret an adaptive filter as a filter that performs the approximation step on-line. Usually, the definition of the performance criterion requires the existence of a reference signal that is usually hidden in the approximation step of fixed-filter
design. This discussion brings the feeling that in the design of fixed (nonadaptive) filters a complete characterization of the input and reference signals is required in order to design the most appropriate filter that meets a prescribed performance. Unfortunately, this is not the usual situation encountered in practice, where the environment is not well defined. The signals that compose the environment are the input and the reference signals, and in cases where any of them is not well defined, the design procedure is to model the signals and subsequently design the filter. This procedure could be costly and difficult to implement on-line. The solution to this problem is to employ an adaptive filter that performs on-line updating of its parameters through a rather simple algorithm, using only the information available in the environment. In other words, the adaptive filter performs a data-driven approximation step.

The subject of this book is adaptive filtering, which concerns the choice of structures and algorithms for a filter that has its parameters (or coefficients) adapted, in order to improve a prescribed performance criterion. The coefficient updating is performed using the information available at a given time.

The development of digital very large scale integration (VLSI) technology allowed the widespread use of adaptive signal processing techniques in a large number of applications. This is the reason why in this book only discrete-time implementations of adaptive filters are considered. Obviously, we assume that continuous-time signals taken from the real world are properly sampled, i.e., they are represented by discrete-time signals with sampling rate higher than twice their highest frequency. Basically, it is assumed that when generating a discrete-time signal by sampling a continuous-time signal, the Nyquist or sampling theorem is satisfied [1]-[9].

1.2 ADAPTIVE SIGNAL PROCESSING

As previously discussed, the design of digital filters with fixed coefficients requires well defined prescribed specifications. However, there are situations where the specifications are not available, or are time varying. The solution in these cases is to employ a digital filter with adaptive coefficients, known as adaptive filters [10]-[17].

Since no specifications are available, the adaptive algorithm that determines the updating of the filter coefficients, requires extra information that is usually given in the form of a signal. This signal is in general called a desired or reference signal, whose choice is normally a tricky task that depends on the application.

Adaptive filters are considered nonlinear systems, therefore their behavior analysis is more complicated than for fixed filters. On the other hand, because the adaptive filters are self designing filters, from the practitioner’s point of view their design can be considered less involved than in the case of digital filters with fixed coefficients.

The general set up of an adaptive-filtering environment is illustrated in Fig. 1.1, where \( k \) is the iteration number, \( x(k) \) denotes the input signal, \( y(k) \) is the adaptive-filter output signal, and \( d(k) \) defines the desired signal. The error signal \( e(k) \) is calculated as \( d(k) - y(k) \). The error signal is
then used to form a performance (or objective) function that is required by the adaptation algorithm in order to determine the appropriate updating of the filter coefficients. The minimization of the objective function implies that the adaptive-filter output signal is matching the desired signal in some sense.

![General adaptive-filter configuration.](image)

The complete specification of an adaptive system, as shown in Fig. 1.1, consists of three items:

1) **Application**: The type of application is defined by the choice of the signals acquired from the environment to be the input and desired-output signals. The number of different applications in which adaptive techniques are being successfully used has increased enormously during the last two decades. Some examples are echo cancellation, equalization of dispersive channels, system identification, signal enhancement, adaptive beamforming, noise cancelling, and control [14]-[20]. The study of different applications is not the main scope of this book. However, some applications are considered in some detail.

2) **Adaptive-Filter Structure**: The adaptive filter can be implemented in a number of different structures or realizations. The choice of the structure can influence the computational complexity (amount of arithmetic operations per iteration) of the process and also the necessary number of iterations to achieve a desired performance level. Basically, there are two major classes of adaptive digital filter realizations, distinguished by the form of the impulse response, namely the finite-duration impulse response (FIR) filter and the infinite-duration impulse response (IIR) filters. FIR filters are usually implemented with nonrecursive structures, whereas IIR filters utilize recursive realizations.

