UNSUPERVISED SIGNAL PROCESSING

Channel Equalization and Source Separation

João M. T. Romano
Romis R. de F. Attux
Charles C. Cavalcante
Ricardo Suyama

CRC Press
Taylor & Francis Group

UNSUPERVISED SIGNAL PROCESSING

Channel Equalization and Source Separation

UNSUPERVISED SIGNAL PROCESSING

Channel Equalization and Source Separation

João M. T. Romano
Romis R. de F. Attux
Charles C. Cavalcante
Ricardo Suyama

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-1946-9 (Ebook-PDF)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To our families.

To the friendly members of DSPCom, past and present.

To the lovely memories of Hélio Drago Romano, Romis Attux, and Francisco Casimiro do Nascimento.

Contents

Foreword

Intelligent systems have made major contributions to the progress of science and technology in recent decades. They find applications in all technical fields and, particularly, in communications, consumer electronics, and control. A distinct characteristic is their high level of complexity, due to the fact that they capitalize on all sorts of scientific knowledge and practical know-how. However, their architecture is rather simple and can be broken down into four basic constituents, namely, sensors, actuators, signal-processing modules, and information-processing modules. The sensors and actuators constitute the interfaces of the system with its environment, while the signal-processing modules link these interfaces with the information-processing modules. Although it is generally recognized that the intelligence of the system lies in the information-processing section, intelligence is also needed in the signal-processing section to learn the environment, follow its evolutions, and cope with its adverse effects. The signal-processing modules deliver the raw data and even the most sophisticated information-processing algorithms perform badly if the quality of the raw data is poor.

From the perspective of signal processing, the most challenging problem is the connection between the signal sources and the sensors, for two main reasons. First, the transmission channels degrade the useful signals, and second, the sources have to be identified and separated from the received mixtures. Channel equalization and source separation can be dealt with separately or jointly. In any case, the quality of the corresponding processing is essential for the performance of the system, because it determines the reliability of the input data to the information-processing modules. Whenever appropriate, the problem is simplified by the introduction of learning phases, during which the algorithms are trained for optimal operation; this is called supervised processing. However, this procedure is not always possible or desirable, and continuous optimization has many advantages in terms of global performance and efficiency. Thus, we arrive at unsupervised signal processing, which is the topic of this book.

Unsupervised signal-processing techniques are described in different categories of books dealing with digital filters, adaptive methods, or statistical signal processing. But, until now, no unified presentation has been available. Therefore, this book is timely and it is an important contribution to the signal-processing literature. Moreover, unifying under a common framework the topics of blind equalization and source separation is particularly appropriate and inspiring from the perspective of both education and research.

Through the remarkable synthesis of the field it provides and the new vision it offers, this book will stimulate progress and contribute to the advent of more useful, efficient, and friendly intelligent systems.

Maurice Bellanger
Académie des Technologies de France
Paris, France

Preface

"At Cambridge, Russell had impressed on me not only the importance of mathematics but the need for a physical sense..."
Norbert Wiener, *I Am a Mathematician*

Perhaps the most fundamental motivation for writing a book is the desire to tell a story in which the author can express himself or herself and be understood by others. This sort of motivation is also present in scientific works, even if the *story* is usually narrated in formal and austere language.

The main motivation for writing this book is to tell something about the work we carry out in the Laboratory of Signal Processing for Communications (DSPCom). This includes the research topics on which we have been working as well as the way we work, which is closely related to the epigraph we chose for this preface.

The work we have developed is founded on the theory of adaptive filtering, having communication systems as the main focus of application. The natural evolution of our studies and researches led us to widen our scope of interest to themes like blind equalization, source separation, machine learning, and bio-inspired algorithms, always with the signal processing–oriented approach that is registered in the DNA of our lab.

Hence, in short, our objective in this book is to provide a unified, systematic, and synthetic presentation of what may be called the theory of *unsupervised signal processing*, with an emphasis on two topics that could be considered as the pillars [137] of such a theory: blind equalization and source separation. These two topics constitute the core of the book. They are based on the foundations of statistical and adaptive signal processing, exposed in Chapters 2 and 3, and they point to more emergent tools in signal processing, like machine learning–based solutions and bio-inspired methods, presented in Chapters 7 and 8.

Clearly, the objective described above represents a stimulating challenge for, at least, two reasons: first, gathering together all the mentioned themes was subject to the risk of dispersion or excessive verbosity, with the consequent lack of interest on the part of the readers; second, the themes of interest on their own have been specifically addressed by renowned specialists in a number of excellent books.

In this sense, we feel obliged to mention that adaptive filter theory is a well-established discipline that has been studied in depth in books like [32, 100, 139, 194, 249, 262, 303], and others. Blind equalization methods and algorithms are presented in detail in [99], and were recently surveyed in [70]. Blind source separation and related aspects like independent component analysis have been treated in very important works such as in [76, 148, 156].

Numerous authors from different scientific communities have written on topics related to machine learning and bio-inspired optimization. We must also mention inspiring works like [12, 137, 138], which deal with both blind deconvolution and separation problems.

In a certain sense, by placing the topics of this book under a similar conceptual treatment and mathematical formalism, we have tried to reap some of the important ideas disseminated and fertilized by the aforementioned authors and others we necessarily omitted in our non-exhaustive citation.

Since the genesis of the book is strongly linked to the work the authors carried out at DSPCom laboratory during more than a decade, words of thankfulness and recognition must be addressed to those who supported and inspired such work. First of all, we would like to thank all researchers, students, and assistants who worked in the lab since its establishment. It seems unreasonable to name everybody, so we decided to include all these friends in the main dedication of the book.

The first author of this book was fortunate in having Professor Maurice Bellanger, from CNAM/Paris, France, as a PhD advisor, a collaborator in many works, and an inspirational figure for us in the process of writing this book. We are grateful to many colleagues and friends for their constant support. Special thanks are due to Professor Paulo S.R. Diniz from the Federal University of Rio de Janeiro (COPPE/UFRJ) and Professor Michel D. Yacoub from FEEC/UNICAMP, first for their personal and professional example, and also for attentively motivating and pushing us to finish the work. A special mention must also be made to the memory of the late Professor Max Gerken from the University of São Paulo (POLI/USP). We also express our gratitude to Professor João C.M. Mota from the Federal University of Ceará (UFC) for many years of fruitful cooperation.

We are indebted to many colleagues in our institution, the School of Electrical and Computer Engineering at the University of Campinas (FEEC/UNICAMP, Brazil). We are particularly thankful to Professor Renato Lopes, Professor Murilo Loiola, Dr. Rafael Ferrari, Dr. Leonardo Tomazeli Duarte, and Levy Boccato for directly influencing the contents of this book, and for carefully reviewing and/or stimulating discussions about many central themes of the book. We would also like to thank Professors Fernando Von Zuben, Christiano Lyra, and Amauri Lopes, who collaborated with us by means of scientific and/or academic partnerships. Our warmest regards are reserved for Celi Pavanatti, for her constant and kind support.

Many friends and colleagues in other institutions influenced our work in different ways. For their direct technical contribution to the book or to our careers, and for their special attention in some key occasions, we would like to thank Professor Francisco R. P. Cavalcanti from UFC; Professors Maria Miranda and Cristiano Panazio from POLI/USP; Professor Leandro de Castro from Universidade Presbiteriana Mackenzie (UPM); Professor Aline Neves from Universidade Federal do ABC (UFABC); Professors Carlos A. F. da Rocha, Leonardo Resende, and Rui Seara from Universidade Federal de Santa Catarina (UFSC); Professor Jacques Szczupak from Pontifícia Universidade Católica do Rio de Janeiro (PUC); Professor Moisés Ribeiro from Universidade Federal de Juiz de Fora (UFJF); Professor Luiz C. Coradine from Universidade Federal de Alagoas (UFAL); Professor Jugurta Montalvão from Universidade Federal de Sergipe (UFS); Dr. Cynthia Junqueira from Comando Geral de Tecnologia Aeroespacial (IAE/CTA); Dr. Danilo Zanatta from NTi Audio AG; Maurício Sol de Castro from Von Braun Center; Professors Madeleine Bonnet, Hisham Abou-Kandil, Bernadette Dorizzi, and Odile Macchi, respectively, from the University Paris-Descartes, ENS/Cachan, IT-SudParis, and CNRS, in France; and Professor Tülay Adali from the University of Maryland in Baltimore, Maryland. We are especially grateful to Professor Simon Haykin from McMaster University in Canada for having given us the unforgettable opportunity of discussing our entire project during the ICA Conference at Paraty in 2009.

The acknowledgment list would certainly be incomplete without mentioning the staff of CRC Press. Our deepest gratitude must be expressed to Nora Konopka, Amber Donley, Vedavalli Karunagaran, Richard Tressider, and Brittany Gilbert for their competence, solicitude, and patience. So many thanks for believing in this project and pushing it from one end to the other!

João M. T. Romano
Romis R. de F. Attux
Charles C. Cavalcante
Ricardo Suyama

Acknowledgments

"Eu não ando só, só ando em boa companhia."
Vinicius de Moraes, Brazilian poet

The authors would like to thank their families and friends for their constant support during the process of writing this book.

João would like to express his gratitude to his lovely wife Maria Inês and children Miriam, Teresa, Filipe, Débora, Maria Beatriz, Marcelo, Ana Laura, and Daniela; to his parents, brothers, and sisters, particularly to Andy Stauffer who reviewed the first draft of this project to be sent to CRC. He would also like to thank his friends at the Cultural Center in Campinas, especially to the kind memory of Professor Francesco Langone. He is grateful to his dear friend Ilma Valadão for his support. Finally, he would like to acknowledge the motivating words of his inspirational friend, Professor José L. Boldrini.

Romis would like to thank Dilmara, Clara, and Marina for their love and for the constant happiness they bring to his life; Dina, Cecília, João Gabriel, Jecy (in memoriam), and Flora for their support on all occasions; Afrânio, Isabel, Beth, Toninho, Naby (in memoriam), Sônia, Ramsa, and his whole family for their warm affection; his students and former students—Cristina, Denis, Diogo, Everton, Filipe, George, Hassan, Hugo, Leonardo, Tiago Dias, Tiago Tavares, and Wesley—for their kindness and patience; Cristiano and Dr. Danilo for the many enriching conversations; the G6 (Alim, Carol, Daniel, Inácio, Lídia, and Theo) for the countless happy moments; all friends and colleagues from FEEC/UNICAMP; his undergraduate and graduate students for the inspiration they bring to his life; and, finally, all his friends (it would be impossible to name them here!).

Charles deeply recognizes the importance of some people in his life for the realization of this book. He expresses his gratefulness to his wife, Erika; to his children, Matheus and Yasmin; to his mother, Ivoneide; to his brothers, Kleber, Rogério, and Marcelo; and to his friend, Josué, who has always pushed him to achieve this result. For you all, my deepest gratitude; without you, nothing would have been possible and worth the effort.

Ricardo expresses his warmest gratitude to Jorge, Cecília, Bruna, Maria (in memoriam), and many others in the family, for the love and constant support; Gislaine, for all the love and patience during the final stages of this work; and all colleagues from DSPCom, for the enriching conversations and friendship.

Authors

João Marcos Travassos Romano is a professor at the University of Campinas (UNICAMP), Campinas, Sao Paulo, Brazil. He received his BS and MS in electrical engineering from UNICAMP in 1981 and 1984, respectively. In 1987, he received his PhD from the University of Paris–XI, Orsay. He has been an invited professor at CNAM, Paris; at University of Paris-Descartes; and at ENS, Cachan. He is the coordinator of the DSPCom Laboratory at UNICAMP, and his research interests include adaptive filtering, unsupervised signal processing, and applications in communication systems.

Romis Ribeiro de Faissol Attux is an assistant professor at the University of Campinas (UNICAMP), Campinas, Sao Paulo, Brazil. He received his BS, MS, and PhD in electrical engineering from UNICAMP in 1999, 2001, and 2005, respectively. He is a researcher in the DSPCom Laboratory. His research interests include blind signal processing, independent component analysis (ICA), nonlinear adaptive filtering, information-theoretic learning, neural networks, bio-inspired computing, dynamical systems, and chaos.

Charles Casimiro Cavalcante is an assistant professor at the Federal University of Ceará (UFC), Fortaleza, Ceara, Brazil. He received his BSc and MSc in electrical engineering from UFC in 1999 and 2001, respectively, and his PhD from the University of Campinas, Campinas, Sao Paulo, Brazil, in 2004. He is a researcher in the Wireless Telecommunications Research Group (GTEL), where he leads research on signal processing for communications, blind source separation, wireless communications, and statistical signal processing.

Ricardo Suyama is an assistant professor at the Federal University of ABC (UFABC), Santo Andre, Sao Paulo, Brazil. He received his BS, MS, and PhD in electrical engineering from the University of Campinas, Campinas, Sao Paulo, Brazil in 2001, 2003, and 2007, respectively. He is a researcher in the DSPCom Laboratory at UNICAMP. His research interests include adaptive filtering, source separation, and applications in communication systems.

1

Introduction

The subject of this book could be summarized by a simple scheme as that depicted in Figure 1.1.

We have an original set of data of our interest that we want, for instance, to transmit, store, extract any kind of useful information from; such data are represented by a quantity $\mathbf{s}$. However, we do not have direct access to $\mathbf{s}$ but have access only to a modified version of it, which we represent by the quantity $\mathbf{x}$. So, we can state that there is a data mapping $\mathcal{H}(\cdot)$ so that the observed data $\mathbf{x}$ are obtained by

$$\mathbf{x} = \mathcal{H}(\mathbf{s}) \tag{1.1}$$

Then our problem consists in finding a kind of inverse mapping $\mathcal{W}$ to be applied in the available data so that we could, based on a certain performance criterion, recover suitable information about the original set of data. We represent this step by another mapping that provides, from $\mathbf{x}$, what we could name an estimate of $\mathbf{s}$, represented by

$$\hat{\mathbf{s}} = \mathcal{W}(\mathbf{x}) \tag{1.2}$$

The above description is generalized on purpose so that a number of different concrete problems could fit it, with also a great variety of approaches to tackle with them. According to the area of knowledge, the aforementioned problem can be considerably relevant in signal processing, telecommunications, identification and control, pattern recognition, Bayesian analysis, and other fields. The scope of this book is clearly *signal processing oriented*, with a focus on two major problems: *channel equalization* and *source separation*. Even thus, such character of the work does not restrict the wide field of application of the theory and tools it presents.

1.1 Channel Equalization

In general terms, an equalization filter or, simply, equalizer, is a device that compensates the distortion due to an inadequate response of a given system. In communication systems, it is well known that any physical

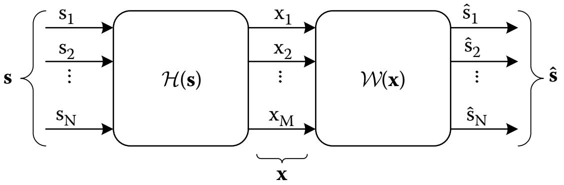


FIGURE 1.1
General scheme.

transmission channel is band-limited, i.e., it necessarily imposes distortion over the transmitted signal if such signal exceeds the allowed passband. Moreover, the channel presents additional impairments since its frequency-response in the passband is often not flat, and is also subject to noise. In the most treatable case, the channel is assumed linear and time-invariant, i.e., the output is obtained by a temporal convolution, and the noise is assumed Gaussian and additive.

In analog communications systems, channel impairments lead to a continuous-time distortion over the transmitted waveform. In digital communication, information is carried by a sequence of symbols, instead of a continuous waveform. Such symbols constitute a given transmission signal in accordance with a given modulation scheme. Hence, the noxious effect of the channel impairments in digital communications is a wrong symbol decision at the receiver.

Since information is conveyed by a sequence of symbols, it is suitable to employ a discrete-time model for the system, so that both the channel and the equalizer may be viewed as discrete-time filters, and the involved signals are numerical sequences. So, the problem may be represented by the scheme in Figure 1.2, where $s(n)$ is the transmitted signal; $\nu(n)$ is the additive noise; $x(n)$ is the received signal, i.e., the equalizer input; and $\hat{s}(n)$ is the estimate of the transmitted signal, provided by the equalizer through the mapping

$$\hat{s}(n) = \mathcal{W}\left[x(n)\right] \tag{1.3}$$

Since the channel is linear, we can characterize it by an impulse response $h(n)$ so that the mapping provided by the channel may be expressed by

$$\mathcal{H}\left[s(n)\right] = s(n) * h(n) \tag{1.4}$$

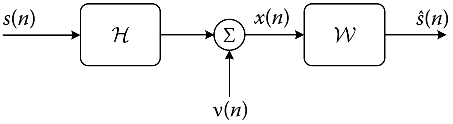


FIGURE 1.2
Equalization scheme.

where $*$ stands for the discrete-time convolution, and then

$$x(n) = s(n) * h(n) + \nu(n) \tag{1.5}$$

Clearly, the desired situation will correspond to a correct recovery of the original sequence $s(n)$, except for a delay and a constant factor, which can include phase rotation if we deal with the most general case of complex symbols. This very ideal situation is named zero-forcing (ZF) condition. As better explained further in the book, it comes from the fact that, in such conditions, all terms associated to the intersymbol interference (ISI) are "forced to zero." So, if the global system formed by the channel and the equalizer establishes a global mapping $\mathcal{G}(\cdot)$, the ZF conditions leads to

$$\mathcal{G}\left[s(n)\right] = \rho s(n - n_0) \tag{1.6}$$

where
n_0 is a delay
ρ is the constant factor

Once ρ and n_0 are known or estimated, the ideal operation under the ZF condition leads to the correct retrieval of all transmitted symbol. However, as we could expect, such a condition is not attainable in practice due to the nonideal character of $\mathcal{W}\left[\cdot\right]$ and to the effect of noise.

Hence, a more suitable approach is to search for the equalizer $\mathcal{W}\left[\cdot\right]$ that provides a minimal quantity of errors in the process of symbol recovery. By considering the stochastic nature of the transmitted information and the noise, the most natural mathematical procedure consists in dealing with the notion of probability of error.

In this sense, the first effective solution is credited to Forney [111], which considered the Viterbi algorithm for symbol recovery in presence of ISI. In its turn, the Viterbi algorithm was conceived for decoding convolutional codes in digital communications, in accordance with a maximum-likelihood (ML) criterion [300].

One year after Forney's paper, the BCJR algorithm, named after its inventors [24], was proposed for decoding, but in accordance with a maximum a posteriori (MAP) criterion. In this case, recovery was carried out symbol-by-symbol basis instead of recovering the best sequence, as in the Viterbi approach.

Once the transmitted symbols are equiprobable, the ML and MAP criteria lead to the same result. So, the Viterbi algorithm minimizes the probability of detecting a whole sequence erroneously, while the BCJR algorithm minimizes the probability of error for each individual symbol. The adaptive (supervised and unsupervised) techniques considered in this book are typically based on a symbol-by-symbol recovery.

We will refer to as Bayesian equalizer the mapping $\mathcal{W}[\cdot]$ that provides the minimal probability of error, considering symbol-by-symbol recovery. It is important to think of the Bayesian equalizer, from now, as our reference of optimality. However, due to its nonlinear character, its mathematical derivation will be postponed to Chapter 7.

Optimal equalizers derived from ML and/or MAP criteria are unfortunately not so straightforward to implement in practice [112], especially in realistic scenarios that involve real-time operation at high bit rates, nonstationary environments, etc. Taking into account the inherent difficulties of a practical communication system, the search for suitable solutions of the equalization problem includes the following steps:

- To implement the mapping $\mathcal{W}$ by means of a linear finite impulse response (FIR) filter followed by a nonlinear symbol-recovering (-decision) device.
- To choose a more feasible, although suboptimum, criterion instead of that of probability of error.
- To derive operative (adaptive, if desirable) procedures to obtain the equalizer in accordance with the chosen criterion.
- To use (as much as possible) prior knowledge about the transmitted signal and/or the channel in the aforementioned procedures.

Taking into account the above steps, the mapping $\mathcal{W}[x(n)]$ will then be accomplished by

$$y(n) = x(n) * w(n) \tag{1.7}$$

and

$$\hat{s}(n) = \Gamma\left[y(n)\right] \tag{1.8}$$

where

$w(n)$ is the equalizer impulse response
$y(n)$ is the equalizer output
$\Gamma(\cdot)$ stands for the decision device

In addition, we can now define the notion of combined response channel+equalizer as

$$g(n) = h(n) * w(n) \tag{1.9}$$

so that the ZF condition can be simply established if we define a vector $\mathbf{g}$, the elements of which are those of the sequence $g(n)$. The ZF condition holds if and only if

$$\mathbf{g} = [0, \ldots, 0, \rho, 0, \ldots, 0]^T \tag{1.10}$$

where the position of ρ in $\mathbf{g}$ is associated with the equalization delay.

As far as the criterion is concerned, the discussion is, in fact, founded on the field of estimation theory. From there, we take two useful possibilities: the minimum-squared error (MSE) and the least-squares (LS) criteria, as our main practical tools. For the operative procedure, we have two distinct possibilities: taking into account the whole transmitted sequence to obtain an optimized equalizer for this set of data (data acquisition first and equalizer optimization then) or proceeding to an adjustment of the equalizer as the data are available at the receiver (joint acquisition and optimization). In this second case, we talk about adaptive equalization. Finally, the use of a priori information is closely related to the possibility of putting into practice a mechanism of supervision or training over the system. If such a mechanism can be periodically implemented, we talk about supervised equalization, while the absence of supervision leads to the unsupervised or blind techniques.

To a certain extent, this book discusses a vast range of possible approaches to pass through these three steps, with a clear emphasis on adaptive and unsupervised methods.

We can easily observe that the problem of channel equalization, as depicted in Figure 1.2, fits the general problem of Figure 1.1, for the particular case of $M = N = 1$. Another particularization is related to the hypothesis over the transmitted signal: as a rule, it is considered to be a sequence of independent and identically distributed (i.i.d.) random variables, which belong to a finite alphabet of symbols. This last aspect clearly imposes the use of a symbol-recovering device. Regarded in this light, the problem is referred to as *SISO* channel equalization, since both the channel and the equalizer are single-input single-output filters.

Nevertheless, we can also consider a communication channel with multiple inputs and/or multiple outputs. A typical and practical case may be a wireless link with multiple antennas at the transmitter and/or at the receiver. In this book, we will specially consider the following cases, to be discussed in Chapter 4:

- A single-input multiple-output (SIMO) channel with a multiple-input single-output (MISO) equalizer, which corresponds to $N = 1$ and $M > 1$ in Figure 1.1.
- A multiple-input multiple-output (MIMO) channel with a multiple-input multiple-output (MIMO) equalizer, which corresponds to $N > 1$ and $M > 1$ in Figure 1.1.

1.2 Source Separation

The research work on SISO blind equalization has been particularly intense during the 1980s. At this time, another challenging problem in signal processing was proposed, that of blind source separation (BSS). In general terms, such a problem can be simply explained by the classical example known as cocktail party phenomenon, where a number of speakers communicate at the same time in the same noisy environment. In order to focus the attention in a specific speaker s_1, a given receiver must retrieve the corresponding signal from a mixture of all signals $\{s_1, \ldots, s_N\}$, where N is the number of speakers. Despite the human ability in performing this task, a technical solution for providing blind separation was unknown until the work of Hérault et al., in 1985 [144].

As stated above, the BSS problem also fits in the scheme of Figure 1.1. The possibility of obtaining proper solutions will depend on the hypothesis we consider for the mapping $\mathcal{H}(\cdot)$ and for the set of original signals, or sources, $\mathbf{s}$. The most tractable case emerges from the following assumptions:

- The mapping $\mathcal{H}(\cdot)$ stands for a linear and memoryless system, with $M = N$.
- The sources $\{s_1, \ldots, s_N\}$ are assumed to be mutually independent signals.
- There is, at most, one Gaussian source.

The main techniques for solving BSS under these assumptions come from the principle of independent component analysis (ICA) [74]. Such techniques are based on searching for a separating system $\mathcal{W}(\cdot)$, the parameters of which are obtained in accordance of a given criterion that imposes statistical independence between the set of outputs $\hat{\mathbf{s}}$. As pointed out in [137], ICA may be viewed as an extension of the well-known principal component analysis (PCA), which deals only with the second-order statistics of the involved signals.

Although blind equalization and source separation problems have originated independently and in somewhat distinct scientific communities, we can clearly observe a certain "duality" between them:

- In SISO channels, the output is a linear combination (temporal convolution) of the elements of the transmitted signal with additive Gaussian noise. In BSS, the set of outputs comes from the linear mixture of signals, among which one can be Gaussian.
- In SISO equalization, we try to recover a sequence of independent symbols that correspond to the transmitted signal. In BSS, we search for a set of independent variables that correspond to original sources.
- In both cases, dealing with second-order statistics is not sufficient: the output of a SISO channel may be whitened, for instance, by a prediction-error filter, while the outputs of the mixing system may be decorrelated by a PCA procedure. However, as we will stress later in the book, neither of these procedures can guarantee a correct retrieval.

The above considerations will become clearer, and will be more rigorously revisited, in the sequence of the chapters. Nevertheless, it is worth remarking these points in this introduction to illustrate the interest in bringing unsupervised equalization and source separation to a common theoretical framework.

On the other hand, BSS can become a more challenging problem as the aforementioned assumptions are discarded. The case of a mixing system with memory corresponds to the more general problem of convolutive mixtures. Such a problem is rather similar to that of MIMO equalization. As a rule in this book, we consider convolutive BSS as a more general problem since, in MIMO channel equalization, we usually suppose that the transmitted signals have the same statistical distributions and belong to a finite alphabet. This is not at all the case in other typical applications of BSS.

If the hypothesis of linear mixing is discarded, the solution of BSS problems will require special care, particularly in applying ICA. Such a solution may involve the use of nonlinear devices in the separating systems, as done in the so-called post-nonlinear model. It is worth mentioning that nonlinear channels can also be considered in communication and different approaches have been proposed for nonlinear equalization, including the widely known decision feedback equalizer (DFE). Overall, our problem will certainly become more intricate when nonlinear mappings take place in $\mathcal{H}(\cdot)$ and/or in $\mathcal{W}(\cdot)$, as we will discuss in more detail in Chapter 6.

Furthermore, other scenarios in BSS deserve the attention of researchers, as those of underdetermined mixtures, i.e., in scenarios in which $M < N$ in Figure 1.1; correlated sources; sparse sources, etc.

1.3 Organization and Contents

We have organized the book as follows:

Chapter 2 reviews the fundamental concepts concerning the characterization of signals and systems. The purpose of this chapter is to emphasize some notions and tools that are necessary to the sequence of the book. For the sake of clarity, we first deal with deterministic concepts and then we introduce statistical characterization tools. Although many readers could be familiar with these subjects, we provide a synthetic presentation of the following topics: signals and systems definitions and main properties; basic concepts of discrete-time signal processing, including the sampling theorem; fundamentals of probability theory, including topics like cumulants, which are particularly useful in the context of unsupervised processing; a review on stochastic processes with a specific topic on discrete-time random signals; and, finally, a section on estimation theory.

In order to establish the foundations of unsupervised signal processing, we present in Chapter 3 the theory of optimal and adaptive filtering in the classic scenario of linear and supervised processing. As already commented, many books are devoted to this rich subject and present it in a more exhaustive fashion. We opt for a brief and, to a certain extent, personal presentation that facilitates the introduction of the central themes of the book. First, we discuss three emblematic problems in linear filter theory: identification, deconvolution, and prediction. From there, the specific case of channel equalization is introduced. Then, as usually done in the literature, we present the Wiener filtering theory as the typical solution for supervised processing and a paradigm for adaptive procedures. The sections on supervised adaptive filtering discuss the celebrated LMS and RLS algorithms, and also the use of structures alternative to the linear FIR filter. Moreover, in Chapter 3 we introduce the notion of optimal and adaptive filtering without a reference signal, as a first step to consider blind techniques. In this context, we discuss the problem of constrained filtering and revisit that of prediction, indicating some relationships between linear prediction and unsupervised equalization.

After establishing the necessary foundations in Chapters 2 and 3, the subject of unsupervised equalization itself is studied in Chapter 4, which deals with single-input single-output (SISO) channels, and in Chapter 5, in which the multichannel case is considered.

Chapter 4 starts with a general discussion on the problem of unsupervised deconvolution, of which blind equalization may be viewed as a particular case. After introducing the specific problem of equalization, we state the two fundamental theorems: Benveniste–Goursat–Ruget and Shalvi–Weinstein. Then we discuss the main adaptive techniques: the so-called Bussgang algorithms that comprise different LMS-based blind techniques, the Shalvi–Weinstein algorithm, and the super-exponential. Among Bussgang techniques, special attention is given to the decision-directed (DD) and Godard/CMA approaches, due to their practical interest in communications schemes. We discuss important aspects about the equilibrium solutions and convergence of these methods, having the Wiener MSE surface as a benchmark for performance evaluation. Finally, based on a more recent literature, we present some results concerning the relationships between constant-modulus, Shalvi–Weinstein, and Wiener criteria.

The problem of blind equalization is extended to the context of systems with multiple inputs and/or outputs in Chapter 5. First, we state some theoretical properties concerning these systems. Then we discuss single-input multiple-output (SIMO) channels, which may be engendered, for instance, by two practical situations: temporal oversampling of the received signal or the use of multiple antennas at the receiver. In the context of SIMO equalization, we discuss equalization conditions in the light of Bezout's identity and the second-order methods for blind equalization. Afterward, we turn our attention to the most general scenario, that of multiple-input multiple-output (MIMO) channels. In such case, special attention is given to multiuser systems, the importance of which is notorious in modern wireless communications.

Chapter 6 deals with blind source separation (BSS), the other central subject for the objectives of this book. We start this chapter by stating the main models to be used and the standard case to be considered first, that of a linear, instantaneous, and noiseless mixture. Then, we introduce a tool of major interest in BSS: the independent component analysis (ICA). The first part of Chapter 6 is devoted to the main concepts, criteria, and algorithms to perform ICA. Afterward, we deal with alternative techniques that exploit prior information as, in particular, the nonnegative and the sparse component decompositions. Then, we leave the aforementioned standard case to consider two relevant problems in BSS: those of convolutive and nonlinear mixtures. Both of them can be viewed as open problems with significant research results in the recent literature. So we focus our brief presentation on some representative methods with emphasis on the so-called post-nonlinear model.

Chapters 4 through 6 establish the fundamental core of the book, as we try to bring together blind equalization and source separation under the same conceptual and formal framework. The two final chapters consider more emergent techniques that can be applied in the solution of those two problems.

The synergy between the disciplines of machine learning and signal processing has significantly increased during the last decades, which is attested by the several regular and specific conferences and journal issues devoted to the subject. From the standpoint of this book, it is quite relevant that a nonnegligible part of this literature is related to unsupervised problems. Chapter 7 presents some instigating connections between nonlinear filtering, machine learning techniques, and unsupervised processing. We start by considering a classical nonlinear solution for adaptive equalization—the DFE structure—since this remarkably efficient approach can be equally used in supervised and blind contexts. Then we turn our attention to more sophisticated structures that present properties related to the idea of universal approximation, like Volterra filters and artificial neural networks. For that, we previously revisit equalization within the framework of a classification problem and introduce an important benchmark in digital transmission: the Bayesian equalizer, which performs a classification task by recovering the transmitted symbols in accordance with the criterion of minimum probability of error. Finally, we discuss two classical artificial neural networks: multilayer perceptron (MLP) and radial basis function (RBF) network. The training process of these networks is illustrated with the aid of classical results, like the backpropagation algorithm and the k-means algorithm.

The methods and techniques discussed all through this book are issued, after all, from a problem of optimization. The solutions are obtained, as a rule, by the minimization or maximization of a given criterion or cost-function. The bio-inspired optimization methods discussed in Chapter 8, however, are part of a different paradigm, as they are founded on a number of complex processes found in nature. These methods are generally characterized by a significant global search potential and do not require significant a priori information about the problem to be solved, which encourages application, for instance, in nonlinear and/or unsupervised contexts. Chapter 8 closes the book by considering this family of techniques, which are finding increasing applications in signal processing. Given the vastness of the subject, we limit our discussion to three potentially suitable approaches, taking into account our domain of interest: genetic algorithms, artificial immune systems, and particle swarm optimization methods.

The book presents enough material for a graduate course, since blind techniques are increasingly present in graduate programs, and can also be used as a complementary reference for undergraduate students. According to the audience, Chapter 2 can be skipped, and even some topics of Chapter 3, if the students have the possibility of attending a specific course on *adaptive filtering theory*. Furthermore, the content of Chapters 7 and 8 can be adapted to the audience and also serves as a complementary material for courses on machine learning and/or optimization. Overall, it is worth emphasizing that a course on unsupervised signal processing theory, comprising blind equalization and source separation, must not be organized in a rigid way, but following the interests of different institutions.

Finally, it is worth emphasizing that adaptive filtering, unsupervised equalization, source separation, and related themes present a number of recent results and open problems. Necessarily, and to preserve the main focus of this book, some of them were omitted or not dealt with in depth.

2

Statistical Characterization of Signals and Systems

The statistical characterization of signals and systems provides an important framework of concepts and mathematical tools that are fundamental to the modern theory of filtering and signal processing. In signal theory, we denote by statistical signal processing the field of study that treats signals as stochastic processes. The word stochastic is etymologically associated with the notion of randomness. Even though such notion gives rise to different interpretations, in our field of study, randomness is related to the concept of uncertainty. Uncertainty on its turn is present in the essence of information signals in their different forms as well as in the several types of disturbances that can affect a system.

The subject of statistical characterization of signals and systems is really extensive and has been built along more than two centuries, as a result of classical works on statistical inference, linear filtering, and information theory. Nevertheless, the purpose of this chapter is rather objective and, in a way, unpretentious: to present the basic foundations and to emphasize some concepts and tools that are necessary to the understanding of the next chapters. With this aim in mind we have chosen five main topics to discuss:

- Section 2.1 is devoted to the basic theory of *signals and systems*. For the sake of systemizing such theory, we first consider signals that do not have randomness in their nature.
- Section 2.2 specifically considers *discrete-time signal processing*, since most methods to be presented in the book tend to be implemented using this approach.
- Section 2.3 discusses the foundations of the *probability theory* in order to introduce the suitable tools to deal with random signals. The main definitions and properties are exposed.
- Section 2.4 then deals with the notion of *stochastic processes* together with some useful properties. An appendix on the correlation matrix properties complements the subject.
- Finally, Section 2.5 discusses the main concepts of *estimation theory*, a major area of statistical signal processing with strong connections with that of optimal filtering, which is the subject of the following chapter.

Historical Notes

The mathematical foundations of the theory of signals and systems have been established by eminent mathematicians of the seventeenth and eighteenth centuries. This coincides, in a way, with the advent of calculus, since the representation of physical phenomena in terms of functions of continuous variables and differential equations gave rise to an appropriate description of the behavior of continuous signals and systems. Furthermore, as mentioned by Alan Oppenheim and Ronald Schafer [219], the classical works on numerical analysis developed by names like Euler, Bernoulli, and Lagrange sowed the seeds of discrete-time signal processing.

The bridge between continuous- and discrete-time signal processing was theoretically established by the sampling theorem, introduced in the works of Harry Nyquist in 1928, D. Gabor in 1946, and definitely proved by Claude Shannon in 1949. Notwithstanding this central result, signal processing was typically carried out by analog systems and in a continuous-time framework, basically due to performance limitations of the existing digital machines. Simultaneously with the development of computers, a landmark result appeared: the proposition of the fast Fourier transform algorithm by Cooley and Tukey in 1965. Indeed, this result has been considered to be one of the most important in the history of discrete-time signal processing, since it opened a perspective of practical implementation of many other algorithms in digital hardware.

Two other branches of mathematics are fundamental in the modern theory of signals and systems: functional analysis and probability theory. Functional analysis is concerned with the study of vector spaces and operators acting upon them, which are crucial for different methods of signal analysis and representation. From it is derived the concept of Hilbert space, the denomination of which is due to John von Neumann in 1929, as a recognition of the work of the great mathematician David Hilbert. This is a fundamental concept to describe signals and systems in a transformed domain, including the Fourier transform, a major tool in signal processing, the principles of which had been introduced one century before by Jean-Baptiste Joseph Fourier.

Probability theory allows extending the theory of signals and systems to a scenario where randomness or incertitude is present. The creation of a mathematical theory of probability is attributed to two great French mathematicians, Blaise Pascal and Pierre de Fermat, in 1654. Along three centuries, important works were written by names like Jakob Bernoulli, Abraham de Moivre, Thomas Bayes, Carl Friedrich Gauss, and many others. In 1812, Pierre de Laplace introduced a host of new ideas and mathematical techniques in his book *Théorie Analytique des Probabilités* [175].

Since Laplace, many authors have contributed to developing a mathematical probability theory precise enough for use in mathematics as well as suitable to be applicable to a wide range of practical problems. The Russian mathematician Andrei Nikolaevich Kolmogorov established a solid landmark in 1933, by proposing the axiomatic approach that forms the basis of modern probability theory [169]. A few years later, in his classical paper [272], Shannon made use of probability in the definition of entropy, in order to "play a central role in information theory as measures of information, choice and uncertainty." This fundamental link between uncertainty and information raised many possibilities of using statistical tools in the characterization of signals and systems within all fields of knowledge concerned with information processing.

2.1 Signals and Systems

Information exchange has been a vital process since the dawn of mankind. If we consider for a moment our routine, we will probably be able to point out several sources of information that belong to our everyday life. Nevertheless, "information in itself" cannot be transmitted. A message must find its proper herald; this is the idea of signal.

We shall define a signal as a function that bears information, while a system shall be understood as a device that produces one or more output signals from one or more input signals. As mentioned in the introduction of this chapter, the proper way to address signals and systems in the modern theory of filtering and signal processing is by means of their statistical characterization, due to the intrinsic relationships between information and randomness. Nevertheless, for the sake of systemizing such theory, we first consider signals that do not have incertitude in their nature.

2.1.1 Signals

In simple terms, a signal can be defined as an information-bearing function. The more we probe into the structure of a certain signal, the more information we are able to extract. A cardiologist can find out a lot about your health by simply glancing at an ECG. Conversely, someone without an adequate training would hardly avoid a commonplace appreciation of the same data set, which leads us to a conclusion: signals have but a small practical value without the efficient means to interpret their content. From this it is easy to understand why so much attention has been paid to the field of signal analysis.

Mathematically, a function is a mapping that associates elements of two sets—the domain and the codomain. The domain of a signal is usually, although not necessarily, related to the idea of time flow. In signal processing, there are countless examples of temporal signals: the electrical stimulus produced by a microphone, the voltage in a capacitor, the daily peak temperature profile of a given city, etc. In a number of cases, signals can be, for instance, functions of space: the gray intensity level of a monochrome image, the set of measures provided by an array of sensors, etc. Also, spatiotemporal signals may be of great interest, the most typical example being a video signal, which is a function of a two-dimensional domain: space and time.

In this book, we deal much more frequently with temporal signals, but some cases of space-time processing, like the use of antenna array in particular channels, are also relevant to the present work. Anyway, it is interesting to expose some important properties concerning the nature of a signal as well as ways of classifying and characterizing them.

2.1.1.1 Continuous- and Discrete-Time Signals

Insofar as the domain of temporal signals is concerned, there are two possibilities of particular relevance: to establish a continuum or an integer set of time values. In the former case, the chosen domain engenders a continuous-time signal, which is mathematically described by a function of a continuous variable, denoted by $x(t)$. Conversely, if time-dependence is expressed by means of a set of integer values, it gives rise to a discrete-time signal, which is mathematically described by a numerical sequence, denoted by $x(n)$. For instance, a signal received by a microphone or an antenna can be assumed to be a continuous-time signal, while a daily stock quote is a discrete-time signal.

2.1.1.2 Analog and Digital Signals

A signal whose amplitude can assume any value in a continuous range is an analog signal, which means that it can assume an infinite number of values. On the other hand, if the signal amplitude assumes only a finite number of values, it is a digital signal.

Figure 2.1 illustrates examples of different types of signals. It should be clear that a continuous-time signal is not necessarily an analog signal, as well as a discrete-time signal may not be digital. The terms continuous-time and discrete-time refer to the nature of the signals along the time, while the terms analog and digital qualify the nature of the signal amplitude. This is shown in Figure 2.1.

2.1.1.3 Periodic and Aperiodic/Causal and Noncausal Signals

A signal $x(t)$ is said to be periodic if, for some positive constant T,

$$x(t) = x(t + T) \tag{2.1}$$

for all t. The smallest value of T for which (2.1) holds is the period of the signal. Signals that do not exhibit periodicity are termed aperiodic signals.

From (2.1), we can notice that a periodic signal should not change if shifted in time by a period T. Also, it must start at $t = -\infty$, otherwise, it would not be

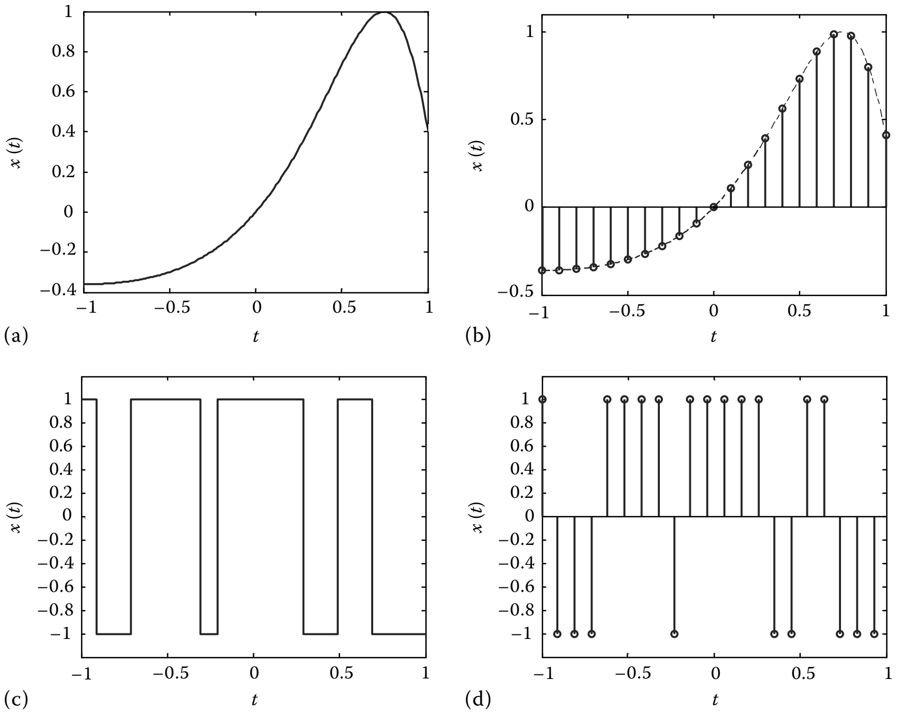


FIGURE 2.1
Examples of analog/digital and continuous-time/discrete-time signals: (a) analog continuous-time signal, (b) analog discrete-time signal, (c) digital continuous-time signal, (d) digital discrete-time signal.

possible to respect the condition expressed in (2.1) for all t. Signals that start at $t = -\infty$ and extend until $t = \infty$ are denoted infinite duration signals.

In addition to these definitions, it is interesting to establish the difference between a causal and a noncausal signal. A signal is causal if

$$x(t) = 0, \quad t < 0 \tag{2.2}$$

and said to be noncausal if the signal starts before $t = 0$.

It is worth mentioning that all definitions also apply for discrete-time signals.

2.1.1.4 *Energy Signals and Power Signals*

An energy signal is a signal that has finite energy, i.e.,

$$\int_{-\infty}^{\infty} |x(t)|^2 \, dt < \infty \tag{2.3}$$

A signal with finite nonzero power, i.e.,

$$\lim_{\alpha\to\infty} \frac{1}{\alpha} \int_{-\alpha/2}^{\alpha/2} |x(t)|^2 \, dt < \infty \tag{2.4}$$

is called a power signal.

All practical signals present finite energy and, thus, are energy signals. Another interesting fact is that a power signal should necessarily be an infinite duration signal, otherwise, its average energy would tend to zero within a long enough time interval.

2.1.1.5 Deterministic and Random Signals

A deterministic signal is a signal whose physical description, either in a mathematical or in a graphical form, is completely known. Conversely, a signal whose values cannot be precisely predicted but are known only in terms of a probabilistic description is a random signal.

2.1.2 Transforms

In daily life, our senses are exposed to all kinds of information when we move, or simply as long as time flows. Since time (and in a way this is also true for space) constitutes the most natural domain in which we observe information, it is also a natural standpoint to represent and analyze a signal, but it is not the only possibility. Sometimes, a great deal of insight on the characteristics of an information-bearing function can be gained by translating it into another support input space.

We can understand a transform as a mapping that establishes a one-to-one relationship between the representations of a given signal in two distinct domains. As a rule, a transform is employed when the present domain wherein a signal is represented is not the most favorable to the study of one or more of its relevant aspects. The domain of representation is strongly related with the mathematical concept of complete orthogonal basis. For instance, if we define the unit impulse function $\delta(t)$ (also known as the *Dirac delta function*) as

$$\delta(t) = 0 \quad t \neq 0 \tag{2.5}$$

$$\int_{-\infty}^{\infty} \delta(\tau) d\tau = 1 \tag{2.6}$$

it is interesting to observe that a continuous-time signal can be written as

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau \tag{2.7}$$

A similar representation can be obtained for discrete-time signals:

$$x(n) = \sum_{-\infty}^{\infty} x(k)\delta(n-k) \tag{2.8}$$

where $\delta(n)$ denotes the discrete-time unit impulse function (also known as the *Kronecker delta function*), and is defined as

$$\delta(n) = \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

It comes, and this is even more evident in the discrete case, that the signal of interest is a linear combination of shifted unit impulse functions. In this sense, these shifted functions can be regarded as a basis for representing the signal.

A change of representation domain corresponds to a change in the basis over which the signal is decomposed. As mentioned before, this can be very important in order to study some characteristics and properties of the signal that are not directly observed in the form, for instance, of a temporal function or sequence. In the classical theory of signal and systems, representation by the complete orthogonal basis composed by complex exponentials deserves special attention. For purely imaginary exponents, the complex exponential functions and sequences are directly associated with the physical concept of frequency, and such representation gives rise to the Fourier transform. For general complex exponents, the corresponding decomposition gives rise to the Laplace transform, in the continuous case, and to the z-transform, in the discrete case, which are both crucial for the study of linear systems. These four important cases are depicted in the sequel.

2.1.2.1 The Fourier Transform of Continuous-Time Signals

As mentioned earlier, the Fourier transform corresponds to the projection of a signal $x(t)$ onto a complete orthogonal basis composed of complex exponentials $\exp(j2\pi ft)$. It is defined as

$$X(f) = \int_{-\infty}^{\infty} x(t)\exp(-j2\pi ft)dt \tag{2.10}$$

while the inverse Fourier transform is given by

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp(j2\pi ft) df \tag{2.11}$$

2.1.2.2 The Fourier Transform of Discrete-Time Signals

The discrete-time counterpart of the Fourier transform correspond to the projection of the sequence $x(n)$ onto an orthogonal basis composed of complex exponentials $\exp(j2\pi fn)$, and is defined as

$$X(\exp(j2\pi f)) = \sum_{n=-\infty}^{\infty} x(n) \exp(-j2\pi fn) \tag{2.12}$$

while the inverse Fourier transform is given by

$$x(n) = \int_{-1/2}^{1/2} X(\exp(j2\pi fn)) \exp(j2\pi fn) df \tag{2.13}$$

2.1.2.3 The Laplace Transform

The basic idea behind the Laplace transform is to build an alternative representation $X(s)$ of a continuous-time signal $x(t)$, from a basis of complex exponentials:

$$X(s) = \int_{-\infty}^{\infty} x(t) \exp(-st) dt \tag{2.14}$$

where $s = \sigma + j2\pi f$. The set of values of s for which the integral shown in (2.14) converges is called region of convergence (ROC) of the Laplace transform.

The inverse Laplace transform is then given by

$$x(t) = \frac{1}{2\pi j} \oint_C X(s) \exp(st) ds \tag{2.15}$$

where C is a suitable contour path.

2.1.2.4 The z-Transform

The z-transform can be understood as the equivalent of the Laplace transform in the context of discrete-time signals. The transform $X(z)$ of a sequence $x(n)$ is defined by

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2.16}$$

where $z = \exp(\sigma + j\,2\pi f)$. The ROC of the z-transform is defined as the values of z for which the summation presented in (2.16) converges.

The inverse z-transform is defined as

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \tag{2.17}$$

where the integral must be evaluated in a path C that encircles all of the poles of $X(z)$.

It is worth mentioning that Equations 2.14 and 2.16 correspond to the so-called bilateral Laplace and z-transforms, which are the most generic representations. For causal signals, it is useful to consider the unilateral transforms, in which the integral and the discrete sum start from zero instead of $-\infty$.

2.1.3 Systems

Having in mind the definition of signal presented in the beginning of Section 2.1.1, we may alternatively define a system as an information-processing device. In Figure 2.2, we present a schematic view of a system.

A system can be fully characterized by its input–output relation, i.e., by the mathematical expression that relates its outputs to its inputs. Assuming that the operator $S[\cdot]$ represents the mapping performed by the system, we may write

$$\mathbf{y} = S[\mathbf{x}] \tag{2.18}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors, respectively. It is interesting to analyze some important classes of systems and the properties that characterize them.

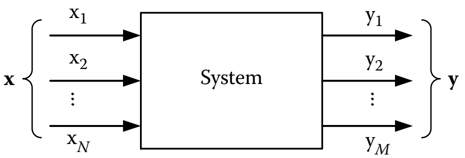


FIGURE 2.2
Schematic view of a system.

2.1.3.1 SISO/SIMO/MISO/MIMO Systems

This classification is based on the number of input and output signals of a system:

- SISO (single-input single-output) systems have a single input signal and a single output signal. Therefore, **x** and **y** become scalars.
- SIMO (single-input multiple-outputs) systems have a single input signal and more than one output signal.
- MISO (multiple-input single-output) systems have multiple input signals and a single output signal.
- Finally, MIMO (multiple-input multiple-output) systems have multiple input and output signals, and form the most general of the four classes.

Throughout the book, the reader will have the opportunity of considering the differences between these classes of systems, the importance of which is patent in modern signal processing techniques.

2.1.3.2 Causal Systems

If the system output depends exclusively on present and past values of the input, the system is said to be *causal*. In other words, causality means that the output of a system at a given instant is not influenced by future values of the input.

When we consider real-time applications, causality will certainly hold. However, when we manipulate acquired data, noncausal systems are acceptable, and may even be desirable in some cases.

2.1.3.3 Invertible Systems

When it is possible to build a mapping that recovers the input signals of a given system from its output, we say that such a system is invertible. This means that it is possible to obtain **x** from **y** using an inverse system cascaded with the original one. This notion will be revisited when we analyze the problems of equalization and source separation.

2.1.3.4 Stable Systems

Stability is also a major concern in system analysis. We shall assume that a system is stable if the response to a bounded input is also bounded. In simple words, if the input signal does not diverge to infinity, the output will not diverge as well. Stability is a common feature in real-world systems, which we suppose to be restricted by conservation laws, but the same may not occur in some mathematical models and algorithms.

2.1.3.5 Linear Systems

In system theory, it is often convenient to introduce some classes of possible operators. A very relevant distinction is established between linear and nonlinear systems. *Linear systems* are those whose defining $S[\cdot]$ operator obeys the following *superposition principle*:

$$S[k_1\mathbf{x}_1 + k_2\mathbf{x}_2] = k_1S[\mathbf{x}_1] + k_2S[\mathbf{x}_2] \tag{2.19}$$

The idea of superposition can be explained in simple terms: the response to a linear combination of input stimuli is the linear combination of the individual responses. Conversely, a nonlinear system is simply one that does not obey this principle.

2.1.3.6 Time-Invariant Systems

Another important feature is time-invariance. A system is said to be time-invariant when its input–output mapping does not vary with time. When the contrary holds, the system is said to be time-variant. Since this characteristic makes the system easier to be dealt with in mathematical terms, most models of practical systems are, with different degrees of fidelity, time-invariant.

2.1.3.7 Linear Time-Invariant Systems

A very special class of systems is that formed by those that are both linear and time-invariant (linear time-invariant, LTI). These systems obey the superposition principle and have an input–output mapping that does not vary with time. The combination of these desirable properties gives rise to the following mathematical result.

Suppose that $x(t)$ and $y(t)$ are, respectively, the input and the output of a continuous-time LTI SISO system. In such case,

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \tag{2.20}$$

where $h(t)$ is the system impulse response, which is the system output when $x(t)$ is equal to the Dirac delta function $\delta(t)$. The symbol $*$ denotes that the output $y(n)$ is the result of the convolution of $x(t)$ with $h(t)$.

Analogously, if $x(n)$ and $y(n)$ are, respectively, the input and the output of a discrete-time LTI SISO system, it holds that

$$y(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{2.21}$$

where $h(n)$ is the system impulse response, i.e., the system output when $x(n)$ is equal to the Kronecker delta function $\delta(n)$. Once more, the symbol $*$ stands for convolution.

2.1.4 Transfer Function and Frequency Response

An important consequence of the fact that the input and the output of a continuous-time LTI SISO system are related by a convolution integral is that their Laplace transforms will be related in a very simple way:

$$Y(s) = H(s)X(s) \tag{2.22}$$

where
$Y(s)$ and $X(s)$ are, respectively, the Laplace transforms of the output and the input
$H(s)$ is the transform of the system impulse response, the so-called transfer function

This means that the input–output relation of an LTI system is the result of a simple product in the Laplace domain.

If the ROCs of $X(s)$, $Y(s)$, and $H(s)$ include the imaginary axis, expression (2.22) can be promptly particularized to the domain of the Fourier analysis. In this case, the following holds:

$$Y(f) = H(f)X(f) \tag{2.23}$$

where
$Y(f)$ and $X(f)$ are, respectively, the Fourier transforms of the output and the input
$H(f)$ is the transform of the system impulse response, which is called frequency response

It is possible to understand several key features of a given LTI system simply by studying the functions $H(s)$ and $H(f)$. For instance, to know the frequency response of a system is the key to understanding how it responds to stimuli at any frequency of the spectrum, and how an input signal characterized by certain frequency content will be processed by it.

The extension to the discrete-time domain is straightforward. If $Y(z)$ and $X(z)$ are the z-transforms of two discrete-time signals related by an LTI system, it is possible to write

$$Y(z) = H(z)X(z) \tag{2.24}$$

and, if the ROCs of $X(z)$, $Y(z)$, and $H(z)$ include the unit circle, expression (2.24) reduces to

$$Y\left[\exp\left(j2\pi f\right)\right] = H\left[\exp\left(j2\pi f\right)\right]X\left[\exp\left(j2\pi f\right)\right] \tag{2.25}$$

2.2 Digital Signal Processing

Discrete-time signals can be characterized and stored very easily. This relies on a very relevant feature of discrete-time signals: given a finite time interval, there is a finite set of values that fully characterize a sequence, whereas the same does not hold for a continuous-time signal. This essential difference is a reflex of the profound structural divergences between the domains of these classes of information-bearing functions.

The world of digital computers excels in storage capacity and potential of information processing, and is essentially a "discrete-time world." Therefore, it is not surprising that digital signal processing is a widespread tool nowadays. Nevertheless, it is also clear that many of our physical models are inherently based on continuous-time signals. The bridge between this "real world" and the existing digital tools is established by the sampling theorem.

2.2.1 The Sampling Theorem

The idea of sampling is very intuitive, as it is closely related to the notion of measure. When we measure our height or weight, we are, in a certain sense, sampling the continuous-time signal that expresses the time-evolution of these variables. In the context of communications, the sampling process produces, from a continuous-time signal, a representative discrete-time signal that lends itself to proper digital processing and storage. Conditions for equivalent representation and perfect reconstruction of the original signal from its samples were achieved through the sampling theorem, proposed by Harry Nyquist (1926), D. Gabor (1946), and Claude Shannon (1949), and are related to two requirements:

1. The continuous-time signal must be band-limited, i.e., its Fourier spectrum must be null for $f > f_M$.
2. The sampling rate, i.e., the inverse of the time-spacing T_S of the samples must be higher than or equal to $2f_M$.

Given these conditions, we are ready to enunciate the sampling theorem [219]

THEOREM 2.1 (Sampling Theorem)

If $x(t)$ is a signal that obeys requirement 1 above, it may be perfectly determined by its samples $x(nT_S)$, n integer, if T_S obeys requirement 2. ■

If these requirements are not complied with, the reconstruction process will be adversely affected by a phenomenon referred to as aliasing [219].

2.2.2 The Filtering Problem

There are many practical instances in which it is relevant to process information, i.e., to treat signals in a controlled way. A straightforward approach to fulfill this task is to design a filter, i.e., a system whose input–output relation is tailored to comply with preestablished requirements. The project of a filter usually encompasses three major stages:

- Choice of the filtering structure, i.e., of the general mathematical form of the input–output relation.
- Establishment of a filtering criterion, i.e., of an expression that encompasses the general objectives of the signal processing task at hand.
- Optimization of the cost function defined in the previous step with respect to the free parameters of the structure defined in the first step.

It is very useful to divide the universe of discrete-time filtering structures into two classes: linear and nonlinear. There are two basic types of linear digital filters: finite impulse response filters (FIR) and infinite impulse response filters (IIR). The main difference is that FIR filters are, by nature, feedforward devices, whereas IIR filters are essentially related to the idea of feedback.

On the other hand, nonlinearity is essentially a negative concept. Therefore, there are countless possible classes of nonlinear structures, which means that the task of treating the filtering problem in general terms is far from trivial.

Certain classes of nonlinear structures (like those of neural networks and polynomial filters, which will be discussed in Chapter 7) share a very relevant feature: they are derived within a mathematical framework related to the idea of universal approximation. Consequently, they have the ability of producing virtually any kind of nonpathological input–output mapping, which is a remarkable feature in a universe as wide as that of nonlinear filters.

A filtering criterion is a mathematical expression of the aims subjacent to a certain task. The most direct expression of a filtering criterion is its associated cost function, the optimization of which leads to the choice and adaptation of the free parameters of the chosen structure.

When both the structure and an adequate cost function are chosen, there remains the procedure of optimizing the function with respect to the free parameters of the filtering device. Although there are many possible approaches, iterative techniques are quite usual in practical applications for, at least, two reasons:

- They avoid the need for explicitly finding closed-form solutions, which, in some cases, can be rather complicated even in static environments.
- Their dynamic nature suits very well the idea of adaptation, which is essential in a vast number of real-world applications.

Adaptation will be a crucial idea in the sequel of this text and, as we will see, the derivation of a number of adaptive algorithms depends on some statistical concepts to be introduced now.

2.3 Probability Theory and Randomness

Up to this moment, signals have been completely described by mathematical functions that generate information from a support input space. This is the essence of deterministic signals. However, this direct mapping between the input and output space cannot be established if uncertainties exist. In such case, the element of randomness is introduced and probabilistic laws must be used to represent information. Thus, it is of great interest to review some fundamental concepts of probability theory.

2.3.1 Definition of Probability

Probability is essentially a measure to be employed in a random experiment. When one deals with any kind of random experiment, it is often necessary to establish some conditions in order that its outcome be representative of the phenomenon under study. In more specific terms, a random experiment should have the following three features [135]:

1. The experiment must be repeatable under identical conditions.
2. The outcome w_i of the experiment on any trial is unpredictable before its occurrence.

3. When a large number of trials is run, statistical regularity must be observed in the outcome, i.e., an average behavior must be identified if the experiment is repeated a large number of times.

The key point of analyzing a random experiment lies exactly in the representation of the statistical regularity. A simple measure thereof is the so-called relative frequency. In order to reach this concept, let us define the following:

- The space of outcomes Ω, or sample space, which is the set of all possible outcomes of the random experiment.
- An event A, which is an element, a subset or a set of subsets of Ω.

Relative frequency is the ratio between the number of occurrences of a specific event and the total number of experiment trials. If an event A occurs $N(A)$ times over a total number of trials N, this ratio obeys

$$0 \leq \frac{N(A)}{N} \leq 1 \tag{2.26}$$

We may state that an experiment exhibits statistical regularity if, for any given sequence of N trials, (2.26) converges to the same limit as N becomes a large number. Therefore, the information about the occurrence of a random event can be expressed by the *frequency definition of probability*, given by

$$\Pr(A) = \lim_{N\to\infty}\left(\frac{N(A)}{N}\right) \tag{2.27}$$

On the other hand, as stated by Andrey Nikolaevich Kolmogorov in his seminal work [170], "The probability theory, as a mathematical discipline, can and should be developed from axioms in exactly the same way as Geometry and Algebra." Kolmogorov thus established the axiomatic foundation of probability theory. According to this elegant and rigorous approach, we can define a *field of probability* formed by the triplet $\{\Omega, \mathcal{F}, \Pr(A)\}$, where Ω is the space of outcomes, $\mathcal{F}$ is a field that contains all possible events of the random experiment,* and $\Pr(A)$ is the probability of event A. This measure is so chosen as to satisfy the following axioms.

Axiom 1: $\Pr(A) \geq 0$
Axiom 2: $\Pr(\Omega) = 1$
Axiom 3: If $A \cap B = \varnothing$, then $\Pr(A \cup B) = \Pr(A) + \Pr(B)$, where $\cap$ and $\cup$ stand for the set operations intersection and union, respectively.

* In the terminology of mathematical analysis, the collection of subsets $\mathcal{F}$ is referred to as a σ-algebra [110].

For a countable infinite sequence of mutually exclusive events, it is possible to enunciate Axiom 3 in the following extended form:

Axiom 3′: For mutually exclusive events $A_1, A_2, \ldots, A_n$,

$$\Pr\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \Pr(A_i)$$

From these three axioms, and using set operations, it follows that

$$\Pr(\bar{A}) = 1 - \Pr(A) \tag{2.28}$$

where $\bar{A}$ stands for the complement of the set A. If $A \cap B \neq \varnothing$, then

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \tag{2.29}$$

In probability theory, an important and very useful concept is that of independence. Two events A_i and A_j, for $i \neq j$, are said to be independent if and only if

$$\Pr(A_i \cap A_j) = \Pr(A_i)\Pr(A_j) \tag{2.30}$$

It is also important to calculate the probability of a particular event given the occurrence of another. Thus, we define the conditional probability of A_i given A_j (supposing $\Pr(A_j) \neq 0$) as

$$\Pr(A_i|A_j) \triangleq \frac{\Pr(A_i \cap A_j)}{\Pr(A_j)} \tag{2.31}$$

It should be noted that, if A_i and A_j are independent, then $\Pr(A_i|A_j) = \Pr(A_i)$. This means that knowledge about the occurrence of A_j does not modify the probability of occurrence of A_i. In other words, the conditional probability of independent events is completely described by their individual probabilities.

Computation of the probability of a given event can be performed with the help of the theorem of total probability. Consider a finite or countably infinite set of mutually exclusive ($A_i \cap A_j = \varnothing$ for all $i \neq j$) and exhaustive $\left(\bigcup_i A_i = A\right)$ events. The probability of an arbitrary event B is given by

$$\Pr(B) = \sum_i \Pr(A_i \cap B) = \sum_i \Pr(A_i)\Pr(B|A_i) \tag{2.32}$$

2.3.2 Random Variables

A deterministic signal is defined in accordance with an established mathematical formula. In order to deal with random signals, it is important to introduce the notions of random variable (r.v.) and probability density function (pdf). In a certain sense, a random signal is a continuous flux or a discrete sequence of r.v.'s, the generation of which is ruled by a pdf. In order to fully understand this idea, let us start by formally defining the concept of random variable.

DEFINITION 2.1 An r.v. is a function $X(\cdot)$ that assigns a number $X(w_i)$, or simply x_i, called value of the r.v., to each possible outcome in the set of observations (sample space).

For a discrete r.v., which directly associates points with probabilities, we have [123]

$$\Pr[X(w_i) = x_i] = \Pr[w_i \in \Omega : X(w_i) = x_i] \tag{2.33}$$

which means that the probability of the value of the r.v., x_i, is the probability of the outcome, w_i, associated with x_i through the function $X(\cdot)$.

For a continuous r.v., since the probability in a particular point tends to zero, we have to resort to a certain range of values to obtain a nonzero value. As a consequence, we write

$$\Pr[X(w_i) \leq x_i] = \Pr[w_i \in \Omega : X(w_i) \leq x_i] \tag{2.34}$$

The definition in (2.34) can be rewritten to eliminate the explicit dependence on a particular outcome. This leads to an important function that characterizes r.v.'s.

DEFINITION 2.2 The cumulative distribution function (cdf)—or simply distribution function—of an r.v. X is given by

$$P_X(x) = \Pr[X(w) \leq x] = \Pr[X \leq x] \tag{2.35}$$

for $-\infty < x < \infty$.

The cdf has the following properties:

$$0 \leq P_X(x) \leq 1 \tag{2.36a}$$

$$P_X(x_1) \leq P_X(x_2) \quad \text{for } x_1 < x_2 \tag{2.36b}$$

$$P_X(-\infty) \triangleq \lim_{x \to -\infty} P_X(x) = 0 \tag{2.36c}$$

$$P_X(\infty) \triangleq \lim_{x \to \infty} P_X(x) = 1 \tag{2.36d}$$

$$\lim_{\substack{\varepsilon \to 0 \\ \varepsilon > 0}} P_X(x + \varepsilon) \triangleq P_X(x^+) = P_X(x) \tag{2.36e}$$

Notice that (2.36a) and (2.36b) show that the cdf is bounded between zero and one and is a monotone nondecreasing function of x.

The pdf, or simply density function, is defined as

$$p_X(x) = \frac{d}{dx}P_X(x). \tag{2.37}$$

The probability of the event $x_1 < X \le x_2$ is given by

$$\begin{aligned}\Pr(x_1 < X \le x_2) &= \Pr(X \le x_2) - \Pr(X \le x_1)\\ &= P_X(x_2) - P_X(x_1)\\ &= \int_{x_1}^{x_2} p_X(x)\, dx \end{aligned}\tag{2.38}$$

According to (2.38), the probability associated with an interval is the area under the pdf in such interval. If we make $x_1 = -\infty$, it is possible to write the cdf in terms of the pdf as follows:

$$P_X(x) = \int_{-\infty}^{x} p_X(\xi)\, d\xi \tag{2.39}$$

Using (2.39) and (2.36), we can see that

$$\int_{-\infty}^{\infty} p_X(x)\, dx = 1 \tag{2.40}$$

As mentioned before, the cdf must always be monotone nondecreasing. This implies that its derivative must always be nonnegative. Furthermore, due to (2.40), we may state that the pdf must always be a nonnegative function whose total area equals one.

Example 2.1 (Uniform Distribution)

A very useful class of random variables is formed by those that possess a uniform distribution. An r.v. X is said to have a uniform distribution if its pdf, in an interval $(a, b]$, is given by

$$p_X(x) = \begin{cases} 0, & x \leqslant a\\ \frac{1}{b-a}, & a < x \leqslant b\\ 0, & x > b \end{cases} \tag{2.41}$$

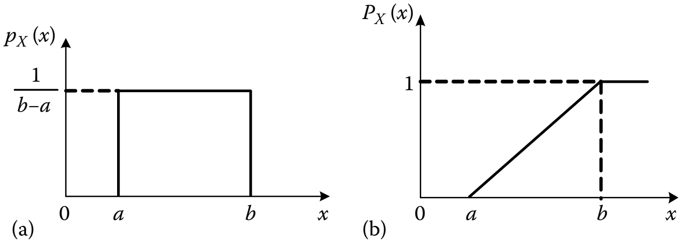


FIGURE 2.3
Example of a uniform distribution in the interval $(a, b]$: (a) probability density function of X and (b) cumulative distribution function of X.

In this case, the cumulative distribution function is given by

$$P_X(x) = \begin{cases} 0, & x \leqslant a \\ \frac{x-a}{b-a}, & a < x \leqslant b \\ 0, & x > b \end{cases} \tag{2.42}$$

Figure 2.3 shows the pdf and cdf of a uniformly distributed r.v. X.

2.3.2.1 *Joint and Conditional Densities*

When we work with random models in practical applications, the number of r.v.'s required to describe the behavior of the events is often greater than one. In this section, we extend the probabilistic concepts exposed so far to the case of multiple r.v.'s. Actually, we shall consider in detail exclusively the particular case of two variables, since the extension to the generic multidimensional case is somewhat direct. If we consider two r.v.'s X and Y, we can define the following distribution.

DEFINITION 2.3 The joint distribution function $P_{X,Y}(x,y)$ is the probability that the r.v. X is less than or equal to a specified value x and that the r.v. Y is less than or equal to a specified value y.

Mathematically, for $(-\infty < X \leq x, -\infty < Y \leq y)$, we write

$$P_{X,Y}(x,y) = Pr\left[X \leq x,\ Y \leq y\right] \tag{2.43}$$

Notice that (2.43) states that the outcome associated to the joint event is a point of the xy-plane. It is worth mentioning that X and Y may be considered as two separate one-dimensional r.v.'s as well as two components of a single two-dimensional r.v.

If $P_{X,Y}(x, y)$ is continuous everywhere, and the partial derivative

$$p_{X,Y}(x, y) = \frac{\partial^2 P_{X,Y}(x, y)}{\partial x \, \partial y} \tag{2.44}$$

exists and is also continuous everywhere, the function $p_{X,Y}(x, y)$ is called joint pdf of the r.v.'s X and Y.

The properties of the joint distribution function are similar to those exposed in the case of a one-dimensional r.v. Hence, $P_{X,Y}(x, y)$ is a monotone nondecreasing function of both variables x and y, the joint pdf $p_{X,Y}(x, y)$ is always nonnegative, and the volume under $p_{X,Y}(x, y)$ must equal one, i.e.,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{X,Y}(\xi, \eta) d\xi \, d\eta = 1 \tag{2.45}$$

When we deal with joint distributions (or densities), we are also interested in finding the distribution (or density) of a specific variable. Using Equation 2.43, we get

$$P_X(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{x} p_{X,Y}(\xi, \eta) d\xi \, d\eta \tag{2.46}$$

and differentiating both sides of the previous equation we have

$$p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x, \eta) d\eta \tag{2.47}$$

We can obtain $P_Y(y)$ and $p_Y(y)$ through a similar procedure. The distributions $P_X(x)$ and $P_Y(y)$ and densities $p_X(x)$ and $p_Y(y)$ are accordingly called marginal distributions and marginal densities of the joint distributions and densities, respectively.

The density of an r.v. when the occurrence of another is given, i.e., the conditional pdf of X, given that $Y = y$, is

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} \tag{2.48}$$

provided that $p_Y(y) > 0$.

Since $p_{X|Y}(x|y)$ is a function of the r.v. X with Y arbitrarily assuming a fixed value y, it fulfills the requirements of a pdf, since

$$p_{X|Y}(x|y) \geq 0$$

and

$$\int_{-\infty}^{\infty} p_{X|Y}(x|y)\, dx = 1$$

An important case arises when the variables X and Y are statistically independent. Then, knowledge about the occurrence of variable Y does not affect the distribution of X, or, in other words, the conditional density $p_{X|Y}(x|y)$ has no information about Y. In mathematical terms, we may write

$$p_{X|Y}(x|y) = p_X(x) \tag{2.49}$$

which means that the conditional density is equivalent to the marginal density when the variables are independent. From this result comes a very useful relation between the pdfs of independent variables:

$$p_{X,Y}(x,y) = p_X(x)p_Y(y) \tag{2.50}$$

i.e., the joint density of independent variables is equal to the product of their marginal densities.

In analogy with (2.48), we can define the conditional pdf of Y, given that $X = x$, as

$$p_{Y|X}(y|x) = \frac{p_{Y,X}(y,x)}{p_X(x)} \tag{2.51}$$

and, using (2.48) and (2.51), we obtain the following relation

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)p_X(x)}{p_Y(y)} \tag{2.52}$$

which is known as Bayes' rule [230].

It is straightforward to modify the approach to derive the equivalent relations in terms of the joint and conditional probability distributions.

2.3.2.2 Function of a Random Variable

At this point, it is useful to recall a classical result regarding the transformation of an r.v. Let $f(\cdot)$ be an arbitrary function, so that

$$Y = f(X)$$

where X is an r.v. whose density function is known.

Then, if $f(\cdot)$ is invertible, the following property of r.v. transformation holds [230]:

$$p_Y(y) = \left.\frac{p_X(x)}{\left|\frac{df(x)}{dx}\right|}\right|_{x=f^{-1}(y)} \tag{2.53}$$

where $f^{-1}(\cdot)$ is the inverse function of $f(\cdot)$. For multiple variables, if the mapping $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is invertible, then we have

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{p_{\mathbf{x}}(\mathbf{f}^{-1}(\mathbf{y}))}{|\det \mathbf{J}_{\mathbf{f}}(\mathbf{x})|} \tag{2.54}$$

where $\mathbf{f}^{-1}(\cdot)$ is the inverse function of $\mathbf{f}(\cdot)$, and $\mathbf{J}_{\mathbf{f}}(\mathbf{y})$ denotes the *Jacobian matrix* associated with $\mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x})\right]$, given by

$$\mathbf{J}_{\mathbf{f}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \tag{2.55}$$

Equation 2.53 can be also generalized when $f(X)$ does not produce a biunivocal mapping from y to x. In this case, it is necessary to separate the functions into regions that present a unique inverse mapping. Therefore, we may write [179]

$$p_Y(y) = \sum_i p_X(x_i) \left.\left|\frac{df(x_i)}{dx_i}\right|^{-1}\right|_{x_i=f^{-1}(y)} \tag{2.56}$$

where i stands for the number of regions the function is divided into, and x_i are the values of x inside the ith region.

2.3.3 Moments and Cumulants

The pdf plays a key role in statistical processing, since it carries all the available statistical information about an r.v. However, the determination of the exact model of the pdf may be a task quite hard to be accomplished in certain cases. Thus, it is relevant to build alternative expressions of the statistical behavior of a random variable that are both representative and straightforward to characterize. The idea of average, as we will show in the following, fits these requirements.

A simple and very important case is the expected value or mean of a random variable X, which is defined as

$$E\{X\} \triangleq \int_{-\infty}^{\infty} x p_X(x) dx \tag{2.57}$$

where $E\{\cdot\}$ denotes the statistical expectation operator. Another important case is the variance or second central moment, defined as

$$\text{var}(X) \triangleq \int_{-\infty}^{\infty} (x - \kappa_1(X))^2 p_X(x) dx \tag{2.58}$$

where $\kappa_1(X) = E\{X\}$.

We can generalize the idea of mean and variance by revisiting the notion of function of an r.v. Let X be an r.v. and $f(\cdot)$ an arbitrary function, so that

$$Y = f(X) \tag{2.59}$$

It is clear that Y is also a random variable and it is possible to define the expected value thereof as

$$E\{Y\} \triangleq \int_{-\infty}^{\infty} y p_Y(y) dy \tag{2.60}$$

or rather

$$E\left\{f(X)\right\} \triangleq \int_{-\infty}^{\infty} f(x) p_X(x) dx \tag{2.61}$$

Equation 2.61 generalizes the concept of the mean of an r.v. to the expectation of an arbitrary function of the same variable: such procedure will be of particular relevance throughout the entire book. At this point, it is worth pointing out a special case: if we make $f(X) = X^n$ in (2.61), we obtain the nth *moment* of the pdf $p_X(x)$, defined as

$$\kappa_n(X) = \int_{\infty}^{\infty} x^n p_X(x)\, dx \tag{2.62}$$

From (2.59) and (2.62) it is clear that $\kappa_1(X) = E\{X\}$.

Even though the moments can be directly computed using (2.62), a special function can be tailored to produce them directly. The moment-generating function or first characteristic function of a real r.v. X is defined as

$$\Omega_X(\omega) \triangleq \int_{-\infty}^{\infty} p_X(x) \exp(j\omega x) dx$$
$$\triangleq E\left\{\exp(j\omega x)\right\} \tag{2.63}$$

If $\Omega_X(\omega)$ is expanded in a Taylor series about the origin, we get to [230]

$$\Omega_X(\omega) \triangleq \sum_{k=0}^{\infty} \frac{\kappa_k}{k!}(j\omega)^k \tag{2.64}$$

where κ_k is the k-order moment. Thus, one can obtain the k-order moment using the following expression

$$\kappa_k = (-j)^k \frac{\partial^k \Omega_X(\omega)}{\partial \omega^k}\bigg|_{\omega=0}$$

Another important statistical measure is given by the cumulants. Cumulants are tailored by specific relationships between the moments of a random variable in order to reveal certain aspects of its pdf as well as to present some useful properties for statistical processing. The cumulants are generated by the second characteristic function or cumulant-generating function, defined by

$$\Upsilon_X(\omega) \triangleq \ln\left[\Omega_X(\omega)\right] \tag{2.65}$$

The Taylor series of $\Upsilon_X(\omega)$ around the origin can be written as

$$\Upsilon_X(\omega) \triangleq \sum_{k=0}^{\infty} \frac{c_k}{k!}(j\omega)^k \tag{2.66}$$

and

$$c_k = (-j)^k \frac{\partial^k \Upsilon_X(\omega)}{\partial \omega^k}\bigg|_{\omega=0} \tag{2.67}$$

is the k-order cumulant c_k [230, 244].

We have described the first and second characteristic functions for the real case. For complex r.v.'s, a straightforward extension is possible, so that the characteristic function becomes [13]

$$\Omega_{X,X^*}(\omega,\omega^*)$$
$$\triangleq \int_{-\infty}^{\infty} p_{X,X^*}(x,x^*) \exp\left[j\left(\frac{\omega x^* + \omega^* x}{2}\right)\right] dx\, dx^*$$
$$\triangleq E\left\{\exp\left[j\left(\frac{\omega x^* + \omega^* x}{2}\right)\right]\right\} \tag{2.68}$$

In this case, the cumulant-generating function is given by

$$\Upsilon_Y(\omega) \triangleq \ln\left[\Omega_{X,X^*}(\omega,\omega^*)\right] \tag{2.69}$$

2.3.3.1 Properties of Cumulants

Cumulants possess a number of properties that are interesting from a signal processing standpoint. Some of these properties are described in the sequel.

- *Invariance and equivariance*
  The first-order cumulant is equivariant, while the others are invariant to shifts, i.e.,

  $$\begin{aligned} c_1(X+\alpha) &= c_1(X)+\alpha \\ c_k(X+\alpha) &= c_k(X) \end{aligned} \tag{2.70}$$

  for an arbitrary constant α.
- *Homogeneity*
  The k-order cumulant is homogeneous of k degree. Thus, for the real case

  $$c_k(\alpha X) = \alpha^k \cdot c_k(X) \tag{2.71}$$

  In the complex case, the k-order cumulant is defined by

  $$c_k(X,X^*) = c_k(\underbrace{X,\ldots,X}_{s \text{ terms}},\underbrace{X^*,\ldots,X^*}_{q \text{ terms}}) \quad \forall\, s+q=k \tag{2.72}$$

  According to (2.72), the homogeneity property for a complex r.v. is given by [14,173]

  $$c_k(\alpha X,\alpha X^*) = (\alpha)^s \cdot \left(\alpha^*\right)^q \cdot c_k(X,X^*) \quad \forall\, s+q=k \tag{2.73}$$

  Hence, for even-order cumulants, we may consider $s=q$, so that the homogeneity condition becomes

  $$c_k(\alpha Y) = |\alpha|^k \cdot c_k(Y) \tag{2.74}$$

- *Additivity*
  If X and Y are statistically independent r.v.'s, the following relation holds:

  $$c_k(X+Y) = c_k(X) + c_k(Y) \tag{2.75}$$

2.3.3.2 Relationships between Cumulants and Moments

We can relate cumulants and moments via the following recursive rule [214]:

$$c_k = \kappa_k - \sum_{i=1}^{k-1} \binom{k-1}{i-1} c_i \cdot \kappa_{k-i} \tag{2.76}$$

Therefore, the kth moment is a kth order polynomial built from the first k cumulants. For instance, we have for k up to 6:

$$\begin{aligned}
\kappa_1 &= c_1 \\
\kappa_2 &= c_2 + c_1^2 \\
\kappa_3 &= c_3 + 3c_2c_1 + c_1^3 \\
\kappa_4 &= c_4 + 4c_3c_1 + 3c_2^2 + 6c_2c_1^2 + c_1^4 \\
\kappa_5 &= c_5 + 5c_4c_1 + 10c_3c_2 + 10c_3c_1^2 + 15c_2^2c_1 + 10c_2c_1^3 \\
\kappa_6 &= c_6 + 6c_5c_1 + 15c_4c_2 + 15c_4c_1^2 + 10c_3^2 + 60c_3c_2c_1 + 20c_3c_1^3 + 15c_2^3 \\
&\quad + 45c_2^2c_1^2 + 15c_2c_1^4 + c_1^6
\end{aligned} \tag{2.77}$$

Clearly, when zero-mean distributions are considered, the terms in c_1 are removed from (2.77). A more detailed description of these relationships can be found in [214].

