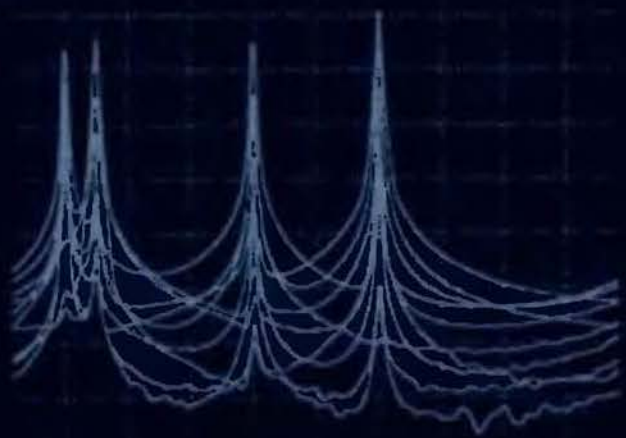
MONSON H. HAYES

STATISTICAL DIGITAL SIGNAL PROCESSING AND MODELING



STATISTICAL DIGITAL SIGNAL PROCESSING AND MODELING

MONSON H. HAYES
Georgia Institute of Technology

JOHN WILEY & SONS, INC.
New York, Chichester, Brisbane, Toronto, Singapore

Acquisitions Editor Steven Elliot
Marketing Manager Debra Riegert
Senior Production Editor Tony VenGraitis
Designer Laura Ierardi
Manufacturing Manager Mark Cirillo
Production Service Ingrao Associates

This book was set in Times Roman by
Techsetters, Inc. and printed and bound
by Hamilton Printing. The cover was
printed by New England Book Components.

Recognizing the importance of preserving
what has been written, it is a policy of
John Wiley & Sons, Inc. to have books
of enduring value published in the United States
printed on acid-free paper, and we exert our best
efforts to that end.

The paper in this book was manufactured by a
mill whose forest management programs include
sustained yield harvesting of its timberlands.
Sustained yield harvesting principles ensure that
the number of trees cut each year does not exceed
the amount of new growth.

ISBN 0-471 59431-8
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

This book is dedicated to my beautiful wife, Sandy, for all of her love, patience, and support, and for all of the sacrifices that she has made so that this book could become a reality.

It is also dedicated to Michael, Kimberly, and Nicole, for the joy that they have brought into my life, and to my parents for all of their love and support of me in all of my endeavors.

CONTENTS

PREFACE

This book is the culmination of a project that began as a set of notes for a graduate level course that is offered at Georgia Tech. In writing this book, there have been many challenges. One of these was the selection of an appropriate title for the book. Although the title that was selected is *Statistical Signal Processing and Modeling*, any one of a number of other titles could equally well have been chosen. For example, if the title of a book is to capture its central theme, then the title perhaps could have been *Least Squares Theory in Signal Processing*. If, on the other hand, the title should reflect the role of the book within the context of a course curriculum, then the title should have been *A Second Course in Discrete-Time Signal Processing*. Whatever the title, the goal of this book remains the same: to provide a comprehensive treatment of signal processing algorithms for modeling discrete-time signals, designing optimum digital filters, estimating the power spectrum of a random process, and designing and implementing adaptive filters.

In looking through the Table of Contents, the reader may wonder what the reasons were in choosing the collection of topics in this book. There are two. The first is that each topic that has been selected is not only important, in its own right, but is also important in a wide variety of applications such as speech and audio signal processing, image processing, array processing, and digital communications. The second is that, as the reader will soon discover, there is a remarkable relationship that exists between these topics that tie together a number of seemingly unrelated problems and applications. For example, in Chapter 4 we consider the problem of modeling a signal as the unit sample response of an all-pole filter. Then, in Chapter 7, we find that all-pole signal modeling is equivalent to the problem of designing an optimum (Wiener) filter for linear prediction. Since both problems require finding the solution to a set of Toeplitz linear equations, the Levinson recursion that is derived in Chapter 5 may be used to solve both problems, and the properties that are shown to apply to one problem may be applied to the other. Later, in Chapter 8, we find that an all-pole model performs a maximum entropy extrapolation of a partial autocorrelation sequence and leads, therefore, to the maximum entropy method of spectrum estimation.

This book possesses some unique features that set it apart from other treatments of statistical signal processing and modeling. First, each chapter contains numerous examples that illustrate the algorithms and techniques presented in the text. These examples play an important role in the learning process. However, of equal or greater importance is the working of new problems by the student. Therefore, at the end of each chapter, the reader will find numerous problems that range in difficulty from relatively simple exercises to more involved problems. Since many of the problems introduce extensions, generalizations, or applications of the material in each chapter, the reader is encouraged to read through the problems that are not worked. In addition to working problems, another important step in the learning process is to experiment with signal processing algorithms on a computer using either real or synthetic data. Therefore, throughout the book the reader will find MATLAB programs that have been written for most of the algorithms and techniques that are presented in the book and, at the end of each chapter, the reader will find a variety of computer exercises that use these programs. In these exercises, the student will study the performance of signal processing algorithms, look at ways to generalize the algorithms or make them more efficient, and write new programs.

Another feature that is somewhat unique to this book concerns the treatment of complex signals. Since a choice had to be made about how to deal with complex signals, the easy thing to have done would have been to consider only real-valued signals, leaving the generalization to complex signals to the exercises. Another possibility would have been to derive all results assuming that signals are real-valued, and to state how the results would change for complex-valued signals. Instead, however, the approach that was taken was to assume, from the beginning, that all signals are complex-valued. This not only saves time and space, and avoids having to jump back and forth between real-valued and complex-valued signals, but it also allows the reader or instructor to easily treat real-valued signals as a special case.

This book consists of nine chapters and an appendix and is suitable for a one-quarter or one-semester course at the Senior or Graduate level. It is intended to be used in a course that follows an introductory course in discrete-time signal processing. Although the prerequisites for this book are modest, each is important. First, it is assumed that the reader is familiar with the fundamentals of discrete-time signal processing including difference equations, discrete convolution and linear filtering, the discrete-time Fourier transform, and the z-transform. This material is reviewed in Chapter 2 and is covered in most standard textbooks on discrete-time signal processing [3,6]. The second prerequisite is a familiarity with linear algebra. Although an extensive background in linear algebra is not required, it will be necessary for the reader to be able to perform a variety of matrix operations such as matrix multiplication, finding the inverse of a matrix, and evaluating its determinant. It will also be necessary for the reader to be able to find the solution to a set of linear equations and to be familiar with eigenvalues and eigenvectors. This is standard material that may be found in any one of a number of excellent textbooks on linear algebra [2,5], and is reviewed in Chapter 2. Also in Chapter 2 is a section of particular importance that may not typically be in a student's background. This is the material covered in Section 2.3.10, which is concerned with optimization theory. The specific problem of interest is the minimization of a quadratic function of one or more *complex* variables. Although the minimization of a quadratic function of one or more *real* variables is fairly straightforward and only requires setting the derivative of the function with respect to each variable equal to zero, there are some subtleties that arise when the variables are complex. For example, although we know that the quadratic function $f(z) = |z|^2$ has a unique minimum that occurs at $z = 0$, it is not clear how to formally demonstrate this since this function is not differentiable. The last prerequisite is a course in basic probability theory. Specifically, it is necessary for the reader to be able to compute ensemble averages such as the mean and variance, to be familiar with jointly distributed random variables, and to know the meaning of terms such as statistical independence, orthogonality, and correlation. This material is reviewed in Chapter 3 and may be found in many textbooks [1,4].

This book is structured to allow a fair amount of flexibility in the order in which the topics may be taught. The first three chapters stand by themselves and, depending upon the background of the student, may either be treated as optional reading, reviewed quickly, or used as a reference. Chapter 2, for example, reviews the basic principles of discrete-time signal processing and introduces the principles and techniques from linear algebra that will be used in the book. Chapter 3, on the other hand, reviews basic principles of probability theory and provides an introduction to discrete-time random processes. Since it is not assumed that the reader has had a course in random processes, this chapter develops all of the necessary tools and techniques that are necessary for our treatment of stochastic signal modeling, optimum linear filters, spectrum estimation, and adaptive filtering.

In Chapter 4, we begin our treatment of statistical signal processing and modeling with the development of a number of techniques for modeling a signal as the output of a linear shift-invariant filter. Most of this chapter is concerned with models for deterministic signals, which include the methods of Padé, Prony, and Shanks, along with the autocorrelation and covariance methods. Due to its similarity with the problem of signal modeling, we also look at the design of a least-squares inverse filter. Finally, in Section 4.7, we look at models for discrete-time random processes, and briefly explore the important application of spectrum estimation. Depending on the interest of the reader or the structure of a course, this section may be postponed or omitted without any loss in continuity.

The initial motivation for the material in Chapter 5 is to derive efficient algorithms for the solution to a set of Toeplitz linear equations. However, this chapter accomplishes much more than this. Beginning with the derivation of a special case of the Levinson recursion known as the Levinson-Durbin recursion, this chapter then proceeds to establish a number of very important properties and results that may be derived from this recursion. These include the introduction of the lattice filter structure, the proof of the stability of the all-pole model that is formed using the autocorrelation method, the derivation of the Schur-Cohn stability test for digital filters, a discussion of the autocorrelation extension problem, the Cholesky decomposition of a Toeplitz matrix, and a procedure for recursively computing the inverse of a Toeplitz matrix that may be used to derive the general Levinson recursion and establish an interesting relationship that exists between the spectrum estimates produced using the minimum variance method and the maximum entropy method. The chapter concludes with the derivation of the Levinson and the split Levinson recursions.

The focus of Chapter 6 is on lattice filters and on how they may be used for signal modeling. The chapter begins with the derivation of the FIR lattice filter, and then proceeds to develop other lattice filter structures, which include the all-pole and allpass lattice filters, lattice filters that have both poles and zeros, and the split lattice filter. Then, we look at lattice methods for all-pole signal modeling. These methods include the forward covariance method, the backward covariance method, and the Burg algorithm. Finally, the chapter concludes by looking at how lattice filters may be used in modeling discrete-time random processes.

In Chapter 7 we turn our attention to the design of optimum linear filters for estimating one process from another. The chapter begins with the design of an FIR Wiener filter and explores how this filter may be used in such problems as smoothing, linear prediction, and noise cancellation. We then look at the problem of designing IIR Wiener filters. Although a noncausal Wiener filter is easy to design, when a causality constraint in imposed on the filter structure, the design requires a spectral factorization. One of the limitations of the Wiener filter, however, is the underlying assumption that the processes that are being filtered are wide-sense stationary. As a result, the Wiener filter is linear and shift-invariant. The chapter concludes with an introduction to the discrete Kalman filter which, unlike the Wiener filter, may be used for stationary as well as nonstationary processes.

Chapter 8 considers the problem of estimating the power spectrum of a discrete-time random process. Beginning with the classical approaches to spectrum estimation, which involve taking the discrete-time Fourier transform of an estimated autocorrelation sequence, we will examine the performance of these methods and will find that they are limited in resolution when the data records are short. Therefore, we then look at some modern approaches to spectrum estimation, which include the minimum variance method, the maximum entropy method, and parametric methods that are based on developing a model for a

random process and using this model to estimate the power spectrum. Finally, we look at eigenvector methods for estimating the frequencies of a harmonic process. These methods include the Pisarenko harmonic decomposition, MUSIC, the eigenvector method, the minimum norm algorithm, and methods that are based on a principle components analysis of the autocorrelation matrix.

Finally, Chapter 9 provides an introduction to the design, implementation, and analysis of adaptive filters. The focus of this chapter is on the LMS algorithm and its variations, and the recursive least squares algorithm. There are many applications in which adaptive filters have played an important role such as linear prediction, echo cancellation, channel equalization, interference cancellation, adaptive notch filtering, adaptive control, system identification, and array processing. Therefore, included in this chapter are examples of some of these applications.

Following Chapter 9, the reader will find an Appendix that contains some documentation on how to use the MATLAB m-files that are found in the book. As noted in the appendix, these m-files are available by anonymous ftp from

```
ftp.eedsp.gatech.edu/pub/users/mhayes/stat_dsp
```

and may be accessed from the Web server for this book at

```
http://www.ece.gatech.edu/users/mhayes/stat_dsp
```

The reader may also wish to browse this Web site for additional information such as new problems and m-files, notes and errata, and reader comments and feedback.

A typical one quarter course for students who have not been exposed to discrete-time random processes might cover, in depth, Sections 3.3–3.7 of Chapter 3; Chapter 4; Sections 5.1–5.3 of Chapter 5; Sections 6.1, 6.2, and 6.4–6.7 of Chapter 6; Chapter 7; and Chapter 8. For students that have had a formal course in random processes, Chapter 3 may be quickly reviewed or omitted and replaced with Sections 9.1 and 9.2 of Chapter 9. Alternatively, the instructor may wish to introduce adaptive approaches to signal modeling at the end of Chapter 4, introduce the adaptive Wiener filter after Chapter 7, and discuss techniques for adaptive spectrum estimation in Chapter 8. For a semester course, on the other hand, Chapter 9 could be covered in its entirety. Starred (*) sections contain material that is a bit more advanced, and these sections may be easily omitted without any loss of continuity.