- **Adaptive FIR filter realizations**: The most widely used adaptive FIR filter structure is the transversal filter, also called tapped delay line, that implements an all-zero transfer function with a canonic direct form realization without feedback. For this realization, the output signal
Chapter 1 Introduction to Adaptive Filtering

$y(k)$ is a linear combination of the filter coefficients, that yields a quadratic mean-square error (MSE = $E[|e(k)|^2]$) function with a unique optimal solution. Other alternative adaptive FIR realizations are also used in order to obtain improvements as compared to the transversal filter structure, in terms of computational complexity, speed of convergence, and finite wordlength properties as will be seen later in the book.

- Adaptive IIR filter realizations: The most widely used realization of adaptive IIR filters is the canonic direct form realization [5], due to its simple implementation and analysis. However, there are some inherent problems related to recursive adaptive filters which are structure dependent, such as pole-stability monitoring requirement and slow speed of convergence. To address these problems, different realizations were proposed attempting to overcome the limitations of the direct form structure. Among these alternative structures, the cascade, the lattice, and the parallel realizations are considered because of their unique features as will be discussed in Chapter 10.

3) Algorithm: The algorithm is the procedure used to adjust the adaptive filter coefficients in order to minimize a prescribed criterion. The algorithm is determined by defining the search method (or minimization algorithm), the objective function, and the error signal nature. The choice of the algorithm determines several crucial aspects of the overall adaptive process, such as existence of sub-optimal solutions, biased optimal solution, and computational complexity.

1.3 INTRODUCTION TO ADAPTIVE ALGORITHMS

The basic objective of the adaptive filter is to set its parameters, $\theta(k)$, in such a way that its output tries to minimize a meaningful objective function involving the reference signal. Usually, the objective function $F$ is a function of the input, the reference, and adaptive-filter output signals, i.e., $F = F[x(k), d(k), y(k)]$. A consistent definition of the objective function must satisfy the following properties:

- Non-negativity: $F[x(k), d(k), y(k)] \geq 0$, $\forall y(k), x(k)$, and $d(k)$;
- Optimality: $F[x(k), d(k), d(k)] = 0$.

One should understand that in an adaptive process, the adaptive algorithm attempts to minimize the function $F$, in such a way that $y(k)$ approximates $d(k)$, and as a consequence, $\theta(k)$ converges to $\theta_o$, where $\theta_o$ is the optimum set of coefficients that leads to the minimization of the objective function.

Another way to interpret the objective function is to consider it a direct function of a generic error signal $e(k)$, which in turn is a function of the signals $x(k)$, $y(k)$, and $d(k)$, i.e., $F = F[e(k)] = F[e(x(k), y(k), d(k))]$. Using this framework, we can consider that an adaptive algorithm is composed of three basic items: definition of the minimization algorithm, definition of the objective function form, and definition of the error signal.
1.3 Introduction to Adaptive Algorithms

1) **Definition of the minimization algorithm for the function** \( F \): This item is the main subject of Optimization Theory [22]-[23], and it essentially affects the speed of convergence and computational complexity of the adaptive process.

In practice any continuous function having high-order model of the parameters can be approximated around a given point \( \theta(k) \) by a truncated Taylor series as follows

\[
F[\theta(k) + \Delta \theta(k)] \approx F[\theta(k)] + g_{\theta}[F[\theta(k)]] \Delta \theta(k) + \frac{1}{2} \Delta \theta^T(k) H_{\theta}[F[\theta(k)]] \Delta \theta(k) \tag{1.1}
\]

where \( H_{\theta}[F[\theta(k)]] \) is the Hessian matrix of the objective function, and \( g_{\theta}[F[\theta(k)]] \) is the gradient vector, further details about the Hessian matrix and gradient vector are presented along the text. The aim is to minimize the objective function with respect to the set of parameters by iterating

\[
\theta(k + 1) = \theta(k) + \Delta \theta(k) \tag{1.2}
\]

where the step or correction term \( \Delta \theta(k) \) is meant to minimize the quadratic approximation of the objective function \( F[\theta(k)] \). The so-called Newton method requires the first and second-order derivatives of \( F[\theta(k)] \) to be available at any point, as well as the function value. These informations are required in order to evaluate equation (1.1). If \( H_{\theta}(\theta(k)) \) is a positive definite matrix, then the quadratic approximation has a unique and well defined minimum point. Such a solution can be found by setting the gradient of the quadratic function with respect to the parameters correction terms, at instant \( k + 1 \), to zero which leads to

\[
g_{\theta}[F[\theta(k)]] = -H_{\theta}[F[\theta(k)]] \Delta \theta(k) \tag{1.3}
\]