2.3.3.3 Joint Cumulants

The joint cumulant of several r.v.'s $X_1, \ldots, X_k$ is defined similarly to (2.67) [214]

$$c\left(x_1^{k_1}, x_2^{k_2}, \ldots, x_p^{k_p}\right) \triangleq (-j)^r \left. \frac{\partial^r \Upsilon(\omega_1, \ldots, \omega_p)}{\partial w_1^{k_1} \ldots \partial w_p^{k_p}} \right|_{\omega_1 = \cdots = \omega_p = 0} \tag{2.78}$$

where $\Upsilon(\omega_1, \ldots, \omega_p)$ represents the second characteristic function of the joint pdf $X_1, \ldots, X_k$. If the variables are independent, their joint cumulant is null, and if all k variables are equal, the joint cumulant is $c_k(X)$.

In order to link the concepts we have presented with the notion of signal, we now consider the evolution of r.v.'s in detail.

2.4 Stochastic Processes

The proper method of including probability theory in the study of information signals consists of defining a sample space associated with a set of functions. This model originates the notion of a stochastic process. Considering time as the support of the signal, the following definition can be posed.

DEFINITION 2.4 A stochastic process $X(t)$ is a collection or ensemble of functions engendered by a rule that assigns a function $x(t, \omega_i)$ or simply $x_i(t)$, called sample of the stochastic process, to each possible outcome in the sample space.

Figure 2.4 illustrates in a classical and intuitive way the concept of stochastic process. From this figure we can observe that

- For a given ω_i we have a single time function or sample function $x_i(t)$. This function represents a specific realization of the random signal, which means that, for stochastic processes, the occurrence of a given signal is a result of a random experiment.
- For a given time instant t_K, the value to be assigned to $X(t_K)$ depends on the choice of the sample function, i.e., depends on ω_i. So $X(t_K)$ is a value corresponding to the outcome of a random experiment, i.e., is an r.v. Hence each sample function $x_i(t)$ is a flux of random variables in time.

As discussed earlier for deterministic signals, time flow can be modeled in terms of a continuum or an integer set of values, which means that random signals can also be continuous- or discrete-time. In order to characterize a

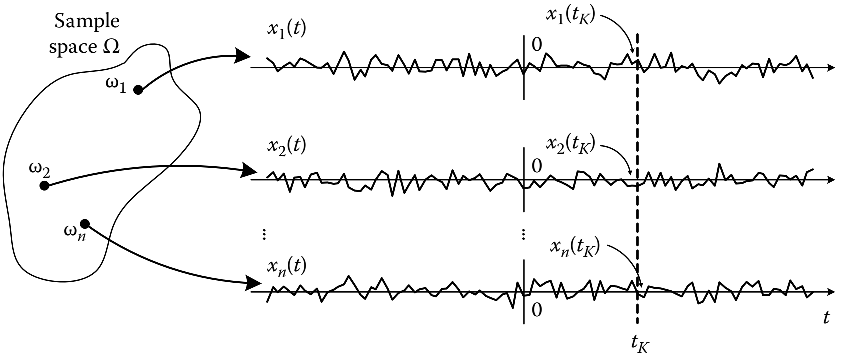


FIGURE 2.4
Several sample functions of a random process.

stochastic process, it is suitable to establish an interval of observation, for instance, $(0, t_K]$, as indicated in Figure 2.4, within which we consider K samples of the process: $X(t_k)$, $k = 1, \ldots, K$. Under these circumstances, a stochastic process actually becomes a family of random signals, which can be characterized by the corresponding joint CDF

$$\begin{aligned} &P_{X(t_1),\ldots,X(t_K)}(x(t_1), \ldots, x(t_K)) \text{ or, for the sake of simplicity,} \\ &\quad P_{X(t_1),\ldots,X(t_K)}(x_1, \ldots, x_K) \end{aligned} \tag{2.79}$$

and joint pdf

$$\begin{aligned} &p_{X(t_1),\ldots,X(t_K)}(x(t_1), \ldots, x(t_K)) \text{ or, for the sake of simplicity,} \\ &\quad p_{X(t_1),\ldots,X(t_K)}(x_1, \ldots, x_K). \end{aligned} \tag{2.80}$$

In many cases, to deal with these multivariable functions is a rather difficult task. It is then suitable to deal with a partial characterization of stochastic processes, as that which comes from a sort of extension of the concept of moments.

2.4.1 Partial Characterization of Stochastic Processes: Mean, Correlation, and Covariance

We can define the mean of a stochastic process $X(t)$ as the mean of the random variable originated when the time index is fixed at t. Since the mean is the first-order moment, we can write

$$\begin{aligned} \kappa_1(X,t) &= E\{X(t)\} \\ &= \int_{-\infty}^{\infty} x p_{X(t)}(x) dx \end{aligned} \tag{2.81}$$

where $p_{X(t)}(x)$ is called the first-order pdf of the process, since it takes into account only one time instant of the sample functions.

An equivalent of the second-order moment of an r.v. in the context of stochastic processes is the autocorrelation function. We can define this function as the statistical expectation of the product of two r.v.'s obtained from the observation of the process at two different time instants, t_1 and t_2. So we may write

$$\begin{aligned} R_X(t_1, t_2) &= E\left\{X(t_1)X^*(t_2)\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2^* p_{X(t_1),X(t_2)}(x_1, x_2)\, dx_1 dx_2 \end{aligned} \tag{2.82}$$

where $p_{X(t_1),X(t_2)}(x_1, x_2)$ is the second-order pdf of the process. We can also define the autocovariance function

$$\begin{aligned} C_X(t_1, t_2) &= E\left\{[X(t_1) - \kappa_1(X, t_1)]\,[X(t_2) - \kappa_1(X, t_2)]^*\right\} \\ &= R_X(t_1, t_2) - \kappa_1(X, t_1) \cdot \kappa_1^*(X, t_2) \end{aligned} \tag{2.83}$$

In order to evaluate the second-order moments in different time instants, let us create a vector $\mathbf{x} = [X(t_1) \quad X(t_2) \quad \cdots \quad X(t_n)]^T$. Then, if we compute $E\left\{\mathbf{x}\mathbf{x}^H\right\}$ for a zero-mean process, we obtain the autocorrelation matrix

$$\begin{aligned} \mathbf{R}_{\mathbf{xx}} &= E\left\{\mathbf{x}\mathbf{x}^H\right\} \\ &= \begin{bmatrix} X(t_1)X^*(t_1) & \cdots & X(t_1)X^*(t_n) \\ \vdots & \ddots & \vdots \\ X(t_n)X^*(t_1) & \cdots & X(t_n)X^*(t_n) \end{bmatrix} \\ &= \begin{bmatrix} R_X(t_1, t_1) & \cdots & R_X(t_1, t_n) \\ \vdots & \ddots & \vdots \\ R_X(t_n, t_1) & \cdots & R_X(t_n, t_n) \end{bmatrix} \end{aligned} \tag{2.84}$$

In the above definition, the superscript $(\cdot)^H$ stands for Hermitian transposition. The autocovariance matrix is obtained if the autocorrelation function is replaced by the autocovariance function in (2.84).

Another important measure is the cross-correlation function, which expresses the correlation between different processes. Given two different stochastic processes $X(t)$ and $Y(t)$, the two cross-correlation functions can be defined as [135]

$$R_{XY}(t_1, t_2) = E\left\{X(t_1)Y^*(t_2)\right\} \tag{2.85}$$

and

$$R_{YX}(t_1, t_2) = E\left\{Y(t_1)X^*(t_2)\right\} \tag{2.86}$$

We can also define a cross-correlation matrix, given by

$$\mathbf{R}_{XY}(t_1, t_2) = \begin{bmatrix} R_X(t_1, t_2) & R_{XY}(t_1, t_2) \\ R_{YX}(t_1, t_2) & R_Y(t_1, t_2) \end{bmatrix} \tag{2.87}$$

So far, there has been a strong dependence of the definitions with respect to multiple time indices. However, some random signals show regularities that can be extremely useful, as we shall now see.

2.4.2 Stationarity

The term stationarity refers to an important property shared by many random processes: their statistical characteristics are considered to be time invariant. We can express this idea in formal terms by using the notion of joint density function of a stochastic process $X(t)$, so that, if

$$p_{X(t_1),\dots,X(t_k)}(x_1,\dots,x_k) = p_{X(t_1+\tau),\dots,X(t_k+\tau)}(x_1,\dots,x_k) \qquad (2.88)$$

is valid for any τ, the process is said to be strict-sense stationary. In other words, it can be defined as follows [135].

DEFINITION 2.5 A stochastic process $X(t)$, initiated at $t = -\infty$, is strict-sense stationary if the joint density of any set of r.v.'s obtained by observing the random process $X(t)$ is invariant with respect to the location of the origin $t = 0$.

Two special cases deserve our attention:

1. $k = 1$

$$p_{X(t)}(x) = p_{X(t+\tau)}(x) = p_X(x) \quad \text{for all } t \text{ and } \tau. \qquad (2.89)$$

The above equation reveals that the first-order density function of a strict-sense stationary process is time independent.

2. $k = 2$ and $\tau = -t_1$

$$p_{X(t_1),X(t_2)}(x_1,x_2) = p_{X(0),X(t_2-t_1)}(x_1,x_2) \quad \text{for all } t_1 \text{ and } t_2. \qquad (2.90)$$

In this case, the second-order density function of a strict-sense stationary process depends only on the time difference between the observation times, and not on the particular instants of observation.

These properties have a great impact on the statistical characterization of a stochastic process, as the following results show.

- The mean of a stationary process is constant:

$$\kappa_1(X,t) = \kappa_1(X) \qquad (2.91)$$

- The autocorrelation and autocovariance functions depend exclusively on the time difference $\tau = t_2 - t_1$:

$$R_X(t_1,t_2) = R_X(\tau) \qquad (2.92)$$

and

$$C_X(t_1, t_2) = R_X(\tau) - \kappa_1^2(X) \tag{2.93}$$

for all t_1 and t_2.

- The autocorrelation matrix is hermitian. It means that the element in the ith row and jth column is equal to the complex conjugate of the element in the jth row and ith column, for all indices i and j, i.e.,

$$\mathbf{R_{xx}} = \begin{bmatrix} R_X(0) & \cdots & R_X^*(t_n - t_1) \\ \vdots & \ddots & \vdots \\ R_X(t_n - t_1) & \cdots & R_X(0) \end{bmatrix} \tag{2.94}$$

  Furthermore, the autocorrelation matrix of a strict-sense stationary process has the property of being a Toeplitz matrix [139].

- Given two stationary processes $X(t)$ and $Y(t)$, their cross-correlation matrix, which carries the information of the different cross-correlation functions, can be written as

$$\mathbf{R}_{XY}(\tau) = \begin{bmatrix} R_X(\tau) & R_{XY}(\tau) \\ R_{YX}(\tau) & R_Y(\tau) \end{bmatrix} \quad \tau = t_2 - t_1 \tag{2.95}$$

  The cross-correlation function does not have the same symmetry inherent to the autocorrelation function. However, the following relationship is verified [139]:

$$R_{XY}^*(\tau) = R_{YX}(-\tau) \tag{2.96}$$

A less stringent condition for stationarity is given when a partial characterization of the stochastic process is carried out by means of the mean and the autocorrelation function. A stochastic process is said to be wide-sense stationary if the mean and the autocorrelation function of the process do not depend on the time instant, i.e., if (2.91) through (2.93) hold, but not necessarily (2.88). Hence, strict-sense stationarity implies wide-sense stationarity, but the converse is not true.

At this point, it is worth mentioning that it is also possible to analyze random signals in the frequency domain. In the beginning of this chapter, we have introduced the Fourier transform to define the spectrum of a deterministic signal. Since a stochastic process is composed by a collection of sample functions, the change of domain will engender a collection of spectra, so that the inherent uncertainty of the random signal will be transferred to the frequency domain. Therefore, it is necessary to resort to statistical measures to evaluate the frequency behavior of a random signal, i.e., employ a transform of moments as a source of information in the frequency domain.

For now, we consider the second-order moment and define the power spectral density $S_X(f)$ of a stationary stochastic process with autocorrelation function $R_X(\tau)$ through the following pair of equations:

$$S_X(f) = \int_{-\infty}^{\infty} R_X(\tau)\exp(-j\,2\pi f\tau)d\tau \tag{2.97}$$

$$R_X(\tau) = \int_{-\infty}^{\infty} S_X(f)\exp(j\,2\pi f\tau)df \tag{2.98}$$

Equations 2.97 and 2.98 are well known in classical signal analysis and usually referred to as Einstein–Wiener–Khintchine relations [135].

2.4.3 Ergodicity

The estimation of statistical moments is a crucial task in the characterization of a random signal. An inherent problem is that their definitions consider the whole ensemble of realizations of the process. For instance, the mean $\kappa_1(X)$ describes the behavior of the r.v. $X(t)$, for a fixed t, taking into account all possible outcomes thereof.

However, in practice, we have access to a limited number of sample functions $x_i(t)$. Notwithstanding, most applications present an interval of observation adequately long to provide accurate measures of time averages. Therefore, we may seriously consider the hypothesis of using these averages to estimate statistical ensemble averages. The fundamental question is if these two entities can be considered equivalent. To answer it, let us consider a wide-sense stationary process $X(t)$ and an interval of observation $[-T, T]$. The time average of a single realization $x(t)$ is given by

$$\kappa_1(x, T) = \frac{1}{2T}\int_{-T}^{T} x(t)dt \tag{2.99}$$

Notice that the time average is an r.v., since it depends on the sample function to be considered in the computation. Thus, we can take the expectation of that measure so that, by interchanging the linear operators and assuming stationarity, we have

$$\begin{aligned} E\{\kappa_1(x, T)\} &= \frac{1}{2T}\int_{-T}^{T} E\{x(t)\}\,dt \\ &= \frac{1}{2T}\int_{-T}^{T} \kappa_1(X)dt \\ &= \kappa_1(X) \end{aligned} \tag{2.100}$$

As a consequence, the time average $\kappa_1(x;T)$ is an unbiased estimate of the mean $\kappa_1(x)$. We may say that the process $X(t)$ is ergodic in the mean if two conditions are satisfied [135]:

1. The time average $\kappa_1(x,T)$ tends to the mean $\kappa_1(x)$ if the interval of observation tends to infinity, i.e.,

$$\lim_{T\to\infty} \kappa_1(x,T) = \kappa_1(X)$$

2. The variance of $\kappa_1(x,T)$ tends to zero if the interval of observation tends to infinity, i.e.,

$$\lim_{T\to\infty} \text{var}[\kappa_1(x,T)] = 0$$

The same procedure can be applied to second-order moments. The temporal autocorrelation function of a sample $x(t)$ is given by

$$R_x(t,T) = \frac{1}{2T}\int_{-T}^{T} x(t+\tau)x^*(t)dt \tag{2.101}$$

This time average is also an r.v., and we say that a process is ergodic in the autocorrelation function if the following conditions hold:

$$\lim_{T\to\infty} R_x(\tau,T) = R_X(\tau)$$

$$\lim_{T\to\infty} \text{var}\,[R_x(\tau,T)] = 0$$

It is also possible to define ergodicity of stochastic processes in terms of higher-order moments. However, in practice, ergodicity in the mean and autocorrelation is usually enough in classical problems of signal analysis.

2.4.4 Cyclostationarity

Nonstationary processes are characterized by a time-variant statistical behavior. However, there is a special class of stochastic processes whose statistics do vary in time, but in a very specific manner: they vary periodically. Processes for which the statistical parameters experience cyclic (periodical) changes are called cyclostationary processes. The following definition holds [7].

DEFINITION 2.6 A stochastic process $X(t)$ is said to be cyclostationary with period T if and only if $p_{X(t_1),\ldots,X(t_k)}(x_1,\ldots,x_k)$ is periodic in t with period T, i.e.,

$$p_{X(t_1),\ldots,X(t_K)}(x_1,\ldots,x_K) = p_{X(t_1+T),\ldots,X(t_K+T)}(x_1,\ldots,x_K)$$

It is also possible to establish two possibilities: strict-sense cyclostationarity, which corresponds to the above definition, and wide-sense (weak) cyclostationarity. A stochastic process $X(t)$ is wide-sense cyclostationary if its mean and autocorrelation function are periodic in t with some period T, i.e.,

$$\kappa_1(X, t+T) = \kappa_1(X, t) \tag{2.102}$$

$$R_X\left(t+\frac{\tau}{2}, t-\frac{\tau}{2}\right) = E\left\{X\left(t+\frac{\tau}{2}\right)X^*\left(t-\frac{\tau}{2}\right)\right\} = R_X\left(t+\frac{\tau}{2}+T, t-\frac{\tau}{2}+T\right) \tag{2.103}$$

for all $\tau \in (-T, T)$.

2.4.5 Discrete-Time Random Signals

A discrete-time random process is a particular kind of random process in which the time variable is of a discrete nature. Formally, a discrete-time random process is defined as follows.

DEFINITION 2.7 A discrete-time stochastic process $X(n)$ is a collection, or ensemble, of functions engendered by a rule that assigns a sequence $x(n, \omega_i)$ or simply $x_i(n)$, called sample of the stochastic process, to each possible outcome of a sample space.

Similarly to its continuous-time counterpart, it is possible to characterize a discrete-time random process by means of first- and second-order moments. Thus, we define the mean of a discrete-time random process as the mean value of the corresponding random variable produced when the time index n is fixed, i.e.,

$$\begin{aligned}\kappa_1(X, n) &= E\{X(n)\} \\ &= \int_{-\infty}^{\infty} x p_{X(n)}(x)dx\end{aligned} \tag{2.104}$$

where $p_{X(n)}(x)$ is the first-order pdf of the process.

Following the presentation in Section 2.4.1, we may also define the autocorrelation function of a discrete-time random process as

$$R_X(n_1, n_2) = E\left\{X(n_1)X^*(n_2)\right\}$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2^* p_{X(n_1),X(n_2)}(x_1, x_2)dx_1dx_2 \tag{2.105}$$

the autocovariance function as

$$C_X(n_1, n_2) = E\left\{[X(n_1) - \kappa_1(X, n_1)]\,[X(n_2) - \kappa_1(X, n_2)]^*\right\}$$
$$= R_X(n_1, n_2) - \kappa_1(X, n_1) \cdot \kappa_1^*(X, n_2) \tag{2.106}$$

and, finally, the autocorrelation matrix, defined for a vector $\mathbf{x} = \left[X(n_1)\quad X(n_2)\quad \cdots \quad X(n_n)\right]^T$, given by

$$\mathbf{R_{xx}} = E\left\{\mathbf{xx}^H\right\}$$
$$= \begin{bmatrix} X(n_1)X^*(n_1) & \cdots & X(n_1)X^*(n_n) \\ \vdots & \ddots & \vdots \\ X(n_n)X^*(n_1) & \cdots & X(n_n)X^*(n_n) \end{bmatrix}$$
$$= \begin{bmatrix} R_X(n_1, n_1) & \cdots & R_X(n_1, n_n) \\ \vdots & \ddots & \vdots \\ R_X(n_n, n_1) & \cdots & R_X(n_n, n_n) \end{bmatrix} \tag{2.107}$$

In Appendix A, we show some important properties of the autocorrelation matrix, which will help us in the development of Chapter 3.

2.4.6 Linear Time-Invariant Systems with Random Inputs

Suppose that a random process $X(t)$ (or, equivalently, a discrete-time random process $X(n)$) is applied as input to an LTI filter with impulse response $h(t)$ ($h(n)$). According to (2.20) (and (2.21)), the output of such system will be given by

$$Y(t) = \int_{-\infty}^{\infty} h(\tau)X(t - \tau)d\tau \tag{2.108}$$

$$Y(n) = \sum_{m=-\infty}^{\infty} h(m)X(n - m) \tag{2.109}$$

In general, it is difficult to describe the probability distribution of the output random process $Y(t)$, even when the probability distribution of the input random process $X(t)$ is completely specified for $-\infty \leq t \leq +\infty$. However, it is useful to perform an analysis in terms of the mean and autocorrelation function of the output signal.

If we assume that the input signal $X(t)$ is a stationary process, then we can evaluate the mean of the output random process $Y(t)$ as follows:

$$\begin{aligned}\kappa_1(Y,t) &= E\left\{\int_{-\infty}^{\infty} h(\tau)X(t-\tau)d\tau\right\} \\ &= \int_{-\infty}^{\infty} h(\tau)E\left\{X(t-\tau)d\tau\right\} \\ &= \int_{-\infty}^{\infty} h(\tau)\kappa_1(X,t-\tau)d\tau \end{aligned} \tag{2.110}$$

and, since we are dealing with a stationary process, we have $\kappa_1(X) = \kappa_1(X,t)$, hence

$$\begin{aligned}\kappa_1(Y) &= \kappa_1(X)\int_{-\infty}^{\infty} h(\tau)d\tau \\ &= \kappa_1(X,t)H(0)\end{aligned} \tag{2.111}$$

where $H(0)$ is the zero frequency response of the system.

We can also evaluate the autocorrelation function of the output signal $Y(t)$. Recalling that

$$R_Y(t_1,t_2) = E\left\{Y(t_1)Y^*(t_2)\right\} \tag{2.112}$$

$$\begin{aligned}R_Y(t_1,t_2) &= E\left\{\int_{-\infty}^{\infty} h(\tau_1)X(t_1-\tau_1)d\tau_1 \int_{-\infty}^{\infty} h^*(\tau_2)X^*(t_2-\tau_2)d\tau_2\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\tau_1)h^*(\tau_2)E\left\{X(t_1-\tau_1)X^*(t_2-\tau_2)\right\}d\tau_1 d\tau_2\end{aligned} \tag{2.113}$$

Now, since we assume that $X(t)$ is stationary, then, as discussed in Section 2.4.2, the autocorrelation function of $X(t)$ is only a function of the difference between the observation times $t_1 - \tau_1$ and $t_2 - \tau_2$. Thus, letting

$\tau = t_1 - t_2$, we get

$$R_Y(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tau_1) h^*(\tau_2) R_X(\tau - \tau_1 + \tau_2) d\tau_1 d\tau_2 \tag{2.114}$$

which can be rearranged to reveal that

$$\begin{aligned} R_Y(\tau) &= \int_{-\infty}^{\infty} h^*(\tau_2) \int_{-\infty}^{\infty} h(\tau_1) R_X(\tau + \tau_2 - \tau_1) d\tau_1 d\tau_2 \\ &= \int_{-\infty}^{\infty} h^*(\tau_2) \left\{ h(t) * R_X(t + \tau_2) \right\} d\tau_2 \\ &= R_X(\tau) * h(-\tau) * h^*(\tau) \end{aligned} \tag{2.115}$$

This result can also be presented in the frequency domain, in terms of the power spectral density or power spectrum $S_Y(f)$, defined as the Fourier transform of the autocorrelation function of $Y(t)$, i.e.,

$$S_Y(f) = \int_{-\infty}^{\infty} R_Y(\tau) \exp(-j2\pi f \tau) d\tau \tag{2.116}$$

Employing (2.115) and (2.116), we reach the conclusion that

$$S_Y(f) = \int_{-\infty}^{\infty} h(\tau_1) \int_{-\infty}^{\infty} h^*(\tau_2) \int_{-\infty}^{\infty} R_X(\tau - \tau_1 + \tau_2) \exp(-j2\pi f \tau) d\tau d\tau_1 d\tau_2 \tag{2.117}$$

Changing variables $t = \tau + \tau_1 - \tau_2$ and $dt = d\tau$ leads to

$$\begin{aligned} S_Y(f) = &\int_{-\infty}^{\infty} h(\tau_1) \exp(j2\pi f \tau_1) d\tau_1 \int_{-\infty}^{\infty} h^*(\tau_2) \exp(-j2\pi f \tau_2) d\tau_2 \\ &\times \int_{-\infty}^{\infty} R_X(t) \exp(-j2\pi f t) dt \\ &= H^*(f) H(f) S_X(f) \\ &= \left| H(f) \right|^2 S_X(f) \end{aligned} \tag{2.118}$$

Notice that $R_Y(0) = E\left\{|Y(t)|^2\right\}$, and, from (2.118), it comes that

$$E\left\{|Y(t)|^2\right\} = \int_{-\infty}^{\infty} \left|H(f)\right|^2 S_X(f)df \tag{2.119}$$

A similar reasoning lead us to the discrete-time counterpart of this result. The autocorrelation of the output of an LTI system with a discrete-time random process as an input is given by

$$R_Y(k) = R_X(k) * h(-n) * h^*(n) \tag{2.120}$$

where, in this case, $*$ stands for the discrete-time convolution, and

$$S_Y(\exp(j\,2\pi f)) = \left|H(\exp(j\,2\pi f))\right|^2 S_X(\exp(j\,2\pi f)) \tag{2.121}$$

2.5 Estimation Theory

Estimation theory is the field of statistical signal processing that deals with the determination of one or more parameters of interest, based on a set of available measured or empirical data. This problem is rather general, and a number of scientific domains derive great benefit from the application of estimation techniques.

From the perspective of this book, it is particularly relevant to the case in which the parameters of interest are associated with a system to be designed or analyzed. This corresponds to applying statistical methods to the optimal filtering problem, subject of Chapter 3. Before that, however, it is useful to present the foundations of estimation theory in general terms.

Different methods can be built according to the hypotheses we assume concerning the parameters to be estimated [70]. If they are considered to be deterministic parameters, we may derive the so-called classical estimation methods, as that of *maximum likelihood* (ML) estimation. Dealing with the parameters to be estimated as r.v.'s gives rise to the Bayesian estimation methods, like the *minimum mean-squared error* (MMSE) and the *maximum a posteriori* (MAP) methods. Finally, an estimation method may be derived regardless of the nature of the unknown parameters, as the *least-squares* (LS) estimation method that may be applied to either random or deterministic parameters.

As a well-established discipline, estimation theory has been treated in important books, in which existing methods are studied in detail [165, 181, 265, 283]. In this section, we provide a brief exposition of the main approaches, having in mind the problems to be discussed in the subsequent chapters.

2.5.1 The Estimation Problem

For the sake of clarity, the estimation problem can be divided into two cases, depending on the number of parameters involved: single-parameter estimation and multiple-parameter estimation.

2.5.1.1 Single-Parameter Estimation

Let us consider that a realization $x(n)$ of a discrete-time stochastic process depends on an unknown parameter θ. The problem of parameter estimation is then to estimate the parameter θ from a finite set of available observations $\{x(0) \quad x(1) \quad \cdots \quad x(N-1)\}$. Hence, we need to construct a function that extracts the parameter from the measurements:

$$\widehat{\theta} = \phi\left[x(0) \quad x(1) \quad \cdots \quad x(N-1)\right] \tag{2.122}$$

where $\phi[\cdot]$ is a deterministic transformation to be determined. One important aspect is that θ can be either deterministic or random, depending on the problem at hand, while $\widehat{\theta}$ is typically random, since it is a function of random variables. The r.v. $\widehat{\theta}$ is called estimator of θ and a realization of such r.v. is called estimate [165].

Let $\mathbf{x} = [x(0) \quad x(1) \quad \cdots \quad x(N-1)]^T$. If θ is assumed to be deterministic, we use the notation $p_X(\mathbf{x}; \theta)$ to emphasize the dependence of the data on θ. When θ is random, $\mathbf{x}$ and θ are related by means of the joint pdf denoted by $p_{X,\theta}(\mathbf{x}, \theta)$. In both cases, $\widehat{\theta}$ is a deterministic function of $\mathbf{x}$ and hence is also statistically dependent on θ, but it is not a deterministic function on θ. This fact tells us that $\partial\widehat{\theta}/\partial\theta = 0$.

2.5.1.2 Multiple-Parameter Estimation

Let us now consider that the discrete stochastic process $x(n)$ depends on a set of L unknown parameters $\theta(0) \quad \theta(1) \quad \cdots \quad \theta(L-1)$. Now the problem consists in finding a transformation

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\phi}\left[x(0) \quad x(1) \quad \cdots \quad x(N-1)\right] \tag{2.123}$$

in order to estimate the unknown parameters, being $\boldsymbol{\phi}(\cdot)$ a vector of L functions to be determined. Similarly to the single-parameter case, $\boldsymbol{\theta}$ can be either deterministic or random while $\widehat{\boldsymbol{\theta}}$ is typically random. Also, $\mathbf{x} = [x(0) \quad x(1) \quad \cdots \quad x(N-1)]^T$ is related to $\boldsymbol{\theta}$ via its pdf $p_X(\mathbf{x}; \boldsymbol{\theta})$ when $\boldsymbol{\theta}$ is deterministic, while $\mathbf{x}$ and θ are related via the joint pdf $p_{X,\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta})$ when $\boldsymbol{\theta}$ is random. Also, $\widehat{\boldsymbol{\theta}}$ is statistically dependent on $\boldsymbol{\theta}$, and it holds that $\partial\widehat{\boldsymbol{\theta}}^T/\partial\boldsymbol{\theta} = \mathbf{0}$.

Since the problem is mathematically stated, it is relevant to mention some important statistical properties of the estimators.

2.5.2 Properties of Estimators

2.5.2.1 Bias

An estimator $\widehat{\theta}$ is said to be unbiased if

$$E\left\{\widehat{\theta}\right\} = E\left\{\theta\right\} \tag{2.124}$$

otherwise, it is said to be biased with

$$\text{Bias}\left(\widehat{\theta}\right) = E\left\{\theta - \widehat{\theta}\right\} = E\left\{\theta\right\} - E\left\{\widehat{\theta}\right\} \tag{2.125}$$

When θ is deterministic, we have $E\left\{\theta\right\} = \theta$.

When multiple parameters are considered and the estimator $\widehat{\boldsymbol{\theta}}$ is unbiased, we have

$$E\left\{\widehat{\boldsymbol{\theta}}\right\} = E\left\{\boldsymbol{\theta}\right\} \tag{2.126}$$

and the bias is given by

$$\text{Bias}\left(\widehat{\boldsymbol{\theta}}\right) = E\left\{\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\right\} = E\left\{\boldsymbol{\theta}\right\} - E\left\{\widehat{\boldsymbol{\theta}}\right\} \tag{2.127}$$

2.5.2.2 Efficiency

For two unbiased estimators $\widehat{\theta}$ and $\widetilde{\theta}$, we say that $\widehat{\theta}$ is more efficient than $\widetilde{\theta}$ if

$$\text{var}\left(\widehat{\theta}\right) \leq \text{var}\left(\widetilde{\theta}\right) \tag{2.128}$$

Additionally, we can define the estimation error as

$$\varepsilon = \theta - \widehat{\theta} \tag{2.129}$$

so that the notion of efficiency is related to achieving an unbiased estimator $\widehat{\theta}$ with the smallest error variance, which is given by $\text{var}\left(\varepsilon\right) = \text{var}\left(\widehat{\theta}\right)$.

For the multiple-parameter case, the estimator $\widehat{\boldsymbol{\theta}}$ is said to be more efficient than $\widetilde{\boldsymbol{\theta}}$, assuming both to be unbiased, if

$$\mathbf{C}_{\widehat{\theta}}\left(\widehat{\theta}\right) \leq \mathbf{C}_{\widetilde{\theta}}\left(\widetilde{\theta}\right) \tag{2.130}$$

where

$$\mathbf{C_x}(\mathbf{x}) = E\left\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^H\right\} \tag{2.131}$$

is the covariance matrix of the vector **x**, which reduces to the autocorrelation matrix if $E\{x\} = 0$. The notation $\mathbf{A} > \mathbf{B}$ ($\mathbf{A} \geq \mathbf{B}$) denotes that matrix $\mathbf{A} - \mathbf{B}$ is positive definite (positive semidefinite) [128].

2.5.2.3 Cramér–Rao Bound

An important result associated with unbiased estimators is the existence of a lower bound of performance in terms of the variance of the estimated parameter $\widehat{\theta}$. This lower bound is the *Cramér–Rao Bound* (CRB) [165], which provides the minimum achievable performance for an estimator in terms of variance of the estimated parameters. It can be shown that, if an unbiased estimator $\widehat{\theta}$ exists, then its variance obeys [70, 165]

$$\mathrm{Var}\left(\widehat{\theta}\right) \geq F^{-1}(\theta) \tag{2.132}$$

where

$$F(\theta) = E\left\{\frac{\partial^2 p_X(\mathbf{x};\theta)}{\partial\theta\,\partial\theta^*}\right\} \tag{2.133}$$

is called the Fisher information. The equality in (2.132) holds if and only if the estimation error is given by

$$\widehat{\theta} - \theta = I(\theta)\frac{\partial p_X(\mathbf{x};\theta)}{\partial\theta} \tag{2.134}$$

where $I(\theta)$ is a nonzero function. Roughly speaking, $F(\theta)$ measures the information present in $p_X(\mathbf{x};\theta)$, and it is intuitive that the more information the pdf provides, the more accurate is the estimator $\widehat{\theta}$ and, consequently, the smaller is its variance.

For multiple parameters, the CRB is written as [70, 165]

$$\mathbf{C}\left(\widehat{\boldsymbol{\theta}}\right) \geq \mathbf{F}^{-1}(\boldsymbol{\theta}) \tag{2.135}$$

where

$$\mathbf{F}(\boldsymbol{\theta}) = E\left\{\left[\frac{\partial p_X(\mathbf{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]\cdot\left[\frac{\partial p_X(\mathbf{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]^H\right\} \tag{2.136}$$

is the Fisher information matrix. For this case, the equality in (2.136) holds if and only if

$$\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta} = I(\boldsymbol{\theta}) \frac{\partial p_X(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{2.137}$$

where $I(\boldsymbol{\theta})$ is a matrix composed of nonzero functions.

Some aspects concerning the estimator properties for infinite data length (asymptotic properties) may also be of interest. For instance, Refs. [70, 165] present relevant discussions on this topic, which is out of our main focus of presenting estimation theory methods as a support of the optimal filtering techniques. Therefore, we now turn our attention to the design of estimators. We will consider the case of multiple-parameter estimation, which is more general and more relevant to our objectives.

2.5.3 Maximum Likelihood Estimation

The ML estimation method consists of finding the estimator that maximizes the likelihood function established between the observed data and the parameters. In other words, for a given set of available measurements $\mathbf{x}$, we search for the parameters $\boldsymbol{\theta}$ that provide the highest probability $p_X(\mathbf{x}|\boldsymbol{\theta})$ with which the observed data would have been generated. Thus, the ML estimator is given by

$$\widehat{\boldsymbol{\theta}}_{\text{ML}} = \arg \max_{\boldsymbol{\theta}} p_X(\mathbf{x}|\boldsymbol{\theta}) \tag{2.138}$$

where the pdf $p_X(\mathbf{x}|\boldsymbol{\theta})$ is the likelihood function. Hence, we need to find the maximum of this function, which is given by its first derivative.

Due to the widespread use of exponential families of pdfs, it is very usual to use the log-likelihood. This being the case, the solution of the likelihood equation is given by

$$\left. \frac{\partial}{\partial \boldsymbol{\theta}} \ln \left[p_X(\mathbf{x}|\boldsymbol{\theta}) \right] \right|_{\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_{\text{ML}}} = \mathbf{0} \tag{2.139}$$

When Equation 2.139 presents several solutions, one must keep the $\widehat{\boldsymbol{\theta}}$ that corresponds to the global maximum.

One important feature of the ML estimator is that it is asymptotically efficient, that is, the ML estimator achieves the CRB when the number of observed data tends to infinity [165].

2.5.4 Bayesian Approach

As mentioned before, the Bayesian approach assumes that the parameters to be estimated are r.v.'s. In such case, we need to have some sort of model and/or a priori information about the parameters distribution $p_\Theta(\boldsymbol{\theta})$. In practice, the main question is how to obtain such a priori information. We, however, will concentrate our efforts on presenting the methodology to obtain the estimator of random parameters.

In essence, the core of Bayesian estimation is related to the a posteriori density $p_{\Theta|X}(\boldsymbol{\theta}|\mathbf{x})$, as our interest lies in finding the distribution of the parameters given the available measurements. We have two main approaches to Bayesian estimation: MAP and MMSE.

2.5.4.1 Maximum a Posteriori Estimation

In a way there is a certain duality between ML and MAP estimators, since, in the latter case, for a given set of available measurements $\mathbf{x}$, we search for the parameters $\boldsymbol{\theta}$ with the highest probability $p_{\Theta|X}(\boldsymbol{\theta}|\mathbf{x})$ of having generated the observed data. Therefore, the MAP estimator is obtained by solving

$$\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p_{\Theta|X}(\boldsymbol{\theta}|\mathbf{x}) \tag{2.140}$$

As previously, we employ the logarithm function in order to facilitate the process of handling exponential families. Then, by using Bayes' theorem, we can maximize the conditional pdf in (2.140) by posing

$$\frac{\partial \ln\left[p_{\Theta|X}(\boldsymbol{\theta}|\mathbf{x})\right]}{\partial\boldsymbol{\theta}} = \frac{\partial \ln\left[p_{X|\Theta}(\mathbf{x}|\boldsymbol{\theta})\right]}{\partial\boldsymbol{\theta}} + \frac{\partial \ln\left[p_{\Theta}(\boldsymbol{\theta})\right]}{\partial\boldsymbol{\theta}} = \mathbf{0} \tag{2.141}$$

We can notice that the MAP estimator takes into account the probabilistic information about $\boldsymbol{\theta}$. Furthermore, if $\boldsymbol{\theta}$ has a uniform distribution, the ML and MAP estimators lead to the same result. Intuitively, this is indeed the case if we have no available a priori information about $\boldsymbol{\theta}$. When $p_\Theta(\boldsymbol{\theta})$ is not uniformly distributed, which means that we have some useful a priori information about the distribution of the parameters, the estimators are different.

Example 2.2 (ML and MAP Estimators)

In order to illustrate the difference between ML and MAP estimators, let us consider a simple and, hopefully, amusing example related to a very popular adult beverage: the wine.

The wine consists of a complex myriad of substances, and several factors contribute to its taste and quality. The grape variety from which the wine was made from is one of them, and wine connoisseurs claim that, with sufficient training, a person should be capable of identifying the variety used to produce the wine simply by tasting it.

Let us now consider a nonprofessional wine taster that tastes a given glass of wine from a South American producer and tries to discern four different

TABLE 2.1

Result of a Hypothetical Pool about the Most Noticeable Flavor Found in the Wine

	Blackcurrant (%)	Plum (%)	Raspberry (%)	Other (%)
Cabernet-Sauvignon	41	45	2	12
Tannat	4	0	79	17
Malbec	31	55	0	14
Merlot	40	30	20	10

grapes traditionally found in this region: Cabernet-Sauvignon, Tannat, Malbec and Merlot. We can state that the *taste* stands for the observed data while the *grape* is the parameter to be estimated.

As a support to his difficult task, the taster relies on the information contained in Table 2.1, which shows the results of a poll in which renowned wine professionals were asked what is the most noticeable flavor in the wines made by this producer.

Let us say that the mentioned taster found a very distinguished touch of plum in the wine being tested. According to Table 2.1, we have that

$$P(\text{taste} = \text{plum}|\text{variety}) = \begin{cases} 0.45, & \text{for variety} = \text{Cabernet-Sauvignon} \\ 0, & \text{for variety} = \text{Tannat} \\ 0.55, & \text{for variety} = \text{Malbec} \\ 0.30, & \text{for variety} = \text{Merlot} \end{cases}$$

Therefore, if the decision is made using the ML criterion, according to (2.139), the "estimated grape" is the variety that

$$\widehat{\text{variety}} = \max_{\text{variety}} P(\text{taste} = \text{plum}|\text{variety}) \qquad (2.142)$$

That is, based on the poll provided by specialists, the taster concludes that, since the taste of plum is predominant, the wine is more likely to be a Malbec.

Now, suppose that, in addition to Table 2.1, the taster is also informed that approximately 40% of the total grape production of this wine maker consists of the cabernet-sauvignon variety, and each one of the other three grapes correspond to 20% of the total production. Since we have prior information, it is reasonable to employ the MAP criterion in this case. In other words, we take into account not only the fact that the flavor of plum is claimed by the specialists to be more likely found if the wine is Malbec, but, since it can also be found in other varieties, we take in account the probability of occurrence of each variety. Hence, the estimate is given by

$$P(\text{variety}|\text{taste} = \text{plum}) \propto P(\text{taste} = \text{plum}|\text{variety})P(\text{variety}) \qquad (2.143)$$

Thus, using Table 2.1 we get

$$P(\text{variety, taste = plum})P(\text{variety}) = \begin{cases} 0.18, & \text{for variety = Cabernet-Sauvignon} \\ 0, & \text{for variety = Tannat} \\ 0.11, & \text{for variety = Malbec} \\ 0.06, & \text{for variety = Merlot} \end{cases}.$$

Thus, in this case, even though the Malbec wine provides a highest probability of finding the flavor of plum, the MAP estimator changes the answer to Cabernet-Sauvignon. This is because the taster now takes into account the marginal probabilities $P(\textit{Cabernet-Sauvignon}) = 0.4$ and $P(\textit{Malbec}) = 0.2$, which increases the probability that the observed flavor of plum is originated by a Cabernet-Sauvignon wine.

2.5.4.2 Minimum Mean-Squared Error

As mentioned before, the estimation error is directly related to the efficiency of the estimator. For multiple parameters, we can define the error vector

$$\epsilon = \theta - \widehat{\theta} \tag{2.144}$$

Whenever the set of parameters θ to be estimated is random, we may think about any "measuring of closeness" between θ and its estimate. A statistical average of the estimation error is not per se a suitable candidate, since it is possible, for example, that a zero-mean error has a significant variance. In other words, the estimator may be unbiased but not efficient. A suitable option is to work with the statistical average of the square of the error, i.e., with the mean-squared error (MSE). Such option originates the method of the MMSE estimation, which consists in finding the $\widehat{\theta}$ that minimizes

$$J_{\text{MSE}}(\widehat{\theta}) = E\left\{\|\epsilon\|^2\right\} = E\left\{\left\|\theta - \widehat{\theta}\right\|^2\right\} \tag{2.145}$$

In (2.145) it should be emphasized that since θ is random, the expectation operator is taken with respect to the joint pdf $p(\mathbf{x}, \theta)$, which means that

$$\begin{aligned} J_{\text{MSE}}(\widehat{\theta}) &= \int\int \left\|\theta - \widehat{\theta}\right\|^2 p_{\Theta,X}(\mathbf{x}, \theta) d\mathbf{x}\, d\theta \\ &= \int \left[\int \left\|\theta - \widehat{\theta}\right\|^2 p_{\Theta|X}(\theta|\mathbf{x}) d\theta\right] p_X(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{2.146}$$

Since $p(\mathbf{x}) \geq 0$ for all $\mathbf{x}$, $J_{\text{MSE}}(\widehat{\theta})$ will be minimized if the term in brackets is minimized for each value of $\mathbf{x}$. Thus,

$$\frac{\partial}{\partial\widehat{\theta}} \int \left\| \theta - \widehat{\theta} \right\|^2 p_{\Theta|X}(\theta|\mathbf{x})d\theta = -2 \int \theta p_{\Theta|X}(\theta|\mathbf{x})d\theta + 2\widehat{\theta} \int p_{\Theta|X}(\theta|\mathbf{x})d\theta \tag{2.147}$$

Finally, setting the derivative equal to zero leads us to the optimum estimator in the MMSE sense, given by [165]

$$\widehat{\theta}_{\text{MSE}} = \int_{-\infty}^{\infty} \theta p_{\Theta|X}(\theta|\mathbf{x})d\theta = E\{\theta|\mathbf{x}\} \tag{2.148}$$

In general, the solution in (2.148) is of a nonlinear nature, which means that $\widehat{\theta}_{\text{MSE}}$ is not necessarily easy to be obtained. A simplifying hypothesis consists in considering the estimator as a linear function of the data. This model is suitable in a number of applications, and an emblematic example of the linear MMSE estimator is the *Wiener filter*, which will be studied in Chapter 3. In the classical Wiener filtering formulation, the parameters to be estimated are the coefficients of a linear combiner. Solution (2.148) is then reduced to a system of linear equations, as we shall see in more detail.

Moreover, the Wiener approach can be extended to that of a sequential MMSE estimator in order to accommodate data vectors that additionally may be nonstationary. This leads to the *Kalman filter*, which brings in its core the ideas of recursivity and adaptivity, to be more deeply discussed in Chapter 3. Appendix B presents the derivation of the Kalman filter.

2.5.5 Least Squares Estimation

In many practical scenarios, to rely on statistical entities is an idealization. However, it is possible to consider an alternative and intuitive solution that is more "data-oriented." This was the approach employed by Johann Carl Friedrich Gauss in 1795 when he established the Method of Least Squares (LS) to study the motion of celestial bodies [119]. LS estimation does not consider any hypothesis about probability distributions, neither for the data nor for the parameters. Only a model for the generation of the measurements is assumed and the parameters are estimated according to this assumption.

Typically, LS estimation can be used to build linear estimators, generalized estimators, and nonlinear estimators. For the linear case, we assume that the measurements are generated from the parameters θ according to the following model:

$$\mathbf{x} = \mathbf{H}\theta + \mathbf{v} \tag{2.149}$$

where

$\mathbf{H}_{N\times M}$ is called observation matrix, assumed to be known
$\mathbf{v}$ is a vector of unknown r.v.'s, called measurement errors or noise

The observation matrix is full rank with $N \geq M$, which means that the number of observations is at least equal to the number of parameters to be estimated.

The estimation method consists of applying the LS criterion to the measurement errors. Taking the linear model in account, it comes that

$$\mathcal{E}_{\text{LS}} = \frac{1}{2}\|\mathbf{v}\|^2 = \frac{1}{2}\left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right)^H \left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right) \tag{2.150}$$

In order to find the estimator that minimizes $\mathcal{E}_{\text{LS}}$, we force to zero its derivatives in relation to the parameter vector. It leads to the following expression:

$$\left(\mathbf{H}^H\mathbf{H}\right)\widehat{\boldsymbol{\theta}}_{\text{LS}} = \mathbf{H}^H\mathbf{x} \tag{2.151}$$

which gives rise to the following expression for $\widehat{\theta}$

$$\widehat{\boldsymbol{\theta}}_{\text{LS}} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{H}^H\mathbf{x} = \mathbf{H}^\dagger\mathbf{x} \tag{2.152}$$

where $\mathbf{H}^\dagger$ is the pseudo-inverse of H, assuming $N > M$ and that $\mathbf{H}$ is a full rank matrix.

An interesting property concerning the LS solution is that the optimal parameters $\widehat{\theta}$ engender a residual vector $\mathbf{v}$ that is orthogonal to the column space of the matrix $\mathbf{H}$. In other words, the optimal parameter vector in the LS sense originates the "shortest possible $\mathbf{v}$," i.e., the shortest distance between the measures and the subspace spanned by the columns of $\mathbf{H}$. This geometrical interpretation establishes the equivalence between the LS method and the so-called orthogonality principle: to minimize the LS criterion corresponds to searching for the parameters that provide a residual vector that is normal to the subspace spanned by the columns of $\mathbf{H}$. Due to this property, Equation 2.151 is commonly referred to as normal equation.

The above approach may be extended to the nonlinear case if we generalize the model in (2.149). For the nonlinear least squares (NLS) estimator, we define the generating data model as

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) + \mathbf{v} \tag{2.153}$$

where $\mathbf{f}$ is a vector of nonlinear functions, continuously differentiable w.r.t the parameter θ. The NLS criterion may be defined as

$$\mathcal{E}_{\text{NLS}} = [\mathbf{x} - \mathbf{f}(\theta)]^H [\mathbf{x} - \mathbf{f}(\theta)] \tag{2.154}$$

and the nonlinear estimator θ minimizes $\mathcal{E}_{\text{LS}}$. Such procedure, however, does not lead to a closed-form solution as in (2.152).

2.6 Concluding Remarks

In this chapter, we discussed the fundamental concepts on signals, systems, and statistical processing to be used in the book.

Basic definitions and useful properties of the classical theory of signals and systems were briefly recalled in a systemized way. Description of signals and systems in the time and the frequency domain was carried out. Specific aspects concerning discrete-time signal processing have been emphasized. The sampling theorem was enunciated and the problem of discrete filtering was discussed.

As far as random signals are concerned, the foundations of probability theory were presented. Random variables were defined and characterized by the cumulative distribution and the pdf's. Partial characterization was provided by means of the important concepts of moments and cumulants. These concepts have been extended toward the notion of stochastic process, a suitable model to represent information signals, which are implicitly endowed with uncertainty. Some special cases of statistical behavior were discussed.

We finished the chapter with a brief section on estimation theory. We focused on the methods that are more relevant for the sequence of the book. In particular, Chapter 3 deals with the search of optimal parameters of a linear filter. This problem can be viewed as a special case of estimation and requires a proper understanding of the concepts and tools discussed in this chapter.

3

Linear Optimal and Adaptive Filtering

As discussed in Chapter 2, linear estimation is a crucial problem in a number of important applications, among which the search for optimal parameters of a filter is particularly relevant. We have also pointed out in Section 2.2 that the project of a filter involves three fundamental steps: to select a suitable structure, to establish a criterion for the filter optimization, and to employ a technique to find the optimal parameters.

In this chapter, we will proceed with the mentioned steps under the following considerations: First, we will focus on discrete-time filters and assume that the filtering structure is a linear combiner, among which the finite impulse response (FIR) filter is a typical case. Moreover, unless stated otherwise, we consider that all signals are real-valued. Second, among the estimation criteria previously discussed, we will focus on the minimum mean-squared error (MMSE) and the LS estimators, since both of them lead to a linear problem with closed-form solutions. Finally, in addition to finding closed solutions, we will be interested in deriving *adaptive techniques* to attain the optimal parameters.

The scenario to be considered throughout this chapter is one of *supervised filtering*, i.e., the optimization process is guided by an available reference signal. This scenario plays an important role either as a performance reference or as a practical solution, in cases in which the reference signal is available. It is also important to establish the foundations of supervised filtering theory in order to pave the way to the study of non-supervised techniques, which is the main interest of this book. This is essentially the objective of this chapter, which is organized as follows:

- In Section 3.1, we discuss the basic idea of supervised filters to clarify their scope of application as well as their limitations. Emphasis is given to three emblematic filtering problems: *identification*, *deconvolution*, and *prediction*. In this context, we introduce the problem of *channel equalization*, which is a central subject of the book.
- Section 3.2 presents a most relevant concept, that of Wiener filtering, which is obtained through the application of the MMSE criterion to a linear FIR structure. The relevance of the Wiener result is briefly considered in the historical notes.
- Section 3.3 describes the so-called *steepest-descent algorithm*, an iterative solution to the Wiener filtering problem, based on the gradient

method. In fact, this technique is presented as a kind of first step toward truly adaptive methods, which constitute our main interest.

- Section 3.4 deals with the adaptive case and presents a most relevant technique: *the least mean square* (*LMS*) *algorithm*. We present LMS in a rather canonical way, i.e., as a stochastic approximation of the steepest-descent method.
- In Section 3.5, we introduce the *method of least squares* (*LS*), which is the application of the LS estimation criterion to a linear FIR filter. In contrast with Wiener theory, the LS solution does not involve statistical averages but depends on a set of available data. The optimal solution can also be obtained in a recursive way, which gives rise to the so-called *recursive least squares* (*RLS*) *algorithm*.
- Although the main interest of this chapter is in linear FIR filters, Section 3.6 discusses alternative approaches, for the sake of completeness. A more in-depth discussion on nonlinear structures will be presented in Chapter 7.
- In Section 3.7, we turn our attention to the problem of filtering when a set of constraints on the filter parameters replaces the reference signal in the optimization process. Our main motivation is presenting the constrained filtering case as a sort of bridge between the linear filter theory and the non-supervised problem.
- Finally, in Section 3.8, we revisit the special case of *linear prediction*, in order to discuss some particularly important properties. This discussion results in a connection between prediction and equalization problems, which will be exploited in subsequent chapters.

Historical Notes

As is usual in textbooks, and for didactic reasons, this chapter starts from a discussion about Wiener theory, which is founded on the MMSE estimation criterion. However, this precedence is not historical, since the LS method, the development of which is attributed to Gauss, dates from 1795 [119], although Legendre first published it in 1810 [178].

The development of estimation theory in a stochastic context is derived from the advances in statistic inference and probability theory. The application of the MMSE criterion in the linear prediction problem gave rise to the *modern filtering theory*, thanks to the works of Kolmogorov [170], in 1939, and of the American mathematician Norbert Wiener during the 1940s, the latter definitely published in 1949 [305]. Kolmogorov oriented his work toward discrete-time stationary processes, and his works were complemented by those of Mark Krein, an eminent pupil of his. Wiener formulated the continuous-time optimal predictor, which required the solution of an integral equation, known as *Wiener–Hopf equation*, previously developed by the two authors in 1931 [306].

The independence, as well as the relationships, between these two seminal works is attested by the following words of Norbert Wiener himself in [305]:

> When I first wrote about prediction theory, I was not aware that some of the main mathematical ideas had already been introduced in the literature. It was not long before I found out that just before the Second World War an important little paper on the same subject had been published by the Russian mathematician Kolmogorov in the Comptes Rendus of the French Academy of Sciences. In this, Kolmogorov confined himself to discrete prediction, while I worked in a continuous time

As posed by Kailath in his review paper on linear filter theory [162], a fundamental reference we use in these brief notes, Wiener stressed "the engineering significance of his ideas" and "was also conscious of the problems of actually building circuits to implement the theoretical solutions." Such concerns, together with the theoretical relevance of the mentioned results, seem to be crucial for the growing application of linear filtering in different fields of engineering from the second half of the last century on.

Two important results are not exposed in this chapter, but must be mentioned as central to the history of filtering. Such results involve recursive solutions to optimal filtering: the *Levinson–Durbin algorithm*, which provides a recursive-in-order procedure and the Kalman filter, which provides a recursive-in-time solution, by applying the sequential MMSE estimator.

In 1947, Levinson [180] formulated in a matrix form the discrete-time version of the Wiener filter. Levinson also proposed a recursive procedure to solve the matrix form of the Wiener–Hopf equation, based on the so-called *Toeplitz structure* of the matrix. Durbin [106] exploited this result in the particular case of finding the parameters of an autoregressive model of a stationary discrete-time series. In this case, the optimal parameters are obtained by the *Yule–Walker equations*, which can be viewed as a particular case of the Wiener–Hopf equations. This recursive solution is known as the Levinson–Durbin algorithm.

In 1960, Rudolf Kalman formulated the optimal filtering problem in terms of state-space concepts, within the framework of a dynamic system model. Such formulation allows the recursive computing of the optimal MMSE filter by considering the time evolution of the state variables. In consequence, the Kalman filter maintains its optimality in both stationary and nonstationary scenarios, while the Wiener and Kolmogorov solutions were derived in stationary contexts. For this reason, it is worth indicating the important role of the Kalman filter in *adaptive filtering theory* (see [262] for interesting reflections about it), even if these two approaches originated from different philosophical standpoints and had distinct motivations. In Appendix B, we provide a brief review of Kalman filters.

The work of Widrow and Hoff in 1960 is considered to be the most important seminal work on adaptive algorithms, since it gave rise to the celebrated LMS algorithm, also known as stochastic gradient algorithm. The latter name is due to the concept of stochastic approximation, established by Robbins and Monro in 1951 [252], which lies in the very essence of the LMS algorithm.

After Widrow and Hoff's work, a significant amount of important results established the vast literature on adaptive systems along the last 50 years. It worth mentioning here the family of algorithms based on the method of least squares. It is interesting to note in [262] how the idea of a recursive least-squares procedure was first discerned by Gauss. In modern times, the original work is credited to Plackett [240]. In 1994, Sayed and Kailath [263] exposed the exact relationships between the RLS algorithm and Kalman filter theory, which led to the interesting perspectives of using Kalman filtering results within the classical adaptive filtering framework.

3.1 Supervised Linear Filtering

The problem of supervised linear filtering can be easily stated: we have a certain signal processing task to fulfill, and this requires that a filtering configuration be properly designed. The filtering structure performs a linear combination of a set of input samples in order to produce an output according to the values of its free parameters. The fundamental question is how these parameters can be chosen in a systematic and efficient way.

For now, we restrict the universe of possible answers by making an important hypothesis: it is possible to have access to samples of the desired response of the filter to a number of input patterns. In other words, we are able to guide the choice process by establishing some sort of comparison between an "ideal output" and the "actual output." Whenever a procedure of this nature is possible, we speak of supervised filtering. Figure 3.1 represents a basic supervised scheme, in which $x(n)$ is the input signal, $d(n)$ is the desired or reference signal, $y(n)$ is the estimated signal, and $e(n)$ is the error signal.

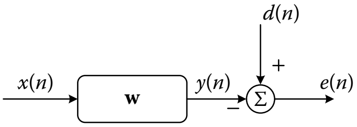


FIGURE 3.1
Supervised filtering scheme.

The estimation of the parameters of a given filter requires a criterion and an optimization procedure to be employed. In principle, there are many paths toward the establishment of a supervised criterion, since there are many ways to explore the information brought by the input signals and the desired output. The Wiener filter is inspired in MMSE estimation, which leads to the minimization of the mean-squared value of the error signal. The LS approach deals with a specific realization of the error signal and does not carry out statistical averages. Before presenting such approaches in detail, it is worth considering some emblematic cases of supervised filtering.

3.1.1 System Identification

The problem of system identification is of great practical importance in several branches of engineering [189], since, in general terms, it consists in building models from available data. Having a model for an unknown system is often very important for analysis, simulation, monitoring, and control of the system. Such a problem may be addressed as a filtering procedure: first, a filtering structure is chosen to serve as the model of the unknown system; second, the parameters of this structure are chosen so that its response be, in some sense, as close as possible to that of the unknown system.

The method can be represented as in Figure 3.2, in which a given input signal feeds both the unknown system and the chosen model. The reference signal $d(n)$ stands for the available measures, which are often subject to additive noise $v(n)$. The model provides an estimated signal so that, by minimizing a cost function related to the error signal, both structures respond as similarly as possible to the same input, which may mean that the unknown system and its proposed model perform similar input–output mappings, if the input signal is adequately chosen.

The input signal must comply with the persistent excitation condition, i.e., it must allow the system to be excited by an adequate variety of modes.

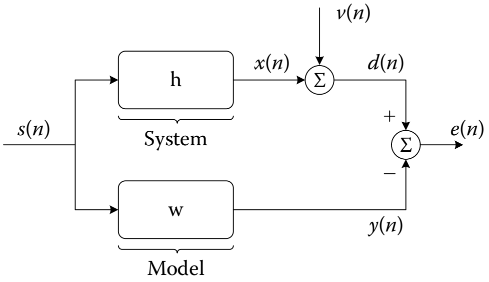


FIGURE 3.2
System identification scheme.

A white noise is classically used as input signal, since its power spectral density is uniformly distributed along the frequency spectrum.

3.1.2 Deconvolution: Channel Equalization

The channel equalization problem can be seen as a dual of the identification problem, as its aim is to find the inverse of an unknown system. A general case of inverse identification or deconvolution is shown in Figure 3.3. In this case, the desired signal is the system input $s(n)$, which is to be recovered.

In the specific case of a digital communication system, we must consider that the signal to be transmitted is composed of a sequence of symbols that belong to a finite alphabet. As a consequence, a nonlinear decision device is inserted at the receiver in order to allow a proper symbol recovery.

The discrete-time representation of a communication channel is usually an FIR filter that performs a linear combination of a set of transmitted samples. This is a suitable model to the so-called intersymbol interference (IIS) phenomenon, which corresponds to a superposition of delayed versions of the transmitted signal. The IIS and the additive noise are fundamental limitations of a transmission system.

The parameters of the equalizer must be optimized in order to yield an output signal that approximates a version of the transmitted information. In this condition (the so-called *open-eye* condition), the transmitted symbols can be recovered by employing a nonlinear quantization device at the output of the equalizer.

It is worth pointing out that channel equalization is by nature an unsupervised problem, since the desired response of the equalizer is the transmitted signal, which is unavailable at the receiver. A possible way to overcome this limitation and allow the application of supervised filtering is the use of training sequences. A training sequence is a signal that is transmitted in spite of being known in advance at the receiver, so that it does not carry any useful information, but is responsible for providing a supervised operation mode. After the optimization procedure, the training sequence is interrupted and the system is switched to the mode of information transmission; then an unsupervised technique can be employed to preserve the open-eye condition.

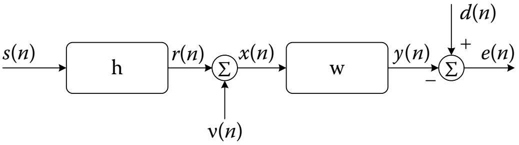


FIGURE 3.3
Equalization scheme.

Since the pioneer work by Lucky in 1965–1966 [190, 191], a vast amount of research efforts has been devoted to this problem. Important contributions were made to different aspect like optimization techniques, equalizer structures, adaptive algorithms, and performance evaluation. An interesting scan of the literature is given by Ding in [99, 245, 246].

3.1.3 Linear Prediction

The idea of estimating future values of a time series from its present and past values engenders one of the most relevant problems in signal processing theory from theoretical and practical standpoints. In simple terms, the problem of one-step prediction consists of finding a mapping $F[\cdot]$ that, when applied to a set of samples of a time series $\mathbf{x}(n-1) = [x(n-1), \ldots, x(n-K)]^T$, yields a suitable estimate of $x(n)$, i.e.,

$$F[\mathbf{x}(n-1)] = \hat{x}(n) \tag{3.1}$$

The generic mapping is typically associated with a certain filtering structure with a set of free parameters. We shall consider the relevant case of linear prediction, when the mapping in (3.1) is a linear combination of the past samples in $\mathbf{x}(n-1)$. Thus, the predictor can be implemented using an FIR filter, the output of which is given by

$$\hat{x}(n) = \sum_{k=0}^{K-1} a_i x(n-1-k) = \mathbf{a}^T \mathbf{x}(n-1) \tag{3.2}$$

where the parameters $a_0, \ldots, a_{K-1}$ are called prediction coefficients. As a consequence, we define the prediction-error signal as

$$e(n) = x(n) - \hat{x}(n) \tag{3.3}$$

where the true value $x(n)$ works as the reference signal, so that the linear prediction problem may be described in terms of a supervised filtering scheme, as shown in Figure 3.4. The mapping from $\mathbf{x}(n-1)$ to $x(n)$ is done by the prediction filter or simply predictor, while the whole structure corresponds to the so-called prediction-error filter (PEF).

The relevance of linear prediction in filtering theory is manifest, even from a historic point of view, since such problem has been dealt with in the fundamental works of Kolmogorov, Wiener, and Levinson, as previously mentioned. Thereafter a great number of important works concerning theoretical results and applications of linear prediction has been presented in the literature. Among them, it is worth mentioning the classical tutorial published by Makhoul in 1975 [197] and the recent book of Vaidyanathan [297]

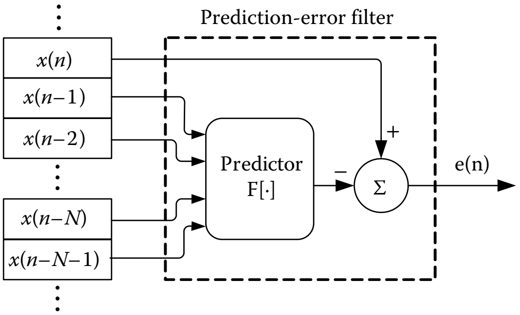


FIGURE 3.4
Scheme of a prediction-error filter.

as excellent ways to have access to the state of the art and to some stimulating insights on the matter.

A central aspect to be discussed in this book is the relationship between prediction and equalization. For this reason, two properties of the PEF will be particularly discussed: that concerning its minimum-phase response and that related to the flat spectral shape of the output prediction error. Such properties emerge from the application of the MMSE criterion to obtain the optimal PEF.

The use of the MMSE criterion in an FIR filter optimization gives rise to the aforementioned Wiener theory, to be formally presented in the sequel. It is worth pointing out that from now on, throughout the chapter, and as a rule in this book, all involved signals are real, zero-mean stationary discrete-time stochastic processes.

3.2 Wiener Filtering

By considering the filtering scheme in Figure 3.1 we may express the output signal as

$$y(n) = \sum_{k=0}^{K-1} w_k x_k(n) = \mathbf{w}^T \mathbf{x}(n) \tag{3.4}$$

where

$\mathbf{w} = [w_0 \ldots w_{K-1}]^T$ is the parameter vector

$\mathbf{x} = [x_0(n) \ldots x_{K-1}(n)]^T$ is the vector that contains the input signals, i.e., the input vector

If we have access to samples of the desired response $d(n)$, it is possible to build an error signal of the form

$$e(n) = d(n) - y(n) \tag{3.5}$$

Now, considering (3.4) and (3.5) together, we note that there is a direct dependence of the error with respect to the free parameters, which is in accordance with our purpose of parameter optimization. If we work with the MMSE criterion, it originates a cost function that relates the mean-squared error (MSE) to the parameter vector **w** in the following form:

$$J_{MSE}(\mathbf{w}) = E\left[e^2(n)\right] = E\left[\left(d(n) - y(n)\right)^2\right] = E\left[\left(d(n) - \mathbf{w}^T\mathbf{x}(n)\right)^2\right] \tag{3.6}$$

which can be rewritten as

$$\begin{aligned} J_{MSE}(\mathbf{w}) &= E\left[e(n)e(n)\right] = E\left[e(n)e^T(n)\right] \\ &= E\left[\left(d(n) - \mathbf{w}^T\mathbf{x}(n)\right)\left(d^T(n) - \mathbf{x}^T(n)\mathbf{w}\right)\right] \end{aligned} \tag{3.7}$$

Further development and proper grouping yields

$$J_{MSE}(\mathbf{w}) = \sigma_d^2 - E\left[d(n)\mathbf{x}^T(n)\right]\mathbf{w} - \mathbf{w}^T E\left[\mathbf{x}(n)d(n)\right] + \mathbf{w}^T E\left[\mathbf{x}(n)\mathbf{x}^T(n)\right]\mathbf{w} \tag{3.8}$$

Aside from the variance of the desired signal, there are three terms that fully characterize the cost function: $E\left[\mathbf{x}(n)\mathbf{x}^T(n)\right]$, which represents the autocorrelation matrix of $x(n)$ (already discussed in Section 2.4.5); $E\left[\mathbf{x}(n)d(n)\right]$ and $E\left[d(n)\mathbf{x}^T(n)\right]$, which correspond to cross-correlation measures. Let us recall their mathematical definitions:

$$\mathbf{R} = E\left[\mathbf{x}(n)\mathbf{x}^T(n)\right] \tag{3.9}$$

$$\mathbf{p} = E\left[\mathbf{x}(n)d(n)\right] \tag{3.10}$$

From (3.9) and (3.10), we may rewrite the cost function (3.8) as

$$J_{MSE}(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^T\mathbf{w} - \mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \tag{3.11}$$

so that the Wiener criterion, i.e., the cost function of the MMSE criterion is established in terms of statistical averages.

3.2.1 The MSE Surface

A relevant aspect of the cost function obtained in (3.11) is that it is a quadratic function of the parameters of the filter and describes an elliptic paraboloid with a single minimum. This means that there is a single parameter vector that minimizes the MSE, the so-called Wiener solution. This is in fact an important reason why the MSE is the dominant metric in linear supervised filtering. In Figure 3.5, we present the MSE cost function and its contours in a typical two-dimensional filtering problem.

The two plots confirm our initial comments and reveal the elliptical character of the cost function contours. Two properties of these elliptical contours deserve attention:

1. Their eccentricity is related to the eigenvalues of the correlation matrix **R** (see Appendix A). The larger the eigenvalues spread, the most significant is the discrepancy between axes. Naturally, if the eigenvalue spread is equal to 1, the contours become circular.
2. The directions of the eigenvectors of the correlation matrix determine the orientation of the axes of the contours.

In order to find the minimum of the MSE cost function, we follow a classical procedure: setting to zero the gradient of J_{MSE}. From (3.11), it comes that

$$\nabla J_{MSE}(\mathbf{w}) = 2\mathbf{R}\mathbf{w} - 2\mathbf{p} \tag{3.12}$$

By forcing it to be equal to the null vector, we obtain

$$\nabla J_{MSE}(\mathbf{w}) = 2\mathbf{R}\mathbf{w} - 2\mathbf{p} = \mathbf{0} \rightarrow \mathbf{R}\mathbf{w} = \mathbf{p} \tag{3.13}$$

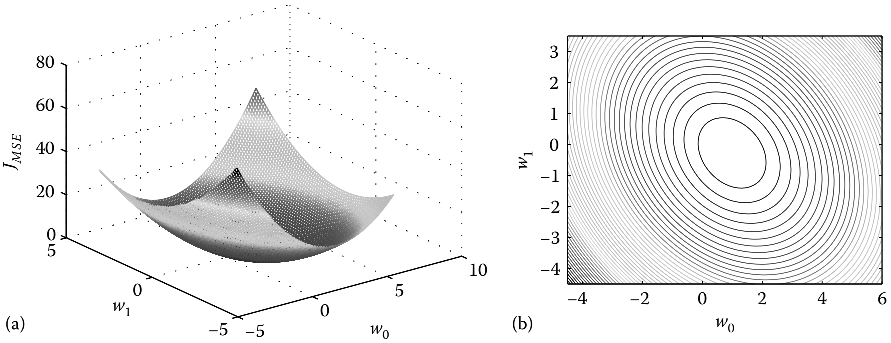


FIGURE 3.5
(a) The MSE cost function and (b) its contours.

Equation (3.13) corresponds to a linear system of equations, known as Wiener–Hopf equations, the solution of which is the Wiener solution:

$$\mathbf{w}_w = \mathbf{R}^{-1}\mathbf{p} \tag{3.14}$$

In a given problem, the application of (3.14) depends on the knowledge of **R** and **p** and on the invertibility of **R**. However, if these conditions are met, the MSE criterion gives rise to a closed-form solution, which is undoubtedly a very strong point.

Now, it is important to keep in mind that to minimize the MSE does not mean that a null-error signal will be produced. In effect, the MSE associated with the Wiener solution is given by

$$\begin{aligned} J_{MSE}(\mathbf{w}_w) &= \sigma_d^2 - \mathbf{p}^T\mathbf{w}_w - \mathbf{w}_w^T\mathbf{p} + \mathbf{w}_w^T\mathbf{R}\mathbf{w}_w \\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{w}_w \\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} \end{aligned} \tag{3.15}$$

As a matter of fact, in most practical cases, the right-hand term in (3.15) does not vanish, which means that the Wiener filter is not able to perfectly reproduce the desired signal. This can be due to factors like the presence of noise, the use of an insufficiently flexible structure, a definitive lack of information about the desired signal in the input signals, etc. Before we proceed to the examples that illustrate it, let us stress a bit more the meaning of the Wiener solution. First, we recall (3.6):

$$J_{MSE}(\mathbf{w}) = E\left[e^2(n)\right] \tag{3.16}$$

From (3.16), it is possible to write

$$\nabla J_{MSE}(\mathbf{w}) = E\left[2e(n)\frac{\partial e(n)}{\partial \mathbf{w}}\right] = -2E\left[e(n)\mathbf{x}(n)\right] \tag{3.17}$$

Forcing (3.17) to be equal to the null vector leads us to the condition

$$E\left[e(n)\mathbf{x}(n)\right] = \mathbf{0} \tag{3.18}$$

This means that the error signal produced by the Wiener solution is orthogonal to all input signals. In a certain sense, the residual error is a parcel of the desired signal that cannot be built due to its being uncorrelated with the input signal. Since the residual error is orthogonal to all input signals, the output of the Wiener filter is also orthogonal to the error signal.

It is useful now to revisit some aforementioned applications in order to discuss concrete examples.

Example 3.1 (Channel Equalization)

Let us consider the problem of channel equalization, described in Figure 3.3, and suppose the following received signal at the output of the channel:

$$x(n) = s(n) + 0.4s(n-1) \tag{3.19}$$

where the transmitted signal $s(n)$ is composed of binary (+1/−1) independent and identically distributed (i.i.d.) samples. The equalizer is an FIR linear filter with K coefficients, and its input vector given by

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-K+1)]^T \tag{3.20}$$

Let us consider that $K = 2$. Recalling that the transmitted signal is composed of uncorrelated samples, we may write the correlation matrix as

$$\mathbf{R} = \begin{bmatrix} 1.16 & 0.4 \\ 0.4 & 1.16 \end{bmatrix} \tag{3.21}$$

since

$$\begin{aligned} r(0) &= E\left[x(n)x(n)\right] = E\left[(s(n) + 0.4s(n-1))(s(n) + 0.4s(n-1))\right] \\ &= E\left[s^2(n)\right] + 0.16E\left[s^2(n-1)\right] = 1.16 \end{aligned} \tag{3.22}$$

and

$$\begin{aligned} r(1) &= E[x(n)x(n-1)] = E\left[(s(n) + 0.4s(n-1))\,(s(n-1) + 0.4s(n-2))\right] \\ &= 0.4E\left[s^2(n-1)\right] = 0.4 \end{aligned} \tag{3.23}$$

Now we need the cross-correlation vector to derive the Wiener solution, but it must be preceded by a clear determination of what is the desired signal. In the training mode a natural choice can be

$$d(n) = s(n) \tag{3.24}$$

The cross-correlation vector is

$$\mathbf{p} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{3.25}$$

as

$$p(0) = E\left[s(n)\,(s(n) + 0.4s(n-1))\right] = E\left[s^2(n)\right] = 1 \tag{3.26}$$

and

$$p(1) = E\left[s(n)\,(s(n-1) + 0.4s(n-2))\right] = 0 \tag{3.27}$$

Finally the Wiener solution is reached:

$$\mathbf{w}_W = \mathbf{R}^{-1}\mathbf{p} = \begin{bmatrix} 0.978 \\ -0.337 \end{bmatrix} \tag{3.28}$$

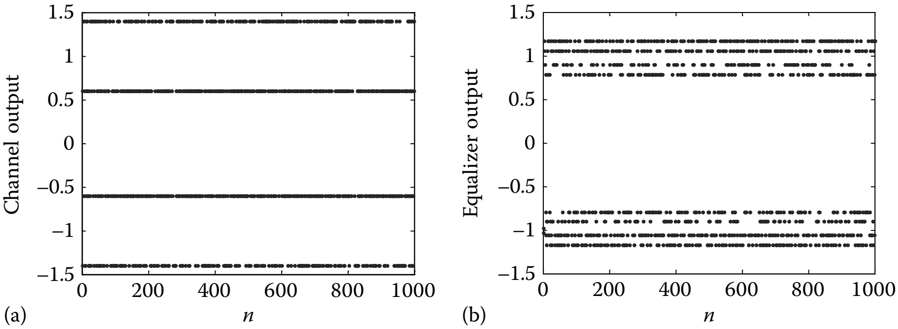


FIGURE 3.6
(a) Channel and (b) equalizer outputs in the absence of noise.

As mentioned above, the optimal MSE is not necessarily null. In fact, this means that some kind of performance analysis of the Wiener solution must always be carried out. In this example, we may inquire whether the Wiener equalizer is efficient enough to mitigate IIS and provide correct symbol recovery. In order to answer this question, let us first calculate the MSE associated with the Wiener solution:

$$J_{MSE}(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^T\mathbf{w}_W = 1 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.978 \\ -0.337 \end{bmatrix} = 0.0216 \tag{3.29}$$

In view of the magnitude of the involved signals, this residual MSE indicates that the equalization task has been carried out efficiently. Such conclusion is confirmed by Figure 3.6, which shows the channel and equalizer outputs. The latter is indeed concentrated around the corrected symbols +1 and −1.

Let us now consider what happens if the channel also introduces additive noise, assumed to be Gaussian and white (which leads to the classical acronym AWGN—additive white Gaussian noise) and, moreover, independent of the transmitted signal. Thus, the received signal will be

$$x(n) = s(n) + 0.4s(n-1) + \nu(n) \tag{3.30}$$

where $\nu(n)$ stands for the AWGN, with variance σ_ν^2. From (3.30), the correlation matrix becomes

$$\mathbf{R} = \begin{bmatrix} 1.16 + \sigma_\nu^2 & 0.4 \\ 0.4 & 1.16 + \sigma_\nu^2 \end{bmatrix} \tag{3.31}$$

while the correlation vector remains as shown in (3.25). The Wiener solution is now given by

$$\mathbf{w}_W = \mathbf{R}^{-1}\mathbf{p} = \begin{bmatrix} 1.16 + \sigma_\nu^2 & 0.4 \\ 0.4 & 1.16 + \sigma_\nu^2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.32}$$

In order to evaluate the noise effect, we consider three different variance levels: $\sigma_1^2 = 0.1$, $\sigma_2^2 = 0.01$, $\sigma_3^2 = 0.001$. Table 3.1 presents the Wiener solutions for the different levels of noise, as well as the residual MSE's.

TABLE 3.1

Comparison between Wiener Solutions for Different Noise Levels

Noise Variance	Wiener Solution	EQM
No noise	$[0.978, -0.337]^T$	0.0216
0.001	$[0.977, -0.337]^T$	0.0227
0.01	$[0.968, -0.331]^T$	0.0322
0.1	$[0.883, -0.280]^T$	0.1174

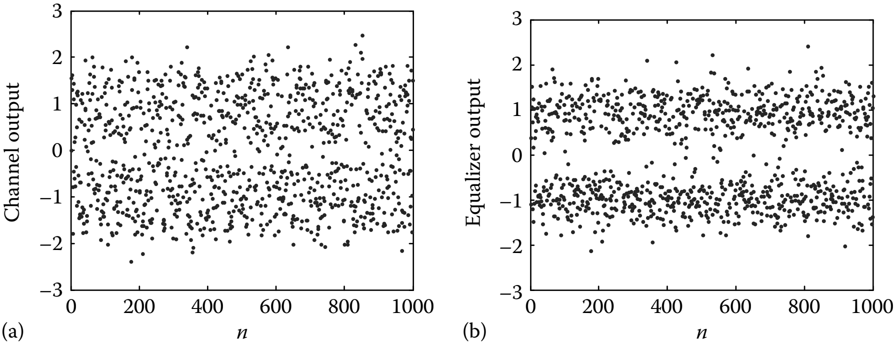


FIGURE 3.7
(a) Channel and (b) equalizer outputs in the presence of noise.

It is clear from Table 3.1 that the addition of noise modifies the performance of the optimal equalizer. In fact, the minimization of the MSE leads the equalizer to attempt to solve two distinct tasks: to cancel the IIS and to mitigate the noxious effects of the noise. Since the number of parameters is fixed, this "double task" becomes more difficult as the additive noise is more significant, which is reflected by an increase in the residual MSE. This is also illustrated in Figure 3.7, in which the channel and equalizer outputs are presented for the case in which $\sigma_\nu^2 = 0.1$. In such case, recovery is not perfect, which confirms that two optimal solutions (those corresponding to Figures 3.6 and 3.7) can present distinct performance behaviors according to the conditions under which the filtering task is accomplished.

Example 3.2 (System Identification)

According to Figure 3.2, let us suppose that the unknown system to be identified is characterized by the following response:

$$x(n) = h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \nu(n) \tag{3.33}$$

where

$\nu(n)$ is the AWGN
$s(n)$ is composed by i.i.d. samples with unit variance

If we consider an efficient model, in this case an FIR filter with three coefficients, the corresponding input vector is

$$\mathbf{s}(n) = \left[s(n) \quad s(n-1) \quad s(n-2)\right]^T \tag{3.34}$$

and the correlation matrix is given by

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.35}$$

To obtain the cross-correlation vector, it is necessary to determine the desired signal. From the above discussion, the error signal must express a comparison between the system and the model outputs. This leads to the natural choice,

$$d(n) = x(n) \tag{3.36}$$

Since the signal $s(n)$ and the noise are mutually independent, it follows that

$$\begin{aligned} p(0) &= E\left[x(n)s(n)\right] \\ &= E\left[\left(h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \nu(n)\right) s(n)\right] \\ &= h_0 E\left[s^2(n)\right] \\ &= h_0 \end{aligned} \tag{3.37}$$

$$\begin{aligned} p(1) &= E\left[x(n)s(n-1)\right] \\ &= E\left[\left(h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \nu(n)\right) s(n-1)\right] \\ &= h_1 E\left[s^2(n-1)\right] \\ &= h_1 \end{aligned} \tag{3.38}$$

and

$$\begin{aligned} p(2) &= E\left[x(n)s(n-2)\right] \\ &= E\left[\left(h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \nu(n)\right) s(n-2)\right] \\ &= h_2 E[s^2(n-2)] \\ &= h_2 \end{aligned} \tag{3.39}$$

so that

$$\mathbf{p} = \begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} \tag{3.40}$$

and finally

$$\mathbf{w}_W = \mathbf{R}^{-1}\mathbf{p} = \begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} \tag{3.41}$$

We can note that the above solution captures the essential information about the system and is not influenced by the noise. The corresponding residual MSE can be calculated by using (3.15). First, the variance of the desired signal is obtained by

$$\begin{aligned}\sigma_d^2 &= E\left[d^2(n)\right] \\ &= E\left[(h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \nu(n))^2\right] \\ &= h_0^2 + h_1^2 + h_2^2 + \sigma_\nu^2 \end{aligned} \tag{3.42}$$

so that

$$\begin{aligned} J_{MSE}(\mathbf{w}_W) = \sigma_d^2 - \mathbf{p}^T\mathbf{w}_W &= h_0^2 + h_1^2 + h_2^2 + \sigma_\nu^2 \\ &- \begin{bmatrix} h_0 & h_1 & h_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} = \sigma_\nu^2 \end{aligned} \tag{3.43}$$

The interpretation of the obtained results is direct: if the order of the model is sufficient, its parameters exactly fit the coefficients of the system, and the residual MSE corresponds to the parcel due to noise, which cannot be modeled. Let us now analyze what happens if a model of insufficient order (for instance, a two-tap FIR filter) is chosen. In such case, the correlation matrix is the 2 × 2 identity matrix, the cross-correlation vector is given by

$$\mathbf{p} = \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} \tag{3.44}$$

and the corresponding solution is

$$\mathbf{w}_W = \mathbf{R}^{-1}\mathbf{p} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} \tag{3.45}$$

The lack of one coefficient leads to an increased MSE, if compared with (3.43):

$$J_{MSE}(\mathbf{w}_W) = \sigma_d^2 - \mathbf{p}^T\mathbf{w}_W = \sigma_\nu^2 + h_2^2 \tag{3.46}$$

This simple example illustrates the difficulties that arise from the choice of a model with insufficient approximation capability.