In teaching from this book, I have found it best to review linear algebra only as it is needed, and to begin the course by spending a couple of lectures on some of the more advanced material from Sections 3.3–3.7 of Chapter 3, such as ergodicity and the mean ergodic theorems, the relationship between the power spectrum and the maximum and minimum eigenvalues of an autocorrelation matrix, spectral factorization, and the Yule-Walker equations. I have also found it best to restrict attention to the case of real-valued signals, since treating the complex case requires concepts that may be less familiar to the student. This typically only requires replacing derivatives that are taken with respect to z^* with derivatives that are taken with respect to z.

Few textbooks are ever written in isolation, and this book is no exception. It has been shaped by many discussions and interactions over the years with many people. I have been fortunate in being able to benefit from the collective experience of my colleagues at Georgia Tech: Tom Barnwell, Mark Clements, Jim McClellan, Vijay Madisetti, Francois Malassenet, Petros Maragos, Russ Mersereau, Ron Schafer, Mark Smith, and Doug Williams. Among

these people, I would like to give special thanks to Russ Mersereau, my friend and colleague, who has followed the development of this book from its beginning, and who will probably never believe that this project is finally done. Many of the ways in which the topics in this book have been developed and presented are attributed to Jim McClellan, who first exposed me to this material during my graduate work at M.I.T. His way of looking at the problem of signal modeling has had a significant influence in much of the material in this book. I would also like to acknowledge Jae Lim and Alan Oppenheim, whose leadership and guidance played a significant and important role in the development of my professional career. Other people I have been fortunate to be able to interact with over the years include my doctoral students Mitch Wilkes, Erlandur Karlsson, Wooshik Kim, David Mazel, Greg Vines, Ayhan Sakarya, Armin Kittel, Sam Liu, Baldine-Brunel Paul, Halük Aydinoglu, Antai Peng, and Qin Jiang. In some way, each of these people has influenced the final form of this book. I would also like to acknowledge all of the feedback and comments given to me by all of the students who have studied this material at Georgia Tech using early drafts of this book. Special thanks goes to Ali Adibi, Abdelnaser Adas, David Anderson, Mohamed-Slim Alouini, Osama Al-Sheikh, Rahmi Hezar, Steven Kogan, and Jeff Schodorf for the extra effort that they took in proof-reading and suggesting ways to improve the presentation of the material.

I would like to express thanks and appreciation to my dear friends Dave and Annie, Pete and Linda, and Olivier and Isabelle for all of the wonderful times that we have shared together while taking a break from our work to relax and enjoy each other's company.

Finally, we thank the following reviewers for their suggestions and encouragement throughout the development of this text: Tom Alexander, North Carolina State University; Jan P. Allenbach, Purdue University; Andreas Antoniou, University of Victoria; Ernest G. Baxa, Clemson University; Takis Kasparis, University of Central Florida; JoAnn B. Koskol, Widener University; and Charles W. Therrien, Naval Postgraduate School.

References

1. H. L. Larson and B. O. Shubert, *Probabilistic Models in Engineering Sciences: Vol. I, Random Variables and Stochastic Processes*, John Wiley & Sons, New York, 1979.
2. B. Nobel and J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
3. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
4. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd Ed, McGraw-Hill, New York, 1991.
5. G. Strang, *Linear Algebra and its Applications*, Academic Press, New York, 1980.
6. R. D. Strum and D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley, MA, 1988.

1 INTRODUCTION

Our ability to communicate is key to our society. Communication involves the exchange of information, and this exchange may be over short distances, as when two people engage in a face-to-face conversation, or it may occur over large distances through a telephone line or satellite link. The entity that carries this information from one point to another is a *signal*. A signal may assume a variety of different forms and may carry many different types of information. For example, an acoustic wave generated by the vocal tract of a human carries speech, whereas electromagnetic waves carry audio and video information from a radio or television transmitter to a receiver. A signal may be a function of a single continuous variable, such as time, it may be a function of two or more continuous variables, such as (x, y, t) where x and y are spatial coordinates, or it may be a function of one or more discrete variables.

Signal processing is concerned with the representation, manipulation, and transformation of signals and the information that they carry. For example, we may wish to enhance a signal by reducing the noise or some other interference, or we may wish to process the signal to extract some information such as the words contained in a speech signal, the identity of a person in a photograph, or the classification of a target in a radar signal.

Digital signal processing (*DSP*) is concerned with the processing of information that is represented in digital form. Although DSP, as we know it today, began to bloom in the 1960s, some of the important and powerful processing techniques that are in use today may be traced back to numerical algorithms that were proposed and studied centuries ago. In fact, one of the key figures whose work plays an important role in laying the groundwork for much of the material in this book, and whose name is attached to a number of terms and techniques, is the mathematician Karl Friedrick Gauss. Although Gauss is usually associated with terms such as the Gaussian density function, Gaussian elimination, and the Gauss-Seidel method, he is perhaps best known for his work in least squares estimation.[1] It should be interesting for the reader to keep a note of some of the other personalities appearing in this book and the dates in which they made their contributions. We will find, for example, that Prony's work on modeling a signal as a sum of exponentials was published in 1795, the work of Padé on signal matching was published in 1892, and Schuster's work

[1]Gauss is less well known for his work on fast algorithms for computing the Fourier series coefficients of a sampled signal. Specifically, he has been attributed recently with the derivation of the radix-2 decimation-in-time Fast Fourier Transform (FFT) algorithm [6].

on the periodogram appeared in 1898. Two of the more recent pioneers whose names we will encounter and that have become well known in the engineering community are those of N. Wiener (1949) and R. E. Kalman (1960).

Since the early 1970s when the first DSP chips were introduced, the field of Digital Signal Processing (DSP) has evolved dramatically. With a tremendously rapid increase in the speed of DSP processors along with a corresponding increase in their sophistication and computational power, digital signal processing has become an integral part of many products and applications. Coupled with the development of efficient algorithms for performing complex signal processing tasks, digital signal processing is radically changing the way that signal processing is done and is becoming a commonplace term.

The purpose of this book is to introduce and explore the relationships between four very important signal processing problems: signal modeling, optimum filtering, spectrum estimation, and adaptive filtering. Although these problems may initially seem to be unrelated, as we progress through this book we will find a number of different problems reappearing in different forms and we will often be using solutions to previous problems to solve new ones. For example, we will find that one way to derive a high-resolution spectrum estimate is to use the tools and techniques derived for modeling a signal and then use the model to estimate the spectrum.

The prerequisites necessary for this book are modest, consisting of an introduction to the basic principles of digital signal processing, an introduction to basic probability theory, and a general familiarity with linear algebra. For purposes of review as well as to introduce the notation that will be used throughout this book, Chapter 2 provides an overview of the fundamentals of DSP and discusses the concepts from linear algebra that will be useful in our representation and manipulation of signals. The following paragraphs describe, in general terms, the topics that are covered in this book and overview the importance of these topics in applications.

DISCRETE-TIME RANDOM PROCESSES

An introductory course in digital signal processing is concerned with the analysis and design of systems for processing deterministic discrete-time signals. A deterministic signal may be defined as one that can be described by a mathematical expression or that can be reproduced repeatedly. Simple deterministic signals include the unit sample, a complex exponential, and the response of a digital filter to a given input. In almost any application, however, it becomes necessary to consider a more general type of signal known as a random process. Unlike a deterministic signal, a random process is an ensemble or collection of signals that is defined in terms of the statistical properties that the ensemble possesses. Although a sinusoid is a deterministic signal, it may also be used as the basis for the random process consisting of the ensemble of all possible sinusoids having a given amplitude and frequency. The randomness or uncertainty in this process is contained in the phase of the sinusoid. A more common random process that we will be frequently concerned with is noise. Noise is pervasive and occurs in many different places and in many forms. For example, noise may be quantization errors that occur in an A/D converter, it may be round-off noise injected into the output of a fixed point digital filter, or it may be an unwanted disturbance such as engine noise in the cockpit of an airplane or the random disturbances picked up by a sonar array on the bottom of the ocean floor. As the name implies, a random process is

distinguished from a deterministic signal because it is random or nondeterministic. Thus, until a particular signal from the ensemble is selected or observed, the exact values of the process are generally unknown. As a result of this randomness it is necessary to use a different language to represent and describe these signals.

A discrete-time random process is an indexed sequence of random variables. Therefore, what is needed is a framework for describing these random variables and the relationships between them. What will be of primary interest will be the notions of an autocorrelation sequence and a power spectrum. The autocorrelation is a second-order statistical characterization of a random process that provides information about how much linear dependence there is between signal values. This dependence may be exploited in the design of systems for predicting a signal. These predictors, in turn, are important in a number of applications such as speech compression, systems for detecting signals in noise, and filters for reducing interference. The power spectrum, on the other hand, is the Fourier transform of the autocorrelation sequence and provides another representation for the second-order statistics of the process. As is the case for deterministic signals, the frequency domain description of random processes provides a different window through which we may view the process and, in some applications, is so important that we will spend an entire chapter on methods for estimating the power spectrum.

SIGNAL MODELING

The efficient representation of signals is at the heart of many signal processing problems and applications. For example, with the explosion of information that is transmitted and stored in digital form, there is an increasing need to compress these signals so that they may be more efficiently transmitted or stored. Although this book does not address the problem of signal compression directly, it is concerned with the representation of discrete-time signals and with ways of processing these signals to enhance them or to extract some information from them. We will find that once it is possible to accurately model a signal, it then becomes possible to perform important signal processing tasks such as extrapolation and interpolation, and we will be able to use the model to classify signals or to extract certain features or characteristics from them.

One approach that may be used to compress or code a discrete-time signal is to find a model that is able to provide an accurate representation for the signal. For example, we may consider modeling a signal as a sum of sinusoids. The model or code would consist of the amplitudes, frequencies, and phases of the sinusoidal components. There is, of course, a plethora of different models that may be used to represent discrete-time signals, ranging from simple harmonic decompositions to fractal representations [2]. The approach that is used depends upon a number of different factors including the type of signal that is to be compressed and the level of fidelity that is required in the decoded or uncompressed signal. In this book, our focus will be on how we may most accurately model a signal as the unit sample response of a linear shift-invariant filter. What we will discover is that there are many different ways in which to formulate such a signal modeling problem, and that each formulation leads to a solution having different properties. We will also find that the techniques used to solve the signal modeling problem, as well as the solutions themselves, will be useful in our finding solutions to other problems. For example, the FIR Wiener filtering problem may be solved almost by inspection from the solutions that we derive for

all-pole signal modeling, and many of the approaches to the problem of spectrum estimation are based on signal modeling techniques.

THE LEVINSON AND RELATED RECURSIONS

In spite of the ever-increasing power and speed of digital signal processors, there will always be a need to develop *fast algorithms* for performing a specific task or executing a particular algorithm. Many of the problems that we will be solving in this book will require that we find the solution to a set of linear equations. Many different approaches have been developed for solving these equations, such as Gaussian elimination and, in some cases, efficient algorithms exist for finding the solution. As we will see in our development of different approaches to signal modeling and as we will see in our discussions of Wiener filtering, linear equations having a Toeplitz form will appear often. Due to the tremendous amount of structure in these Toeplitz linear equations, we will find that the number of computations required to solve these equations may be reduced from order n^3 required for Gaussian elimation to order n^2. The key in this reduction is the Levinson and the more specialized Levinson-Durbin recursions. Interesting in their own right, what is perhaps even more important are the properties and relationships that are hidden within these recursions. We will see, for example, that embedded in the Levinson-Durbin recursion is a filter structure known as the lattice filter that has many important properties that make them attractive in signal processing applications. We will also find that the Levinson-Durbin recursion forms the basis for a remarkably simple stability test for digital filters.

LATTICE FILTERS

As mentioned in the previous paragraph, the lattice filter is a structure that emerges from the Levinson-Durbin recursion. Although most students who have taken a first course in digital signal processing are introduced to a number of different filter structures such as the direct form, cascade, and parallel structures, the lattice filter is typically not introduced. Nevertheless, there are a number of advantages that a lattice filter enjoys over these other filter structures that often make it a popular structure to use in signal processing applications. The first is the modularity of the filter. It is this modularity and the stage-by-stage optimality of the lattice filter for linear prediction and all-pole signal modeling that allows for the order of the lattice filter to be easily increased or decreased. Another advantage of these filters is that it is trivial to determine whether or not the filter is minimum phase (all of the roots inside the unit circle). Specifically, all that is required is to check that all of the filter coefficients (reflection coefficients) are less than 1 in magnitude. Finally, compared to other filter structures, the lattice filter tends to be less sensitive to parameter quantization effects.

WIENER AND KALMAN FILTERING

There is always a desire or the need to design the optimum filter—the one that will perform a given task or function better than any other filter. For example, in a first course in DSP one learns how to design a linear phase FIR filter that is optimum in the Chebyshev sense of

minimizing the maximum error between the frequency response of the filter and the response of the ideal filter [7]. The Wiener and Kalman filters are also optimum filters. Unlike a typical linear shift-invariant filter, a Wiener filter is designed to process a given signal $x(n)$, the input, and form the best estimate in the mean square sense of a related signal $d(n)$, called the desired signal. Since $x(n)$ and $d(n)$ are not known in advance and may be described only in a statistical sense, it is not possible to simply use $H(e^{j\omega}) = D(e^{j\omega})/X(e^{j\omega})$ as the frequency response of the filter. The Kalman filter may be used similarly to recursively find the best estimate. As we will see, these filters are very general and may be used to solve a variety of different problems such as prediction, interpolation, deconvolution, and smoothing.