The most commonly used optimization methods in the adaptive signal processing field are:

- **Newton’s method**: This method seeks the minimum of a second-order approximation of the objective function using an iterative updating formula for the parameter vector given by

\[
\theta(k + 1) = \theta(k) - \mu H_{\theta}^{-1}[F[e(k)]] g_{\theta}[F[e(k)]] \tag{1.4}
\]

where \( \mu \) is a factor that controls the step size of the algorithm, i.e., it determines how fast the parameter vector will be changed. The reader should note that the direction of the correction term \( \Delta \theta(k) \) is chosen according to equation (1.3). The matrix of second derivatives of \( F[e(k)] \), \( H_{\theta}[F[e(k)]] \) is the Hessian matrix of the objective function, and \( g_{\theta}[F[e(k)]] \) is the gradient of the objective function with respect to the adaptive filter coefficients. It should be noted that the error \( e(k) \) depends on the parameters \( \theta(k) \). If the function \( F[e(k)] \) is originally quadratic, there is no approximation in the model of equation (1.1) and the global minimum of the objective function would be reached in one step if \( \mu = 1 \). For nonquadratic functions the value of \( \mu \) should be reduced.

- **Quasi-Newton methods**: This class of algorithms is a simplified version of the method above described, as it attempts to minimize the objective function using a recursively calculated estimate of the inverse of the Hessian matrix, i.e.,

\[
\theta(k + 1) = \theta(k) - \mu S(k) g_{\theta}[F[e(k)]] \tag{1.5}
\]
where $S(k)$ is an estimate of $H^{-1}_θ\{F[e(k)]\}$, such that

$$\lim_{k \to \infty} S(k) = H^{-1}_θ\{F[e(k)]\}$$

A usual way to calculate the inverse of the Hessian estimate is through the matrix inversion lemma (see, for example [21] and some chapters to come). Also, the gradient vector is usually replaced by a computationally efficient estimate.

- Steepest-descent method: This type of algorithm searches the objective function minimum point following the opposite direction of the gradient vector of this function. Consequently, the updating equation assumes the form

$$\theta(k + 1) = \theta(k) - \mu g_θ\{F[e(k)]\} \quad (1.6)$$

Here and in the open literature, the steepest-descent method is often also referred to as gradient method.

In general, gradient methods are easier to implement, but on the other hand, the Newton method usually requires a smaller number of iterations to reach a neighborhood of the minimum point. In many cases, Quasi-Newton methods can be considered a good compromise between the computational efficiency of the gradient methods and the fast convergence of the Newton method. However, the Quasi-Newton algorithms are susceptible to instability problems due to the recursive form used to generate the estimate of the inverse Hessian matrix. A detailed study of the most widely used minimization algorithms can be found in [22]-[23].

It should be pointed out that with any minimization method, the convergence factor $\mu$ controls the stability, speed of convergence, and some characteristics of residual error of the overall adaptive process. Usually, an appropriate choice of this parameter requires a reasonable amount of knowledge of the specific adaptive problem of interest. Consequently, there is no general solution to accomplish this task. In practice, computational simulations play an important role and are, in fact, the most used tool to address the problem.

2) Definition of the objective function $F[e(k)]$: There are many ways to define an objective function that satisfies the optimality and non-negativity properties formerly described. This definition affects the complexity of the gradient vector and the Hessian matrix calculation. Using the algorithm’s computational complexity as a criterion, we can list the following forms for the objective function as the most commonly used in the derivation of an adaptive algorithm:

- Mean-Square Error (MSE): $F[e(k)] = E[|e(k)|^2]$;
- Least Squares (LS): $F[e(k)] = \frac{1}{k+1} \sum_{i=0}^{k} |e(k - i)|^2$;
- Weighted Least Squares (WLS): $F[e(k)] = \sum_{i=0}^{k} \lambda^i |e(k - i)|^2$, $\lambda$ is a constant smaller than 1;
- Instantaneous Squared Value (ISV): $F[e(k)] = |e(k)|^2$.
The MSE, in a strict sense, is only of theoretical value, since it requires an infinite amount of
information to be measured. In practice, this ideal objective function can be approximated by the
other three listed. The LS, WLS, and ISV functions differ in the implementation complexity and
in the convergence behavior characteristics; in general, the ISV is easier to implement but presents
noisy convergence properties, since it represents a greatly simplified objective function. The LS is
convenient to be used in stationary environment, whereas the WLS is useful in applications where
the environment is slowly varying.