To summarize the discussion, we can observe that the use of the supervised MMSE criterion together with a linear structure makes the Wiener approach a kind of paradigm for optimal filtering methods. However, it is important to keep in mind two assumptions on which Wiener filter is founded: the involved signals are wide-sense stationary; and the statistical averages, **R** and **p**, are known. This means that the process of acquisition of the involved signals, $s(n)$ and $x(n)$, precedes the process of optimization of the filter, i.e., the calculation of the optimal parameters via (3.14).

Now our attention must be turned to two aspects that are really frequent in many practical cases: the need for real-time operation, and the presence of nonstationary signals. In fact, these two requests violate the assumptions presented above. The real-time constraint will require methods that provide a joint process of acquisition and optimization, while the nonstationary context will inhibit the use of a closed-form solution, as in (3.14), since there will be no sense in dealing with fixed values of statistical correlations. This new scenario leads us to the frontier between *optimal* and adaptive filtering, or rather, between methods that are based on closed-form solutions and those based on iterative/recursive solutions for the linear filtering problem. A classical and didactic way to verify this is to consider first a simple iterative method to attain the Wiener solution.

3.3 The Steepest-Descent Algorithm

We are interested in establishing a kind of learning process that eventually leads to the optimal solution. The answer to this question is directly related to the optimization theory: in many practical problems, the only feasible optimization approach is exactly to resort to iterative processes. To obtain the Wiener solution was indeed part of an optimization task, albeit a closed-form solution prevented us from considering iterative approaches. However, in view of the questions we have just raised, it appears to be natural that we have to turn our attention toward them.

The iterative approach to be considered now is based on a simple idea, i.e., to use the gradient vector of the cost function as a guide to the learning process. This is the core of the steepest-descent approach [139], which allows that a local minimum be found by taking successive steps in a direction opposite to that indicated by the gradient vector. Mathematically, the steepest-descent algorithm is an iterative optimization process of the form

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla J(\mathbf{w}) \tag{3.47}$$

where

$J(\mathbf{w})$ is the cost function to be optimized
μ is the step size

The application of this method within the Wiener filtering problem is basically a matter of using the calculated gradient vector (3.12) in (3.47). This leads to

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla J_{MSE}\left[\mathbf{w}(n)\right] = \mathbf{w}(n) - 2\mu\left[\mathbf{R}\mathbf{w}(n) - \mathbf{p}\right] \tag{3.48}$$

From now on, we will consider that the factor 2 will be incorporated to the step size, so that the iterative process to attain the Wiener solution is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\left[\mathbf{R}\mathbf{w}(n) - \mathbf{p}\right] \tag{3.49}$$

In order to evaluate the effectiveness of such iterative procedure, we must analyze the equilibrium points of the steepest-descent algorithm as well as its properties of convergence. In view of this objective, we can note that (3.49) corresponds to a linear dynamical system whose state variables are the filter coefficients. The equilibrium points of a discrete-time dynamical system are the points that are invariant to the iterative process. In other words, these points are the solutions of the following equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) \rightarrow \mu\left[\mathbf{R}\mathbf{w}(n) - \mathbf{p}\right] = 0. \tag{3.50}$$

which can be simplified to yield the equilibrium point

$$\mathbf{w_e} = \mathbf{R}^{-1}\mathbf{p}. \tag{3.51}$$

Thus, the system has a single equilibrium point, which is not surprising, since the system is linear and this point is exactly the Wiener solution. However, the convergence is not guaranteed since we have not yet verified under what conditions this equilibrium point is stable. In order to proceed with this verification using concepts of dynamical system theory, let us first write (3.49) in the following form:

$$\mathbf{w}(n+1) = \left[\mathbf{I} - \mu\mathbf{R}\right]\mathbf{w}(n) + \mu\mathbf{p} \tag{3.52}$$

The stability of such a system depends on the eigenstructure of the following matrix [139]:

$$\mathbf{B} = \mathbf{I} - \mu\mathbf{R}. \tag{3.53}$$

If all the eigenvalues of $\mathbf{B}$ lie inside the unit circle, the Wiener solution will be a stable equilibrium point. On the other hand, if a single eigenvalue is outside the unit circle, the entire scheme will be compromised. Clearly, the stability of the steepest-descent algorithm is strongly dependent on the step size μ, which serves as a sort of control parameter. The eigenvalues of $\mathbf{B}$ are

$$\lambda_{\mathbf{B}} = 1 - \mu\lambda_{\mathbf{R}} \tag{3.54}$$

where $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{R}}$ stand for a generic pair of eigenvalues. If all eigenvalues of $\mathbf{B}$ are to be inside the unit circle, it is necessary that the following condition hold:

$$|1 - \mu\lambda_{\mathbf{R}}| < 1 \tag{3.55}$$

for all eigenvalues of **R**. The most stringent case will be reached when the largest eigenvalue is considered, i.e.,

$$\left|1 - \mu\lambda_{\mathbf{R}_{\max}}\right| < 1 \tag{3.56}$$

which means that

$$-1 < 1 - \mu\lambda_{\mathbf{R}_{\max}} < 1 \tag{3.57}$$

Since both μ and $\lambda_{\mathbf{R}_{\max}}$ are nonnegative, the relevant inequality is

$$-1 < 1 - \mu\lambda_{\mathbf{R}_{\max}} \tag{3.58}$$

which yields

$$\mu < \frac{2}{\lambda_{\mathbf{R}_{\max}}} \tag{3.59}$$

This condition relates the convergence of the algorithm to a statistical property of the input signal. This is a particular characteristic of gradient-based iterative techniques. Let us now consider an example to complete this topic.

Example 3.3 (Channel Equalization Revisited)

Let us return to the noiseless scenario of Example 3.1, but now our aim is to search for the optimal filter using the steepest-descent approach. First, let us verify the step-size upper bound. An analysis of the correlation matrix presented in (3.21) yields

$$\lambda_{\mathbf{R}_{\max}} = 1.56 \rightarrow \mu < \frac{2}{1.56} \rightarrow \mu < 1.282 \tag{3.60}$$

Then, we arbitrarily choose two step-sizes, $\mu = 0.1$ and $\mu = 1$. We also consider the initial condition $\mathbf{w}(0) = [0, 0]^T$ and a number of 1000 iterations per run. In Figure 3.8, we present the time evolution of the coefficients for both step-sizes.

The figure clearly shows that in the case of the steepest-descent algorithm, the step-size is basically related to the convergence rate of the algorithm: the larger the step-size, the faster the convergence (within the stability bounds). In other words, the step-size regulates the characteristics of the transient response of the dynamic system, whereas the Wiener solution to be reached determines the equilibrium point to which the algorithm converges.

Another interesting way to study the evolution of the coefficients is to analyze it against the frame of the contours of the MSE cost function. In Figure 3.9 we plot the trajectories associated with both choices of the step-size. These trajectories reveal a clear limitation of the steepest-descent method, which arises from its

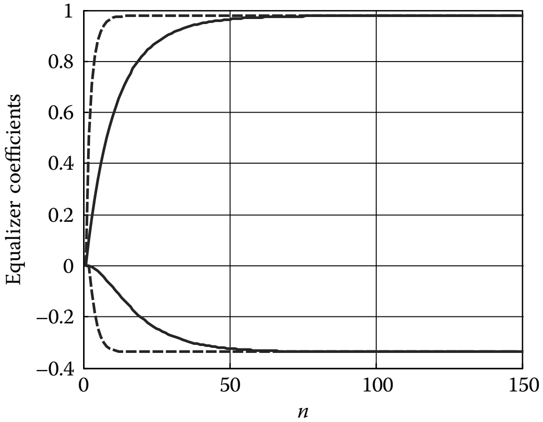


FIGURE 3.8
Time evolution of the free parameters for $\mu = 0.1$ (solid line) and $\mu = 0.5$ (dashed line).

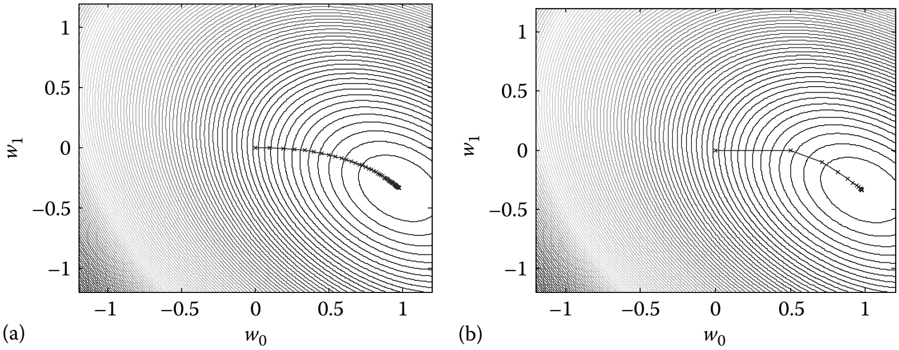


FIGURE 3.9
Convergence of the steepest-descent algorithm: (a) $\mu = 0.1$ and (b) $\mu = 0.5$.

exclusive use of the first-order derivatives: the trajectory is not directly oriented from the initial condition to the Wiener solution, but follows the gradient direction, i.e., a direction orthogonal to the contours.

The example also gives us a favorable opportunity to discuss a bit more about the shape of the contours of the MSE cost function. As previously mentioned, this shape is associated with an eigenanalysis of the correlation matrix and the axes of the elliptical contours are determined by its eigenvectors. In Figure 3.10, we revisit Figure 3.5 and include the directions of the two eigenvectors of the correlation matrix to illustrate this point. Such illustration indicates that the directions of the eigenvectors are, in a certain sense, "special directions of convergence." The direction associated with the minor axis works as a favorable direction, i.e., the best path to take if we are standing on a certain contour. In contrast, the direction associated with the major axis is a kind of unfavorable direction.

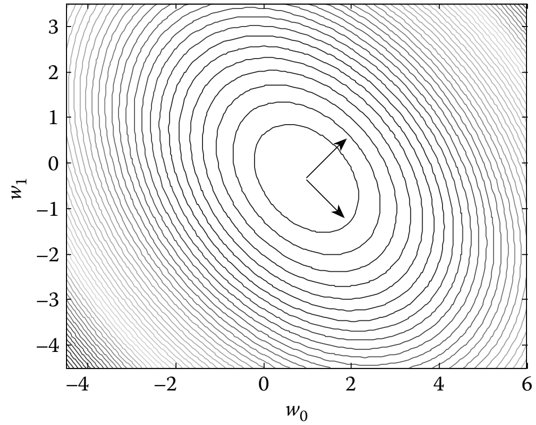


FIGURE 3.10
Contours of the MSE cost function and directions of the eigenvectors of **R**.

The discussed example illustrates the efficiency of the steepest-descent algorithm in searching for the optimal Wiener solution. However, it is worth reconsidering the motivations in using such procedure, i.e., the idea of carrying out the processes of data acquisition and filter optimization jointly. In fact, it is evident from (3.49) that the steepest-descent algorithm does not accomplish such requirement, since the knowledge of statistical correlation (matrix **R** and vector **p**) is as indispensable to calculate the gradient vector as to solve the Wiener–Hopf equations. This means that an alternative methodology must be introduced in order to cross over the frontier from optimal (fixed) to adaptive filtering definitively. To proceed with such discussion, the fundamental idea of stochastic algorithm is now introduced.

3.4 The Least Mean Square Algorithm

The pioneer works on adaptive filtering date from the 1950s. From that time, this field of research has been significantly developed and originated a wide range of applications, methods, algorithms, and tools of analysis. Moreover, adaptive filtering became a well-established discipline in modern signal processing theory, with a number of relevant and classical textbooks, like [32,100,139,194,262,304] to mention a few.

Among the great number of efficient techniques, the *LMS* algorithm is classically considered to be the "most popular" one as well as the basis of many others, the so-called LMS-based algorithms. The most usual and accessible way to introduce the LMS algorithm is by means of the concept of stochastic approximation. Such concept was first posed and theoretically justified by Robbins and Monro in the context of providing iterative parameter estimation based on random observations [252].

The Robbins–Monro problem is closely related to the idea of finding a set of optimal filter coefficients while a random signal is acquired. So the LMS algorithm is in fact an application of the stochastic approximation principle to the steepest-descent algorithm (which is in fact also known as deterministic-gradient algorithm), using a stochastic estimation of the gradient vector and a fixed step-size. This simple and extremely efficient idea is historically attributed to Widrow and Hoff [303]. The stochastic approximation employed in LMS is straightforward: it consists in replacing the correlation matrix and the cross-correlation vector by instantaneous and unbiased estimates. In doing so, we get

$$\hat{\mathbf{R}} = \mathbf{x}(n)\mathbf{x}^T(n) \tag{3.61}$$

and

$$\hat{\mathbf{p}} = d(n)\mathbf{x}(n) \tag{3.62}$$

so that the stochastic gradient vector becomes

$$\hat{\nabla}J\left[\mathbf{w}(n)\right] = \left(\mathbf{x}(n)\mathbf{x}^T(n)\right)\mathbf{w}(n) - d(n)\mathbf{x}(n) \tag{3.63}$$

If we apply the above expressions in the steepest-descent algorithm, it follows that

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\left[\hat{\mathbf{R}}\mathbf{w}(n) - \hat{\mathbf{p}}\right] \\ &= \mathbf{w}(n) - \mu\left[\left(\mathbf{x}(n)\mathbf{x}^T(n)\right)\mathbf{w}(n) - d(n)\mathbf{x}(n)\right]\end{aligned} \tag{3.64}$$

or rather,

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu[y(n) - d(n)]\mathbf{x}(n) \\ &= \mathbf{w}(n) + \mu[d(n) - y(n)]\mathbf{x}(n)\end{aligned} \tag{3.65}$$

which is the expression of the LMS algorithm.

Now, it is useful to revisit the previous example to illustrate the application of the LMS and to establish some comparisons with the steepest-descent algorithm.

Example 3.4 (Channel Equalization with the LMS Algorithm)

Let us return to the equalization problem studied in Example 3.3. Now, we will use $\mu = 0.1$ and the same initial condition adopted for the LMS algorithm.

Figure 3.11 depicts the time evolution of the equalizer coefficients for the two discussed techniques. The evolution of the LMS against the contours of the MSE is presented in Figure 3.12.

These curves clearly reveal the stochastic character of the LMS, which is a direct consequence of the employed approximation. Nevertheless, the LMS is capable of leading the coefficient vector to the vicinity of the Wiener solution. The path toward the goal is notwithstanding much more "winding," and the convergence is characterized by a persistent degree of stochastic

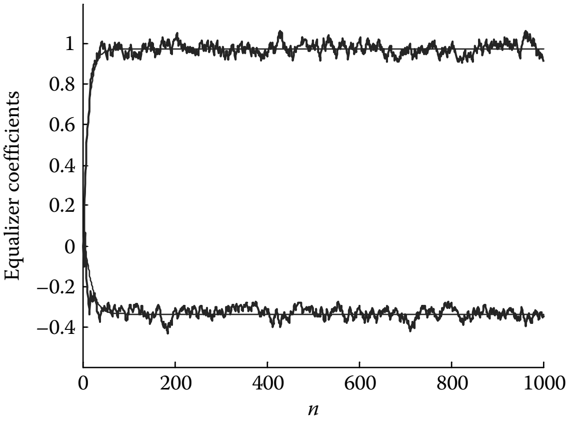


FIGURE 3.11
Time evolution of the equalizer coefficients—LMS (solid line) and steepest descent (dashed line).

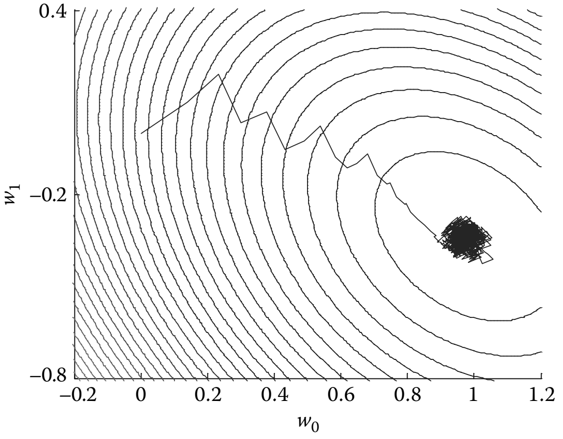


FIGURE 3.12
Convergence of the LMS algorithm against the contours of the MSE surface.

fluctuation around the optimum. These issues must be taken into account whenever a performance analysis of the LMS algorithm is carried out.

The random behavior of the LMS algorithm causes its convergence analysis to be considerably harder than that of the steepest-descent algorithm. The crucial point is the potential presence of a persistent fluctuation around the optimal point, which is due to the stochastic nature of the algorithm, and leads us to consider the concepts of convergence in the mean and convergence in the mean squared sense.

A classical approach to perform the convergence analysis of the LMS algorithm is based on the so-called independence theory, which takes into account some simplifying hypotheses [100,139,304]. In a way, the agreement between some analytical and experimental results justify the hypotheses of the independence theory, even if they are not rigorously true in the context of adaptive FIR filters.

By using independence theory, the maximum value of the step-size obtained for the LMS is similar to that exposed in (3.59). A more conservative value can be obtained if the maximum eigenvalue is replaced with the sum of all eigenvalues, which corresponds to the trace of the correlation matrix, or rather, to the total input power. This reveals that the stability of the LMS is also related to the correlation structure of the input signal.

A second point requires the introduction of the notion of misadjustment of the algorithm defined by

$$\mathcal{M} = \frac{J_{MSE}(\mathbf{w}(\infty)) - J_{MSE}\left(\mathbf{w}_w\right)}{J_{MSE}\left(\mathbf{w}_w\right)} \tag{3.66}$$

which provides a measure of the discrepancy between the steady-state MSE obtained via the LMS and the ideal steady-state MSE. This misadjustment grows with the increase of the step-size, which creates a tradeoff between speed of convergence and precision. This is an important aspect to be taken into account by the designer whenever the LMS is used [100].

Recently, Haykin introduced an alternative approach for the convergence analysis of LMS [140]. Such approach is based on a Markovian representation of the LMS algorithm and provides a quite interesting result: the LMS algorithm performs a Brownian motion around the optimal Wiener solution after a large enough number of iterations. The assumptions employed in Haykin's approach are less restrictive than those required by independence theory, and they are as follows:

1. The step-size μ is sufficiently small.
2. The error signal produced by the Wiener filter is uncorrelated.
3. The input vector $\mathbf{x}(n)$ and the desired signal $d(n)$ are jointly Gaussian.

By using these hypotheses together with the so-called Kushner's direct-averaging method [172], and applying an orthogonality transformation over the coefficient vector of the filter, it can be shown that the resulting transformed vector follows a stochastic recursion recognized as the discrete-time version of the Langevin equation [140]. In thermodynamics, such equation describes the Brownian motion of a particle in a viscous fluid. So, Haykin's result explains the Brownian motion executed by the LMS algorithm around the optimal Wiener solution, after a large enough number of iterations.

3.5 The Method of Least Squares

In the previous sections, the methods for finding the optimal parameters of a filter were established within a statistical framework: the input and desired signals are considered to be stochastic processes; the MSE criterion is a statistical average, and the same is valid for the correlation measures **R** and **p**; the stochastic approximation replaces these statistical averages by instantaneous estimates to make possible the development of a fully adaptive technique, the LMS. In a few words, and in accordance with classical authors like Haykin [139], we can say that the LMS and other algorithms derived from it are based on the Wiener filter theory. But this is not the only approach to derive adaptive algorithms.

Certainly, in many practical scenarios, it is an idealization to rely on statistical entities, particularly when the generative model of the data is not known. Nonetheless, it is possible to consider an alternative approach that is more "data-oriented", i.e., based on a given temporal realization of the involved signals, so that a filtering criterion could be built from the available data, for instance, by using time averages, which can be promptly calculated.

In other words, the key is no longer the optimal filter with respect to a measure that takes into account the ensemble underlying the involved stochastic processes, but the optimal filter for the available data we have access to. Although both approaches are deeply interrelated, there is an essential conceptual difference.

In view of the above comments, we may propose a new supervised cost function based on the sum of the squares of an error signal:

$$J_{LS}(\mathbf{w}) = \sum_{k=1}^{N_{samples}} \lambda(k)e^2(k) \tag{3.67}$$

where

$e(n)$ is the error signal

$\lambda(n)$ is a weight factor that can control the degree of relevance of the error produced in the instant n

We assume that $\lambda(n)$ takes the form of a forgetting factor, i.e., that it tends to attenuate the relevance of older samples, which is justifiable, for instance, whenever one deals with a time-varying environment:

$$\lambda(n) = \lambda^{N_{samples}-n} \tag{3.68}$$

where

$0 < \lambda \leq 1$ is a parameter that controls the degree of penalization of older samples (if $\lambda = 1$, they are not penalized at all)

$N_{samples}$ is the number of available samples

The cost function given by (3.67) can be rewritten as

$$\begin{aligned} J_{LS}(\mathbf{w}) &= \sum_{k=1}^{N_{samples}} \lambda(k)e^2(k) \\ &= \sum_{k=1}^{N_{samples}} \lambda(k)e(k)e^T(n) \\ &= \sum_{k=1}^{N_{samples}} \lambda(k)\left[d(k) - \mathbf{w}^T\mathbf{x}(k)\right]\left[d^T(k) - \mathbf{x}^T(k)\mathbf{w}\right] \end{aligned} \tag{3.69}$$

Further manipulation leads to

$$\begin{aligned} J_{LS}(\mathbf{w}) = \sum_{k=1}^{N_{samples}} \Big\{&\lambda(k)d^2(k) - \lambda(k)d(k)\mathbf{x}^T(k)\mathbf{w} - \lambda(k)\mathbf{w}^T\mathbf{x}(k)d(k) \\ &+ \lambda(k)\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}\Big\} \end{aligned} \tag{3.70}$$

which finally yields

$$J_{LS}(\mathbf{w}) = \sum_{k=1}^{N_{samples}} \lambda(k)d^2(k) - \boldsymbol{\pi}(N_{samples})^T\mathbf{w} - \mathbf{w}^T\boldsymbol{\pi}(N_{samples}) + \mathbf{w}^T\Psi(N_{samples})\mathbf{w} \tag{3.71}$$

where

$$\Psi(N_{samples}) = \sum_{k=1}^{N_{samples}} \lambda(k)\mathbf{x}(k)\mathbf{x}^T(k) \tag{3.72}$$

and

$$\boldsymbol{\pi}(N_{samples}) = \sum_{k=1}^{N_{samples}} \lambda(k)\mathbf{x}(k)d(k) \tag{3.73}$$

It can be noted that $\Psi(N_{samples})$ and $\boldsymbol{\pi}(N_{samples})$ are in fact temporal estimates of the correlation matrix and of the cross-correlation vector, respectively. The gradient of (3.71) is

$$\nabla J_{LS}(\mathbf{w}) = -2\boldsymbol{\pi}(N_{samples}) + 2\Psi(N_{samples})\mathbf{w} \tag{3.74}$$

and, finally, the optimal solution is found by imposing the null-gradient condition,

$$\mathbf{w}_{LS} = \Psi^{-1}(N_{samples})\boldsymbol{\pi}(N_{samples}) \tag{3.75}$$

So, the presented approach, which in fact corresponds to the application of the classical LS method to the filtering problem, leads to a solution quite similar to that of the Wiener–Hopf equations. For this very reason, it is worth emphasizing the conceptual contrast: from the standpoint of the LS approach, the solution (3.75) is exact and optimal for the available dataset. On the other hand, the solution could also be interpreted as an attempt of estimating the Wiener solution, which, in itself, belongs to the "ideal world" of the ensemble averages.

There remains the task of building an adaptive algorithm for the LS method. A promising step is to consider the possibility of updating the optimal solution on an iterative basis. In other words, when new data becomes available, the algorithm must update the optimal solution in order to take them into account, which produces an adaptive scheme in a rather natural way, as presented below.

3.5.1 The Recursive Least-Squares Algorithm

We may formulate a preliminary algorithm to calculate the matrix (3.72) and the vector (3.73) at each time instant using all available data up to the present sample and providing the optimal parameter vector via (3.75). This solution is conceptually satisfactory, but it gives us the clear impression of being significantly demanding from a computational standpoint, particularly under the perspective of dealing with huge amounts of data. A more pleasant situation would be reached if the calculations were made in accordance with an iterative “Kalman-like” spirit. Interestingly, this is not particularly difficult if we rewrite (3.72) as

$$\Psi(n) = \sum_{k=1}^{n} \lambda(k)\mathbf{x}(k)\mathbf{x}^T(k) = \lambda(n)\mathbf{x}(n)\mathbf{x}^T(n) + \sum_{k=1}^{n-1} \lambda(k)\mathbf{x}(k)\mathbf{x}^T(k) \tag{3.76}$$

and (3.73) as

$$\boldsymbol{\pi}(n) = \sum_{k=1}^{n} \lambda(k)\mathbf{x}(k)d(k) = \lambda(n)\mathbf{x}(n)d(n) + \sum_{k=1}^{n-1} \lambda(k)\mathbf{x}(k)d(k) \tag{3.77}$$

If we replace $\lambda(k)$ by the forgetting factor, as posed in (3.68), it follows that

$$\begin{aligned}\Psi(n) &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{x}(k)\mathbf{x}^T(k) = \mathbf{x}(n)\mathbf{x}^T(n) + \lambda \sum_{k=1}^{n-1} \lambda^{n-k-1}\mathbf{x}(k)\mathbf{x}^T(k) \\ &= \mathbf{x}(n)\mathbf{x}^T(n) + \lambda\Psi(n-1)\end{aligned} \tag{3.78}$$

and

$$\begin{aligned}\boldsymbol{\pi}(n) &= \sum_{k=1}^{n} \lambda^{n-k}\mathbf{x}(k)d(k) = \mathbf{x}(n)d(n) + \lambda \sum_{k=1}^{n-1} \lambda^{n-k-1}\mathbf{x}(k)d(k) \\ &= \mathbf{x}(n)d(n) + \lambda\boldsymbol{\pi}(n-1)\end{aligned} \tag{3.79}$$

From now on, as shown in the last equations, the time indices are incorporated into Ψ and Π to emphasize their characteristic of temporal estimations that depend on the quantity of available data.

From (3.78), the temporal autocorrelation matrix is obtained recursively, but it must be inverted to provide the optimal solution. In order to avoid direct matrix inversion, it is suitable to make use of an elegant mathematical result known as matrix inversion lemma [128]. Using this lemma, it is possible to update the inverse of Ψ directly by

$$\Psi^{-1}(n) = \lambda^{-1}\Psi^{-1}(n-1) - \frac{\lambda^{-2}\Psi^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\Psi^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n)} \tag{3.80}$$

Such iterative calculation, together with some definitions of auxiliary variables, leads to the RLS algorithm, depicted in Algorithm 3.1 [139]:

Algorithm 3.1: RLS Algorithm

1. Initialize $\mathbf{w}(0)$ and $\Psi(0)$. Typically, the initial value of the matrix $\mathbf{P}(0)$, which corresponds to the inverse of Ψ, is taken to be a diagonal matrix with small values.
2. For each new received sample, update the weights according to the following equations:

$$\gamma(n) = \frac{\lambda^{-1}\Psi^{-1}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n)} \tag{3.81}$$

$$\xi(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n) \tag{3.82}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \gamma(n)\xi(n) \tag{3.83}$$

$$\mathbf{P}(n) = \lambda^{-1}\Psi^{-1}(n-1) - \lambda^{-1}\gamma(n)\mathbf{x}^T(n)\Psi^{-1}(n-1) \tag{3.84}$$

Although a thorough convergence analysis of the RLS transcends the scope of this book, it is worth mentioning at least two characteristics of this technique in comparison with the LMS. First, the rate of convergence of the RLS does not depend on the eigenvalue spread of the correlation matrix, so that it converges faster than LMS techniques with stationary correlated input signals. Also, if we consider a stationary environment and assume $\lambda = 1$, the idea of misadjustment is not so strongly applicable in the case of the RLS, as the excess MSE tends to zero when the number of available samples tends to infinity [139]. This is, to a certain extent, expected, as we are working with temporal averages that will tend to approximate the expected values in the exposed limit.

3.6 A Few Remarks Concerning Structural Extensions

So far, our discussion was carried out under a well-defined structural framework: that associated with a linear combiner, whereof an FIR filter is a particular case. However, the idea of employing an MSE cost function is not fundamentally related to a particular filtering structure. In principle, this approach could be used in the context of a generic input–output mapping with free parameters. Although a complete analysis of this subject is beyond the scope of this chapter, two possible structural extensions with respect to the linear combiner deserve to be briefly discussed: the use of an infinite impulse response (IIR) filter and of a nonlinear structure.

3.6.1 Infinite Impulse Response Filters

The use of IIR filters in signal processing leads to a very general linear formulation. In particular, the use of IIR filters can lead to models more parsimonious in terms of number of free parameters than those obtained employing FIR filtering. In addition to that, it is possible to consider the idea of adapting the free parameters of a generic IIR filter of the form

$$\sum_{k=0}^{K_a} a_k y(n-k) = \sum_{p=0}^{K_b} b_k x(n-p) \tag{3.85}$$

via the Wiener paradigm, i.e., choosing the values of a_i, $i = 1, \ldots, K_a$ and b_j, $j = 1, \ldots, K_b$ that minimize the MSE cost function.

The first step of this procedure consists in calculating the gradient of the MSE with respect to the free parameters of the filter. The main difficulty lies in the need for differentiating the filter output with respect to its parameters, because the existence of feedback imposes a clear dependence with respect to past values of the output. In an adaptive process, such values will be dependent on past values of the parameters [274].

This fact can be dealt with by resorting to certain approximations. However, even with these approximations and the concrete perspective of obtaining an estimate of the gradient vector, there are two additional problems that must be dealt with and that are generally absent from the FIR framework: the existence of local minima in the MSE cost function and the possibility of unstable behavior during the adaptation process [275]. These difficulties open the possibility that more powerful search methods be considered whenever one deals with IIR adaptive filtering [20].

A complete and rigorous work on adaptive IIR filters can be found in [249]. Now, it is suitable to extend our discussion beyond the context of linear structures, since it will be useful in the sequel of the book.

3.6.2 Nonlinear Filters

Typically, an option for a nonlinear structure is justifiable in solving problems that require more flexible mappings than those engendered by linear devices. In other words, the choice for a nonlinear structure ideally tends to allow a more satisfactory level of performance, but a crucial question is the optimization of its parameters.

Whenever a structure is characterized by a nonlinear dependence of its output with respect to the free parameters, it is no longer licit to expect that the MSE cost function present a single minimum. In fact, the existence of local minima and other equilibrium points like saddle points becomes the rule. This is the case, for instance, in some neural-network-based approaches, in the context of which it might be desirable to employ techniques with a significant global search potential, e.g., genetic algorithms, instead of those based on the derivatives of the MSE surface.

This more complicate scenario is, to a certain extent, avoided whenever one chooses a nonlinear structure that is linear with respect to its free parameters. In such case, it is possible to employ the classical supervised tools and results discussed in this chapter in a quite direct manner. On the other hand, to build a device that is both linear with respect to the parameters and efficient may require some structural choices that are not always simple to deal with. A suitable model to be used in this context is, for instance, a Volterra filter [201].

The problem of nonlinear filtering is discussed in more detail in Chapter 7, in which the above mentioned topics will be revisited. For now, such brief comments on the theme are useful to illustrate how the straightforward idea of MSE optimization becomes intricate with the introduction of nonlinear devices.

3.7 Linear Filtering without a Reference Signal

Throughout this chapter, the linear filtering problem has been established in a supervised context, which is characterized by the presence of a desired or reference signal $d(n)$ that guides the process of optimization and/or adaptation of the system at hand. The existence of a reference signal, together with an appropriate criterion, leads to a linear solution and to a convex cost function.

However, the explicit use of reference signal is not necessarily easy, or even feasible in some practical problems. Such problems can be separated in two very distinct classes:

- The reference signal is indeed desired but unavailable, so that we must obtain a certain amount of a priori knowledge about its nature as well as its statistic properties. This is the essence of unsupervised or blind processing, which normally leads to nonlinear optimization problems and multimodal cost functions. This book deals with this scenario from the next chapter on.
- The reference signal is not desired per se or even unnecessary for a given task. So, it can be purposely replaced by a set of constraints on the filter coefficients that make possible the optimization and/or adaptation process. Such procedure can be understood as a kind of "missing link" between the "separated worlds" of supervised and unsupervised filtering.

3.7.1 Constrained Optimal Filters

A typical problem in which a reference signal is replaced by a set of suitable constraints is the so-called linearly constrained minimum variance (LCMV) filter, in which the minimization process is not carried out with respect to an error signal, but directly over the output signal. In order to present the problem, let us consider again the output of a linear combiner, given by

$$y(n) = \mathbf{w}^H\mathbf{x}(n) \tag{3.86}$$

Notice that, in this case, we are assuming that all signals are complex-valued.

Clearly, the direct minimization of $E\left[|y(n)|^2\right]$ leads to the trivial solution $\mathbf{w} = 0$. However, a set of nontrivial coefficients can be obtained by the following procedure:

$$\text{Minimize } E\left[|y(n)|^2\right] = E\left[\mathbf{w}^H\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}\right] = \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{3.87}$$

subject to

$$\mathbf{C}^H\mathbf{w} = \mathbf{g} \tag{3.88}$$

The resulting optimization problem corresponds to the minimization of a quadratic form given a set of linear constraints, which can be performed in accordance with the method of *Lagrange multipliers*. The use of Lagrange multipliers requires the minimization of the following expression [139]:

$$\frac{1}{2}\mathbf{w}^H\mathbf{R}\mathbf{w} + \boldsymbol{\lambda}^H\left(\mathbf{C}^H\mathbf{w} - \mathbf{g}\right) \tag{3.89}$$

In order to find all the relevant parameters, we must set to zero the gradient of (3.89), which leads to

$$\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} = \mathbf{0} \tag{3.90}$$

Then, from (3.89), the Lagrange multipliers are

$$\boldsymbol{\lambda} = -\left(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{g} \tag{3.91}$$

and the optimal coefficients are given by

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{g} \tag{3.92}$$

Hence, we reach a closed-form solution, which depends on the autocorrelation matrix, as well as in (3.14), but also on the parameters defining the

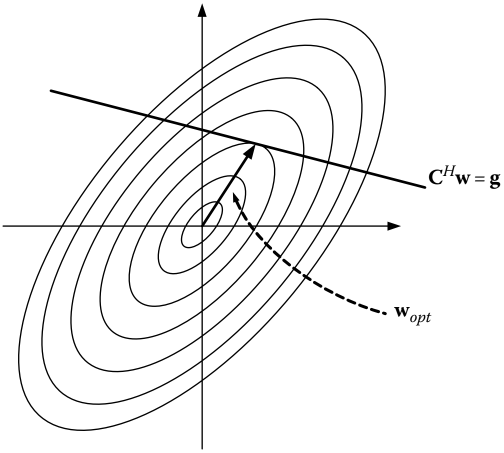


FIGURE 3.13
Illustration of the solution to the LCMV filtering problem.

constraints. In fact, it can be observed that (3.88) defines a hyperplane that intercepts the elliptical paraboloid defined by Equation 3.87, so that the intersection establishes the optimal point in (3.92), as illustrated by Figure 3.13. Having the problem been stated and solved, it is useful to discuss a relevant scenario of application.

Example 3.5 (Minimum Variance Distortionless Response Beamformer)

Let the linear combiner of Equation 3.86 correspond to the model of an antenna array, i.e., a set of antennas disposed in accordance with some geometric pattern, for instance a uniform linear array (ULA), as illustrated in Figure 3.14. Each antenna is supposed to be omnidirectional and is endowed with a complex gain that, assuming a baseband model, is able to modify the phase and the amplitude

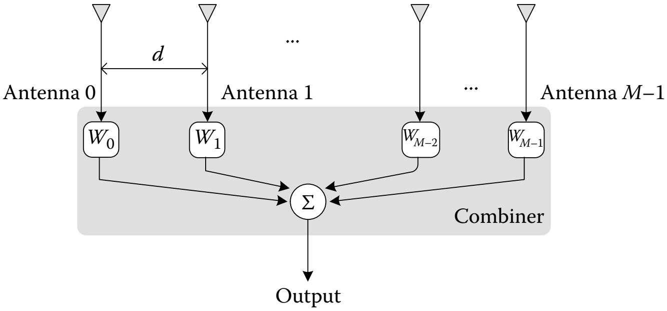


FIGURE 3.14
Structure of a ULA.

of the captured signal. When a plane wave impinges on the ULA, the structure performs spatial sampling and filtering of the incident signal.

The objective of a minimum variance distortionless response (MVDR) beamformer is twofold: to preserve a given target signal and to minimize the interference of other sources. The desired signal is supposed to propagate according to a known direction of arrival (DOA), while interference signals possess different incidence angles. Hence, the MVDR beamformer provides a selective filtering in the DOA domain, as well as temporal filters provide selectivity in the frequency domain. The response, or radiation pattern, of the array is given as a function of the electric angle, defined by

$$\theta_i = \frac{2\pi d}{\lambda}\phi_i \tag{3.93}$$

where

d is the spacing between adjacent elements of the array
λ is the wavelength of the incident signal
ϕ_i is the direction of arrival, which lies within the range $(-\pi/2, \pi/2]$

Under these conditions, let us suppose a given signal of interest with electric angle θ_o, being all other signals considered to be interferers. The received vector due to the transmitted signal $s_o(n)$ is given by $\mathbf{a}(\theta_o)s_o(n)$, where $\mathbf{a}(\theta_o)$ is the so-called steering vector, defined by

$$\mathbf{a}(\theta_o) = \left[1, e^{-j\theta_o}, e^{-j2\theta_o}, \ldots\right]^T \tag{3.94}$$

Thus, considering all signals and corresponding directions of arrivals, the received vector can be expressed as

$$\mathbf{x}(n) = \mathbf{A}s(n) \tag{3.95}$$

where $\mathbf{A}$ is a matrix of the form

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\theta_1} & e^{-j\theta_2} & \cdots & e^{-j\theta_N} \\ e^{-j2\theta_1} & e^{-j2\theta_2} & \cdots & e^{-j2\theta_N} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{3.96}$$

Since the aim is to eliminate all impinging signals except the desired signal, the ULA will work as a notch filter in the electric angle domain. With this purpose, we can formulate the problem of finding the coefficients of the array as

$$\text{Minimize } E\left[|y(n)|^2\right] = E\left[\mathbf{w}^H\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}\right] = \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{3.97}$$

subject to

$$\mathbf{w}^H\mathbf{a}(\theta_o) = 1 \tag{3.98}$$

So that the optimal parameters are

$$\mathbf{w}_{opt} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta_o)}{\mathbf{a}^H(\theta_o)\mathbf{R}^{-1}\mathbf{a}(\theta_o)} \tag{3.99}$$

The approach can be extended to several desired signals employing a matrix of constraints, in which each column is associated to a given DOA. Similar procedures can be employed in different applications, as posing constraints in the frequency domain for temporal filters, and even in using constraints to incorporate particular filtering behaviors [250]. Finally, adaptive algorithms can also be derived to find the optimal solution in (3.99), as shown in the sequel.

3.7.2 Constrained Adaptive Filters

As in the problem of Wiener filtering, for several cases it is important to derive adaptive procedures for finding the LCMV solution. This can be accomplished in lines very similar to those whereby we reached the LMS and RLS algorithms. As a matter of fact, the two algorithms we shall briefly present are counterparts to those classical adaptive techniques.

A pioneer result was presented in 1972, when Frost proposed the LMS-based algorithm for LCMV filtering [114], summarized in Algorithm 3.2.

Algorithm 3.2: LMS-Frost Algorithm

1. Initialize $\mathbf{w}(0)$
2. Update the weights according to the following equations:

$$\mathbf{w}(n+1) = \mathbf{P}\left[\mathbf{w}(n) - \mu\mathbf{x}(n)y^*(n)\right] + \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{g} \qquad (3.100)$$

where

$$\mathbf{P} = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H \qquad (3.101)$$

It can be noted that the term in brackets in (3.100) is analogous to the unconstrained LMS algorithm. The role of the matrix **P** and vector **q** consists in keeping the updated coefficients in the hyperplane defined by the constraints.

Griffiths and Jim proposed an alternative and important approach in 1982, the so-called general sidelobe canceller (GSC) [131]. In contrast with the LCMV approach, the GSC converts the constrained problem into an unconstrained one. A closed-form solution can also be obtained and either the LMS or the RLS algorithms can implement the technique in an adaptive context, since we return to unconstrained optimization.

Finally, in 1996, Resende et al. proposed an RLS-based algorithm for LCMV filtering [251]. As in the Frost approach, their technique deals directly with constrained optimization and provides faster convergence with an increase of the computational cost, which depends on the number of constraints. Robustness to round-off errors is also discussed in the mentioned reference, as this is a well-known concern of RLS techniques.

The works by Frost, Griffiths, and Resende provide important results to the theory of LCMV adaptive filtering, and a number of interesting proposals and applications in array processing and communications can be derived from these approaches [84]. Moreover, constraints may be introduced in order to provide some specific behavior for the filter response. For instance, in [250], the method is used to derive a linear phase adaptive filter. Furthermore, the PEF introduced in Section 3.1.3 can be obtained from a constrained optimization procedure, so that the PEF can be understood in the context of linear filtering without reference signal, as shown in the next section.

3.8 Linear Prediction Revisited

The parameters of a linear predictor, i.e., the prediction coefficients, can be derived from the MMSE criterion by minimizing mean-square prediction error $E\left[e(n)^2\right]$, where $e(n)$ is defined in (3.3). The calculation of the optimal prediction coefficients is straightforward and, by comparing (3.3) with (3.5), we observe that the linear prediction appears as a particular case of Wiener filtering where the desired signal is the input $x(n)$ itself

$$d(n) = x(n) \tag{3.102}$$

while the input vector is now composed of the past samples of the input signal

$$\mathbf{x}(n-1) = [x(n-1), x(n-2), \ldots, x(n-K)]^T \tag{3.103}$$

The MMSE procedure leads to the optimal Wiener solution as given in (3.14). However the cross-correlation vector between the desired and the input signals is now composed by the input autocorrelation elements:

$$\mathbf{p} = [r(1), r(2), \ldots, r(K)]^T \tag{3.104}$$

where $r(k) = E\left[x(n)x(n-k)\right]$.

Clearly, since we assume that all signals are stationary, the autocorrelation matrix remains unchanged with respect to the temporal index of the input vector. Hence, the vector of the optimal prediction coefficients is given by

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \tag{3.105}$$

With this solution at hand, it is suitable to discuss two interesting properties mentioned in the beginning of this chapter.

3.8.1 The Linear Prediction-Error Filter as a Whitening Filter

As previously discussed, the prediction error $e(n)$ depends on the present and past values of the input signal $x(n)$. Assuming that we are using an infinite number of past samples to estimate the current sample, we have

$$e_f(n) = x(n) - \sum_{k=1}^{\infty} a_k^* \, x(n-k) \tag{3.106}$$

where the prediction coefficients are obtained by minimizing the MSE cost function defined in (3.6). Therefore, the following relationship holds

$$\frac{\partial \, \mathrm{E}\left[e_f(n)^2\right]}{\partial a_k} = 0 \tag{3.107}$$

which can be expressed as

$$2\,\mathrm{E}\left[e_f(n)\frac{\partial e_f(n)}{\partial a_k}\right] = 0 \tag{3.108}$$

Differentiating (3.106) and substituting in (3.108) yields

$$\mathrm{E}\left[e_f(n)\, x(n-k)\right] = 0, \quad k \geq 1 \tag{3.109}$$

This last expression shows that the PEF produces an output sample that is orthogonal to all past samples of $x(n)$. On the other hand, according to (3.106), any past output sample, $e_f(n-k)$, also represents a linear combination of $x(n-k)$ and all its past samples. Thus, $e_f(n-k)$ is also orthogonal to $e_f(n)$, i.e.,

$$\mathrm{E}\left[e_f(n)\, e_f(n-k)\right] = 0, \quad k \geq 1 \tag{3.110}$$

Equation 3.110 reveals that a PEF with a sufficiently large number of coefficients will produce uncorrelated output samples, hence acting as a *whitening filter* [32].

3.8.2 The Linear Prediction-Error Filter Minimum Phase Property

Let the transfer function of a PEF be given by

$$A_f(z) = \sum_{k=0}^{K} a_{K,k}^{*} z^{-k} \tag{3.111}$$

Let also z_i, $i = 1, \ldots, K$ denote the zeros of $A_f(z)$. Then, we can rewrite (3.111) as

$$A_f(z) = B(z)(1 - z_b\, z^{-1}) \tag{3.112}$$

where z_b is a zero of $a_f(z)$, and

$$B(z) = \prod_{\substack{i=1 \\ i \neq b}}^{K} 1 - z_i z^{-1} \tag{3.113}$$

Thus, the MSE can be rewritten in the following form:

$$\begin{aligned} \mathrm{E}\left[\left|e_f(n)\right|^2\right] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_f(e^{j\omega}) d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\omega}) \left|A_f(e^{j\omega})\right|^2 d\omega \end{aligned} \tag{3.114}$$

where

$S_f(e^{j\omega})$ and $S_x(e^{j\omega})$ denote the power spectral densities of the prediction error and the input signal, respectively

$A_f(e^{j\omega})$ represents the frequency response of the PEF

Substituting (3.112) in (3.114), and expressing z_b in terms of its absolute value ρ_b and phase ω_b, it comes

$$\begin{aligned} \mathrm{E}\left[\left|e_f(n)\right|^2\right] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\omega}) \left|B(e^{j\omega})\right|^2 \left|1 - z_b e^{-j\omega}\right|^2 d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\omega}) \left|B(e^{j\omega})\right|^2 \left[1 - 2\rho_b \cos(\omega - \omega_b) + \rho_b^2\right] d\omega \end{aligned} \tag{3.115}$$

Now, assuming that all parameters of $b(z)$ are already optimized according to the Wiener criterion, then

$$\frac{\partial\, \mathrm{E}\left[\left|e_f[n]\right|^2\right]}{\partial \rho_b} = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(e^{j\omega})\left|Bs(e^{j\omega})\right|^2 [-2\cos(\omega-\omega_b)+2\rho_b]\, d\omega = 0 \tag{3.116}$$

Since both $S_x(e^{j\omega})$ and $|\,b(e^{j\omega})|^2$ will always be nonnegative, then $\rho_b - \cos(\omega-\omega_b)$ must, necessarily, assume positive and negative values in order that (3.116) be valid. Therefore, since $|\cos(\omega-\omega_b)| \le 1$ for any ω, ρ_b should be less than 1, which means that all zeros z_i $(i = 1, \ldots, L)$ should necessarily be located inside the unit circle in the complex z-plane. In conclusion, the FEP is a minimum-phase filter.

3.8.3 The Linear Prediction-Error Filter as a Constrained Filter

As previously mentioned, the PEF provides a direct mapping between the input signal and the error signal. Such mapping can be implemented by an FIR filter

$$\begin{aligned} e(n) &= x(n) - \sum_{k=1}^{K} w_{f,k}x(n-k) \\ &= \mathbf{w}_f^T\mathbf{x}(n) \end{aligned} \tag{3.117}$$

where

$$\mathbf{w}_f = [1, -w_1, -w_2, \ldots, -w_K]^T \tag{3.118}$$

By doing so, the search for the optimal PEF can be carried out in an LCMV context, in accordance with (3.87) and (3.88). Let us define

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \tag{3.119}$$

and

$$\mathbf{g} = 1 \tag{3.120}$$

Then, from (3.92), the following solution is reached

$$\mathbf{w}_f = \frac{\mathbf{R}^{-1}\mathbf{C}}{\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}} \tag{3.121}$$

i.e., the prediction coefficients are given by (3.105). This is an alternative way to build the PEF and highlights that *the PEF design* can be seen as a problem of linear filtering without a reference signal.

The above results open some perspectives that deserve to be further exploited: in fact, in the problem of channel equalization, it is usual to assume that the transmitted signal is a sequence of independent and identically distributed (i.i.d.) random variables. Therefore, the transmitted signal is uncorrelated, which means that the equalizer works as a whitening filter. This requirement naturally leads to the idea of considering the use of an FEP in the equalization process, when a training sequence is not available.

Equalization without training sequence, i.e., *unsupervised* or *blind equalization* is a central subject of this book, which will be studied from now on.

3.9 Concluding Remarks

In this chapter, we presented the fundamental concepts of optimal and adaptive filtering, which constitute the theoretical basis of the studies on unsupervised signal processing.

After a brief presentation on the principles and motivations of supervised filtering, we exposed the classical Wiener theory in the context of the search of optimal parameters of an FIR linear filter and derived the Wiener–Hopf equations. As an illustration, we applied the results in some representative examples, like system identification and channel equalization.

Instead of using the Wiener–Hopf equations, the optimal parameters may also be obtained via iterative procedures. We developed the steepest-descent technique, a well-established way to reach the domain of truly adaptive algorithms. Based on the Robbin–Monro principle of stochastic approximation, we derived the LMS algorithm, and briefly discussed some convergence issues of this celebrated technique, including some recent results provided in [136].

Afterward, we considered another family of optimal and adaptive filtering methods, based on the LS criterion, in which we work with a given set of available data and not with statistic averages. We derived the normal equations that provide the optimal LS solution for the filter parameters. By using an appropriate update of these averages as new data is available, we derived the recursive LS algorithm.

Although the focus of this chapter is on linear FIR filters, we provided a brief discussion about alternative structures, including IIR and nonlinear filters. A more in-depth presentation on nonlinear filtering will be given in Chapter 7.

After the discussion about alternative structure, we turned our attention to the problem of optimal and adaptive filtering without a reference signal, since it is the very essence of unsupervised signal processing. In the context of this chapter, we considered the case in which the reference signal is replaced by a set of linear constraints on the filter coefficients. In this case, the optimization process still leads to a linear solution. This is why we considered such problem as a link between supervised filtering and the "truly blind" techniques, where the original information must be recovered from the observed signal.

Finally, we closed the chapter by revisiting linear prediction theory, deriving the optimal solution for the prediction coefficients in the MMSE context. We have shown some important properties of the error prediction filter, not only for the sake of completeness, but also to establish some relationships between prediction and equalization that will be useful in the sequel.

It is worth emphasizing that many issues of this chapter are discussed in a number of important works. However, it is important to put together some results in order to establish a solid foundation and suitable links before starting to consider the problem of unsupervised filtering.

4

Unsupervised Channel Equalization

As pointed out in Chapter 3, supervised signal processing is characterized by the presence of a desired or reference signal that guides the process of parameter optimization and/or adaptation. The reference signal can also be replaced by a set of constraints, as shown in the discussion on LCMV and prediction-error filters (PEF). Even so, the problem can be solved within a framework of linear optimal filtering.

A different scenario occurs when the process of parameter optimization and/or adaptation cannot be guided because a reference signal is by no means available. This situation establishes the problem of unsupervised filtering, the solution of which is not so directly attained by the linear methods previously discussed.

Unsupervised signal processing has been an exciting theme of research for at least three decades. It finds potential applications in practically all fields where more classical techniques of digital signal processing have been employed: telecommunications, speech and audio processing, radar and sonar, biomedical signals and images, geophysics, etc. In such fields, unsupervised methods can be required for channel identification and equalization, source separation, image deconvolution, data clustering, and others.

This chapter presents the foundations of unsupervised filtering methods and focuses on the problem of channel equalization, which is an unsupervised problem by nature, as we have already mentioned. In addition to that, such a problem is characterized by the requirement of real-time and low computational burden, due to the practical operation conditions of a communication system. For this reason, we are particularly oriented to unsupervised adaptive techniques, although batch methods provide interesting results in a number of off-line applications [70,137,138]. In order to properly present the fundamental principles of unsupervised signal processing and expose the main techniques to be used in channel equalization, this chapter is organized as follows:

- Section 4.1 establishes the general problem of *unsupervised deconvolution*, whereof equalization is a particular case. This section poses some important conceptual statements in order to pave the way for the subsequent theoretical results.
- Section 4.2 presents two main results: the *Benveniste–Goursat–Ruget* and the *Shalvi–Weinstein* (SW) *theorems*. Such theorems establish

theoretical conditions for blind equalizations, over which effective methods and algorithms may be implemented.

- Once the theoretical foundation is established, we turn our attention to adaptive techniques in Section 4.3. We discuss the so-called *Bussgang algorithms*, which play a central role in this chapter, devoted to SISO channels. Among the great number of proposals found in the literature, we focus on a representative group: the *decision-directed* (DD), *Sato*, and *Godard algorithms*.
- Section 4.4 presents the *Shalvi–Weinstein algorithm* (SWA), which differs from the previous ones in its underlying mathematical basis, since it is derived from the corresponding SW theorem. The section includes both constrained and unconstrained versions of the SWA.
- Section 4.5 presents the *super-exponential algorithm* (SEA) which is also derived from SW framework, with the aim of accelerating the convergence of the techniques presented previously.
- In Section 4.6, we turn our attention to the study of *equilibrium solutions* of blind equalization criteria. Due to their practical relevance, we give specific attention to the DD and the constant modulus (CM) criteria.
- Finally, in Section 4.7, we discuss the *relationships between the CM and the SW criteria* and consider the relationships between these criteria and supervised approaches.

Historical Notes

The application of adaptive solutions in digital communications and, specifically, in channel equalization, dates back to the 1960s, with the work of Lucky [190,191], who is referred to as the inventor of the adaptive equalizer.

Lucky was the first to propose the so-called zero-forcing (ZF) method to be applied in FIR equalization. The ZF criterion is an ingenious approach that is rather intuitive for communications engineers, since it aims to minimize (force to zero) the intersymbol interference (ISI) caused by the dispersive effect of the channel. The ZF algorithm proposed by Lucky is an adaptive procedure to adjust the coefficients of the FIR equalizer, so that the IIS be set to zero. In a noiseless situation, the optimal ZF equalizer tends to be the inverse of the channel.

In a second proposal, Lucky extended his approach to the tracking mode of operation and introduced the so-called decision-directed equalizer. As discussed later in this chapter, the DD method is an unsupervised but not robust strategy, since its effectiveness depends on the initial condition of the equalizer coefficients.

The first application of a robust unsupervised strategy is credited to Sato in 1975 [260]. Sato proposed an adaptive equalizer to work with pulse amplitude modulation (PAM) signals. The approach gave rise to a number of interesting algorithms, particularly throughout the 1980s, many of which have been derived from an intuitive starting point.

A first theoretical landmark was the work of Benveniste et al. [41] in 1980, which stated fundamental conditions for blind deconvolution. Moreover, the authors proposed a class of unsupervised algorithms, which encompasses the Sato algorithm, and studied their convergence properties. Complexity was an issue, as the method required the equalization of probability density functions (pdfs) or, equivalently, of all the infinite higher-order statistics of the involved signals.

Also in 1980, Dominique Godard proposed a new class of cost functions to be applied to complex signals, such as quadrature amplitude modulation (QAM) signals [118]. Later, in 1983, Treichler and Agee exploited the structural properties of the transmitted signal to design a cost function. In particular, the idea of restoring the CM properties of some modulations was used in the constant modulus algorithm (CMA) [292], probably the most investigated unsupervised adaptive algorithm for blind equalization. Interestingly, the CMA is identical to one of the members of the class of algorithms proposed by Godard. That is why, in general, credit is given to both works for the formulation of the approach.

The CM, Sato, and other algorithms were shown to belong to the class of the so-called Bussgang algorithms, introduced by Godfrey and Rocca [125] and Bellini and Rocca [36]. In [35], Bellini provides an interesting survey of Bussgang methods. The term "blind equalization" seems to have been first introduced by Benveniste and Goursat in a paper that appeared in 1984 [40], in which the authors proposed an update procedure composed by a combination between both the DD and Sato algorithms. In 1987, Picchi and Prati proposed the "stop-and-go" algorithm [239], which, again, combined the Sato and DD strategies to reach a procedure that continues or stops the adaptation process, depending on a reliability criterion.

A second theoretical landmark occurred in 1990 when Shalvi and Weinstein [269] significantly simplified the conditions for blind deconvolution as previously stated by Benveniste et al. Before their work, the general belief was that infinite statistics were required to guarantee ZF equalization. Shalvi and Weinstein showed that ZF equalization can be achieved if only two statistics of the involved signals are equalized. Actually, they proved that, if the fourth-order cumulant (kurtosis) is maximized and the second-order cumulant remains the same, then the recovered signal would be a scaled and rotated version of the transmitted signal. Later, they also proved that other higher-order statistics could be used to ensure perfect equalization [271]. This result was very important to provide theoretical support to the proposition of blind equalization criteria and algorithms with low complexity burden. In 1993, the same authors proposed a cumulant-based algorithm called super-exponential algorithm [270].

Many other important works and approaches have been presented in the vast literature on the subject. A nice and helpful scan of the literature can be found in [99].

4.1 The Unsupervised Deconvolution Problem

The importance of convolution is notorious, as it models the input–output mapping of a linear time-invariant system. Deconvolution is the inverse mathematical operation that allows the recovery of the input signal from the output signal. If the system response is available, it is relatively straightforward to develop both time-domain and frequency-domain algorithms to perform deconvolution.

An early application of deconvolution has been in seismic signal processing. Robinson developed a pioneer work on the subject in his PhD thesis at MIT [253]. His research attracted the attention of Wiener and Levinson, both working at MIT at that time. In fact, Robinson's work represented the first successful application of the recently developed Wiener theory on prediction and filtering. His aim was to obtain information about the structure of the Earth by the estimation of the impulse response of a layered earth model, i.e., an FIR model.

The problem became intricate though, since an estimate of the input signal, the so-called seismic wavelet, was not available. Such lack of information characterizes the problem as unsupervised. To solve it, Robinson derived the predictive deconvolution procedure by considering two simplifying hypotheses [139]: (1) the seismic wavelet is the impulse response of an all-pole system, so that it is necessarily minimum phase; (2) the impulse response of the layered earth model behaves like a white noise, so that it has a flat spectral shape.

Indeed, if we compare the above hypotheses with the properties of the PEF given in Sections 3.8.1 and 3.8.2, we verify that the desired impulse response of the model may be recovered as a prediction-error signal when the output of the model (i.e., the measured signal that is recorded in a seismograph) is applied to the input of a PEF.

Robinson's approach can be generalized as follows. If a given signal $x(n)$ obeys a convolution relationship,

$$x(n) = s(n) * w(n) \tag{4.1}$$

where $s(n)$ is an uncorrelated (white) signal and $w(n)$ is a minimum-phase impulse response, we can recover $s(n)$ and $w(n)$ in an unsupervised way by obtaining the prediction-error signal

$$e_f(n) = x(n) - \sum_{k=1}^{K} a_k x(n-k) \tag{4.2}$$

Except for a constant factor, this signal corresponds to the recovered version of the input signal if K is large enough to provide an effective whitening of $x(n)$. The prediction coefficients a_k, which can be obtained with the Wiener procedure described in Section 3.2, lead to the identification of the minimum-phase system, the frequency response of which is given by

$$W(\exp(j2\pi f)) = \frac{\sigma_s^2}{\left|1 - \sum_{k=1}^{K} a_k^* \exp(-j2\pi fk)\right|} \tag{4.3}$$

where σ_s^2 denotes the constant power spectral density of the transmitted signal $s(n)$.

The above deconvolution method shows that Wiener theory can be used in an unsupervised context if the signal to be recovered is white and the system to be identified/inverted is minimum phase. The procedure is carried out by means of a PEF, hence the terminology predictive deconvolution. Actually, this is another way of reaching the result exposed in Section 3.7, in which the PEF is viewed as an unsupervised filter.

It is worth highlighting a theoretical aspect: as stated above, predictive deconvolution deals only with the second-order statistics of the involved signals. From Equations 2.121 and 4.1, and if $s(n)$ is white, it comes that

$$S_x(f) = \sigma_s^2 \left|W(f)\right|^2 \tag{4.4}$$

where N_S stands for the constant power spectral density of $s(n)$. Equation 4.4 makes clear that second-order statistics provide only information exclusively about the magnitude, and not about the phase components of the signal spectra and of the system response. In other words, it is not possible to carry out a complete unsupervised deconvolution and/or identification procedure using only second-order statistics, unless we have additional informations about the phase behavior. In the mentioned seismic application, predictive deconvolution works due to the minimum-phase hypothesis, which corresponds to additional information without which the method fails. This is an important conceptual point to well understand the theoretic foundations of unsupervised equalization.

4.1.1 The Specific Case of Equalization

Figure 4.1 illustrates the case of unsupervised equalization, where $s(n)$ is the transmitted signal, $h(n)$ is the channel, $\nu(n)$ is an additive noise, generally supposed to be white and Gaussian, $w(n)$ stands for the equalizer response, and $y(n)$ for its output signal. The output $y(n)$ is applied to a nonlinear decision device in order to provide the recovered signal $\hat{s}(n)$.

If compared to the general case of unsupervised deconvolution, the problem of equalization is characterized by two typical assumptions:

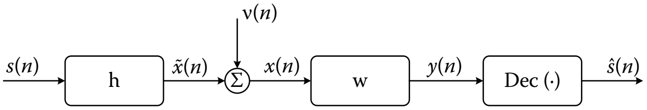


FIGURE 4.1
Unsupervised equalization scheme.

- The transmitted signal $s(n)$ belongs to a finite symbol alphabet $\mathcal{A}$.
- The transmitted signal is composed of a sequence of independent and identically distributed (i.i.d.) random variables.

The first assumption comes directly from the nature of the application, since we necessarily employ some kind of digital modulation scheme. In practice, $y(n)$ does not belong to $\mathcal{A}$, which explains the need for a nonlinear device to map $y(n)$ to $\hat{s} \in \mathcal{A}$.

In a way, the aim of the equalizer is to provide a signal $y(n)$ whose values are close to the symbol levels associated with $\mathcal{A}$. The more the equalizer compensates the channel distortions, the more $y(n)$ is concentrated around the symbol levels. A classical and ingenious technique to evaluate this performance in digital communication is the *eye pattern* [28,245]. Figure 4.2 shows, for a binary transmission scheme, the eye pattern before and after an optimized equalizer. The name "eye pattern" comes from the peculiar shape of the diagram, and the two illustrated conditions are referred to as *closed-eye* and *open-eye* conditions, respectively. In supervised equalization, as mentioned in Section 3.1.2, the role of the training period is to allow an open-eye condition to be reached in order that the system may safely work in the information transmission mode. In this mode, the equalizer is optimized using an unsupervised technique, the aim of which is to preserve the open-eye condition. In the absence of a training procedure, the unsupervised equalizer must be optimized from an initial closed-eye condition.

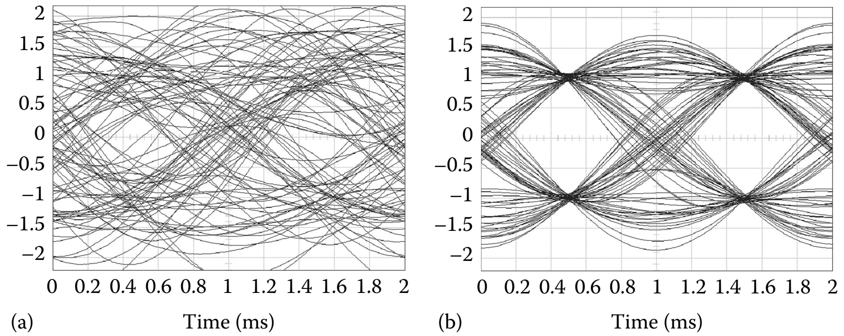


FIGURE 4.2
Eye pattern before and after an optimized equalizer: (a) closed-eye condition and (b) open-eye condition.

The second of the above assumptions is usual in digital communication, where scrambling procedures are normally carried out at the transmitters. As a consequence of this hypothesis, the spectrum of $s(n)$ is considered to be flat (white), as statistical independence implies decorrelation. However, the converse is not true, which means that the use of a whitening filter as an equalizer is not sufficient to guarantee the recovery of the transmitted i.i.d. sequence. Nevertheless, if the channel is minimum phase, the PEF, which works as a whitening filter, is capable of providing a scaled version of $s(n)$, in a procedure similar to that of predictive deconvolution.

In fact, in the absence of a priori knowledge about the phase-response behavior of the channel, the PEF or any other whitening filtering can only ensure magnitude equalization. The phase distortion remains to be compensated, which is classically carried out using an all-pass structure. Predictive equalization can be developed based on this principle by a nonlinear configuration that combines the use of a PEF with a blind phase equalizer. Such an approach, originally proposed by da Rocha et al. [49, 81, 195], is further discussed in Chapter 7.

The above remarks clearly reveal the limitations of the predictive deconvolution principle in the blind equalization problem, which is important for well understanding the sequence of results in this chapter. Before addressing them, let us summarize the key points of the discussion so far:

- A whitening procedure deals only with second-order statistics and can only provide magnitude equalization; it does decorrelate the received signal but does not ensure the recovery of the original i.i.d. sequence.
- If the phase response of the channel is known a priori, it is possible to perform blind equalization by dealing only with second-order statistics. In the specific case of a minimum-phase channel, the PEF is the whitening filter that works as the optimum equalizer.
- In the absence of such additional information, the recovery of an i.i.d. signal requires more than second-order whitening. As discussed in Chapter 2, the joint cumulant of independent random variables is null. This means that the concept of whitening must be, in a way, extended to encompass all higher-order statistics when we search for the optimum equalizer in an unsupervised mode.

These points can now be further investigated with the aid of two theorems that constitute the theoretical foundation of blind equalization.

4.2 Fundamental Theorems

As commented in our brief historical, Sato's algorithm was a first practical solution for blind channel equalization. Nevertheless, this proposition was not supported by a solid theoretical justification, which was only provided by the works of Benveniste–Goursat–Ruget [41] and Shalvi–Weinstein [269]. In this section, we present and discuss both results.

4.2.1 The Benveniste–Goursat–Ruget Theorem

The work in [41] first stated theoretical conditions for blind equalization. Its main result was a well-known theorem named after its authors.

THEOREM 4.1 (Benveniste–Goursat–Ruget)

Let the transmitted signal be composed of non-Gaussian i.i.d. samples and both channel and equalizer be linear time-invariant filters, in a noiseless scenario. Under these conditions, if the pdfs of the transmitted signal and of the equalizer output are equal, then the channel will have been perfectly equalized, i.e.,

$$y(n) = \pm\alpha s(n-d) \tag{4.5}$$

α is a unit magnitude complex constant. ■

This result is crucial, since it establishes the viability of obtaining an efficient equalizer with the sole aid of statistical properties and without any knowledge of the channel impulse response. The non-Gaussianity hypothesis is required, since a linearly filtered Gaussian process remains Gaussian [230] and, in such a case, pdf matching would ensure only power normalization.

4.2.2 The Shalvi–Weinstein Theorem

The BGR theorem provides a solid theoretical basis to blind equalization. However, a decade later, Shalvi and Weinstein demonstrated that the condition of equality between the pdf's of the transmitted and equalized signals was excessively stringent. Under assumptions equivalent to those exposed in the last section, they demonstrated that it is possible to perform blind equalization with the aid only of the cumulants of the involved signals, as stated in the following theorem.

THEOREM 4.2 (Shalvi–Weinstein)

Let the transmitted signal be composed of non-Gaussian i.i.d. samples and both the channel and the equalizer be linear time-invariant filters, in a noiseless scenario. Under these conditions, if $E\left\{|s(n)|^2\right\} = E\left\{|y(n)|^2\right\}$ and a nonzero cumulant of order higher than 2 of $s(n)$ and $y(n)$ are equal, then the channel will have been perfectly equalized. ■

The importance of the SW theorem is notorious as it assures that blind equalization can be accomplished by simply comparing the cumulants of the transmitted and equalized signals. In addition, to providing simplified conditions for blind equalization, the theorem gives support to the proposition of feasible algorithms, as shown further in this chapter.

4.3 Bussgang Algorithms

The Benveniste–Goursat–Ruget and SW theorems clearly state the need for dealing with the information brought by higher-order statistics of the involved signals in order to guarantee blind equalization. In this section, we concentrate our attention on algorithms that make implicit use of higher-order statistics. These algorithms, commonly referred to as Bussgang algorithms [36, 125], employ some sort of prior information about the pdf of the transmitted sequence $s(n)$ to obtain an estimate of the transmitted symbol. Such estimate plays the role of the "pilot signal" in the adaptive algorithm.

The general form of these algorithms is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \left\{\psi[y(n)] - y(n)\right\} \mathbf{x}(n) \tag{4.6}$$

where

$\mathbf{w}(n)$ is the parameter vector at instant n
$\mathbf{x}(n)$ is the equalizer input vector
μ is the step-size

This is similar to the least mean square (LMS) update rule given in (3.65), but the pilot signal is replaced by $\psi[y(n)]$, which corresponds to a memoryless nonlinear estimator for the transmitted signal $s(n)$.

In order to obtain the optimal estimator, let us consider the usual assumption that both channel and equalizer are modeled as linear and time-invariant systems, so that the equalizer output is given in terms of the combined channel + equalizer response $g(n)$, i.e.,

$$\begin{aligned} y(n) &= w(n) * x(n) \\ &= w(n) * h(n) * s(n) \\ &= g(n) * s(n) \end{aligned} \tag{4.7}$$

where $g(n) = w(n) * h(n)$. Then, the equalizer output can be rewritten as

$$y(n) = g(n) * s(n) = g(0)s(n) + \sum_{i=-\infty, i\neq 0}^{\infty} g(n-i)s(i)$$
$$= g(0)s(n) + \eta(n) \tag{4.8}$$

where $\eta(n)$ is the so-called convolutional noise [135], which is null only if the ZF condition is attained.

If the pdf of $\eta(n)$ is known beforehand, the maximum likelihood (ML) estimate $\hat{s}(n)$ (vide Section 2.5) of the transmitted symbol is given by

$$\hat{s}(n)_{\mathrm{ML}} = \psi\left[y(n)\right] = \arg\max_{s(n)} p_y\left(y(n)|s(n)\right) \tag{4.9}$$

where $p_y\left(y(n)|s(n)\right)$ is the conditional distribution of the equalizer output given the transmitted signal $s(n)$. It should be noted that this conditional distribution depends on the channel and the equalizer, which are unknown. Thus, the derivation of an ML estimator depends on additional assumptions that provide an adequate characterization of the convolutional noise distribution [35].

A first simplifying assumption is that the convolutional noise presents a Gaussian distribution, which can be justified in terms of the central limit theorem [230], if we consider that the combined response $g(n)$ is long enough. In this case, the ML estimator becomes the minimum variance estimator [135,166], given by

$$\psi\left[y(n)\right] = E\left[s(n)|y(n)\right] \tag{4.10}$$

We have that

$$\mu_\eta = E\left\{\eta(n)\right\} = 0 \tag{4.11}$$

and

$$\sigma_\eta^2 = E\left\{\eta(n)^2\right\} = E\left\{s(n)^2\right\} \sum_{i\neq 0} g(i)^2 \tag{4.12}$$

Thus, we can consider that

$$p_\eta\left(\eta(n)\right) = \frac{1}{\sqrt{2\pi}\,\sigma_\eta} \exp\left\{-\frac{\eta(n)^2}{2\sigma_\eta^2}\right\} \tag{4.13}$$

The output signal $y(n)$ is simply the sum of $g(0)s(n)$ and $\eta(n)$. Hence, its pdf equals the convolution between $p_\eta\left(\eta(n)\right)$ and the pdf of $g(0)s(n)$.

If we assume that $s(n)$ is an M-PAM signal uniformly distributed over $\{-(M-1), \ldots, -3, -1, 1, 3, \ldots, (M-1)\}$, then

$$p_y\left(y(n)\right) = \sum_{i=1}^{M} p_\eta\left[y(n) + (2i - M - 1)g(0)\right] \tag{4.14}$$

Using Bayes' rule, we get

$$p_s\left(s(n)|y(n)\right) = \frac{p_y(y(n)|s(n))p_s(s(n))}{p_y(y(n))} \tag{4.15}$$

From (4.8) and (4.13) we obtain that

$$p_y(y(n)|s(n)) = p_\eta\left[y(n) - g(0)s(n)\right] \tag{4.16}$$

and, hence

$$p_s\left(s(n)|y(n)\right) = \frac{p_\eta\left[y(n) - g(0)s(n)\right]p_s(s(n))}{\sum_{i=1}^{M} p_\eta\left[y(n) + (2i - M - 1)g(0)\right]} \tag{4.17}$$

By evaluating the expectation given in (4.10), and after some algebraic manipulations, we get the following expression for the estimator [135]:

$$\psi\left[y(n)\right] = \frac{\sum_{i=1}^{M/2} (2i-1)\exp\left[-\frac{g(0)^2(2i-1)^2}{2\sigma_\eta^2}\right]\sinh\left[\frac{g(0)(2i-1)}{\sigma_\eta^2}y(n)\right]}{\sum_{i=1}^{M/2} \exp\left[-\frac{g(0)^2(2i-1)^2}{2\sigma_\eta^2}\right]\cosh\left[\frac{g(0)(2i-1)}{\sigma_\eta^2}y(n)\right]} \tag{4.18}$$

In order to gain some insight on the role of the nonlinear estimator, let us turn our attention to the case in which the transmitted signal belongs to a 4-PAM modulation [35]. In Figure 4.3, we present some examples of (4.18) for $M = 4$ and different variances of the convolutional noise. We can note that the estimator has a "quasi-linear" region for large values of σ_η^2, which means that, under the adopted hypotheses, the details of the chosen modulation are not particularly relevant. However, when the convolutional noise variance is small, the estimator becomes rather similar to the mapping performed by the decision device [135]. In fact, the difference between the different proposed algorithms resides in the memoryless estimator $\psi[y(n)]$ employed in the adaptation. In the sequence, we describe three of them: the DD, Sato, and Godard algorithms.

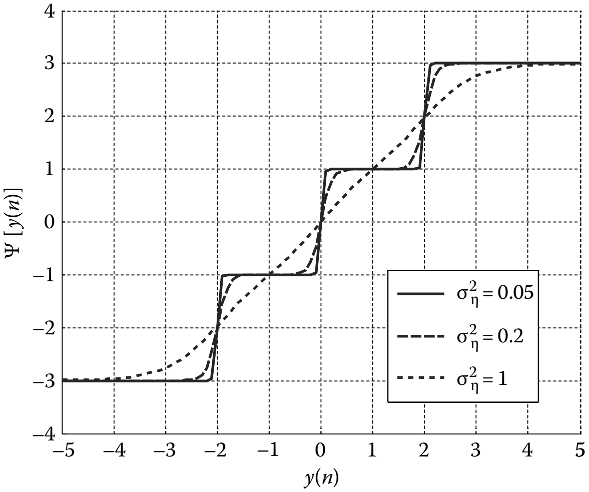


FIGURE 4.3
Bussgang estimators for a 4-PAM modulation and different convolutional noise variances.

4.3.1 The Decision-Directed Algorithm

The DD algorithm can be considered as the simplest of Bussgang techniques, since it employs the decision-device output as an estimate of the desired response, i.e.,

$$\psi_{\text{DD}}[y(n)] = dec\left[y(n)\right] \tag{4.19}$$

leading to an adaptation rule given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left\{dec\left[y(n)\right] - y(n)\right\}\mathbf{x}(n) \tag{4.20}$$

In this case, the corresponding cost function being minimized by the iterative method in (4.20) is

$$J_{\text{DD}}\left(\mathbf{w}\right) = E\left[\left(y(n) - dec\left(y(n)\right)\right)^2\right] \tag{4.21}$$

Intuitively, the technique is founded on the assumption that the output of the decision-device should be a useful estimate of the signal we wish to recover. This situation is reasonable if we begin the adaptation of the equalizer from a good initial condition, i.e., a condition capable of leading to a satisfactory open-eye condition. A classical possibility to make this possible is to employ the DD algorithm together with a supervised method. In this case, the supervised method is responsible for using an available training sequence to reduce as much as possible the level of intersymbol interference (ISI) present in the received signal, which, ideally, would give rise to an open-eye condition. After the conclusion of the supervised training period, the DD algorithm could be employed to refine the obtained equalizer parameters, further reducing the ISI level and tracking eventual channel variations.

4.3.2 The Sato Algorithm

In 1975, Yoichi Sato proposed a blind algorithm for the recovery of M-PAM multilevel signals [260] that was based on recovering the most significant bit of the modulation and to treat the remaining information as a kind of noise. In accordance with this idea, the following nonlinearity was proposed:

$$\psi_{\text{Sato}}\left[y(n)\right] = \gamma \operatorname{sign}\left[y(n)\right] \tag{4.22}$$

where

$$\gamma = \frac{E\left[s^2(n)\right]}{E\left[|s(n)|\right]} \tag{4.23}$$

is a "gain control" scaling factor. The update rule then becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left\{\gamma \operatorname{sign}\left[y(n)\right] - y(n)\right\}\mathbf{x}(n) \tag{4.24}$$

which represents an LMS-like algorithm for the minimization of the following cost function:

$$J_{\text{Sato}}(\mathbf{w}) = E\left[\left(\gamma \operatorname{sign}\left[y(n)\right] - \left|y(n)\right|\right)^2\right] \tag{4.25}$$

Sato's proposal is historically very important as a first robust technique that allows operation in a completely blind fashion.

4.3.3 The Godard Algorithm

In 1980, Dominique Godard proposed a blind equalization criterion whose main feature was its immunity to phase recovery errors [124]. The goal was to develop a method capable of reducing the distortions introduced by the channel to a level that could be handled by conventional methods like the DD algorithm.

The idea behind Godard's proposal was to establish a measure of dispersion around a predetermined value without having to resort to phase information, which can be accomplished considering the following cost function:

$$J_G^{(p,q)}(\mathbf{w}) = E\left[\left(\left|y(n)\right|^p - R_p\right)^q\right] \tag{4.26}$$

where R_p is a predetermined constant value, given by

$$R_p = \frac{E\left[|s(n)|^{2p}\right]}{E\left[|s(n)|^p\right]} \tag{4.27}$$

A particular case of great interest is that originated by the choice $p = q = 2$, generally referred to as the CM criterion, whose cost function is

$$J_{CM} = E\left[\left(\left|y(n)\right|^2 - R_2\right)^2\right] \tag{4.28}$$

where

$$R_2 = \frac{E\left[|s(n)|^4\right]}{E\left[|s(n)|^2\right]} \tag{4.29}$$

is a constant that depends on a priori statistical information about the transmitted signal. Using the standard stochastic approximation for the gradient vector, we obtain the CMA:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\upsilon^*(n)\mathbf{x}(n) \tag{4.30}$$

where

$$\upsilon(n) = y(n)\left[R_2 - |y(n)|^2\right] \tag{4.31}$$

From (4.30) and (4.6), it is possible to notice that the CMA employs the following estimator:

$$\begin{aligned}\psi_{\text{CM}}\left[y(n)\right] &= \upsilon(n) + y(n)\\ &= y(n)\left[1 + R_2 - |y(n)|^2\right]\end{aligned} \tag{4.32}$$

To summarize the explanation, we present in Table 4.1 the estimators and corresponding cost functions of the algorithms discussed in this section. It is important to mention that even though these algorithms are related by (4.6), they may exhibit very different convergence characteristics. Convergence aspects of these techniques are discussed in Section 4.6.