SPECTRUM ESTIMATION

As mentioned earlier in this chapter, the frequency domain provides a different window through which one may view a discrete-time signal or random process. The power spectrum is the Fourier transform of the autocorrelation sequence of a stationary process. In a number of applications it is necessary that the power spectrum of a process be known. For example, the IIR Wiener filter is defined in terms of the power spectral densities of two processes, the input to the filter and the desired output. Without prior knowledge of these power spectral densities it becomes necessary to estimate them from observations of the processes. The power spectrum also plays an important role in the detection and classification of periodic or narrowband processes buried in noise.

In developing techniques for estimating the power spectrum of a random process, we will find that the simple approach of Fourier transforming the data from a sample realization does not provide a statistically reliable or high-resolution estimate of the underlying spectrum. However, if we are able to find a model for the process, then this model may be used to estimate the spectrum. Thus, we will find that many of the techniques and results developed for signal modeling will prove useful in understanding and solving the spectrum estimation problem.

ADAPTIVE FILTERS

The final topic considered in this book is adaptive filtering. Throughout most of the discussions of signal modeling, Wiener filtering, and spectrum estimation, it is assumed that the signals that are being processed or analyzed are stationary; that is, their statistical properties are not varying in time. In the real world, however, this will never be the case. Therefore, these problems are reconsidered within the context of nonstationary processes. Beginning with a general FIR Wiener filter, it is shown how a gradient descent algorithm may be used to solve the Wiener-Hopf equations and design the Wiener filter. Although this algorithm is well behaved in terms of its convergence properties, it is not generally used in practice since it requires knowledge of the process statistics, which are generally unknown. Another approach, which has been used successfully in many applications, is the stochastic gradient algorithm known as LMS. Using a simple gradient estimate in place of the true gradient in a gradient descent algorithm, LMS is efficient and well understood in terms of its convergence properties. A variation of the LMS algorithm is the perceptron algorithm used

in pattern recognition and is the starting point for the design of a neural network. Finally, while the LMS algorithm is designed to solve the Wiener filtering problem by minimizing a mean square error, a deterministic least squares approach leads to the development of the RLS algorithm. Although computationally much more involved than the stochastic gradient algorithms, RLS enjoys a significant performance advantage.

There are many excellent textbooks that deal in much more detail and rigor with the subject of adaptive filtering and adaptive signal processing [1, 3, 4, 5, 8]. Here, our goal is to simply provide the bridge to that literature, illustrating how the problems that we have solved in earlier chapters may be adapted to work in nonstationary environments. It is here, too, that we are able to look at some important applications such as linear prediction, channel equalization, interference cancelation, and system identification.

CLOSING

The problems that are considered in the following chapters are fundamental and important. The similarities and relationships between these problems are striking and remarkable. Many of the problem solving techniques presented in this text are powerful and general and may be successfully applied to a variety of other problems in digital signal processing as well as in other disciplines. It is hoped that the reader will share in the excitement of unfolding these relationships and embarking on a journey into the world of statistical signal processing.

References

1. S. T. Alexander, *Adaptive Signal Processing*, Springer-Verlag, New York, 1986.
2. M. Barnsley, *Fractals Everywhere*, Academic Press, New York, 1988.
3. M. G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker, Inc., New York, 1987.
4. P. M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, Boca Raton, FL, 1993.
5. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood-Cliffs, NJ, 1986.
6. M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the history of the Fast Fourier Transform," *IEEE ASSP Magazine*, vol. 1, no. 4, pp. 14–21, October 1984.
7. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
8. B. Widrow and S. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

BACKGROUND 2

2.1 INTRODUCTION

There are two goals of this chapter. The first is to provide the reader with a summary as well as a reference for the fundamentals of discrete-time signal processing and the basic vector and matrix operations of linear algebra. The second is to introduce the notation and terminology that will be used throughout the book. Although it would be possible to move directly to Chapter 3 and only refer to this chapter as a reference as needed, the reader should, at a minimum, peruse the topics presented in this chapter in order to become familiar with what is expected in terms of background and to become acquainted with the terminology and notational conventions that will be used.

The organization of this chapter is as follows. The first part contains a review of the fundamentals of discrete-time signal processing. This review includes a summary of important topics in discrete-time signal manipulation and representation such as linear filtering, difference equations, digital networks, discrete-time Fourier transforms, z-transforms, and the DFT. Since this review is not intended to be a comprehensive treatment of discrete-time signal processing, the reader wishing a more in-depth study may consult any one of a number of excellent texts including [7, 8, 11]. The second part of this chapter summarizes the basic ideas and techniques from linear algebra that will be used in later chapters. The topics that are covered include vector and matrix manipulations, the solution of linear equations, eigenvalues and eigenvectors, and the minimization of quadratic forms. For a more detailed treatment of linear algebra, the reader may consult any one of a number of standard textbooks such as [6, 10].

2.2 DISCRETE-TIME SIGNAL PROCESSING

In this brief overview of the fundamentals of discrete-time signal processing, we focus primarily on the specification and characterization of discrete-time signals and systems. Of particular importance are the topics of discrete-time filters, the transform analysis of discrete-time signals and systems, and digital filter flowgraphs. Since the goal of this section is only to review these topics while introducing notation and terminology, the discussions will be cursory with most of the mathematical details being left to the references.

2.2.1 Discrete-Time Signals

A discrete-time signal is an indexed sequence of real or complex numbers.[1] Thus, a discrete-time signal is a function of an integer-valued variable, n, that is denoted by $x(n)$. Although the independent variable n need not necessarily represent "time" (n may, for example, correspond to a spatial coordinate or distance), $x(n)$ is generally referred to as a function of time. Since a discrete-time signal is undefined for noninteger values of n, the graphical representation of $x(n)$ will be in the form of a *lollipop* plot, as shown in Fig. 2.1.

Discrete-time signals may arise as a result of sampling a continuous-time signal, such as speech, with an analog-to-digital (A/D) converter. For example, if a continuous-time signal $x_a(t)$ is sampled at a rate of $f_s = 1/T_s$ samples per second, then the sampled signal $x(n)$ is related to $x_a(t)$ as follows

$$x(n) = x_a(nT_s)$$

Not all discrete-time signals, however, are obtained in this manner. In particular, some signals may be considered to be naturally occurring discrete-time sequences since there is no physical analog-to-digital converter that is converting an analog signal into a discrete-time signal. Examples of signals that fall into this category include daily stock market prices, population statistics, warehouse inventories, and the Wolfer sunspot numbers.

Although most information-bearing signals of practical interest are complicated functions of time, there are three simple yet important discrete-time signals that are frequently used in the representation and description of more complicated signals. These are the unit sample, the unit step, and the complex exponential. The *unit sample*, denoted by $\delta(n)$, is defined by

$$\delta(n) = \begin{cases} 1 & ; \quad n = 0 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

and plays the same role in discrete-time signal processing that the unit impulse plays in continuous-time signal processing. The unit sample may be used to decompose an arbitrary signal $x(n)$ into a sum of weighted (scaled) and shifted unit samples as follows

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

This decomposition is the discrete version of the *sifting property* for continuous-time signals. The *unit step*, denoted by $u(n)$, is defined by

$$u(n) = \begin{cases} 1 & ; \quad n \geq 0 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

and is related to the unit sample by

$$u(n) = \sum_{k=-\infty}^{n} \delta(k)$$

Finally, a *complex exponential* is the periodic signal

$$e^{jn\omega_0} = \cos(n\omega_0) + j\sin(n\omega_0)$$

[1] Discrete-time signals may be either deterministic or random (stochastic). Here it is assumed that the signals are deterministic. In Chapter 3 we will consider the characterization of discrete-time random processes.

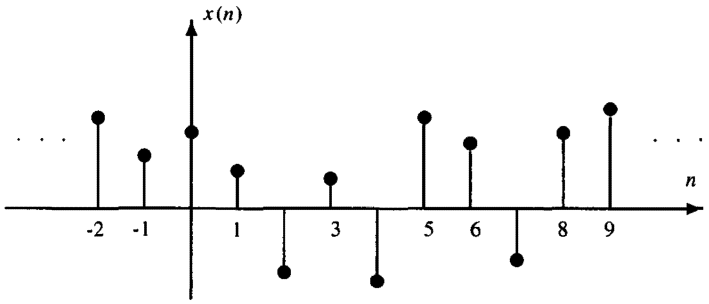


Figure 2.1 *The graphical representation of a discrete-time signal* $x(n)$.

where ω_0 is some real constant. Complex exponentials are useful in the Fourier decomposition of signals as described in Section 2.2.4.

Discrete-time signals may be conveniently classified in terms of their duration or extent. For example, a discrete-time sequence is said to be a *finite length sequence* if it is equal to zero for all values of n outside a finite interval $[N_1, N_2]$. Signals that are not finite in length, such as the unit step and the complex exponential, are said to be *infinite length sequences*. Since the unit step is equal to zero for $n < 0$, it is said to be a *right-sided sequence*. In general, a right-sided sequence is any infinite-length sequence that is equal to zero for all values of $n < n_0$ for some integer n_0. Similarly, an infinite-length sequence $x(n)$ is said to be *left sided* if, for some integer n_0, $x(n) = 0$ for all $n > n_0$. An example of a left-sided sequence is

$$x(n) = u(n_0 - n) = \begin{cases} 1 & ; \quad n \leq n_0 \\ 0 & ; \quad n > n_0 \end{cases}$$

which is a time-reversed and delayed unit step. An infinite-length signal that is neither right sided nor left sided, such as the complex exponential, is referred to as a *two-sided sequence*.

2.2.2 Discrete-time Systems

A discrete-time system is a mathematical operator or mapping that transforms one signal (the input) into another signal (the output) by means of a fixed set of rules or functions. The notation $T[-]$ will be used to represent a general system, such as the one shown in Fig. 2.2, in which an input signal $x(n)$ is transformed into an output signal $y(n)$ through the transformation $T[-]$. The input/output properties of such a system may be specified in any one of a number of different ways. The relationship between the input and output, for example, may be expressed in terms of a concise mathematical rule or function such as

$$y(n) = x^2(n)$$

or

$$y(n) = 0.5y(n-1) + x(n)$$

It is also possible, however, to describe a system in terms of an algorithm that provides a sequence of instructions or operations that is to be applied to the input signal values. For

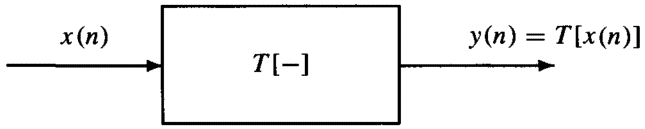


Figure 2.2 *The representation of a discrete-time system as a transformation $T[-]$ that maps an input signal $x(n)$ into an output signal $y(n)$.*

example,

$$\begin{aligned} y_1(n) &= 0.5y_1(n-1) + 0.25x(n) \\ y_2(n) &= 0.25y_2(n-1) + 0.5x(n) \\ y_3(n) &= 0.4y_3(n-1) + 0.5x(n) \\ y(n) &= y_1(n) + y_2(n) + y_3(n) \end{aligned}$$

is a sequence of instructions that defines a third-order recursive digital filter in parallel form. In some cases, a system may conveniently be specified in terms of a table that defines the set of all possible input/output signal pairs of interest. In the Texas Instruments *Speak and Spell*™, for example, pushing a specific button (the input signal) results in the synthesis of a given spoken letter or word (the system output).

Discrete-time systems are generally classified in terms of the properties that they possess. The most common properties include linearity, shift-invariance, causality, stability, and invertibility, which are described in the following sections.

Linearity and Shift-Invariance. The two system properties that are of the greatest importance for simplifying the analysis and design of discrete-time systems are *linearity* and *shift-invariance*. A system $T[-]$ is said to be *linear* if, for any two inputs $x_1(n)$ and $x_2(n)$ and for any two (complex-valued) constants a and b,

$$T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)]$$

In other words, the response of a linear system to a sum of two signals is the sum of the two responses, and scaling the input by a constant results in the output being scaled by the same constant. The importance of this property is evidenced in the observation that if the input is decomposed into a superposition of weighted and shifted unit samples,

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

then the output is

$$\boxed{y(n) = \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)] = \sum_{k=-\infty}^{\infty} x(k)h_k(n)} \tag{2.1}$$

where $h_k(n) = T[\delta(n-k)]$ is the response of the system to the delayed unit sample $\delta(n-k)$. Equation (2.1) is referred to as the *superposition sum* and it illustrates that a linear system is completely specified once the signals $h_k(n)$ are known.

The second important property is *shift-invariance*.[2] A system is said to be shift-invariant if a shift in the input by n_0 results in a shift in the output by n_0. Thus, if $y(n)$ is the response

[2]Some authors use the term *time-invariance* instead of shift-invariance. However, since the independent variable, n, may represent something other than time, such as distance, we prefer to use the term *shift-invariance*.

of a shift-invariant system to an input $x(n)$, then for any shift in the input, $x(n - n_0)$, the response of the system will be $y(n - n_0)$. In effect, shift-invariance means that the properties of the system do not change with time.