3) Definition of the error signal $e(k)$: The choice of the error signal is crucial for the algorithm
definition, since it can affect several characteristics of the overall algorithm including computational
complexity, speed of convergence, robustness, and most importantly for the IIR adaptive filtering
case, the occurrence of biased and multiple solutions.

The minimization algorithm, the objective function, and the error signal as presented give us a
structured and simple way to interpret, analyze, and study an adaptive algorithm. In fact, almost all
known adaptive algorithms can be visualized in this form, or in a slight variation of this organization.
In the remaining parts of this book, using this framework, we present the principles of adaptive
algorithms. It may be observed that the minimization algorithm and the objective function affect
the convergence speed of the adaptive process. An important step in the definition of an adaptive
algorithm is the choice of the error signal, since this task exercises direct influence in many aspects
of the overall convergence process.

1.4 APPLICATIONS

In this section, we discuss some possible choices for the input and desired signals and how these
choices are related to the applications. Some of the classical applications of adaptive filtering are
system identification, channel equalization, signal enhancement, and prediction.

In the system identification application, the desired signal is the output of the unknown system when
excited by a broadband signal, in most cases a white-noise signal. The broadband signal is also used
as input for the adaptive filter as illustrated in Fig. 1.2. When the output MSE is minimized, the
adaptive filter represents a model for the unknown system.

The channel equalization scheme consists of applying the originally transmitted signal distorted by
the channel plus environment noise as the input signal to an adaptive filter, whereas the desired signal
is a delayed version of the original signal as depicted in Fig. 1.3. This delayed version of the input
signal is in general available at the receiver in a form of standard training signal. In a noiseless case,
the minimization of the MSE indicates that the adaptive filter represents an inverse model (equalizer)
of the channel.

In the signal enhancement case, a signal $x(k)$ is corrupted by noise $n_1(k)$, and a signal $n_2(k)$
correlated to the noise is available (measurable). If $n_2(k)$ is used as an input to the adaptive filter
with the signal corrupted by noise playing the role of the desired signal, after convergence the output
Chapter 1 Introduction to Adaptive Filtering

Figure 1.2 System identification.

Figure 1.3 Channel equalization.

Figure 1.4 Signal enhancement ($n_1(k)$ and $n_2(k)$ are noise signals correlated to each other).

error will be an enhanced version of the signal. Fig. 1.4 illustrates a typical signal enhancement setup.

Finally, in the prediction case the desired signal is a forward (or eventually a backward) version of the adaptive-filter input signal as shown in Fig. 1.5. After convergence, the adaptive filter represents a model for the input signal, and can be used as a predictor model for the input signal.

Further details regarding the applications discussed here will be given in the following chapters.
Example 1.1

Before concluding this chapter, we present a simple example in order to illustrate how an adaptive filter can be useful in solving problems that lie in the general framework represented by Fig. 1.1. We chose the signal enhancement application illustrated in Fig. 1.4.

In this example, the reference (or desired) signal consists of a discrete-time triangular waveform corrupted by a colored noise. Fig. 1.6 shows the desired signal. The adaptive-filter input signal is a white noise correlated with the noise signal that corrupted the triangular waveform, as shown in Fig. 1.7.

The coefficients of the adaptive filter are adjusted in order to keep the squared value of the output error as small as possible. As can be noticed in Fig. 1.8, as the number of iterations increase the error signal resembles the discrete-time triangular waveform shown in the same figure (dashed curve).
Figure 1.7 Input signal.

Figure 1.8 Error signal (continuous line) and triangular waveform (dashed line).