TABLE 4.1

Estimators and Corresponding Cost Functions Associated with the Bussgang Algorithms

Algorithm	**Estimator $\psi(y(n))$**	**Cost Function**
DD	$dec(y(n))$	$J_{DD} = E\left[\left(y(n) - dec\left(y(n)\right)\right)^2\right]$
Sato	$\gamma \, \text{sign}\left[y(n)\right]$	$J_{Sato} = E\left[(\gamma \, \text{sign}\left[y(n)\right] - \lvert y(n) \rvert)^2\right]$
CMA	$y(n)\left[1 + R_2 - \lvert y(n)\rvert^2\right]$	$J_{CM} = E\left[\left(\lvert y(n)\rvert^2 - R_2\right)^2\right]$

4.4 The Shalvi–Weinstein Algorithm

The work by Shalvi and Weinstein [269] was relevant not only as a theoretical foundation to the problem of blind equalization, but also as the source of important criteria and adaptive methods. The SWA is an implementation based on the ideas presented in Section 4.2.2, and can be derived in two different versions: constrained and unconstrained. In the sequence we develop both approaches.

4.4.1 Constrained Algorithm

The constrained algorithm can be seen as a direct interpretation of the Shalvi–Weinstein theorem. First, we should notice that if the variance of the input and output signals are the same, we should have $\sum_i |g(i)|^2 = 1$, and the following relations hold [269]:

- $\sum_i |g(i)|^4 \leq 1$
- $\sum_i |g(i)|^4 = 1$ if and only if $\mathbf{g}$ corresponds to a perfect equalization solution, i.e., $g(n) = [0, \ldots, 0, 1, 0, \ldots, 0]$

Since equality only occurs if the ZF condition is attained, the theorem naturally provides the following equalization criterion:

$$\begin{cases} \text{maximize} & J_{SW_c}(\mathbf{w}) \left|c_4\left(y(n)\right)\right| \\ \text{subject to} & c_2\left(y(n)\right) = c_2\left(s(n)\right) \end{cases} \tag{4.33}$$

Notice that the constraint is equivalent to $\sum_i |g(i)|^2 = 1$, which is a requirement of the SW theorem.

The SWA is then derived employing a stochastic approximation proposed in [41] to deal with constrained problems. In this kind of approximation, the algorithm requires spectral prewhitening of the channel output. Each iteration is composed of the following two steps [269]:

$$\tilde{\mathbf{w}}(n+1) = \tilde{\mathbf{w}}(n) + \mu \nabla_{\tilde{\mathbf{w}}} J_{SW_c}(\mathbf{w}) \tag{4.34}$$

and

$$\tilde{\mathbf{w}}(n+1) = \frac{\mathbf{w}(n+1)}{\|\mathbf{w}(n+1)\|} \tag{4.35}$$

where
μ is the step size
$\nabla_{\mathbf{w}} J$ is the gradient of the criterion

The normalization procedure is required to ensure that the power constraint is respected.

In order to obtain $\nabla_{\mathbf{w}} J_{SW_c}(\mathbf{w})$, let us first rewrite the criterion as [269]

$$\begin{aligned} J_{SW_c} &= |c_4(y)| = \operatorname{sgn}\left[c_4\left(y(n)\right)\right] c_4\left(y(n)\right) \\ &= \operatorname{sgn}\left[c_4\left(s(n)\right)\right]\left[E\left\{|y(n)|^4\right\} - 2E^2\left\{|y(n)|^2\right\} - \left|E\left\{y(n)^2\right\}\right|^2\right] \end{aligned} \tag{4.36}$$

where we have assumed that $\operatorname{sgn}\left[c_4\left(y(n)\right)\right] = \operatorname{sgn}\left[c_4\left(s(n)\right)\right]$. Since we assume that the power constraint is respected, the term involving $E\left\{|y(n)|^2\right\}$ is constant and can be neglected. Moreover, if we consider that both the input and the output are real-valued, the term $E\left\{y(n)^2\right\}$ is also constant and can be neglected (if the signals are complex-valued, it is required that $E\left\{s(n)^2\right\} = 0$, and, in this case, $E\left\{y(n)^2\right\} = 0$). These considerations reduce the criterion to [269]

$$J_{SW_c} = \operatorname{sgn}\left[c_4\left(s(n)\right)\right] E\left\{|y(n)|^4\right\} \tag{4.37}$$

We may write the output signal as

$$y(n) = \sum_i w(i)\tilde{x}(n-i) \tag{4.38}$$

where $\tilde{x}(n)$ is the channel output after a prewhitening step. Hence, after replacing (4.38) in (4.37) and performing the differentiation, we obtain

$$\frac{\partial J}{\partial w(i)} = 4 \operatorname{sgn}\left[c_4\left(s(n)\right)\right] E\left\{|y(n)|^2 y(n)\tilde{x}(n-i)\right\} \tag{4.39}$$

Finally, the update rule is obtained by replacing (4.39) in (4.34) and removing the expectation operator:

$$\tilde{\mathbf{w}}(n+1) = \mathbf{w}(n) + \mu \operatorname{sgn}\left[c_4\left(s(n)\right)\right] |y(n)|^2 y(n) \tilde{\mathbf{x}}(n) \tag{4.40a}$$

$$\mathbf{w}(n+1) = \left(\frac{1}{\sqrt{\sum_i \tilde{w}(i)^2}}\right) \tilde{\mathbf{w}}(n+1) \quad \text{(normalization)} \tag{4.40b}$$

where $\tilde{\mathbf{x}}(n)$ is the $M \times 1$ vector containing the present and past samples of the prewhitened sequence.

Despite the simplicity of the constrained version of the SWA presented in (4.40), the prewhitening requirement can be very stringent or even prohibitive in some cases. Therefore, in the sequel, we describe an unconstrained version that does not demand a prewhitening step.

4.4.2 Unconstrained Algorithm

Another possibility, developed in [269], is to transform (4.33) into an unconstrained optimization problem. The main idea behind this approach is to include a penalty term related to the constraint into the cost function. The derivation starts by considering the following potential function:

$$\begin{aligned} \phi(\mathbf{g}) &= F(\mathbf{g}) + f\left(\|\mathbf{g}\|^2\right) \\ &= \sum_i |g(i)|^4 + f\left(\sum_i |g(i)|^2\right) \end{aligned} \tag{4.41}$$

where $F(\mathbf{g}) = \sum_i |g(i)|^4$ and $f : [0, \infty] \rightarrow \mathbb{R}$ is a measurable function so that

$$l(x) = x^2 + f(x) \tag{4.42}$$

is monotonically increasing in $0 \leq x < 1$, monotonically decreasing for $x > 1$, and has a unique maximum at $x = 1$.

In order to verify if $\phi(\mathbf{g})$ has a maximum if and only if $\mathbf{g}$ represents the ZF solution, we can notice that [269]

$$\begin{aligned} \phi(\mathbf{g}) &= \sum_i |g(i)|^4 + f\left(\sum_i |g(i)|^2\right) \\ &\leq \left(\sum_i |g(i)|^2\right)^2 + f\left(\sum_i |g(i)|^2\right) = l\left(\sum_i |g(i)|^2\right) \end{aligned} \tag{4.43}$$

where the equality holds if and only if $\mathbf{g}$ has at most one nonzero element. Moreover, from (4.42), we have

$$l\left(\sum_i |g(i)|^2\right) \leq l(1) \tag{4.44}$$

being the equality valid if and only if $\sum_i |g(i)|^2 = 1$. Hence, from (4.43) and (4.44), we know that only for ideal solutions the equalities are satisfied.

Now, for the definition of a cost function, it is necessary to specify the $l(\cdot)$ function. Let us consider the choice

$$l(x) = 2\alpha x - \alpha x^2, \quad \alpha > 0 \tag{4.45}$$

which, as required, monotonically increases in $0 \leq x < 1$ and decreases for $x > 1$. By a direct substitution of (4.45) into (4.42), and performing some algebraic manipulations, we obtain

$$\begin{aligned} J &= \frac{c_4(y(n))}{c_4(s(n))} - (1+\alpha)\frac{E^2\left\{|y(n)|^2\right\}}{E^2\left\{|s(n)|^2\right\}} + 2\alpha\frac{E\left\{|y(n)|^2\right\}}{E\left\{|s(n)|^2\right\}} \\ &= \frac{1}{c_4(s(n))}\left[E\left\{|y(n)|^4\right\} - \left|E\left\{|y(n)|^2\right\}\right|^2 - \left(2 + \frac{(1+\alpha)c_4(s(n))}{E^2\left\{|s(n)|^2\right\}}\right)E^2\left\{|y(n)|^2\right\}\right. \\ &\quad \left. + 2\alpha\frac{c_4(s(n))}{E\left\{|s(n)|^2\right\}}E\left\{|y(n)|^2\right\}\right] \end{aligned} \tag{4.46}$$

Equation 4.46 can still be rewritten as

$$\begin{aligned} J_{SW_c} = \operatorname{sgn}\left[c_4(s(n))\right]\Big\{ & E\left\{|y(n)|^4\right\} - \left|E\left\{y(n)^2\right\}\right|^2 \\ & + \gamma_1 E^2\left\{|y(n)|^2\right\} + 2\gamma_2 E\left\{|y(n)|^2\right\}\Big\} \end{aligned} \tag{4.47}$$

where

$$\gamma_1 = -\left[2 + \frac{(1+\alpha)c_4(s(n))}{E^2\left\{|s(n)|^2\right\}}\right] \tag{4.48}$$

and

$$\gamma_2 = \alpha\frac{c_4(s(n))}{E\left\{|s(n)|^2\right\}} \tag{4.49}$$

Differentiating (4.47) we get to

$$\frac{\partial J_{SW_c}}{\partial w_i} = 4\,\mathrm{sgn}(c_4(s(n)))\Big[E\left\{|y(n)|^2 y(n)x^*(n-i)\right\} - E\left\{y^2(n)\right\}E\left\{y^*(n)x^*(n-i)\right\}$$

$$+\gamma_1 E\left\{|y(n)|^2\right\}E\left\{y(n)x^*(n-i)\right\} + \gamma_2 E\left\{y(n)x^*(n-i)\right\}\Big] \tag{4.50}$$

The problem is how to estimate the expectations $E\left\{y^2(n)\right\}$ and $E\left\{|y(n)|^2\right\}$ in (4.50). The original proposal in [269] uses an empirical average for those expectations and, for the correlations, employs a stochastic approximation. This leads to the following adaptation procedure:

$$\mathbf{w}(n+1) = \mathbf{w}(n)\mu\left(|y(n)|^2 - \frac{E\left\{|s(n)|^4\right\}}{E\left\{|s(n)|^2\right\}}\right)y(n)\mathbf{x}^*(n)$$

$$\left[\left(|y(n)|^2 + \gamma_1|\widehat{y^2(n)}| + \gamma_2\right)y(n) - \widehat{y^2(n)}y^*(n)\right]\mathbf{x}^*(n) \tag{4.51a}$$

$$\widehat{y^2(n)} = (1-\mu_1)y^2\widehat{(n-1)} + \mu_1 y^2(n) \tag{4.51b}$$

$$\widehat{|y^2(n)|} = (1-\mu_2)|y^2\widehat{(n-1)}| + \mu_2|y(n)|^2 \tag{4.51c}$$

where μ_1 and μ_2 are the step sizes for the estimation of $E\left\{y^2(n)\right\}$ and $E\left\{|y(n)|^2\right\}$, respectively.

It is worth noting that if we consider $E\{s(n)^2\} = 0$ and $c_4(s(n)) < 0$, which is the case for digital modulation signals, we have

$$\alpha = -\frac{E|s(n)|^4}{c_4(s(n))} \tag{4.52}$$

and the update rule in (4.51) becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\left(|y(n)|^2 - \frac{E\{s(n)^4\}}{E\{s(n)^2\}}\right)y(n)\mathbf{x}(n) \tag{4.53}$$

which is the Godard/CMA algorithm [124,292]. More about the relationships between blind equalization algorithms will be said in Section 4.7.

4.5 The Super-Exponential Algorithm

Shalvi and Weinstein [270] proposed the SEA as an alternative to accelerate the convergence of the techniques discussed in Section 4.4. The algorithm is derived from a modification of the constrained criterion in (4.33), which leads to

$$\text{maximize} \; \frac{\left|c_4\left(y(n)\right)\right|}{c_2\left(y(n)\right)^2} \tag{4.54}$$

which characterizes a normalized criterion. The cost function can be rewritten in terms of the cumulants of the transmitted signal as

$$\frac{\left|c_4\left(y(n)\right)\right|}{c_2\left(y(n)\right)^2} = \frac{c_4\left(s(n)\right)}{c_2\left(s(n)\right)^2} \underbrace{\frac{\sum_l |g(l)|^4}{\left(\sum_l |g(l)|^2\right)^2}}_{f_4(\mathbf{g})} \tag{4.55}$$

where we define

$$f_4(\mathbf{g}) = \frac{\sum_l |g(l)|^4}{\left(\sum_l |g(l)|^2\right)^2} \tag{4.56}$$

for which we have

$$0 \leq f_4(\mathbf{g}) \leq 1 \tag{4.57}$$

so that $f_4(\mathbf{g}) = 1$ only if $\mathbf{g}$ presents a single non-null element, which corresponds to the ZF solution. Moreover, it is possible to show that

$$f_4(\mathbf{g}) = f_4(\alpha\mathbf{g}) \tag{4.58}$$

for any non-null constant α.

Since the statistical characteristics of the transmitted signal are known, the criterion in (4.54) is equivalent to the maximization of $f_4(\mathbf{g})$, a criterion whose origin can be traced back to the works of Wiggins [307] and Donoho [101] about signal deconvolution.

In order to maximize $f_4(\mathbf{g})$, the SEA explores a simple iteration employing the *Hadamard exponent* of a vector. Let us consider, for the moment, that $\mathbf{g}$ is real-valued. The mth Hadamard exponent of $\mathbf{g}$ is defined componentwise by

$$(\mathbf{g}^{\odot m})_k = \mathbf{g}(k)^m \tag{4.59}$$

If $\mathbf{g}$ presents a unique dominant term in the kth position, the mth Hadamard exponent $(\mathbf{g}/\mathbf{g}(k))^{\odot m}$ will converge to a ZF condition as m tends to infinity.

Thus, the following iteration

$$\boldsymbol{\nu} = \mathbf{g}^{\odot m} \tag{4.60}$$

$$\mathbf{g} = \frac{\boldsymbol{\nu}}{||\boldsymbol{\nu}||} \tag{4.61}$$

converges asymptotically to the ideal ZF response, when initialized by a unit-norm response (g) having a unique dominant element.

Since the combined response depends on the unavailable channel impulse response **h**, the iterative procedure is not practical* and must be adjusted in order to work in the equalizer domain. To do so, let us consider that both the channel and the equalizer can be represented by finite impulse responses, whose coefficients form vectors **h** and **w**, respectively. In this case, the combined response in terms of the channel and equalizer coefficients is

$$\mathbf{g} = \mathcal{H}\mathbf{w} \tag{4.62}$$

where

$$\mathcal{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{M-1} & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{M-1} & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{M-1} \end{bmatrix} \tag{4.63}$$

is the so-called channel convolution matrix.

The aim of the adaptive method would then be to yield **w** such that $\mathbf{g} = \mathcal{H}\mathbf{w}$ be an ideal solution, i.e., $\mathbf{g} = [0, \ldots, 0, 1, 0, \ldots, 0]$. However, it is important to notice that since we are assuming an FIR equalizer, there may not exist a ZF equalizer for a given channel, which leads us to the concept of the attainable set of a channel. The attainable set $\mathcal{T}$ is given by

$$\mathcal{T} \triangleq \left\{ \mathbf{g} : \mathbf{g} = \mathcal{H}\mathbf{w}, \forall \mathbf{w} \in \Re^L \right\} \tag{4.64}$$

where L is the length of the equalizer. Thus, for a given global response, we may obtain its projection onto the attainable set by applying a projection operator, given by

$$\mathbf{P}_T = \mathcal{H}\left(\mathcal{H}^H\mathcal{H}\right)^{-1}\mathcal{H}^H \tag{4.65}$$

* We refer the interested reader to [70,270], which contain a more detailed discussion regarding the convergence of this iterative procedure.

Considering the previous discussion, in [248] it is shown that the iterative procedure composed of

$$\boldsymbol{\nu} = \mathbf{P}_T(\mathbf{g}^{\odot<3>}) \tag{4.66}$$

$$\mathbf{g} = \frac{\boldsymbol{\nu}}{||\boldsymbol{\nu}||} \tag{4.67}$$

where $\mathbf{g}^{\odot<3>}$ is defined componentwise as

$$(\mathbf{g}^{\odot<3>})_k = |\mathbf{g}(k)|^2\mathbf{g}(k) \tag{4.68}$$

will converge to the same stationary points of $f_4(\mathbf{g})$ for an initial choice of a unit-norm vector $\mathbf{g} \in \mathcal{T}$. It is also demonstrated that this kind of iteration may converge to the ZF solution at a super-exponential rate, and, hence, the name of the algorithm.

Since we are dealing with finite length equalizers, it is necessary to look for a solution in the equalizer parameter space $\widetilde{\mathbf{w}}$ such that $\mathcal{H}\widetilde{\mathbf{w}}$ be as close as possible to the solution obtained by the iteration in (4.66), i.e.,

$$\min_{\widetilde{\mathbf{w}}} ||\mathcal{H}\widetilde{\mathbf{w}} - \mathbf{g}^{\odot<3>}||^2 \tag{4.69}$$

The solution of this optimization problem is given by

$$\widetilde{\mathbf{w}} = \left(\mathcal{H}^H\mathcal{H}\right)^{-1}\mathcal{H}^H\mathbf{g}^{\odot<3>} \tag{4.70}$$

and the normalization step in the iteration in (4.66) becomes

$$\widetilde{\mathbf{w}} = \frac{1}{\sqrt{\widetilde{\mathbf{w}}^H\mathcal{H}^H\mathcal{H}\widetilde{\mathbf{w}}}}\widetilde{\mathbf{w}} \tag{4.71}$$

The procedure still depends on the unknown channel and $\mathbf{g}$. Therefore, it is necessary to express these quantities in terms of the cumulants of the equalizer input and output signals. Employing the cumulant properties described in Chapter 2, we may obtain the following relationships:

$$c\left(x(n-j), x^*(n-i)\right) = c_2\left(s(n)\right)\left(\mathcal{H}^T\mathcal{H}\right)_{ij} \tag{4.72}$$

where $(\cdot)_{ij}$ is the (i,j)th element of the matrix in the argument, and

$$\begin{aligned} &c\left(y(n), y(n), y^*(n), x^*(n-i)\right) \\ &\quad = c\left(s(n), s(n), s^*(n), x^*(n-i)\right)\left(\mathcal{H}^H\mathcal{H}\right)^{-1}\mathcal{H}^H\mathbf{g}^{\odot<3>} \end{aligned} \tag{4.73}$$

Using these two results in (4.70) and (4.71), we finally get to

$$\widetilde{\mathbf{w}}(n+1) = \mathbf{R}^{-1}\mathbf{b} \tag{4.74a}$$

$$\mathbf{w}(n+1) = \frac{\widetilde{\mathbf{w}}(n+1)}{\sqrt{\widetilde{\mathbf{w}}^{T}(n+1)\mathbf{R}\widetilde{\mathbf{w}}(n+1)}} \tag{4.74b}$$

where matrix **R** is square with a dimension equal to the equalizer length, having elements given by [270]

$$r_{ij} = \frac{c_2\left(x(n-i);x(n-j)\right)}{c_2(s(n))} = \frac{\text{cov}\left(x(n-i);x(n-j)\right)}{\text{var}(s(n))} \tag{4.75}$$

and **b** is a column vector with dimension equal to the equalizer length, its elements being given by

$$b_i = \frac{c\left(y(n),y(n),y^*(n),x^*(n-l)\right)}{c_4(s(n))} \tag{4.76}$$

In the algorithm shown in (4.74a) and (4.74b), we only need to know the cumulants of the input and to compute the joint cumulant of the input and output signals. This can be done by means of empirical cumulants, as shown and discussed in [70, 270, 271]. In this sense, the property of convergence of the SEA shows some advantages in terms of speed, and there are no significant hypotheses about the signal model, an exception being the usual non-Gaussianity assumption.

4.6 Analysis of the Equilibrium Solutions of Unsupervised Criteria

In Chapter 3 we presented and discussed the Wiener theory, founded on an MSE criterion, the importance of which can be justified on at least two bases: conceptual soundness and mathematical tractability. Therefore, this approach establishes a benchmark against which any other solution could be compared. Now, if the delay is considered an additional free parameter in the search of the optimal equalizer, the Wiener procedure acquires a multimodal character, as typically occurs with the unsupervised criteria. In order to well explain such idea, we take the following example.

Let us consider a binary $(+1/-1)$ i.i.d. signal transmitted by a noiseless FIR channel whose transfer function is $H(z) = 1+1.5z^{-1}$. We are interested in

TABLE 4.2

Wiener Solutions for Different Equalization Delays

Delay (d)	Wiener Solution	MSE
<0	$[0, 0]^T$	1
0	$[0.391, -0.180]^T$	0.609
1	$[0.406, 0.120]^T$	0.271
2	$[-0.271, 0.586]^T$	0.120
>3	$[0, 0]^T$	1

finding a two-tap equalizer using the Wiener approach. Following the procedure shown in Example 3.1, we reach the solutions, and their corresponding MSEs, shown in Table 4.2.

We can observe that each delay produces a different solution with a distinct residual MSE. In addition to that, it is important to point out that the values of the third column reveal a significant discrepancy between the performances of the three nontrivial solutions. The results show that it is also necessary to select an optimal equalization delay. This feature is to be considered in the subsequent analysis.

4.6.1 Analysis of the Decision-Directed Criterion

As discussed in Section 4.3.1, the DD algorithm deals with the following error signal to drive an LMS-based updating:

$$e(n) = dec\left[y(n)\right] - y(n) \tag{4.77}$$

From this general idea, some conclusions can be immediately reached:

- Since the estimate presented in (4.77) does not depend on the equalization delay, it is possible for the DD algorithm to reach generic solutions as far as this parameter is concerned.
- Under the condition

  $$dec\left[y(n)\right] = s(n-d) \tag{4.78}$$

  there is an equivalence between DD and Wiener criteria.
- It is harder to say something about the behavior of the DD algorithm when the above condition does not hold.

As this initial discussion indicates, the analysis of the equilibrium solutions of the DD algorithm will depend on the possibility of establishing relationships with solutions provided by the Wiener criterion (in some cases, ZF solutions will also be relevant).

The work by Mazo [205] is a classical reference in the study of the DD criterion. In his work, he deals with the analysis of the DD cost function in the combined channel + equalizer domain g considering the binary case. The analysis reveals that even in the combined channel + equalizer domain, there are two classes of minima: one formed by ideal ZF solutions, the good minima; and one class with solutions that are not capable of eliminating the ISI, the spurious minima.

Therefore, if the DD algorithm is not adequately initialized, there is a serious risk of ill convergence. The existence of spurious minima in the DD algorithm may be intuitively understood, since if an adequate initial condition is not provided, the resulting decision errors will decisively compromise the entire process. This is exactly the reason why this blind technique is so intimately associated with a preliminary supervised training period.

4.6.2 Elements of Contact between the Decision-Directed and Wiener Criteria

As previously outlined, the key to a perfect equivalence between the DD and Wiener criteria is the situation in which there are no decision errors. Therefore, if a certain parameter vector **w** leads to that condition, we have

$$J_W(\mathbf{w}) = J_{DD}(\mathbf{w}) \tag{4.79}$$

In a region around such solution, in which the equalizer still does not lead to decision errors, we will still have the equivalence $J_W(\mathbf{w}) = J_{DD}(\mathbf{w})$, i.e., the DD surface will be identical to the Wiener paraboloid. The same is valid for regions around other Wiener minima, as long as they do not incur decision errors. This situation is illustrated in Figure 4.4, which provides a contour plot of the cost function J_{DD} in the scenario depicted in Example 3.1.

The figure reveals that the best minima of the DD cost function are exactly the two Wiener solutions with the smallest residual MSE (vide Table 4.2) and their symmetrics. These two solutions are marked in Figure 4.4 (the best solution and its symmetric are marked with an "o" sign and the second best Wiener solution and its symmetric are marked with a "*" sign).

These two Wiener solutions are exactly the only ones that are able to attain a situation with no decision errors (Figure 4.5). The existence of symmetric versions of all minima in this case is unavoidable, since J_{DD} is invariant to changes in the sign of all elements of a given parameter vector.

Other minima of J_{DD} are visible that do not correspond to any of the three Wiener solutions exposed in Table 4.2: these are exactly the spurious minima mentioned above. In the regions around these minima there are decision errors, and they are harder to characterize, since the link with the Wiener criterion is lost.

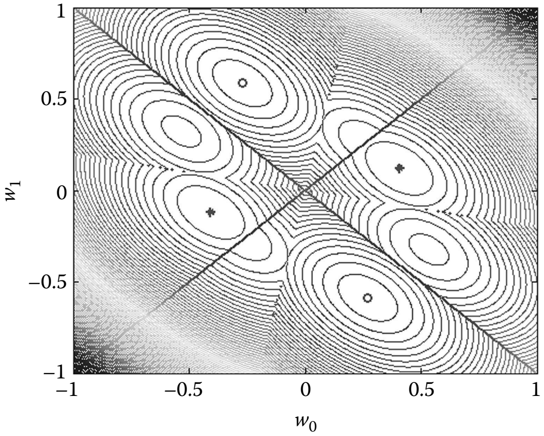


FIGURE 4.4
Contours of the DD cost function.

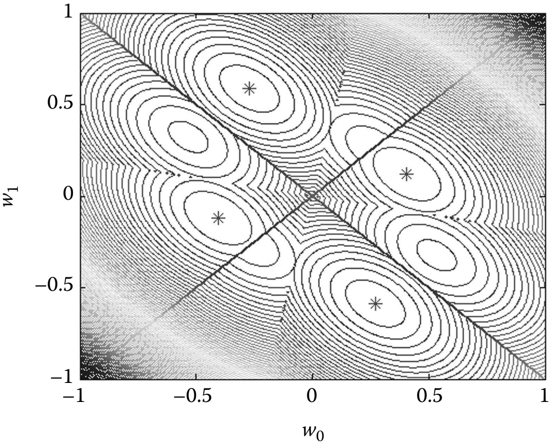


FIGURE 4.5
Contours of the DD cost function and Wiener solutions.

Hence, a natural conclusion is that the DD algorithm is a technique that may be safely adopted given that a fair initial condition is provided.

4.6.3 Analysis of the Constant Modulus Criterion

The difficulty regarding initialization is quite appealing, and, indeed, it was in the spirit of this question that Godard's seminal work [124] must be understood. Even though Godard's work provided a family of cost functions given by (4.26), only two cost functions are commonly used: the one from the choice $p = 1$ and $q = 2$, i.e., which gives rise to a kind of modified Sato algorithm [139], and the one that emerges from the choice $p = q = 2$, i.e.,

$$J_{CM}(\mathbf{w}) = E\left[R_2 - |y(n)|^2\right]^2 \tag{4.80}$$

which leads to the *CMA* [292]. Although the first of these techniques is an interesting object of study [33, 34], we will concentrate our attention on the second one.

The name of this technique, first presented in the work by Treichler and Agee [292], comes from the fact that if the transmitted signal is generated from a CM modulation, like FM and PSK, then the CM criterion will attempt to restore this CM property. However, it is important to remark that the technique is useful even for other modulations.

Besides the CM recovery property, another aspect that deserves attention is the fact that only the modulus of the equalizer output is taken into account by the criterion, which means that the problems of equalization and phase recovery are decoupled when the CM criterion is employed. The practical implications of this aspect are considered in Godard's work, but, for now, it suffices to say that it makes all solutions with the same modulus undistinguishable.

4.6.4 Analysis in the Combined Channel + Equalizer Domain

In his work, Godard carries out an analysis in the combined channel + equalizer domain, which leads to the conclusion that the ZF solutions are global minima of the CM cost function. However, there was more to be said on the subject, and it was only in 1985 that Foschini presented a more detailed analysis of the CM criterion in this domain [113].

The key element in Foschini's analysis was to look for equilibrium points of the CMA in the combined domain with a variable number χ of nonzero elements. By considering the gradient of J_{CM} and its second-order derivatives, he showed the existence of three classes of solutions with

1. $\chi = 0$, i.e., the null vector, which is a maximum
2. $\chi = 1$, which are the ZF minima
3. $\chi > 1$, which are saddle points

The first two kinds of solutions are not surprising. The maximum is present in all scenarios in which the CMA is used and this is why the initialization $\mathbf{w}(0) = \mathbf{0}$ cannot be considered. The class for which $\chi = 1$ was previously found by Godard. The novelty undoubtedly lies in the third class: there are equilibrium points that do not correspond to a satisfactory equalization condition. It would be tempting to bring Mazo's analysis of the DD solutions to our mind; nonetheless, in this case, the third class is not composed of spurious minima, but of saddle points that are unable to misguide the adaptation process.

Naturally, the analysis discussed so far was not carried out in the most realistic scenario, but it was possible to think that the transition from the combined domain to the equalizer domain would not significantly compromise the general conclusions. In fact, Foschini's work contains two ideas of both practical and historical relevance. The first of these ideas is related to the so-called center-spike initialization, in which the initial value of **w** is taken to be the null vector, except for the central element, which is unitary, e.g.:

$$\mathbf{w} = [0,0,\ldots,0,1,0,\ldots,0]^T \tag{4.81}$$

The second idea is even more intuitive: Foschini considered that a generic (although bounded) increase in the number of equalizer taps would be a valuable bridge between the equalizer and combined domains. This indicates the possibility that an arbitrarily large equalizer would lead to a scenario similar to that discovered in the analysis of the combined space. This is, a priori, an appealing view, since an increase in the number of taps opens the perspective of reaching solutions in the combined domain that are closer to ZF solutions.

An important progress regarding the conditions for the equivalence between the combined and the equalizer domains was given in [97]. In this work, Ding et al. showed that there are two fundamental possibilities for the equivalence:

- When the channel is trivial, i.e., is simply a gain. In this case, there is no ISI and, as a consequence, the combined response will be completely equivalent to the equalizer response.
- When the equalizer is doubly infinite. In this case, the equalizer is capable of perfectly inverting an FIR channel, which, once more, creates a perfect equivalence between domains.

4.6.4.1 Ill-Convergence in the Equalizer Domain

A very relevant work by Ding et al. [98] shows that the CM cost function presents solutions distinct to those listed by Foschini that would not reduce ISI, even in a scenario in which an ideal solution is attainable. This work considers a noiseless channel modeled by an all-pole filter whose transfer function is

$$H(z) = \frac{1}{1 + \alpha z^{-N}} \tag{4.82}$$

Given this channel model, the ZF condition is achieved using an FIR equalizer, and no infinitely long equalizers were required. Thus, since perfect equalization is attained for $\mathbf{w} = \pm\,[1, 0, \ldots, \alpha]^T$, it is expected that they also represent ideal solutions of J_{CM}. This assumption, as proved by the authors, is entirely correct. In addition, the authors consider possible solutions given by

$$\mathbf{w}_{eq} = \pm\,\varrho\,[0, \ldots, 0, 1]^T \tag{4.83}$$

where

$$\varrho = \sqrt{R_2 \frac{E\left[|x(n)|^2\right]}{E\left[|x(n)|^4\right]}} \tag{4.84}$$

This solution is not able to reduce ISI, and it is proven to be a minimum of J_{CM} with the aid of an analysis of the derivatives of the cost function.

Let us consider an example, in which the channel transfer function is

$$H(z) = \frac{1}{1 + 0.6z^{-1}} \tag{4.85}$$

From the above reasoning, we are correctly led to expect that the points

$$\mathbf{w} = \pm\,[1, 0.6]^T \tag{4.86}$$

be global minima of J_{CM}. However, there shall also exist solutions like (4.83), solutions that are complete failures insofar as the equalization task is concerned. As shown in Figure 4.6, the minima at $\mathbf{w} = \pm\,[1, 0.6]^T$ are indeed the best and lead to perfect equalization. On the other hand, as the work of Ding et al. anticipates, there is a pair of "shallow minima" that have exactly the form described in (4.83).

Finally, it is worth mentioning that the authors also show, with the aid of simulations, that the bad minima continue to exist in the presence of noise and that there are local minima even for an FIR channel model. These minima, however, are not ineffective minima like the ones discussed above. As a matter of fact, not all local minima are necessarily bad minima. It is important to keep in mind that even the Wiener criterion possesses multiple solutions depending on the perspective brought by distinct equalization delays.

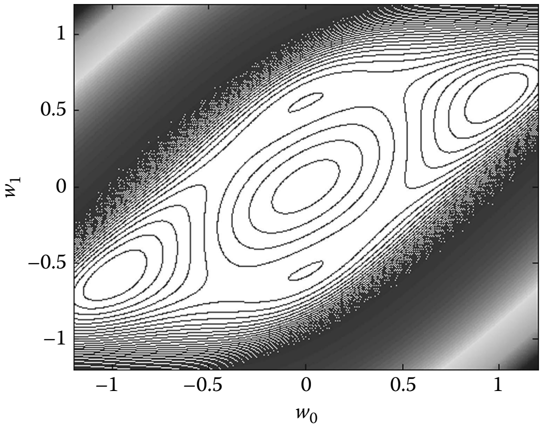


FIGURE 4.6
Contours of the CM cost function for the AR case.

4.7 Relationships between Equalization Criteria

Some of the previously discussed criteria used in blind equalization, even though developed from very different motivations, exhibit a close relationship. Interestingly enough, it is also possible to obtain some relationships between blind equalization criteria and the minima found using the Wiener criterion. In this section we expose some ideas related to these relationships.

4.7.1 Relationships between the Constant Modulus and Shalvi–Weinstein Criteria

After having proposed the constrained criterion described in (4.36), Shalvi and Weinstein analyze it in the combined channel + equalizer domain. They conclude that the equilibrium points (aside from the trivial null vector) are twofold: a family of ZF maxima and a set of saddle points. Later, when unconstrained criteria are discussed, a new analysis leads once more to these solutions. If we recall the discussion carried out in Section 4.6.4, we are compelled to state that the results of Shalvi and Weinstein are essentially equivalent to Foschini's analysis of the CM criterion. In other words, both the CM and SW criteria give rise to equivalent solutions in the combined domain.

In the same paper, more is said about the equivalence between the CM criterion and the SW methods. Under their fundamental assumptions, Shalvi and Weinstein show that it is possible to obtain the CM criterion and the CMA as particular cases of their methods.

These proofs of equivalence are compromised by the limitation of the scenarios in which they were obtained. Five more years would pass before the appearance of Li and Ding's work [183], which contained a more general proof of the equivalence between the SW and CM criteria. This equivalence is concerned with the position of the critical points of the respective cost functions, i.e., with the equilibrium points of the associated algorithms. The proof was carried out in the context of a geometrical framework, and was associated with other relevant theoretical results.

Four years later, in 1999, Regalia would revisit this problem and provide with another simple and elegant proof.

4.7.1.1 Regalia's Proof of the Equivalence between the Constant Modulus and Shalvi–Weinstein Criteria

Regalia's proof is remarkable in the sense that it possesses two features that are not trivially brought together: simplicity and generality. His assumptions were exiguous—in his own words, "little more than stationarity to fourth order of the equalizer input"—and his goal was ambitious: to relate the CM and SW criteria by means of a transformation of their cost functions, which is more than analyzing critical points.

First, we must choose a representation of the parameter vectors in terms of polar coordinates, a representation that allows us to decouple magnitudes and directions. In this representation, the equalizer parameter vector $\mathbf{w}$ can be thought of as

$$\mathbf{w} = \rho\overline{\mathbf{w}} \tag{4.87}$$

where

ρ is the norm of $\mathbf{w}$

$\overline{\mathbf{w}}$ is a unit-norm parameter vector

It is also possible to extend this idea to the output signal

$$y(n) = \rho\overline{y}(n) \tag{4.88}$$

where

$y(n)$ is the equalizer output

$\overline{y}(n)$ is a unit-norm version thereof

Under these circumstances, a normalized version of the SW cost function [269,293] is written as

$$J_{SW}(\Theta) = \frac{c_4\left(y(n)\right)}{\left(E\left[y^2(n)\right]\right)^2} = \frac{c_4\left(\overline{y}(n)\right)}{\left(E\left[\overline{y}^2(n)\right]\right)^2} \tag{4.89}$$

The dependence of J_{SW} on Θ indicates that the cost function depends only on the angles of the polar representation. Analogously, the CM cost function becomes

$$J_{CM}(\rho, \Theta) = E\left[\left(R_2 - |\rho|^2 \left|\bar{y}(n)\right|^2\right)^2\right] \tag{4.90}$$

An analysis of Equations 4.89 and 4.90 reveals the usefulness of employing polar coordinates: it becomes explicit that the SW criterion does not depend on the norm of the parameter vector, whereas the CM criterion shows this dependence. On the other hand, this fact poses a difficulty insofar as the way of comparing these two approaches is concerned.

Regalia's idea was to perform the optimization of $J_{CM}(\rho, \Theta)$ with respect to ρ by differentiating it with respect to this parameter and equaling the result to zero:

$$\frac{\partial J_{CM}(\rho, \Theta)}{\partial \rho} = 4\rho\left[\rho^2 E\left[\left|\bar{y}(n)\right|^4\right] - R_2 E\left[\left|\bar{y}(n)\right|^2\right]\right] = 0 \tag{4.91}$$

The resulting equation has the following solutions:

$$\rho_{opt} = 0 \ or \ \rho_{opt} = \pm\sqrt{R_2 \frac{E\left[\left|\bar{y}(n)\right|^2\right]}{E\left[\left|\bar{y}(n)\right|^4\right]}} \tag{4.92}$$

The first of these solutions is the well-known maximum at $\mathbf{w} = \mathbf{0}$, whereas the two other conditions are of interest (minima and saddle points). Using these values in (4.90), we get to

$$J_{CM}(\Theta) = R_2^2\left(1 - \frac{\left[E\left(\left|\bar{y}(n)\right|^2\right)\right]^2}{E\left[\left|\bar{y}(n)\right|^4\right]}\right) \tag{4.93}$$

which can be rewritten as

$$J_{CM}(\Theta) = R_2^2\left(1 - \frac{1}{2 + \frac{c_4(\bar{y}(n))}{\left[E\left(\left|\bar{y}(n)\right|^2\right)\right]^2} + \frac{E\left[(\bar{y}(n))^2\right]E\left[(\bar{y}^*(n))^2\right]}{\left[E\left[\left|\bar{y}(n)\right|^2\right]\right]^2}}\right) \tag{4.94}$$

At this point, it is necessary to take into account whether we deal with real- or complex-valued signals. From now on, we will assume that the former case is valid (however, the results obtained for complex signals are quite similar). Under this assumption we get to

$$J_{CM}(\Theta) = R_2^2\left[1 - \frac{1}{3 + J_{SW}(\Theta)}\right] \tag{4.95}$$

This expression reveals that the CM cost function can be understood as a sort of deformation of the SW cost function. The rest of the analysis amounts essentially to a study of the nature of this deformation. First, let us study the deformation with respect to monotonicity. This can be done with the aid of the following expression:

$$\frac{dJ_{CM}(\Theta)}{dJ_{SW}(\Theta)} = \frac{R_2^2}{[3 + J_{SW}(\Theta)]^2} \tag{4.96}$$

Since this expression is always greater than zero, the deformation is monotonic. Furthermore, the expression reveals that all stationary points in Θ of the two cost function coincide, i.e., that

$$\left.\frac{dJ_{CM}(\Theta)}{d\Theta}\right|_{\Theta_1} = 0 \Leftrightarrow \left.\frac{dJ_{SW}(\Theta)}{d\Theta}\right|_{\Theta_1} = 0 \tag{4.97}$$

Finally, it is possible to show that

$$J_{CM}(\Theta_1) - J_{CM}(\Theta_2) = R_2^2 \frac{J_{SW}(\Theta_1) - J_{SW}(\Theta_2)}{[3 + J_{SW}(\Theta_1)]\,[3 + J_{SW}(\Theta_2)]} \tag{4.98}$$

which means that the deformation is order-preserving, i.e., that

$$J_{CM}(\Theta_1) > J_{CM}(\Theta_2) \Leftrightarrow J_{SW}(\Theta_1) > J_{SW}(\Theta_2) \tag{4.99}$$

As (4.97) shows, all stationary points of the two cost functions coincide. In addition to that, (4.96) and (4.99) reveal that the classification of each point is identical, which renders the equivalence even stronger. However, we should notice that this equivalence is restricted to the stationary points: the criteria in their entirety are not identical, which means that there exist performance aspects that may be different for both methods and their correspondent algorithms.

4.7.2 Some Remarks Concerning the Relationship between the Constant Modulus/Shalvi–Weinstein and the Wiener Criteria

The relationships between the DD and Wiener criteria and between the CM and SW criteria are established within similar statistical frameworks: in the

TABLE 4.3

Equalization Delay and Wiener Solutions

Delay (d)	Wiener Solution	MSE
<0	$[0, 0]^T$	1
0	$[0.391, -0.180]^T$	0.609
1	$[0.406, 0.120]^T$	0.271
2	$[-0.271, 0.586]^T$	0.120
>3	$[0, 0]^T$	1

first case, a second-order framework that emerges under conditions in which the ISI is eliminated, and in the second case, a fourth-order framework. However, a more difficult task arises when it is necessary to interrelate these two frameworks, which is the case, for instance, when we seek to compare the CM and Wiener criteria.

Let us consider the same scenario discussed in Section 4.6. The Wiener solutions and corresponding MSEs for this case are presented again in Table 4.3 for convenience.

The solutions obtained with the CM criterion in this scenario are close to some of the Wiener minima. In Figure 4.7, we present the contours of the CM cost function for this case. We notice that the function has a pair of global minima, $[0.246, -0.522]^T$ and $[-0.246, 0.522]^T$, and a pair of local minima, $[0.378, 0.065]^T$ and $[-0.378, -0.065]^T$. Notice that one minimum of each

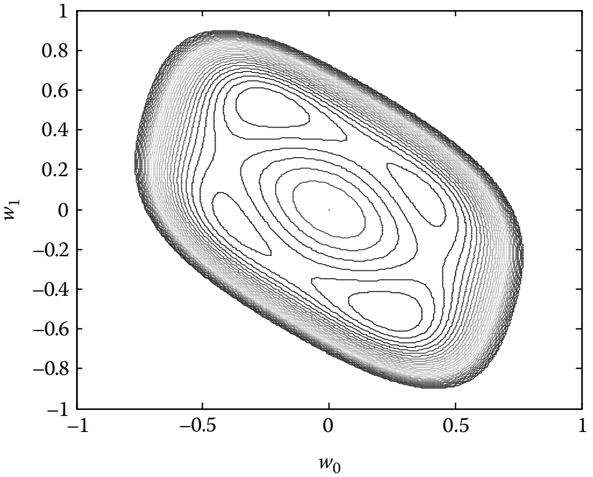


FIGURE 4.7
Contour plots of the CM cost function.

pair is close to one of the best Wiener solutions, which are $[-0.271, 0.586]^T$—for $d = 2$—, the best of the three, and $[0.406, 0.120]^T$ (for $d = 1$), and that the worst Wiener solution (for $d = 0$) has no counterpart in the CM cost function.

This situation is also observed in other scenarios, and some works in the literature indeed studied the relationship between the minima of these criteria and their cost values [264, 312, 313]. An interesting idea that arises from these works is to focus on CM minima close to Wiener solutions associated with certain delays.

In fact, the application of the CM criterion to problems in which an FIR filter is used to equalize an FIR channel often leads to an analytical framework in which CM minima are more or less closely related to a set of good Wiener solutions. If we recall that perfect inversion solutions should be present in both criteria, it is expected that Wiener solutions associated with a small MSE should have counterparts in the CM cost function. On the other hand, solutions obtained from equalization delays that lead to a large MSE might not even be present therein, a fact that is also observed in the previous example.

The analysis in a general case is rather complex, and certain aspects thereof can still be considered to be open. However, if we only consider the special case of binary signals with i.i.d. samples, it is possible to obtain a simple relationship between the CM and Wiener criteria. Under this assumption, the CM cost function can be expressed as

$$J_{CM}(\mathbf{w}) = E\left\{\left[y(n) - s(n-d)\right]^2 \left[y(n) + s(n-d)\right]^2\right\} \tag{4.100}$$

where, in this case, $E\left\{s^4(n-d)\right\} = 1$.

It is useful to recall the important Cauchy–Schwarz inequality [78], which guarantees that

$$E\{ab\}^2 \leq E\left\{a^2\right\} E\left\{b^2\right\} \tag{4.101}$$

This result, together with (4.100), leads to

$$\begin{aligned} J_{CM}^2(\mathbf{w}) &\leq E\left\{\left[y(n) - s(n-d)\right]^4\right\} E\left\{\left[y(n) + s(n-d)\right]^4\right\} \\ &\leq J_F(\mathbf{w}) J_F(-\mathbf{w}) \end{aligned} \tag{4.102}$$

As $J_{CM} \geq 0$, we may write

$$J_{CM} \leq \sqrt{J_F(\mathbf{w}) J_F(-\mathbf{w})} \tag{4.103}$$

where

$$J_F(\mathbf{w}) = E\left\{\left[y(n) - s\left(n-d\right)\right]^4\right\} \tag{4.104}$$

which corresponds to the cost function associated with the least mean-fourth algorithm (a discussion about the least mean-fourth supervised approach can be found in [301]).

Equation 4.103 provides an upper bound for the CM cost function that is composed of a product of cost functions based on the fourth-order moment of the error signal. In order to better understand this relationship, in Figure 4.8 we show the upper bound and the original cost function. It is possible to observe that the upper bound changes for different values of d. This may be understood as being indicative of the fact that the CM cost function is associated with different equalization delays, as mentioned earlier. An overall upper bound composed of the surfaces associated with all equalization delays is also illustrated Figure 4.8d.

Concluding, in simple terms, we believe it is valid to state that the analysis of the CM criterion from the standpoint of supervised approaches should

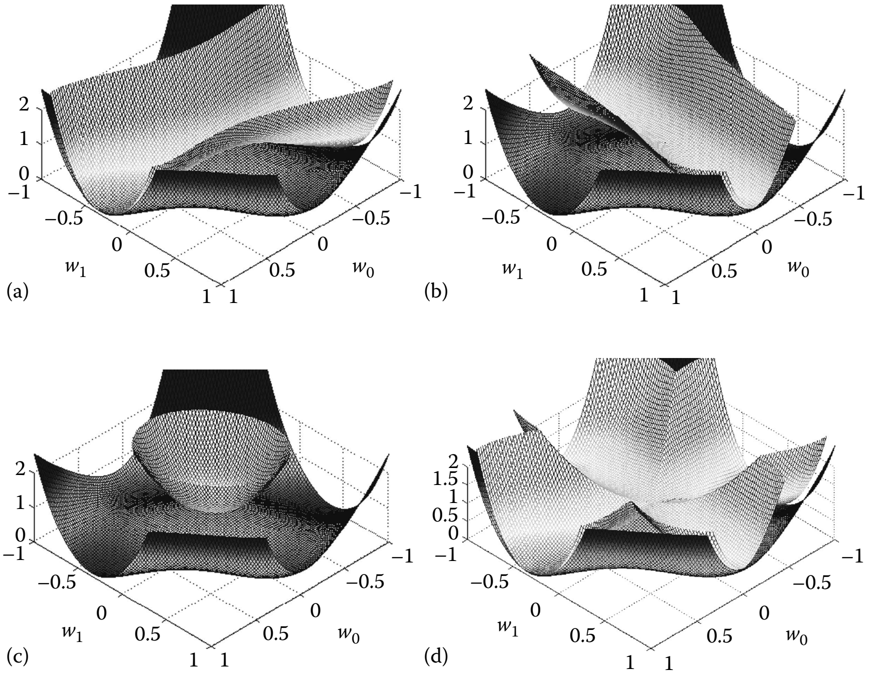


FIGURE 4.8
Derived upper bound for the channel $H(z) = 1 + 1.5z^{-1}$ and different equalization delays: (a) $d = 0$, (b) $d = 1$, (c) $d = 2$, and (d) overall upper bound.

be based on searching for good equalization solutions as well as for a parallel between them and CM minima. Over all, perhaps the main conclusion to be drawn from the comparison between supervised and blind criteria is that of not thinking of a local minimum of a blind criterion necessarily as a bad minimum, especially for the CM criterion, which does not possess spurious minima like those found in approaches like DD [99].

4.8 Concluding Remarks

In this chapter, we stated the problem of unsupervised signal processing, in particular that of channel equalization, in the context of SISO systems. This scenario of interest was introduced after an initial and more general presentation on unsupervised deconvolution, where we discussed the role of second-order statistics in order to pave the way to the subsequent theoretical results.

The central theoretical results are essentially established by the theorems of Benveniste–Goussat–Ruget and Shalvi–Weinstein, enounced in this chapter. Such theorems clearly establish the need of higher-order statistics to provide equalization in an unsupervised mode.

The chapter particularly focuses on the methods that make an implicit use of the higher-order statistics, since our major interest is dealing with adaptive techniques. The Bussgang algorithms constitute a large family of techniques, among which we emphasize DD, Sato, and Godard/CMA algorithms. Also, we took the SW theorem as a starting point to derive SW constrained and unconstrained algorithms, as well as the super-exponential.

If the first part of the chapter was devoted to presentation and derivation of the main algorithms, the second part was concerned with the analysis of the corresponding criteria.

First, we exposed the most classical results of the literature about the equilibrium points of DD and CM criteria. Later, we discussed equivalence relationships between criteria. In addition to the elegance of some results presented in the last sections, it is worth pointing out that such analyses are important to provide an adequate understanding about the operation of the algorithms and, as a consequence, to envisage a suitable practical use of them.

5

Unsupervised Multichannel Equalization

The blind equalization criteria studied in Chapter 4 for the SISO scenario consider either implicit or explicit use of higher-order statistics. In such a framework, the resulting cost function generally presents local minima, so that the performance of the optimization methods often depends on appropriate initializations. Another limiting aspect in SISO equalization was concerned to *equalizability*, in the sense that zero-forcing (ZF) conditions cannot be perfectly attained in the usual assumptions of FIR channel and FIR equalizer.

The works on unsupervised processing in multichannel scenarios have been intensified from the beginning of the 1990s for both practical and theoretical reasons. From a theoretical standpoint, the use of models with multiple inputs and/or multiple outputs revealed the possibility of overcoming the drawbacks of SISO case mentioned above. In fact, two instigating results have been brought out:

- Dealing with multichannel configurations, it is possible, under certain conditions, to attain perfect (ZF) equalization even if both channel and equalizer are FIR structures.
- Also, optimal equalizer can be obtained in unsupervised mode from optimization criteria based only on second-order statistics of the involved signals.

This last and important result becomes conceptually clearer if we consider, for instance, that in SISO case second-order based methods could be effective if we disposed of any suitable prior information, e.g., the phase-response of the channel. In the multichannel case, this additional information is related to some kind of diversity in the system.

The multichannel structure itself is endowed with an inherent spatial diversity, which can be in practice associated to the use of multiple transmitter and/or receiver disposed in the space. As presented further in this chapter, proceeding with the optimization of the parameters of the multichannel structure, we get to optimal solutions that only depend on the autocorrelations and cross-correlations of the involved signals.

On the other hand, a multichannel model can be used to represent temporal diversity, which may be provided by the process of oversampling. Thus, the multichannel model will in fact describe a SISO communication system, operating with sampling rate higher than symbol rate. As presented further in this chapter, the oversampled signals are cyclostationary, and the second-order statistics of cyclostationary signals do provide phase-spectrum information.

In addition to the theoretical aspects commented above, multichannel configurations receive an increasing practical interest in modern communications systems, due to the effectiveness of the aforementioned diversity techniques. Spatial diversity is increasingly applied in wireless communications, by using multiple antennas in the transmitter and/or the receiver, as well as in array of sensors in general. Dealing with temporal diversity by the employment of oversampling and fractionally spaced (FS) equalizers also brings systemic advantages as a lower sensitivity to sampling timing errors and noise amplification.

The above considerations serve as a motivation for this chapter, which is, in a way, built in two distinct parts. First, we systemize the aforementioned theoretical points of multichannel equalization, giving emphasis to the SIMO channel (and MISO equalizer) case, since the extension to the MIMO scenario is not a crucial theoretical issue. Then, as far as MIMO channel is concerned, we give special attention to the problem of multiuser processing, which is particularly important in modern wireless communications systems. It is also worth pointing out that the problem of MIMO equalization itself will be, in a way, revisited in the next chapter when we deal with convolutive source separation. With this in mind, we organize the chapter as follows:

- Section 5.1 states the problem in a general scenario, considering both multiple input and multiple output channels. We establish the requirements for ZF MIMO equalization based on two theorems: the first one introduces the notion of *Smith form*, and the second one uses this definition to provide equalizability conditions.
- As commented above, our theoretical derivations emphasize the case of SIMO channel equalization, which is presented in Section 5.2 as a particularization of the previous section. The main results are
  - We present SIMO channel model and its relationships with the two most typical cases: multiple antennas and temporal oversampling.
  - We discuss the cyclostationary behavior of the oversampled signal, showing that its second-order statistics convey phase information. From there, we formulate the solution based on FS equalizers.
  - We particularize the Smith form to the SIMO case and enounce the *Bezout's identity*, showing how the ZF condition can be attained when we deal with an FIR SIMO channel and an FIR MISO equalizer.
- Once the problem is described, we discuss in Section 5.3 the solutions for blind equalization of SIMO channels. First, we comment on the use of methods based on higher-order statistics. Then, we

present in more detail two well-established second-order methods: *subspace decomposition* and *linear prediction*.

- In Section 5.4, we change the main focus from equalization itself to the problem of *multiuser processing*, which is closely related to MIMO configurations. The problem consists in recovering a given transmitted signal that is subject to the channel impairments and also to the effect of interfering signals arriving at the same receiver. We present two types of methods: the first one involves the equalization criterion together with an auxiliary term that penalizes the correlation between the received signals; the other methodology relies on the use of an orthogonalization approach based on the Gram–Schmidt procedure.

Historical Notes

The evolution of the blind equalization algorithms from SISO to SIMO systems can be traced back to the work from Tong et al. in 1991 [287]. In this work, they proposed a method for blind equalization and identification of SIMO channels using only the second-order statistics of the involved signals. After the so-called TXK algorithm, a number of algorithms and methods using second-order statistics were proposed in the 1990s, making use of prediction error structures, subspace methods, and fractionally spaced equalizers. Among which we can mention the references in [207,211,229,289]. Moreover, it is important to mention the works of Gardner [116, 117], also in the 1990s, which exploit second-order statistics, but in the context of the cyclic spectrum of cyclostationary signals.

The methods based on higher-order statistics for SIMO systems have been concentrated on the generalization of the SISO algorithms. The consideration of FS equalizers led to the proposal of the fractionally-spaced constant modulus algorithm (FS-CMA) by Li and Ding [184], as one of the most known algorithms on such class. A variant of the super-exponential algorithm for multichannel are also reported in [129].

As far as MIMO channels are concerned, results are also reported by the literature since the early 1990s, with methods based on both second-order and higher-order statistics. Among important contributions to the theme, we can we can point out the works by Hua and Tugnait [146, 296], and other techniques that consider subspace methods, matrix pencil decomposition, channel decorrelation, etc. [15, 130, 193, 207, 309].

The problem of multiuser processing in a MIMO environment is in fact closely related to those of source separation and independent component analysis, to be considered in the next chapter, although these topics and their corresponding scientific communities made their own independent way. The approach of considering independent transmitted signals in a MIMO channel scenario has founded interesting works as [96, 186, 294]. It is worth mentioning the multiuser techniques based on the penalization of correlations between receiver signals, as proposed in [65, 228].

Again, this brief historic is necessarily non-exhaustive since, in spite of the relatively short period of time, many important authors contributed with this nice theme. From the aforementioned references it will be not difficult to access others and, once more, we must mention the scan of the literature provided by [99] as a necessary complement.

5.1 Systems with Multiple Inputs and/or Multiple Outputs

In Chapter 2, we briefly recalled some basic aspects of system theory. Among these aspects, we commented about the systems classification in accordance with the number of input and output signals: SISO, SIMO, MISO, and MIMO systems.

In the equalization scenarios discussed in both Chapters 3 and 4, we considered a single information source and a single receiver. In this case, the channel model is given by a SISO system that is fully characterized by its transfer function. In this context, the definitions of its zeros and poles is straightforward, as well as the conditions for perfect inversion of the channel impulse response.

The concept, however, becomes more involved in a scenario in which we have more than a single pair of transmitted and received signals, as depicted in Figure 5.1.

In this model, signal propagation from each source of information to each receiving device is modeled as a SISO channel, which means that the received signals can be expressed as

$$\begin{aligned} x_i(n) &= \sum_j \sum_\tau h_{ij}(\tau) s_j(n-\tau) \\ &= \sum_j h_{ij}(n) * s_j(n), \end{aligned} \tag{5.1}$$

for $i = 1, \ldots, N$, where $h_{ij}(n)$ denotes the impulse response of the subchannel linking the ith source and the jth sensor. Thus, we may express the vector $\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_M(n)]$ of received signals as

$$\mathbf{x}(n) = \sum_l \mathbf{H}(l)\mathbf{s}\left(n-l\right), \tag{5.2}$$

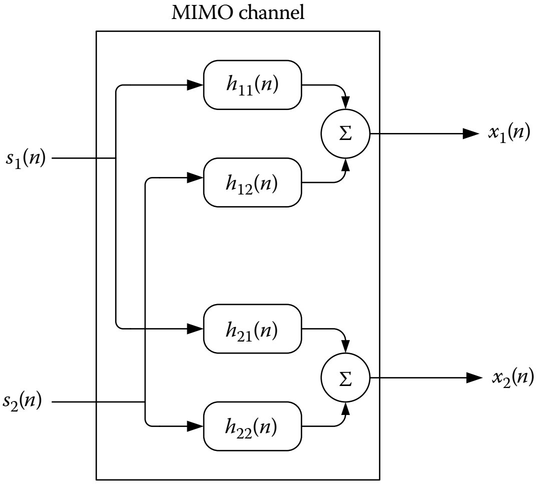


FIGURE 5.1
MIMO scenario with two transmitters and two receivers.

where the matrices $\mathbf{H}(n)$ are composed of the coefficients $h_{ij}(n)$ of all subchannels.

In fact, Equation 5.2 is very similar to that which describes the output of a SISO system, given in (2.21). The output of a MIMO system can thus be interpreted as the convolution between a sequence of matrices $\mathbf{H}(n)$ representing the channel and the vector $\mathbf{s}(n)$ composed of the sources. Hence, we may write

$$\mathbf{x}(n) = \mathbf{H}(n) * \mathbf{s}(n) + \boldsymbol{\nu}(n) \tag{5.3}$$

We can also define the z-transform of the channel impulse response as

$$\mathbf{H}(z) \triangleq \sum_k \mathbf{H}(k) z^{-k} \tag{5.4}$$

which represents a polynomial matrix, since each of its elements is a polynomial in z. A representation of (5.3) is provided in Figure 5.2.

An important classification of the MIMO systems is concerned with the nature of combination of the input (source) signals, which leads to the following cases:

- *Instantaneous*: In this case, the resulting signals at the receiver, apart from the noise, are a combination of the input signals at a given time instant, i.e., the received signal at a time instant n only depends on a

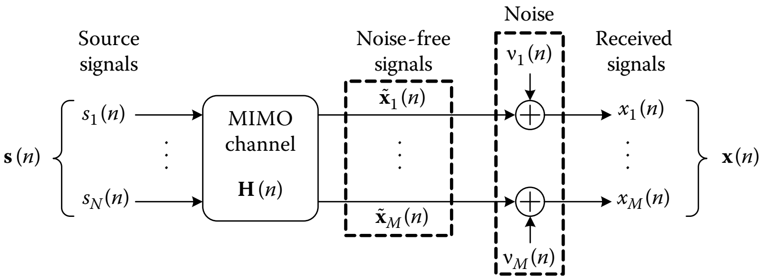


FIGURE 5.2
Received signal model for a MIMO system.

combination of the input signals at the same time instant. Hence, we would have the noise-free signals given by

$$\begin{aligned}\tilde{\mathbf{x}}(n) &= \mathbf{H}(n) * \mathbf{s}(n) \\ &= \mathbf{H}\mathbf{s}(n) \qquad (5.5)\end{aligned}$$

It is usual in the literature to relate this case with narrowband signals, since they practically do not suffer from temporal dispersion.

- *Convolutive*: The received signals at time instant n are a combination of the input signals at instant n and at certain past time instants. In such situations, the noise-free signals are computed according to the general expression (5.3). This model is often associated in the literature with a broadband signal that is susceptible to the effect of linear filtering.

The equalizer may follow the same formulation so that we have

$$\mathbf{y}(n) = \mathbf{W}(n) * \mathbf{x}(n) \qquad (5.6)$$

$$= \mathbf{W}(n) * \mathbf{H}(n) * \mathbf{s}(n) \qquad (5.7)$$

$$= \mathbf{G}(n) * \mathbf{s}(n) \qquad (5.8)$$

where $\mathbf{W}(n)$ and $\mathbf{G}(n)$ denote the equalizer and the combined channel + equalizer impulse responses, respectively.

5.1.1 Conditions for Zero-Forcing Equalization of MIMO Systems

As discussed in Chapter 4, we should obtain an equalizer capable of yielding a signal that is a delayed and possibly scaled version of the transmitted signal: this is the essence of the ZF condition for the SISO case.

However, differently from that case, in a MIMO scenario, it is necessary to take into account that there is more than one signal involved in the equalization problem. We may consider that a perfect equalization condition is achieved if each one of the recovered signals represents a delayed and possibly scaled version of the transmitted signals, even if the order of the signals in the received vector $\mathbf{y}(n)$ is different from the original order in $\mathbf{s}(n)$ [153].

Taking into account these possible ambiguities, we may state that a MIMO channel $\mathbf{H}(z)$ is perfectly equalized, i.e., that the ZF condition is achieved, if there exists an equalizer $\mathbf{W}(z)$ such that

$$\mathbf{W}^H(z)\,\mathbf{H}(z) = \mathbf{PD}\boldsymbol{\Lambda}(z), \tag{5.9}$$

where

$\mathbf{P}$ and $\mathbf{D}$ represent, respectively, a permutation matrix and a non-singular diagonal matrix

$\boldsymbol{\Lambda}(z)$ denotes a diagonal matrix composed of monic monomials z^{-l_i}, $l_i \geq 0$

The existence of a ZF equalizer in the SISO case is readily verified by observing the location of zeros and poles of the channel transfer function. Moreover, if a given SISO channel is modeled as an FIR filter, it is well-known that the ZF condition is not attained, in general, by an FIR equalizer. On the other hand, in a clear contrast with the SISO case, it is possible to obtain the ZF condition with a MIMO-FIR equalizer even if the channel is also a MIMO-FIR structure.

In order to present the main result regarding the attainability of ZF solutions in the MIMO case, let us consider from now on that the channel can be modeled as a MIMO-FIR structure. In analogy with SISO channels, we may also define the zeros of a MIMO-FIR structure. The definition employs the Smith form of $\mathbf{H}(z)$, given by [115].

THEOREM 5.1 (Smith Form)

Let $\mathbf{H}(z)$ denote an $M \times N$ polynomial matrix, in which each element represents a polynomial in z. Then, there exists a finite number of elementary operations over the rows and columns of $\mathbf{H}(z)$ that reduce it to a diagonal polynomial matrix. It hence can be expressed as

$$\mathbf{H}(z) = \mathbf{Q}(z)\,\Gamma(z)\,\mathbf{P}(z), \tag{5.10}$$

where $\mathbf{Q}(z)_{M\times M}$ and $\mathbf{P}(z)_{N\times N}$ are unimodular polynomial matrices in z, i.e., their determinant does not depend on the value of z, and $\Gamma(z)$ is defined as

$$\Gamma(z) = \begin{bmatrix} \gamma_0(z) & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \gamma_1(z) & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots & \vdots & \vdots \\ \vdots & \vdots & 0 & \gamma_{\rho-1}(z) & 0 & \cdots & 0 \\ 0 & 0 & \vdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{5.11}$$

where each element is a monic polynomial $\gamma_i(z)$. In addition to that, it is verified that $\gamma_i(z)$ divides $\gamma_{i+1}(z)$, $0 \leq i \leq \rho - 2$, ρ being the normal rank of $\mathbf{H}(z)$. For a given matrix $\mathbf{H}(z)$, $\Gamma(z)$ is unique, and represents the Smith form of $\mathbf{H}(z)$. ■

Given the Smith form of $\mathbf{H}(z)$, the zeros of the channel are defined as the roots of $\gamma_i(z)$ [25]. Then, the following theorem provides the conditions for the ZF equalization of MIMO-FIR channels [25].

THEOREM 5.2 (Equalizability of MIMO-FIR Channels)

Let $\mathbf{H}_{M\times N}(z)$ denote a MIMO-FIR channel, with $M \geq N$. Considering its associated Smith form given by (5.10), the channel is perfectly equalizable, i.e., the ZF condition is attainable by means of MIMO-FIR equalizers if, and only if, $\gamma_i(z) = z^{-d_i}$ for some $d_i \geq 0$. ■

Theorem 5.2 states that it is possible to obtain a ZF equalizer even if both channel and equalizer are FIR structures, a situation very distant from that found in the SISO case. Considering the definition of system zeros of a MIMO system, the condition under which it is possible to perfectly equalize it is equivalent to the nonexistence of finite zeros associated with the system [153].

It is important to mention that this condition evokes the idea of a trivial SISO channel, i.e., one that only imposes a delay to the input signal. However, it is interesting to imagine that for a MIMO-FIR structure, even if the sub-channels linking each transmitter-receiver are not trivial channels, the global system can be perfectly inverted by an FIR equalizer.

5.2 SIMO Channel Equalization

Let us now consider a particular case of the general MIMO model, which is illustrated in Figure 5.3. In this scenario, we have a SIMO channel model, in which a single signal is transmitted by multiple subchannels.

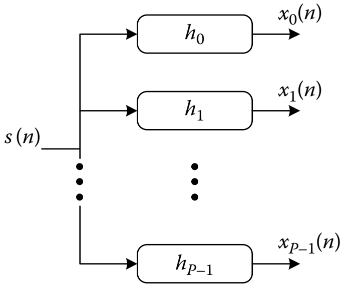


FIGURE 5.3
General SIMO model.

It should be noticed that since the same information is transmitted through different subchannels, all received sequences will be distinctly distorted versions of the same message, which accounts for a certain signal diversity. Therefore, it is reasonable to assume that more information about the transmitted signal will be available at the receiver end. It is like looking at the same landscape through different blurry windows: each sight reveals you a different detail that perhaps is not so clear in the others.

Interestingly enough, two practical scenarios can be formulated using the SIMO channel framework. The first one is illustrated in Figure 5.4a, in which we have a system with a single transmit antenna and several antennas at the receiver end. In such a case, each possible channel linking the transmit antenna to one of the receive antennas can be regarded as a subchannel, as indicated in Figure 5.4. The spatial diversity introduced by means of a set of antennas at the receiver also gives the system a certain degree of robustness: if one of these channels happens to severely degrade the signal, other received versions may still provide the necessary information to correctly estimate the original message.

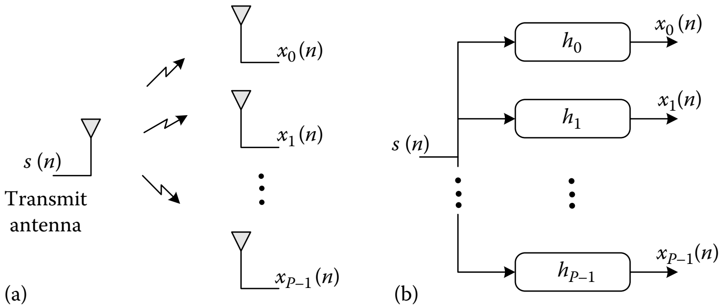


FIGURE 5.4
(a) A system with multiple receiving antennas and (b) its SIMO model.

The other context in which the SIMO channel model arises is related to the idea of *oversampling*, described in more detail in Section 5.2.1.

5.2.1 Oversampling and the SIMO Model

In the previous chapters, whenever we have expressed the received signal as the convolution of a symbol sequence and the channel impulse response $h(n)$, we have assumed that the received signal was sampled at the symbol rate, which is the case in several communication systems. However, if the sampling rate is made higher than the symbol rate, the received signal can also be represented in terms of the outputs of a SIMO channel model.

To explain this connection, let us first consider the received baseband signal, expressed by

$$x(t) = \sum_{i=-\infty}^{\infty} s(i)h(t - iT) + \nu(t), \tag{5.12}$$

where

$h(t)$ stands for the channel impulse response
$\nu(t)$ is the additive noise

In the usual case, when the signal is sampled at the symbol rate $\frac{1}{T}$, the received sequence will be given by

$$x(nT) = \sum_{i=-\infty}^{\infty} s(i)\, h((n - i)T) + \nu(nT). \tag{5.13}$$

If we sample the received signal at a rate P times higher than the symbol rate, i.e.,

$$\frac{1}{T_s} = P\frac{1}{T} \tag{5.14}$$

then the *oversampled* sequence $x(nT_s)$ will be given by

$$x(nT_s) = x\left(n\frac{T}{P}\right) = \sum_{i=-\infty}^{\infty} s(i)\, h\left(n\frac{T}{P} - iT\right) + \nu\left(n\frac{T}{P}\right). \tag{5.15}$$

The relationship between the oversampled sequence and the SIMO channel model is revealed when we consider sequences composed of samples of $x(t)$ spaced by one symbol period. For example, the samples $x(0), x(T), x(2T), \ldots$ represent the sequence $\{x(nT)\}$, while $x(T_s)$, $x(T_s + T)$,

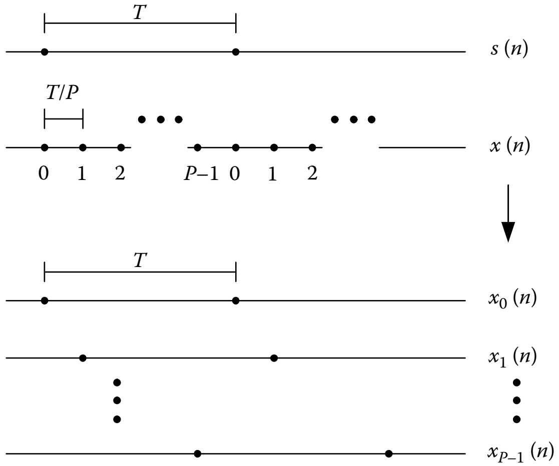


FIGURE 5.5
Relationship between the oversampled sequence and P sequences sampled at the baud rate.

$x\,(T_s + 2T), \ldots$ correspond to the sequence $\{x\,(nT + T_s)\}$, and so forth, as illustrated in Figure 5.5.

Therefore, the samples of the pth sequence $\left\{x_p\,(n)\right\}, p = 0, \ldots, P-1$, can be expressed as

$$x_p\,(nT) = x\left(nT + \frac{pT}{P}\right) = x\left(\left(nP + p\right)T_s\right). \tag{5.16}$$

and, from (5.13) and (5.16), we get to

$$x_p\,(nT) = \sum_i s\,(i)\,h\left(nT - iT + \frac{pT}{P}\right). \tag{5.17}$$

Thus, the pth sequence is the result of the convolution between the transmitted symbols and the impulse response of the corresponding subchannel, whose impulse response is given by

$$h_p\,(nT) = h\,(t)\Bigg|_{t=nT+\frac{p}{P}T}, \quad p = 0, \ldots, P-1, \tag{5.18}$$

and the pth received sequence $\{x_p(nT)\}$ can be rewritten as

$$x_p\left(kT\right) = \sum_i s\,(i)\,h_p\,(nT - iT). \tag{5.19}$$

As in the scenario with multiple antennas, we have a SIMO channel model representing the observed signals. In the oversampling approach, however, we explore the temporal diversity of the signals.

5.2.2 Cyclostationary Statistics of Oversampled Signals

In a certain sense, when we consider the SIMO channel model, we are implicitly exploring a particular characteristic of digitally modulated signals: they are wide-sense cyclostationary, i.e., as discussed in Section 2.4.4, their mean and autocorrelation functions are periodic [230]. Gardner [116, 117] first explored this characteristic, showing that it is possible to recover the channel phase information from the cyclic autocorrelation function, and, later, by Tong et al. [287–289], who proposed methods for blind channel identification and equalization.

Since the digitally modulated signal is cyclostationary, it is possible to show that

$$R_x\left(t+\frac{\tau}{2},t-\frac{\tau}{2}\right)=R_x\left(t+\frac{\tau}{2}+T,t-\frac{\tau}{2}+T\right) \tag{5.20}$$

which means that its autocorrelation function is periodic, with period T, for any value of τ. Hence, it can be expressed in terms of a Fourier series [7,118], whose coefficients are given by

$$n\text{th coefficient}=\frac{1}{T}\int_{-T/2}^{T/2} R_x\left(t+\frac{\tau}{2},t-\frac{\tau}{2}\right)\exp\left(-j2\pi\frac{n}{T}t\right)dt \tag{5.21}$$

The nth coefficient can be defined as a cyclic correlation function $R_x^{\alpha}(\tau)$, defined as

$$R_x^{\alpha}(\tau)=\frac{1}{T}\int_{-T/2}^{T/2} R_x\left(t+\frac{\tau}{2},t-\frac{\tau}{2}\right)\exp(-j2\pi\alpha t)dt \tag{5.22}$$

where $\alpha=\frac{n}{T}$ is known as the cyclic frequency.

It is interesting to notice that, for stationary signals that do not present a periodic autocorrelation, $R_x^{\alpha}(\tau)\equiv 0, \forall\alpha\neq 0$. On the other hand, for second-order cyclostationary signals—or simply cyclostationary signals—the cyclic correlation will be different from zero for some values of α.

In analogy with stationary signals, we can define a correlation spectral density $S_x^{\alpha}(f)$, as the Fourier transform of (5.22)

$$S_x^{\alpha}(f)=\int_{\tau=-\infty}^{\tau=\infty} R_x^{\alpha}(2\pi f\tau)\,d\tau \tag{5.23}$$

From (5.23), it is possible to show that [99]

$$S_x^{\alpha}(f) = H\left(f + \frac{\alpha}{2}\right) H^*\left(f - \frac{\alpha}{2}\right) \sigma_s^2 + \sigma_\nu^2 \delta(\alpha T) \tag{5.24}$$

where σ_s^2 and σ_ν^2 are, respectively, the transmitted signal and additive noise powers. Notice that for $\alpha = 0$, (5.24) is reduced to the power spectral density of $x(t)$:

$$S_x(f) = \sigma_s^2 \left|H(f)\right|^2 + \sigma_\nu^2 \tag{5.25}$$

It directly seems from (5.25) that the power spectral density does not provide any information about the phase of the channel transfer function. However, rewriting (5.24) for $\alpha = \pm 1/T, \pm 2/T, \ldots$, we get to

$$S_x^{\alpha}(f) = H\left(f + \frac{\alpha}{2}\right) H^*\left(f - \frac{\alpha}{2}\right) \sigma_s^2 \tag{5.26}$$

where it becomes clear that the correlation spectral density provides information about the phase of the channel transfer function. Therefore, it is possible to identify both the magnitude and phase responses of a channel based solely on second-order statistics of the oversampled received signal.

5.2.3 Representations of the SIMO Model

In order to discuss methods for blind equalization and identification in this chapter, it is convenient to represent the outputs of the SIMO system in a matrix form. Depending on how the output samples are organized in a vector, it is possible to obtain at least two different representations, discussed in the following.

5.2.3.1 Standard Representation

Let us assume that each subchannel is modeled by an L-tap filter, and let us define the vectors

$$\mathbf{h}_p = \begin{bmatrix} h_p(0) & h_p(1) & \cdots & h_p(L-1) \end{bmatrix}^T \tag{5.27}$$

$$\tilde{\mathbf{x}}_p(n) = \begin{bmatrix} x_p(n) & x_p(n-1) & \cdots & x_p(n-K+1) \end{bmatrix}^T \tag{5.28}$$

$$\tilde{\boldsymbol{\nu}}_p(n) = \begin{bmatrix} \nu_p(n) & \nu_p(n-1) & \cdots & \nu_p(n-K+1) \end{bmatrix}^T \tag{5.29}$$

representing the pth subchannel impulse response, a vector containing K samples of the corresponding output vector and a vector containing K samples of the additive noise, respectively. Using this notation and (5.19), and assuming the presence of noise, we can express the output vector related to the pth subchannel as

$$\tilde{\mathbf{x}}_p(n) = \tilde{\boldsymbol{\mathcal{H}}}_p \mathbf{s}(n) + \tilde{\mathbf{v}}_p(n), \tag{5.30}$$

with the convolution matrix associated with the pth subchannel, given by

$$\tilde{\boldsymbol{\mathcal{H}}}_p = \begin{bmatrix} h_p(0) & h_p(1) & \cdots & h_p(L-1) & 0 & \cdots & 0 \\ 0 & h_p(0) & \cdots & h_p(L-2) & h_p(L-1) & \cdots & 0 \\ & & \ddots & & & \ddots & \\ 0 & 0 & \cdots & h_p(0) & h_p(1) & \cdots & h_p(L-1) \end{bmatrix}. \tag{5.31}$$

Finally, stacking the P vectors corresponding to the subchannels, one can obtain the following:

$$\begin{bmatrix} \tilde{\mathbf{x}}_0(n) \\ \tilde{\mathbf{x}}_1(n) \\ \vdots \\ \tilde{\mathbf{x}}_{P-1}(n) \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\mathcal{H}}}_0 \\ \tilde{\boldsymbol{\mathcal{H}}}_1 \\ \vdots \\ \tilde{\boldsymbol{\mathcal{H}}}_{P-1} \end{bmatrix} s(n) + \begin{bmatrix} \tilde{\mathbf{v}}_0(n) \\ \tilde{\mathbf{v}}_1(n) \\ \vdots \\ \tilde{\mathbf{v}}_{P-1}(n) \end{bmatrix}, \tag{5.32}$$

or, simply,

$$\tilde{\mathbf{x}}(n) = \tilde{\boldsymbol{\mathcal{H}}} \mathbf{s}(n) + \tilde{\mathbf{v}}(n). \tag{5.33}$$

5.2.3.2 Representation via the Sylvester Matrix

Another possible way to represent the SIMO model is obtained by arranging the samples in a different manner. Let us define the vector

$$\mathbf{h}(n) = \begin{bmatrix} h_0(n) & h_1(n) & \cdots & h_{P-1}(n) \end{bmatrix}^T \tag{5.34}$$

containing the samples of the impulse response of all subchannels at a given time instant n. In a similar fashion, let us define the vectors associated with the outputs and the noise as follows:

$$\mathbf{x}(n) = \begin{bmatrix} x_0(n) & x_1(n) & \cdots & x_{P-1}(n) \end{bmatrix}^T \tag{5.35}$$

$$\mathbf{v}(n) = \begin{bmatrix} \nu_0(n) & \nu_1(n) & \cdots & \nu_{P-1}(n) \end{bmatrix}^T \tag{5.36}$$

Then, we can express the subchannel outputs as

$$\begin{bmatrix} x_0(n) \\ \vdots \\ x_{P-1}(n) \end{bmatrix} = \begin{bmatrix} h_0(0) & \dots & h_0(L-1) \\ \vdots & & \vdots \\ h_{P-1}(0) & \dots & h_{P-1}(L-1) \end{bmatrix} \begin{bmatrix} s(n) \\ \vdots \\ s(n-L+1) \end{bmatrix} + \begin{bmatrix} \nu_0(n) \\ \vdots \\ \nu_{P-1}(n) \end{bmatrix} \tag{5.37}$$

or, in compact form,

$$\mathbf{x}(n) = \mathbf{Hs}(n) + \boldsymbol{\nu}(n) \tag{5.38}$$

Notice that the matrix $\mathbf{H}$ can also be written as

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} h_0(0) & \dots & h_0(N-1) \\ \vdots & & \vdots \\ h_{P-1}(0) & \dots & h_{P-1}(N-1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{h}_0^T \\ \vdots \\ \mathbf{h}_{P-1}^T \end{bmatrix} \end{aligned} \tag{5.39}$$

Stacking K vectors containing the received signals and noise samples in vectors $\mathbf{x}(n)$ and $\mathbf{v}(n)$, in the form

$$\mathbb{x}(n) = \begin{bmatrix} \mathbf{x}(n)^T & \cdots & \mathbf{x}(n-K+1)^T \end{bmatrix}^T \tag{5.40}$$

$$\mathbb{v}(n) = \begin{bmatrix} \boldsymbol{\nu}(n)^T & \cdots & \boldsymbol{\nu}(n-K+1)^T \end{bmatrix}^T \tag{5.41}$$

$$\mathbb{s}(n) = \begin{bmatrix} \mathbf{s}(n)^T & \cdots & \mathbf{s}(n-L-K+2)^T \end{bmatrix}^T \tag{5.42}$$

we get to the following matrix equation:

$$\mathbb{x}(n) = \mathbb{H}\mathbb{s}(n) + \mathbb{v}(n) \tag{5.43}$$

where $\mathbb{H}$ is defined as

$$\mathbb{H} = \begin{bmatrix} \mathbf{h}(0) & \mathbf{h}(1) & \cdots & \mathbf{h}(N-1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}(0) & \cdots & \mathbf{h}(N-2) & \mathbf{h}(N-1) & & \mathbf{0} \\ \vdots & & \ddots & & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}(0) & \mathbf{h}(1) & \cdots & \mathbf{h}(N-1) \end{bmatrix}. \tag{5.44}$$

5.2.4 Fractionally Spaced Equalizers and the MISO Equalizer Model

Let us assume the equalizer to be an FIR filter, composed of KP coefficients, i.e.,

$$\mathbf{w} = \begin{bmatrix} w(0) & w(T_s) & w(2T_s) & \ldots & w(T) & \ldots & w((KP-1)T_s) \end{bmatrix}^T. \tag{5.45}$$

The difference in this case is that two adjacent elements are related to samples delayed by a fraction of the symbol interval, the reason why this is called an *FS equalizer*. Its output is given by

$$y\left(nT_s\right) = \sum_{l=0}^{KP-1} w\left(lT_s\right) x\left(nT_s - lT_s\right). \tag{5.46}$$

It should be noted that the output is also oversampled, i.e., its samples are spaced by a fraction of the symbol period. Therefore, to obtain a sequence of T-spaced recovered symbols one should decimate the sequence by the same oversampling factor P. This decimation factor originates P different baud-spaced sequences

$$y_i\left(nT\right) = y\left(nT + \frac{iT}{P}\right), \quad i = 0, \ldots, P-1 \tag{5.47}$$

and each one, in accordance with (5.46), can be expressed as

$$y_i\left(nT\right) = \sum_{l=0}^{LP-1} w\left(\frac{lT}{P}\right) x\left(nT + \frac{iT}{P} - \frac{lT}{P}\right). \tag{5.48}$$

Notice that the coefficients $w\left(lT_s\right)$ can also be arranged into P baud-spaced sequences, each one related to a subequalizer, i.e.,

$$w_k\left(lT\right) = w\left(lT + \frac{k}{P}T\right) = w\left(\left(lP + k\right)T_s\right) \quad \begin{matrix} l = 0, \ldots, K-1 \\ k = 0, \ldots, P-1 \end{matrix} \tag{5.49}$$

Replacing (5.49) in (5.46), we get

$$\begin{aligned} y_i(nT) &= \sum_{l=0}^{L-1}\sum_{k=0}^{P-1} w_k(lT)\, u\left(nT + i\frac{T}{P} - \left(lT + k\frac{T}{P}\right)\right) \\ &= \sum_{l=0}^{L-1}\sum_{k=0}^{P-1} w_k(lT)\, u\left(\left(n-l\right)T + \left(i-k\right)T_s\right). \end{aligned} \tag{5.50}$$

Assuming that we are interested in the sequence $\left\{y_{P-1}(nT)\right\}$, we have

$$y(nT) = y_{P-1}(nT) = \sum_{k=0}^{P-1}\sum_{l=0}^{K-1} w_k(lT)\, x\left(\left(n-l+1\right)T - \left(k+1\right)T_s\right) \tag{5.51}$$

which can be understood as the sum of the outputs of all P subequalizers.