A system that is both linear and shift-invariant is called a *linear shift-invariant* (LSI) system. For a shift-invariant system, if $h(n) = T[\delta(n)]$ is the response to a unit sample $\delta(n)$, then the response to $\delta(n - k)$ is $h(n - k)$. Therefore, for an LSI system the superposition sum given in Eq. (2.1) becomes

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n - k) \tag{2.2}$$

which is the *convolution sum*. In order to simplify notation, the convolution sum is often expressed as

$$y(n) = x(n) * h(n)$$

Since the convolution sum allows one to evaluate the response of an LSI system to an arbitrary input $x(n)$, LSI systems are uniquely characterized by their response, $h(n)$, to a unit sample. This signal is referred to as the *unit sample response* of the system.

Causality. A system property that is important for real-time applications is *causality*. A system is said to be *causal* if, for any n_0, the response of the system at time n_0 depends only upon the values of the input for $n \leq n_0$. For a causal system it is not possible for changes in the output to precede changes in the input. The system described by the equation $y(n) = x(n) + x(n - 1)$, for example, is causal since the value of the output at any time $n = n_0$ depends only on the input $x(n)$ at time n_0 and at time $n_0 - 1$. The system described by $y(n) = x(n) + x(n + 1)$, on the other hand, is noncausal since the output at time $n = n_0$ depends on the value of the input at time $n_0 + 1$. If a system is linear and shift-invariant then it will be causal if and only if $h(n) = 0$ for $n < 0$.

Stability. In many applications, it is important for a system to have a response, $y(n)$, that is bounded in amplitude whenever the input is bounded. A system with this property is said to be *stable* in the Bounded-Input Bounded-Output (BIBO) sense. More specifically, a system is BIBO stable if, for any bounded input, $|x(n)| \leq A < \infty$, the output is bounded, $|y(n)| \leq B < \infty$. In the case of a linear shift-invariant system, stability is guaranteed whenever the unit sample response is absolutely summable

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \tag{2.3}$$

For example, an LSI system with $h(n) = a^n u(n)$ is stable whenever $|a| < 1$.

Invertibility. A system property that is important in applications such as channel equalization and deconvolution is *invertibility* (see Section 4.4.5 for the design of an FIR least squares inverse filter). A system is said to be *invertible* if the input to the system may be uniquely determined by observing the output. In order for a system to be invertible, it is necessary for distinct inputs to produce distinct outputs. In other words, given any two inputs $x_1(n)$ and $x_2(n)$ with $x_1(n) \neq x_2(n)$, it must be true that $y_1(n) \neq y_2(n)$. For example,

the system defined by

$$y(n) = x(n)g(n)$$

is invertible if and only if $g(n) \neq 0$ for all n. Specifically, if $g(n)$ is nonzero for all n, then, given $y(n)$, the input may be reconstructed by $x(n) = y(n)/g(n)$.

2.2.3 Time-Domain Descriptions of LSI Filters

An important class of linear shift-invariant systems are those for which the input $x(n)$ and output $y(n)$ are related by a linear constant coefficient difference equation (LCCDE) of the form

$$y(n) + \sum_{k=1}^{p} a(k)y(n-k) = \sum_{k=0}^{q} b(k)x(n-k) \tag{2.4}$$

In this difference equation, p and q are integers that determine the *order* of the system and $a(1), \ldots, a(p)$ and $b(0), \ldots, b(q)$ are the *filter coefficients* that define the system. The difference equation is often written in the form

$$\boxed{y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k)} \tag{2.5}$$

which clearly shows that the output $y(n)$ is equal to a linear combination of past output values, $y(n-k)$ for $k = 1, 2, \ldots, p$, along with past and present input values, $x(n-k)$ for $k = 0, 1, \ldots, q$. For the special case of $p = 0$, the difference equation becomes

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k) \tag{2.6}$$

and the output is simply a weighted sum of the current and past input values. As a result, the unit sample response is finite in length

$$h(n) = \sum_{k=0}^{q} b(k)\delta(n-k)$$

and the system is referred to as a *Finite length Impulse Response* (*FIR*) system. However, if $p \neq 0$ then the unit sample response is, in general, infinite in length and the system is referred to as an *Infinite length Impulse Response* (*IIR*) system. For example, if

$$y(n) = ay(n-1) + x(n)$$

then the unit sample response is $h(n) = a^n u(n)$.

2.2.4 The Discrete-Time Fourier Transform

Frequency analysis of discrete-time signals and systems provides an important analysis and design tool and often provides insights into the solution of problems that would not otherwise be possible. Of central importance in the frequency analysis of discrete-time signals is the Discrete-Time Fourier Transform (DTFT). The DTFT of a signal $x(n)$ is the

complex-valued function of the continuous (frequency) variable, ω, defined by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-jn\omega} \tag{2.7}$$

In order for the DTFT of a signal to be defined, however, the sum in Eq. (2.7) must converge. A sufficient condition for the sum to converge uniformly to a continuous function of ω is that $x(n)$ be absolutely summable

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \tag{2.8}$$

Although most signals of interest will have a DTFT, signals such as the unit step and the complex exponential are not absolutely summable and do not have a DTFT. However, if we allow the DTFT to contain generalized functions then the DTFT of a complex exponential is an impulse

$$x(n) = e^{jn\omega_0} \longrightarrow X(e^{j\omega}) = 2\pi u_0(\omega - \omega_0) \quad ; \quad |\omega| < \pi$$

where $u_0(\omega - \omega_0)$ is used to denote an impulse at frequency $\omega = \omega_0$. Similarly, the DTFT of a unit step is

$$u(n) \longrightarrow U(e^{j\omega}) = \frac{1}{1 - e^{-j\omega}} + \pi u_0(\omega) \quad ; \quad |\omega| < \pi$$

The DTFT possesses some symmetry properties of interest. For example, since $e^{-jn\omega}$ is periodic in ω with a period of 2π, it follows that $X(e^{j\omega})$ is also periodic with a period of 2π. In addition, if $x(n)$ is real-valued then $X(e^{j\omega})$ will be conjugate symmetric

$$X(e^{j\omega}) = X^*(e^{-j\omega})$$

The DTFT is, in general, a complex-valued function of ω. Therefore, it is normally represented in polar form in terms of its magnitude and phase

$$X(e^{j\omega}) = \left|X(e^{j\omega})\right| e^{j\phi_x(\omega)}$$

For real signals, the conjugate symmetry of $X(e^{j\omega})$ implies that the magnitude is an even function, $|X(e^{j\omega})| = |X(e^{-j\omega})|$ and that the phase is an odd function, $\phi_x(\omega) = -\phi_x(\omega)$.

A discrete-time Fourier transform of special importance is the DTFT of the unit sample response of a linear shift-invariant system,

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jn\omega} \tag{2.9}$$

This DTFT is called the *frequency response* of the filter, and it defines how a complex exponential is changed in amplitude and phase by the system. Note that the condition for the existence of the DTFT given in Eq. (2.8) is the same as the condition for BIBO stability of an LSI system. Therefore, it follows that the DTFT of $h(n)$ exists for BIBO stable systems.

The DTFT is an invertible transformation in the sense that, given the DTFT $X(e^{j\omega})$ of a signal $x(n)$, the signal may be recovered using the Inverse DTFT (IDTFT) as follows

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{jn\omega} d\omega \tag{2.10}$$

In addition to providing a method for computing $x(n)$ from $X(e^{j\omega})$, the IDTFT may also be viewed as a decomposition of $x(n)$ into a linear combination of complex exponentials.

There are a number of useful and important properties of the DTFT. Perhaps the most important of these is the *convolution theorem*, which states that the DTFT of a convolution of two signals

$$y(n) = x(n) * h(n)$$

is equal to the product of the transforms,

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

Another useful property is *Parseval's theorem*, which states that the sum of the squares of a signal, $x(n)$, is equal to the integral of the square of its DTFT,

$$\boxed{\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega} \tag{2.11}$$

Some additional properties of the DTFT are listed in Table 2.1. Derivations and applications of these properties may be found in references [7, 8, 11].

2.2.5 The z-Transform

The z-transform is a generalization of the discrete-time Fourier transform that allows many signals not having a DTFT to be described using transform techniques. The z-transform of a discrete-time signal $x(n)$ is defined as

$$\boxed{X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}} \tag{2.12}$$

where $z = re^{j\omega}$ is a complex variable. Note that when $r = 1$, or $z = e^{j\omega}$, the z-transform

Table 2.1 Properties of the DTFT

Property	Sequence	Transform
	$x(n)$	$X(e^{j\omega})$
Delay	$x(n - n_0)$	$e^{-j\omega n_0}X(e^{j\omega})$
Modulation	$e^{j\omega_0 n}x(n)$	$X(e^{j(\omega-\omega_0)})$
Conjugation	$x^*(n)$	$X^*(e^{-j\omega})$
Time reversal	$x(-n)$	$X(e^{-j\omega})$
Convolution	$x(n) * y(n)$	$X(e^{j\omega})Y(e^{j\omega})$
Multiplication	$x(n)y(n)$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})d\theta$
Multiplication by n	$nx(n)$	$j\frac{d}{d\omega}X(e^{j\omega})$
Parseval	$\sum_{n=-\infty}^{\infty} x(n)y^*(n)$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})Y^*(e^{j\omega})d\omega$

becomes the discrete-time Fourier transform,

$$X(e^{j\omega}) = X(z)\big|_{z=e^{j\omega}} = \sum_{n=-\infty}^{\infty} x(n)e^{-jn\omega}$$

As with the DTFT, the z-transform is only defined when the sum in Eq. (2.12) converges. Since the sum generally does not converge for all values of z, associated with each z-transform is a *region of convergence* that defines those values of z for which the sum converges. For a finite length sequence, the sum in Eq. (2.12) contains only a finite number of terms. Therefore, the z-transform of a finite length sequence is a polynomial in z and the region of convergence will include all values of z (except possibly $z = 0$ or $z = \infty$). For right-sided sequences, on the other hand, the region of convergence is the exterior of a circle, $|z| > R_-$, and for left-sided sequences it is the interior of a circle, $|z| < R_+$, where R_- and R_+ are positive numbers. For two-sided sequences, the region of convergence is an annulus

$$R_- < |z| < R_+$$

Just as for the discrete-time Fourier transform, the z-transform has a number of important and useful properties, some of which are summarized in Table 2.2. In addition, a symmetry condition that will be of interest in our discussions of the power spectrum is the following. If $x(n)$ is a conjugate symmetric sequence,

$$x(n) = x^*(-n)$$

then it z-transform satisfies the relation

$$\boxed{X(z) = X^*(1/z^*)} \tag{2.13}$$

This property follows by combining the conjugation and time-reversal properties listed in Table 2.2. Finally, given in Table 2.3 are some closed-form expressions for some commonly found summations. These are often useful in evaluating the z-transform given in Eq. (2.12).

A z-transform of special importance in the design and analysis of linear shift-invariant systems is the *system function*, which is the z-transform of the unit sample response,

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

Table 2.2 *Properties of the z-Transform*

Property	Sequence	Transform
	$x(n)$	$X(z)$
Delay	$x(n - n_0)$	$z^{-n_0}X(z)$
Multiplication by α^n	$\alpha^n x(n)$	$X(z/\alpha)$
Conjugation	$x^*(n)$	$X^*(z^*)$
Time reversal	$x(-n)$	$X(z^{-1})$
Convolution	$x(n) * h(n)$	$X(z)H(z)$
Multiplication by n	$nx(n)$	$-z\frac{d}{dz}X(z)$

Table 2.3 **Closed-form Expressions for Some Commonly Encountered Series**

$\sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a}$	$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a} \quad ; \quad \lvert a\rvert < 1$
$\sum_{n=0}^{N-1} na^n = \frac{(N-1)a^{N+1} - Na^N + a}{(1-a)^2}$	$\sum_{n=0}^{\infty} na^n = \frac{a}{(1-a)^2} \quad ; \quad \lvert a\rvert < 1$
$\sum_{n=0}^{N-1} n = \frac{1}{2}N(N-1)$	$\sum_{n=0}^{N-1} n^2 = \frac{1}{6}N(N-1)(2N-1)$

For an FIR filter described by a LCCDE of the form given in Eq. (2.6), the system function is a polynomial in z^{-1},

$$H(z) = \sum_{k=0}^{q} b(k)z^{-k} = b(0)\prod_{k=1}^{q}(1 - z_k z^{-1}) \tag{2.14}$$

and the roots of this polynomial, z_k, are called the *zeros* of the filter. Due to the form of $H(z)$ for FIR filters, they are often referred to as *all-zero* filters. For an IIR filter described by the general difference equation given in Eq. (2.5), the system function is a ratio of two polynomials in z^{-1},

$$H(z) = \frac{\sum_{k=0}^{q} b(k)z^{-k}}{1 + \sum_{k=1}^{p} a(k)z^{-k}} = b(0)\frac{\prod_{k=1}^{q}(1 - z_k z^{-1})}{\prod_{k=1}^{p}(1 - p_k z^{-1})} \tag{2.15}$$

The roots of the numerator polynomial, z_k, are the zeros of $H(z)$ and the roots of the denominator polynomial, p_k, are called the *poles*. If the order of the numerator polynomial is zero, $q = 0$, then

$$H(z) = \frac{b(0)}{1 + \sum_{k=1}^{p} a(k)z^{-k}} = \frac{b(0)}{\prod_{k=1}^{p}(1 - p_k z^{-1})} \tag{2.16}$$

and $H(z)$ is called an *all-pole* filter.

If the coefficients $a(k)$ and $b(k)$ in the system function are real-valued (equivalently, if $h(n)$ is real-valued) then

$$H(z) = H^*(z^*)$$

and the poles and zeros of $H(z)$ will occur in conjugate pairs. That is, if $H(z)$ has a pole (zero) at $z = a$, then $H(z)$ will have a pole (zero) at $z = a^*$. Some useful z-transform pairs may be found in Table 2.4.

2.2.6 Special Classes of Filters

There are several special classes of filters that we will be encountering in the following chapters. The first are filters that have *linear phase*. These filters are important in applications

Table 2.4 *Some Useful z-Transform Pairs*

Sequence	Transform	Region of Convergence
$\delta(n)$	1	All z
$\alpha^n\left[u(n) - u(n-N)\right]$	$\dfrac{1-\alpha^N z^{-N}}{1-\alpha z^{-1}}$	$\lvert z\rvert > 0$
$\alpha^n u(n)$	$\dfrac{1}{1-\alpha z^{-1}}$	$\lvert z\rvert > \alpha$
$-\alpha^n u(-n-1)$	$\dfrac{1}{1-\alpha z^{-1}}$	$\lvert z\rvert < \alpha$
$\alpha^{\lvert n\rvert}$	$\dfrac{1-\alpha^2}{(1-\alpha z^{-1})(1-\alpha z)}$	$\alpha < \lvert z\rvert < 1/\alpha$

such as speech and image processing and have a frequency response of the form

$$H(e^{j\omega}) = A(e^{j\omega})e^{j(\beta-\alpha\omega)} \tag{2.17}$$

where $A(e^{j\omega})$ is a real-valued function of ω and α and β are constants.[3] In order for a *causal* filter to have linear phase and be realizable using a *finite-order* linear constant coefficient difference equation, the filter must be FIR [2, 7]. In addition, the linear phase condition places the constraint that the unit sample response $h(n)$ be either conjugate symmetric (Hermitian),

$$h^*(n) = h(N-1-n)$$

or conjugate antisymmetric (anti-Hermitian)

$$h^*(n) = -h(N-1-n)$$

These constraints, in turn, impose the constraint that the zeros of the system function $H(z)$ occur in conjugate reciprocal pairs,

$$H^*(z^*) = \pm z^{N-1}H(1/z)$$

In other words, if $H(z)$ has a zero at $z = z_0$, then $H(z)$ must also have a zero at $z = 1/z_0^*$.

Another filter having a special form is the *allpass filter*. Allpass filters are useful in applications such as phase equalization and have a frequency response with a constant magnitude

$$|H(e^{j\omega})| = A$$

For an allpass filter having a rational system function, $H(z)$ must be of the form

$$H(z) = z^{-n_0}A\prod_{k=1}^{N}\frac{z^{-1}-\alpha_k^*}{1-\alpha_k z^{-1}}$$

Thus, if $H(z)$ has a zero (pole) at $z = \alpha_k$, then $H(z)$ must also have a pole (zero) at $z = 1/\alpha_k^*$.

[3]The term linear phase is often reserved for the special case in which $\beta = 0$ and $A(e^{j\omega})$ is nonnegative. Filters of the form given in Eq. (2.17) are then said to have *generalized linear phase*. Here we will not be concerned about this distinction.

Finally, a stable and causal filter that has a rational system function with all of its poles and zeros *inside* the unit circle is said to be a *minimum phase filter*. Thus, for a minimum phase filter, $|z_k| < 1$ and $|p_k| < 1$ in Eq. (2.15). Note that a minimum phase filter will have a stable and causal inverse, $1/H(z)$, which will also be minimum phase. This property will be useful in the development of the spectral factorization theorem in Section 3.5 and in the derivation of the causal Wiener filter in Chapter 7.

2.2.7 Filter Flowgraphs

A LCCDE defines the relationship between the filter input, $x(n)$, and the filter output, $y(n)$. In the implementation of a filter, either in hardware or software, there are many different ways in which the computations necessary to compute the value of the output $y(n)$ at time n may be ordered. For example, the difference equation

$$y(n) = 0.2y(n-1) + x(n) + 0.5x(n-1)$$

and the pair of coupled difference equations

$$\begin{aligned} w(n) &= 0.2w(n-1) + x(n) \\ y(n) &= w(n) + 0.5w(n-1) \end{aligned}$$

are two implementations of a first-order IIR filter having a system function

$$H(z) = \frac{1 + 0.5z^{-1}}{1 - 0.2z^{-1}}$$

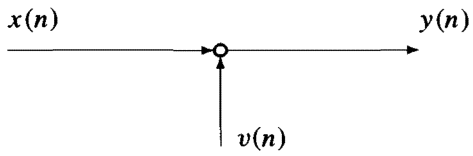
In describing these different filter implementations it is convenient to use *flowgraphs* or *digital networks* that show how a given system is realized in terms of interconnections of the basic computational elements of the filter, i.e., adders, multipliers, and delays. The notation that will be used to represent these computational elements is shown in Fig. 2.3. As an example, shown in Fig. 2.4*a* is a fourth-order IIR filter implemented in *direct form* and in Fig. 2.4*b* is an Nth-order FIR filter, referred to as a *tapped delay line*.

One structure of particular importance that will be developed in Chapter 6 is the *lattice filter*. As we will see, this structure has a number of useful and important properties compared to other filter structures.

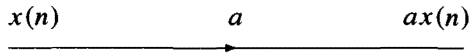
2.2.8 The DFT and FFT

The discrete-time Fourier transform provides a Fourier representation of a discrete-time signal that is useful as a design and analysis tool. However, since the DTFT is a function of a continuous variable, ω, it is not directly amenable to digital computation. For finite length sequences, there is another representation called the *Discrete Fourier Transform* (*DFT*) that is a function of an integer variable, k, and is therefore easily evaluated with a digital computer. For a finite-length sequence $x(n)$ of length N that is equal to zero outside the interval $[0, N-1]$, the *N-point DFT* is

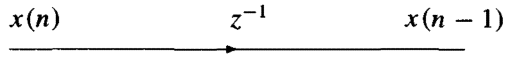
$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \tag{2.18}$$



(a) An adder.

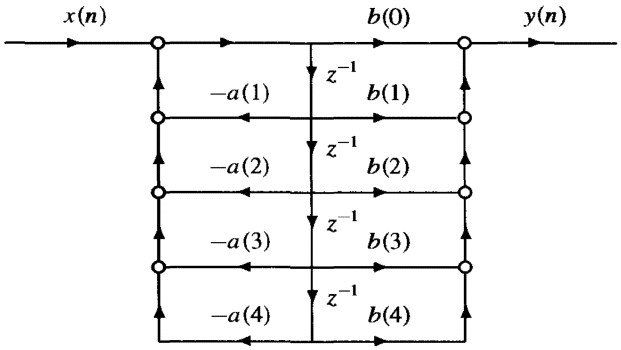


(b) Multiplication by a constant.

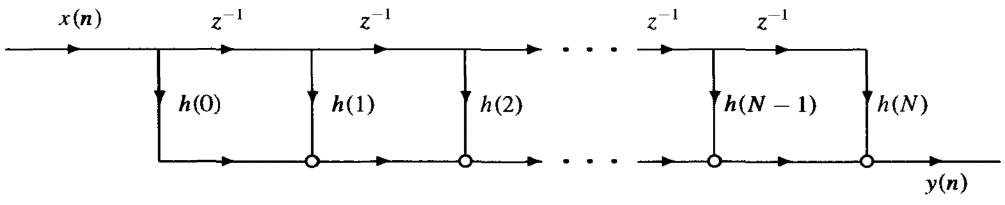


(c) A unit delay.

Figure 2.3 *Flowgraph notation for digital networks.*



(a) Direct form II filter structure for a 4th order IIR filter.



(a) Tapped delay line filter structure for an FIR filter.

Figure 2.4 *Filter flowgraphs for a fourth-order IIR and an Nth-order FIR filter.*

Note that the DFT of $x(n)$ is equal to the discrete-time Fourier transform sampled at N frequencies that are equally spaced between 0 and 2π,

$$X(k) = X(e^{j\omega})|_{\omega=2\pi k/N} \tag{2.19}$$

The Inverse Discrete Fourier Transform (IDFT)

$$\boxed{x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}} \tag{2.20}$$

provides the relationship required to determine $x(n)$ from the DFT coefficients $X(k)$.

Recall that the product of the discrete-time Fourier transforms of two signals corresponds, in the time domain, to the (linear) convolution of the two signals. For the DFT, however, if $H(k)$ and $X(k)$ are the N-point DFTs of $h(n)$ and $x(n)$, respectively, and if $Y(k) = X(k)H(k)$, then

$$y(n) = \sum_{k=0}^{N} x((k))_N h((n-k))_N$$

which is the N-point *circular convolution* of $x(n)$ and $h(n)$ where

$$x((n))_N \equiv x(n \bmod N)$$

In general, circular convolution of two sequences is not the same as linear convolution. However, there is a special and important case in which the two are the same. Specifically, if $x(n)$ is a finite-length sequence of length N_1 and $h(n)$ is a finite-length sequence of length N_2, then the linear convolution of $x(n)$ and $h(n)$ is of length $L = N_1 + N_2 - 1$. In this case, the N-point circular convolution of $x(n)$ and $h(n)$ will be equal to the linear convolution provided $N \geq L$.

In using the DFT to perform convolutions or as a method for evaluating samples of the discrete-time Fourier transform in real-time applications, it is useful to know what the computational requirements would be to find the DFT. If the DFT is evaluated directly using Eq. (2.19), then N complex multiplications and $N - 1$ complex additions would be required for each value of k. Therefore, an N-point DFT requires a total of N^2 complex multiplications and $N^2 - N$ complex additions. [4] This number may be reduced significantly, however, by employing any one of a number of *Fast Fourier Transform* (*FFT*) algorithms [7]. For example, if N is a power of 2, $N = 2^{\mu}$, a radix-2 decimation-in-time FFT algorithm requires approximately $\frac{N}{2}\log_2 N$ complex multiplications and $N \log_2 N$ complex additions. For large N, this represents a significant reduction.

2.3 LINEAR ALGEBRA

In many of the mathematical developments that will be encountered in later chapters, it will be convenient to use vector and matrix notation for the representation of signals and the operations that are performed on them. Such a representation will greatly simplify many of the mathematical expressions from a notational point of view and allow us to draw upon

[4] Since we are not taking into account the fact that some of these multiplications are by ± 1, the actual number is a bit smaller than this.

many useful results from linear algebra to simplify or solve these expressions. Although it will not be necessary to delve too deeply into the theory of linear algebra, it will be important to become familiar with the basic tools of vector and matrix analysis. Therefore, in this section we summarize the results from linear algebra that will prove to be useful in this book.

2.3.1 Vectors

A vector is an array of real-valued or complex-valued numbers or functions. Vectors will be denoted by lowercase bold letters, such as **x**, **a**, and **v** and, in all cases, these vectors will be assumed to be column vectors. For example,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

is a column vector containing N scalars. If the elements of **x** are real, then **x** it is said to be a *real vector*, whereas if they are complex, then **x** is said to be a *complex vector*. A vector having N elements is referred to as an N-dimensional vector. The transpose of a vector, $\mathbf{x}^T$, is a row vector

$$\mathbf{x}^T = \begin{bmatrix} x_1, & x_2, \ldots, & x_N \end{bmatrix}$$

and the *Hermitian transpose*, $\mathbf{x}^H$, is the complex conjugate of the transpose of **x**

$$\mathbf{x}^H = (\mathbf{x}^T)^* = \begin{bmatrix} x_1^*, & x_2^*, \ldots, & x_N^* \end{bmatrix}$$

Vectors are useful for representing the values of a discrete-time signal in a concise way. For example, a finite length sequence $x(n)$ that is equal to zero outside the interval $[0, N-1]$ may be represented in vector form as

$$\mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{2.21}$$

It is also convenient, in some cases, to consider the set of vectors, $\mathbf{x}(n)$, that contain the signal values $x(n), x(n-1), \ldots, x(n-N+1)$,

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix} \tag{2.22}$$

Thus, $\mathbf{x}(n)$ is a vector of N elements that is parameterized by the time index n.

Many of the operations that we will be performing on vectors will involve finding the *magnitude* of a vector. In order to find the magnitude, however, it is necessary to define a *norm* or *distance metric*. The *Euclidean* or $\mathcal{L}_2$ *Norm* is one such measure that, for a vector

$\mathbf{x}$ of dimension N, is

$$\boxed{\|\mathbf{x}\|_2 = \left\{\sum_{i=1}^{N} |x_i|^2\right\}^{1/2}} \tag{2.23}$$

Other useful norms include the $\mathcal{L}_1$ norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$$

and the $\mathcal{L}_\infty$ norm

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

In this book, since the $\mathcal{L}_2$ norm will be used almost exclusively, the vector norm will be denoted simply by $\|\mathbf{x}\|$ and will be interpreted as the $\mathcal{L}_2$ norm unless indicated otherwise.