A diagram illustrating the channel and equalizer structure is provided in Figure 5.6.

Using the definition in (5.51), the output is given in matrix form by

$$y(n) = \mathbb{w}^H \mathbb{x}(n) \tag{5.52}$$

where

$$\mathbb{w} = \left[\, \mathbf{w}^H(0) \quad \mathbf{w}^H(1) \quad \cdots \quad \mathbf{w}^H(L-1) \,\right]^H \tag{5.53}$$

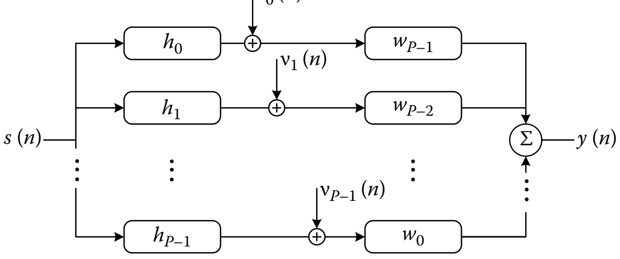


FIGURE 5.6
Multichannel model and subequalizers.

and

$$\mathbf{w}(n) = \begin{bmatrix} w_{P-1}(n) \\ w_{P-2}(n) \\ \vdots \\ w_0(n) \end{bmatrix}, \quad n = 0, \ldots, K-1 \tag{5.54}$$

Using this formulation, obtaining the Wiener solution is a process similar to that developed in Section 3.2. We can express the Wiener–Hopf equations (3.13) for the multichannel model as

$$\mathbf{R}_{\mathbb{x}} \mathbb{w}_{opt} = \mathbf{p} \Rightarrow \mathbb{w}_{opt} = (\mathbf{R}_{\mathbb{x}})^{-1} \mathbf{p} \tag{5.55}$$

where the autocorrelation matrix is given by

$$\mathbf{R}_{\mathbb{x}} = E\left[\mathbb{x}(n)\, \mathbb{x}^H(n)\right]$$

and the cross-correlation vector between the desired and the received signals is

$$\mathbf{p} = E\left[\mathbb{x}(n)\, s^*\left(n-d\right)\right]$$

5.2.5 Bezout's Identity and the Zero-Forcing Criterion

The condition stated in Theorem 5.2 is also valid for this particular case of a SIMO channel, thus providing a suitable condition for ZF equalization. Let the polynomial vector associated with the channel be defined as

$$\mathbf{h}(z) = \sum_{i=0}^{L-1} \mathbf{h}(i)\, z^{-i} = \begin{bmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_{P-1}(z) \end{bmatrix} \tag{5.56}$$

where $\mathbf{h}(i)$ is given in (5.34), and the polynomial vector related to the equalizer is

$$\mathbf{w}(z) = \sum_{i=0}^{K-1} \mathbf{w}(i)\, z^{-i} = \begin{bmatrix} W_{P-1}(z) \\ W_{P-2}(z) \\ \vdots \\ W_0(z) \end{bmatrix} \tag{5.57}$$

where $\mathbf{w}(n)$ is defined in (5.54).

Then, in the spirit of (5.10), the polynomial vector can be rewritten in terms of its Smith form as

$$\mathbf{h}(z) = \mathbf{Q}(z) \begin{bmatrix} \gamma(z) \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{5.58}$$

Hence, the condition for ZF equalization is reduced to $\gamma(z) = z^{-d}$.

It is interesting to note that $\gamma(z)$ represents the greatest common divisor of all subchannels $h_i(z)$, which means that the roots of $\gamma(z)$ are the common zeros of all subchannels of the SIMO system. In other words, an FIR SIMO channel is perfectly equalized by another FIR structure if and only if the subchannels have no zeros in common. This result, even though explained here in terms of the Smith form of the channel, is also a consequence of *Bezout's identity* [163].

THEOREM 5.3 (Bezout's Identity)

Let the $\mathbf{h}(z)$ and $\mathbf{w}(z)$, defined in (5.56) and (5.57), denote the polynomial vectors associated with the channel and equalizer, respectively. If the polynomials $H_0(z), \ldots, H_{P-1}(z)$ do not share common zeros, then there exists a set of polynomials $W_0(z), \ldots, W_{P-1}(z)$ such that

$$\mathbf{w}^H(z)\mathbf{h}(z) = z^{-d} \tag{5.59}$$

where d is an arbitrary delay. ■

In the time domain, Bezout's identity implies

$$\mathbb{w}^H \mathbb{H} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{5.60}$$

where $\mathbb{H}_L$ is defined in (5.44). We can note that (5.60) is equivalent to a ZF condition. Also, from (5.60), it is possible to notice that we have $L+K-1$ equations and LP unknowns. In this system, $L+K-1$ is the number of positions in the combined channel+equalizer response and KP is the length of the equalizer. Since the number of unknowns should be at least equal to the number of equations, we find a first constraint on the length of the equalizer, given by [229]

$$KP \geq L+K-1 \Rightarrow K \geq \left\lceil \frac{L-1}{P-1} \right\rceil \tag{5.61}$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

5.3 Methods for Blind SIMO Equalization

Previous discussions confirm the possibility of carrying out blind equalization in multichannel scenarios by dealing only with second-order statistics. Nevertheless, it does not discard the possible interest in using higher-order techniques due, for instance, to their characteristics of robustness. This option is discussed next, and in the sequence we present the two main second-order methods, those of subspace decomposition and linear prediction.

5.3.1 Blind Equalization Based on Higher-Order Statistics

The idea of FS equalization was originally conceived as an alternative to reduce the sensitivity to sampling timing errors and noise amplification [184], for both supervised and blind equalizers. In the latter case, a special attention was paid to the CMA, which is commonly termed FS-CMA. However, the relationship between oversampling and the multichannel model opened a new perspective to the study of FS-CMA.

The main characteristic associated with the FS-CMA is the possibility of effective global convergence. In fact, in the absence of noise, if all subchannels share no common zeros and the condition about the length of the equalizer is respected, then all minima of the CM cost function correspond to ZF solutions [184]. In other words, in contrast with the baud-rate equalization case, the ZF condition is blindly attainable with finite length equalizers.

Different works contributed to the study of the behavior of the FS-CMA. In [184], the authors prove the global convergence behavior of FS-CMA under the aforementioned conditions. Moreover, a brief discussion is presented about the influence of noise, which violates the ideal conditions, in the minima location. The discussion about convergence is extended in [95], which analyzes the equilibrium points of cost functions of other Bussgang algorithms. Important results regarding the convergence analysis of the FS-CMA can be found in [155]. The relationship between equalizers obtained with the CM and Wiener criteria is investigated in [132,312].

5.3.2 Blind Equalization Based on Subspace Decomposition

Schemes based solely on second-order moments of the received signal have an advantage over cumulant-based methods with respect to computational complexity. The existing methods to solve the problem of SIMO blind equalization/identification based exclusively on second-order statistics can be grouped in two main classes: those based on subspace decomposition and those based on linear prediction. In the following, we will present the main characteristics associated with both approaches.

In simple terms, subspace methods rely on the decomposition of the autocorrelation matrix of the received signal [287, 289]. For instance, in [211], the main idea is to explore the orthogonality between signal and noise subspaces. The problem consists of estimating a vector $LP \times 1$ composed of all coefficients from all P subchannels, organized in the following manner:

$$\mathbf{h}_S = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_{P-1} \end{bmatrix}^T \tag{5.62}$$

Estimation is based on a set of K observations of the received signals, and is founded on the following theorem [139,288].

THEOREM 5.4

The convolution matrix $\tilde{\mathcal{H}}$ associated with the channel has full column rank if and only if the following conditions hold:

- The polynomials composed by the coefficients of each subchannel have no common zeros.
- At least one of the polynomials has maximum degree $L - 1$.
- The length K of the vector containing samples from each subchannel should be $K > L - 1$.

■

We assume that the input symbol vector $\mathbf{s}(n)$ and the noise vector $\tilde{\mathbf{v}}(n)$ present in (5.33) are obtained from mutually independent wide-sense stationary processes. The transmitted signal is zero mean, and its correlation matrix is defined as

$$\mathbf{R}_s = E\left[\mathbf{s}(n)\,\mathbf{s}^H(n)\right] \tag{5.63}$$

which has a full column rank. The vector of noise samples is zero mean, and has a correlation matrix

$$\mathbf{R}_{\mathcal{V}} = E\left[\tilde{\mathbf{v}}(n)\tilde{\mathbf{v}}(n)^H\right] = \sigma_v^2\mathbf{I} \tag{5.64}$$

with σ_v^2 being the noise variance, supposed to be known.

The identification process is based on the correlation matrix, defined by

$$\begin{aligned}
\mathbf{R}_{\tilde{\mathbf{x}}} &= E\left[\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^H(n)\right] \\
&= E\left[(\tilde{\mathcal{H}}\mathbf{s}(n) + \tilde{\mathbf{v}}(n))(\tilde{\mathcal{H}}\mathbf{s}(n) + \tilde{\mathbf{v}}(n))^H\right] \\
&= E\left[\tilde{\mathcal{H}}\mathbf{s}(n)\mathbf{s}(n)^H\tilde{\mathcal{H}}^H\right] + E\left[\tilde{\mathbf{v}}(n)\tilde{\mathbf{v}}(n)^H\right] \\
&= \tilde{\mathcal{H}}\mathbf{R}_s\tilde{\mathcal{H}}^H + \mathbf{R}_{\mathcal{V}}
\end{aligned} \tag{5.65}$$

The matrix $\mathbf{R}_{\tilde{\mathbf{x}}}$ can be rewritten in terms of its eigenvectors

$$\mathbf{R}_{\tilde{\mathbf{x}}} = \sum_{i=0}^{KP-1} \lambda_i q_i q_i^H \tag{5.66}$$

where λ_i are the eigenvalues, in a decreasing order: $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{KP-1}$.

Similarly, the space spanned by the eigenvectors of $\mathbf{R}_{\tilde{\mathbf{x}}}$ can be separated into two subspaces:

- The signal subspace $\mathcal{S}$, spanned by the eigenvectors associated with $\lambda_0, \lambda_1, \ldots, \lambda_{L+K-2}$, which are given by $q_{signal_i} = q_i, i = 0, \ldots, L+K-2$.
- The noise subspace $\mathcal{L}$, spanned by the eigenvectors associated with $\lambda_{L+K-1}, \ldots, \lambda_{KP-1}$, which are given by $q_{noise_i} = q_{L+K+i-1}$, $i = 0, \ldots, KP-L-K$.

The noise subspace is the orthogonal complement of the signal subspace. The signal subspace is also spanned by the columns of the convolution matrix $\tilde{\mathcal{H}}$. Hence, the columns of $\tilde{\mathcal{H}}$ are orthogonal to every vector in the noise subspace, which implies

$$\tilde{\mathcal{H}}^H \mathbf{q}_{noise_i} = \mathbf{0}, \quad i = 0, \ldots, KP-L-K \tag{5.67}$$

The previous result can also be verified in a different way. By definition

$$\mathbf{R}_{\tilde{\mathbf{x}}}\mathbf{q}_{noise_i} = \sigma_v^2 \mathbf{q}_{noise_i}, \quad i = 0, \ldots, KP-L-K \tag{5.68}$$

Substituting (5.65) into (5.68) and making $\mathbf{R}_{\tilde{\mathbf{v}}} = \sigma_{\tilde{\mathbf{v}}}^2 \mathbf{I}$ yields

$$\tilde{\mathcal{H}} \mathbf{R}_{\mathbf{s}} \tilde{\mathcal{H}}^H \mathbf{q}_{noise_i} = \mathbf{0}, \quad i = 0, \ldots, KP - L - K \tag{5.69}$$

Since both $\mathbf{R}_{\mathbf{s}}$ and $\tilde{\mathcal{H}}$ have full column rank, (5.69) implies (5.67). Equation 5.67 forms the basis of the subspace decomposition method proposed by Moulines et al. [211], based upon the following premises:

- Knowledge about the eigenvectors associated with the $KP-L-K+1$ lowest eigenvalues of the correlation matrix $\mathbf{R}_{\tilde{\mathbf{x}}}$
- Orthogonality between the columns of the channel convolution matrix $\tilde{\mathcal{H}}$ (unknown) and the noise subspace

In order to obtain the criterion itself, the orthogonality principle expressed in (5.67) is first rewritten in its scalar form:

$$\left\| \tilde{\mathcal{H}}^H \mathbf{q}_{noise_i} \right\|^2 = \mathbf{q}_{noise_i}^H \tilde{\mathcal{H}} \tilde{\mathcal{H}}^H \mathbf{q}_{noise_i} = 0, \quad i = 0, \ldots, KP - L - K \tag{5.70}$$

Then, making an analogy with the "modular" structure of $\tilde{\mathcal{H}}_M$, the eigenvector $KP \times 1$ $\mathbf{q}_{noise_i}$ can be partitioned as follows:

$$\mathbf{q}_{noise_i} = \begin{bmatrix} \mathbf{q}_{noise_i,0}^T & \mathbf{q}_{noise_i,1}^T & \cdots & \mathbf{q}_{noise_i,P-1}^T \end{bmatrix}^T \tag{5.71}$$

where $\mathbf{q}_{noise_i,p}$, $p = 0, \ldots, P-1$ is a vector $K \times 1$. Based on the same structure of the convolution matrix of each subchannel $\tilde{\mathcal{H}}_p$, we define the following matrix $L \times L + K - 1$ associated with $\mathbf{q}_{noise_i,p}$:

$$\mathcal{Q}_{noise_i,p} = \begin{bmatrix} \mathbf{q}_{noise_i,p}(0) & \cdots & \mathbf{q}_{noise_i,p}(K-1) & & \mathbf{0} & \\ & \ddots & & & \ddots & \\ \mathbf{0} & & \mathbf{q}_{noise_i,p}(0) & & \cdots & \mathbf{q}_{noise_i,p}(K-1) \end{bmatrix} \tag{5.72}$$

and we can also define a matrix $LP \times L + K - 1$ $\mathcal{Q}_{noise_i}$:

$$\mathcal{Q}_{noise_i} = \begin{bmatrix} \mathcal{Q}_{noise_i,0} \\ \mathcal{Q}_{noise_i,1} \\ \vdots \\ \mathcal{Q}_{noise_i,P-1} \end{bmatrix}, \quad i = 0, \ldots, KP - L - K \tag{5.73}$$

Finally, it can be shown [211] that

$$\mathbf{q}_{noise_i}^H \tilde{\boldsymbol{\mathcal{H}}}\tilde{\boldsymbol{\mathcal{H}}}^H \mathbf{q}_{noise_i} = \mathbf{h}_S^H \boldsymbol{\mathcal{Q}}_{noise_i} \boldsymbol{\mathcal{Q}}_{noise_i}^H \mathbf{h}_S \tag{5.74}$$

Therefore, the orthogonality relation (5.70) can also be expressed as

$$\mathbf{h}_S^H \boldsymbol{\mathcal{Q}}_{noise_i} \boldsymbol{\mathcal{Q}}_{noise_i}^H \mathbf{h}_S = 0, \quad i = 0, \ldots, KP - L - K \tag{5.75}$$

In practice, only estimates $\hat{\mathbf{q}}_{noise_i}$ of the eigenvectors associated with the noise subspace are required. Then, the estimate of the channel coefficient vector $\mathbf{h}_S$ is obtained minimizing the following quadratic form:

$$J_{quad}\left(\mathbf{h}_S\right) = \sum_{i=0}^{KP-L-K} \left\| \tilde{\boldsymbol{\mathcal{H}}}^H \hat{\mathbf{q}}_{noise_i} \right\|^2 = \mathbf{h}_S^H \boldsymbol{\mathcal{Q}}_{noise} \mathbf{h}_S \tag{5.76}$$

where $\boldsymbol{\mathcal{Q}}_{noise}$ is a $LP \times LP$ matrix given by

$$\boldsymbol{\mathcal{Q}}_{noise} = \sum_{i=0}^{KP-L-K} \hat{\boldsymbol{\mathcal{Q}}}_{noise_i} \hat{\boldsymbol{\mathcal{Q}}}_{noise_i}^H \tag{5.77}$$

and matrix $\hat{\boldsymbol{\mathcal{Q}}}_{noise_i}$ is defined by (5.71) through (5.73), replacing the eigenvectors associated with the noise subspace with their estimates.

The minimization must be subject to appropriate constraints, in order to avoid the trivial solution $\mathbf{h}_S = 0$. In [211], the following criteria are suggested.

- Quadratic constraint: minimize $J_{quad}(\mathbf{h}_S)$ subject to $\|\mathbf{h}_S\| = 1$. The optimal solution is given by the unit-norm eigenvector associated with the smallest eigenvalue of $\boldsymbol{\mathcal{Q}}_{noise}$.
- Linear constraint: minimize $J_{quad}(\mathbf{h}_S)$ subject to $l^H \tilde{\boldsymbol{\mathcal{H}}} = 1$, where l is a $LP \times 1$ vector.

The first criterion can be considered to be the natural choice, although it has a higher computational complexity due to the eigenvector estimation. The second criterion, in spite of demanding a lower computational burden, depends on the appropriate choice of an arbitrary vector $\boldsymbol{l}$—the solution is proportional to $\boldsymbol{\mathcal{Q}}^{-1}\boldsymbol{l}$.

The quadratic form (5.76) can also be expressed in terms of the eigenvectors associated with the signal subspace:

$$
\begin{aligned}
J_{quad}\left(\mathbf{h}_S\right) &= K\left\|\mathbf{h}_S\right\|^2 - \sum_{i=0}^{K+L-2}\left\|\tilde{\mathcal{H}}_K^H \hat{\mathbf{q}}_{signal_i}\right\|^2 \\
&= K\left\|\mathbf{h}_S\right\|^2 - \mathbf{h}_S^H\left(\sum_{i=0}^{K+L-2}\hat{\mathcal{Q}}_{signal_i}\hat{\mathcal{Q}}_{signal_i}^H\right)\mathbf{h}_S \\
&= K\left\|\mathbf{h}_S\right\|^2 - \mathbf{h}_S^H\mathcal{Q}_{signal}\mathbf{h}_S
\end{aligned}
\tag{5.78}
$$

where

$$
\mathcal{Q}_{signal_i,p} = \begin{bmatrix}
\mathbf{q}_{signal_i,p}(0) & \cdots & \mathbf{q}_{signal_i,p}(K-1) & & \mathbf{0} \\
 & \ddots & & \ddots & \\
\mathbf{0} & & \mathbf{q}_{signal_i,p}(0) & \cdots & \mathbf{q}_{signal_i,p}(K-1)
\end{bmatrix}
\tag{5.79}
$$

is the convolution matrix associated with the signal subspace. The minimization of (5.78) subject to $\left\|\mathbf{h}_S\right\| = 1$ is equivalent to the maximization of

$$
\tilde{J}_{quad}\left(\mathbf{h}_S\right) = \mathbf{h}_S^H\mathcal{Q}_{signal}\mathbf{h}_S \quad \text{subject to } \left\|\mathbf{h}_S\right\| = 1
\tag{5.80}
$$

For the quadratic norm constraint, both forms (5.76) and (5.80) provide the same solution. However, the evaluation of (5.80) involves $K+L-1$ terms, while the calculation of (5.76) involves $KP-L-K+1$ terms, which is favorable to the method based on the signal subspace. The main drawback pointed out in [211] relates to the case in which the subchannels present zeros very close to each other, a condition that compromises all premises on which this method is founded.

5.3.3 Blind Equalization Based on Linear Prediction

We revisit here the problem discussed in Chapter 3, in order to introduce the notation used in the presentation of an equalization method for the SIMO case. The idea is to obtain an estimate for the received vector $\mathbf{x}(n)$ as a linear combination of the vectors $\mathbf{x}(n-1) \quad \ldots \quad \mathbf{x}(n-K+1)$, i.e., the components of $\mathbb{x}(n-1)$. The estimate can be expressed as

$$
\begin{aligned}
\hat{\mathbf{x}}(n) &= \mathbf{A}^H(1)\mathbf{x}(n-1) + \cdots + \mathbf{A}^H(K-1)\mathbf{x}(k-K+1) \\
&= \mathbf{A}^H\mathbb{x}\left(n-1\right)
\end{aligned}
\tag{5.81}
$$

where **A** is a $P(K-1) \times P$ matrix composed by the $(K-1)$ $P \times P$ matrices of prediction coefficients:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^H(1) & \cdots & \mathbf{A}^H(K-1) \end{bmatrix}^H \tag{5.82}$$

The forward prediction error is then given by

$$\mathbf{e}_f(n)\Big|_{\mathbb{x}(n-1)} = \mathbf{x}(n) - \hat{\mathbf{x}}(n)\Big|_{\mathbb{x}(n-1)} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^H \end{bmatrix} \mathbb{x}(n) \tag{5.83}$$

The operation of a forward linear multichannel predictor is illustrated in Figure 5.7.

A $P \times P$ matrix with the forward prediction-error variance is defined by

$$\sigma^2_{\mathbf{e}_f} = E\left[\mathbf{e}_f(n)\,\mathbf{e}_f^H(n)\right] = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^H \end{bmatrix} \mathbf{R}_{\mathbb{x}}^{(K)}(n) \begin{bmatrix} \mathbf{I} & -\mathbf{A}^H \end{bmatrix}^H \tag{5.84}$$

where

$$\mathbf{R}_{\mathbb{x}}^{(K)}(n) = E\left[\mathbb{x}(n)\,\mathbb{x}(n)^H\right] \tag{5.85}$$

and K indicates the number of time instances taken into account. The minimization of the variance of the prediction error leads to the following optimization problem:

$$\min_{\mathbf{A}} \begin{bmatrix} \mathbf{I}_P & -\mathbf{A}^H \end{bmatrix} \mathbf{R}_{\mathbb{x}}^{(K)}(n) \begin{bmatrix} \mathbf{I}_P & -\mathbf{A}^H \end{bmatrix}^H = \sigma^2_{e_f} \tag{5.86}$$

and results, according to the Yule–Walker equations, in

$$\begin{bmatrix} \mathbf{I}_P & -\mathbf{A}^H \end{bmatrix} \mathbf{R}_{\mathbb{x}}^{(K)}(n) = \begin{bmatrix} \sigma^2_{e_f} & \mathbf{0}_P & \cdots & \mathbf{0}_P \end{bmatrix} \tag{5.87}$$

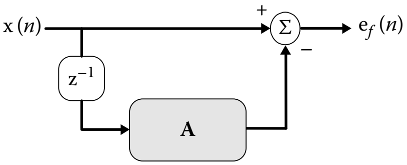


FIGURE 5.7
Structure of a forward linear predictor.

Since the correlation matrix presents a Toeplitz structure, it can be partitioned into

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A}^H \end{bmatrix} \cdot \begin{bmatrix} \mathbf{r}_0 & \mathbf{r} \\ \mathbf{r}^H & \mathbf{R}_{\mathbb{X}}^{(K-1)}(n) \end{bmatrix} = \begin{bmatrix} \sigma_{e_f}^2 & \mathbf{0}_P & \cdots & \mathbf{0}_P \end{bmatrix} \tag{5.88}$$

Finally, we get to the equations that allow us to obtain the prediction coefficients and the prediction-error variance based on second-order statistics of the received signal:

$$\begin{cases} \sigma_{e_f}^2 = \mathbf{r}_0 - \mathbf{r}\left(\mathbf{R}_{\mathbb{X}}^{(K-1)}(n)\right)^{-1}\mathbf{r}^H \\ \mathbf{A}_{L-1} = \left(\mathbf{R}_{\mathbb{X}}^{(K-1)}(n)\right)^{-1}\mathbf{r}^H \end{cases} \tag{5.89}$$

From this, it is possible to obtain an expression for the ZF equalizer with zero equalization delay. The development here assumes an ideal noiseless case. Let us rewrite Equation 5.43 for the sake of convenience:

$$\mathbb{x}(n) = \mathbb{H}\mathbb{s}(n) \tag{5.90}$$

The estimation of the prediction error in terms of $\mathbb{x}(n-1)$ is now done in terms of $\mathbf{s}(n-1)$:

$$\begin{aligned} \mathbf{e}_f(n)\big|_{\mathbb{x}(n-1)} &= \mathbf{e}_f(n)\big|_{\mathbf{s}(n-1)} \\ &= \mathbf{x}(n) - \hat{\mathbf{x}}(n)\big|_{\mathbf{s}(n-1)} \\ &= \sum_{i=0}^{L-1}\mathbf{h}(i)s(n-i) - \sum_{i=0}^{L-1}\mathbf{h}(i)\hat{s}(n-i)\Big|_{\mathbf{s}(n-1)} \\ &= \mathbf{h}(0)\tilde{s}(n)\big|_{\mathbf{s}(n-1)} \end{aligned} \tag{5.91}$$

with $\tilde{s}(n)$ representing the prediction error that arises when $s(n)$ is estimated with samples of the vector $\mathbf{s}(n-1)$.

From Equation 5.91, if the transmitted signal is assumed to be composed of i.i.d. samples (or, at least, uncorrelated), $\tilde{s}(n)$ will correspond to $s(n)$, i.e., the prediction-error filter represents a ZF equalizer for a null delay.

Such a result consolidates the possibility of using second-order statistics in SIMO channels blind equalization, as linear prediction is essentially a second-order method. Also, it allows the use of adaptive techniques, which is perfectly compatible with predictive configurations. In fact, as far as the SIMO case is concerned, both second-order and higher-order adaptive methods are presented in the literature. The first ones have all the already mentioned advantages of dealing with a friendlier framework, while the second ones are disposed to robustness, for instance in regard to the indeterminacy of the equalization delays.

As far as the theoretical principles of multichannel equalization are concerned, the extension from SIMO to MIMO case does not present significant novelties, although notation and mathematical formalism may become more intricate. So, in the following discussion, we emphasize a scenario that is particularly typical in MIMO communications and especially important in modern wireless systems, that one of multiuser processing.

5.4 MIMO Channels and Multiuser Processing

In a wireless communications, sets of transmitting and receiving antennas take place together in a same environment, which characterize a MIMO scenario. In a given station, the received signals may be subject to spatial interference due to other signals as well as to temporal distortion caused by the system convolutive character of the channel. Such temporal dispersion leads to the well-known effect of intersymbol interference (ISI), while the spatial dispersion is usually called multiple access interference (MAI). The methods for removing these interferences in an unsupervised fashion typically make use of some assumptions [295]:

1. The transmitted signals $s_i(n)$, $i = 1, \ldots, N$ are mutually independent and i.i.d.
2. The channel is modeled as a linear MIMO system and there are at least as many antennas as receiving signals.
3. The noise is a zero-mean, ergodic, and stationary Gaussian sequence independent of $s_i(n)$.

Figure 5.8 illustrates a general MIMO equalization scheme.

In this case, we process and recover all transmitted signals at the same time, and the receiver output can be expressed as

$$\begin{aligned}\mathbf{y}(n) &= \mathbf{W}(n) * \mathbf{x}(n) = \mathbf{W}(n) * \mathbf{H}(n) * \mathbf{s}(n) + \mathbf{v}'(n)\\ &= \mathbf{G}(n) * \mathbf{s}(n) + \mathbf{v}'(n),\end{aligned} \tag{5.92}$$

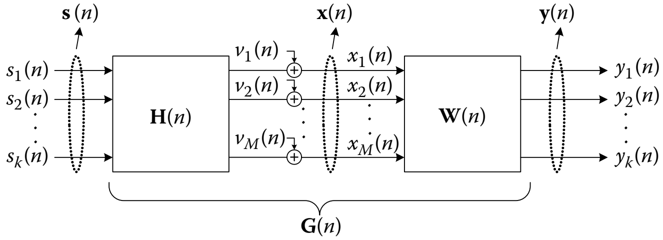


FIGURE 5.8
General MIMO equalization scheme.

where $\mathbf{W}(n)$ and $\mathbf{H}(n)$ denote, respectively, the MIMO equalizer and channel impulse responses, and

$$\mathbf{G}(n) = \mathbf{W}(n) * \mathbf{H}(n) \tag{5.93}$$

is the combined channel + equalizer matrix and $\boldsymbol{\nu}'(n) = \mathbf{W}(n) * \boldsymbol{\nu}(n)$ is the filtered noise.

It is also possible to process the signals in separate, by means of MISO filters. Each filter, in this case, aims to recover a single signal, so that the MIMO system is in fact reduced to a collection of MISO ones. Despite the interest of such an approach, which is usually refereed to as deflation [70,89] in the related literature, this chapter focuses in the MIMO filter solution.

When we use unsupervised MIMO equalization to recover the signals transmitted by several users, an additional difficulty may arise from the fact that all of them belong to the same finite alphabet and present the same statistical distribution. Due to the characteristics of wireless propagation, it follows that in the receiver that is closer users experience a higher signal power when compared to the more distant ones. This phenomenon is known as near–far effect [65,187,236]. This may lead to recovering errors, as a blind algorithm may "view" the higher power signal as the desired one, while the rest tends to be considered as interference. This would lead to the recovery of multiple copies of this only signal [228]. In order to deal with this limitation, it is necessary to employ additional strategies over the criterion for unsupervised MIMO equalization. In the sequel, we discuss two main families that have been adopted in most of the works of the literature.

5.4.1 Multiuser Detection Methods Based on Decorrelation Criteria

The objective of multiuser detection (MUD) methods based on decorrelation criteria is to force the estimates of different users to be mutually decorrelated as possible at the output of the MIMO equalizer. This aims to ensure that the equalizer will provide a correct estimate of all source signals. The key point here is how to force the decorrelation in a general adaptive procedure. We will discuss this subject in the sequel.

5.4.1.1 The Multiuser Constant Modulus Algorithm

The multiuser constant modulus algorithm (MU-CMA) was proposed in [228] as a generalization of the classical CMA to the multiuser case. In order to cope with the problem posed by the near–far effect, the authors combine the CMA with an auxiliary criterion based on the decorrelation of the estimates. Such technique allows removing MAI and ISI, avoiding the replication of source signals. So, the optimization process operates jointly on the decorrelation of the several equalizer outputs and on the minimization of the CM cost function.

A possible MU-CMA cost function for the *k*th user in the purely spatial case, is given by

$$J_{\text{MU-CMA}}\left(\mathbf{w}_k\right) = J_{\text{CMA}}\left(\mathbf{w}_k\right) + \gamma \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left|r_{ij}\right|^2, \tag{5.94}$$

where

$$r_{ij} = E\left\{y_i(n) y_j^*(n)\right\} \tag{5.95}$$

is the cross-correlation between the *i*th and *j*th equalizer outputs, γ is a decorrelation factor, and $J_{\text{CMA}}\left(\mathbf{w}_k\right)$ is the CMA cost function defined in Section 4.3.

Calculating the gradient of (5.94), we obtain

$$\nabla J_{\text{MU-CMA}}\left(\mathbf{w}_k\right) = E\left\{y_k(n)\left[\left|y_k(n)\right|^2 - 1\right]\mathbf{x}^*(n)\right\} + \gamma \sum_{\substack{i=1 \\ i \neq k}}^{N} r_{ik} E\left\{y_i(n)\mathbf{x}^*(n)\right\}, \tag{5.96}$$

where we assume a normalized (unit) power for the signals.

We must note that r_{ij} and $E\left\{y_i(n)\mathbf{x}^*(n)\right\}$ need to be estimated, and this can be performed with the aid of temporal averages [139]. So the technique is complemented by the following procedure [67]:

$$\widehat{\mathbf{R}}_y(n+1) = \varsigma \widehat{\mathbf{R}}_y(n) + (1-\varsigma)\mathbf{y}(n)\mathbf{y}^H(n) \tag{5.97a}$$

$$\widehat{\mathbf{P}}(n+1) = \varsigma \widehat{\mathbf{P}}(n) + (1-\varsigma)\mathbf{x}^*(n)\mathbf{y}^T(n), \tag{5.97b}$$

where $\mathbf{y}(n) = \begin{bmatrix} y_1(n) & \cdots & y_N(n) \end{bmatrix}^T$ and $\varsigma < 1$ is a smoothing factor.

Accordingly, the adaptation for spatial processing is given by the following expression:

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \mu\left(1 - \left|y_k(n)\right|^2\right) y_k(n)\mathbf{x}^*(n) - \gamma \sum_{\substack{i=1\\ i\neq k}}^{N} \widehat{r}_{ik}(n)\widehat{\mathbf{p}}_i(n), \quad (5.98)$$

where
$\widehat{r}_{ik}(n)$ is the (i,k)th element of the matrix $\widehat{\mathbf{R}}_y(n)$
$\widehat{\mathbf{p}}_i(n)$ is the ith column of matrix $\widehat{\mathbf{P}}(n)$, given in (5.97)

To take the effect of ISI into account, we have to modify the equations in order that the decorrelation term comprises the different time instants and mitigate the replication of a sequence of the same source signal with different delays. We can then write the following:

$$J_{\text{MU-CMA}}\left(\mathcal{W}_k\right) = J_{\text{CMA}}\left(\mathcal{W}_k\right) + \gamma \sum_{i=1}^{K} \sum_{\substack{j=1\\ j\neq i}}^{K} \sum_{\ell=-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \left|r_{ij}\left(\ell\right)\right|^2, \quad (5.99)$$

where

$$r_{ij}(\ell) = E\left\{y_i(n)y_j(n-\ell)\right\} \quad (5.100)$$

is the cross-correlation between the signals from the ith and jth outputs of the space-time MIMO equalizer with lag ℓ, and $\frac{\Delta}{2}$ is the maximum estimated delay for which the signals associated with multiple users must be uncorrelated.

Making use of the points of contact between spatial and space-time processing, we can write

$$\mathcal{W}_k(n+1) = \mathcal{W}_k(n) + \mu\left(1 - \left|y_k(n)\right|^2\right) y_k(n)\mathcal{X}(n) - \gamma \sum_{\substack{i=1\\ i\neq k}}^{K} \sum_{\ell=-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \widehat{r}_{ik,\ell}(n)\widehat{\mathbf{p}}_{i,\ell}(n), \quad (5.101)$$

and

$$\mathbf{R}_{y,\ell}(n+1) = \varsigma\mathbf{R}_{y,\ell}(n) + (1-\varsigma)\mathbf{y}(n)\mathbf{y}^T(n-\ell) \quad (5.102a)$$

$$\mathbf{P}_{\ell}(n+1) = \varsigma\mathbf{P}_{\ell}(n) + (1-\varsigma)\mathcal{X}(n)\mathbf{y}^T(n-\ell) \quad (5.102b)$$

$$\mathbf{y}(n-\ell) = \begin{bmatrix} y_1(n-\ell) & \cdots & y_K(n-\ell) \end{bmatrix}^T \tag{5.102c}$$

$$\ell = -\frac{\Delta}{2}, \ldots, \frac{\Delta}{2}, \tag{5.102d}$$

where $\widehat{r}_{ik,\ell}(n)$ is the cross-correlation between the ith and jth outputs with delay equal to ℓ, which corresponds to the (i,j)th element of the matrix $\mathbf{R}_{y,\ell}(n)$, and $\widehat{\mathbf{p}}_{i,\ell}(n)$ is the ith column of the matrix $\mathbf{P}_\ell(n)$.

In terms of performance, the MU-CMA suffers from some aspects that are related to the structure of the CMA, as discussed in [139, 223]. For instance we may mention

- Relatively low convergence speed.
- The regularization factor must be chosen in order to take into account the trade-off between steady-state error and number of erroneous recoveries (which is related to the number of non-recovered sources) [65].

5.4.1.2 The Fast Multiuser Constant Modulus Algorithm

The algorithm proposed in [65] aims to improve the performance of the MU-CMA at the cost of an increase in the associated computational complexity. Its development is based on a recursive version of the MU-CMA, and the technique is called fast multiuser constant modulus algorithm (FMU-CMA) or least-squares with adaptive decorrelation constant modulus algorithm (LSAD-CMA) [176].

The FMU-CMA also uses the decorrelation approach to force the recovery of different source signals, and employs a recursive expression to optimize a time-averaged version of Equation 5.94. The adaptive algorithm can be described with the aid of the following equations:

$$\mathbf{w}_k(n) = \mathbf{R}_{xy,k}^{-1}(n)\mathbf{d}_{xy,k}(n) \tag{5.103a}$$

$$\mathbf{R}_{xy,k}(n+1) = \zeta\mathbf{R}_{xy,k}(n) + (1-\zeta)\left|y_k(n)\right|^2 \mathbf{x}^*(n)\mathbf{x}^T(n) \tag{5.103b}$$

$$\mathbf{d}_{xy,k}(k+1) = \zeta\mathbf{d}_{xy,k}(n) + (1-\zeta)\rho_2 y_k(n)\mathbf{x}^*(n) - \gamma \sum_{\substack{i=1\\ i\neq k}}^{N} \widehat{r}_{ik}(n)\widehat{\mathbf{p}}_i(n), \tag{5.103c}$$

where ζ is a smoothing term and $\widehat{r}_{ik}(n)$ and $\widehat{\mathbf{p}}_i(n)$ are obtained from (5.97).

The recursive procedure in (5.103) improves the convergence speed of the constant modulus approach at the cost of increasing the implementation complexity of the algorithm, thus partially solving one of the points raised about the performance of the MU-CMA. The other point, namely the tradeoff between steady-state error and number of erroneous recovery, requires a modification of the cost function.

The solution proposed in [66] is the inclusion of an adaptive regularization factor that promotes a self-adjust of the MAI. This procedure improves the performance of the steady-state error, since, intuitively, we can realize that the decorrelation factor may be larger in the initial steps: when the users have been reasonably separated, the equalizer outputs should already be significantly uncorrelated, and the decorrelation factor may be reduced in order to improve the steady-state performance. The proposal includes, then, a criterion in which γ varies in time and is dependent on the level of cross-correlation between the estimated user signals.

The average level of correlation per user is measured by

$$\bar{r}_k(n) = \frac{1}{K-1} \sum_{\substack{i=1 \\ i \neq k}}^{K} |\widehat{r}_{ik}(n)|^2. \tag{5.104}$$

The adaptive decorrelation factor is then obtained by employing a nonlinear mapping that allows a saturation of the values of $\bar{r}_k(n)$. Such mapping is given by [66]

$$\gamma_k(n) = \tanh\left[\bar{r}_k(n)\right], \tag{5.105}$$

where $\tanh(\cdot)$ stands for the hyperbolic tangent function. Another possibility is to consider a normalization of the matrix $\mathbf{R}_y(n)$ shown in Equation 5.97a and obtain $\bar{r}_k(n)$ from the normalized values of $\mathbf{R}_y(n)$, thus rendering the nonlinear mapping unnecessary. In this case, we may use

$$\gamma_k(n) = \bar{r}_k(n). \tag{5.106}$$

Finally, it is also possible to develop a space-time version of the FMU-CMA by generalizing the equations to include the temporal dispersion as well, as described in the MU-CMA case.

5.4.1.3 The Multiuser pdf Fitting Algorithm (MU-FPA)

The multiuser fitting pdf criterion, proposed in [63,64], is based on the estimation of the equalizer output pdfs, using as a reference a parametric model for the pdf of the ideally recovered signals. Hence, the procedure consists of minimizing the divergence between two density functions (the output signal pdf and the corresponding parametric model) using the *Kullback–Leibler divergence* (KLD). The KLD is given by the following expression [78]:

$$D_{p_Y(y\|\Phi(y))} = \int_{-\infty}^{\infty} p_Y(y) \cdot \ln\left(\frac{p_Y(y)}{\Phi(y)}\right) dy, \tag{5.107}$$

where the function $\Phi(y)$ is the parametric model that fits the statistical behavior of an ideally recovered signal [64].

Since we deal with discrete symbols in the presence of Gaussian noise, it is suitable to pose

$$\Phi(y) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \sum_{i=1}^{S} \exp\left(-\frac{|y - s_i|^2}{2\sigma_r^2}\right) \cdot P(a_i), \tag{5.108}$$

where
S is the cardinality of the transmitted alphabet
$P(s_i)$ is the probability of occurrence of a symbol s_i
σ_r^2 is the variance of the Gaussian kernels we assume in the model

In this sense, the algorithm can be understood as an attempt to "equalize" the pdfs of the transmitted signal and the filter output.

The proposed cost function, to be minimized, is

$$\begin{aligned} J_{\text{FPC}}(\mathbf{w}(n)) &= -\int_{-\infty}^{\infty} p_Y(y) \ln\left(\Phi(y)\right) dy \\ &= -E\left\{\ln\left(\Phi(y)\right)\right\} \end{aligned} \tag{5.109}$$

where FPC stand for fitting pdf criterion.

A stochastic version for filter adaptation is given by

$$\nabla J_{\text{FPC}}(\mathbf{w}(n)) = \frac{\sum_{i=1}^{S} \exp\left(-|y(n) - s_i|^2/2\sigma_r^2\right) P(a_i)\left(y(n) - a_i\right)\mathbf{x}(n)}{\sigma_r^2 \sum_{i=1}^{S} \exp\left(-|y(n) - s_i|^2/2\sigma_r^2\right) P(a_i)} \tag{5.110}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla J_{\text{FPC}}(\mathbf{w}(n)),$$

where μ is the algorithm step-size. This adaptive algorithm is referred to as fitting pdf algorithm (FPA).

Based on the approach proposed in [228], we can use the criterion of explicit decorrelation of beamformer outputs to establish the cost function for the multiuser fitting pdf criterion (MU-FPC). Thus, we obtain the following criterion for the MIMO scenario:

$$J_{\text{MU-FPC}}\left(\mathbf{w}_k(n)\right) = J_{\text{FPC}}\left(\mathbf{w}_k(n)\right) + \gamma \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left| r_{ij} \right|^2, \tag{5.111}$$

where
γ is the decorrelation factor
$r_{ij} = E\left\{y_i(n)y_j^*(n)\right\}$ is the cross-correlation between the ith and jth outputs

Furthermore, we can adapt the decorrelation weight γ, as discussed in the Section 5.4.1.2, in order to increase the convergence rate and decrease steady-state error. The adaptation procedure of the algorithm for the kth user becomes

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) - \mu \nabla J_{\text{FPC}}\Big(\mathbf{w}_k(n)\Big) - \gamma(n) \sum_{\substack{i=1 \\ i \neq k}}^{N} \widehat{r}_{ik}(n) \mathbf{p}_i(n), \tag{5.112}$$

where
$\widehat{r}_{ik}(n)$ is the (i, k) element of matrix $\mathbf{R}_{yy}(n)$
$\mathbf{p}_i(n)$ is the ith column of matrix $\mathbf{P}(n)$

Such matrices may be computed as

$$\begin{aligned} \mathbf{R}_{yy}(n+1) &= \varsigma \mathbf{R}_{yy}(n) + (1-\varsigma)\mathbf{y}(n)\mathbf{y}^H(n) \\ \mathbf{P}(n+1) &= \varsigma \mathbf{P}(n) + (1-\varsigma)\mathbf{x}(n)\mathbf{y}^H(n), \end{aligned} \tag{5.113}$$

where
$\mathbf{y}(n) = \begin{bmatrix} y_1(n) & \cdots & y_N(n) \end{bmatrix}^T$ is the output vector
ς is a forgetting factor

The decorrelation weight is updated by [65]

$$\begin{aligned} \bar{r}_k(n) &= \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq k}}^{N} |\widehat{r}_{ik}|^2 \\ \gamma(n) &= \tanh\left[\bar{r}_k(n)\right] \end{aligned} \tag{5.114}$$

For space-time multiuser processing, it is also necessary to guarantee decorrelation of the signals in a time interval in order to remove the ISI. With that in mind, the criterion becomes [63]

$$J_{\text{MU-FPC}}\Big(\mathcal{W}_k(n)\Big) = J_{\text{FPC}}\Big(\mathcal{W}_k(n)\Big) + \gamma(n)\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\ell=-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \left|r_{ij,\ell}\right|^2, \tag{5.115}$$

where

$$r_{ij,\ell} = E\left\{y_i(n)y_j^*(n-\ell)\right\} \tag{5.116}$$

is the cross-correlation between the ith and jth outputs of the space-time receivers for a lag ℓ, and $\frac{\Delta}{2}$ is the maximum lag for which the output signals of the different filters must be decorrelated.

The MU-FPA, for space-time processing, is then given by [62]

$$\mathcal{W}_k(n+1) = \mathcal{W}_k(n) - \mu\nabla J_{\text{FPC}}\Big(\mathcal{W}_k(n)\Big)$$

$$- \gamma(n)\sum_{\substack{i=1\\ i\neq k}}^{N}\sum_{\ell=-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \widehat{r}_{ik,\ell}(n)\widehat{\mathbf{p}}_{i,\ell}(n) \tag{5.117}$$

$$\mathbf{R}_{y,\ell}(n+1) = \varsigma\mathbf{R}_{y,\ell}(n) + (1-\varsigma)\mathbf{y}(n)\mathbf{y}^H(n-\ell) \tag{5.118}$$

$$\mathbf{P}_{\ell}(n+1) = \varsigma\mathbf{P}_{\ell}(n) + (1-\varsigma)\boldsymbol{\mathcal{X}}(n)\mathbf{y}^H(n-\ell) \tag{5.119}$$

$$\mathbf{y}(n-\ell) = \begin{bmatrix} y_1(n-\ell) & \cdots & y_K(n-\ell) \end{bmatrix}^T \tag{5.120}$$

$$\ell = -\frac{\Delta}{2}, \ldots, \frac{\Delta}{2}, \tag{5.121}$$

where

$\widehat{r}_{ik,\ell}(n)$ is the cross-correlation between the ith and jth user estimates with lag ℓ at time index n, given by the (i,j)th element of matrix $\mathbf{R}_{y,\ell}(n)$

$\widehat{\mathbf{p}}_{i,\ell}(n)$ is the ith column of matrix $\mathbf{P}_{\ell}(n)$

α is a smoothing factor related to the process of learning the involved statistics

5.4.2 Multiuser Detection Methods Based on Orthogonalization Criteria

Another family of algorithms tries to replace the direct decorrelation of the equalizer outputs with an alternative condition that guarantees the correct recovery of the source signals even in the presence of the near–far effect. The approach is based on a constrained optimization procedure in which the global response is subject to preserving the structure of a perfect recovery condition [75]. Let us present two strategies that use this approach in the sequel.

5.4.2.1 The Multiuser Kurtosis Algorithm

The multiuser kurtosis maximization (MUK) criterion, proposed in [224,225], is based on the SW theorem, previously discussed in Chapter 4, and uses a set of necessary conditions for the blind equalization of several source signals. Such conditions are the following:

1. $s_l(n)$ is i.i.d. and zero mean $(l = 1, \ldots, N)$.
2. $s_l(n)$ and $s_q(n)$ are statistically independent for $l \neq q$, with the same pdf.
3. $\left|c_4\left[y_l(n)\right]\right| = \left|c_4\left[s(n)\right]\right| \quad (l = 1, \ldots, N)$.
4. $E\left\{\left|y_l(n)\right|^2\right\} = \sigma_s^2 \quad (l = 1, \ldots, N)$.
5. $E\left\{y_l(n)y_q(n)\right\} = 0, \quad l \neq q$.

where

$c_4\left[s(n)\right]$ and σ_s^2 are, respectively, the kurtosis and the variance (power) of the source signals

$c_4\left[\cdot\right]$ is the kurtosis operator, as defined in Chapter 4

We can note that Condition 5 aims to ensure the same desired condition in the decorrelation approach defined by (5.94).

Furthermore, the MUK technique does not employ the decorrelation approach such as the MU-CMA. In order to attain correct identification of the different signals, the MUK takes an additional criterion based on the orthogonalization of the combined channel+equalizer matrix $\mathbf{G}$ in such a way that $\mathbf{G}^H\mathbf{G} = \mathbf{I}$. In this context, the criterion can be written as [224,225]

$$\begin{cases} \max\limits_{\mathbf{G}} \quad J_{\text{MUK}}\left(\mathbf{G}\right) = \sum\limits_{k=1}^{N} \left|c_4\left[y_k\right]\right| \\ \\ \text{subject to: } \mathbf{G}^H\mathbf{G} = \mathbf{I} \end{cases} . \tag{5.122}$$

We can then divide this algorithm into two stages:

Equalization step: kurtosis maximization, in the spirit of the SW theorem. This stage is associated with a matrix $\mathbf{W}^e$.

Separation step: is responsible for promoting the uncorrelation of the estimates of the several users. This stage is associated with a matrix $\mathbf{W}$.

In order to force the global response matrix to be orthogonal, a Gram–Schmidt orthogonalization procedure is used in the matrix $\mathbf{W}^e$ [225]. This procedure, carried out iteratively, forces the equalizer outputs to be uncorrelated.

Consequently, the stochastic gradient of the MUK criterion is given by [224,225]

$$\nabla J_{\text{MUK}}(\mathbf{G}) = 4\,\text{sign}\left(c_4\left[s(n)\right]\right)\sum_{k=1}^{N}\left\{\left|y_k(n)\right|^2 y_k(n)\mathbf{x}^*(n)\right\}. \tag{5.123}$$

We can observe the similarity of the above equation with Equation 4.37.

Hence, at the first step (equalization), an adaptation of $\mathbf{W}(n)$ is performed in the direction of the instantaneous gradient, in a very similar way to the SW algorithm, leading to

$$\mathbf{W}^e(n+1) = \mathbf{W}(n) + \mu\,\text{sign}\left(c_4(s(n))\right)\mathbf{x}(n)\mathcal{Y}(n), \tag{5.124}$$

where $\mathcal{Y}(n) = \begin{bmatrix} \left|y_1(n)\right|^2 y_1(n) & \cdots & \left|y_K(n)\right|^2 y_K(n) \end{bmatrix}$ and μ is a step-size.

Once the equalization stage is carried out, the constraint related to the orthogonalization of $\mathbf{G}$ must be respected. In addition, the receiver data must be whitened, in the space or in both space and time domains, which means that the matrix $\mathbf{H}$ must be unitary. The goal of such hypothesis is also to ensure a constant variance (power) of the transmitted data, hence respecting the conditions for signal recovery.

The orthogonalization step, for the kth user, is given by

$$\mathbf{w}_k(n+1) = \frac{\mathbf{w}_k^e(n+1) - \sum_{l=1}^{k-1}\left[\mathbf{w}_l^T(n+1)\mathbf{w}_k^e(n+1)\right]\mathbf{w}_l(n+1)}{\left\|\mathbf{w}_k^e(n+1) - \sum_{l=1}^{k-1}\left[\mathbf{w}_l^T(n+1)\mathbf{w}_k^e(n+1)\right]\mathbf{w}_l(n+1)\right\|}, \tag{5.125}$$

where $\|\cdot\|$ stands for the l_2-norm of the vector.

To carry out this step, it is usual to employ the Schur algorithm [90–92, 134] to the covariance matrix of the noise-free signal $\mathbf{R}_{\tilde{x}}$, which, due to its symmetry, provides the following decomposition:

$$\mathbf{R}_{\tilde{x}} = \mathbf{LDL}^H, \tag{5.126}$$

where

$\tilde{x}$ represents the noise-free signals

$\mathbf{L}$ is a unitary matrix

$\mathbf{D}$ is a diagonal matrix with real entries

A noise-free estimation of

$$\mathbf{R}_x = E\left\{\mathbf{x}\mathbf{x}^H\right\} \tag{5.127}$$

may be performed by estimating the noise variance, σ_v^2, by means of the average of the lowest eigenvalues of $\widehat{\mathbf{R}}_x$, and then making

$$\widehat{\mathbf{R}}_{\widetilde{x}} = \widehat{\mathbf{R}}_x - \sigma_v^2 \mathbf{I}_M \tag{5.128}$$

Notice that this possibility is related to the concept of subspace decomposition presented in Section 5.3.2.

Ideally, we must have $\mathbf{R}_{\widetilde{x}} = \mathbf{H}\mathbf{H}^H$, assuming normalized and statistically independent sources; hence, we would have that $\mathbf{L}\mathbf{D}^{\frac{1}{2}}$ is equal to $\mathbf{H}$, up to a unitary ambiguity matrix $\mathbf{U}$, which means that

$$\widetilde{\mathbf{L}} = \mathbf{H}\mathbf{U}, \tag{5.129}$$

where $\widetilde{\mathbf{L}}$ is built as the matrix that contain the N larger norm columns of $\mathbf{L}\mathbf{D}^{\frac{1}{2}}$ [90, 134, 227]. Finally, the prewhitening is performed by $\widetilde{\mathbf{L}}^{\dagger}$, where † denotes the pseudo-inverse, given by [139]

$$\widetilde{\mathbf{L}}^{\dagger} = \left(\widetilde{\mathbf{L}}^H \widetilde{\mathbf{L}}\right)^{-1} \widetilde{\mathbf{L}}^H \tag{5.130}$$

For space-time processing, a space-time prewhitening is demanded in order to allow the use of the MUK in MIMO equalization processing. After the prewhitening algorithm, the signal model is analogous to the case of spatial processing: the only difference is related to the dimensions of the involved vectors.

The development of the whitening transformation in this case is performed as in (5.126), respecting the order of the vectors. As shown in [226] and also in [227], we may use a temporal prediction method in order to replace the temporal equalization step in the prewhitening processing.

5.5 Concluding Remarks

In this chapter, we extended the problem of unsupervised signal processing to the scenario of multichannel systems. This scenario is first introduced under the light of a system-theoretic analysis, where we present the notion of Smith form to establish equalizability conditions.

In contrast to the SISO case, the central theoretical results that arise when we consider the multichannel configuration are essentially as follows: the possibility of perfect (ZF) equalization even if both channel and equalizer are FIR structures; and the possibility of relying on methods based only on the second-order statistics of the signals. We emphasized these results in the more treatable case of SIMO channels equalization.

The possibility of perfect (ZF) equalization, even with an FIR SIMO channel and an FIR MISO equalizer, was demonstrated by using the Bezout's identity. The result about the use of second-order statistics is closely related to dealing with the cyclic spectra of the cyclostationary signals engendered by oversampling operation. Indeed, we have shown that a SIMO configuration might be used to model the case of SISO channel with oversampled signal.

As far as the methods for blind equalization are concerned, we gave special attention to the two well-established second-order methods, those of subspace decomposition and linear prediction, and commented about robust methods that opt for employing higher-order statistics.

Finally, we discussed the problem of multiuser processing, where a given transmitted signal suffers from channel impairments together with interference effects, and presented some proposed solutions. This problem, and that of MIMO equalization in general, is in fact closely related to those of source separation, to be considered next.

6

Blind Source Separation

MIMO channels were studied in Chapter 5 as a general case of the blind equalization problem. However, since we were dealing with digital communication channel, two hypotheses were implicitly presented: the transmitted signals were modeled as a sequence of symbols from a finite alphabet and these sequences were assumed to be i.i.d.

A more general scenario occurs when we discard the two aforementioned hypotheses and replace the idea of a MIMO transmission channel by a generic MIMO system that engenders both mixture and distortion of a set of input signals. The recovery of these original signals after the mixing process constitutes the problem of source or signal separation and, similarly to the equalization problem, we talk about *blind source separation* (BSS) when the recovery is carried out by unsupervised methods.

Interest in source separation techniques has intensively grown from their genesis in the beginning of the 1980s until nowadays. From a theoretical standpoint, the general BSS problem remarkably captures the notion of information extraction that, in a way, embodies the ensemble of methods and tools considered in this book. As a consequence, and from a more practical standpoint, source separation techniques are relevant in a great number of applications. To mention a few, BSS is concerned with understanding and extracting information from data as diverse as neuronal activities and brain images, communications, audio, video, and sensor signals in general.

A major tool to perform BSS is the so-called *independent component analysis* (ICA), the relevance of which can be confirmed by the fact that ICA was also the name of the most prestigious conference on source separation and its applications, from its first version, in 1999, to its latest version in 2009. Nevertheless signal or source separation involves not only ICA or blind techniques, since semi-blind and factorization methods that make use of prior information about the problem at hand must be considered in a number of applications. This is reflected in the new name *latent variable analysis* (LVA), used in the aforementioned conference in its ninth edition in 2010, which emphasizes the general character of the problems in signal processing and information extraction related to the theme.

The structure of this chapter is influenced by a general view on the subject and by the idea of joining together key theoretical elements and models of practical significance. Our aim is to present the theoretical foundations of BSS together with some celebrated methods and algorithms in an accessible and synthetic way. We have no intention, however, of dealing with all aspects of an area that has received attention and contribution from such different standpoints and research communities, like signal processing, machine learning, and statistics. Hence, the chapter is organized as follows:

- In Section 6.1, we present the BSS problem in a general form and discuss some modeling elements that are important to define specific models. Particular attention is given to the standard case of a linear, instantaneous, and noiseless mixture, since such a model serves as a starting point to the following discussions.
- Section 6.2 introduces the very relevant conceptual link between BSS and ICA. We state fundamental theorems and definitions, describe the preprocessing procedure of data whitening, and then present a number of important criteria to perform ICA, with the aid of which practical algorithms can be implemented.
- Section 6.3 complements the previous one by bringing a study on some of the most well-known algorithms for performing ICA. The section starts with the algorithm proposed by Jeanny Hérault and Christian Jutten, a true landmark, as commented in the historic notes. Among other relevant techniques, we present the important principle of *principal component analysis* (PCA) in order to introduce the nonlinear PCA algorithm.
- As previously pointed out, there exist different approaches that go beyond ICA and strictly blind techniques. Section 6.4 is devoted to some of these techniques that are suitable to the source separation problem, particularly by exploiting prior information about the task at hand. We present two key factorization methods: the nonnegative and the sparse component decompositions.
- Section 6.5 extends the discussion on BSS to the case of *convolutive mixtures*. Convolutive mixtures are an extension of the standard linear and instantaneous model that accounts for the existence of mixture components originated by delayed versions of the sources. The temporal ingredient added to the model widens the applicability of BSS and also gives rise to connections with the equalization problem.
- Section 6.6 deals with *nonlinear mixtures*. This scenario constitutes a more intricate extension of the standard case, since the need for inverting nonlinear models via ICA may lead to certain ambiguities that require special care. In order to overcome such limitations, we introduce the post-nonlinear (PNL) model together with some separation approaches related to it, like that of Gaussianization.
- Finally, we close the chapter with the concluding remarks exposed in Section 6.7.

Historical Notes

The origin of BSS is generally traced to the early 1980s,* particularly to the efforts of Hérault, Jutten, and Ans in the study of the problem of motion coding [144].† Throughout this decade, the BSS problem gradually attracted the interest of the signal processing community, mainly in France and afterward in Europe. From the 1990s, interest in BSS and its application was widespread among the international community and the theme become a "hot topic" in a number of important conferences and journals.

An important landmark was established with Comon's work on ICA [74]. Relationships between BSS and ICA opened perspectives of building new criteria and algorithms to perform BSS.

Important early work concerning the use of higher-order statistics is associated with the names of Lacoume [174] and Cardoso [56], the latter being responsible for the proposal of the JADE algorithm. The approach of nonlinear PCA, which has conceptual affinities with neural network theory, took shape with works like [164, 217]. Another classical approach that has strong connections with the field of neural networks [188] is the information maximization (Infomax) method, the origin of which is due to Bell and Sejnowski [31]. Our list is completed by mentioning Cichocki and Unbenhauen's important algorithm [72], to the pioneer efforts using the idea of natural gradient [9] and to the extremely popular FastICA method [149].

Other approaches to source separation were developed considering different assumptions about the signals. For instance, in [38, 237], the idea of exploring the correlation structure of the sources is addressed. In [242], the authors consider the problem of separating signals that are always nonnegative, giving rise to the so-called nonnegative ICA. In [45] the notion of sparsity was used to perform blind identification of a mixture, an approach that is now referred to as sparse component analysis (SCA).

The problem of convolutive source separation is conceptually similar to that of blind deconvolution in a MIMO domain. A formulation in terms of deconvolution or equalization is very common in communications, where transmitted signals are modeled by an i.i.d. sequence of symbols that belong to a finite alphabet. Nevertheless, there are many other problems that make use of BSS models and methods. From a historical point of view, it is important to mention applications like audio signals [199], astronomical data [59, 158, 215], and brain images [17, 53, 133]. Reference [235] provides an interesting survey on the subject.

* More details about historical aspects of BSS can be found in [148] and [160].

† In [160], the work of Bar-Ness et al. [27] is mentioned as a possible independent effort in the field of communications.

The problem of nonlinear BSS has received a great deal of attention from the 1990s on. Important efforts were established within a neural computation framework, like Burel's work [50] and proposals based on self-organizing maps [221]. A key reference in the context of post-nonlinear mixtures is Taleb and Jutten's work [280], and the study of this class of mixtures is still a prolific research subject [3,281].

6.1 The Problem of Blind Source Separation

Figure 6.1 depicts the general problem of BSS, in which a set of information signals is submitted to the mixing and eventually distorting effect of a MIMO system. The resulting signals are captured by a set of sensors. The purpose of source separation techniques is to recover the original signals from the sensor outputs, by means of an appropriate separating system. Similarly to the equalization case, we talk about BSS when the mixing system is unknown and the desired signals are not available for any kind of training procedure.

The problem may be formulated as follows: let us consider a set of N signals, denoted sources, whose samples form a source vector $\mathbf{s}(n)$, and a set of M signals, the observations, organized in a vector $\mathbf{x}(n)$. The observations represent, in general, a mixture of the different source signals and can be expressed as

$$\mathbf{x}(n) = \mathbf{F}\left(\mathbf{s}(n), \ldots, \mathbf{s}(n-L), \mathbf{n}(n), n\right) \tag{6.1}$$

where $\mathbf{F}(\cdot)$ is the mixing mapping.

Clearly, if $\mathbf{F}(\cdot)$ is known a priori, the sources can be estimated by obtaining the optimal inverse mapping (assuming that it exists and the noise is negligible). Moreover, if it is possible to rely on a training sequence, it should

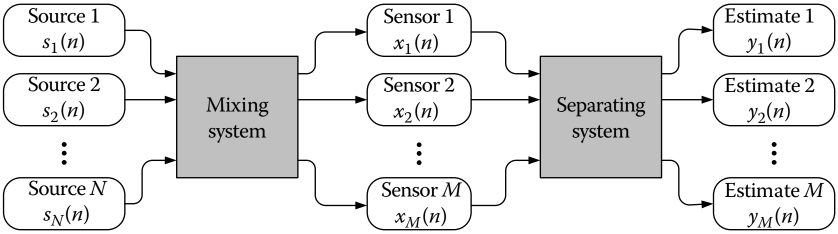


FIGURE 6.1
The general source separation problem.

be possible to identify the mixing mapping in a supervised manner and then obtain the inverse mapping. The challenge, however, is to obtain such an estimate based only on the observed samples, with a minimal amount of information about the signals and the mixing system.

Even though the solution to the general problem stated above would fit several different applications, there is no general solution. Nevertheless, it is possible to treat particular models that can be classified according to some specific characteristics.

- *Linear or nonlinear:* A mixing system is said to be linear if the mapping $\mathbf{F}(\cdot)$ obeys the superposition principle (presented in (2.19), and repeated here for the sake of clarity)

$$\mathbf{F}(\alpha_1\mathbf{s}_1(n) + \alpha_2\mathbf{s}_2(n)) = \alpha_1\mathbf{F}(\mathbf{s}_1(n)) + \alpha_2\mathbf{F}(\mathbf{s}_2(n)) \tag{6.2}$$

  for all constants α_1 and α_2 and source vectors $\mathbf{s}_1(n)$ and $\mathbf{s}_2(n)$. Otherwise, the system is said to be *nonlinear*. The case of nonlinear mixtures will be discussed in more detail in Section 6.6.
- *Memoryless or convolutive:* Similarly to the nomenclature introduced in Chapter 5, if the observations represent a mixture of source samples at different time instants, i.e., $L > 0$, a linear mixing system gives rise to a convolutive mixture. For $L = 0$, the simplest case, we are dealing with a memoryless or instantaneous mixture.
- *Number of sources and number of sensors:* In the situation in which the number of sensors is greater than the number of sources $(M < N)$ we have an overdetermined mixture. Similarly, if there are less sensors than sources $(M > N)$, we deal with an underdetermined mixture.

Let us consider the simplest case, in which the mixing process is modeled by a noiseless linear memoryless system, i.e., that of a linear instantaneous mixture. In this case, the model is completely characterized by a mixing matrix $\mathbf{A}_{M\times N}$, and the observation vector is given by

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) \tag{6.3}$$

which means that the observations are linear combinations of the sources.

In order to present the basic principles related to the problem of BSS, let us assume this noiseless scenario and that there are as many sensors as sources $(\mathbf{A}_{N\times N})$. Moreover, unless stated otherwise, the source signals are considered to be real-valued zero-mean stationary stochastic processes [230]. Therefore, whenever it is possible, the time index will be omitted.

Under the aforementioned conditions, and assuming that the mixing matrix is invertible, signal separation is achieved obtaining a separating matrix $\mathbf{W}$ such that

$$\begin{aligned}\mathbf{y} &= \mathbf{W}\mathbf{x}\\ &= \mathbf{W}\mathbf{A}\mathbf{s}\\ &= \mathbf{s}\end{aligned} \tag{6.4}$$

i.e., $\mathbf{W} = \mathbf{A}^{-1}$. The goal of BSS techniques consists in obtaining $\mathbf{W}$ without any explicit knowledge about the sources and/or the mixing matrix $\mathbf{A}$. For that, a key hypothesis is that the sources are considered to be random signals statistically independent between each other. This hypothesis was determinant to the development of several methods based on ICA, which is discussed in the sequel.

6.2 Independent Component Analysis

Mathematically, the elements of a random vector $\mathbf{s}$ are statistically independent if

$$p_{\mathbf{s}}(\mathbf{s}) = p_{s_1}(s_1)p_{s_2}(s_2)\dots p_{s_N}(s_N) \tag{6.5}$$

where

$p_{\mathbf{s}}(\mathbf{s})$ is the joint probability density function (pdf) of its elements $s_1, s_2, \dots, s_N$

$p_{s_i}(s_i)$ is the marginal pdf of the ith source

In his seminal work [74], Comon showed that if the sources are mutually independent, it is possible to obtain the separating matrix in an unsupervised manner. The main idea is to look for a matrix $\mathbf{W}$ such that the estimated signals $\mathbf{y} = \mathbf{W}\mathbf{x}$ are also mutually independent. The proof of this property is based on a theorem due to Darmois–Skitovich [161], which can be enunciated as follows.

THEOREM 6.1 (Darmois–Skitovich)

Let $s_1, s_2, \dots, s_N$ be a set of zero-mean and statistically independent random variables. Also, let y_1 and y_2 be defined as

$$\begin{aligned}y_1 &= a_1 s_1 + a_2 s_2 + \cdots + a_N s_N\\ y_2 &= b_1 s_1 + b_2 s_2 + \cdots + b_N s_N\end{aligned} \tag{6.6}$$

If y_1 and y_2 are mutually independent, then all variables s_i for which $a_i b_i \neq 0$ will be Gaussian random variables. ■

Thus, according to this theorem, it would not be possible to obtain independent random variables from a linear mixture of non-Gaussian sources. Consequently, if we consider that the coefficients a_i and b_i are the parameters of the overall input–output mapping **WA**, it is clear that the estimate will only be independent, assuming that none of them has a Gaussian distribution, if the sources are no longer mixed.

Therefore, source separation can be achieved via ICA, which can be formally defined as follows.

DEFINITION 6.1 (ICA) The ICA of a random vector $\mathbf{x} = [x_1 x_2 \ldots x_M]^T$ consists of determining a matrix **W** such that the elements of $\mathbf{y} = \mathbf{Wx}$ be as statistically independent as possible, in the sense of optimizing a cost function that expresses, direct or indirectly, the notion of independence between signals.*

Therefore, source recovery relies on two important concepts: those of statistical independence and of non-Gaussianity. Nonetheless, it is important to note that the solution found using ICA will recover the sources up to scale and permutation ambiguities. That is because if a vector **s** is composed of independent random variables, so will be a vector that is just a permutation of **s**, or even a scaled version thereof. In other words,

$$\mathbf{y} = \mathbf{\Lambda P s} \tag{6.7}$$

will also present independent components,

where
Λ is a diagonal matrix
P is a permutation matrix, will also present independent components

Hence, the conditions under which the sources can be recovered using ICA may be summarized in the following theorem [74].

THEOREM 6.2 (Separability)

The system presented in (6.3) is separable by ICA, i.e., it is possible to obtain **W** such that $\mathbf{y} = \mathbf{Wx}$ correspond to the sources up to scale and permutation ambiguities, if and only if the mixing matrix **A** is full rank and there is, at most, one Gaussian source. ■

The first condition regarding the rank of **A** is self-evident, since we are looking for a separating matrix that should, in some sense, invert the mixing

* In the context of source separation, the cost function is also termed a *contrast function* [74].

matrix. However, the sources can only be blindly recovered as long as there is, at most, one Gaussian source. The reason why Gaussian sources are not allowed should become clearer in Section 6.2.1, in which we discuss the limitations of second-order statistics in the source separation problem.

6.2.1 Preprocessing: Whitening

As discussed in Chapter 2, decorrelation is a less restrictive concept than statistical independence, so that, if two random variables are statistically independent, they will also be uncorrelated. Without loss of generality, we may consider the case in which the sources are zero mean, with unit variance, so that

$$E\left\{\mathbf{s}\mathbf{s}^T\right\} = \mathbf{I} \tag{6.8}$$

Under these conditions, the autocorrelation matrix of the observations is given by

$$\begin{aligned}\mathbf{R}_\mathbf{x} &= E\left\{\mathbf{x}\mathbf{x}^T\right\}\\ &= \mathbf{A}\mathbf{R}_\mathbf{s}\mathbf{A}^T\\ &= \mathbf{A}\mathbf{A}^T\end{aligned} \tag{6.9}$$

In its turn, the mixing matrix **A** can be expressed as follows, by employing the singular value decomposition (SVD) [128]:

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{V}^T \tag{6.10}$$

Since **V** is a unitary matrix, it comes from (6.9) and (6.10) that

$$\mathbf{R}_\mathbf{x} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \tag{6.11}$$

Now, let us consider the data transform

$$\bar{\mathbf{x}} = \mathbf{T}\mathbf{x} \tag{6.12}$$

where

$$\mathbf{T} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T \tag{6.13}$$

Then, the transformed data $\bar{\mathbf{x}}(n)$ presents an autocorrelation matrix given by

$$\begin{aligned}\mathbf{R}_{\bar{\mathbf{x}}} &= E\left\{\bar{\mathbf{x}}\bar{\mathbf{x}}^T\right\}\\ &= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{R}_{\mathbf{x}}\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\\ &= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\\ &= \mathbf{I} \end{aligned} \tag{6.14}$$

i.e., the transformed data is uncorrelated.* In this case, it is said that the data was *whitened* or *spherized*.

It is important to mention that the whitening transformation is not unique, since $\mathbf{T} = \mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{V}^T$, for any orthogonal matrix $\mathbf{Q}$, will also produce uncorrelated data. So it is clear that the whitening transformation $\mathbf{T}$ does not provide the original sources, which are not only uncorrelated, but also independent. Two consequences of this result deserve to be commented in the sequel.

First, we can easily conclude that it is not possible to recover the original independent sources by dealing only with second-order statistics. This is in fact a similar result to that obtained when we discussed and compared the features of equalization and prediction in the precedent chapters. In the BSS problem, the whitening procedure cannot guarantee a correct retrieval of independent sources in the same way that, in SISO blind equalization, it could not guarantee a correct retrieval of the i.i.d. transmitted signal. It is worth mentioning, incidentally, that the standpoint of directly looking for independence still seems not to be properly exploited in SISO equalization methods.

Second, the aforementioned result indicates why it is not possible to separate Gaussian sources. In fact, in the case in which we have, for instance, a mixture of two zero-mean Gaussian sources, to obtain uncorrelated estimates also implies obtaining independent signals. However, since the whitening transform is not unique, the obtained independent signals do not correspond necessarily to the original sources. In other words, if two or more Gaussian sources are present, we cannot guarantee that obtaining independent signal lead us to recover the sources, so that ICA is not, in this case, a suitable method for BSS.

Even though obtaining uncorrelated data does not lead to source separation, this procedure can be seen as a preprocessing step in BSS

* Signals are uncorrelated if the covariance matrix is diagonal. In this case, since we are dealing with zero-mean signals, the autocorrelation matrix and the covariance matrix are the same, and hence, signals are uncorrelated.

algorithms [148]. Let us consider again the whitening transform in (6.12), which leads us to

$$
\begin{aligned}
\bar{\mathbf{x}} &= \mathbf{T}\mathbf{x} \\
&= \boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^{T}\mathbf{A}\mathbf{s} \\
&= \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{T}\mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{V}^{T}\mathbf{s} \\
&= \mathbf{V}\mathbf{s}
\end{aligned}
\tag{6.15}
$$

where **V** is an orthogonal matrix. We can observe that the whitening process reduces the problem to one in which the mixing matrix is orthogonal, and hence, limits the search for the separating matrix to the group of orthogonal matrices. In order to illustrate the effect of preprocessing, we present the following example.

Example 6.1 (Two-Source Mixture)

Let us consider two independent sources uniformly distributed in the interval $\left[-\sqrt{3}, \sqrt{3}\right]$. The joint pdf of the sources is then given by

$$
p\left(x, y\right) = \begin{cases} \frac{1}{12}, & -\sqrt{3} \leq x, y \leq \sqrt{3} \\ 0, & \text{otherwise} \end{cases}
$$

and is illustrated in Figure 6.2a.

Let the mixing matrix be

$$
A = \begin{bmatrix} 1 & 0.5 \\ 0.3 & 0.9 \end{bmatrix}
$$

Then, the distribution of the observed data becomes as illustrated in Figure 6.2b. Notice that the mixing matrix distorts and rotates the original source distribution. Interestingly, after a prewhitening procedure, the distribution of the transformed data, as shown in Figure 6.2c, resembles the original distribution, except for a rotation factor. So, the remaining step consists in determining the orthogonal matrix corresponding to the rotation that will restore the source distribution.

6.2.2 Criteria for Independent Component Analysis

As previously discussed, the very essence of ICA, as discussed in Section 6.2, consists of determining a separating structure that provides estimates as independent as possible. Different criteria were proposed in order to implement the idea of ICA, and in the following we discuss some of them.

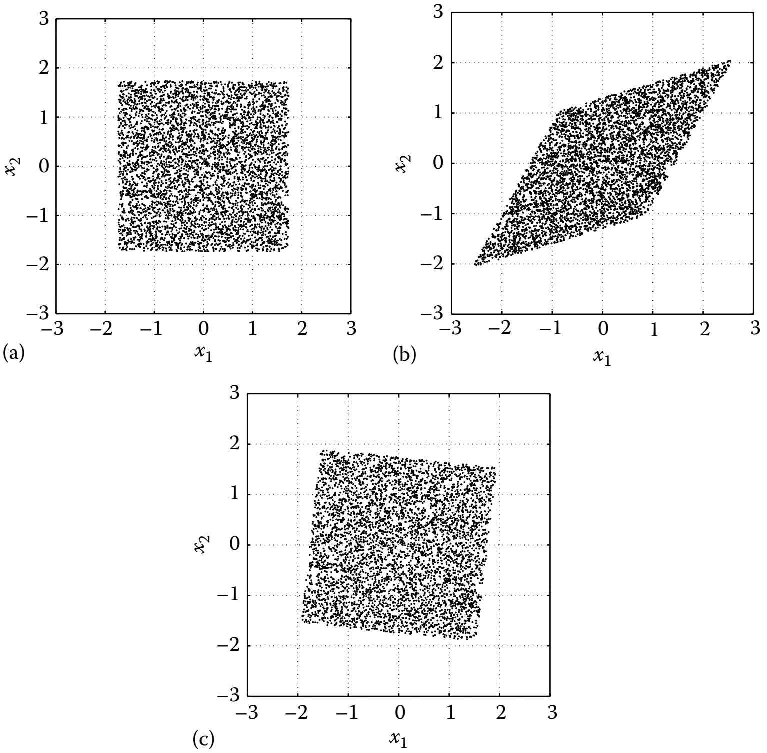


FIGURE 6.2
Joint distributions of the uniformly distributed original sources, of mixtures of them, and of the prewhitened mixtures: (a) joint source distribution, (b) joint distribution of the observations, and (c) joint distribution of the prewhitened data.

6.2.2.1 Mutual Information

The information-theoretic concept of mutual information [78] can be used to quantify independence between random variables. In order to clarify this idea and properly define mutual information, it is important to begin with the definition of another information-theoretic concept.

DEFINITION 6.2 (Differential Entropy) Let α denote a random variable characterized by its pdf $p_\alpha(\alpha)$. The differential entropy, or simply entropy, $H(\alpha)$ is defined by [272]

$$H(\alpha) = -E\left[\log p_\alpha(\alpha)\right] = -\int_{-\infty}^{\infty} p_\alpha(\tau)\log p_\alpha(\tau)\,d\tau \tag{6.16}$$

The entropy conveys the idea of uncertainty of a random variable and it is possible to show that the Gaussian distribution presents the largest entropy among all distributions with the same mean and variance. On the other hand, if we restrict this comparison only to distributions with finite support, the largest entropy will be obtained for a uniform distribution [78].

It is worth mentioning that $H(\alpha)$ can also be defined for discrete random variables. In this case,

$$H(\alpha) = -\sum_i p_i \log p_i \tag{6.17}$$

and presents similar properties to the differential entropy.

In a similar fashion, one can define the conditional entropy as follows.

DEFINITION 6.3 (Conditional Entropy) Let $p_{\alpha|\beta}(\alpha|\beta)$ denotes the conditional pdf of α given β. The *conditional entropy* is given by

$$H(\alpha|\beta) = -E\left[\log p_{\alpha|\beta}(\alpha|\beta)\right] = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{\alpha,\beta}(\tau,\upsilon)\log p_{\alpha|\beta}(\tau|\upsilon)\,d\tau d\upsilon \tag{6.18}$$

which is related to the uncertainty of a random variable given the observation of another random variable.

From Definitions 6.2 and 6.3, we can finally define the mutual information $I(\alpha,\beta)$ as follows.

DEFINITION 6.4 (Mutual Information) The mutual information between two random variables α and β is defined by

$$I(\alpha,\beta) = H(\alpha) - H(\alpha|\beta) = H(\beta) - H(\beta|\alpha) \tag{6.19}$$

From (6.16) and (6.18), one can show that the mutual information can also be expressed as

$$I(\alpha,\beta) = \int p_{\alpha,\beta}(\alpha,\beta)\log\frac{p_{\alpha,\beta}(\alpha,\beta)}{p_\alpha(\alpha)\,p_\beta(\beta)}d\alpha d\beta \tag{6.20}$$

It is particularly useful to interpret the mutual information in terms of the so-called Kulback–Leibler divergence [78], defined in (5.107) and rewritten here for the sake of convenience,

$$D\left(p_x(x) \parallel q_y(y)\right) = \int p_x(x) \log \frac{p_x(\alpha)}{q_y(\alpha)} d\alpha \tag{6.21}$$

which is always nonnegative and being zero only if the pdfs $p_x(x)$ and $q_y(y)$ are identical. From (6.20) and (6.21) it is possible to show that

$$I(\alpha, \beta) = D\left(p_{\alpha,\beta}(\alpha, \beta) \parallel p_\alpha(\alpha) p_\beta(\beta)\right) \tag{6.22}$$

Equation 6.22 reveals that the mutual information can be interpreted as a measure of proximity between the joint distribution of α and β and the product of their marginal distributions. Furthermore, since the Kullback–Leibler divergence is zero if and only if the distributions are identical, one concludes that the mutual information equals zero if and only if α and β are independent. Therefore, mutual information quantifies the degree of statistical dependence between random variables, exactly what is needed to implement an ICA method.

The concept of mutual information can be extended to a vector of N random variables $\alpha_1, \ldots, \alpha_N$. The mutual information among all elements is defined as

$$I(\alpha_1, \ldots, \alpha_N) = D\left(p_{\alpha_1,\ldots,\alpha_N}(\alpha_1, \ldots, \alpha_N) \parallel p_{\alpha_1}(\alpha_1) \ldots p_{\alpha_N}(\alpha_N)\right) \tag{6.23}$$

and can also be expressed as

$$I(\alpha_1, \alpha_2, \ldots, \alpha_n) = \sum_{i=1}^{N} H(\alpha_i) - H(\alpha_1, \alpha_2, \ldots, \alpha_n) \tag{6.24}$$

indicating that the minimization of the mutual information is equivalent to making $H(\alpha_1, \alpha_2, \ldots, \alpha_n)$ as close as possible to the sum of marginal entropies [148].

Hence, evaluating the mutual information between the outputs of the separating system depicted in Figure 6.1 it is possible to obtain, for linear mixtures:

$$\begin{aligned} I(\mathbf{y}) &= \sum_{i=1}^{N} H(y_i) - H(\mathbf{y}) \\ &= \sum_{i=1}^{N} H(y_i) - H(\mathbf{x}) - \log\left|\det \mathbf{W}\right| \end{aligned} \tag{6.25}$$

It is important to note that $H(\mathbf{x})$ does not depend on the elements of $\mathbf{W}$, which means that minimization of the mutual information leads to the following criterion for source separation:

$$\min_{\mathbf{W}} \sum_{i=1}^{N} H\left(y_i\right) - \log\left|\det \mathbf{W}\right| \tag{6.26}$$

6.2.2.2 A Criterion Based on Higher-Order Statistics

Another possible criterion to implement ICA explores the joint cumulants of the involved signals [54,74]. The main idea is to explore the fact that the joint cumulant $c(X_1, \ldots, X_k)$ of a set of independent variables $X_1, \ldots, X_k$ is always null, for any order (see Section 2.3.3). For instance, considering only two random variables X and Y, in order to ensure that they are independent, the following should hold:

$$c(\underbrace{X, \ldots, X,}_{s\ terms} \underbrace{Y, \ldots, Y}_{q\ terms}) = c(X^s, Y^q) = 0 \tag{6.27}$$

for any $s, q = 1, \ldots, \infty$.

Even though independence between signals, in general, is related to all joint cumulants, signal separation can be achieved using only fourth-order cumulants [74]. The restriction, however, is that there be no more than one source with null kurtosis. Notice that this restriction includes the Gaussian distribution as a special, and perhaps the most representative, case. Therefore, for a wide range of applications, BSS can be performed solely based on the information brought by the fourth-order joint cumulants.

Let $c(y_i, y_j, y_k, y_l)$ denote the fourth-order joint cumulant between signals y_i, y_j, y_k, and y_l. Then, according to the previous discussion, the separating structure should be chosen such that $c(y_i, y_j, y_k, y_l)$ be minimal for any combination of indices i, j, k, and l, except for $i=j=k=l$, the case in which $c(y_i, y_j, y_k, y_l) = c_4(y_i)$. Therefore, a possible optimization criterion to reach this condition is given by

$$\min_{\mathbf{W}} \sum_{\Omega} \left|c(y_i, y_j, y_k, y_l)\right|^2 \tag{6.28}$$

where Ω denotes all possible combinations of i, j, k, and l, except for $i=j=k=l$.

If we consider that the data has been prewhitened, so that we should look for an orthogonal separating matrix, it is possible to show that the criterion in (6.28) is equivalent to

$$\max_{\mathbf{W}} \sum_{i=1,\ldots,N} \left|c(y_i, y_i, y_i, y_i)\right|^2 \tag{6.29}$$

where N is the number of signals to be recovered.

6.2.2.3 Nonlinear Decorrelation

Independence between signals can also be verified by means of the nonlinear correlation, which can be seen as an extension of the concept of correlation presented in Section 2.4.1. The nonlinear correlation between two random variables is defined as

$$\eta(X,Y) = E\left\{f_1(X)f_2(Y)\right\} \tag{6.30}$$

with $f_1(\cdot)$ and $f_2(\cdot)$ representing two arbitrary nonlinear functions.

If X and Y are independent, we have

$$E\left\{f_1(X)f_2(Y)\right\} = E\left\{f_1(X)\right\}E\left\{f_2(Y)\right\} \tag{6.31}$$

The converse statement is only true if (6.31) holds for *all* continuous functions $f_1(\cdot)$ and $f_2(\cdot)$ that are zero outside a finite interval [230]. Nevertheless, it is possible to employ the notion of nonlinear correlation to obtain very simple BSS methods.

Let us consider that both $f_1(\cdot)$ and $f_2(\cdot)$ are smooth functions with derivatives of all orders around the origin. In these conditions, (6.30) can be expressed in terms of the Taylor expansion of these nonlinear functions as [148]

$$E\left\{f_1(X)f_2(Y)\right\} = \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} f_1^{(k)}f_2^{(l)}E\{X^kY^l\} \tag{6.32}$$

where $f_1^{(i)}$ and $f_2^{(i)}$ denote the coefficients of the Taylor series. Hence, if X and Y are independent and either $E\{X^k\}=0$ or $E\{Y^k\}=0$, for all k, then the nonlinear correlation is zero, and the variables are said to be nonlinearly decorrelated.

Therefore, if we are trying to separate two signals y_i and y_j, the following criterion could be employed:

$$\min E\left\{f_1\left(y_i\right)f_2\left(y_j\right)\right\} \tag{6.33}$$

It is important to note that the condition that either $E\{X^k\}=0$ or $E\{Y^k\}=0$ implicitly requires that $f_1(\cdot)$ or $f_2(\cdot)$ be an odd function, which means that the corresponding Taylor series has only odd powers, otherwise the aforementioned condition would imply that even moments like the variance are zero.

It is worth pointing out that the nonlinear decorrelation may not be effective in all cases, since we cannot guarantee that an arbitrary pair of nonlinear functions will lead to independent signals.

6.2.2.4 Non-Gaussianity Maximization

According to the central limit theorem [230], the pdf of a sum of independent random variables tends, under certain conditions, to a Gaussian distribution [77]. From this fact, we can expect that the pdf of a sum of two random variables will be "closer" to a Gaussian distribution than any of the two original variables [148]. This idea is illustrated in Figure 6.3, which shows a mixture of two independent sources, one with uniform distribution and the other with a Laplacian distribution. We can observe that both resulting signals present a distribution that resembles a Gaussian distribution.

Considering the model presented in (6.3), if we are interested in recovering only one of the sources, we can employ a separating vector $\mathbf{w}_1$ such that

$$\begin{aligned} y_1 &= \mathbf{w}_1^T \mathbf{x} \\ &= \mathbf{w}_1^T \mathbf{A} \mathbf{s} \\ &= \mathbf{q}_1^T \mathbf{s} = \sum_{i=1}^{N} q_i s_i \end{aligned} \tag{6.34}$$

where $\mathbf{q} = \mathbf{w}_1^T \mathbf{A}$ represents the joint effect of the mixing system and the separating structure. Notice that, in a perfect source recovery condition, $\mathbf{q}$ should be a vector with a single nonzero element.

Since $y_1 = \mathbf{q}_1^T \mathbf{s}$ is a linear combination of the sources, one can assume that its pdf will be closer to a Gaussian pdf than that of any one of the sources s_i.

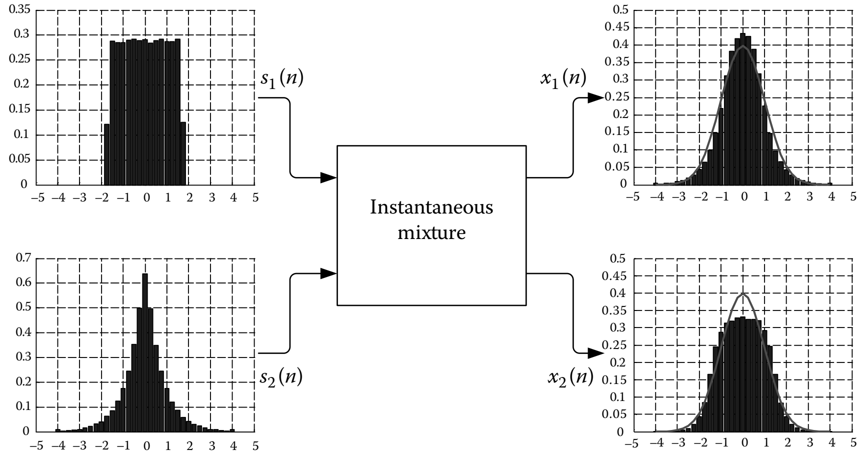


FIGURE 6.3
Mixtures between uniform- and Laplacian-distributed signals.

Thus, one may say that y_1 will be "less" Gaussian if its distribution is equal to that of one of the sources, i.e., when $\mathbf{q}$ presents only one nonzero element. Therefore, to obtain a vector $\mathbf{w}_1$ that maximizes the non-Gaussianity of $\mathbf{w}_1^T\mathbf{x}$ should also lead to source recovery.

The classical way of quantifying the gaussianity of a distribution is to use the *kurtosis*, defined in Section 2.3.3. Most distributions present a nonzero kurtosis value, the Gaussian being one of the few exceptions. In fact, it is usual to classify pdfs according to their kurtosis value: if $K(x) > 0$, it is said that x has a super-Gaussian distribution; if $K(x) < 0$, x has a sub-Gaussian distribution. Hence, one criterion that expresses the maximization of the non-Gaussian character of a signal is given by

$$\max_{\mathbf{w}} \left| K\left(y_i\right) \right| \tag{6.35}$$

Another possibility of quantifying the non-Gaussianity is by means of the concept of negentropy [148], defined as follows:

DEFINITION 6.5 (Negentropy) The negentropy of a random variable Y is defined as

$$J_{Negentropy}\left(Y\right) = H\left(Y_{Gauss}\right) - H\left(Y\right) \tag{6.36}$$

where Y_{Gauss} represents a Gaussian variable with the same mean and variance of Y.

The notion of negentropy is based on the fact that a Gaussian random variable will present the largest entropy over all other distributions with the same mean and variance [230]. Therefore, the negentropy will always assume nonnegative values, being zero only if y is Gaussian. In this sense, the negentropy quantifies the degree of proximity between the pdf of y and that of a Gaussian variable.

One interesting point regarding the non-Gaussianity maximization is that it can be used to estimate the sources individually. Due to this feature, techniques based on this approach are usually associated within the framework of blind source extraction (BSE) [79, 185]. In BSE, we are not interested in all sources, but only in a subset of the signals contained in the mixture. If the number of signals to be extracted is the same as the total number of sources, we reach again with the BSS problem.

The procedure for the extraction of more than one source can be carried out with two distinct strategies, both exploring the fact that the extracting vectors $\mathbf{w}_i$, obtained considering whitened data, will necessarily be orthogonal.

1. Serial estimation: In this approach, the components are sequentially estimated. This means that all extracting vectors $\mathbf{w}_i$ must be, necessarily, orthogonal to the previously obtained vectors. For this purpose, one can employ the Gram–Schmidt orthogonalization method [128]. This serial approach is also known in the literature as the deflation approach [89].
2. Parallel estimation: In this case, a certain number of sources will be estimated at once, adapting in parallel the vectors $\mathbf{w}_i$. However, since it is required that all vectors be orthogonal, an additional step of orthonormalization is required, and the Gram–Schmidt procedure can be employed again [148].

6.2.2.5 The Infomax Principle and the Maximum Likelihood Approach

Another interesting approach to perform ICA is the so-called Infomax principle, introduced in the context of BSS by Bell and Sejnowski [31], even though key results had already been established in a different context [212]. The approach is based on the concepts issued from the field of neural networks. Neural networks will be discussed in more detail in Chapter 7, but for the moment, it suffices to consider that one possible structure of a neural network is composed of a linear portion and a set of nonlinearities.

Let us consider the structure depicted in Figure 6.4, where $\mathbf{A}$ represents the mixing system. The separating system is an artificial neural network composed by a linear part (the matrix $\mathbf{W}$) and a set of nonlinearities $f_i(\cdot)$, each one applied to a particular output y_i, so that we define the vector.

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} f_1(y_1) & f_2(y_2) & \cdots & f_N(y_N) \end{bmatrix}^T \tag{6.37}$$

The nonlinear functions $f_i(\cdot)$ are monotonically increasing, with $f_i(-\infty) = 0$ and $f_i(\infty) = 1$.

According to the Infomax principle, the coefficients of the neural network should be adjusted in order to maximize the amount of information that flows from the inputs to the outputs, which means that $\mathbf{W}$ should be chosen to maximize the mutual information between $\mathbf{x}$ and $\mathbf{z}$, thus leading to

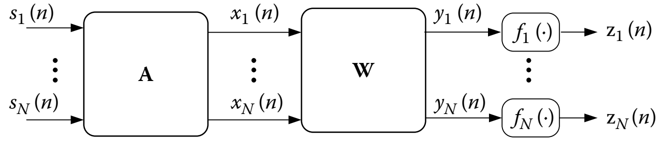


FIGURE 6.4
Structure of an artificial neural network.

the following criterion:

$$\max_{\mathbf{W}} I(\mathbf{x}, \mathbf{z}) \tag{6.38}$$

When the noise is negligible, it can be shown that the maximization of the mutual information corresponds to the maximization of the joint entropy of the outputs,

$$\max_{\mathbf{W}} H(\mathbf{z}) \tag{6.39}$$

a strategy known in the literature as *MaxEnt* [31]. In fact, if we rewrite the joint entropy of **z** as

$$H(\mathbf{z}) = \sum_{i=1}^{N} H(z_i) - I(\mathbf{z}) \tag{6.40}$$

we may notice that the maximization of the joint entropy is related to the maximization of the marginal entropies and to the minimization of the mutual information between the elements of **z**. Thus, maximization of the joint entropy will tend, in general, to reduce the statistical dependence between the network outputs, and, consequently, the dependence between the elements of **y**.

The nonlinear functions have two important roles: First, they limit the value of variables z_i, and hence the value of their entropies. Second, it is possible to verify that obtaining independent outputs is closely related to an adequate choice of these nonlinear functions.

From Figure 6.4 we may show that

$$H(\mathbf{z}) = H(\mathbf{x}) + E\left\{\sum_{i=1}^{N} \log(f_i'(\mathbf{w}_i\mathbf{x}))\right\} + \log(|\det(\mathbf{W})|) \tag{6.41}$$

where
$f_i'(\cdot)$ denotes the derivative of $f_i(\cdot)$
$\mathbf{w}_i$ denotes the *i*th row of **W**

Since the joint entropy of **x** does not depend on the separating matrix, the Infomax principle applied to the BSS problem leads to the following optimization problem:

$$\max_{\mathbf{W}} J_{Infomax}(\mathbf{W}) \triangleq E\left\{\sum_{i=1}^{N} \log(f_i'(\mathbf{w}_i\mathbf{x}))\right\} + \log(|\det(\mathbf{W})|) \tag{6.42}$$

Even though this is not apparent, the Infomax approach to source separation is closely related to the maximum likelihood method [55, 56]. In [56], Cardoso shows that, in the context of BSS, the log likelihood function is expressed by

$$J_{ML}(\mathbf{W}) \propto E\left\{\sum_{i=1}^{N} \log\left(p_{s_i}\left(\mathbf{w}_i\mathbf{x}\left(j\right)\right)\right)\right\} + \log\left(\left|\det(\mathbf{W})\right|\right) \tag{6.43}$$

Comparing (6.42) and (6.43), it is clear that both approaches have very similar cost functions, the only difference being the nonlinearities. In fact, the cost functions are exactly the same when $f_i(\cdot)$ equals the cumulative distribution function (CDF) of the ith source. If that is the case, the pdf of $f_i\left(y_i\right)$ will be uniform in the interval $[0, 1]$ when y_i is equal to s_i or to some other source with the same pdf [56].

If the likelihood function is rewritten in terms of the Kullback–Leibler divergence, i.e.,

$$J_{ML}(\mathbf{W}) = -D\left(p_{\mathbf{y}}\left(\mathbf{y}\right) \| p_{\mathbf{s}}\left(\mathbf{s}\right)\right) \tag{6.44}$$

it becomes clear that the maximum likelihood approach is, in a certain sense, a pdf matching criterion: matrix **W** should be chosen in order that the distribution of the estimates be "as close as possible" to the distribution of the sources. In the Infomax approach, however, the true pdfs of the sources are replaced by the derivatives of nonlinear functions $f_i(\cdot)$. Nonetheless, it is important to remark that even if there is no perfect match between these functions and the pdf of the sources, it can still be possible to separate the signals [56].

6.3 Algorithms for Independent Component Analysis

The ICA criteria exposed in Section 6.2 indicate theoretical solutions to the BSS problem. However, the effectiveness of such solutions depends on finding feasible algorithms to implement them in practical scenarios. This section discusses some classical algorithms, starting from Hérault and Jutten's seminal approach.

6.3.1 Hérault and Jutten's Approach

The proposal of Hérault and Jutten [143, 144] is considered to be the first algorithm capable of extracting signals from linear mixtures. The method is inspired in elements of neural networks and based on the structure presented in Figure 6.5, which is a fully interconnected neural network composed of linear neurons.

From Figure 6.5, the output signal can be expressed as

$$y_1 = x_1 - m_{21}y_2 \tag{6.45}$$

$$y_2 = x_2 - m_{12}y_1 \tag{6.46}$$

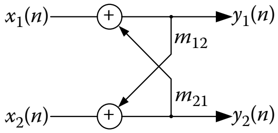


FIGURE 6.5
Fully interconnected linear neural network used as separating structure.

or, equivalently, using matrix notation

$$\mathbf{y} = \mathbf{x} - \mathbf{M}\mathbf{y} \tag{6.47}$$

where $\mathbf{M}$ is the weight matrix composed of elements m_{ij}, with $m_{ij} = 0$ for $i = j$.

The update law for $\mathbf{M}$ is given by

$$m_{ij} \leftarrow m_{ij} - \mu E\{f(y_i)g(y_j)\} \tag{6.48}$$

Such update law employs the idea of nonlinear decorrelation discussed in Section 6.2: the algorithm stops updating the weights when the nonlinear correlation between the outputs is null.

As discussed earlier, nonlinear decorrelation does not guarantee, in all cases, that the signals are mutually independent. In practice, the effective use of the algorithm is restricted to scenarios with a limited number of sources, and there may be convergence problems even for the case with two sources, as studied in [72, 93, 278]. Nevertheless, Herault and Jutten's algorithm is one of the simplest BSS algorithms, and had a major importance in the development of the BSS research field.

6.3.2 The Infomax Algorithm

The Infomax algorithm, also known as the Bell–Sejnowski (BS) algorithm, is derived from the Infomax principle, discussed in Section 6.2.2.5, and employs a steepest-descent approach to update the free parameters. Thus, the first step to build the adaptation rule is to obtain the gradient of the cost function:

$$\frac{\partial J_{Infomax}(\mathbf{W})}{\partial \mathbf{W}} = E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T\} + (\mathbf{W}^T)^{-1} \tag{6.49}$$

where $\mathbf{g}(\cdot) = [g_1(\cdot) \ldots g_N(\cdot)]$ is a vector of functions such that

$$g_i(x) = \frac{d \log\left(f_i'(x)\right)}{dx} \tag{6.50}$$

Then, the update rule for the separating matrix **W** will be given by

$$\mathbf{W} \leftarrow \mathbf{W} + \mu\{E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T\} + (\mathbf{W}^T)^{-1}\} \tag{6.51}$$

where μ is the learning step. The stochastic version of this algorithm is obtained by simply ignoring the expectation operator in (6.51), leading to the following update rule:

$$\mathbf{W} \leftarrow \mathbf{W} + \mu\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T + (\mathbf{W}^T)^{-1}\} \tag{6.52}$$

This algorithm is known both as the Infomax algorithm and as the BS algorithm.

6.3.3 Nonlinear PCA

The nonlinear PCA algorithm is a direct extension of the classical PCA, a well-known tool in the field of data analysis, and is widely used in applications like the data compression.

In a data compression problem, the main goal could be to represent a random vector $\mathbf{x} = [x_1, \ldots, x_M]^T$ in a lower-dimensional subspace. In this context, PCA can be understood as a tool to provide orthonormal basis vectors $\mathbf{w}_i$ of a subspace of this kind, by minimizing

$$J_{PCA}(\mathbf{W}) = E\left\{ \|\mathbf{x} - \sum_{i=1}^{N_{Dimensions}} \left(\mathbf{w}_i^T\mathbf{x}\right)\mathbf{w}_i\|^2 \right\} \tag{6.53}$$

The projections of the original data onto these basis vectors are known as *principal components* and the above cost function represents the compression error, which decreases as the number of principal components increases.

Assuming that **x** is zero mean, the solution to this optimization problem is given by the eigenvectors of $\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\}$ [94] (vide Appendix A), and the principal components are thus mutually uncorrelated.

According to our previous discussion in Section 6.2.1 regarding the use of second-order statistics in the BSS problem, it is clear that PCA could only be used as a tool for the prewhitening step. Nevertheless, by including nonlinearities in (6.53), following the same idea described in the nonlinear decorrelation approach, it is possible to implicitly make use of higher-order statistic and, thus, obtain signals that are not only uncorrelated but also independent. The resulting method is known as nonlinear PCA, or NPCA [148], and its cost function is given by

$$J_{NPCA}(\mathbf{W}) = E\left\{ \|\mathbf{x} - \sum_{i=1}^{N_{Dimensions}} \left(g_i\left(\mathbf{w}_i^T\mathbf{x}\right)\right)\mathbf{w}_i\|^2 \right\} \tag{6.54}$$

where $g_i(\cdot)$ is a nonlinear function. In this case, the nonlinearity is included in the projection onto the basis vectors $\mathbf{w}_i$, but there are other possibilities [148]. In matrix notation, (6.54) can be expressed as

$$J_{NPCA}(\mathbf{W}) = E\{\|\mathbf{x} - \mathbf{W}^T\mathbf{g}(\mathbf{W}\mathbf{x})\|^2\} \tag{6.55}$$

where $\mathbf{g}(\cdot) = [g_1(\cdot) \ldots g_N(\cdot)]$.

If $\mathbf{x}$ has been prewhitened, the separating matrix $\mathbf{W}$ will be orthogonal and (6.55) reduces to

$$J_{NPCA}(\mathbf{W}) = \sum_{i=1}^{N} E\{[y_i - g_i(y_i)]^2\} \tag{6.56}$$

It is interesting to notice that (6.56) is very similar to the Bussgang algorithms [213] discussed in Section 4.3.

The cost function defined in (6.55) can be minimized by any optimization method. However, the original proposal employs a recursive least squares (RLS, vide Section 3.5.1) approach [222]. The NPCA algorithm employing this approach is given in Algorithm 6.1.

Algorithm 6.1: Nonlinear PCA

1. Randomly initialize $\mathbf{W}(0)$ and $\mathbf{P}(0)$;
2. While a stopping criterion is not met, do:

$$\mathbf{z}(n) = \mathbf{g}(\mathbf{W}(n-1)\bar{\mathbf{x}}(n)) \tag{6.57}$$

$$\mathbf{h}(n) = \mathbf{P}(n-1)\mathbf{z}(n) \tag{6.58}$$

$$\mathbf{m}(n) = \mathbf{h}(n)/(\lambda + \mathbf{z}^T(n)\mathbf{h}(n)) \tag{6.59}$$

$$\mathbf{P}(n) = \lambda^{-1}\Upsilon[\mathbf{P}(n-1) - \mathbf{m}(n)\mathbf{h}(n)^T] \tag{6.60}$$

$$\mathbf{e}(n) = \mathbf{z}(n) - \mathbf{W}(n-1)^T\mathbf{z}(n) \tag{6.61}$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{m}(n)\mathbf{e}(n)^T \tag{6.62}$$

where $\Upsilon[\mathbf{J}]$ denotes an operator that generates a new symmetric matrix with the same upper-triangular portion of $\mathbf{J}$, $\bar{\mathbf{x}}(n)$ denotes the whitened data and λ is the forgetting factor of RLS algorithm.

6.3.4 The JADE Algorithm

The essence of the JADE (Joint Approximate Diagonalization of Eigenmatrices) [54] algorithm lies in the information provided by the joint cumulant of the signals in a very interesting optimization procedure based on the Jacobi method for matrix diagonalization [128].

Suppose that the observed data has been prewhitened, i.e., $\bar{\mathbf{x}} = \mathbf{U}\mathbf{s}$, $\mathbf{U}$ being an orthogonal matrix. First, it is necessary to introduce the concept of a cumulant matrix related to matrix $\mathbf{M}$, $Q^{\bar{\mathbf{x}}}(\mathbf{M})$, whose elements are defined as

$$Q_{ij}^{\bar{\mathbf{x}}}(\mathbf{M}) = \sum_k \sum_l c(x_i, x_j, x_k, x_l)\mathbf{M}_{kl} \tag{6.63}$$

Considering the properties of cumulants (seen in Chapter 2), one can deduce the following [57]:

$$Q^{\bar{\mathbf{x}}}(\mathbf{M}) = \mathbf{U}\triangle(\mathbf{M})\mathbf{U}^T \tag{6.64}$$

where $\triangle(\mathbf{M})$ is given by

$$\triangle(\mathbf{M}) = diag\left\{c_4(s_1)\mathbf{a}_1^T\mathbf{M}\mathbf{a}_1, \ldots, c_4(s_N)\mathbf{a}_N^T\mathbf{M}\mathbf{a}_N\right\} \tag{6.65}$$

$\mathbf{a}_i$ being the ith column of the mixing matrix $\mathbf{A}$.

Equation 6.64 can be regarded as the eigendecomposition of $Q^{\mathbf{x}}(\mathbf{M})$. Hence, if all eigenvalues are distinct, the decomposition is unique [128] and matrix $\mathbf{U}$ can be easily determined by a decomposition algorithm. In practice, however, it is not possible to determine whether a matrix $\mathbf{M}$ will render distinct eigenvalues of $Q^{\mathbf{x}}(\mathbf{M})$ or not, since this depends on the unknown mixing matrix $\mathbf{A}$.

In order to overcome this difficulty, JADE algorithm employs a joint diagonalization approach. The main idea is to jointly diagonalize a set of cumulant matrices, corresponding to different matrices $\mathbf{M}_i$. Mathematically, the associated cost function is defined as

$$D(\mathbf{U}) = \sum_i \Omega(\mathbf{U}^T Q^{\bar{\mathbf{x}}}(\mathbf{M_i})\mathbf{U}) \tag{6.66}$$

where $\Omega(\cdot)$ represents the quadratic sum of all elements that are not in the main diagonal. Matrices $\mathbf{M}_i$ should ideally be chosen such that $Q^{\bar{\mathbf{x}}}(\mathbf{M}_i) = \lambda\mathbf{M}_i$. If that is the case, all relevant information regarding the joint cumulants is taken into account. Then, the optimization of (6.66) is performed via a Jacobi method for matrix diagonalization, which can be seen in detail in [60].

6.3.5 Equivariant Adaptive Source Separation/Natural Gradient

As pointed in Chapters 3 and 4, a number of optimization algorithms in signal processing is based on the gradient method, the main idea of which is to explore the gradient of a given cost function to find its minimum (or maximum). Following this procedure, the adaptation of a matrix **W** has the general form

$$\mathbf{W} \leftarrow \mathbf{W} \pm \mu \left. \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} \right|_{\mathbf{W}} \tag{6.67}$$

where the sign of the update term depends on whether we are dealing with a maximization or minimization problem, and $J(\mathbf{W})$ denotes a generic cost function.

In [58], another approach is presented. Cardoso and Laheld employ a serial adaptation, which consists of updating the separating matrix according to

$$\mathbf{W} \leftarrow \big(\mathbf{I} - \lambda \boldsymbol{\Psi}(\mathbf{y})\big)\,\mathbf{W} \tag{6.68}$$

where
$\boldsymbol{\Psi}(\cdot)$ maps a vector onto a matrix,
λ represents the learning step

Hence, the increment is made by left-multiplying a matrix, instead of adding a term to the previous separating matrix.

Therefore, the adaptation rule in (6.68) suggests that we can redefine the concept of gradient. In the standard case, the gradient at **W** can be understood as being the first-order term of a Taylor series of $J(\mathbf{W} + \mathbf{D})$:

$$J(\mathbf{W} + \mathbf{D}) \approx J(\mathbf{W}) + tr\left(\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}^{T} \mathbf{D}\right) \tag{6.69}$$

where **D** corresponds to an increment. On the other hand, the relative gradient can be defined in a similar fashion from the expansion of $J(\mathbf{W}+\mathbf{DW})$

$$\begin{aligned} J(\mathbf{W} + \mathbf{DW}) &\approx J(\mathbf{W}) + tr\left(\mathbf{W}\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}^{T} \mathbf{D}\right) \\ &\approx J(\mathbf{W}) + tr\left(\frac{\partial_R J(\mathbf{W})}{\partial \mathbf{W}}^{T} \mathbf{D}\right) \end{aligned} \tag{6.70}$$

i.e., the relative gradient is given by

$$\frac{\partial_R J(\mathbf{W})}{\partial \mathbf{W}} = \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} \mathbf{W}^T \tag{6.71}$$

For a given cost function $J(\mathbf{W}) = E\{\mathbf{f}(\mathbf{y})\}$, it is possible to build the following update rule from the relative gradient:

$$\mathbf{W} \leftarrow \mathbf{W} - \lambda E\{\mathbf{f}'(\mathbf{y})\mathbf{y}^T\}\mathbf{W}, \tag{6.72}$$

where $\mathbf{f}'(y) = [f'(y_1) \ldots f'(y_N)]$. In addition to that, by including an orthogonality constraint on $\mathbf{W}$, the update rule becomes

$$\mathbf{W} \leftarrow \mathbf{W} - \lambda E\{\mathbf{y}\mathbf{y}^T - \mathbf{I} + \mathbf{f}'(\mathbf{y})\mathbf{y}^T - \mathbf{y}\mathbf{f}'(\mathbf{y})^T\}\mathbf{W} \tag{6.73}$$

which corresponds to the so-called Equivariant Adaptive Source Separation (EASI) algorithm.

It is interesting to notice that the notion of natural gradient, developed by Amari [8], is very similar to that of the relative gradient, leading to expressions similar to those obtained by Cardoso and Laheld. In [8], Amari defines the gradient taking into account the structure subjacent to the parameter space, i.e., the nonsingular matrices space. In this space, the gradient direction is given by

$$\frac{\partial_{Natural} J(\mathbf{W})}{\partial \mathbf{W}} = \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} \tag{6.74}$$

which is very similar to the relative gradient.

6.3.6 The FastICA Algorithm

Another contribution whose innovation is mainly related to the optimization method is the FastICA algorithm [147, 149], a very popular tool for ICA. The formal derivation of the algorithm considers that signals have been prewhitened, and also considers an approximation for the negentropy of a zero-mean unit variance signal

$$J(y) = \left(E\left\{f(y)\right\} - E\left\{f(\nu)\right\}\right)^2 \tag{6.75}$$

where
$f(\cdot)$ represents a nonquadratic function
ν represents a zero-mean unit variance Gaussian variable

Thus, if we are estimating only one signal, we should look for a unit-norm vector $\mathbf{w}$ that maximizes the negentropy, i.e.

$$\max\left(E\left\{f(y)\right\}-E\left\{f(\nu)\right\}\right)^2 \quad s.t. \quad \|\mathbf{w}\|^2=\mathbf{w}^T\mathbf{w}=1 \tag{6.76}$$

Since $E\left\{f(\nu)\right\}$ does not depend on $\mathbf{w}$, one may notice that the optima of $E\left\{f(y)\right\}$ will also correspond to some optima of $\left(E\left\{f(y)\right\}-E\left\{f(\nu)\right\}\right)^2$. In other words, taking into account the constraint, the solution is given by the point where the gradient of the following Lagrangian is null:

$$\frac{\partial}{\partial\mathbf{w}}L(\mathbf{w})=E\left\{\bar{\mathbf{x}}f'(\mathbf{w}^T\bar{\mathbf{x}})\right\}-\lambda\mathbf{w}=0 \tag{6.77}$$

where
λ denotes the Lagrange multiplier
$\bar{\mathbf{x}}$ represents the vector of prewhitened observed data
$f'(\cdot)$ is the derivative of $f(\cdot)$

In order to solve this problem using Newton's method [148], we must evaluate the second-order derivatives of the cost function, which are given by

$$\frac{\partial^2}{\partial\mathbf{w}\partial\mathbf{w}^T}L(\mathbf{w})=E\left\{\bar{\mathbf{x}}\bar{\mathbf{x}}^Tf''(\mathbf{w}^T\bar{\mathbf{x}})\right\}+\beta\mathbf{I} \tag{6.78}$$

where $f''(\cdot)$ is the second-order derivative of $f(\cdot)$. The FastICA algorithm is then obtained considering an approximation for (6.78), given by

$$E\left\{\bar{\mathbf{x}}\bar{\mathbf{x}}^Tf''(\mathbf{w}^T\bar{\mathbf{x}})\right\}\approx E\left\{\bar{\mathbf{x}}\bar{\mathbf{x}}^T\right\}E\left\{f''(\mathbf{w}^T\bar{\mathbf{x}})\right\}=E\left\{f''(\mathbf{w}^T\bar{\mathbf{x}})\right\} \tag{6.79}$$

Then, following a Newton-like iteration to solve this optimization we obtain, using (6.77) and (6.78), the following update rule for $\mathbf{w}$:

$$\mathbf{w}\leftarrow\mathbf{w}-\left[E\left\{\bar{\mathbf{x}}f''(\mathbf{w}^T\bar{\mathbf{x}})\right\}-\lambda\mathbf{w}\right]\left[E\left\{f''(\mathbf{w}^T\bar{\mathbf{x}})\right\}-\lambda\right]^{-1} \tag{6.80}$$

Finally, multiplying (6.80) by $\lambda-E\left\{\mathbf{w}^T\bar{\mathbf{x}}f'(\mathbf{w}^T\bar{\mathbf{x}})\right\}$ on both sides and performing some algebraic simplifications, we obtain the final update rule used in the FastICA:

$$\mathbf{w}\leftarrow E\left\{\bar{\mathbf{x}}f''(\mathbf{w}^T\bar{\mathbf{x}})\right\}-E\left\{f''(\mathbf{w}^T\bar{\mathbf{x}})\right\}\mathbf{w} \tag{6.81}$$

The FastICA algorithm for the estimation of a single source considering negentropy maximization is given in Algorithm 6.2.

The re-normalization step of $\mathbf{w}(n)$ after each iteration is employed to increase the convergence rate and cope with stability issues of the algorithm. If more than one source is being estimated at the same time, i.e., if parallel estimation is performed, instead of a normalization step we should guarantee that all separating vectors $\mathbf{w}_i(n)$ are orthonormal between each other. The algorithm for this other case is given in Algorithm 6.3.

Algorithm 6.2: FastICA Algorithm for the Estimation of a Single Source

1. Preprocessing: center and whiten data. The preprocessed data form vector $\bar{\mathbf{x}}$;
2. Randomly initialize $\mathbf{w}(n)$, keeping a unit-norm constraint;
3. While a stoping criterion is not met, do:

$$\mathbf{w}'(n) = E\left\{\bar{\mathbf{x}}f'\left(\mathbf{w}(n-1)^T\bar{\mathbf{x}}\right)\right\} - \mathbf{w}(n-1)E\left\{f''\left(\mathbf{w}^T\bar{\mathbf{x}}\right)\right\} \tag{6.82}$$

$$\mathbf{w}(n) = \frac{\mathbf{w}'(n)}{\|\mathbf{w}'(n)\|} \tag{6.83}$$

Algorithm 6.3: FastICA Algorithm for the Estimation of Several Sources

1. Preprocessing: center and whiten data. The preprocessed data form vector $\bar{\mathbf{x}}$;
2. Randomly initialize $\mathbf{w}_i(0)$, keeping a unit-norm constraint;
3. While a stoping criterion is not met, do:

$$\mathbf{w}_i(n) = E\left\{\bar{\mathbf{x}}f'\left(\mathbf{w}_i(n-1)^T\bar{\mathbf{x}}\right)\right\} - \mathbf{w}_i(n-1)E\left\{f''\left(\mathbf{w}_i(n-1)^T\bar{\mathbf{x}}\right)\right\} \tag{6.84}$$

4. Organize $\mathbf{w}_i(n)$ into a matrix $\mathbf{W}'(n) = (\mathbf{w}_1(n), \mathbf{w}_2(n), \ldots, \mathbf{w}_N(n))^T$, and perform an orthogonalization procedure:

$$\mathbf{W}(n) \leftarrow \left(\mathbf{W}'(n)\mathbf{W}'(n)^T\right)^{-\frac{1}{2}}\mathbf{W}'(n) \tag{6.85}$$

6.4 Other Approaches for Blind Source Separation

Independently of the algorithm we use to perform BSS, all methods based on ICA rely on two hypotheses: that the sources are mutually independent, and that at most one of them is Gaussian. However, in some applications, one may have access to more a priori information about the mixing system and the sources.

In this section, we will discuss other approaches that are based on distinct assumptions about the sources and/or the mixing process. Under these additional hypotheses, it is possible to devise specific methods that may present advantages over ICA.

6.4.1 Exploring the Correlation Structure of the Sources

A class of simple but effective algorithms for source separation can be derived if the power spectral densities of the sources are mutually distinct, i.e., if their correlation structure obeys

$$E\left\{s_i\left(k\right)s_i\left(k-l\right)\right\} \neq E\left\{s_j\left(k\right)s_j\left(k-l\right)\right\} \tag{6.86}$$

for $i \neq j$ and some $l \neq 0$.

Differently from the ICA approach, these algorithms explore the time information of the observed signals in order to recover the sources. An interesting point of this new approach is that, in this case, source separation can be carried out using only second-order statistics. In order to understand this fact, let us consider that data has been prewhitened. Then, the autocorrelation matrix of the observations for a given delay l is given by

$$\mathbf{R}_{\bar{\mathbf{x}}}\left(l\right) = E\left\{\bar{\mathbf{x}}(n)\bar{\mathbf{x}}^T(n-l)\right\} = \mathbf{U}\mathbf{R}_{\mathbf{s}}(l)\mathbf{U}^T \tag{6.87}$$

where

$$\mathbf{R}_{\mathbf{s}}\left(l\right) = E\left\{\bar{\mathbf{s}}(n)\bar{\mathbf{s}}^T(n-l)\right\} \tag{6.88}$$

denotes the autocorrelation matrix of the sources for a delay l.

If the sources are mutually independent (or, at least, uncorrelated), $\mathbf{R}_{\mathbf{s}}\left(l\right)$ is diagonal, which means that (6.87) represents the eigendecomposition of $\mathbf{R}_{\bar{\mathbf{x}}}\left(l\right)$. Thus, the decomposition of $\mathbf{R}_{\bar{\mathbf{x}}}\left(l\right)$ reveals the rotation matrix that leads to source separation. This idea is the very essence of the AMUSE algorithm [286] and depends on the correct choice of the delay l for which $\mathbf{R}_{\mathbf{s}}\left(l\right)$ presents distinct eigenvalues. An extension of this method would be to consider the information contained in the autocorrelation matrix for different values of l, leading to algorithms like SOBI [37] and TDSEP [316].

It is important to notice that even though the aforementioned methods are simple and efficient in recovering the sources, they rely on the hypothesis that all sources have distinct power spectra. However, in the situation that they present different but similar power spectra, the methods may not obtain good results in practice [71, 148]. On the other hand, it is not necessary that signals be independent, only uncorrelated. In addition to that, there is no restriction about Gaussian sources.

Another scenario that can employ second-order based methods occurs when the source signals are nonstationary processes, like speech signals. In this situation, it can be shown that decorrelation between the estimated signals is a sufficient condition to guarantee source separation [204, 238].

6.4.2 Nonnegative Independent Component Analysis

Another class of BSS algorithms can be derived if the sources do not assume negative (or, equivalently, positive) values, a natural condition in applications like, for example, image analysis [71].

The hypothesis of nonnegativity or positivity of the signals is present in data analysis methods like the nonnegative matrix factorization (NMF) [177], and positive matrix factorization (PMF) [220]. Nonetheless, it is not always possible to guarantee that the factorization is unique; thus, the direct application of such methods in BSS requires additional constraints on the original signals.

In this sense, nonnegative independent component analysis considers that the sources are both independent and nonnegative, leading to very simple criteria for source separation. The central result related to this approach is summarized in the following theorem [241].

THEOREM 6.3 (Separability of Mixtures of Nonnegative Sources)

Let $\mathbf{s}$ be a random vector of independent elements, each one with unity variance. Each element assumes only nonnegative values and $P(s_i < \delta) > 0$ for all $\delta > 0$. Let also $\mathbf{y} = \mathbf{U}\mathbf{s}$, where $\mathbf{U}$ is an orthogonal matrix, i.e., ($\mathbf{U}\mathbf{U}^\mathbf{T} = \mathbf{I}$). Then, $\mathbf{U}$ will be a permutation matrix if and only if all elements of $\mathbf{y}$ are nonnegative. Therefore, if this condition holds, $\mathbf{y}$ will be exactly the vector of source signals up to a permutation ambiguity. ■

From Theorem 6.3, it is possible to recover the sources based on cost functions that express the nonnegativity of the elements of $\mathbf{y}$. Figure 6.6 shows an example with two uniform sources distributed in the interval [0, 1]. After the prewhitening step, the sources will be recovered by determining the

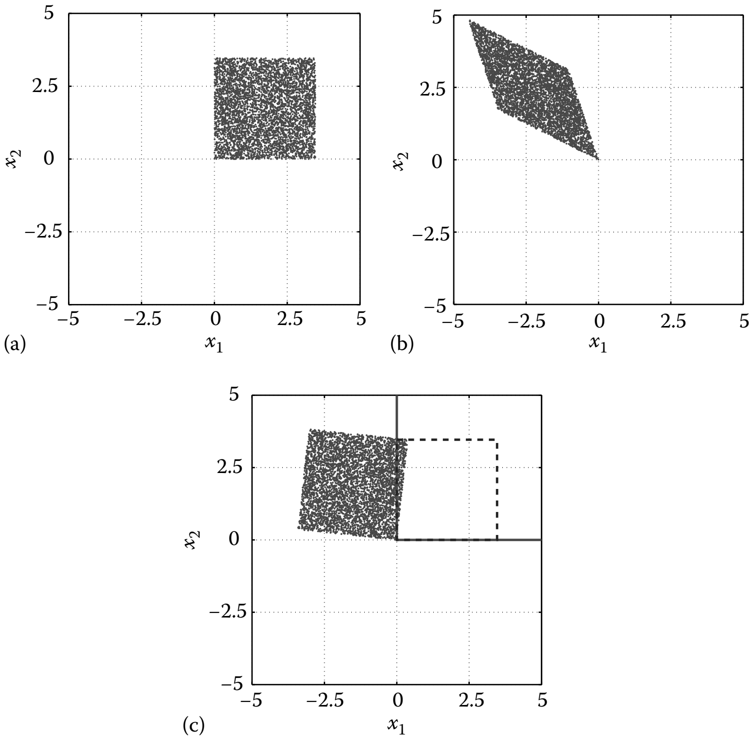


FIGURE 6.6
Distributions in the context of nonnegative source separation. (a) Sources distribution, (b) observations distribution, and (c) distribution after whitening step.

rotation matrix that restores the nonnegativity condition of the signals. Some algorithms exploring this idea can be found in [218,242].

6.4.3 Sparse Component Analysis

SCA is based on the hypothesis that the sources present, in some domain, a significant degree of sparseness. The concept of sparseness is not a consensus but, for our purposes, it suffices to consider that sparse sources are signals that assume values close to zero most of the time [45], as depicted in Figure 6.7. The great interest in this approach lies in the fact that it makes possible to identify the mixing matrix, or even to separate the sources, in an underdetermined context [120,182] as discussed in the sequel.

Let us consider the example presented in Figure 6.8, in which is shown the joint distribution of the observations in a scenario with 3 sources and only 2 sensors. Figure 6.8a brings the sample distribution for uniform sources, while in Figure 6.8b, the sources are sparse.

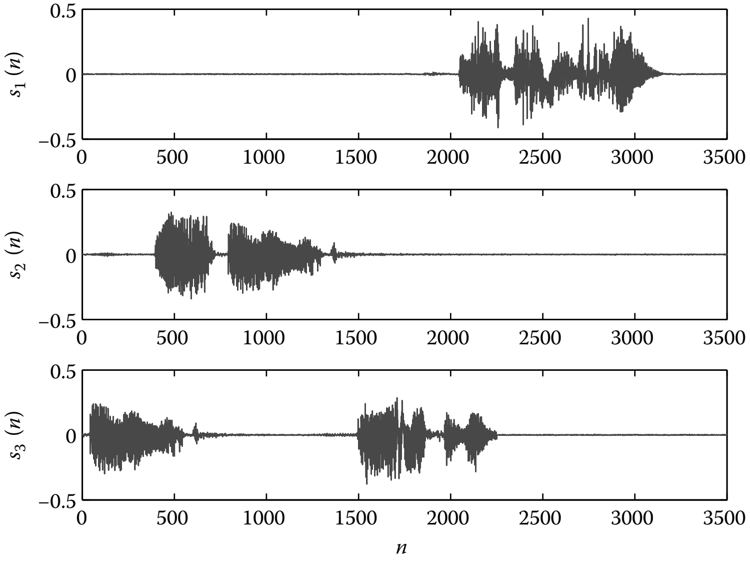


FIGURE 6.7
Some examples of sparse sources.

Due to sparseness, it is likely that only one of the sources presents a value considerably larger than zero at each time instant. Hence, the observations tend to be

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \end{bmatrix} \begin{bmatrix} s_1 \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix} \approx \mathbf{a}_1 s_1 \tag{6.89}$$

where
$\mathbf{a}_i$ represents the ith column of $\mathbf{A}$
ε represents a small value

Thus, the observed data will tend to concentrate along the directions defined by the columns of the mixing matrix, as indicated in Figure 6.8b.

In this situation, in order to identify the mixing matrix, one should look for methods that estimate these directions, like clustering methods [141]. Another interesting point is that it should also be possible to identify, without a priori information, the number of sources in a mixture by blindly estimating the number of clusters [18, 19].

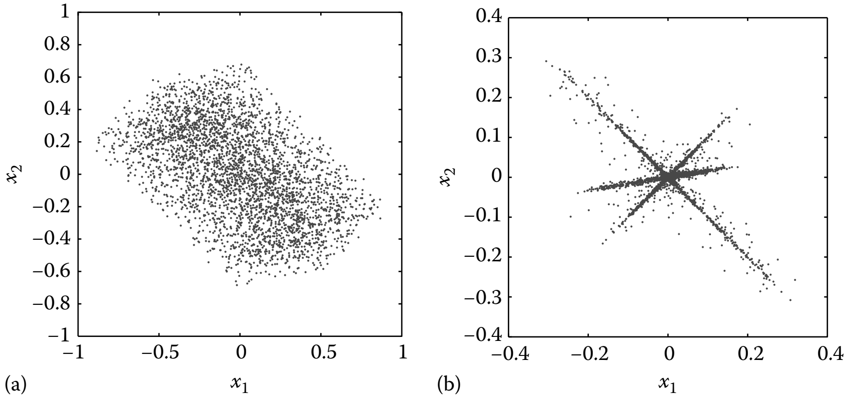


FIGURE 6.8
Underdetermined mixtures (three sources and two sensors): (a) nonsparse and (b) sparse sources.

Even though the discussion so far considered only sparse sources in the time domain, SCA can be applied more generally on data that are sparse in some domain. For instance, in audio applications, the signals are often sparse in the time–frequency domain. Thus, one should be employ a linear transform like the short-time Fourier or a wavelet transform prior to processing the data [120,216,258].

6.4.4 Bayesian Approaches

The Bayesian approach for source separation [168, 298, 299] also takes advantage of prior information about the sources to perform the required estimation. In this case, the notion of using a priori information is in fact inherent to the methodology, since the Bayesian approach consists in mixing the a posteriori probability given by $p(\mathbf{A}, \mathbf{s}|\mathbf{x})$. Thus, using Bayes' rule we have

$$p(\mathbf{A}, \mathbf{s}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{A}, \mathbf{s})p(\mathbf{A})p(\mathbf{s}) \tag{6.90}$$

where
$p(\mathbf{s})$ and $p(\mathbf{A})$ denote, respectively, the pdf of the sources and the mixing matrix coefficients
$p(\mathbf{x}|\mathbf{A}, \mathbf{s})$ is the data likehood function [165,168]

From (6.90), we can notice that this approach is extremely flexible, since any additional information about the problem at hand can be included in the optimization criterion through density priors. Such additional information may be related to the pdf of the elements in the mixing matrix or to the pdf of the sources. In this sense, the Bayesian approach may also be used with more elaborated mixing models, like the case with noise and the underdetermined mixing model [104,299,314].

On the other hand, the maximization of (6.90) is a great challenge, since the function being optimized is, in general, non-convex, and the number of parameters is usually large.

6.5 Convolutive Mixtures

The linear and instantaneous mixing model has been employed in several practical and theoretical instances [148]. However, there are situations in which the mixing process is better formulated in terms of a linear system with memory, which gives rise to the linear convolutive mixture model. Typical examples are found in audio signal processing, in which the signals are recorded using a set of microphones in a reverberating environment [210], and also in wireless communication systems.

Signal propagation from each source to each sensor is modeled by a transfer function, as illustrated in Figure 6.9. In this case, the received signal can be expressed by

$$\begin{aligned} x_i\left(k\right) &= \sum_j \sum_\tau a_{ij}(\tau) s_j\left(k-\tau\right) \\ &= \sum_j a_{ij}(n) * s_j\left(n\right) \end{aligned} \tag{6.91}$$

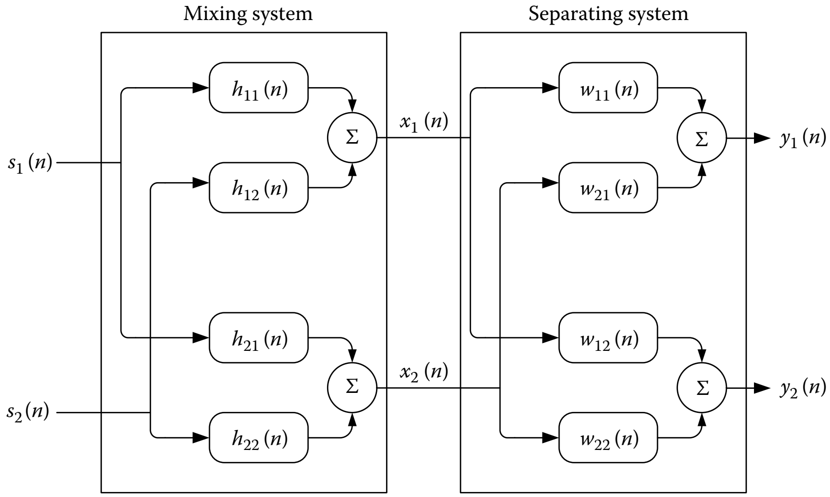


FIGURE 6.9
Convolutive mixture model with two sources and two sensors.

for $i = 1, \ldots, N$, where $a_{ij}(n)$ denotes the impulse response of the channel linking the ith source to the jth sensor. Signal separation is then achieved by means of a set of filters $w_{ij}(n)$, as indicated in Figure 6.9.

Since the mixing model as well as the separating system are MIMO systems, the notation presented in Chapter 5 can also be used. Thus, the system transfer function is given by

$$\mathbf{A}[z] = \mathbf{A}(0) + \mathbf{A}(1)z^{-1} + \cdots + \mathbf{A}(L-1)z^{-L+1} \tag{6.92}$$

and the observations and estimated signals can be expressed respectively by

$$\mathbf{x}(k) = \sum_{l} \mathbf{A}(l)\mathbf{s}(k-l) \tag{6.93}$$

and

$$\mathbf{y}(k) = \sum_{l} \mathbf{W}(l)\mathbf{x}(k-l) \tag{6.94}$$

where we assumed both mixing and separating systems as linear feedforward structures. It is worth mentioning that other structures, like those with a feedback loop, can also be considered [159].

Let us now consider two important classes of methods that can be applied to solve the problem of separation of linear convolutive mixtures.

6.5.1 Source Separation in the Time Domain

The methods based on the principle of ICA can be extended to the convolutive case [311]. Therefore, assuming that all sources are mutually independent, the separating system must be designed to yield signals as independent as possible from each other.

It is important to mention that in the case of instantaneous mixtures, independence between two estimated signals $y_1(n)$ and $y_2(n)$ is ensured by the independence between the random variables associated with time instant n. When dealing with convolutive mixtures, the problem becomes more involved: due to the presence of memory in the mixing process, we must ensure statistical independence between two stochastic processes, i.e., $y_1(n)$ and $y_2(n-l)$ must be independent for all n and l.

If the mutual information is employed as a measure of independence, the cost function to be optimized in the convolutive case is

$$J_{\text{Convolutive}}(\mathbf{w}(n)) = \sum_{m} I\left(y_1(k), y_2(k-m)\right) \tag{6.95}$$

An important aspect is related to the ambiguities of the solution. As in the instantaneous case, separation of convolutive mixtures also exhibits a permutation indeterminacy, which is only solved if additional information about the sources is provided. Nonetheless, the amplitude ambiguity is now replaced by a filtering indeterminacy. This occurs because if $y_1(n)$ and $y_2(n)$ are mutually independent processes, so will be $\hat{y}_1(n) = h_1(n) * y_1(n)$ and $\hat{y}_2(n) = h_2(n) * y_2(n)$, for any invertible filters $h_1(n)$ and $h_2(n)$. Therefore, unless additional information about the original sources is available, the method is susceptible to providing distorted versions of the sources.

A number of different methods based on a time-domain approach can be found in the literature. An extension of Hérault and Jutten's algorithm is proposed in [159], in which the coefficients of the separating structure, as shown in Figure 6.5, are replaced by linear filters. Other methods have been modified to cope with the convolutive mixture model, like the natural gradient algorithm [10,11], Infomax [290,291], as well as the FastICA [285].

6.5.2 Signal Separation in the Frequency Domain

Another class of methods for the case of convolutive mixtures deals with separation in the frequency domain [198,276].

The observed signals $x_i(n)$, sampled at a frequency f_s, can be represented in the frequency domain $\bar{x}_i(f,\tau)$ by means of a short-time Fourier transform (STFT) with a finite number of points, i.e.,

$$\bar{x}_i(f,\tau) = \sum_{l=-\frac{L}{2}}^{\frac{L}{2}+1} x_i(\tau + l)\, w_{\text{window}}(l)\, e^{-j2\pi f l} \tag{6.96}$$

where

$f \in \left\{0, \frac{1}{L}f_s, \ldots, \frac{L-1}{L}f_s\right\}$ denotes the set of frequency bins
$w_{\text{window}}(n)$ is a windowing function
τ represents a time index

Therefore, the observed signal can be approximately represented as the result of a linear instantaneous mixture in a frequency bin, i.e.,

$$\bar{x}_i(f,\tau) = \sum_j \bar{a}_{ij}(f)\, \bar{s}_j(f,\tau) \tag{6.97}$$

where

$\bar{a}_{ij}(n)$ is the frequency response between the jth source and the ith sensor
$\bar{s}_j(f,\tau)$ denotes the STFT of the jth source

Finally, organizing (6.97) into matrix notation we get to

$$\bar{\mathbf{x}}(f,\tau) = \bar{\mathbf{A}}(f)\, \bar{\mathbf{s}}(f,\tau) \tag{6.98}$$

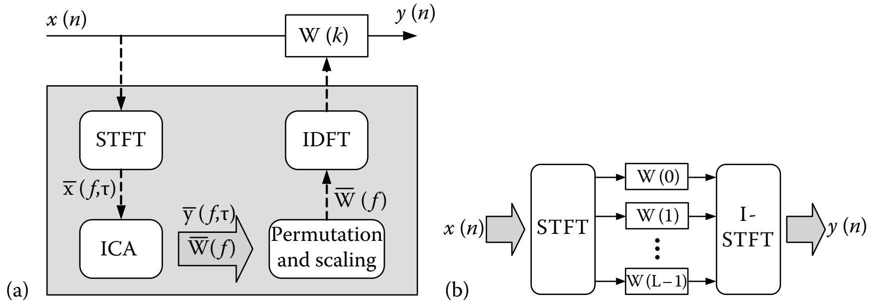


FIGURE 6.10
Strategies for source separation in the frequency domain: (a) separation in the frequency domain and (b) filter design in the frequency domain.

Hence, the convolutive problem is converted into an equivalent problem composed of a set of linear instantaneous mixtures, for which several methods have been presented in this chapter. Figure 6.10 brings two possible strategies that explore this principle. In Figure 6.10a, separation is performed in the frequency domain and then the signals are transformed back to the time domain, while, in Figure 6.10b, only the separating filters are obtained in the frequency domain, but separation itself is performed in the time domain.

Since separation is carried out independently in each frequency bin, each solution is subject to arbitrary scaling and permutation ambiguity, i.e.,

$$\mathbf{y}(f,\tau) = \mathbf{P}(f)\boldsymbol{\Lambda}(f)\mathbf{s}(f,\tau) \tag{6.99}$$

where
$\mathbf{P}(f)$ accounts for the permutation
$\boldsymbol{\Lambda}(f)$ for the scaling ambiguities

The amplitude indeterminacy is the origin of the filter ambiguity mentioned earlier. In general, this ambiguity can only be solved if additional information is available. An example of this condition occurs in digital communications, where the distribution of the sources is, in most of the cases, known a priori, and the samples can be considered to be i.i.d. Even if the sources cannot be perfectly recovered, it is usual, specially in audio applications, to employ the minimum distortion principle [202, 203] in order to determine the separation filters and reduce the distortion due to the inherent filtering indeterminacy.

The second indeterminacy, related to the permutation of the recovered signals, may also cause a severe distortion. Since the solution of ICA in each frequency bin is subject to a permutation ambiguity, if the permutations are not consistent over all frequency bins, the reconstruction of the signals in the time domain will yield estimates that are composed of different sources in each frequency bin. Therefore, an additional procedure to perform the spectral alignment is required prior to the signal recovery in time domain.

The solutions proposed in the literature explore particular characteristics of the signals and of the separating matrices to perform this adjustment. An example would be to assume that the transition for separating matrices in frequency domain is smooth, i.e., there is a small variation between coefficients of matrices $\bar{\mathbf{A}}(f_{k-1})$ and $\bar{\mathbf{A}}(f_k)$. Therefore, the permutation problem could be solved by determining, for each frequency bin f_k, the permutation which leads to a minimum distance between the estimated mixing matrix $\bar{\mathbf{A}}(f_k)$ and the estimated mixing matrix $\bar{\mathbf{A}}(f_{k-1})$ from the previous frequency bin [29].

Another method is effective when the source signals arrive at the sensor array from different directions. Using additional information about the sensor array geometry, it is possible to estimate the direction of arrival (DOA) of the signals, which is then employed to perform the spectral alignment by clustering the components according to the DOA [152, 171].

Other additional information can also be employed, like the spectra of the sources [151, 268]. For natural sounds, for instance, spectral components close to each other can be strongly correlated, and this fact can be explored to indicate how the components must be permutated in order to correctly reconstruct the signals. This approach, allied with spatial information brought by the sensor array geometry, can lead to methods with better results [261].

6.6 Nonlinear Mixtures

In this section, we will discuss the BSS problem for nonlinear mixture models, in particular the PNL model, for which the ICA background can still be applied.

The general form of a memoryless nonlinear mixture is given by a mapping $\mathbf{F} : \Re^N \rightarrow \Re^M$ such that

$$\mathbf{x}(n) = \mathbf{F}(\mathbf{s}(n)) \tag{6.100}$$

Therefore, our goal is to obtain another mapping

$$\mathbf{y}(n) = \mathbf{G}(\mathbf{x}(n)) \tag{6.101}$$

such that $\mathbf{y}(n)$ be a precise estimate of the sources, $\mathbf{G}(\cdot)$ being a specially tailored nonlinear mapping.

6.6.1 Nonlinear ICA

A tempting solution consists in extending ICA methods that are efficient in the linear case. Under the same hypothesis of independence between the sources, the separating structure must be adjusted to recover independent signals. This approach was given the name of nonlinear ICA.

Consider a mapping $\mathcal{H}(\mathbf{s}) = \mathbf{G}(\mathbf{F}(\mathbf{s}))$, which represents the composition between mixing and separating mappings, such that

$$\mathcal{H}(\mathbf{s}) = \left[h_1(s_{\rho(i)}), \ldots, h_n(s_{\rho(i)})\right] \tag{6.102}$$

where

h_i denotes an arbitrary nonlinear function
ρ denotes a permutation operator over $\{1, 2, \ldots, n\}$

This particular mapping is known as a trivial mapping, and transforms a vector of independent components into another vector with the same property, since $\mathcal{H}(\mathbf{s})$ represents a set of functions of single components of $\mathbf{s}$ that are mutually independent. Therefore, a mapping $\mathbf{G}(\cdot)$ that leads to an overall trivial mapping $\mathcal{H}(\mathbf{s})$ is also a nonlinear ICA solution. Notice that, in this case, there are ambiguities related to permutation, and also a residual nonlinear distortion due to h_i.

Nonlinear ICA solutions are not restricted to trivial mappings, and hence, do not ensure the separation. A very simple example of this condition is shown in Example 6.2 [157].

Example 6.2

Let us consider two independent sources s_1 and s_2, with pdf $p_{s_1}(s_1) = s_1 \exp\left(-s_1^2/2\right)$, i.e., a Rayleigh distribution, and $p_{s_2}(s_2) = 2/\pi$, a uniform pdf over $s_2 \in [0, \pi/2)$.

The joint pdf of s_1 and s_2 is expressed by

$$p_{s_1 s_2}(s_1, s_2) = \begin{cases} \frac{2}{\pi} s_1 \exp\left(\frac{-s_1^2}{2}\right), & s_2 \in [0, \pi/2) \\ 0, & s_2 \in (-\infty, 0) \text{ or } s_2 \in [\pi/2, +\infty) \end{cases} \tag{6.103}$$

Consider now the following mapping $\boldsymbol{\mathcal{H}}$:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \boldsymbol{\mathcal{H}}(\mathbf{s}) = \begin{bmatrix} s_1 \cos(s_2) \\ s_1 \sin(s_2) \end{bmatrix} \tag{6.104}$$

Since the Jacobian of the transformation is not diagonal, $\mathbf{y}$ will represent a mixture of the elements of $\mathbf{s}$. Now, let us express the joint pdf of $\mathbf{y}$ [230]:

$$p_{y_1 y_2}(y_1, y_2) = \frac{p_{s_1 s_2}(s_1, s_2)}{|\det \mathbf{J}_{\boldsymbol{\mathcal{H}}}(\mathbf{y})|} \tag{6.105}$$

where $\mathbf{J}_{\mathcal{H}}(\mathbf{y})$ corresponds to the Jacobian of $\mathcal{H}(\cdot)$, which is given by

$$|\det \mathbf{J}_{\mathcal{H}}(\mathbf{y})| = \left|\det\left(\begin{bmatrix} \cos(s_2) & \sin(s_2) \\ s_1\sin(s_2) & -s_1\cos(s_2) \end{bmatrix}\right)\right|$$
$$= |-s_1(\cos^2(s_2)+\sin^2(s_2))| = |s_1| \tag{6.106}$$

Substituting (6.103) and (6.106) in (6.105), and noticing that $s_1^2 = y_1^2 + y_2^2$, we obtain

$$p_{y_1y_2}(y_1, y_2) = \frac{2}{\pi}\exp\left(\frac{-(y_1^2+y_2^2)}{2}\right) = \left(\sqrt{\frac{2}{\pi}}\exp\left(\frac{-y_1^2}{2}\right)\right)\left(\sqrt{\frac{2}{\pi}}\exp\left(\frac{-y_2^2}{2}\right)\right) \tag{6.107}$$

Thus, we notice that $\mathbf{y}$ is composed of independent components—since its pdf is the product of functions of the pdf of y_1 and y_2—even though $\mathbf{y}$ represents a mixture of s_1 and s_2.

The above example shows that it is possible to obtain nonlinear mixing mappings that preserve independence. This result was first observed by Darmois back in 1951, in a nonlinear factor analysis context [281]. In the context of BSS, though, this problem was studied in [150].

In a certain sense, we could argue that the main difficulty is due to the great flexibility of nonlinear mappings. Since there is no reference signal to guide the adaptation of the separating system, we saw that it is possible to recover independent signals without separating the sources. Thus, in order to avoid these undesirable solutions, the existing algorithms for the nonlinear case try to restrict the degree of flexibility of these mappings. In this context, a possible approach to solve the nonlinear mixing problem is to consider nonlinear mappings to which all theoretical backgrounds developed for the linear case are still valid. In such case, even though the developed solutions are restricted to a smaller number of practical scenarios, several existing tools remain applicable.

A first attempt in this direction was made in [6, 310], in which only mild nonlinearities—provided by a multilayer perceptron—were considered. Nonetheless, even under this assumption, it is possible to obtain undesirable solutions [200], indicating that other constraints must be taken into account.

6.6.2 Post-Nonlinear Mixtures

Taleb and Jutten introduced the PNL model in [280], which is depicted in Figure 6.11.

The PNL model is particularly useful when the sensors present some sort of nonlinear distortion. Mathematically, the observed signals $\mathbf{x}$ are given by

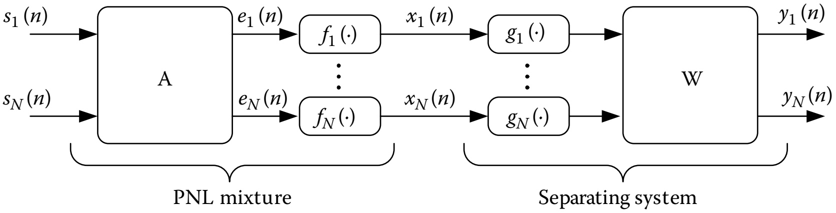


FIGURE 6.11
Post-nonlinear model.

$$\mathbf{x} = \mathbf{f}(\mathbf{As}) \tag{6.108}$$

where $\mathbf{f}(\mathbf{e}) = \left[f_1(e_1), f_2(e_2), \ldots, f_N(e_N)\right]$.

The natural choice for the separating system would be a similar structure, composed of a set of nonlinearities $g_i(\cdot)$ and a separating matrix $\mathbf{W}$, as illustrated in Figure 6.11. The output of this structure is expressed by

$$\mathbf{y} = \mathbf{Wg}(\mathbf{x}) \tag{6.109}$$

where $\mathbf{g}(\mathbf{x}) = \left[g_1(x_1), g_2(x_2), \ldots, g_N(x_N)\right]$.

The conditions under which the PNL model is separable using ICA can be stated in terms of the following theorem [3,281].

THEOREM 6.4 (Separability of the PNL Model)

Consider the PNL model depicted in Figure 6.11, and the following hypotheses:

- Matrix $\mathbf{A}$ is invertible and effectively mixes the sources, i.e., there are at least two nonzero elements in each row and column.
- $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are monotonic functions; consequently, $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ will also be monotonic.
- There is, at most, one Gaussian source.
- The joint pdf of the sources is differentiable and its derivative is continuous over all its support.

Under these conditions, if the elements of $\mathbf{y}$ are statistically independent, then $\mathbf{h}(\cdot)$ will be composed of linear functions and $\mathbf{WA} = \mathbf{\Lambda P}$. ■

Thus, in addition to the usual conditions for separability for a linear mixing model, it is required that the nonlinearities be monotonic functions, which implies that they are invertible.

6.6.3 Mutual Information Minimization

Since ICA can still be applied to this case, mutual information will be a valid criterion for source separation within the PNL model. Based on (6.24), the mutual information of $\mathbf{y}$ can be defined as

$$I(\mathbf{y}) = \sum_{i=1}^{N} H(y_i) - H(\mathbf{y}) \tag{6.110}$$

where $H(y_i)$ is the marginal entropy of the ith element of $\mathbf{y}$.

From Figure 6.11, the joint entropy $H(\mathbf{y})$ can be expressed in terms of the joint entropy of $\mathbf{x}$:

$$H(\mathbf{y}) = H(\mathbf{x}) + E\left\{\log |J|\right\} \tag{6.111}$$

where J represents the Jacobian of the nonlinear transformation. Applying this result to (6.110), one can obtain [281]

$$I(\mathbf{y}) = \sum_{i=1}^{N} H(y_i) - H(\mathbf{x}) - \log\left|\det \mathbf{W}\right| - E\left\{\log \prod_{i} \left|g_i'(x_i)\right|\right\} \tag{6.112}$$

where $g_i'(\cdot)$ denotes the derivative of $g_i(\cdot)$. It is important to point out that this relation is only valid for invertible nonlinearities $g_i(\cdot)$.

In order to minimize (6.112) a gradient-based algorithm was proposed in [281]. It follows that

$$\frac{\partial I(\mathbf{y})}{\partial \mathbf{W}} = -E\{\Psi(\mathbf{y})\mathbf{e}^T\} - (\mathbf{W}^T)^{-1} \tag{6.113}$$

where

$$\Psi(\mathbf{y}) = [\psi_{y_1}(y_1), \ldots, \psi_{y_N}(y_N)] \tag{6.114}$$

for

$$\psi_{y_i}(y_i) = \left(\frac{p_{y_i}(y_i)'}{p_{y_i}(y_i)}\right) \tag{6.115}$$

which is called the score function.

In order to obtain the update rule for the nonlinear section of the separating system, it is necessary to consider parametrized nonlinear functions

$g_i(\theta_i, x_i)$, $i = 1, \dots, N$, with θ_i denoting the parameters. The gradient with respect to θ_i will be given by

$$\frac{\partial I(\mathbf{y})}{\partial \theta_i} = -E\left\{\frac{\partial \log\left|g_i'(\theta_i, x_i)\right|}{\partial \theta_i}\right\} - E\left\{\left(\sum_{k=1}^{N} \psi_{y_k}(y_k) w_{ki}\right)\frac{\partial g_i(\theta_i, x_i)}{\partial \theta_i}\right\} \tag{6.116}$$

where w_{ki} corresponds to the element (k, i) of $\mathbf{W}$.

The complete derivation of the algorithm depends on the structure of the parametric nonlinear functions $g_i(\theta_i, x_i)$. The approach taken in [281] was to employ artificial neural networks, a flexible nonlinear structure to be discussed in Chapter 7.

Another point worth mentioning is that the derivatives of $I(\mathbf{y})$ with respect to $\mathbf{W}$ and to the parameters θ heavily depend on the score functions $\psi_{y_i}(y_i)$. Unfortunately, since we are dealing with a blind separation problem, these functions are unknown, and must be estimated from the output vector $\mathbf{y}$. The complete description of the algorithm can be found in [281].

6.6.4 Gaussianization

The PNL model clearly presents two distinct sections, one linear and the other nonlinear. This characteristic suggests that the parameters of the separating structure could be separately adapted: the Gaussianization approach explores this idea.

The Gaussianization method [277, 315] relies on the hypothesis that linearly mixed signals tend to a Gaussian distribution. This hypothesis is derived from the central limit theorem, as the mixed signals are linear combinations of independent sources. Once these linearly mixed signals are subject to a nonlinearity, the resulting observations will certainly not present a Gaussian distribution. Figure 6.12 illustrates this idea, showing the distribution of a linear mixture of 10 signals with uniform pdf and the result after a nonlinearity $f(x) = \tanh(x)$.

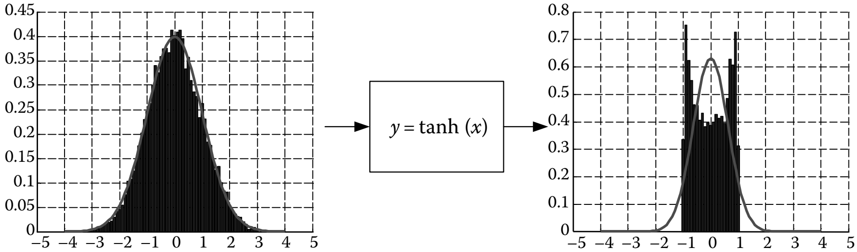


FIGURE 6.12
Illustration of the idea underlying Gaussianization approaches.

It is clear that the nonlinearly distorted signal does not present a Gaussian distribution anymore. Thus, considering the PNL model, in order to obtain the separating system, first the nonlinearities $g_i(\cdot)$ should be adjusted so that the signals after each nonlinearity $g_i(\cdot)$ present a Gaussian distribution, which can be accomplished by

$$g_i(\cdot) = F_{Gauss}^{-1}(F_{x_i}(\cdot)) \tag{6.117}$$

where

$F_{x_i}(x_i)$ denotes the cumulative distribution function of x_i

$F_{Gauss}^{-1}(\cdot)$ the inverse of a cumulative distribution function of a normalized Gaussian, i.e., the quantile function

Once the observations have been "Gaussianized," they should represent a linear mixture of the sources, and hence, the sources could be recovered by using any ICA method developed for linear mixtures.

The results of this approach heavily depend on the validity of the Gaussianity assumption, which is sounder when a large number of sources is present in the mixture. However, in situations in which this assumption is not representative, there will be a considerable distortion in the transformed data, and ICA methods will not be efficient.

6.7 Concluding Remarks

In this chapter, we studied the instigating problem of BSS. As a natural progression on the theme of blind equalization, BSS widens the scope of our study in terms of theoretical tools as well as of potential of practical applications.

We first presented the problem in its general form before reaching the linear and instantaneous mixture model, which was a reference in the ensuing discussion. The starting point of this discussion was ICA, a widespread BSS tool with major theoretical and historical importance and a significant degree of applicability.

In order to discuss some fundamental criteria to perform ICA, we introduced important concepts and definitions. BSS makes use of important notions, issues from information theory like mutual information and negentropy, the definitions of which have been stated. From there, we enounced the Infomax principle. We discussed on about the role of high-order statistics in BSS and described the procedure of whitening preprocessing and the notion of nonlinear decorrelation.

Based on the discussed criteria, we presented some well-established algorithms to perform ICA: Hérault–Jutten, Infomax algorithm, the nonlinear PCA, JADE, EASI, and the FastICA algorithm.

It was also important to present some alternative paths toward solving the BSS problem, as the incorporation of a priori knowledge of multiple natures. In this sense, we presented BSS methods exploring the fact that the power spectral densities of the sources are mutually distinct. Two other important methods we showed were those based on nonnegativity (nonnegative ICA) and sparsity (SCA) properties of the source signals. The Bayesian approach was also discussed.

Finally, we considered two extensions of the linear and instantaneous model: convolutive mixtures and nonlinear mixtures. To deal with the first one, can employ methods derived from the linear and instantaneous case, as long as we take into account the so-called filtering indeterminacy, which generally requires additional information about the original sources. Methods for convolutive source separation in both time and frequency domains were discussed. Concerning nonlinear mixtures, we discussed limitations of nonlinear ICA and introduced the so-called PNL model, which allows the efficient use of certain separation approaches, like those based on mutual information minimization and on Gaussianization.

It is worth emphasizing that source separation is a theme that places itself at the intersection of many science and engineering disciplines and applications. For this reason, different and rather rich standpoints about theoretical and practical aspects are available in the existing literature. In particular, important works like [12, 76, 137, 148, 156] must be considered in order to attain a more complete view on the subject. Our purpose in this chapter is signal processing oriented, and aims to show BSS as an essential area in unsupervised signal processing theory, as well as to put together, in a synthetic way, a number of definitions, criteria, and algorithms that can be useful in the study and/or research of different kinds of readers.

7

Nonlinear Filtering and Machine Learning

In the previous chapters, we generally considered linear structures to perform both supervised and unsupervised tasks. In spite of their interesting trade-off between mathematical tractability and performance, linear structures may be unsatisfactory in some problems, as in the previously discussed case of nonlinear mixtures separation. In fact, a gradual emergence of more complex signal processing scenarios leads to an increasing interest in the use of *nonlinear filtering structures*, which are able to provide more flexible input–output mappings.

As far as the equalization problem is concerned, the essential goal is the correct recovery of the transmitted symbols. As already commented, the fact that these symbols belong to a finite alphabet leads to the need of a nonlinear decision device at the receiver. Moreover, as a matter of fact, the effectiveness of the transmission link must be evaluated in terms of symbol *error rate* (SER). As indicated in Chapter 1, the equalizer we obtain in accordance with a criterion of minimum SER is the *Bayesian equalizer* [2,68], which is essentially a nonlinear device.

Along the previous chapters, we studied suboptimal, and more treatable, criteria to provide equalization, since the complexity in implementing the Bayesian solution tends to be prohibitive. Such criteria may also be applied in association with a nonlinear structure to improve the capability of correct retrieval. This is the case, for instance, of a nonlinear solution intensively applied in the area of adaptive equalization: the *decision-feedback equalizer* (DFE). As discussed in this chapter, this ingenious approach is a nonlinear structure specially tailored in view of the practical problem of channel equalization.

Nevertheless, equalization can be studied in a more general theoretical framework of nonlinear filtering. In this sense, the suitable methodology consists in fitting equalization in the context of classification task. At this point, our problem of interest touches the rather large field of machine learning, since from this standpoint, solutions based on neural networks and related approaches will emerge.

The above considerations justify our motivation in bringing together, in a specific chapter, some approaches we associate to the notions of nonlinear filtering and machine learning. Both themes are really vast, and to present a detailed analysis of all nonlinear filtering and/or machine learning approaches, with several up-to-date references in all possible contexts, is clearly out of the focus of the book. So, the objective of this chapter is to provide an overview of some representative structures and algorithms, with a clearly specific regard on our problems of interest. Hence, the chapter is organized as follows:

- In Section 7.1, we discuss the DFE. The DFE is a very efficient device for performing deconvolution whenever one deals with digital signals. In a certain sense, we may consider DFE as a historical link between linear and nonlinear approaches in equalization, which justifies our choice for it as the starting point of the chapter. An alternative solution proposed for blind equalization, the so-called predictive DFE, is also presented.
- In Section 7.2, we turn our attention to a class of nonlinear devices based on polynomial expansions: the *Volterra filters*. These filters are based on polynomial expansions and have, as generic approximators, a wider scope of application than that associated with the DFE. On the other hand, they also preserve points of contact with the linear filtering theory as far as the parameter optimization is concerned.
- In Section 7.3, we analyze the problem of *digital equalization as a classification task*. As mentioned above, this point of view leads us to search for synergy with machine learning-based solutions. In addition, this formulation also allows the formal derivation of the Bayesian equalizer, which is our fundamental reference of optimality.
- In Section 7.4, we present the two main neural approaches to be considered in our applications: the *multilayer perceptron* (MLP) and the *radial-basis function* (RBF) network. In addition to that, we state the similarities between the MLP and the Bayesian filter.

Historical Notes

The proposal of the DFE is attributed to Austin in [21]. From this time, a really significant number of articles, patents, and book chapters have dealt with the subject. As an example for reference, we can mention the interesting surveys in [30, 246]. Later, a number of works has been devoted to the use of blind criteria for DFE updating, among which we can mention, e.g., [61].