A vector may be *normalized* to have unit magnitude by dividing by its norm. For example, assuming that $\|\mathbf{x}\| \neq 0$, then

$$\mathbf{v}_x = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

is a *unit norm* vector that lies in the same *direction* as $\mathbf{x}$. For a vector whose elements are signal values, $x(n)$, the norm squared represents the energy in the signal. For example, the squared norm of the vector $\mathbf{x}$ in Eq. (2.21) is

$$\|\mathbf{x}\|^2 = \sum_{n=0}^{N-1} |x(n)|^2$$

In addition to providing a metric for the *length* of a vector, the norm may also be used to measure the distance between two vectors

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \left\{\sum_{i=1}^{N} |x_i - y_i|^2\right\}^{1/2}$$

Given two complex vectors $\mathbf{a} = [a_1, \ldots, a_N]^T$ and $\mathbf{b} = [b_1, \ldots, b_N]^T$, the *inner product* is the scalar defined by

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^H \mathbf{b} = \sum_{i=1}^{N} a_i^* b_i$$

For real vectors the inner product becomes

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^{N} a_i b_i$$

The inner product defines the geometrical relationship between two vectors. This relationship is given by

$$\boxed{\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta} \tag{2.24}$$

where θ is the angle between the two vectors. Thus, two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$ are said to be *orthogonal* if their inner product is zero

$$\langle \mathbf{a}, \mathbf{b} \rangle = 0$$

Two vectors that are orthogonal and have unit norm are said to be *orthonormal.*

Example 2.3.1 ***Inner Product***

Consider the two unit norm vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The inner product between $\mathbf{v}_1$ and $\mathbf{v}_2$ is

$$< \mathbf{v}_1, \mathbf{v}_2 > = \frac{1}{\sqrt{2}} = \cos\theta$$

where $\theta = \pi/4$. Therefore, as we know from Euclidean geometry, $\mathbf{v}_1$ and $\mathbf{v}_2$ form an angle of 45 degrees with respect to each other. If, on the other hand,

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

then the inner product is zero and $\mathbf{v}_1$ and $\mathbf{v}_2$ are orthogonal.

Note that since $|\cos\theta| \leq 1$, the inner product between two vectors is bounded by the product of their magnitudes

$$\boxed{|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\| \, \|\mathbf{b}\|} \tag{2.25}$$

where equality holds if and only if $\mathbf{a}$ and $\mathbf{b}$ are colinear, i.e., $\mathbf{a} = \alpha\mathbf{b}$ for some constant α. Equation (2.25) is known as the *Cauchy-Schwarz inequality.* Another useful inequality is the following

$$\boxed{2|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2} \tag{2.26}$$

with equality holding if and only if $\mathbf{a} = \pm\mathbf{b}$. This inequality may be established by noting that, for any two vectors $\mathbf{a}$ and $\mathbf{b}$,

$$\|\mathbf{a} \pm \mathbf{b}\|^2 \geq 0$$

Expanding the norm we have

$$\|\mathbf{a} \pm \mathbf{b}\|^2 = \|\mathbf{a}\|^2 \pm 2\langle \mathbf{a}, \mathbf{b} \rangle + \|\mathbf{b}\|^2 \geq 0$$

from which the inequality easily follows.

One of the uses of the inner product is to represent, in a concise way, the output of a linear shift-invariant filter. For example, let $h(n)$ be the unit sample response of an FIR filter of order $N-1$ and let $x(n)$ be the filter input. The filter output is the convolution of $h(n)$ and $x(n)$,

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Therefore, expressing $x(n)$ in vector form using Eq. (2.22) and writing $h(n)$ in vector form as

$$\mathbf{h} = \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix}$$

then $y(n)$ may be written as the inner product

$$y(n) = \mathbf{h}^T\mathbf{x}(n)$$

2.3.2 Linear Independence, Vector Spaces, and Basis Vectors

Linear dependence and linear independence of a set of vectors are extremely important concepts in linear algebra. A set of n vectors, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is said to be *linearly independent* if

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n = 0 \tag{2.27}$$

implies that $\alpha_i = 0$ for all i. If a set of nonzero α_i can be found so that Eq. (2.27) holds, then the set is said to be *linearly dependent*. If $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is a set of linearly dependent vectors, then it follows that at least one of the vectors, say $\mathbf{v}_1$, may be expressed as a linear combination of the remaining vectors

$$\mathbf{v}_1 = \beta_2\mathbf{v}_2 + \beta_3\mathbf{v}_3 + \cdots + \beta_n\mathbf{v}_n$$

for some set of scalars, β_i. For vectors of dimension N, no more than N vectors may be linearly independent, i.e., any set containing more than N vectors will always be linearly dependent.

Example 2.3.2 *Linear Independence*

Given the following pair of vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

it is easily shown that $\mathbf{v}_1$ and $\mathbf{v}_2$ are linearly independent. Specifically, note that if we are to find values for the scalars α and β such that

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 = \alpha_1 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

then it follows that

$$\begin{aligned} \alpha_1 + \alpha_2 &= 0 \\ 2\alpha_1 &= 0 \end{aligned}$$

The only solution to these equations is $\alpha_1 = \alpha_2 = 0$. Therefore, the vectors are linearly

independent. However, adding the vector

$$\mathbf{v}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

to the set results in a set of linearly *dependent* vectors since

$$\mathbf{v}_1 = \mathbf{v}_2 + 2\mathbf{v}_3$$

Given a set of N vectors,

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$$

consider the set of all vectors $\mathcal{V}$ that may be formed from a linear combination of the vectors $\mathbf{v}_i$,

$$\mathbf{v} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$$

This set forms a *vector space*, and the vectors $\mathbf{v}_i$ are said to *span* the space $\mathcal{V}$. Furthermore, if the vectors $\mathbf{v}_i$ are linearly independent, then they are said to form a *basis* for the space $\mathcal{V}$, and the number of vectors in the basis, N, is referred to as the *dimension* of the space. For example, the set of all real vectors of the form $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ forms an N-dimensional vector space, denoted by R^N, that is spanned by the basis vectors

$$\begin{aligned} \mathbf{u}_1 &= [1, 0, 0, \ldots, 0]^T \\ \mathbf{u}_2 &= [0, 1, 0, \ldots, 0]^T \\ &\vdots \\ \mathbf{u}_N &= [0, 0, 0, \ldots, 1]^T \end{aligned}$$

In terms of this basis, any vector $\mathbf{v} = [v_1, v_2, \ldots, v_N]^T$ in R^N may be uniquely decomposed as follows

$$\mathbf{v} = \sum_{i=1}^{N} v_i \mathbf{u}_i$$

It should be pointed out, however, that the basis for a vector space is not unique.

2.3.3 Matrices

An $n \times m$ matrix is an array of numbers (real or complex) or functions having n rows and m columns. For example,

$$\mathbf{A} = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \end{bmatrix}$$

is an $n \times m$ matrix of numbers a_{ij} and

$$\mathbf{A}(z) = \{a_{ij}(z)\} = \begin{bmatrix} a_{11}(z) & a_{12}(z) & a_{13}(z) & \cdots & a_{1m}(z) \\ a_{21}(z) & a_{22}(z) & a_{23}(z) & \cdots & a_{2m}(z) \\ a_{31}(z) & a_{32}(z) & a_{33}(z) & \cdots & a_{3m}(z) \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1}(z) & a_{n2}(z) & a_{n3}(z) & \cdots & a_{nm}(z) \end{bmatrix}$$

is an $n \times m$ matrix of functions $a_{ij}(z)$. If $n = m$, then $\mathbf{A}$ is a *square* matrix of n rows and n columns. In some cases, we will have a need to consider matrices having an infinite number of rows or columns. For example, recall that the output of an FIR linear shift-invariant filter with a unit sample response $h(n)$ may be written in vector form as follows

$$y(n) = \mathbf{h}^T\mathbf{x}(n) = \mathbf{x}^T(n)\mathbf{h}$$

If $x(n) = 0$ for $n < 0$, then we may express $y(n)$ for $n \geq 0$ as

$$\mathbf{X}_0\mathbf{h} = \mathbf{y}$$

where $\mathbf{X}_0$ is a *convolution matrix* defined by[5]

$$\mathbf{X}_0 = \begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \\ x(N-1) & x(N-2) & x(N-3) & \cdots & x(0) \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}$$

and $\mathbf{y} = [y(0),\ y(1),\ y(2),\ \ldots]^T$. Note that, in addition to its structure of having equal values along each of the diagonals, $\mathbf{X}_0$ has $N - 1$ columns and an infinite number of rows.

Matrices will be denoted by bold upper case letters, e.g., $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{H}(z)$. On occasion it will be convenient to represent an $n \times m$ matrix $\mathbf{A}$ as a set of m column vectors,

$$\mathbf{A} = [\mathbf{c}_1,\ \mathbf{c}_2, \ldots, \mathbf{c}_m]$$

or a set of n row vectors

$$\mathbf{A} = \begin{bmatrix} \mathbf{r}_1^H \\ \mathbf{r}_2^H \\ \vdots \\ \mathbf{r}_n^H \end{bmatrix}$$

A matrix may also be partitioned into submatrices. For example, an $n \times m$ matrix $\mathbf{A}$ may be partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

where $\mathbf{A}_{11}$ is $p \times q$, $\mathbf{A}_{12}$ is $p \times (m - q)$, $\mathbf{A}_{21}$ is $(n - p) \times q$, and $\mathbf{A}_{22}$ is $(n - p) \times (m - q)$.

[5]The subscript on $\mathbf{X}_0$ is used to indicate the value of the index of $x(n)$ in the first entry of the convolution matrix.

Example 2.3.3 *Partitioning Matrices*

Consider the 3 × 3 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$}} \\ 0 & & \end{array}\right]$$

This matrix may be partitioned as

$$\mathbf{A} = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{0} & \mathbf{A}_{22} \end{array}\right]$$

where $\mathbf{0} = [0, 0]^T$ is a zero vector of length 2 and

$$\mathbf{A}_{22} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

is a 2 × 2 matrix.

If **A** is an $n \times m$ matrix, then the *transpose*, denoted by $\mathbf{A}^T$, is the $m \times n$ matrix that is formed by interchanging the rows and columns of **A**. Thus, the (i, j)th element becomes the (j, i)th element and vice versa. If the matrix is square, then the transpose is formed by simply reflecting the elements of **A** about the diagonal. For a square matrix, if **A** is equal to its transpose,

$$\mathbf{A} = \mathbf{A}^T$$

then **A** is said to be a *symmetric* matrix. For complex matrices, the *Hermitian transpose* is the complex conjugate of the transpose of **A** and is denoted by $\mathbf{A}^H$. Thus,

$$\mathbf{A}^H = (\mathbf{A}^*)^T = (\mathbf{A}^T)^*$$

If a square complex-valued matrix is equal to its Hermitian transpose

$$\mathbf{A} = \mathbf{A}^H$$

then the matrix is said to be *Hermitian*. A few properties of the Hermitian transpose are

1. $(\mathbf{A} + \mathbf{B})^H = \mathbf{A}^H + \mathbf{B}^H$
2. $(\mathbf{A}^H)^H = \mathbf{A}$
3. $(\mathbf{AB})^H = \mathbf{B}^H\mathbf{A}^H$

Equivalent properties for the transpose may be obtained by replacing the Hermitian transpose H with the transpose T.

2.3.4 Matrix Inverse

Let **A** be an $n \times m$ matrix that is partitioned in terms of its m column vectors

$$\mathbf{A} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m]$$

The *rank* of $\mathbf{A}$, $\rho(\mathbf{A})$, is defined to be the number of linearly independent columns in $\mathbf{A}$, i.e., the number of linearly independent vectors in the set $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$. One of the properties of $\rho(\mathbf{A})$ is that the rank of a matrix is equal to the rank of its Hermitian transpose, $\rho(\mathbf{A}) = \rho(\mathbf{A}^H)$. Therefore, if $\mathbf{A}$ is partitioned in terms of its n row vectors,

$$\mathbf{A} = \begin{bmatrix} \mathbf{r}_1^H \\ \mathbf{r}_2^H \\ \vdots \\ \mathbf{r}_n^H \end{bmatrix}$$

then the rank of $\mathbf{A}$ is equivalently equal to the number of linearly independent row vectors, i.e., the number of linearly independent vectors in the set $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$. A useful property of the rank of a matrix is the following:

Property. The rank of $\mathbf{A}$ is equal to the rank of $\mathbf{A}\mathbf{A}^H$ and $\mathbf{A}^H\mathbf{A}$,

$$\rho(\mathbf{A}) = \rho(\mathbf{A}\mathbf{A}^H) = \rho(\mathbf{A}^H\mathbf{A})$$

Since the rank of a matrix is equal to the number of linearly independent rows and the number of linearly independent columns, it follows that if $\mathbf{A}$ is an $m \times n$ matrix then

$$\rho(\mathbf{A}) \leq \min(m, n)$$

If $\mathbf{A}$ is an $m \times n$ matrix and $\rho(\mathbf{A}) = \min(m, n)$ then $\mathbf{A}$ is said to be of *full rank*. If $\mathbf{A}$ is a square matrix of full rank, then there exists a unique matrix $\mathbf{A}^{-1}$, called the inverse of $\mathbf{A}$, such that

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

where

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

is the *identity matrix*, which has ones along the main diagonal and zeros everywhere else. In this case $\mathbf{A}$ is said to be *invertible* or *nonsingular*. If $\mathbf{A}$ is not of full rank, $\rho(\mathbf{A}) < n$, then it is said to be *noninvertible* or *singular* and $\mathbf{A}$ does not have an inverse. Some properties of the matrix inverse are as follows. First, if $\mathbf{A}$ and $\mathbf{B}$ are invertible, then the inverse of their product is