The use of polynomial filters was also an important initial step toward a more widespread adoption of nonlinear models in signal processing applications. Some pioneer efforts in the context of equalization and echo cancellation, for instance, can be traced to the 1970s and 1980s [5,39,43,107,284]. The proposal of iterative learning algorithms in the context of polynomial filters has also been a prolific field of research in the last decades (see [201] for a more detailed account of the subject).

A complete historical analysis of the development of the neural approaches is beyond the scope of the book, but it is worth presenting a brief overview on the subject; a more detailed view can be found, for instance, in [136]. The seminal paper by McCulloch and Pitts [206] established the most relevant relationship between logical calculus and neural network computation and gave rise to a mathematical model of the neuron. In 1949, another decisive step was taken when Donald Hebb's book [142], *The Organization of Behavior*, related the learning process to synaptic modifications controlled by the firing patterns of interrelated neurons.

In 1958, Frank Rosenblatt proposed a neural network for pattern recognition that was called perceptron. The perceptron was, in simple terms, an adaptive linear classifier whose learning algorithm was founded on a beautiful mathematical result, the perceptron convergence theorem, which ensured proper operation for linearly separable patterns. Interestingly, in 1960, Widrow and Hoff used the LMS algorithm to adapt the parameters of a perceptron-like structure, originating the Adaline (adaptive linear element).

After a period in which the interest for research in neural networks suffered a decrease—in spite of important contributions related, for instance, to self-organization [136]—the field experienced a sort of revival with proposals like the Hopfield network and the backpropagation algorithm (BPA) [256, 257, 302], a fundamental result to the effective application of MLPs. Later, in 1988, Broomhead and Lowe [48] were responsible for introducing a new multilayer structure, the RBF network, which is also considered a fundamental neural approach for function approximation.

The revival of the field of neural networks in the 1980s led to a significant interest for the use of these structures in the context of signal processing. A consequence of this tendency was the popularization of the view of the equalization problem as a classification task [68]. As a matter of fact, a huge number of works have applied neural networks in the context of equalization. Among them, it is relevant to mention the seminal papers written by Gibson and Cowan (1990) [122], Theodoridis et al. (1992) [282], and Chen et al. [68, 69].

7.1 Decision-Feedback Equalizers

In digital transmission, a classical nonlinear solution to the equalization problem is the DFE, first proposed by Austin in 1967 [21]. The structure is composed of two filters, a feedforward filter (FFF) and a feedback filter (FBF), which are combined in a manner that also includes the action of a decision device. Figure 7.1 brings a scheme of the structure.

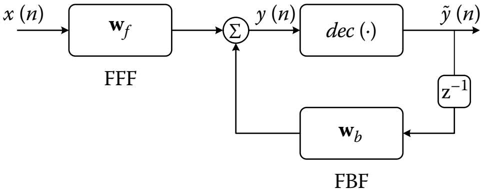


FIGURE 7.1
Structure of a DFE.

The operation of the DFE can be understood in terms of the interaction between the effect of the FFF, the assumption of correct decisions, and the use of the previously recovered symbols to cancel intersymbol interference via the FBF. Mathematically, the input–output relationship of a DFE is given by

$$y(n) = \mathbf{w}_f^T \mathbf{x}(n) + \mathbf{w}_b^T \hat{\mathbf{y}}(n-1) \tag{7.1}$$

where $\hat{\mathbf{y}}(n-1) = \left[dec\left(y(n-1)\right), dec\left(y(n-2)\right), \ldots\right]$.

Conceptually, the DFE behaves as a suboptimal alternative to an equalizer based on the idea of maximum-likelihood sequence estimation (MLSE) [111], being particularly attractive, in terms of computational effort, for channels with long impulse responses.

An important remark is that the structure of the DFE is extremely well-suited to deal with the problem of intersymbol interference removal. This can be clearly understood if we consider, for instance, a case in which a binary signal (+1 or −1) is sent through a noiseless FIR channel with parameter vector $\mathbf{h} = [1 \;\; 0.9 \;\; 0.8]^T$. Even if we consider the FFF to be simply a unit gain, assuming that the FBF has coefficients $\mathbf{w}_b = [-0.9 \;\; -0.8]^T$ and that the past decisions led to correct estimates of $s(n-1)$ and $s(n-2)$, we have that the filter output will originate a correct estimate of $s(n)$. This behavior is clearly similar to that of a linear IIR filter, but the existence of the decision device inside the loop of the DFE makes a crucial difference: it creates a saturation effect that prevents the unavoidable instability we have if a linear structure is used to invert a nonminimum-phase channel.

Even for minimum-phase channels, the decision device in the feedback loop avoids the well-known problem of noise enhancement. This problem emerges in noisy channels with spectral nulls, as the use of a linear structure poses a dilemma: in terms of ISI reduction, it should be useful to have a frequency response with significant gains to compensate the spectral nulls; however, this tends to produce a more substantial noise power at the equalizer output. In its turn, due to its nonlinear character, the DFE is not subjected to amplification or attenuation framework that characterizes the convolution-based behavior of linear equalizers. Another relevant fact is that whenever the decision device performs a correct estimate, no noise parcel is fed back, which decisively contributes to mitigate any enhancement effect.

Several learning strategies can be applied to adapt the parameters of a DFE. It is possible to implement both supervised algorithms, as those presented in Chapter 3 [30], and unsupervised techniques, like Bussgang algorithms [234].

In the supervised context, it is possible to perform parameter adaptation by employing a standard LMS algorithm for both filters:

$$\mathbf{w}_f(n+1) = \mathbf{w}_f(n) + \mu e(n)\mathbf{x}(n) \tag{7.2}$$

$$\mathbf{w}_b(n+1) = \mathbf{w}_b(n) + \mu e(n)\hat{\mathbf{y}}(n-1) \tag{7.3}$$

$e(n) = d(n) - y(n)$ being a standard error signal. Notice that the algorithm treats the past decisions as conventional input signals, without highlighting any dependence with respect to the equalizer weights, which evokes ideas underlying equation-error methodologies in IIR filtering [249, 274].

An analogous approach can be followed to apply an unsupervised algorithm. The expressions, for instance, for the constant modulus algorithm (CMA) are similar to those presented in Section 4.3 [121]:

$$\mathbf{w}_f(n+1) = \mathbf{w}_f(n) + \mu y(n)[R_2 - |y(n)|^2]\mathbf{x}(n) \tag{7.4}$$

$$\mathbf{w}_b(n+1) = \mathbf{w}_b(n) + \mu y(n)[R_2 - |y(n)|^2]\hat{\mathbf{y}}(n-1) \tag{7.5}$$

Finally, it is important to remark that the performance of these algorithms, as well as of the structure itself, is strongly related to the error propagation phenomenon. As the name indicates, error propagation takes place when wrong symbol estimates are generated by the decision device, which means that the process of interference intersymbol removal via feedback loop that characterizes the DFE is compromised [121].

7.1.1 Predictive DFE Approach

As discussed in Chapter 3, the prediction-error filter (PEF) is able to provide equalization if the channel is minimum-phase. For general phase responses, the PEF only equalize the magnitude response of the channel, and an additional processing is required to compensate the phase distortions.

In order to extend the use of the predictive approach to nonminimum-phase channels, Rocha et al. proposed in the structure presented in Figure 7.2.

In this figure, a forward PEF, which is always minimum-phase as already discussed in Section 3.7, is used as a magnitude equalizer. The proposal

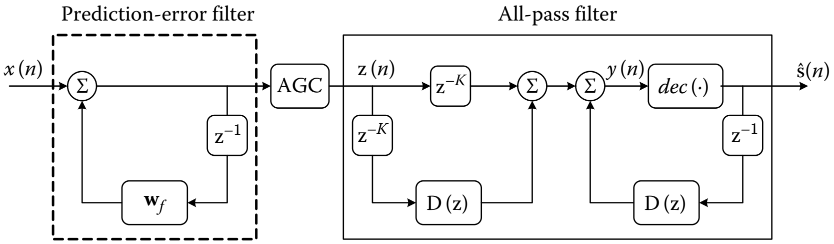


FIGURE 7.2
A predictive equalization approach employing a decision-feedback loop.

includes a recursive (IIR) predictor in order to reduce the number of updated parameters and to attempt for perfect magnitude inversion, when dealing with typical FIR channels. The output of the PEF must be normalized by means of an automatic gain control (AGC) in order to restore the correct signal power.

After this first stage, the signal at the output of the AGC behaves as a prediction-error signal, the power of which is equal to that of the transmitted signal. Such signal presents two characteristics: it tends to be uncorrelated, i.e., its spectrum tends to be flat, due to the magnitude equalization; and it remains subject to phase distortion, which is normally compensated by all-pass filters. Then the proposal includes an all-pass configuration with a decision device in the feedback loop, following the same spirit of the DFE structure [195]. The inclusion of the decision device in the feedback loop provides the nonlinear character of the configuration, which is crucial to the efficiency of the approach.

If we employ an optimization criterion based on the Shalvi–Wenstein criterion, the parameters of both sections of the equalizer are updated by [195]

$$\mathbf{w}_f(n+1) = \mathbf{w}_f(n) + \mu e(n)x(n) \tag{7.6}$$

$$\mathbf{d}(n+1) = \mathbf{d}(n) - \gamma \left\{ \left[y(n)^* |y(n)|^2 - v(n)^* y(n) \right] \mathbf{z}(n) \right. \tag{7.7}$$

$$\left. - \left[y(n)^* |y(n)|^2 - v(n) y(n)^* \right] \hat{\mathbf{s}}(n)^* \right\} \tag{7.8}$$

$$v(n) = v(n-1) + \mu_2 \left(y^2(n) - v(n-1) \right) \tag{7.9}$$

where

$$\mathbf{z}(n) = [z(n-K+1), z(n-K+2), \ldots, z(n)]^T \tag{7.10}$$

and

$$\hat{\mathbf{s}}(n) = \left[\hat{s}(n-1), \hat{s}(n-2), \ldots, \hat{s}(n-K)\right]^T \tag{7.11}$$

It is worth emphasizing that by compensating the phase distortion, the nonlinear all-pass maps an uncorrelated signal into an i.i.d sequence of symbols. In fact, the final goal of equalization is the retrieval of the i.i.d. transmitted symbols and, in this sense, the potential of nonlinear approaches can be exploited in different structures, as shown in the sequel.

7.2 Volterra Filters

The general problem of nonlinear filtering can be conceived as one of finding suitable representations for certain mappings, responses, or behaviors that are particularly useful in information extraction tasks. Representing general functions in terms of other functions belonging to a pre-defined class is a well-known procedure in mathematics and engineering, as the concept of Taylor series [47] attests.

The Volterra series, named after the Italian mathematician Vito Volterra [201], can be thought of, in simple terms, as an extension of the idea of Taylor series to allow the modeling of responses endowed with memory. In other words, the Volterra series is a mathematical tool that can be used to model a broad class of nonlinear systems, in analogy with the Taylor series, which can be used to describe a broad class of nonlinear functions. Having this in mind, we discuss in this section the employ of this general model to design filters with effective processing capability.

In order to model a nonlinear discrete-time SISO system, the general form of the Volterra series is expressed by [201]

$$\begin{aligned} y(n) = h_0 &+ \sum_{k_1} h_1(k_1)x(n-k_1) + \sum_{k_1}\sum_{k_2} h_2(k_1,k_2)x(n-k_1)x(n-k_2) \\ &+ \sum_{k_1}\sum_{k_2}\sum_{k_3} h_3(k_1,k_2,k_3)x(n-k_1)x(n-k_2)x(n-k_3) + \cdots \end{aligned} \tag{7.12}$$

where

$y(n)$ denotes the output of the system
$h_i(k_1, \ldots, k_j)$ are the so-called Volterra kernels
$x(n)$ is the input signal

An example of a second-order truncated version of this series is shown in Figure 7.3.

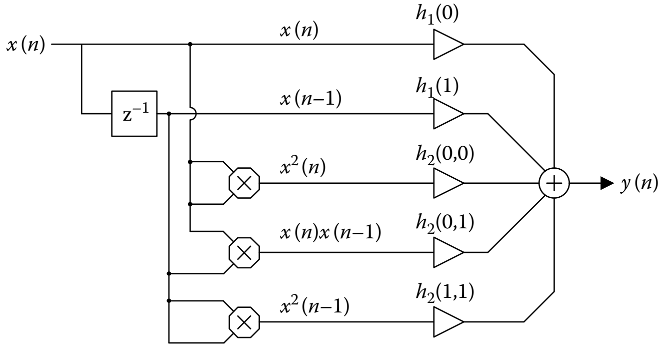


FIGURE 7.3
Structure of a second-order Volterra model with two inputs.

Expression (7.12) reveals that the series is based on the idea of representing the system response in terms of polynomial components, which contain products of different delayed versions of the input signal. At this point, it is important to highlight two aspects:

1. The structure of the Volterra series is responsible for generating a menace of "explosion of dimensionality" associated with the gradual increase of the approximation order and the number of inputs.
2. The nonlinear input–output relationship established by a truncated version of the Volterra series is linear with respect to the free parameters $h_i(\cdot)$.

The second item is a key factor when it comes to the problem of choosing the values $h_i(\cdot)$ that are optimal for a given application. Indeed, since the Volterra filter is linear with respect to its free parameters, we can employ the Wiener framework as well as the LMS and RLS algorithms, as discussed in Chapter 3. From the standpoint of parameter adaptation, the Volterra approach is not so significantly different from that based on FIR filters, which is certainly a positive feature. It is not necessary, for instance, to be concerned with the possibility of local minima when the MSE cost function is employed.

As an illustration, if the Volterra kernels of the generic structure presented in (7.12) are adapted by an LMS procedure, the update expression of the algorithm will be, for a generic parameter [201],

$$h_i(k_1, k_2, \ldots, k_j, n+1) = h_i(k_1, \ldots, k_j, n) + \mu e(n) x(n-k_1) \cdots x(n-k_j) \quad (7.13)$$

with $e(n) = d(n) - y(n)$ being the conventional error signal. In simple terms, this LMS corresponds to a version capable of dealing with linear combiners characterized by an input vector containing polynomial combinations.

Another fact that contributed for the adoption of Volterra filtering solutions is related to the convergence properties of the Volterra series. In [201], we find a clarifying discussion on the subject, a discussion that brings important conclusions about the discrete-time case, which is the focus of our interest here. Using a result by Stone and Weierstrass [279], Mathews and Sicuranza [201] remark that a discrete-time, time-invariant, causal and finite-memory system endowed with the property that small changes in the input produce small changes in the output can be uniformly approximated within a uniformly bounded set of input signals using a truncated version of the Volterra series. This means that a Volterra filter is capable of establishing an input–output relationship that will be arbitrarily close to that established by the referred generic system. From a filtering standpoint, this means that the filters have the necessary flexibility to build the mappings that will be necessary to solve many practical problems, in a spirit similar to that of the universal approximation property that will be discussed in the context of neural networks, presented in Section 7.4.

Before introducing the neural network configurations, it is suitable to state the notion of classification and relate it with the problem of equalization.

7.3 Equalization as a Classification Task

As discussed in Chapter 4, the optimal equalization solution in terms of minimization of the bit-error probability is provided by the MAP criterion.

It is interesting to point out that since the parameters being estimated by the MAP criterion, i.e., the transmitted symbols, may only assume a finite number of possibilities, the Bayesian equalizer essentially performs a *classification task*, associating each received vector to a possible transmitted symbol.

In order to clarify this statement, let us assume that the signal is of a discrete nature and the channel is modeled as an FIR filter. In the absence of noise, such a scenario generates a finite number of possible values for the received samples. For instance, let us suppose that the transmitted signal $s(n) \in \{+1/-1\}$, the channel has L coefficients, and the equalizer has N inputs. In such a case, the signal $x(n)$ will depend on $\{s(n), s(n-1), \ldots, s(n-L+1)\}$. Since the equalizer will employ $\{x(n), \ldots, x(n-K+1)\}$, it means that the equalizer output depends on $\{s(n), \ldots, s(n-K-L+2)\}$. Therefore, in the noiseless case, the equalizer has 2^{K+L-1} possible input vectors.

For example, if we consider a channel with $h(z) = 0.5 + 1.0z^{-1}$, so that the input vector is composed by $K=2$ samples, it is possible to observe eight different equalizer input vectors, which are summarized in Table 7.1. Each equalizer input vector, which we call channel state vector and denote as $\mathbf{c}_i$, is generated from a triple $s(n)$, $s(n-1)$, and $s(n-2)$ of the transmitted signal.

TABLE 7.1
Table of States and Labels

				$\mathbf{x}(n)$		
$\mathbf{c}_j$	$s(n)$	$s(n-1)$	$s(n-2)$	$x(n)$	$x(n-1)$	$f_1(\mathbf{x}(n))$
$\mathbf{c}_1$	+1	+1	+1	1.5	1.5	+1
$\mathbf{c}_2$	+1	+1	−1	1.5	−0.5	+1
$\mathbf{c}_3$	+1	−1	+1	−0.5	0.5	+1
$\mathbf{c}_4$	+1	−1	−1	−0.5	−1.5	+1
$\mathbf{c}_5$	−1	+1	+1	0.5	1.5	−1
$\mathbf{c}_6$	−1	+1	−1	0.5	−0.5	−1
$\mathbf{c}_7$	−1	−1	+1	−1.5	0.5	−1
$\mathbf{c}_8$	−1	−1	−1	−1.5	−1.5	−1

Thus, from a classification standpoint, each of the 2^{K+L-1} vectors must be properly associated with one of the possibilities of the alphabet of the transmitted signal. Therefore, we may think of each $\mathbf{c}_i$ as a point in the input space, and the problem of building an equalizer becomes a matter of defining an adequate mapping $f_1(\mathbf{x}(n))$ that provides a SER as small as possible. It is important to remark that the labeling of these points is not necessarily unique, since there are many different delayed versions of $s(n)$ that we may be interested in recovering: each labeling poses a distinct classification problem and some problems may be easier than others. This fact can be understood as another instance to assess the relevance of a good choice of the equalization delay, as discussed in Section 4.6.

However, the received signal generally contains additive noise, usually modeled as a zero-mean white stochastic process. In this case, the input space will contain a number of data clouds around the centers $\mathbf{c}_i$. Figure 7.4a

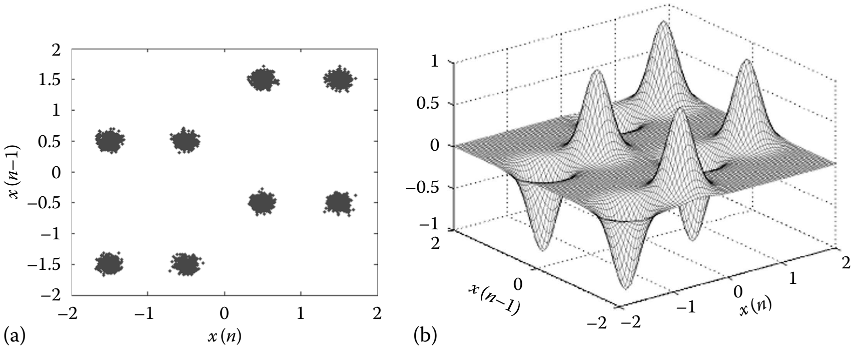


FIGURE 7.4
Example illustrating the received vectors and the corresponding nonlinear mapping provided by an equalizer: (a) possible input vectors and (b) centers and mapping.

depicts the distribution of the input vectors in the noisy case. The circular symmetry exhibited by the clouds is a consequence of the white and Gaussian nature of the noise, while the degree of dispersion is related to the adopted SNR.

Figure 7.4b depicts a possible nonlinear mapping $f_1(\cdot)$, which, in this case, labels each channel state vector with the corresponding value of $s(n)$, as shown in Table 7.1.

7.3.1 Derivation of the Bayesian Equalizer

In order to determine the Bayesian equalizer, let us consider a scenario in which the transmitted signal is composed of i.i.d. samples from a 2-PAM constellation. The equalizer input vector is real, i.e., $\mathbf{x}(n) \in \mathbb{R}^K$. The conditional pdf of $\mathbf{x}(n)$ given a certain channel state $\mathbf{c}_j$ is given by

$$p\left(\mathbf{x}\left(n\right)|\mathbf{c}_j\right) = \left(2\pi\sigma_\eta^2\right)^{-K/2} \exp\left(\frac{-\left\|\mathbf{x}\left(n\right) - \mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right) \tag{7.14}$$

The minimum-error probability is obtained if the estimate of the transmitted symbol is given by the MAP criterion, i.e., maximizing $P\left(s(n-d) = s|\mathbf{x}(n)\right)$, $s(n) \in \{+1, -1\}$ [233]. Thus, the decision is made comparing the a posteriori pdf

$$y\left(n\right) = f_{Bayes}\left(\mathbf{x}\left(n\right)\right) = P\left(s\left(n-d\right) = +1|\mathbf{x}\left(n\right)\right) - P\left(s\left(n-d\right) = -1|\mathbf{x}\left(n\right)\right). \tag{7.15}$$

It can be noticed that (7.15) is positive for

$$P\left(s(n-d) = +1|\mathbf{x}(n)\right) \geq P\left(s(n-d) = -1|\mathbf{x}(n)\right) \tag{7.16}$$

and negative for

$$P\left(s(n-d) = +1|\mathbf{x}(n)\right) < P\left(s(n-d) = -1|\mathbf{x}(n)\right) \tag{7.17}$$

which suggests a decision device of the type

$$\hat{s}\left(n-d\right) = \operatorname{sgn}\left(y\left(n\right)\right) = \begin{cases} +1 & se\ y\left(n\right) \geq 0 \\ -1 & se\ y\left(n\right) < 0 \end{cases} \tag{7.18}$$

Using Bayes' rule, the a posteriori probability of the transmitted signal is given by

$$P\left(s\left(n-d\right)=s|\mathbf{x}\left(n\right)\right)=\frac{p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=s\right)P\left(s\left(n-d\right)=s\right)}{p\left(\mathbf{x}\left(n\right)\right)} \tag{7.19}$$

where

$P(s(n-d)=s)$ is the a priori probability of the transmitted signal

$p(\mathbf{x}(n)|s(n-d)=s)$ is the conditional pdf of the channel output vector given a transmitted symbol

The conditional and a priori probability can be calculated in terms of the channel and the noise statistics. Since we assume an i.i.d. transmitted sequence, $P(s(n-d)=+1)=P(s(n-d)=-1)=1/2$. The conditional pdf $p(\mathbf{x}(n)|s(n-d)=+1)$ is the sum of the pdfs for each one of the states $\mathbf{c}_j \in \mathbf{C}_d^+$,

$$\begin{aligned}p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=+1\right)&=\frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+}p\left(\mathbf{x}\left(n\right)|\mathbf{c}_j\right)\\&=\frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\left(2\pi\sigma_\eta^2\right)^{-K/2}\exp\left(\frac{-\left\|\mathbf{x}\left(n\right)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right)\end{aligned} \tag{7.20}$$

where $\mathbf{C}_d^+$ is the set of states for which $s(n-d)=+1$ and $Pr(\mathbf{c}_j)=\frac{1}{N_s}$. Similarly, $p(\mathbf{x}(n)|s(n-d)=-1)$ is expressed as

$$\begin{aligned}p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=-1\right)&=\frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^-}p\left(\mathbf{x}\left(n\right)|\mathbf{c}_j\right)\\&=\frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^-}\left(2\pi\sigma_\eta^2\right)^{-K/2}\exp\left(\frac{-\left\|\mathbf{x}\left(n\right)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right)\end{aligned} \tag{7.21}$$

where $\mathbf{C}_d^-$ is the set of states for which $s(n-d)=-1$.

Substituting (7.19) into (7.15), we get

$$\begin{aligned}f_{Bayes}\left(\mathbf{x}\left(n\right)\right)&=\frac{p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=+1\right)P\left(s\left(n-d\right)=+1\right)}{p\left(\mathbf{x}\left(n\right)\right)}\\&\quad-\frac{p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=-1\right)P\left(s\left(n-d\right)=-1\right)}{p\left(\mathbf{x}\left(n\right)\right)}\\&=\frac{1}{2p\left(\mathbf{x}\left(n\right)\right)}\left[p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=+1\right)-p\left(\mathbf{x}\left(n\right)|s\left(n-d\right)=-1\right)\right]\end{aligned} \tag{7.22}$$

Moreover, since $\frac{1}{2p(\mathbf{x}(n))}$ is a positive scaling factor and does not influence the sign of $f(\mathbf{x}(n))$, (7.22) can be rewritten as

$$f_1\left(\mathbf{x}(n)\right) = p\left(\mathbf{x}(n)\,|s\left(n-d\right) = +1\right) - p\left(\mathbf{x}(n)\,|s\left(n-d\right) = -1\right) \tag{7.23}$$

Using (7.20) and (7.21) in (7.23), we have

$$\begin{aligned}
f\left(\mathbf{x}(n)\right) =& \frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+} p\left(\mathbf{x}(n)\,|\mathbf{c}_j\right) - \frac{1}{N_s}\sum_{\mathbf{c}_i\in\mathbf{C}_d^-} p\left(\mathbf{x}(n)\,|\mathbf{c}_i\right)\\
=& \frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\left(2\pi\sigma_\eta^2\right)^{-K/2}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right)\\
&- \frac{1}{N_s}\sum_{\mathbf{c}_i\in\mathbf{C}_d^-}\left(2\pi\sigma_\eta^2\right)^{-K/2}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_i\right\|^2}{2\sigma_\eta^2}\right)\\
=& \left\{\sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i\in\mathbf{C}_d^-}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_i\right\|^2}{2\sigma_\eta^2}\right)\right\}\\
&\times\frac{1}{N_s}\left(2\pi\sigma_\eta^2\right)^{-K/2}
\end{aligned} \tag{7.24}$$

The term $\frac{1}{N_s}\left(2\pi\sigma_\eta^2\right)^{-K/2}$ in (7.24) is always positive and, hence, can be suppressed without altering the decision function sign, which can be written as [233]

$$\begin{aligned}
f\left(\mathbf{x}(n)\right) =& \sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i\in\mathbf{C}_d^-}\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_i\right\|^2}{2\sigma_\eta^2}\right)\\
=& \sum_{j=1}^{N_s} w_j\exp\left(\frac{-\left\|\mathbf{x}(n)-\mathbf{c}_j\right\|^2}{2\sigma_\eta^2}\right)
\end{aligned} \tag{7.25}$$

where $w_i = +1$ if $\mathbf{c}_j \in \mathbf{C}_d^+$ and $w_i = -1$ if $\mathbf{c}_j \in \mathbf{C}_d^-$. Equation 7.25 is the Bayesian equalizer decision function. As can be noted, this decision function is nonlinear and completely defined in terms of the channel states and noise statistics.

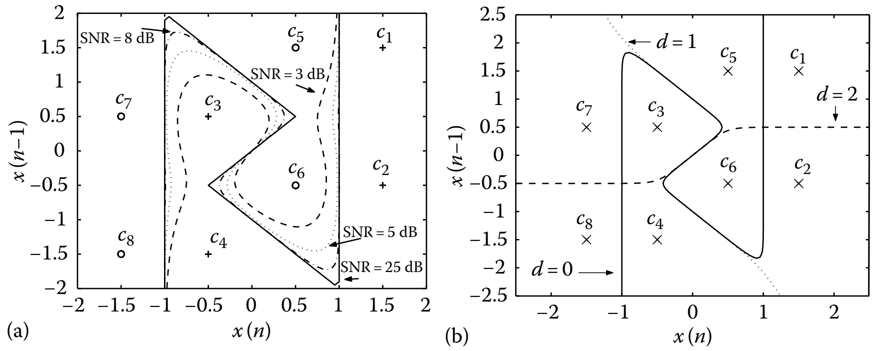


FIGURE 7.5
Decision boundaries for different (a) SNRs and (b) equalization delays. The (+) sign indicates a center associated with the transmission of a −1 symbol.

In a situation of this sort, the problem of building the optimal equalizer can be directly understood in terms of finding the separating surface that leads to the smallest possible SER. In fact, the nonlinear mapping previously provided in Figure 7.4b corresponds to the Bayesian equalizer that recovers the transmitted signal without delay and for SNR = 15 dB.

In Figure 7.5a, we present the centers shown in Table 7.1, numbered and labeled according to the choice $d = 0$, and the decision boundaries between the two classes generated by the Bayesian equalizer. The boundary reveals that for the adopted delay, an efficient equalizer must have a strongly nonlinear character. It is also interesting to notice that under the conditions in question, the boundary, in some regions, lies approximately in the middle of a pair of centers that are the nearest neighbors to each other. Another interesting point is that the decision boundary does not seem to be very sensitive to noise if SNR > 8 dB. There is a slight difference between the boundaries corresponding to SNR = 8 dB and SNR = 25 dB, and they asymptotically tend to a composition of line segments.

The decision boundaries for different equalization delays are shown in Figure 7.5b. In this case, the decision boundaries are very different from each other. For $d = 0$, the channel states can be correctly classified only if the decision boundary is nonlinear, while for $d = 1$ and $d = 2$, it should be possible to classify the channel states by means of a linear boundary. This example illustrates that even for a linear channel, equalization may be a nonlinear classification problem.

The nonlinear nature of the Bayesian equalizer, even for linear channels, highlights the fact that nonlinear filtering structures could be considered to approximate the mapping provided by the optimal equalizer. One possibility of this sort is to employ artificial neural networks, to be discussed in the following.

7.4 Artificial Neural Networks

Artificial neural networks are nonlinear adaptive devices inspired by models of the behavior of the brain. In a nutshell, these networks can be understood as being originated by the interconnection of relatively simple nonlinear processing units, the neurons. In view of the scope of the book, we restrict our attention to a pair of classical neural networks: the MLP and the RBF network.

7.4.1 A Neuron Model

We start by considering a neuron model that bears a significant degree of resemblance to the seminal McCulloch–Pitts proposal [206]. This neuron is the basic unit underlying the important solution known as perceptron [254], which played a most relevant historical role in the context of learning efficient classifiers. The neuron model is simple: a number of input stimuli are linearly combined according to a set of weights and afterward suffer the action of a nonlinear memoryless function. Intuitively, the model evokes the effects of the synapses, stimuli integration, and the existence of a threshold related to activation. Figure 7.6 illustrates this model.

Mathematically, the input–output response of the neuron is given by

$$y(n) = \phi\left(\mathbf{w}^T\mathbf{x}(n)\right) \tag{7.26}$$

where

$\mathbf{w}$ is a synaptic weight vector

$\mathbf{x}(n)$ is the neuron input vector

$\phi(\cdot)$ is the nonlinear function known as activation function

The activation function contains a threshold parameter that can be eliminated by considering the existence of an input always fixed at +1 and an

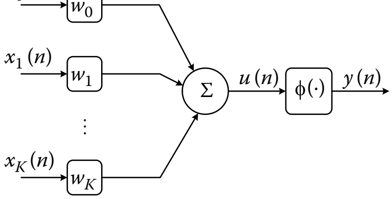


FIGURE 7.6
Neuron model.

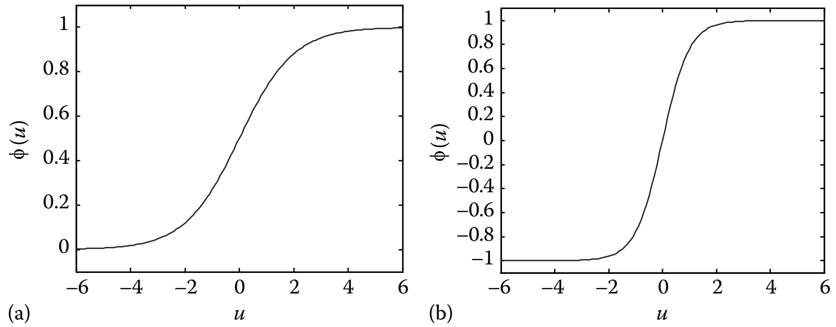


FIGURE 7.7
Typical activation functions: (a) logistic function and (b) hyperbolic tangent function.

additional weight, the bias term. Typical activation functions are the logistic function, given by

$$\phi(u) = \frac{1}{1 + \exp(-\beta u)} \tag{7.27}$$

and the hyperbolic tangent function

$$\phi(u) = \tanh(-\beta u) \tag{7.28}$$

both of which are shown in Figure 7.7. They can be thought of as being smooth versions of the step function that characterizes the classical perceptron model.

Structurally, the presented neuron model may be viewed as a linear combiner followed by a nonlinear device. The mapping performed by the neuron is controlled by the values of the synaptic weights, and we can immediately devise the use of the adaptive filtering framework to adapt it. However, our interest is not to deal with a single neuron, but with an entire network; this must be considered in the algorithm derivation as discussed in the sequel. First, it is relevant to say a few words about the architecture of neural networks.

7.4.2 The Multilayer Perceptron

The network we are attempting to build is based on the idea of organizing the neurons in multiple layers, so that the outputs of a given layer are the inputs of the next, except for the first (input) and the last (output) layers. The input layer contains the sources of all signals to be processed by the network, while the output layer contains the result of the processing carried out by the network. Between them, signals can be carefully nonlinearly transformed into different spaces up to the point in which the output layer is ready to properly

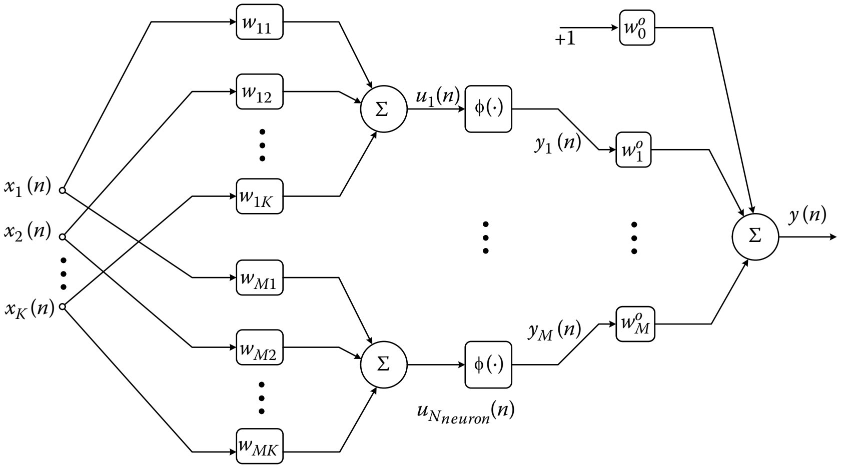


FIGURE 7.8
Example of an MLP.

generate the desired response. Such an organization leads to a most useful neural network architecture, the so-called MLP [136]. Figure 7.8 illustrates an MLP structure with a single hidden (intermediate) layer containing N_{neuron} neurons, K inputs, and a linear output layer with a single output.

We notice a pattern of full connection between neurons of different layers, which leaves to the task of choosing the synaptic weights the possibility of working with all options. As demonstrated by Cybenko [80], a network of this kind is a universal approximator, i.e., is capable of approximating with an arbitrary degree of precision any mapping in a bounded region. This result is presented in more detail in the following theorem.

THEOREM 7.1 (Universal Approximation Theorem for MLPs)

Let $\phi(\cdot)$ be a nonconstant, bounded, and monotome-increasing continuous function. Let I_{m_0} denote the m_0-dimensional unit hypercube $[0,1]^{m_0}$. The space of continuous functions on I_{m_0} is denoted by $C(I_{m_0})$. Then, given any function $f \in C(I_{m_0})$ and $\epsilon > 0$, there exists an integer m_1 and sets of real constants α_i, b_i, and w_{ij}, where $i = 1, \ldots, m_1$ and $j = 1, \ldots, m_0$, such that we may define

$$F(x_1, \ldots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \phi \left(\sum_{j=1}^{m_0} w_{i,j} x_j + b_i \right) \tag{7.29}$$

as an approximate realization of the function f; that is,

$$|F(x_1, \ldots, x_{m_0}) - f(x_1, \ldots, x_{m_0})| < \varepsilon \tag{7.30}$$

for all $x_1, x_2, \ldots, x_{m_0}$ that lie in the input space. ■

Since a given network architecture with universal approximation property is established, a fundamental question concerns the method to obtaining its parameters, i.e., the synaptic weights of the network. The MLP can be thought of as a nonlinear filter and, consequently, we may consider the methodology we have used earlier: to choose a criterion and derive a suitable algorithm. In this sense, our classical procedure consists in obtaining the gradient of the cost function with respect to the synaptic weights, in the spirit of the steepest-descent algorithm, as seen in Section 3.4. This is valid, but a difficulty emerges, as it is not straightforward to verify the influence of the error signal built in the output layer on weights belonging to previous layers. From there arises a specific method for adapting the parameters of an MLP, which is presented next.

7.4.2.1 The Backpropagation Algorithm

The context in which the BPA is defined is that of supervised filtering. In such scenario, we can count on having at hand a set of $N_{samples}$ available samples of input stimuli together with the corresponding desired values. For this data set, it is possible to build a cost function given by

$$J_{BP_{averaged}} = \frac{1}{N_{samples}} \sum_{n=1}^{N_{samples}} \big(d(n) - y(n)\big)^2 \tag{7.31}$$

where $d(n)$ is the desired output. It is worth noting that we are dealing directly with a time-average, similarly to the least-squares procedure described in Section 3.5.1, without resorting to statistical expectations that characterize the Wiener approach. The objective is to minimize the cost function in (7.31) with respect to all weights, which is carried out by cleverly employing the chain rule in the process of differentiation. In order to simplify such process, let us consider for a while the case of $N_{samples} = 1$, so that the cost function becomes the instantaneous quadratic error, expressed by

$$J_{BP}(n) = e^2(n) = \big(d(n) - y(n)\big)^2 \tag{7.32}$$

In fact, Equation 7.32 is a stochastic approximation of the MSE, as that used in the derivation of the LMS algorithm. In order to differentiate $J_{BP}(n)$ with respect to the weights of the hidden layer, we make use of the chain rule. From Figure 7.8, if we mentally build a signal-flow graph from each focused weight up to the output, we can obtain

$$\begin{aligned}\frac{\partial J_{BP}(n)}{\partial w_{ij}} &= \frac{\partial J(n)}{\partial e(n)} \cdot \frac{\partial e(n)}{\partial y(n)} \cdot \frac{\partial y(n)}{\partial u^o(n)} \cdot \frac{\partial u^o(n)}{\partial y_i(n)} \cdot \frac{\partial y_i(n)}{\partial u_i(n)} \cdot \frac{\partial u_i(n)}{\partial w_{ij}} \\ &= 2 \cdot e(n) \cdot (-1) \cdot (1) \cdot w_i^o \cdot \phi' \left[u_i(n)\right] \cdot x_j(n) \\ &= -2e(n) w_i^o \phi' \left[u_i(n)\right] x_j(n) \end{aligned} \tag{7.33}$$

Notice that we may express the output in a compact form as

$$y(n) = \begin{bmatrix} w_0^o & \mathbf{w}^{oT} \end{bmatrix} \begin{bmatrix} 1 & \phi\left(\mathbf{y}^{int}(n)\right) \end{bmatrix}^T \tag{7.34}$$

where $\mathbf{y}^{int}(n) = \left[1, y_1(n), \ldots, y_{N_{neuron}}(n)\right]$ and $\mathbf{w}^{oT} = \left[w_1^o, w_2^o, \ldots, w_{N_{neuron}}^o\right]$. Thus, the gradient of $J_{BP}(n)$ related to the weights in the output layer $\mathbf{w}^o$ and the bias w_0^o are given by

$$\frac{\partial J_{BP}(n)}{\partial \mathbf{w}^o} = -2e(n)\mathbf{y}^{int}(n) \tag{7.35}$$

$$\frac{\partial J_{BP}(n)}{\partial w_0^o} = -2e(n) \tag{7.36}$$

Now, if we define

$$\mathbf{x}(n) = [x_1(n) \quad x_2(n) \quad \cdots \quad x_K(n)]^T \tag{7.37}$$

we can express the gradient of $J_{BP}(n)$ with respect to the ith set of weights present in the hidden layer by

$$\frac{\partial J_{BP}(n)}{\partial \mathbf{w}_i} = -2e(n) w_i^o \phi' \left[u_i(n)\right] \mathbf{x}(n), \quad i = 1, \ldots, N_{neuron} \tag{7.38}$$

being $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{iK}]^T$.

Having found the gradient vector with respect to the weights of the hidden layer and of the output layer, we are in a position to update all weights of our network in the spirit of the steepest-descent method. However, we must not forget that we purposely calculated the derivatives with respect to an instantaneous squared error, which means that the resulting BPA will be conceptually similar to the LMS procedure. This approach constitutes the online BPA, which is particularly suitable to real-time applications. It is expressed by Algorithm 7.1.

Algorithm 7.1: Online Backpropagation Algorithm

```
1. Initialize the synaptic weights
2. While the stopping criterion is not met, do
   (a) Given the input vector $\mathbf{x}(n)$ and the present weights, obtain the
       network output $y(n)$. Calculate the error $e(n) = d(n) - y(n)$.
   (b) Adapt the weights of the output layer using (7.35) and (7.36):
       $\mathbf{w}^o(n+1) = \mathbf{w}^o(n) + 2\mu e(n)\mathbf{y}^i(n)$    (7.39)
       $w_0^o(n+1) = w_0^o(n) + 2\mu e(n)$    (7.40)
   (c) Adapt the weights of the hidden layer using (7.38):
       $\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + 2\mu e(n) w_i^o \phi'[u_i(n)]\,\mathbf{x}(n) \quad i = 1, \ldots, N_{neuron}$    (7.41)
3. End (while)
```

On the other hand, we may also compute an average gradient from the multiple instantaneous gradients originated from a given data set. This procedure engenders the batch BPA, which excels in terms of precision in the determination of the gradient search direction for the entire data set, but is more complex and therefore less suited to online applications.

Both online and batch BPA are in fact steepest-descent-based methods, the derivation of which makes use of the first-order derivatives of the cost function. In addition to them, it is worth mentioning that the use of second-order algorithms as well as of methods for optimizing the step-size are widespread in the field of neural networks. Some approaches to be considered by those interested in using neural networks in signal processing tasks are presented in [44].

In the process of adaptation of the parameters of an MLP, we must keep in mind that the structure is nonlinear with respect to its free parameters, which leads to an MSE surface that can be multimodal. Consequently, it is possible that iterative algorithms based on the derivatives of the cost function, like the BPA and second-order methods, suffer from the drawback of local convergence. The impact of this possibility will depend on the quality of such minima, since a local minimum is not necessarily a bad one. The character of the cost function draws our attention to the importance of the weight initialization process when dealing with the MLP. Local convergence is an issue, but other factors like eventual saturating effects associated with the activation functions can also be relevant in terms of speed of convergence. This explains the use of strategies that advocate an initialization employing small random values, as described in [140].

Algorithm 7.2: Batch Backpropagation Algorithm

```
1. Initialize the synaptic weights
2. While the stopping criterion is not met, do
   (a) For $n = 1$ to $N_{samples}$
       (i) Given the input vector $x(n)$ and the present weights, obtain the network output $y(n)$. Calculate the error $e(n) = d(n) - y(n)$.
       (ii) Calculate the derivatives presented in (7.35). Store the results.
       (iii) Calculate the derivatives presented in (7.38). Store the results.
   (b) End (for)
   (c) Obtain the average, with respect to all $N_{samples}$ patterns, of the gradient terms calculated according to (7.35) and stored in step 2.1.2. This average will be referred to as $\nabla_o[J_{BP}]$.
   (d) Obtain the average, with respect to all $N_{samples}$ patterns, of the gradient terms calculated according to (7.38) and stored in step 2.1.3. This average will be referred to as $\nabla_i[J_{BP}]$.
   (e) Adapt the weights of the output layer:
           $\mathbf{w}^o(n+1) = \mathbf{w}^o(n) - \mu\nabla_o[J_{BP}]$    (7.42)
           $w_0^o(n+1) = w_0^o(n) - \mu\frac{\partial J_{BP}}{\partial w_0^o}$    (7.43)
   (f) Adapt the weights of the hidden layer:
           $\mathbf{w}_i(n+1) = \mathbf{w}_i(n) - \mu\nabla_i[J_{BP}] \quad i = 1, \ldots, N_{neuron}$    (7.44)
   (g) Randomly rearrange the elements of the training set of $N_{samples}$ input–output patterns.
   (h) End (while)
```

7.4.3 The Radial-Basis Function Network

The RBF neural network is the other classical structure to be studied in this chapter. In order to well understand its behavior, it is interesting to establish contrasts between the types of approximation built by the RBF and MLP networks.

The RBF network possesses a structure with three layers: the standard input layer, a hidden layer in which the nonlinear processing is carried out,

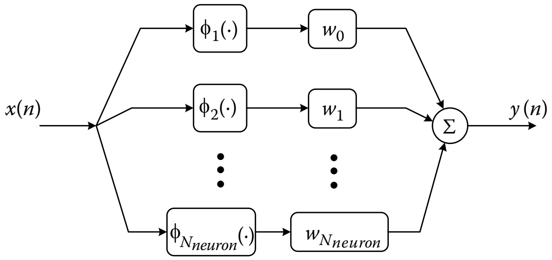


FIGURE 7.9
Structure of an RBF neural network.

and a linear output layer. Roughly, the main difference between the MLP and the RBF lies in the hidden layer, which is formed by neurons with a model distinct from that shown in (7.26). Figure 7.9 depicts the RBF neural network.

In contrast with the MLP, the RBF builds a generic nonlinear mapping by placing nonlinear functions with a characteristic radial decreasing (or increasing) pattern around certain positions (typically referred to as centers) and linearly combining them. Mathematically, this is expressed by

$$y(n) = \mathbf{w}^T \Phi \left[\mathbf{x}(n)\right] \tag{7.45}$$

where $\mathbf{x}(n)$ is the input vector, $\mathbf{w}$ is the weight vector of the linear output layer, and

$$\Phi\left[\cdot\right] = \left[\Phi_1(\cdot) \quad \Phi_2(\cdot) \quad \ldots\right]^T \tag{7.46}$$

is the vector containing the so-called RBFs. Two examples of RBFs are given in Figure 7.10: the Gaussian function, given by

$$\phi(u) = \exp\left(-\frac{(u-\mu)^2}{\sigma^2}\right) \tag{7.47}$$

and the multiquadratic function, given by

$$\phi(u) = \sqrt{\frac{(u-\mu)^2 + \sigma^2}{\sigma}} \tag{7.48}$$

where μ and σ the center and dispersion.

In Figure 7.11, examples of the nonlinear mapping provided by an RBF network are presented. It is important to remark that the approximation scheme that characterizes RBF networks also leads to a universal

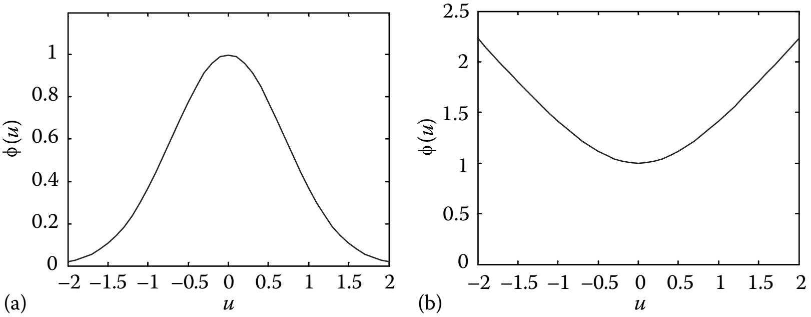


FIGURE 7.10
Examples of one-dimensional RBFs with $\sigma = 1$ and $\mu = 0$: (a) Gaussian function and (b) multiquadratic function.

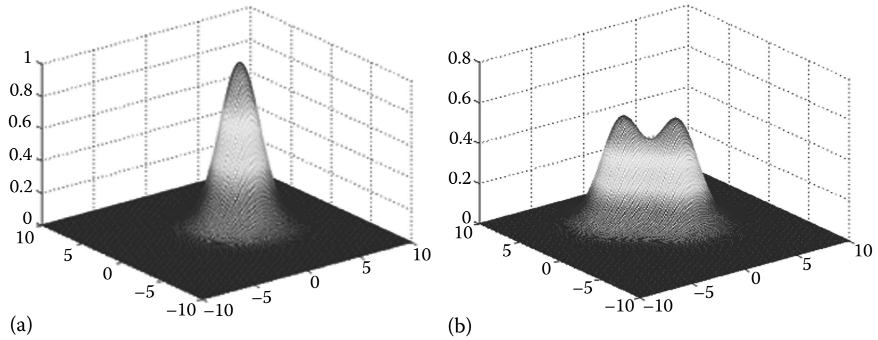


FIGURE 7.11
Examples of a mapping built using an RBF network: (a) output of a single RBF neuron and (b) output of an RBF network with 2 neurons.

approximation theorem similar to that discussed in the context of MLPs [231].

Among the several possible choices, the classical option is for Gaussian functions, so that, in the multidimensional case, we may have, for instance:

$$\Phi\left(\mathbf{x}(n)\right) = \exp\left(-\frac{\|\mathbf{x}(n) - \boldsymbol{\mu}\|^2}{\sigma^2}\right) \tag{7.49}$$

where

μ denotes a mean vector that corresponds to a center
the variance σ^2 expresses a degree of dispersion around the center

The problem of finding the optimal parameters of RBF networks is somewhat different from that associated with the MLP network. In the case of the MLP, all weights are "conceptually equivalent" in that they all belong to structurally similar neurons. This is not the case with RBF networks, for which the nature of the centers and variances are distinct from that of the output weights. This certainly does not mean that the entire set of parameters cannot be jointly adapted [140]. However, this procedure will be related to a complex MSE surface with a removable potential for multimodality. So, it is interesting to consider the option of separating both adaptation processes.

By carrying out a separated optimization, if we suppose the hidden layer to have been properly designed, the remaining problem of finding the output weights will be linear with respect to the free parameters. Thus, we will be back within the framework of linear supervised filtering, and it will be possible to obtain a closed-form solution or to make use of algorithms like LMS and RLS (vide Chapter 3). Since a given choice of the centers and variances potentially establishes the optimal solution of the output weights, we are led to the conclusion that the crux of this approach is the project of the hidden layer.

There are many possibilities for carrying out center placement, for instance, possibilities that range from a uniform distribution to the use of some clustering process [44]. If the idea is to use a clustering method, a classical option is to employ the *k*-means algorithm [105], the summary of which is given in Algorithm 7.3.

Algorithm 7.3: *k*-Means Algorithm

```
1. Initialize the centers
2. While the stopping criterion is not met
   (a) For $n = 1$ to $N_{samples}$
         (i) Determine the distance between each center and the input
             vector $\mathbf{x}(n)$.
        (ii) Assume that the center closest to the input pattern is labeled
             as the $k$th center. Update it according to the following rule:

                 $\mathbf{c}_k \leftarrow \mathbf{c}_k + \mu[\mathbf{x}(n) - \mathbf{c}_k]$        (7.50)

       (iii) End for
3. Randomly rearrange the data set.
4. End while
```

The *k*-means algorithm [105] is founded on a process that can be described as competitive learning [140], in which the centers compete for the right of representing the data. The winner is updated in a manner that tends to reinforce the suitableness of its representation, the outcome of the method being, ideally, a center distribution that expresses key features of the data distribution. In particular, if the data are structured in clusters, a satisfactory center placement should define them properly.

The variance of the RBFs can be determined from the cluster variances or be chosen in accordance with heuristic rules like, for instance [136]

$$\sigma = \frac{d_{max}}{\sqrt{2N_{neuron}}} \tag{7.51}$$

where

d_{max} is the maximum distance between two centers
N_{neuron} is the number of centers, and, consequently, of neurons

The adaptation approach based on this separation is more tractable in many concrete cases, and this is one reason behind its popularity in the literature.

In the signal processing context of equalization, there is another important feature in using RBF networks: they are structurally analogous to the Bayesian equalizer, which, as discussed in Section 7.3, is optimal in terms of minimum SER. An interesting point is that the separate adaptation process discussed above originates a practical recipe for designing a Bayesian filter [68]:

- Use a clustering algorithm to determine the center of the data clouds, which, ideally, correspond to the combinations generated by the channel, i.e., the channel states.
- Use a priori knowledge about the system model, or a heuristic rule, or information brought by the clustering process to determine the noise variance.
- Employ a supervised process to associate each cluster center with the appropriate label ($+1/-1$ in our standard binary case).

It is also important to remark that an RBF network is not the only nonlinear structure structurally similar to the Bayesian filter: fuzzy filters, for instance, are also a viable choice [108,232,233].

7.5 Concluding Remarks

Throughout this chapter we carried out a study of representative approaches in the field of nonlinear signal processing. The aim was to provide a general view of the features of these techniques, with emphasis in their applicability in equalization. So, we first considered the DFE, which is extremely related to this application, and then we widened the scope to the analysis of more general nonlinear structures, as Volterra filter and artificial neural networks.

As far as artificial neural networks are concerned, we focused on two main structures, MLP and RBF, and their corresponding algorithms: backpropagation and *k*-means.

An important issue of this chapter was the analysis of the equalization problem as a classification task. This allowed us to employ neural networks as a method of solution. And, it was also possible to accomplish the derivation of the Bayesian equalizer. As discussed in Chapter 1, this equalizer is the fundamental reference of optimal performance, but it cannot easily be derived in the framework of linear filtering theory, as the suboptimal solutions based on MSE and similar criteria. On the other hand, nonlinear approaches as, for instance, the RBF can provide practical methods for implementing the Bayesian filter.

8

Bio-Inspired Optimization Methods

The ensemble of methods and techniques discussed all through this book are, after all, closely related to solving an optimization problem. The proposed solutions have been currently obtained by the minimization or maximization of a given criterion or cost function. In this chapter, we discuss a different paradigm, the foundations of which come from the study of auto-regulation processes observed in nature. The so-called bio-inspired optimization methods are generally characterized by their global search potential, and do not require significant a priori information about the problem to be solved. Such characteristics encourage their application in the nonlinear and/or unsupervised problems we are dealing with.

The field of bio-inspired techniques for optimization is certainly a broad one. In this chapter, and taking into account the applications in mind, we have decided to concentrate our attention on three classes of tools: *genetic algorithms* (GAs), *artificial immune systems* (AISs), and *particle swarm* methods. The first class is certainly a most emblematic bio-inspired optimization approach. On the other hand, the two are currently the object of attention of many researchers due to some desirable features that, in our opinion, are valid to expose.

In order to better expose our purposes, Section 8.1 provides a general discussion about some points that could motivate the use of bio-inspired tools in some signal processing problems. Then the rest of the chapter is organized as follows:

- In Section 8.2, we discuss the class of the *GAs*. These algorithms are based on elements of the modern synthesis of evolution theory. They are also an interesting starting point for our discussion, given their historical importance and their widespread use in many practical domains.
- In Section 8.3, we analyze another class of bio-inspired methods that can also be considered as an *evolutionary technique*: that of AISs. The analysis is focused in the so-called opt-aiNet, an interesting optimization tool that presents some points of contact with the GAs [86].
- In Section 8.4, we present another branch of techniques, the inspiration of which comes from the collective behavior observed in the nature. To illustrate this branch, we choose the approach known as *particle swarm optimization* (PSO), which is particularly suited to the continuous-valued problems that characterize most filtering applications.

Historical Notes

The history of evolutionary computation began with a number of different, albeit conceptually interrelated, attempts to incorporate the idea of evolution into a number of problem-solving techniques. The origins of the field can be considered to be in the 1950s and 1960s [22]. Three branches that gradually affirmed themselves can be highlighted: evolutionary programming (EP) [109], evolution strategies (ESs) [247,266,267], and GAs [145].

The proposal of GAs is indelibly associated with the name of John Holland, whose 1975 book, *Adaptation in Natural and Artificial Systems* [145], is a classic of the field. In the next decades, the study of GAs attracted exponentially growing attention, which led to the establishment of a vast community of researchers in the field.

AISs [83,87] are a more recent field of research, although it can be safely considered to be a well-established one (an example of this assertion is that the important *International Conference on Artificial Immune Systems* has reached, in 2009, its eighth edition). AISs compose a vast repertoire of computational tools that are being broadly applied in many domains, such as pattern recognition, autonomous navigation and control, data analysis (clustering and data mining), and optimization [87]. Since our interest lies in immune-inspired optimization tools, we would like to highlight the work of de Castro and von Zuben [88], and also the work of de Castro and Timmis [86], who developed an artificial immune network for optimization, known as opt-aiNet.

Kennedy and Eberhart's work can be considered to be the origin of the approach known as PSO [167], although not of the field of swarm intelligence, which other branches like ant-colony approach [102, 103]. Swarm intelligence can be related to important notions like the emergence of collective behavior, which accounts for what can be considered a new computation paradigm. PSO is, nowadays, a well-established optimization tool, with a number of variants and versions [73,273].

8.1 Why Bio-Inspired Computing?

The origins of adaptive filter theory are intimately related to certain hypotheses such as the use of linear feedforward structures and of a mean-squared error criterion, which give shape to the Wiener theory. In the context of the Wiener framework, we are faced with an optimization scenario that suits very well the classical tools based on the derivatives of the MSE cost function, which form the basis of the LMS and RLS algorithms, as well of many other techniques that were not discussed in this work.

In the course of our previous exposition, we had contact with a number of very relevant instances in which the standard form of the Wiener paradigm no longer holds (vide, for example, Chapter 7). In such cases, a relevant question emerges: how well-suited will standard optimization techniques be?

We are not in a position to provide a definitive answer to this question. However, it is important to reflect on a pair of situations in which the limitations of classical nonlinear optimization methods become more evident:

- As outlined in several points of the book, there are at least three potential sources of emergence of multimodal cost functions in adaptive filtering: the use of nonlinear structures, the use of recurrent structures, and the use of criteria based on higher-order statistics. In multimodal contexts, it is possible that a classical optimization method based on the derivatives of the cost function lead to suboptimal solutions, which may be undesirable.
- There are some cases in which it is difficult to obtain the derivatives of the cost function. In such cases, in which it is possible to have access only to values of the cost function at certain points, it is not feasible, in general, to resort to standard adaptive solutions. This may occur, for instance, when one employs complex recurrent structures, piecewise linear filters, and filtering criteria based on certain information-theoretical entities.

The existence of limitations of this sort has opened a very interesting perspective: to use bio-inspired optimization methods in signal processing.* These methods possess a significant global search potential and, in general, require no a priori information about the problem to be solved, except for a comparative measure of quality of the solutions [85]. Such features certainly open many perspectives of building innovative proposals in the field, and we believe that researchers working with subjects related to adaptive filtering must be aware of the potential (and, naturally, of the drawbacks) of these methods.

The nature of these initial remarks should not give the reader the impression that we consider the methods that will be described in the sequel better than the classical optimization approaches that form the basis of the classical adaptive filtering theory. As a matter of fact, the existence of no free lunch theorems [308] indicates that there is no optimization technique that is the most adequate to solve all problems. The view we consider the most useful for any researcher in the field of signal processing is this: know as many methods as you can and try to choose the most useful in the context of the problem you wish to solve.

* Some efforts in this direction can be found, for instance, in [1, 52].

8.2 Genetic Algorithms

GAs are optimization tools based on the synergy between the idea of natural selection and the framework established by modern genetics [22]. In very simple terms, GAs can be considered as a populational optimization tool founded on three conceptual pillars: existence of a selection mechanism, possibility of information interchange between solutions, and existence of spurious changes in their parameters.* The first pillar is the main point of contact between the GA and the problem to be solved, since it is the cost function to be optimized that determines the *fitness* of each solution. The second pillar is related to the idea of *crossover*. Finally, the third pillar is the essence of the idea of *mutation*. These concepts are discussed in more detail in the sequel.

8.2.1 Fundamental Concepts and Terminology

An optimization problem is characterized by the existence of a cost function and certain number of free parameters with respect to which this function is to be optimized, in order to provide the best parameter configuration or, at least, a satisfactory one. The operation of an optimization algorithm can be understood, in this context, as a continuous process of sampling the cost function, i.e., generating/evaluating solutions in a search space, which can be defined as the space generated by all possible instances of the free parameters. The modus operandi of an algorithm is thus related to the manner whereby it performs this sampling [308].

In the case of GAs, the tonic is to conceive the optimization task as being the results of the operation of evolutionary mechanisms. To allow this, a first step is to relate the cost function to a fitness function, the role of which is to create a numerical index to quantify the adequacy of a certain individual, within the simulated evolution framework, to the environment. As, from the standpoint of the optimization task, the fittest means simply the best possible solution, the fitness function can be simply the cost function itself. However, it can also be, if convenient, a mapping of the cost function. This can be done, for instance, if we deal with a minimization problem and need to translate this problem into the evolutionary framework, which typically assumes higher fitness values to be associated with better solutions.

Each solution to the given problem is assumed to correspond to an individual, and the parameters of a given solution must be somehow coded to correspond to its genotype. Having this in view, we freely interchange the

* This description encompasses, in a certain sense, a classical GA, for, modernly, it is acceptable to describe, in some cases, an evolutionary algorithm without a recombination operator as a GA. Notice also that it would have been possible to define the threefold foundation of a GA in terms of selection, reproduction, and variability.

words *solution* and individual. Just like in natural evolution, we have not a single individual, but a population. This means that GAs are populational, that is, perform optimization by manipulating a set of solutions, not a single one, as is the case with many classical nonlinear optimization methods [192].

8.2.2 A Basic Genetic Algorithm

The basic terminology introduced above will also be useful later in the discussion of AISs and particle swarm. At present, let us describe the main steps of a simple implementation of a GA, the pseudo-code of which is shown in Algorithm 8.1. In the subsequent sections, the fundamental operation will be analyzed in more detail.

Algorithm 8.1: Basic GA

```
1. Initialize the population;
2. Evaluate the initial population;
3. While the stopping criterion is not met, do:
   (a) Evaluate the fitness of all individuals;
   (b) Select individuals for crossover;
   (c) Apply the crossover operator;
   (d) Apply the mutation operator;
   (e) Select individuals that will form the new population.
```

After the fitness function has been defined in accordance with the peculiarities of the optimization problem, the next step is to build an initial population. This is usually done in a random fashion, although any sort of a priori information about the problem may be incorporated if this is relevant. Afterward, the main loop of the algorithm starts.

The first step of this loop, described in item (a), is the calculation of the fitness of each individual. This step is crucial, as it is the fitness measure that defines how adequate a given solution (individual) is to the problem to be solved. The role of the fitness measure is particularly pronounced in all selection steps, as it is exactly the quality of a given solution that must guide the optimization process.

The step presented in item (b) is related to the nature of the crossover operator. As indicated earlier, crossover means that more than one individual will be combined to generate an offspring of new solutions that will bring information about new points of the search space. Typically, it is necessary to select pairs of solutions that are subject to this combination, and this is the role of the item in question. A usual way to perform the selection in GAs is by privileging the fittest individuals. By doing so, we assume that a combination of better solutions is an interesting way to generate new solutions and carry on the optimization task.

In item (c), we have the crossover step properly. The idea here is to devise and apply a suitable mechanism of combination between solutions. The original mechanism proposed in [145] has significant points of contact with the biological crossover. However, we may choose other ways for building the operator having in view, for example, specific features of the problem at hand.

The mutation operator is applied in item (d). In biological terms, mutation corresponds to a sort of spurious modification of the genetic material, and this operator is imbued with this notion. The implementation of a mutation scheme is generally done in practice with the aid of stochastic factors, which means, from a computational standpoint, that random number generators should be used. The process has a twofold random character: a randomly selected individual is subject to a random modification of its parameters, i.e., of the values attributed to the variables to be optimized. The point is that a spurious or random parameter modification originates new individuals that will bring novel information to the optimization process. It is important to point out that the occurence of both crossover and mutation is, in certain implementations, regulated by a predefined probability.

In item (e), we introduce a final selection step, which is required if the processes of crossover and mutation led to some sort of increase in the number of focused individuals, i.e., to an intermediate population of solutions with a cardinality larger than that of the original one. In this step, it is demanded that a new generation be formed by elements chosen, in accordance with some criterion, from the original individuals and the products of the crossover and mutation steps. The point is to prevent an unbounded population increase and, at the same time, to make use of the additional information about the cost function brought by the new solutions. It is possible to devise deterministic selection mechanisms, e.g., choosing the best individuals; or stochastic selection mechanisms, e.g., choosing the individuals according to a statistic distribution that favors the fittest.

Finally, we can also employ additional steps like the reintroduction of the best solution found so far if it has been lost, which gives rise to the concept of *elitism*, or even the periodic introduction of fresh individuals.

8.2.3 Coding

As already commented, each individual corresponds to a solution of the focused problem, so that evolutionary approaches are, in general, populational. A solution must be coded in a genotypic level, i.e., in terms of a *chromosome* to which certain phenotypic features may be associated. In other words, the phenotype constitutes a higher-level description of a given solution, while the genotype is a possible codification of such a solution. An adequate coding can drastically simplify the optimization problem [209].

There are countless possibilities for coding solutions, and we consider here two well-known possibilities: binary and real coding.

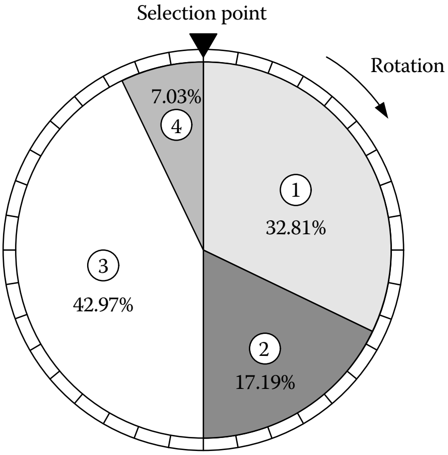
In binary coding, all solutions correspond to a string of bits. This coding is associated with certain pioneering efforts in the field of evolutionary computation [145], and also bears a certain conceptual resemblance with the structure of the biological chromosome. Nevertheless, when a problem is not inherently binary, it is necessary to devise a mechanism to code the solution in terms of bits. In this sense, ideas like Gray codes [22] can be suitable.

It is also possible to work with real coding, as in the so-called evolution strategies, for which each individual is represented by a vector of real numbers [42]. With such a coding, a closer relationship between low-level and high-level representation is achieved, and this may lead to a search modus operandi that can be quite different from that observed in the binary case.

8.2.4 Selection Operators

Selection mechanisms play a key role in the core of the GA. A classical option is to select an individual via the roulette wheel method [26,126], in which the probability of selection of a given member of the population is proportional to its fitness. The name comes from the analogy between this method and a roulette in which each outcome is associated with a different area, as shown in Figure 8.1.

The method implementation is straightforward: using the fitness values of all individuals, it is necessary to create a random number generator that assigns to each individual a probability of selection proportional to its fitness. This can be done, for instance, by using a uniformly distributed random generator together with a properly chosen set of intervals, which correspond



#	Fitness	% of total
1	0.84	32.81
2	0.44	17.19
3	1.1	42.97
4	0.18	7.03
Total	2.56	100

FIGURE 8.1
Illustration of the roulette wheel selection method.

to the areas of the different regions of the wheel in the figure. Algorithm 8.2 describes the procedure.

Algorithm 8.2: Roulette Wheel Selection Method

- Feed the roulette wheel random number generator with the fitness values of the individuals that will take part in the selection process.
- Run the generator and select the individual associated with the obtained index.

An implementation to select multiple individuals may include additional mechanisms, like a lookup table that prevents multiple selections of the same individual.

The roulette wheel method can be considered efficient from the standpoint of speed of convergence, as it has the potential of strongly biasing the selection process toward individuals with fitness values expressively above the average. However, it suffers from the threat of loss of population diversity. Under a selection mechanism that decisively favors a relatively high fitness measure, it is possible that, after a certain while, the entire population be significantly similar to the best individual so far. This can hamper the development of other solutions that could lead to better performance levels. The idea of diversity is essential to allow that multiple optima of a given cost function be properly explored, and this has a decisive impact in the capability of avoiding local minima, which is essential in many signal processing tasks [23].

Other classical selection mechanisms are responsible for mitigating this effect, which is related to the idea known in the literature as selective pressure [22]. Among these methods we may highlight two: the rank and tournament selection methods [127].

The method of selection based on rank generates the probabilities not directly from the fitness measure, but from the position occupied by the individuals in a ranking based on this measure. Therefore, for instance, an enormous difference in the fitness of the best and the second best individuals will not necessarily lead to an enormous difference in the probability of selection of these two solutions. The conclusion is that such a strategy has the potential of reducing selective pressure. In fact, if a high selective pressure is not desirable, the same is valid for a significantly low selective pressure, so that the objective is to attain a solution that will lead to a satisfactory performance. Naturally, this task is extremely complex, since the ideal choice varies from case to case, but the user of a GA must keep in mind the need of being aware of the role of the selection mechanism. In the case of the rank-based mechanism, the choice of different functions to map the position of an individual onto its probability of selection is a tool to control the performance of the operator [127]. The steps described in Algorithm 8.3 give an idea of an implementation of the method to select a single individual.

Algorithm 8.3: Rank Selection Method

- Choose a mapping of the fitness values into the probabilities of selection that is appropriate from the standpoint of some objective (e.g., to maintain population diversity).
- Apply the mapping to the fitness values of the individuals that will take part in the selection process.
- Run a random generator using the mapped probability values and select the individual associated with the obtained index.

The last selection procedure to be considered here is tournament selection [82]. As the name indicates, this method creates a series of competitions, based on fitness, between groups of individuals of the population, and uses the outcome of these tournaments to define those that are selected. A key aspect is that the procedure allows control over the selective pressure. This can be understood if we consider, for instance, a competition with q individuals of a population with, for instance, 10 solutions. If $q = 10$, the selection mechanism will be deterministic, since the individual with the highest fitness value will always be chosen. On the other hand, if $q = 1$, the fitness will have no impact on the choice of an individual, since the choice of the participants is based on a uniform distribution. Finally, $1 < q < 10$, we will reach a controllable intermediate level of selective pressure. This elegant control mechanism is responsible for the wide applicability of this method. The pseudo-code in Algorithm 8.4 gives an idea of the method, when applied to the selection of a single individual.

Algorithm 8.4: Tournament Selection Method

- Define the number of tournaments and the number of individuals taking part in a tournament.
- Start the tournaments. Use a uniform random generator to define which individuals will be chosen to take part and keep a table of the winners, directly defined by the fitness measure.
- Choose the individual with the largest number of wins.

Different criteria can be used to deal with draws and with the selection of multiple individuals, but this is not part of the structure of the tournament itself. It is also important to point out that alternative tournament schemes can be defined [23].

8.2.5 Crossover and Mutation Operators

There are many ways to perform crossover, but their essence is fundamentally the same: a pair of chromosomes is combined in accordance with a given rule and originates descendants. Naturally, an infinity of approaches fit this scheme, and the aim of this section is to describe some classical solutions.

In the case of binary coding, the most traditional crossover methodology is the single-point crossover. Basically, two individuals are selected in accordance with a given mechanism. A position is randomly chosen and the material after this position is swapped between the two parents, which gives rise to a pair of new individuals. Figure 8.2 illustrates the process.

In the case of real coding, it is natural to use arithmetic operations in the recombination process [208, 209], which, from a given a pair of parents $\mathbf{p}_1$ and $\mathbf{p}_2$, generates the following descendants:

$$\begin{aligned} \mathbf{p}_1' &= \alpha\mathbf{p}_1 + (1-\alpha)\mathbf{p}_2 \\ \mathbf{p}_2' &= \alpha\mathbf{p}_2 + (1-\alpha)\mathbf{p}_1 \end{aligned} \tag{8.1}$$

Figure 8.3 illustrates the effect of this kind of crossover operator. The value can be randomly chosen using a uniform distribution in the interval [0, 1]. In some cases, the value may also be allowed to be outside this interval, which may propitiate an exploration of a wider region of the search space.

Mutation operators are characterized by originating spurious modifications, in analogy with the role mutation plays in the biological scenario. In the binary coding case, the classical mutation operator is based on the idea of causing, with a given probability, a bit 0 to become a bit 1 or a bit 1 to become a bit 0. As in nature, an apparently simple modification of this kind may engender significant phenotypic alterations, and this may lead to remarkable

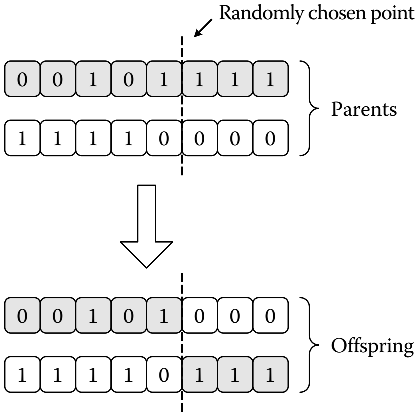


FIGURE 8.2
Example of single-point crossover.

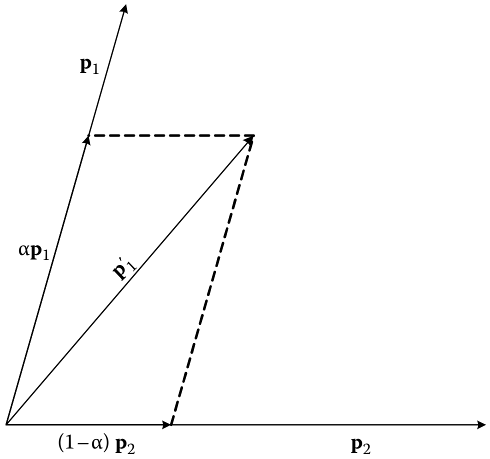


FIGURE 8.3
Example of an arithmetic crossover.

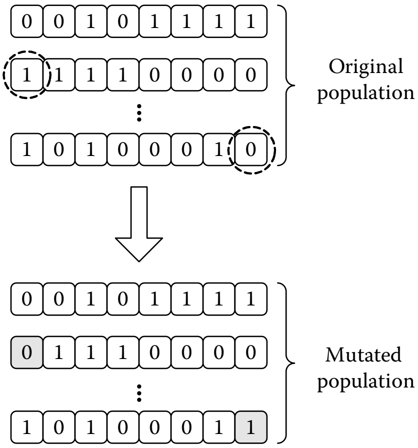


FIGURE 8.4
Binary mutation.

improvements in the quality of the obtained solutions. Figure 8.4 illustrates the process.

In the real coding case, mutation is typically implemented with the aid of continuous random variables. We may consider, for instance, that the relationship between the mutated and the original solutions obeys the following form:

$$\mathbf{x}' = \mathbf{x} + \mathbf{m} \tag{8.2}$$

where **m** is a random function or simply a random variable. It is fairly usual to generate **m** as a Gaussian vector with uncorrelated components. However, the use of a carefully chosen correlation matrix is part of the history of evolutionary techniques (vide, for instance, [255]), and it is also possible to resort to other probability density functions if any feature of the problem suggests their use. Figure 8.5 illustrates an example in a two-dimensional search space where **x** corresponds to the point $(0.5, 0.5)$ and the cloud around it are the solutions generated by three different mutation operators.

A selection process may immediately follow the application of the crossover and mutation operators, and we may consider two interesting possibilities. A first one is related to the idea of elitism and consists in reintroducing the best individual found so far. A second alternative is to allow, in the process of restoring the population size, the introduction of a combination of good, reasonable, and bad solutions, which typically increases the population diversity and, consequently, the global search potential [209].

Much more could be said about GAs, but a more systematic explanation certainly transcends our objectives. On the other hand, it is important

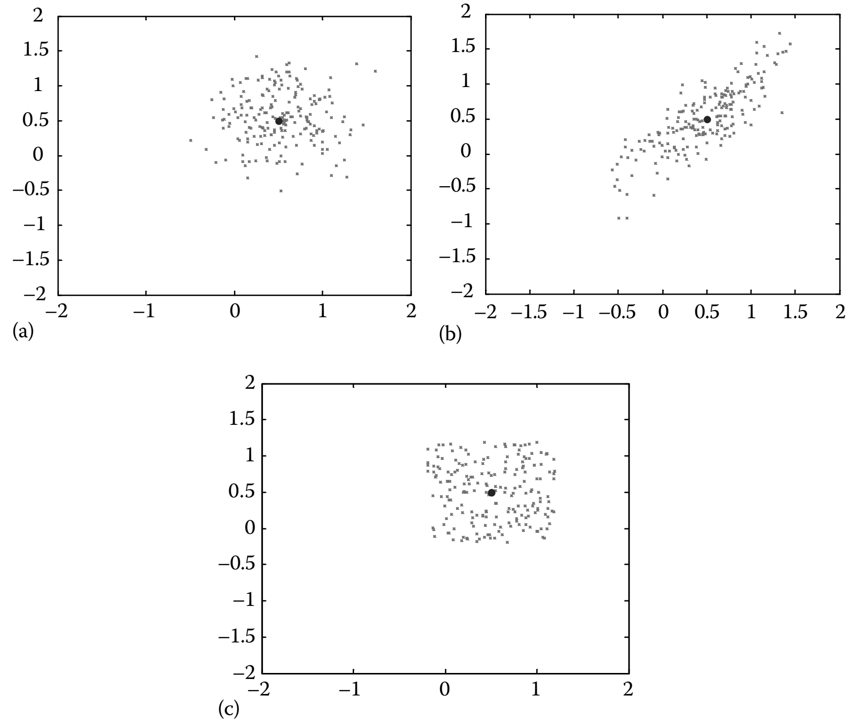


FIGURE 8.5
Samples generated by real mutation operators built from distinct joint pdfs: (a) white Gaussian mutation, (b) correlated Gaussian mutation, and (c) uniform mutation.

to illustrate the application of AG in an example related to the scope of this book. In this example, we employ a GA to perform the search for the equalizer that minimizes the CM cost function, discussed in Chapter 4.

Example 8.1 Blind Equalization Using Genetic Algorithms

Let us consider two nonminimum phase channels

$$h_1(z) = 1 + 0.4z^{-1} + 0.9z^{-2} + 1.4z^{-3} \tag{8.3}$$

and

$$h_2(z) = 1 + 1.2z^{-1} - 0.3z^{-2} + 0.8z^{-3} \tag{8.4}$$

Each individual represents a possible equalizer, and real coding is employed. The simulations were run using the following set of parameters:

- Population size: 30 individuals
- Number of crossovers per generation: 10
- Probability of occurence of mutation: 0.1
- Stopping criterion: 2000 generations

In a first test, a 5-tap filter is employed to the equalizer channel $h_1(z)$. Thus, each individual in the population corresponds to a possible filter, and is represented by a vector with five elements. The solutions obtained in this scenario were used to initialize a DD algorithm, which ideally converges to the closest Wiener solution.

Notice that this test is carried out in a relatively small search space, and the filter order cannot considerably reduce the MSE. Table 8.1 has been built from the outcomes of 50 simulations. The frequency of each solution is also indicated, i.e., the number of trials in which the algorithm converged to a given solution. The results reveal that the method has always provided a solution rather close to the global Wiener minimum, which clearly indicates a very good performance.

Let us now turn our attention to a larger and more complex search space with an 8-tap filter, for which Table 8.2 brings the corresponding results. We notice that convergence to good minima was predominant, with an expressive global convergence rate.

The last scenario is formed by $h_2(z)$ and a 7-tap equalizer, an order high enough to provide a condition close to the ZF one. Table 8.3 shows the corresponding results. Again, we have a good global convergence rate, thus revealing once more the method efficiency.

GAs are probably the most widespread evolutionary approaches, but there are many different paradigms of this class that can be efficiently used.

TABLE 8.1

Solution for a 5-Tap Equalizer for Channel $h_1(z)$

Solution	Residual MSE	Frequency
[0.2183, −0.1873, −0.0596, −0.2804, 0.5892]	0.1751	100

TABLE 8.2

Solution for an 8-Tap Equalizer for Channel $h_1(z)$

Solution	MSE	Frequency (%)
[0.198, −0.142, −0.118, −0.227, 0.520, 0.115, −0.113, 0.085]	0.129	48
[−0.119, 0.053, −0.023, 0.283, −0.237, −0.047, −0.305, 0.615]	0.140	22
[−0.024, 0.226, −0.148, −0.106, −0.276, 0.554, 0.119, −0.067]	0.145	12
[−0.024, −0.025, 0.233, −0.141, −0.086, −0.301, 0.547, 0.091]	0.153	10
[−0.135, −0.143, 0.371, 0.199, −0.146, 0.171, −0.134, 0.057]	0.189	4
[−0.025, −0.112, −0.146, 0.383, 0.161, −0.123, 0.173, −0.101]	0.195	4

TABLE 8.3

Solution for a 7-Tap Equalizer for Channel $h_2(z)$

Solution	MSE	Frequency (%)
[−0.084, 0.154, −0.230, 0.406, 0.289, −0.052, −0.153]	0.0312	48
[0.142, −0.248, 0.383, 0.307, −0.039, −0.145, −0.042]	0.0458	40
[−0.228, 0.412, 0.346, −0.067, −0.166, −0.054, 0.062]	0.0917	8
[0.040, −0.085, 0.158, −0.225, 0.374, 0.265, −0.071]	0.0918	2
[−0.037, 0.047, −0.0745, 0.173, −0.245, 0.359, 0.255]	0.1022	2

In Section 8.3, we turn our attention to a class of optimization tools derived from theories concerning the immune system and that belong to the broad field of AISs. Interest in these techniques is particularly justified by an attractive balance between local and global search mechanisms.