$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

Second, the Hermitian transpose of the inverse is equal to the inverse of the Hermitian transpose

$$(\mathbf{A}^H)^{-1} = (\mathbf{A}^{-1})^H$$

Finally, a formula that will be useful for inverting matrices that arise in adaptive filtering

algorithms is the *Matrix Inversion lemma* [3]

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \tag{2.28}$$

where it is assumed that $\mathbf{A}$ is $n \times n$, $\mathbf{B}$ is $n \times m$, $\mathbf{C}$ is $m \times m$, and $\mathbf{D}$ is $m \times n$ with $\mathbf{A}$ and $\mathbf{C}$ nonsingular matrices. A special case of this lemma occurs when $\mathbf{C} = 1$, $\mathbf{B} = \mathbf{u}$, and $\mathbf{D} = \mathbf{v}^H$ where $\mathbf{u}$ and $\mathbf{v}$ are n-dimensional vectors. In this case the lemma becomes

$$(\mathbf{A}+\mathbf{uv}^H)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{uv}^H\mathbf{A}^{-1}}{1+\mathbf{v}^H\mathbf{A}^{-1}\mathbf{u}} \tag{2.29}$$

which is sometimes referred to as *Woodbury's Identity* [5]. As a special case, note that for $\mathbf{A} = \mathbf{I}$, Eq. (2.29) becomes

$$(\mathbf{I}+\mathbf{uv}^H)^{-1} = \mathbf{I} - \frac{1}{1+\mathbf{v}^H\mathbf{u}}\mathbf{uv}^H \tag{2.30}$$

2.3.5 The Determinant and the Trace

If $\mathbf{A} = a_{11}$ is a 1×1 matrix, then its *determinant* is defined to be $\det(\mathbf{A}) = a_{11}$. The determinant of an $n \times n$ matrix is defined recursively in terms of the determinants of $(n-1) \times (n-1)$ matrices as follows. For any j

$$\det(\mathbf{A}) = \sum_{i=1}^{n}(-1)^{i+j}a_{ij}\det(\mathbf{A}_{ij})$$

where $\mathbf{A}_{ij}$ is the $(n-1) \times (n-1)$ matrix that is formed by deleting the ith row and the jth column of $\mathbf{A}$.

Example 2.3.4 *The Determinant*

For the 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

the determinant is

$$\det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}$$

and for the 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

the determinant is

$$\begin{aligned}\det(\mathbf{A}) &= a_{11}\det\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} - a_{12}\det\begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} + a_{13}\det\begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \\ &= a_{11}[a_{22}a_{33} - a_{23}a_{32}] - a_{12}[a_{21}a_{33} - a_{31}a_{23}] + a_{13}[a_{21}a_{32} - a_{31}a_{22}]\end{aligned}$$

The determinant may be used to determine whether or not a matrix is invertible. Specifically,

> ***Property.*** An $n \times n$ matrix $\mathbf{A}$ is invertible if and only if its determinant is nonzero,
>
> $$\det(\mathbf{A}) \neq 0$$

Some additional properties of the determinant are listed below. It is assumed that $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ matrices.

1. $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$.
2. $\det(\mathbf{A}^T) = \det(\mathbf{A})$
3. $\det(\alpha\mathbf{A}) = \alpha^n \det(\mathbf{A})$, where α is a constant.
4. $\det(\mathbf{A}^{-1}) = \dfrac{1}{\det(\mathbf{A})}$, assuming that $\mathbf{A}$ is invertible.

Another function of a matrix that will occasionally be useful is the *trace*. Given an $n \times n$ matrix, $\mathbf{A}$, the *trace* is the sum of the terms along the diagonal,

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$$

2.3.6 Linear Equations

Many problems addressed in later chapters, such as signal modeling, Wiener filtering, and spectrum estimation, require finding the solution or solutions to a set of linear equations. Many techniques are available for solving linear equations and, depending on the form of the equations, there may be "fast algorithms" for efficiently finding the solution. In addition to solving the linear equations, however, it is often important to characterize the form of the solution in terms of existence and uniqueness. Therefore, this section briefly summarizes the conditions under which a unique solution exists, discusses how constraints might be imposed if multiple solutions exist, and indicates how an approximate solution may be obtained if no solution exists.

Consider the following set of n linear equations in the m unknowns x_i, $i = 1, 2, \ldots, m$,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m &= b_n \end{aligned}$$

These equations may be written concisely in matrix form as follows

$$\mathbf{Ax} = \mathbf{b} \tag{2.31}$$

where $\mathbf{A}$ is an $n \times m$ matrix with entries a_{ij}, $\mathbf{x}$ is an m-dimensional vector containing the unknowns x_i, and $\mathbf{b}$ is an n-dimensional vector with elements b_i. A convenient way to view Eq. (2.31) is as an expansion of the vector $\mathbf{b}$ in terms of a linear combination of the column

vectors $\mathbf{a}_i$ of the matrix $\mathbf{A}$, i.e,

$$\mathbf{b} = \sum_{i=1}^{m} x_i \mathbf{a}_i \tag{2.32}$$

As discussed below, solving Eq. (2.31) depends upon a number of factors including the relative size of m and n, the rank of the matrix $\mathbf{A}$, and the elements in the vector $\mathbf{b}$. The case of a square matrix ($m = n$) is considered first.

Square Matrix: $m = n$. If $\mathbf{A}$ is a square $n \times n$ matrix then the nature of the solution to the linear equations $\mathbf{Ax} = \mathbf{b}$ depends upon whether or not $\mathbf{A}$ is singular. If $\mathbf{A}$ is nonsingular, then the inverse matrix $\mathbf{A}^{-1}$ exists and the solution is uniquely defined by

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

However, if $\mathbf{A}$ is singular, then there may either be no solution (the equations are inconsistent) or many solutions.

Example 2.3.5 *Linear Equations—Singular Case*

Consider the following pair of equations in two unknowns x_1 and x_2,

$$x_1 + x_2 = 1$$
$$x_1 + x_2 = 2$$

In matrix form, these equations are

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Clearly, the matrix $\mathbf{A}$ is singular, $\det(\mathbf{A}) = 0$, and no solution exists. However, if the second equation is modified so that

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

then there are many solutions. Specifically, note that for any constant α, the vector

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

will satisfy these equations.

For the case in which $\mathbf{A}$ is singular, the columns of $\mathbf{A}$ are linearly dependent and there exists nonzero solutions to the homogeneous equations

$$\mathbf{Az} = \mathbf{0} \tag{2.33}$$

In fact, there will be $k = n - \rho(\mathbf{A})$ linearly independent solutions to the homogeneous equations. Therefore, if there is *at least one* vector, $\mathbf{x}_0$, that solves Eq. (2.31), then any vector of the form

$$\mathbf{x} = \mathbf{x}_0 + \alpha_1 \mathbf{z}_1 + \cdots + \alpha_k \mathbf{z}_k$$

will also be a solution where $\mathbf{z}_i$, $i = 1, 2, \ldots, k$ are linearly independent solutions of Eq. (2.33).

Rectangular Matrix: $n < m$. If $n < m$, then there are fewer equations than unknowns. Therefore, provided the equations are not inconsistent, there are many vectors that satisfy the equations, i.e., the solution is *underdetermined* or *incompletely specified*. One approach that is often used to define a unique solution is to find the vector satisfying the equations that has the minimum norm, i.e.,

$$\min \|\mathbf{x}\| \quad \text{such that} \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

If the rank of $\mathbf{A}$ is n (the rows of $\mathbf{A}$ are linearly independent), then the $n \times n$ matrix $\mathbf{A}\mathbf{A}^H$ is invertible and the *minimum norm solution* is [10]

$$\boxed{\mathbf{x}_0 = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b}} \tag{2.34}$$

The matrix

$$\mathbf{A}^+ = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}$$

is known as the *pseudo-inverse* of the matrix $\mathbf{A}$ for the underdetermined problem. The following example illustrates how the pseudoinverse is used to solve an underdetermined set of linear equations.

Example 2.3.6 ***Linear Equations—Underdetermined Case***

Consider the following equation in the four unknowns x_1, x_2, x_3, x_4,

$$x_1 - x_2 + x_3 - x_4 = 1 \tag{2.35}$$

This equation may be written in matrix form as

$$\mathbf{A}\mathbf{x} = [1, \ -1, \ 1, \ -1]\mathbf{x} = \mathbf{b}$$

where $\mathbf{b} = 1$ and $\mathbf{x}$ is the vector containing the unknowns x_i. Clearly, the solution is incompletely specified since there are many solutions that satisfy this equation. However, the minimum norm solution is unique and given by Eq. (2.34). Specifically, since

$$\mathbf{A}\mathbf{A}^H = 4$$

and $(\mathbf{A}\mathbf{A}^H)^{-1} = \frac{1}{4}$ then the minimum norm solution is

$$\mathbf{x} = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

If the following equation is added to Eq. (2.35)

$$x_1 + x_2 + x_3 + x_4 = 1$$

then there are two equations in four unknowns with

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

and $\mathbf{b} = [1, \ 1]^T$. Again, it is easy to see that the solution is incompletely specified. Since

$$\mathbf{A}\mathbf{A}^H = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

then $(\mathbf{A}\mathbf{A}^H)^{-1} = \frac{1}{4}\mathbf{I}$ and the minimum norm solution is

$$\mathbf{x} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b} = \frac{1}{4}\mathbf{A}^H\mathbf{b} = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Rectangular Matrix: $m < n$. If $m < n$, then there are more equations than unknowns and, in general, no solution exists. In this case, the equations are *inconsistent* and the solution is said to be *overdetermined*. The geometry of this problem is illustrated in Fig. 2.5 for the case of three equations in two unknowns. Since an arbitrary vector $\mathbf{b}$ cannot be represented in terms of a linear combination of the columns of $\mathbf{A}$ as given in Eq. (2.32), the goal is to find the coefficients x_i that produce the best approximation to $\mathbf{b}$,

$$\hat{\mathbf{b}} = \sum_{i=1}^{m} x_i \mathbf{a}_i$$

The approach that is commonly used in this situation is to find the *least squares solution*, i.e., the vector $\mathbf{x}$ that minimizes the norm of the error

$$\|\mathbf{e}\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \tag{2.36}$$

As illustrated in Fig. 2.5, the least squares solution has the property that the error,

$$\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{x}$$

is *orthogonal* to each of the vectors that are used in the approximation for $\mathbf{b}$, i.e., the column vectors of $\mathbf{A}$. This orthogonality implies that

$$\mathbf{A}^H\mathbf{e} = 0 \tag{2.37}$$

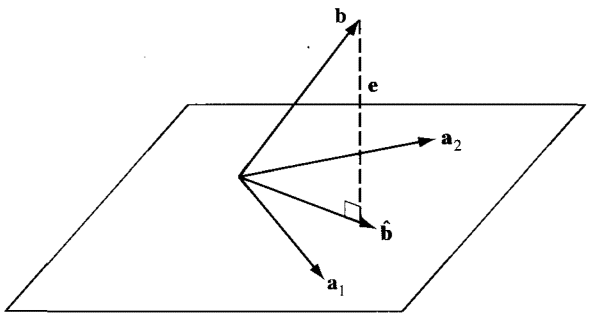


Figure 2.5 *Geometrical illustration of the least squares solution to an overdetermined set of linear equations. The best approximation to* $\mathbf{b}$ *is formed when the error* $\mathbf{e}$ *is orthogonal to the vectors* $\mathbf{a}_1$ *and* $\mathbf{a}_2$.

or,

$$\boxed{\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b}} \tag{2.38}$$

which are known as the *normal equations* [10]. If the columns of $\mathbf{A}$ are linearly independent ($\mathbf{A}$ has full rank), then the matrix $\mathbf{A}^H\mathbf{A}$ is invertible and the *least squares solution* is

$$\boxed{\mathbf{x}_0 = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}} \tag{2.39}$$

or,

$$\mathbf{x}_0 = \mathbf{A}^+\mathbf{b}$$

where the matrix

$$\mathbf{A}^+ = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$$

is the *pseudo-inverse* of the matrix $\mathbf{A}$ for the overdetermined problem. Furthermore, the *best approximation* $\hat{\mathbf{b}}$ to $\mathbf{b}$ is given by the *projection* of the vector $\mathbf{b}$ onto the subspace spanned by the vectors $\mathbf{a}_i$,

$$\boxed{\hat{\mathbf{b}} = \mathbf{A}\mathbf{x}_0 = \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}} \tag{2.40}$$

or

$$\hat{\mathbf{b}} = \mathbf{P}_A\,\mathbf{b}$$

where

$$\boxed{\mathbf{P}_A = \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H} \tag{2.41}$$

is called the *projection matrix*. Finally, expanding the square in Eq. (2.36) and using the *orthogonality condition* given in Eq. (2.37) it follows that the minimum least squares error is

$$\boxed{\min \|\mathbf{e}\|^2 = \mathbf{b}^H\mathbf{e} = \mathbf{b}^H\mathbf{b} - \mathbf{b}^H\mathbf{A}\mathbf{x}_0} \tag{2.42}$$

The following example illustrates the use of the pseudoinverse to solve an overdetermined set of linear equations.