8.3 Artificial Immune Systems

The AIS we discuss here is an artificial immune network proposed to solve multimodal optimization tasks, the opt-aiNet. The opt-aiNet is conceptually related to the adaptive immune response of organisms, i.e., to the ability that the defense system has of increasing their potential of responding to intruders previously met. The algorithm, first proposed in [86], is based on two main concepts: clonal selection and affinity maturation [4, 51], and the idea of immune network [154]. The synergy between these elements gives rise to a tool with mechanisms for local refinement, diversity maintenance, and population-size control, which establish an interesting option to deal with multimodal signal processing problems.

The combination between the clonal selection principle [51] and the notion of affinity maturation [4] is computationally responsible for a process of refinement of all solutions via replication followed by hypermutation. The immune network theory [154] brings the possibility that there be eigenbehaviors of the immune system that may be relevant, even in the absence of invaders, and introduces a mechanism of insertion/pruning that will be decisive in terms of multimodal search potential and computational parsimony.

Each solution, which is considered to be real-coded, is thought of as being related to the structure of a given antibody. The affinity between an antibody and the antigen to be identified is measured by the cost function, also referred to as fitness function. Finally, the affinity between antibodies is quantified via a Euclidean distance measure. The search process combines the cloning/mutation process and the outcomes of the interactions between the antibodies themselves. The above description can be clarified by an analysis of the pseudocode presented in Algorithm 8.5.

In steps 2(a) to 2(e), we have essentially a process of mutation proportional to the fitness value that has a decisive role in terms of local search, although it may as well have some impact in terms of global search. Notice that mutation is applied (N_c times) to each individual, and that a deterministic selection procedure is responsible for locally keeping the population size intact. The fact that the mutation is proportional to the fitness value is responsible for imposing more pronounced modifications to "less adapted" individuals, which establishes a certain control over the convergence rate of the technique.

In step 2(f), we have a key feature of the method, an interesting mechanism to verify if the search process has been accomplished or if it is still leading to improvements and to modify the population structure in accordance with this verification. If there is not a significant amount of variation in the average fitness, a process of pruning is activated, in order to remove individuals that are very close to each other (step 2(g)) or, in other words, that are located in the same region of the search space. This mechanism is very similar to the idea underlying certain niching methods [196, 259], in which, for instance, a given radius defines a niche, and only the fittest individual of each niche is allowed to survive. This reduces the population to a minimum in terms of maintenance of relevant information about the cost function.

A removal of redundant solutions is followed by the introduction of new randomly generated antibodies (step 2(h)) that have the potential of occupying previously unexplored regions of the search space.

The algorithm combines two apparently contradictory aims: global search potential and efficiency in the use of computational resources. Naturally, the proper operation of the algorithm depends on the difficult task of properly tuning fundamental parameters, like the affinity threshold.

Both GA and AIS provide a view of the potential of evolutionary algorithms as optimization tools. Next, we consider a bio-inspired approach that is based on a different analogy and that has received a great deal of attention in the last years.

Algorithm 8.5: The opt-aiNet algorithm

1. (*Initialization*): randomly initialize a population with a small number of individuals;
2. While the stopping criterion is not met do:
   - (a) (*Fitness evaluation*) Determine the fitness of each individual of the population and normalize the fitness vector;
   - (b) (*Replication*) Generate a number of copies (offspring) of each individual;
   - (c) (*Mutation*) Mutate each of these copies in a manner that be inversely proportional to the fitness of its parent cell, and also keep the parent cell. The mutation is given by

     $$\begin{aligned} \mathbf{c}' &= \mathbf{c} + \alpha \mathbf{N}(0,1) \\ \alpha &= \frac{1}{\beta} \exp(-\text{fit}^*) \end{aligned} \tag{8.5}$$

     where

     $\mathbf{c}'$ is a mutated version of $\mathbf{c}$, $\mathbf{N}(0,1)$ is a zero-mean Gaussian random vector with uncorrelated unit variance elements

     β is a parameter that controls the decay of an inverse exponential function

     fit^* is the fitness of an individual normalized to lie in the interval $[0,1]$

     A mutation is only accepted if the mutated individual $\mathbf{c}'$ is within the proper domain;
   - (d) (*Fitness evaluation*) Determine the fitness of all new (mutated) individuals of the population;
   - (e) (*Selection*) For each clone group formed by the parent individual and its mutated offspring, select the individual with highest fitness and calculate the average fitness of the selected population;
   - (f) (*Local convergence*) If the average fitness of the population is not significantly different from that at the previous iteration, then continue; otherwise return to step 2(a);
   - (g) (*Network interactions*) Determine the affinity (degree of similarity measured via the Euclidean distance) of all individuals of the population. Suppress all those individuals whose affinities are less than a suppression threshold σ_s, except the best of them, and determine the number of network individuals, termed memory cells, after suppression;
   - (h) (*Diversity introduction*) Introduce a percentage of randomly generated individuals and return to step 2

8.4 Particle Swarm Optimization

The study of the behavior of social animals is receiving a significant attention in the field of optimization with the increasing interest in applications of the notion of swarm intelligence to the solution of a number of relevant tasks [46]. In this section, we concentrate our attention on a search methodology that has been used with success in contexts similar to those that typically arise in signal processing applications: PSO [167].

As in the previous methods, a given cost function provides the fundamental source of information about the problem to be solved. The classical implementation of the PSO method is based on an update rule that combines individual knowledge about the problem with knowledge of a collective nature.

Each of these kinds of knowledge originates a search direction, and these directions are linearly combined in the update expression of each individual, which is usually coded as a real-valued vector. We incorporate collective knowledge by establishing an a priori neighborhood that will be the essential pattern of relationship between individuals. For instance, in Figure 8.6a we assume that all individuals are neighbors, while in Figure 8.6b each individual is supposed to have two neighbors.

Each solution is considered as a particle whose position is given by its parameter vector. The optimization process can be conceived in terms of the movement of the set of particles that forms a population. This movement must be determined by the cost function, so that "good regions" of the cost function become more attractive than "bad regions," and this must have some impact on the behavior of the population as a whole. For each solution, we have access to the corresponding value of the cost function, which will be used by the algorithm. Furthermore, in analogy to the accumulation

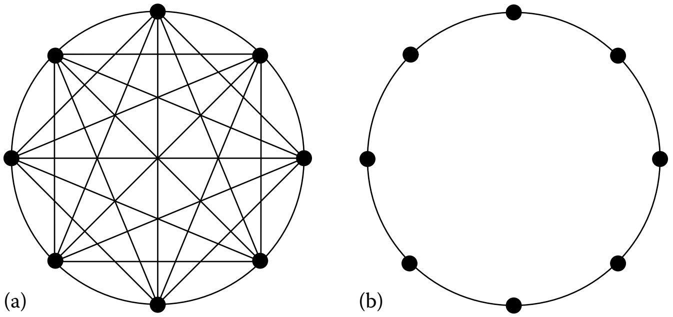


FIGURE 8.6
Examples of possible neighborhood topologies: (a) "ring-like neighborhood" and (b) complete neighborhood.

of knowledge in an individual life, we assume that each particle "remembers" the best position it has so far occupied. Hence, we may establish a search direction that indicates a path from the place presently occupied by the particle to the best place it has ever occupied:

$$\mathit{direction}_1 = \mathbf{p}_i - \mathbf{x}_i(n) \tag{8.6}$$

where

$\mathbf{p}_i$ is the best (in terms of the cost function to be optimized) position visited by the ith particle so far

$\mathbf{x}_i(n)$ is its position at the instant n

However, collective life is not guided exclusively by individual experience, but also by the experience accumulated by the group [85, 167]. In our algorithm, this is modeled via the inclusion of another search direction, which indicates to each individual the path toward the best solution reached by its neighbors:

$$\mathit{direction}_2 = \mathbf{g}_i - \mathbf{x}_i(n) \tag{8.7}$$

where $\mathbf{g}_i$ is the best position visited by any neighbor of the ith particle.

The two calculated directions are included in the determination of the vector $\mathbf{v}_i(n+1)$, i.e., the vector that will be responsible for updating the position of the particle whose current position is $\mathbf{x}_i(n)$. This vector is commonly referred to as the velocity vector of the particle, and its update classically follows the rule

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \boldsymbol{\varphi}_1 \odot \mathit{direction}_1 + \boldsymbol{\varphi}_2 \odot \mathit{direction}_2 \tag{8.8}$$

where

$\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ are vectors of uniformly distributed positive numbers

$\odot$ represents the Hadamard product

Then, the updated position of the ith particle is

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{v}_i(n+1) \tag{8.9}$$

We usually impose fixed upper and lower limits to each value of the velocity vector in order to avoid unlimited expansion of its elements. It is also important to mention that (8.8) corresponds to the classical update expression for the velocity vector. Other alternatives are found in the literature, like those based on an inertia term or a global constriction coefficient [85]. Algorithm 8.6 summarizes a PSO technique.

Algorithm 8.6: Particle Swarm Optimization Algorithm

```
1. Initialize the population and the velocity vectors of all individuals;
2. While stopping criterion is not met, do:
   (a) For each particle $i$
       i. If $f(\mathbf{x}_i(n)) > f(\mathbf{p}_i(n))$ then $\mathbf{p}_i(n+1) = \mathbf{x}_i(n)$
       ii. For all neighbors $j$
           A. If $f(\mathbf{p}_j(n)) > f(\mathbf{g}_i(n))$ then $\mathbf{g}_i(n+1) = \mathbf{p}_j(n)$
   (b) $\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \varphi_1 \odot (\mathbf{p}_i(n+1) - \mathbf{x}_i(n)) + \varphi_2 \odot (\mathbf{g}_i(n+1) - \mathbf{x}_i(n))$, and all elements of $\mathbf{v}_i(n+1)$ must belong to the interval defined by the minimum and the maximum velocities;
   (c) $\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{v}(n+1)$
```

The PSO algorithm is becoming an increasingly popular bio-inspired tool to solve optimization problems, and many variants of the classical algorithm we have studied are being proposed in the literature [73, 273]. Our objective in this section is to give an idea of the basic mechanisms underlying the technique. Such mechanisms combine local and global search potential, which indicates a great potential for application in modern signal processing.

As an illustrative example, we employ in the sequel both AISs and particle swarm in a blind source separation problem, in which a PNL model is employed for the mixing process, as discussed in Section 6.6.

Example 8.2 Blind Source Separation Using AIS and PSO

The mixing system is defined by

$$\mathbf{A} = \begin{bmatrix} 1 & 0.6 \\ 0.5 & 1 \end{bmatrix} \quad \text{and} \quad \begin{array}{l} f_1(e_1) = \tanh(2e_1) \\ f_2(e_2) = 2\sqrt[5]{e_2} \end{array} \tag{8.10}$$

The separating system to be optimized consists of a square matrix **W** and a polynomial of fifth order, only with odd powers, given by

$$y = ax^5 + bx^3 + cx \tag{8.11}$$

Thus, each individual in the population is represented by a set of 10 parameters—4 values that define **W** and 3 values for each nonlinearity (coefficients *a*, *b*, and *c*).

Since the separability property [281] of the PNL model requires that $\mathbf{g}(\mathbf{f}(\cdot))$ be a monotonic function, the coefficients of each polynomial were restricted to be positive. The parameters of the opt-aiNet were set to the following values:

- Initial population: 10 Individuals
- Number of clones (N_c): 7 Clones
- β: 60
- Suppression threshold: 2

On the other hand, for the particle swarm algorithm, the parameters used in the experiment are

- Number of particles: 60
- $AC_1 = AC_2 = 2.05$
- $V_{max} = 0.1$
- $V_{min} = -0.1$

In Figure 8.7a, the joint distribution of the mixture signals is shown. For this situation, we considered 2000 samples of the mixtures in the training stage. Figure 8.7b depicts the joint distribution of the recovered signals using the artificial immune network.

We can observe that a residual nonlinear distortion remains, as it is impossible to invert the hyperbolic tangent using the chosen polynomial. Nevertheless, the obtained distribution is approximately uniform, indicating that the separation task was accomplished. Similar results are obtained with the particle swarm.

In Table 8.4, we depict the residual MSE between the estimated signal and the corresponding source. It can be noted that both algorithms were able to build the separating system quite well, yielding small residual errors.

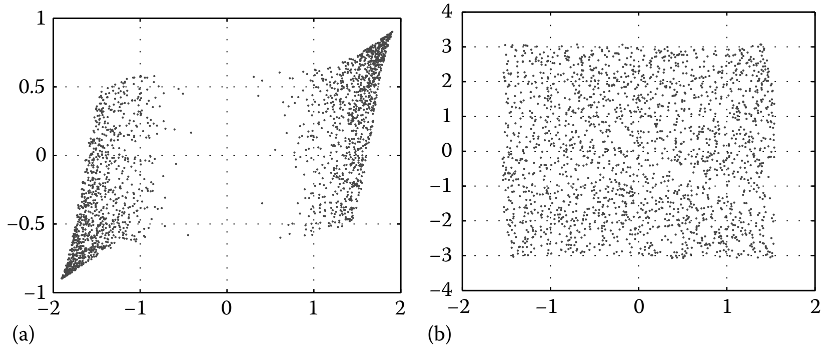


FIGURE 8.7
Distributions of the (a) mixtures and of the (b) recovered sources.

TABLE 8.4

Residual MSE of the Estimated Sources Using an Artificial Immune Network and a Particle Swarm Algorithm

Algorithm	MSE—Source 1	MSE—Source 2
Artificial immune network	0.11×10^{-2}	0.65×10^{-2}
Particle swarm	0.73×10^{-2}	0.53×10^{-2}

8.5 Concluding Remarks

The natural tendency to explore alternative techniques and the requirements of some concrete applications have led signal processing theory toward a more widespread use of nonlinear structures and unsupervised adaptation criteria. Extensions of this nature are likely to originate optimization tasks of a nature considerably different from that associated with the more classical frameworks. Then, the search of alternative optimization methods to improve the global search potential may become attractive.

In this chapter, we discussed some approaches belonging to the field of bio-inspired computing. Our intention, it is important to remark, was not to provide a complete view of the field, but to make use of three important classes of methods: GAs, AISs, and PSO. The main goal is to analyze essential features of these tools and evaluate their potential of application in our problems of interest.

We observed that a GA has the following features: there is local search, there are elements of global search, and there is the influence of selection in the survival of certain solutions in consonance with the characteristics of the function to be optimized. It is worth noting that the method requires information about the cost function itself, but usual requirements such as continuity, differentiability, etc., do not play a key role here. Hence, it is particularly attractive when we deal with cost functions that are very difficult, or even impossible, to manipulate. Certainly, in GA and other techniques, complexity is a factor to be considered in many online applications. Nevertheless, it does not discard the theoretical interest of the methods neither their applications in contexts where the real-time operation at high information rate is not an issue.

Overall, we hope this chapter could serve to instigate a larger group of the signal processing community to consider bio-inspired methods as viable components of their toolkit.

Appendix A: Some Properties of the Correlation Matrix

Let us first consider the context of the linear combiner as described in Figure A.1 and the general case of complex-valued signals.

Let $\mathbf{x}(n)$ denote the input vector of the linear combiner. The correlation matrix associated therewith is defined as

$$\mathbf{R} = E\left[\mathbf{x}(n)\mathbf{x}^H(n)\right] \tag{A.1}$$

where $E[\cdot]$ is the statistical expectation operator. A first remark is that the correlation matrix is a $K \times K$ square matrix, K being the number of input signals that compose $\mathbf{x}(n)$.

A.1 Hermitian Property

In accordance with (A.1), it is straightforward to verify that the matrix $\mathbf{R}$ must be Hermitian, as the correlation between, for instance, inputs $x_1(n)$ and $x_2(n)$, is equal to the complex conjugate of the correlation between $x_2(n)$ and $x_1(n)$. In other words,

$$E\left[x_1(n)x_2^*(n)\right] = \left\{E\left[x_2(n)x_1^*(n)\right]\right\}^* \tag{A.2}$$

A.2 Eigenstructure

First, let us present the classical definition of eigenvalues and eigenvectors of a matrix $\mathbf{R}$:

$$\mathbf{R}\mathbf{q}_i = \lambda_i \mathbf{q}_i \quad \textit{for } i = 1, \ldots, K \tag{A.3}$$

where λ_i and $\mathbf{q}_i$ represent, respectively, the eigenvalues and eigenvectors of $\mathbf{R}$.

The K eigenvectors $\mathbf{q}_1, \ldots, \mathbf{q}_K$ are orthogonal to each other, thus forming a suitable basis for signal representation, which establishes the so-called

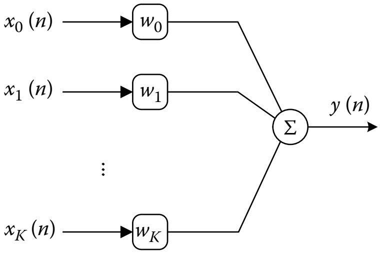


FIGURE A.1
Representation of a linear combiner.

Karhunen–Loève decomposition [230]. We can also define

$$\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_K] \tag{A.4}$$

as the matrix whose columns are the eigenvectors. It is readily seen that $\mathbf{Q}$ is orthogonal, i.e.,

$$\mathbf{Q}^H = \mathbf{Q}^{-1} \tag{A.5}$$

An implication of this property is the use of $\mathbf{Q}$ as a whitening matrix, as discussed in Section 6.2.1.

Considering (A.4), it is possible to represent matrix $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H \tag{A.6}$$

where $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal is composed of the eigenvalues of the correlation matrix.

Another property of interest is the relationship between the eigenvalues and their respective eigenvectors, which is expressed in the form of a Rayleigh quotient, defined by

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i} \tag{A.7}$$

The quotient is employed in the definition of eigenfilters, which are used in noise reduction tasks as well as in high-definition methods of spectral analysis [139].

Now, if we consider $y(n)$ as the output of a linear combiner, the coefficients of which are the elements of a vector $\mathbf{w}$, we can write

$$E\left[|y(n)|^2\right] = E\left[(\mathbf{w}^H\mathbf{x}(n))(\mathbf{w}^H\mathbf{x}(n))^H\right] = E\left[\mathbf{w}^H\mathbf{x}\mathbf{x}^H\mathbf{w}\right] = \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{A.8}$$

Since the mean-square value of $y(n)$ is necessarily nonnegative, we have

$$\mathbf{w}^H\mathbf{R}\mathbf{w} \geq 0 \tag{A.9}$$

From (A.9), we conclude that the matrix **R** is positive semidefinite [139], and, as a consequence, all of its eigenvalues are nonnegative.

A.3 The Correlation Matrix in the Context of Temporal Filtering

Some additional features may be commented about the structure of **R** when the input vector is formed by delayed versions of a same signal, i.e., in the case of temporal filtering.

As already presented, we have in such case

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-K+1)]^T \tag{A.10}$$

so that the elements of the correlation matrix are given by

$$r(k) = E\left[x(n)x^*(n-k)\right] \tag{A.11}$$

Since $x(n)$ is assumed to be stationary, the elements $r(k)$ are a function exclusively of the lag between samples, and do not depend on the time instant. Consequently, the columns and rows of matrix **R** are formed from a same vector $\mathbf{r} = [r(0), r(1), \ldots, r(K-1)]^T$, the elements of which are $r(k)$, for $k = 1, \ldots, N$, by performing subsequent circular shifts and appropriate complex conjugation:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1)^* & \cdots & r(K)^* \\ r(1) & r(0) & \cdots & r(K-1)^* \\ \vdots & \vdots & \ddots & \vdots \\ r(K) & r(K-1) & \cdots & r(0) \end{bmatrix} \tag{A.12}$$

This fact implies the so-called Toeplitz structure of **R** in this case. This property is particularly interesting in recursive methods for matrix inversion and for solving linear equation systems. For instance, the Levinson–Durbin algorithm, which recursively yields the coefficients of an optimal linear predictor, is based on the Toeplitz structure of the correlation matrix.

Appendix B: Kalman Filter

In simple terms, the Kalman filter is an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements. There is a vast literature covering several aspects of the Kalman filter [16, 139, 165]. In this appendix, we will focus on a more intuitive view of this versatile tool.

B.1 State-Space Model

The set of variables that provide a complete representation about the internal configuration of the system at a given time instant is known as the system state [16, 165]. More rigorously, one can define the system state at a time instant n_0 as the minimum amount of information that, along with the input for $n \geq n_0$, uniquely determines the system outputs for $n \geq n_0$ [165].

In order to perform state estimation, one should build a mathematical representation of the system at hand. It is usual to represent the dynamical system in terms of a pair of equations: the process equation, which defines the dynamics of the state variables; and the measurement equation, which describes the observation vector. In mathematical terms we express these two equations as

$$\mathbf{x}(n) = \mathbf{F}(n)\mathbf{x}(n-1) + \mathbf{G}(n)\mathbf{u}(n) \quad \text{(process equation)} \tag{B.1}$$

$$\tilde{\mathbf{y}}(n) = \boldsymbol{\mathcal{X}}(n)\mathbf{x}(n) + \tilde{\mathbf{n}}(n) \quad \text{(measurement equation)} \tag{B.2}$$

where

$\mathbf{x}(n)$ represents the state vector
$\mathbf{F}(n)$ represents the state transition matrix
$\mathbf{u}(n)$ is a process noise vector
$\mathbf{G}(n)$ represents the input matrix
$\boldsymbol{\mathcal{X}}(n)$ is the observation matrix
$\tilde{\mathbf{n}}(n)$ is an additive measurement noise
$\tilde{\mathbf{y}}(n)$ is the observed vector at time instant n

Figure B.1 illustrates the signal flow in a linear dynamical system.

Noise vectors $\mathbf{u}(n)$ and $\tilde{\mathbf{n}}(n)$ can be considered to be white, uncorrelated, zero-mean processes, with covariance matrices given by

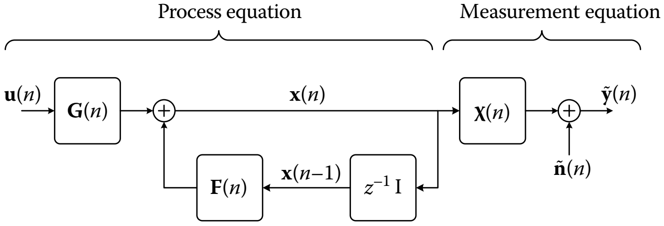


FIGURE B.1
Representation of a linear, discrete-time dynamical system.

$$\mathrm{E}\left[\mathbf{u}(n)\mathbf{u}(k)^H\right] = \begin{cases} \mathbf{Q}(n), & n = k \\ \mathbf{0}, & n \neq k \end{cases} \quad \text{and } \mathrm{E}\left[\tilde{\mathbf{n}}(n)\tilde{\mathbf{n}}(n)^H\right] = \begin{cases} \mathbf{R}(n), & n = k \\ \mathbf{0}, & n \neq k \end{cases} \tag{B.3}$$

Hence, given the state-space model of a system, the objective of the Kalman filter can be formally stated as follows.

DEFINITION B.1 (Discrete-Time Kalman Filtering Problem) Consider the linear, finite-dimensional, discrete-time system represented by (B.1) and (B.2), defined for $n \geq 0$. Let the noise sequences $\{\mathbf{u}(n)\}$ and $\{\tilde{\mathbf{n}}(n)\}$ be independent, zero-mean, Gaussian white processes with covariance matrices given by (B.3). Using the entire observed data $\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n)$, find the minimum mean-square estimate of the state $\mathbf{x}(i)$.

The above definition encompasses three slightly different problems that can be solved by the Kalman filter. If $i = n$, i.e., we want to estimate the current state based on observations up to time index n, we have a filtering problem. On the other hand, if $i > n$ it means we are facing a prediction problem. Finally, if $1 \leq i < n$ we have a so-called smoothing problem, in which we observe a longer sequence of observations to estimate the state.

B.2 Deriving the Kalman Filter

A well known result from the estimation theory is that the minimum mean-squared error (MMSE) estimator is given by the conditional mean

$$\hat{\mathbf{x}}_{\mathrm{MMSE}} = E\left[\mathbf{x} | \{\tilde{\mathbf{y}}\}\right] \tag{B.4}$$

where $\{\tilde{\mathbf{y}}\}$ denotes the observation sequence. The Kalman filter can be interpreted as a tool that implements a sequential MMSE estimator, exploring the dynamical system model to fulfill this aim recursively.

In general terms, the recursive procedure employed by the Kalman filter can be divided into two steps: one prediction step, in which the process equation is used to provide an estimate of the state vector $\mathbf{x}(n)$ based on the estimate obtained at time instant $n-1$; and a filtering step, which takes the predicted value and update the estimate based on the measurement equations and the observed signals at time instant n.

From (B.4), the best estimate at time instant n, given the observations $\tilde{\mathbf{y}}$ up to this instant, is given by

$$\hat{\mathbf{x}}(n|n) = E\left[\mathbf{x}(n)|\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n)\right] \tag{B.5}$$

On the other hand, if we have access to all measurements except for the sample at time instant n, we can compute an a priori estimate given by the conditional mean

$$\hat{\mathbf{x}}(n|n-1) = E\left[\mathbf{x}(n)|\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n-1)\right] \tag{B.6}$$

Both $\hat{\mathbf{x}}(n|n)$ and $\hat{\mathbf{x}}(n|n-1)$ are estimates of $\mathbf{x}(n)$. The difference is that $\hat{\mathbf{x}}(n|n-1)$ is computed prior to the observation of $\tilde{\mathbf{y}}(n)$, while $\hat{\mathbf{x}}(n|n)$ is obtained taking into account the information in $\tilde{\mathbf{y}}(n)$.

Using the state equations, we can express (B.6) as

$$\begin{aligned}
\hat{\mathbf{x}}(n|n-1) &= E\left[\mathbf{F}(n)\mathbf{x}(n-1) + \mathbf{G}(n)\mathbf{u}(n)|\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n-1)\right] \\
&= \mathbf{F}(n)E\left[\mathbf{x}(n-1)|\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n-1)\right] \\
&\quad + \mathbf{G}(n)E\left[\mathbf{u}(n)|\tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots, \tilde{\mathbf{y}}(n-1)\right] \\
&= \mathbf{F}(n)\hat{\mathbf{x}}(n-1|n-1) + \mathbf{G}(n)E\left[\mathbf{u}(n)\right] \\
&= \mathbf{F}(n)\hat{\mathbf{x}}(n-1|n-1)
\end{aligned} \tag{B.7}$$

since $\mathbf{u}(n)$ is zero-mean and is independent from the observations.

One possible way of evaluating the estimates $\hat{\mathbf{x}}(k|k-1)$ for each instant n is to compute the estimation error covariance matrix $\mathbf{P}(n|n-1)$, given by

$$\mathbf{P}(n|n-1) = E\left[\big(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1)\big)\big(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1)\big)^H\right] \tag{B.8}$$

Using (B.1), (B.3), and (B.7), one obtains

$$\begin{aligned}\mathbf{P}(n|n-1) &= E\Big[\big(\mathbf{F}(n)\mathbf{x}(n-1)+\mathbf{G}(n)\mathbf{u}(n)-\mathbf{F}(n)\hat{\mathbf{x}}(n-1|n-1)\big)\\ &\qquad \big(\mathbf{F}(n)\mathbf{x}(n-1)+\mathbf{G}(n)\mathbf{u}(n)-\mathbf{F}(n)\hat{\mathbf{x}}(n-1|n-1)\big)^H\Big]\\ &= E\Big[\big(\mathbf{F}(n)\left(\mathbf{x}(n-1)-\hat{\mathbf{x}}(n-1|n-1)\right)+\mathbf{G}(n)\mathbf{u}(n)\big)\\ &\qquad \Big(\mathbf{F}(n)\left(\mathbf{x}(n-1)-\hat{\mathbf{x}}(n-1|n-1)\right)+\mathbf{G}(n)\mathbf{u}^H(n)\Big)\Big]\\ &= \mathbf{F}(n)\mathbf{P}(n-1|n-1)\mathbf{F}^H(n)+\mathbf{G}(n)\mathbf{Q}(n)\mathbf{G}^H(n) \qquad \text{(B.9)}\end{aligned}$$

where the a posteriori covariance matrix, $\mathbf{P}(n-1|n-1)$, is given by

$$\mathbf{P}(n-1|n-1) = E\left[\left(\mathbf{x}(n-1)-\hat{\mathbf{x}}(n-1|n-1)\right)\left(\mathbf{x}(n-1)-\hat{\mathbf{x}}(n-1|n-1)\right)^H\right] \qquad \text{(B.10)}$$

Observing (B.7) and (B.9), we notice that the estimation error covariance matrix and the state are updated based solely on the system dynamical model and the previous estimate. Hence, these two equations correspond to the prediction step in the Kalman filter.

The next step in the estimation process is to refine the state and error covariance matrix estimates, taking into account the observed sample at time instant n. To this end, let us consider a recursive linear estimator

$$\hat{\mathbf{x}}(n|n) = \mathbf{C}(1,n)\hat{\mathbf{x}}(n|n-1)+\mathbf{C}(2,n)\tilde{\mathbf{y}}(n) \qquad \text{(B.11)}$$

where

$\hat{\mathbf{x}}(n|n-1)$ denotes the estimate obtained by the prediction step
$\tilde{\mathbf{y}}(n)$ denotes the observed vector at time instant n
$\mathbf{C}(1,n)$ and $\mathbf{C}(2,n)$ represent the time-varying coefficients of the linear combination

Using (B.2), it is possible to show that the coefficients $\mathbf{C}(1,n)$ and $\mathbf{C}(2,n)$ that minimize the MSE at each step are given by [16,139,165]

$$\begin{aligned}\mathbf{C}(1,n) &= \mathbf{I}-\mathbf{K}(n)\boldsymbol{\mathcal{X}}(n)\\ \mathbf{C}(2,n) &= \mathbf{K}(n),\end{aligned} \qquad \text{(B.12)}$$

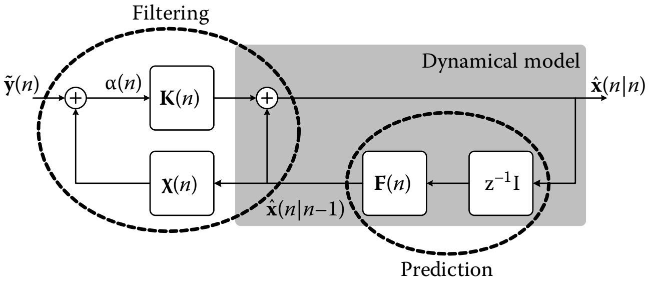


FIGURE B.2
Block diagram of the Kalman filter.

where $\mathbf{K}(n)$ is the *Kalman gain*, computed as

$$\mathbf{K}(n) = \mathbf{P}(n|n-1)\boldsymbol{\mathcal{X}}(n)^H \left(\boldsymbol{\mathcal{X}}(n)\mathbf{P}(n|n-1)\boldsymbol{\mathcal{X}}(n)^H + \mathbf{R}(n)\right)^{-1} \tag{B.13}$$

Thus, the optimum recursive linear estimator is expressed by

$$\hat{\mathbf{x}}(n|n) = (\mathbf{I} - \mathbf{K}(n)\boldsymbol{\mathcal{X}}(n))\,\hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\tilde{\mathbf{y}}(n) \tag{B.14}$$

Reordering the terms in (B.14), we get to

$$\begin{aligned}\hat{\mathbf{x}}(n|n) &= \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\left(\tilde{\mathbf{y}}(n) - \mathcal{X}(n)\hat{\mathbf{x}}(n|n-1)\right)\\ &= \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\boldsymbol{\alpha}(n)\end{aligned} \tag{B.15}$$

Therefore, from (B.15), it is possible to interpret the estimation provided by the Kalman filter as being a linear combination of $\mathbf{x}(n)$ based on past samples, $\mathbf{x}(n|n-1)$, and a correction term $\boldsymbol{\alpha}(n) = \tilde{\mathbf{y}}(n) - \boldsymbol{\mathcal{X}}(n)\hat{\mathbf{x}}(n|n-1)$, which represents the error committed when we try to estimate $\tilde{\mathbf{y}}(n)$ from $\hat{\mathbf{x}}(n|n-1)$.

Figure B.2 illustrates the operation of the Kalman filter, indicating the relationship between the prediction and filtering steps, and the central role played by the system model in the estimation. If we compare Figures B.1 and B.2 it becomes clear that the correction term $\mathbf{K}(n)\boldsymbol{\alpha}(n)$ can be viewed as an estimate of the process noise $\mathbf{G}(n)\mathbf{u}(n)$, which is fed into the system model to provide the filtered estimate $\mathbf{x}(n|n)$.

To conclude our exposition of the Kalman filter, we just need to show how the estimation error covariance matrix is updated in the filtering step. As shown in [16, 139, 165], matrix $\mathbf{P}(n|n)$ is given by

$$\mathbf{P}(n|n) = [\mathbf{I} - \mathbf{K}(n)\boldsymbol{\mathcal{X}}(n)]\,\mathbf{P}(n|n-1) \tag{B.16}$$

Combining (B.7), (B.9), (B.13), (B.15), and (B.16), we obtain the Kalman filter, whose algorithm is summarized in Algorithm B.1.

Algorithm B.1: Kalman Filter

`Prediction`

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{F}(n)\hat{\mathbf{x}}(n-1|n-1) \tag{B.17a}$$

$$\mathbf{P}(n|n-1) = \mathbf{F}(n)\mathbf{P}(n-1|n-1)\mathbf{F}^H(n) + \mathbf{G}(n)\mathbf{Q}(n)\mathbf{G}^H(n) \tag{B.17b}$$

`Filtering`

$$\mathbf{K}(n) = \mathbf{P}(n|n-1)\mathcal{X}(n)^H\left(\mathcal{X}(n)\mathbf{P}(n|n-1)\boldsymbol{\mathcal{X}}^H(n) + \mathbf{R}_n\right)^{-1} \tag{B.18a}$$

$$\boldsymbol{\alpha}(n) = \tilde{\mathbf{y}}(n) - \boldsymbol{\mathcal{X}}(n)\hat{\mathbf{x}}(n|n-1) \tag{B.18b}$$

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\boldsymbol{\alpha}(n) \tag{B.18c}$$

$$\mathbf{P}(n|n) = \left[\mathbf{I} - \mathbf{K}(n)\mathcal{X}(n)\right]\mathbf{P}(n|n-1) \tag{B.18d}$$

References

1. R. Poli and S. Cagnoni. Genetic and evolutionary computation for signal processing and image analysis. Special issue, *EURASIP Journal on Applied Signal Processing*, vol. 8, 2003.
2. K. Abend and B. D. Fritchman. Statistical detection for communication channels with intersymbol interference. *Proceedings of the IEEE*, 58:779–785, 1970.
3. S. Achard and C. Jutten. Identifiability of post-nonlinear mixtures. *IEEE Signal Processing Letters*, 12(5):423–426, May 2005.
4. G. L. Ada and J. V. Nossal. The clonal selection theory. *Scientific American*, 256:50–57, 1987.
5. O. Agazzi, D. Messerschmitt, and D. Hodges. Nonlinear echo cancellation of data signals. *IEEE Transactions on Communications*, 30(11):2421–2433. November 1982.
6. L. B. Almeida. Linear and nonlinear ICA based on mutual information. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium* (*AS-SPCC*), Lake Louise, Canada, 2000, pp. 117–122.
7. J. Altuna. Cyclostationarity blind equalisation in mobile communications. PhD thesis, The University of Edinburgh, Edinburgh, Scotland, October 1997.
8. S.-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.
9. S.-I. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems 8* (*Proceedings of NIPS '95*), MIT Press, Cambridge, MA, 1996, pp. 757–763.
10. S.-I. Amari, A. Cichocki, and H. H. Yang. Blind signal separation and extraction: Neural and information theoretic approaches, *Unsupervised Adaptive Filtering* Volume I: *Blind Source Separation*, John Wiley & Sons, New York, 2000, pp. 63–138.
11. S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Proceedings of the IEEE Workshop on Signal Processing Advances in Wireless Communication* (*SPAWC '97*), Paris, France, 1997, pp. 101–104.
12. S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 2001.
13. P.-O. Amblard, M. Gaeta, and J.-L. Lacoume. Statistics for complex variables and signals—Part I: Variables. *Signal Processing*, 53:1–13, 1996.
14. P.-O. Amblard, M. Gaeta, and J.-L. Lacoume. Statistics for complex variables and signals—Part II: Signals. *Signal Processing*, 53:15–25, 1996.
15. S. An and Y. Hua. Blind signal separation and blind system identification of irreducible MIMO channels. In *Proceedings of the Sixth IEEE International Symposium on Signal Processing and Its Applications*, Kuala Lumpur, Malaysia, August 2001, pp. 276–279.
16. B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Dover, Mineola, NY, 1979. ISBN: 0-486-43938-0.

17. J. Anemüller, T. J. Sejnowski, and S. Makeig. Complex independent component analysis of frequency-domain electroencephalographic data. *Neural Networks*, 16(9):1311–1323, 2003.
18. S. Arberet, R. Gribonval, and F. Bimbot. A robust method to count and locate audio sources in a stereophonic linear instantaneous mixture. *Independent Component Analysis and Blind Signal Separation*, volume 3889/2006 of *Lecture Notes in Computer Science*, Nara, Japan, Springer, Berlin, Germany, 2006, Chapter 67.
19. S. Arberet, R. Gribonval, and F. Bimbot. A robust method to count and locate audio sources in a stereophonic linear anechoic mixture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP '07*), Honolulu, HI, 2007.
20. R. R. F. Attux, L. N. de Castro, F. J. Von Zuben, and J. M. T. Romano. A paradigm for blind IIR equalization using the constant modulus criterion and an artificial immune network. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing* (*NNSP*), Toulouse, France, 2003.
21. M. Austin. Decision feedback equalization for digital communications over dispersive channels. Technical Report 461, MIT Research Laboratory of Electronics, Cambridge, MA, 1967.
22. T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York, 1996.
23. T. Bäck, D. B. Fogel, and Z. Michalewicz, eds. *Evolutionary Computation 1: Basic Algorithms and Operators*. Taylor & Francis, New York, 2000.
24. L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Transactions on Information Theory*, 20(2):284–287, March 1974.
25. E. Bai and Z. Ding. Zero-forcing equalizability of FIR and IIR multichannel systems with and without perfect measurements. *IEEE Transactions on Communications*, 48(1):699–701, January 2000.
26. J. E. Baker. Reducing bias and inefficiency in the selection algorithm. in [icga2], 1987. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, Hillsdale, NJ, 1987, pp. 14–21.
27. Y. Bar-Ness, Y. Carlin, and M. L. Steinberger. Bootstrapping adaptive interference cancelers: Some practical limitations. In *Proceedings of the Globecom Conference*, Miami, FL, 1982, pp. 1251–1255.
28. J. R. Barry, D. G. Messerschmitt, and E. A. Lee. *Digital Communication*. Springer, New York, 2003.
29. W. Baumann, B.-U. Köhler, D. Kolossa, and R. Orglmeister. Real time separation of convolutive mixtures. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation* (*ICA 2001*), San Diego, CA, 2001, pp. 65–69.
30. C. A. Belfiore and J. H. Park, Jr. Decision feedback equalization. *Proceedings of the IEEE*, 67(8):1143–1156, August 1979.
31. A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.
32. M. Bellanger. *Adaptive Filters and Signal Analysis*. Dekker, New York, 1988.
33. M. Bellanger. Which constant modulus criterion is better for blind adaptive filtering: CM(1,2) or CM(2,2)? [equalizer example]. In *Proceedings of the IEEE*

International Conference on Acoustics, Speech, and Signal Processing (*ICASSP '05*), vol. 4, Philadelphia, PA, pp. iv/29–iv/32, 18–23 March 2005.

34. M. Bellanger. On the performance of two constant modulus algorithms in equalization with non-cm signals. In *Proceedings of the IEEE International Symposium on Circuits and Systems* (*ISCAS 2007*), New Orleans, LA, pp. 3475–3478, 27–30 May 2007.
35. S. Bellini. *Bussgang Techniques for Blind Deconvolution and Equalization*, Prentice Hall, Englewood Cliffs, NJ, 1994, pp. 8–52, Chapter 1.
36. S. Bellini and F. Rocca. *Blind Deconvolution: Polyspectra or Bussgang Techniques?* Elsevier Science Publishers, North-Holland, Amsterdam, the Netherlands, 1986, pp. 251–263.
37. A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, February 1997.
38. A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. Second-order blind separation of correlated sources. In *Proceedings of the International Conference Digital Signal Processing*, Cyprus, 1993, pp. 346–351.
39. S. Benedetto, E. Biglieri, and R. Daffara. Modelling and evaluation of nonlinear satelite links—Volterra series approach. *IEEE Transactions on Aerospace and Electron Systems*, 15:494–506, July 1979.
40. A. Benveniste and M. Goursat. Blind equalizers. *IEEE Transaction on Communications*, COM-32(8):871–883, August 1984.
41. A. Benveniste, M. Goursat, and G. Ruget. Robust identification of a non-minimum phase system: Blind adjustment of a linear equalizer in data communications. *IEEE Transactions on Automatic Control*, AC-25(3):385–399, June 1980.
42. H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Journal Natural Computing*, 1(1):3–52, 2002.
43. E. Biglieri, A. Gresho, R. D. Giltin, and T. L. Lim. Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission. *IEEE Journal of Selected Areas in Communication*, SAC-2:765–777, September 1984.
44. C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, U.K., 1995.
45. P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353–2362, 2001.
46. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Oxford, U.K., 1999.
47. C. B. Boyer and U. C. Mezbach. *A History of Mathematics*. Wiley, New York, 1991.
48. D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.
49. J.-M. Brossier. *Signal et Communication Numérique: égalisation et synchronisation* (Collection Traitement du Signal). HERMES, Paris, 1997.
50. G. Burel. Blind separation of sources: A non-linear neural algorithm. *Neural Networks*, 5:937–947, 1992.
51. F. M. Burnet. *Clonal Selection Theory of Acquired Immunity*. Cambridge University Press, Cambridge, U.K., 1959.
52. S. Cagnoni. *Evolutionary Image Analysis and Signal Processing*. Springer, Berlin, Germany, 2009.

53. V. D. Calhoun, G. D. Pearlson, and T. Adali. Independent component analysis applied to fMRI data: A generative model for validating results. *VLSI Signal Processing*, 37(2–3):281–291, 2004.
54. J.-F. Cardoso. Blind beamforming for non-Gaussian signals. *IEE Proceedings*, 140(6):362–370, December 1993.
55. J.-F. Cardoso. Infomax and maximum likelihood of blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, April 1997.
56. J.-F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, October 1998.
57. J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, January 1999.
58. J.-F. Cardoso and B. H. Laheld. Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017–3030, December 1996.
59. J.-F. Cardoso, M. Le Jeune, J. Delabrouille, M. Betoule, and G. Patanchon. Component separation with flexible models application to multichannel astrophysical observations. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):735–746, October 2008.
60. J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.
61. R. A. Casas, Zhi Ding, R. A. Kennedy, C. R. Johnson, Jr., and R. Malamut. Blind adaptation of decision feedback equalizers based on the constant modulus algorithm. In *1995 Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers, 1995*, Washington, DC, vol. 1, pp. 698–702, October–1 November 1995.
62. C. C. Cavalcante. On blind source separation: Proposals and analysis of multi-user processing strategies. PhD thesis, State University of Campinas (UNICAMP)—DECOM, Campinas, SP—Brazil, April 2004.
63. C. C. Cavalcante, F. R. P. Cavalcanti, and J. C. M. Mota. Adaptive blind multiuser separation criterion based on log-likelihood maximisation. *IEE Electronics Letters*, 38(20):1231–1233, September 2002.
64. C. C. Cavalcante and J. M. T. Romano. Multi-user pdf estimation based criteria for adaptive blind separation of discrete sources. *Signal Processing*, 85(5):1059–1072, May 2005.
65. F. R. P. Cavalcanti and J. M. T. Romano. Blind multiuser detection in space division multiple access systems. *Annales des Télécommunications*, 84(7–8):411–419, July–August 1999.
66. F. R. P. Cavalcanti and J. M. T. Romano. Using adaptive array for collision resolution in slotted ALOHA packet radio systems. *Journal of Communications and Networks*, 2(4):344–350, December 2000.
67. F. R. P. Cavalcanti, J. M. T. Romano, and A. L. Brandão. Least-squares CMA with decorrelation for fast blind multiuser signal separation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP '99*), vol. V, pp. 2527–2530, Phoenix, AZ, 15–19 May 1999.
68. S. Chen, B. Mulgrew, and P. M. Grant. A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on Neural Networks*, 4(4):570–579, July 1993.
69. W. Chen, T. Nemoto, T. Kobayashi, Y. Honda, T. Saito, and E. Kasuya. Foetal ECG extraction from maternal body surface measurement using independent component analysis. In *Proceedings of the 23rd Annual International Conference of*

the IEEE Engineering in Medicine and Biology Society, 2001, Istanbul, Turkey, vol. 2, pp. 1990–1993, 25–28 October 2001.

70. C.-Y. Chi, C.-H. Chen, and C.-Y. Chen. *Blind Equalization and System Identification: Batch Processing Algorithms, Performance and Applications*. Springer, London, U.K., 2006.
71. A. Cichocki and S.-I. Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley & Sons, Chichester, U.K., 2002.
72. A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Transaction on Circuits and Systems—I: Fundamental Theory and Applications*, 43:894–906, 1996.
73. M. Clerc and J. Kennedy. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.
74. P. Comon. Independent component analysis: A new concept? *Signal Processing*, 36(3):287–314, April 1994.
75. P. Comon. Contrasts for multichannel blind deconvolution. *IEEE Signal Processing Letters*, 3(7):209–211, July 1996.
76. P. Comon and C. Jutten, eds. *Separation de Sources, tome 1: Concepts de base et Analyse en Composantes Independantes*. Hermes, France, 2007.
77. J. X. Cong. Historical development of central limit theorem (CLT). Technical report, Rice University, Houston, TX, February 2003. Disponível on-line em http://www.stat.rice.edu/~blairc/seminar/Files/julieTalk.pdf
78. T. M. Cover and J. A. Thomas. *Elements of Information Theory*, 2nd edn. John Wiley & Sons, Hoboken, NJ, 2006.
79. S. A. Cruces-Alvarez, A. Cichocki, and S.-I. Amari. From blind signal extraction to blind instantaneous signal separation: Criteria, algorithms, and stability. *IEEE Transactions on Neural Networks*, 15(4):859–873, July 2004.
80. G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* (*MCSS*), 2(4):303–314, December 1989.
81. C. A. F. da Rocha, O. Macchi, and J. M. T. Romano. An adaptive nonlinear IIR filter for self-learning equalization. In *Proceedings of the IEEE International Telecommunications Symposium—ITS 94*, Rio de Janeiro, Brazil, 1994, pp. 184–190.
82. D. Dasgupta and Z. Michalewicz, eds. *Evolutionary Algorithms in Engineering Applications*. Springer, Berlin, Germany, 2001.
83. D. Dasgupta and F. Nino. *Immunological Computation: Theory and Applications*. CRC Press, Boca Raton, FL, 2008.
84. M. L. R. de Campos, S. Werner, and J. A. Apolinário, Jr. Constrained adaptive filters. *Adaptive Antenna Arrays: Trends and Applications*, Springer-Verlag, Berlin, Germany, 2004, pp. 46–62, Chapter 3.
85. L. N. de Castro. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. Chapman & Hall/CRC, Boca Raton, FL, 2006.
86. L. N. de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proceedings of IEEE Conference on Evolutionary Computation*, vol. 1, Washington, DC, 2002, pp. 699–704.
87. L N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, London, U.K., 2002.
88. L. N. de Castro and F. J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, June 2002.

89. N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45:59–83, 1995.
90. L. Deneire. Estimation Aveugle de Canal et Accès Multiple par Répartition Spatiale. PhD thesis, École National des Télécommunications – ENST Paris, Paris, France, December 1998.
91. L. Deneire and D. T. M. Slock. A deterministic Schur method for multichannel blind identification. In *Proceedings of Second IEEE Workshop on Signal Processing Advances in Wireless Communications* (*SPAWC '99*), Annapolis, MD, 9–12 May 1999.
92. L. Deneire and D. T. M. Slock. A Schur method for multiuser multichannel blind identification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP '99*), Phoenix, AZ, 15–19 March 1999, pp. 2905–2908.
93. Y. Deville. A unified stability analysis of the Hérault-Jutten source separation neural network. *Signal Processing*, 51(3):229–233, 1996.
94. K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks*. John Wiley & Sons, New York, 1996.
95. Z. Ding. On convergence analysis of fractionally spaced adaptive blind equalizers. *IEEE Transactions on Signal Processing*, 45(3):650–657, March 1997.
96. Z. Ding and T. Nguyen. Stationary points of a kurtosis maximization algorithm for blind signal separation and antenna beamforming. *IEEE Transactions on Signal Processing*, 48(6):1587–1596, June 2000.
97. Z. Ding, C. R. Johnson, Jr., and R. A. Kennedy. On the (non)existence of undesirable equilibria of godard blind equalizers. *IEEE Transactions on Signal Processing*, 40(10):2425–2432, October 1992.
98. Z. Ding, R. A. Kennedy, B. D. O. Anderson, and C. R. Johnson, Jr. Ill-convergence of Godard blind equalizers in data communication systems. *IEEE Transactions on Communications*, 39(9):1313–1327, September 1991.
99. Z. Ding and Ye (Geoffrey) Li. *Blind Equalization and Identification*. (Signal Processing and Communications Series). Marcel Dekker, New York, 2001.
100. Paulo Sérgio Ramirez Diniz. *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd edn. Springer, 2008.
101. D. Donoho. *On Minimum Entropy Deconvolution*. Academic Press, New York, 1981, pp. 565–608.
102. M. Dorigo. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
103. M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26(1):29–41, 1996.
104. L. Tomazeli Duarte. Design of Smart Chemical Sensor Arrays: An Approach Based on Source Separation Methods. PhD thesis, Institut Politechnique de Grenoble, Grenoble, France, 2009.
105. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, 2nd edn. Wiley-Interscience, New York, 2000.
106. J. Durbin. The fitting of time series models. *Review of the International Statistical Institute*, 28:233–244, 1960.
107. D. D. Falconer and G. J. Doschini. Theory of minimum mean-square-error QAM system employing decision feedback equalisation. *Bell System Technical Journal*, 53:1821–1849, November 1973.

108. R. Ferrari, C. M. Panazio, R. R. F. Attux, C. C. Cavalcante, L. N. de Castro, F. J. Von Zuben, and J. M. T. Romano. Unsupervised channel equalization using fuzzy prediction-error filters. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing (NNSP 03)*, Toulouse, France, 2003.
109. L. J. Fogel. *Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming*. Wiley-Interscience, New York, 1999.
110. G. B. Folland. *Real Analysis—Modern Techniques and Their Applications*. John Wiley & Sons, New York, 1999.
111. G. D. Forney, Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3): 268–278, 1973.
112. G. D. Forney, Jr. The Viterbi algorithm: A personal history. In *Proceedings of the Viterbi Conference on Advancing Technology through Communications Sciences*, Los Angeles, CA, USA, 8–9 March 2005.
113. G. J. Foschini. Equalizing without altering or detecting data. *Bell System Technical Journal*, 64:1885–1911, October 1985.
114. O. L. Frost III. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60:926–935, 1972.
115. F. R. Gantmacher. *The Theory of Matrices*, volume 1. Chelsea Publishing Company, New York, 1977.
116. W. A. Gardner. A new method of channel identification. *IEEE Transactions on Communications*, 39(6):813–817, June 1991.
117. W. A. Gardner. Exploitation of spectral redundancy in cyclostationary signals. *IEEE Signal Processing Letters*, 8:14–36, April 1991.
118. W. A. Gardner, ed. *Cyclostationary in Communications and Signal Processing*. IEEE Press, Piscataway, NJ, 1994.
119. C. F. Gauss. *Theoria motus corporum coelestium in sectionis conicis solem ambientum*. Hamburg (Transl.: Dover, New York: 1963), 1809.
120. P. G. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992–996, July 2005.
121. M. Ghosh. Blind decision feedback equalization for terrestrial television receivers. *Proceedings of the IEEE*, 86(10):2070–2081, October 1998.
122. G. J. Gibson and C. F. N. Cowan. On the decision regions of multilayer perceptrons. *Proceedings of the IEEE*, 78:1590–1599, 1990.
123. J. D. Gibson. *Principles of Digital and Analog Communications*, 2nd edn. Prentice-Hall, Upper Saddle River, NJ, 1993.
124. D. N. Godard. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Transactions on Communications*, COM-28(11):1867–1875, 1980.
125. R. Godfrey and F. Rocca. Zero memory non-linear deconvolution. *Geophysical Prospecting*, 29:189–228, April 1981.
126. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, Reading, MA, 1989.
127. D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Proceedings of the Foundations of Genetic Algorithms Workshop* (*FOGA*), San Mateo, CA, 1991.
128. G. H. Golub and C. F. Van Loan. *Matrix Computations* (Johns Hopkins Series in the Mathematical Sciences). Johns Hopkins University Press, Baltimore, MD, 1996.

129. J. Gomes and V. Barroso. A super-exponential algorithm for blind fractionally spaced equalization. *IEEE Signal Processing Letters*, 3(10):283–285, October 1996.
130. A. Gorokhov and P. Loubaton. Subspace-based techniques for blind separation of convolutive mixtures with temporally correlated sources. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(9):813–820, September 1997.
131. L. J. Griffiths and C. W. Jim. An alternative approach to linearly constrained adaptive beamforming. *IEEE Transactions on Antennas and Propagation*, AP-30:27–34, 1982.
132. M. Gu and L. Tong. Geometrical characterizations of constant modulus receivers. *IEEE Transactions on Signal Processing*, 47(10):2745–2756, 1999.
133. L. K. Hansen. ICA if fMRI based on a convolutive mixture model. In *Proceedings of the Ninth Annual Meeting of the Organization for Human Brain Mapping* (*HBM 2003*), New York, June 2003.
134. S. Haykin. *Modern Filters*. Macmillan, New York, 1989.
135. S. Haykin. *Communication Systems*, 3rd edn. John Wiley & Sons, New York, 1994.
136. S. Haykin. *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall, Upper Saddle River, NJ, 1998.
137. S. Haykin, ed. *Unsupervised Adaptive Filtering*, volume I: *Blind Source Separation* (Series on Adaptive and Learning Systems for Signal Processing, Communications and Control). John Wiley & Sons, New York, 2000.
138. S. Haykin, ed. *Unsupervised Adaptive Filtering*, volume II: *Blind Deconvolution* (Series on Adaptive and Learning Systems for Signal Processing, Communications and Control). John Wiley & Sons, New York, 2000.
139. S. Haykin. *Adaptive Filter Theory*, 4th edn. Prentice Hall, Englewood Cliffs, NJ, 2001.
140. S. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, Englewood Cliffs, NJ, 2008.
141. Z. He and A. Cichocki. K-EVD clustering and its applications to sparse component analysis. *Independent Component Analysis and Blind Signal Separation*. Springer, Berlin, Germany, 2006, pp. 90–97, Chapter 12.
142. D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.
143. J. Hérault and C. Jutten. *Réseaux Neuronaux et Traitement du Signal*. (Traitement du Signal). HERMES, Paris, France, 1994.
144. J. Hérault, C. Jutten, and B. Ans. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In *Proceedings of the 10th GRETSI Symposium on Signal and Image Processing*, Nice, France, May 1985, pp. 1017–1022.
145. J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
146. Y. Hua and J. K. Tugnait. Blind identifiability of FIR-MIMO systems with colored input using second-order statistics. *IEEE Signal Processing Letters*, 7(12):348–350, December 2000.
147. A. Hyvrinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.
148. A. Hyvrinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. (Adaptive and Learning Systems for Signal Processing, Communications, and Control). Wiley-Interscience, New York, 2001.

149. A. Hyvrinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.
150. A. Hyvrinen and P. Pajunen. Nonlinear independent components analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.
151. S. Ikeda and N. Murata. An approach to blind source separation of speech signals. In *Proceedings of the ICANN '98*, Skovde, Sweden, 1998, pp. 1855–1865.
152. M. Z. Ikram and D. R. Morgan. A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP 2002*), Orlando, FL, 2002, pp. 881–884.
153. Y. Inouye and R.-W. Liu. A system theoretic foundation for blind equalization of an FIR MIMO channel system. *IEEE Transactions on Circuits and Systems*, 49(4):425–436, April 2002.
154. N. K. Jerne. Towards a network theory of the immune system. *Annals of Immunology*, 125C(1–2):373–389, 1974.
155. C. R. Johnson, Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas. Blind equalization using the constant modulus criterion: A review. *Proceedings of the IEEE*, 86(10):1927–1950, October 1998.
156. C. Jutten and P. Comon, eds. *Separation de Sources, tome 2: Au dela de l'aveugle, et Applications*. Hermes, Paris, France, 2007.
157. C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. In *Proceedings of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation* (*ICA 2003*), Nara, Japan, 2003, pp. 245–256.
158. C. Jutten, S. Moussaoui, and F. Schmidt. How to apply ICA on actual data? Example of mars hyperspectral image analysis. In *2007 15th International Conference on Digital Signal Processing*, Wales, U.K., July 2007, pp. 3–12.
159. C. Jutten, L. Nguyen Thi, E. Dijkstra, E. Vittoz, and J. Caelen. Blind separation of sources: An algorithm for separation of convolutive mixtures. In *Proceedings of the International Signal Processing Workshop on Higher Order Statistics*, Chamrousse, France, 1991.
160. C. Jutten and A. Taleb. Source separation: From dusk till dawn. In *Proceedings of the International Symposium Independent Component Analysis and Blind Signal Separation*, Helsinki, Finland, 2000, pp. 15–26.
161. A. M. Kagan, Y. V. Linnik, and C. R. Rao. *Characterization Problems in Mathematics Statistics*. John Wiley & Sons, New York, 1973.
162. T. Kailath. A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, 20(2):146–181, March 1974.
163. T. Kailath. *Linear Systems*, 1st edn. Prentice Hall, Englewood Cliffs, NJ, 1980.
164. J. Karhunen, E. Oja, R. Vigario, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Transactions on Neural Networks*, 8:486–504, May 1997.
165. S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1 of Prentice Hall Signal Processing Series. Prentice-Hall, Upper Saddle River, NJ, 1993.
166. S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. 2 of Prentice Hall Signal Processing Series. Prentice-Hall, Upper Saddle River, NJ, 1998.

167. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, Washington, DC, 1995, pp. 1942–1948.
168. K. H. Knuth. A Bayesian approach to source separation. In *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation* (*ICA '99*), Aussois, France, January 1999, pp. 283–288.
169. A. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Julius Springer, Berlin, Germany, 1933.
170. A. N. Kolmogorov. Sur l'interpolation et l'extrapolation des suites stationnaires. *Comptes Rendus de l'Académie des Sciences*, 208:2043–2045, 1939.
171. S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura. Evaluation of frequency-domain blind signal separation using directivity pattern under reverberant conditions. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (*ICASSP '00*), Istanbul, Turkey, 2000, pp. 3140–3143.
172. H. J. Kushner. *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic Systems Theory*. MIT Press, Cambridge, MA, 1984.
173. J.-L. Lacoume, P.-O. Amblard, and P. Comon. *Statistiques d'Ordre Supérieur pour le Traitement du Signal*. (Traitement du Signal). Masson, Paris, 1997.
174. J.-L. Lacoume and P. Ruiz. Sources identification: A solution based on the cumulants. In *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling, 1988*, Minneapolis, MA, August 1988, pp. 199–203.
175. P.-S. M. Laplace. *Théorie Analytique des Probabilités*. Courcier, Paris, France, 1812.
176. J. Leary. Least-squares multi-user CMArray: A new algorithm for blind adaptive beamforming. In *Proceedings of IEEE 31st Asilomar Conference on Signals, Systems and Computer*, Pacific Grove, CA, November 1997, pp. 902–905.
177. D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:708–791, 1999.
178. A. M. Legendre. Méthode des moindres quarrés, pour trouver le milieu le plus probable entre les résultats de différentes observations. *Mémorial Institute France*, 1:149–154, 1810.
179. Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*, 2nd edn. Addison Wesley, Reading, MA, 1994.
180. N. Levinson. The Wiener RMS (root-mean-square) error criterion in filter design and prediction. *Journal of Mathematical Physics*, 25:261–278, 1947.
181. B. C. Levy. *Principles of Signal Detection and Parameter Estimation*. Springer, New York, 2008.
182. Y. Li, A. Cichocki, and S.-I. Amari. Sparse component analysis for blind source separation with less sensors than sources. In *Proceedings of the fourth International Symposium on Independent Component Analysis and Blind Signal Separation* (*ICA 2003*), Kyoto, Japan, 2003, pp. 89–94.
183. Y. Li and Z. Ding. Convergence analysis of finite length blind adaptive equalizers. *IEEE Transactions on Signal Processing*, 43(9):2120–2129, September 1995.
184. Y. Li and Z. Ding. Global convergence of fractionally spaced Godard (CMA) adaptive equalizers. *IEEE Transactions on Signal Processing*, 44(4):818–826, April 1996.
185. Y. Li and J. Wang. Sequential blind extraction of instantaneously mixed sources. *IEEE Transactions on Signal Processing*, 50(5):997–1006, May 2002.

186. Ye (Geoffrey) Li and K. J. Ray Liu. Adaptive blind source separation and equalization for multiple-input/multiple-output systems. *IEEE Transactions on Information Theory*, 44(7):2864–2876, November 1998.
187. J. C. Liberti Jr. and T. S. Rappaport. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. (Prentice Hall Comm. Engineering and Emerging Technologies Series). Prentice-Hall, Englewood Cliffs, NJ, 1999.
188. R. Linsker. Local synaptic learning rules suffice to maximise mutual information in a linear network. *Neural Computation*, 4:691–702, 1992.
189. L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 1987.
190. R. W. Lucky. Automatic equalization for digital communication. *Bell Systems Technical Journal*, 44:547–588, April 1965.
191. R. W. Lucky. Techniques for adaptive equalization of digital communication systems. *Bell Systems Technical Journal*, 45:255–286, February 1966.
192. D. G. Luenberger. *Optimization by Vector Space Methods*. (Series in Decision and Control). John Wiley & Sons, New York, 1969.
193. W.-K. Ma, P. C. Ching, and K. M. Wong. Maximum likelihood detection for multicarrier systems employing non-orthogonal pulse shapes. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000* (*ICASSP '00*), vol. 5, Istanbul, Turkey, May 2000, pp. 2489–2492.
194. O. Macchi. *Adaptive Processing: The LMS Approach with Applications in Transmission*. Wiley, New York, 1995.
195. O. Macchi, C. A. Faria, and J. M. T. Romano. Egalisation adaptative autodidacte par rétroprédiction et prédiction. In *Proceedings of the Colloque GRETSI*, Juan Les Pins, France, 1993.
196. S. W. Mahfoud. Niching methods for genetic algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 1995.
197. J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63:561–580, 1975.
198. S. Makino, H. Sawada, R. Mukai, and S. Araki. Blind source separation of convolutive mixtures of speech in frequency domain. *IEICE Transaction of Electronics, Communications and Computer Sciences*, E88-A(7):1640–1655, 2005.
199. S. Makino, H. Sawada, R. Mukai, and S. Araki. Blind source separation of convolutive mixtures of audio signals in frequency domain *Topics in Acoustic Echo and Noise Control*, Springer, Berlin, Germany, 2006, pp. 51–89, Chapter 3.
200. M. Babaiezadeh Malmiri. On blind source separation in convolutive and nonlinear mixtures. PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, September 2002.
201. V. J. Mathews and G. L. Sicuranza. *Polynomial Signal Processing*. (Wiley Series in Telecommunications and Signal Processing). John Wiley & Sons, New York, 2000.
202. K. Matsuoka. Minimal distortion principle for blind source separation. In *Proceedings of the 41st SICE Annual Conference* (*SICE'2002*), vol. 4, Osaka, Japan, 2002, pp. 2138–2143.
203. K. Matsuoka and S. Nakashima. Minimal distortion principle for blind source separation. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation* (*ICA 2001*), San Diego, CA, 2001, pp. 722–727.
204. K. Matsuoka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

205. J. E. Mazo. Analysis of decision-directed equalizer convergence. *Bell System Technical Journal*, 59(10):1857–1876, December 1980.
206. W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
207. K. A. Meraim, E. Moulines, and P. Loubaton. Prediction error method for second-order blind identification. *IEEE Transactions on Signal Processing*, 45(3):694–705, March 1997.
208. H. Mhlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.
209. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, New York, 1998.
210. N. Mitianoudis and M. E. Davies. Audio source separation of convolutive mixtures. *IEEE Transactions on Speech and Audio Processing*, 11(5):489–497, September 2003.
211. E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue. Subspace methods for blind equalization of FIR filters. *IEEE Transactions on Signal Processing*, 43(2):516–525, February 1995.
212. J.-P. Nadal and N. Parga. Nonlinear neurons in the low-noise limit: A factorial code maximises information transfer. *Network: Computation in Neural Systems*, 5(4):565–581, November 1994.
213. A. O. Neves, R. R. F. Attux, R. Suyama, M. D. Miranda, and J. M. T. Romano. Sobre critérios para equalizao não-supervisionada. *SBA: Controle & Automação Sociedade Brasileira de Automãtica*, 17(3):278–299, 2006.
214. C. L. Nikias and A. P. Petropulu. *Higher-Order Spectra Analysis*. Prentice Hall PTR, Englewood Cliffs, NJ, 1993.
215. D. Nuzillard and A. Bijaoui. Blind source separation and analysis of multispectral astronomical images. *Astronomy and Astrophysics Supplement Series*, 147(1):129–138, November 2000.
216. P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology* (*IJIST*), 15:18–33, 2005.
217. E. Oja. The nonlinear PCA learning rule and signal separation—Mathematical analysis. *Neurocomputing*, 17:22–45, 1997.
218. E. Oja and M. D. Plumbley. Blind separation of positive sources by globally convergent gradient search. *Neural Computation*, 16(9):1811–1825, September 2004.
219. A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*, 1st edn. (Prentice Hall Signal Processing Series). Prentice Hall, Englewood Cliffs, NJ, 1989.
220. P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
221. P. Pajunen. Nonlinear independent component analysis by self-organizing maps. In *Proceedings of the International Conference on Neural Information Processing* (*ICONIP '96*), Hong Kong, China, September 1996, pp. 1207–1210.
222. P. Pajunen and J. Karhunen. Least-squares methods for blind source separation based on nonlinear PCA. *International Journal of Neural Systems*, 8(5–6):601–612, October–December 1997.

223. C. B. Papadias. Methods for blind equalization and identification of linear channels. PhD thesis, École National Supérieure des Télécommunications (ENST), France, March 1995.
224. C. B. Papadias. *Blind Separation of Independent Sources Based on Multiuser Kurtosis Optimization Criteria*, vol. 2. John-Wiley & Sons, New York, 2000, pp. 147–179, Chapter 4.
225. C. B. Papadias. Globally convergent blind source separation based on a multiuser kurtosis maximization criterion. *IEEE Transactions on Signal Processing*, 48(12):3508–3519, December 2000.
226. C. B. Papadias. New unsupervised processing techniques for wideband multiple transmitter/multiple receiver systems. In *Conference Record of the 34th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, October, 29th–November, 1st 2000, pp. 759–763.
227. C. B. Papadias. Unsupervised receiver processing techniques for linear space-time equalization of wideband multiple input/multiple output channels. *IEEE Transactions on Signal Processing*, 52(2):472–482, February 2004.
228. C. B. Papadias and A. J. Paulraj. A constant modulus algorithm for multiuser signal separation in presence of delay spread using antenna array. *IEEE Signal Processing Letters*, 4(6):178–181, June 1997.
229. C. B. Papadias and D. T. M. Slock. Fractionally spaced equalization of linear polyphase channels and related blind techniques based on multichannel linear prediction. *IEEE Transactions on Signal Processing*, 47(3):641–654, March 1999.
230. A. Papoulis. *Probability, Random Variables and Stochastic Processes*, 3rd edn. (Electrical & Electronic Engineering Series). McGraw-Hill International, New York, 1991.
231. J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991.
232. S. K. Patra. *Development of Fuzzy System Based Channel Equalisers*. PhD thesis, University of Edinburgh, Edinburgh, U.K., August 1998.
233. S. K. Patra and B. Mulgrew. Fuzzy techniques for adaptive nonlinear equalization. *Signal Processing*, 80:985–1000, 2000.
234. P. Paulraj, C. B. Papadias, and A. Paulraj. Decision-feedback equalization and identification of linear channels using blind algorithms of the Bussgang type. In *Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers*, vol. II, Pacific Grove, CA, 1995, pp. 335–340.
235. M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra. A survey of convolutive blind source separation methods. In *Springer Handbook of Speech Processing*. Springer Press, November 2007.
236. P. Petrus. Novel adaptive array algorithms and their impact on cellular system capacity. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, March 1997.
237. D. Pham and P. Garat. Sparation aveugle de sources temporellement corrles. In *Proceedings of the GRETSI*, Juan Les Pins, France, 1993, pp. 317–320.
238. D. T. Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9):18371848, 2001.
239. G. Picchi and G. Prati. Blind equalization and carrier recovery using a 'Stop-and-Go' decision-directed algorithm. *IEEE Transactions on Communications*, COM-35(9):877–887, September 1987.

240. R. L. Plackett. A historical note on the method of least squares. *Biometrika*, 36(3–4):458–460, 1949.
241. M. Plumbley. Conditions for nonnegative independent component analysis. *IEEE Signal Processing Letters*, 9(6):177–180, June 2002.
242. M. D. Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543, 2003.
243. M. D. Plumbley. Algorithms for nonnegative independent component analysis. *Neural Networks, IEEE Transactions on*, 14(3):534–543, May 2003.
244. S. Prasad and N. C. Menicucci. Fisher information with respect to cumulants. *IEEE Transactions on Information Theory*, 50(4):638–642, April 2004.
245. J. G. Proakis. *Digital Communications*, 3rd edn. (Electrical Engineering Series). McGraw-Hill, New York, 1995.
246. S. U. H. Qureshi. Adaptive equalization. *Proceedings of the IEEE*, 73(9):1349–1387, September 1985.
247. I. Rechenberg. Cybernetic solution path of an experimental problem. In *Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.* Ministry of Aviation, Royal Aircraft Establishment, Farnborough Hants, U.K., August 1965.
248. P. A. Regalia. On the equivalence between the Godard and Shalvi-Weinstein schemes of blind equalization. *Signal Processing*, 73(1–2):185–190, 1999.
249. P. A. Regalia. *Adaptive IIR Filtering in Signal Processing and Control*. Marcel Dekker, New York, 1995.
250. L. S. Resende, J. M. T. Romano, and M. G. Bellanger. Fast linear phase adaptive filtering using constrained least squares algorithm. In *Proceedings of the EUSIPCO'92*, Brussels, Belgium, 1992.
251. L. S. Resende, J. M. T. Romano, and M. G. Bellanger. A fast least-squares algorithm for linearly constrained adaptive filtering. *IEEE Transactions on Signal Processing*, 44(5):1168–1174, May 1996.
252. H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.
253. E. A. Robinson. Predictive decomposition of time series with applications to seismic exploration. PhD thesis, Department of Geology and Geophysics, MIT, Cambridge, MA, 1954.
254. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psycholical Review*, 65:386–408, 1958.
255. G. Rudolph. On correlated mutations in evolution strategies. In *Parallel Problem Solving from Nature—Proceedings of the Second Conference PPSN*, Amsterdam, the Netherlands, 1992, pp. 105–114. Elsevier.
256. D. R. Rumelhart and J. L McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1: *Foundations*. MIT Press, Cambridge, MA, 1986.
257. D. R. Rumelhart and J. L McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 2: *Psychological and Biological Models*. MIT Press, Cambridge, MA, 1986.
258. S. Sanei, S. Constantinides, C. C. Took, and B. Song. Incorporating frequency warping into sparse component analysis. In *Proceedings of the 13th European Signal Processing Conference* (*EUSIPCO 2005*), Turkey, 2005.
259. B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, September 1998.
260. Y. Sato. A method of self-recovering equalization for multi-level amplitude modulation. *IEEE Transactions on Communications*, 23:679–682, June 1975.

261. H. Sawada, R. Mukai, S. Araki, and S. Makino. A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE Transaction on Speech and Audio Processing*, 12(5):530–538, 2004.
262. A. H. Sayed. *Fundamentals of Adaptive Filtering*. Wiley-Interscience, Hoboken, NJ, 2003.
263. A. H. Sayed and T. Kailath. A state-space approach to adaptive RLS filtering. *IEEE Signal Processing Magazine*, 11(3):18–60, July 1994.
264. P. Schniter and C. R. Johnson, Jr. Bounds or the MSE performance of constant modulus estimators. *IEEE Transactions on Information Theory*, 46(7):2544–2560, November 2000.
265. T. Schonhoff and A. Giordano. *Detection and Estimation Theory*. Prentice Hall, Englewood Cliffs, NJ, 2006.
266. H.-P. Schwefel. Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strmungstechnik. Master's thesis, Technical University of Berlin, Berlin, Germany, March 1965.
267. H.-P. Schwefel. Evolutionsstrategie und numerische optimierung. PhD thesis, Technical University of Berlin, Berlin, Germany, 1975.
268. C. Serviere and D.-T. Pham. A novel method for permutation correction in frequency-domain in blind separation of speech mixtures, *Independent Component Analysis and Blind Signal Separation*, Springer, Berlin, Germany, 2004, pp. 807–815, Chapter 101.
269. O. Shalvi and E. Weinstein. New criteria for blind deconvolution of non-minimum phase systems (channels). *IEEE Transactions on Information Theory*, 36(2):312–321, March 1990.
270. O. Shalvi and E. Weinstein. Super-exponential methods for blind deconvolution. *IEEE Transactions on Information Theory*, 39(2):504–519, March 1993.
271. O. Shalvi and E. Weinstein. *Universal Methods for Blind Deconvolution*. Prentice-Hall, Englewood Cliffs, NJ, 1994, pp. 121–180.
272. C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, July 1948.
273. Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, New York, 1998, pp. 69–73.
274. J. J. Shynk. Adaptive IIR filtering. *IEEE ASSP Magazine*, April:4–21, 1989.
275. J. J. Shynk and R. P. Gooch. Convergence properties of the multistage CMA adaptive beamformer. In *Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, November 1993, pp. 622–626.
276. P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, 22:21–34, 1998.
277. J. Sol-Casals, M. Babaie-Zadeh, C. Jutten, and D.-T. Pham. Improving algorithm speed in PNL mixture separation and Wiener system inversion. In *Proceedings of the International Symposium on Independent Component Analysis and Blind Signal Separation* (*ICA 2003*), Nara, Japan, 2003.
278. E. Sorouchyari. Blind separation of sources, part III: Stability analysis. *Signal Processing*, 24:21–29, 1991.
279. M. H. Stone. The generalized Weierstrass approximation theorem. *Mathematics Magazine*, 21(4):167–184, 1948.

280. A. Taleb and C. Jutten. Nonlinear source separation: The post-nonlinear mixtures. In *Proceedings of the ESANN*, Bruges, Bélgica, 1997.
281. A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820, October 1999.
282. S. Theodoridis, C. M. S. See, and C. F. N. Cowan. Nonlinear channel equalization using clustering techniques. In *Proceedings of the IEEE International Conference on Communications* (*ICC '92*), vol. 3, Chicago, IL, June 1992, pp. 1277–1279.
283. C. W. Therrien. *Discrete Random Signals and Statistical Signal Processing* (Prentice-Hall Signal Processing Series). Prentice-Hall International, Englewood Cliffs, NJ, 1992.
284. E. J. Thomas. Some considerations on the application of the volterra representation of nonlinear networks to adaptive echo cancellers. *Bell System Technical Journal*, 50:2799–2805, 1971.
285. J. Thomas, Y. Deville, and S. Hosseini. Time-domain fast fixed-point algorithms for convolutive ICA. *IEEE Signal Processing Letters*, 13(4):228–231, 2006.
286. L. Tong, Ruey wen Liu, and V. C. Soon. Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems*, 38(5):499–509, May 1991.
287. L. Tong, G. Xu, and T. Kailath. A new approach to blind identification and equalization of multipath channels. In *Proceedings of the 25th Asilomar Conference on Signals, Systems and Computers*, vol. 2, Piscataway, NJ, November 1991, pp. 856–860.
288. L. Tong, G. Xu, and T. Kailath. Fast blind equalization via antenna arrays. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing,* (*ICASSP '93*), Minneapolis, MN, April 1993, pp. 272–275.
289. L. Tong, G. Xu, and T. Kailath. Blind identification and equalization based on second-order statistics: A time domain approach. *IEEE Transactions on Information Theory*, 40(2):340–349, March 1994.
290. K. Torkkola. Blind separation of convolved sources based on information maximization. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, Kyoto, Japan, 1996, pp. 1–10.
291. K. Torkkola. Blind separation of delayed and convolved sources *Unsupervised Adaptive Filtering* volume I: *Blind Source Separation*. John Wiley & Sons, New York, 2000, pp. 321–375, Chapter 8.
292. J. R. Treichler and B. G. Agee. A new approach to multipath correction of constant modulus signals. *IEEE Transactions on Adaptive Speech and Signal Processing*, ASSP-31(4):349–472, April 1983.
293. J. K. Tugnait. Comments on "New criteria for blind deconvolution of nonminimum phase systems (channels)". *IEEE Transactions of Information Theory*, 38(1):210–213, January 1992.
294. J. K. Tugnait. On blind separation of convolutive mixtures of independent linear signals. In *Proceedings of the 8th Workshop on Statistical Signal and Array Processing*, Corfu, Greece, 24–26 June 1996, pp. 312–315.
295. J. K. Tugnait. Adaptive blind separation of convolutive mixtures of independent linear signals. *Signal Processing*, 73:139–152, 1999.
296. J. K. Tugnait. On linear predictors for MIMO channels and related blind identification and equalization. *IEEE Signal Processing Letters*, SPL-5:289–291, November 1998.

297. P. P. Vaidyanathan. *The Theory of Linear Prediction*. Morgan & Claypool Publishers, San Rafael, CA, 2008.
298. H. Valpola. Nonlinear independent component analysis using ensemble learning: Theory. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation* (*ICA 2000*), Finlandia, 2000, pp. 251–256.
299. H. Valpola and P. Pajunen. Fast algorithms for Bayesian independent component analysis. In *Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation*, Helsinki, Finland 2000, pp. 233–237.
300. A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.
301. E. Walach and B. Widrow. The least mean fourth (LMF) adaptive algorithm and its family. *IEEE Transactions on Information Theory*, IT-30(2):275–283, March 1984.
302. P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Harvard University, Cambridge, MA, 1974.
303. B. Widrow and M. E. Hoff, Jr. Adaptive switching circuits. *IRE WESCON Convention Record*, 4:96–104, 1960.
304. B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1985.
305. N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, New York, 1949.
306. N. Wiener and W. Hopf. Uber eine klasse singularer integraleichunger. *Sitz. Akademie Wissenschaft*, Berlin, Germany, 1931, pp. 696–706.
307. R. A. Wiggins. Minimum entropy deconvolution. *Geoexploration*, 16:21–35, 1978.
308. D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.
309. S. An, Y. Hua, and Y. Xiang. Blind identification and equalization of FIR MIMO channels by BIDS. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, pp. 2157–2160.
310. H. H. Yang, S.-I. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in non-linear mixture. *Signal Processing*, 64:291–300, 1998.
311. D. Yellin and E. Weinstein. Criteria for multichannel signal separation. *IEEE Transactions on Signal Processing*, 42(8):2158–2168, August 1994.
312. H. H. Zeng, L. Tong, and C. R. Johnson, Jr. Relationships between the constant modulus and Wiener receivers. *IEEE Transactions on Information Theory*, 44(4):1523–1538, July 1998.
313. H. H. Zeng, L. Tong, and C. R. Johnson, Jr. An analysis of constant modulus receivers. *IEEE Transactions on Signal Processing*, 47(11):2990–2999, November 1999.
314. Y. Zhanga, X. Shia, and C. H. Chenb. A Gaussian mixture model for underdetermined independent component analysis. *Signal Processing*, 86(7):1538–1549, July 2006.

315. A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Miller. Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation. In *Proceedings of the International Symposium on Independent Component Analysis and Blind Signal Separation* (*ICA 2003*), Nara, Japan, 2003.

316. A. Ziehe and K. Muller. TDSEP—An efficient algorithm for blind separation using time structure. In *Proceedings of the International Conference on Artificial Neural Networks* (*ICANN '98*), Sucia, Skovde, Sweden, 1998, pp. 675–680.