Example 2.3.7 *Linear Equations—Overdetermined Case*

Consider the set of three equations in two unknowns

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Since the rank of $\mathbf{A}$ is two, the least squares solution is unique. With

$$\mathbf{A}^H\mathbf{A} = \begin{bmatrix} 6 & 5 \\ 5 & 6 \end{bmatrix}$$

and

$$(\mathbf{A}^H\mathbf{A})^{-1} = \frac{1}{11}\begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix}$$

the least squares solution is

$$\mathbf{x}_0 = \frac{1}{11}\begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{4}{11}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The error, $\mathbf{e}$, is

$$\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{x}_0 = \frac{1}{11}[-1, -1, 3]^T$$

and the least square error is

$$\|\mathbf{e}\|^2 = \mathbf{b}^H\mathbf{e} = \frac{1}{11}$$

A general treatment of the problem of solving m linear equations in n unknowns may be found in references [6, 10], including the cases for which $\mathbf{A}\mathbf{A}^H$ and $\mathbf{A}^H\mathbf{A}$ are not invertible in the minimum norm and least squares solutions, respectively.

2.3.7 Special Matrix Forms

In this section, we describe some of the special types of matrices and symmetries that will be encountered in the following chapters. The first is a *diagonal matrix*, which is a square matrix that has all of its entries equal to zero except possibly those along the main diagonal. Thus, a diagonal matrix has the form

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{2.43}$$

and may be written concisely as

$$\mathbf{A} = \text{diag}\{a_{11}, a_{22}, \ldots, a_{nn}\}$$

A diagonal matrix with ones along the diagonal is referred to as the *identity matrix* and will be represented by $\mathbf{I}$,

$$\mathbf{I} = \text{diag}\{1, 1, \ldots, 1\} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

If the entries along the diagonal in Eq. (2.43) are replaced with matrices,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{22} & \cdots & 0 \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & \mathbf{A}_{kk} \end{bmatrix}$$

then $\mathbf{A}$ is said to be a *block diagonal matrix*. The matrix in Example 2.3.3 is an example of a 3×3 block diagonal matrix.

Another matrix that will be useful is the *exchange matrix*

$$\mathbf{J} = \begin{bmatrix} 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 1 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

which is symmetric and has ones along the *cross-diagonal* and zeros everywhere else. Note that since $\mathbf{J}^2 = \mathbf{I}$ then $\mathbf{J}$ is its own inverse. The effect of multiplying a vector $\mathbf{v}$ by the exchange matrix is to reverse the order of the entries, i.e.,

$$\mathbf{J}\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_n \\ v_{n-1} \\ \vdots \\ v_1 \end{bmatrix}$$

Similarly, if a matrix $\mathbf{A}$ is multiplied on the left by the exchange matrix, the effect is to reverse the order of each column. For example, if

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

then

$$\mathbf{J}^T\mathbf{A} = \begin{bmatrix} a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$$

Similarly, if $\mathbf{A}$ is multiplied on the right by $\mathbf{J}$, then the order of the entries in each row is reversed,

$$\mathbf{AJ} = \begin{bmatrix} a_{13} & a_{12} & a_{11} \\ a_{23} & a_{22} & a_{21} \\ a_{33} & a_{32} & a_{31} \end{bmatrix}$$

Finally, the effect of forming the product $\mathbf{J}^T\mathbf{AJ}$ is to reverse the order of each row and column,

$$\mathbf{J}^T\mathbf{AJ} = \begin{bmatrix} a_{33} & a_{32} & a_{31} \\ a_{23} & a_{22} & a_{21} \\ a_{13} & a_{12} & a_{11} \end{bmatrix}$$

thereby *reflecting* each element of $\mathbf{A}$ about the central element.

An *upper triangular* matrix is a square matrix in which all of the terms below the diagonal are equal to zero, i.e., with $\mathbf{A} = \{a_{ij}\}$ then $a_{ij} = 0$ for $i > j$. The following is an example of a 4 × 4 upper triangular matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

A *lower triangular* matrix, on the other hand, is a square matrix in which all of the entries above the diagonal are equal to zero, i.e., $a_{ij} = 0$ for $i < j$. Clearly, the transpose of a lower triangular matrix is upper triangular and vice versa. Some additional properties of lower and upper triangular matrices are as follows.

1. The determinant of a lower triangular matrix or an upper triangular matrix is equal to the product of the terms along the diagonal

$$\det(\mathbf{A}) = \prod_{i=1}^{n} a_{ii}$$

2. The inverse of an upper (lower) triangular matrix is upper (lower) triangular.
3. The product of two upper (lower) triangular matrices is upper (lower) triangular.

A matrix having a tremendous amount of structure that will play a prominent role in many of our discussions throughout this book is one that is referred to as a *Toeplitz* matrix. An $n \times n$ matrix $\mathbf{A}$ is said to be Toeplitz if all of the elements along each of the diagonals have the same value, i.e.,

$$a_{ij} = a_{i+1,j+1} \quad ; \quad \text{for all } i < n \text{ and } j < n$$

An example of a 4 × 4 Toeplitz matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 1 & 3 & 5 \\ 4 & 2 & 1 & 3 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

Note that all of the entries in a Toeplitz matrix are completely defined once the first column and the first row have been specified. A convolution matrix is an example of a Toeplitz matrix. A matrix with a similar property is a *Hankel matrix*, which has equal elements along the diagonals that are perpendicular to the main diagonal, i.e.,

$$a_{ij} = a_{i+1,j-1} \quad ; \quad \text{for all } i < n \text{ and } j \leq n$$

An example of a 4 × 4 Hankel matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 7 & 4 \\ 5 & 7 & 4 & 2 \\ 7 & 4 & 2 & 1 \end{bmatrix}$$

Another example of a Hankel matrix is the exchange matrix $\mathbf{J}$.

Toeplitz matrices are a special case of a larger class of matrices known as *persymmetric* matrices. A persymmetric matrix is symmetric about the cross-diagonal, $a_{ij} = a_{n-j+1,n-i+1}$.

An example of a persymmetric but non-Toeplitz matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 2 & 4 & 5 \\ 4 & 4 & 2 & 3 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

If a Toeplitz matrix is symmetric, or Hermitian in the case of a complex matrix, then all of the elements of the matrix are completely determined by either the first row or the first column of the matrix. An example of a symmetric Toeplitz matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 3 & 1 & 3 & 5 \\ 5 & 3 & 1 & 3 \\ 7 & 5 & 3 & 1 \end{bmatrix}$$

Since we will be dealing frequently with symmetric Toeplitz and Hermitian Toeplitz matrices it will be convenient to adopt the notation

$$\mathbf{A} = \text{Toep}\{a(0),\ a(1), \ldots, a(p)\}$$

for the Hermitian Toeplitz matrix that has the elements $a(0),\ a(1), \ldots, a(p)$ in the first *column*. For example,

$$\mathbf{A} = \text{Toep}\{1,\ j,\ 1-j\} = \begin{bmatrix} 1 & -j & 1+j \\ j & 1 & -j \\ 1-j & j & 1 \end{bmatrix}$$

Symmetric Toeplitz matrices are a special case of a larger class of matrices known as *centrosymmetric* matrices. A centrosymmetric matrix is both symmetric and persymmetric. An example of a centrosymmetric matrix that is not Toeplitz is

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & 6 \\ 3 & 2 & 4 & 5 \\ 5 & 4 & 2 & 3 \\ 6 & 5 & 3 & 1 \end{bmatrix}$$

There are many interesting and useful properties of Toeplitz, persymmetric, and centrosymmetric matrices. For example, if $\mathbf{A}$ is a symmetric Toeplitz matrix, then

$$\mathbf{J}^T\mathbf{A}\mathbf{J} = \mathbf{A}$$

whereas if $\mathbf{A}$ is a Hermitian Toeplitz matrix, then

$$\mathbf{J}^T\mathbf{A}\mathbf{J} = \mathbf{A}^*$$

Of particular interest will be the following properties that are concerned with the inverse.

- **Property 1.** The inverse of a symmetric matrix is symmetric.
- **Property 2.** The inverse of a persymmetric matrix is persymmetric.
- **Property 3.** The inverse of a Toeplitz matrix is not, in general, Toeplitz. However, since a Toeplitz matrix is persymmetric, the inverse will always be persymmetric. Furthermore, the inverse of a symmetric Toeplitz matrix will be centrosymmetric.

These symmetry properties of matrix inverses along with those previously mentioned are summarized in Table 2.5.

Table 2.5 ***The Relationship between the Symmetries of a Matrix and Its Inverse***

Matrix	Inverse
Symmetric	Symmetric
Hermitian	Hermitian
Persymmetric	Persymmetric
Centrosymmetric	Centrosymmetric
Toeplitz	Persymmetric
Hankel	Symmetric
Triangular	Triangular

Finally, we conclude with the definition of orthogonal and unitary matrices. A real $n \times n$ matrix is said to be *orthogonal* if the columns (and rows) are orthonormal. Thus, if the columns of $\mathbf{A}$ are $\mathbf{a}_i$,

$$\mathbf{A} = \left[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\right]$$

and

$$\mathbf{a}_i^T \mathbf{a}_j = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

then $\mathbf{A}$ is orthogonal. Note that if $\mathbf{A}$ is orthogonal, then

$$\mathbf{A}^T\mathbf{A} = \mathbf{I}$$

Therefore, the inverse of an orthogonal matrix is equal to its transpose

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

The exchange matrix is an example of an orthogonal matrix since $\mathbf{J}^T\mathbf{J} = \mathbf{J}^2 = \mathbf{I}$.

In the case of a complex $n \times n$ matrix, if the columns (rows) are orthogonal,

$$\mathbf{a}_i^H \mathbf{a}_j = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

then

$$\mathbf{A}^H\mathbf{A} = \mathbf{I}$$

and $\mathbf{A}$ is said to be a *unitary* matrix. The inverse of a unitary matrix is equal to its Hermitian transpose

$$\mathbf{A}^{-1} = \mathbf{A}^H$$

2.3.8 Quadratic and Hermitian Forms

The *quadratic form* of a real symmetric $n \times n$ matrix $\mathbf{A}$ is the scalar defined by

$$Q_A(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} x_j$$

where $\mathbf{x}^T = [x_1, x_2, \ldots, x_n]$ is a vector of n real variables. Note that the quadratic form is a quadratic function in the n variables $x_1, x_2, \ldots, x_n$. For example, the quadratic form of

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

is

$$Q_A(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} = 3x_1^2 + 2x_1x_2 + 2x_2^2$$

In a similar fashion, for a Hermitian matrix, the Hermitian form is defined as

$$Q_A(\mathbf{x}) = \mathbf{x}^H\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i^* a_{ij} x_j$$

If the quadratic form of a matrix $\mathbf{A}$ is positive for all nonzero vectors $\mathbf{x}$,

$$Q_A(\mathbf{x}) > 0$$

then $\mathbf{A}$ is said to be *positive definite* and we write $\mathbf{A} > 0$. For example, the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

which has the quadratic form

$$Q_A(\mathbf{x}) = [x_1, x_2]\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2x_1^2 + 3x_2^2$$

is positive definite since $Q_A(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

If the quadratic form is nonnegative for all nonzero vectors $\mathbf{x}$

$$Q_A(\mathbf{x}) \geq 0$$

then $\mathbf{A}$ is said to be *positive semidefinite*. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

is positive semidefinite since $Q_A(x) = 2x_1^2 \geq 0$, but it is not positive definite since $Q_A(\mathbf{x}) = 0$ for any vector $\mathbf{x}$ of the form $\mathbf{x} = [0, x_2]^T$. A test for positive definiteness is given in Section 2.3.9 (see Property 3 on p. 43).

In a similar fashion, a matrix is said to be *negative definite* if $Q_A(\mathbf{x}) < 0$ for all nonzero $\mathbf{x}$, whereas it is said to be *negative semidefinite* if $Q_A(\mathbf{x}) \leq 0$ for all nonzero $\mathbf{x}$. A matrix that is none of the above is said to be *indefinite*.

For any $n \times n$ matrix $\mathbf{A}$ and for any $n \times m$ matrix $\mathbf{B}$ having full rank, the definiteness of $\mathbf{A}$ and $\mathbf{B}^H\mathbf{A}\mathbf{B}$ will be the same. For example, if $\mathbf{A} > 0$ and $\mathbf{B}$ is full rank, then $\mathbf{B}^H\mathbf{A}\mathbf{B} > 0$. This follows by noting that for any vector $\mathbf{x}$,

$$\mathbf{x}^H\left(\mathbf{B}^H\mathbf{A}\mathbf{B}\right)\mathbf{x} = (\mathbf{B}\mathbf{x})^H\,\mathbf{A}\,(\mathbf{B}\mathbf{x}) = \mathbf{v}^H\mathbf{A}\mathbf{v}$$

where $\mathbf{v} = \mathbf{B}\mathbf{x}$. Therefore, if $\mathbf{A} > 0$, then $\mathbf{v}^H\mathbf{A}\mathbf{v} > 0$ and $\mathbf{B}^H\mathbf{A}\mathbf{B}$ is positive definite (The constraint that $\mathbf{B}$ have full rank ensures that $\mathbf{v} = \mathbf{B}\mathbf{x}$ is nonzero for any nonzero vector $\mathbf{x}$).

2.3.9 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors provide useful and important information about a matrix. For example, given the eigenvalues it is possible to determine whether or not a matrix is positive definite. The eigenvalues may also be used to determine whether or not a matrix

is invertible as well as to indicate how sensitive the determination of the inverse will be to numerical errors. Eigenvalues and eigenvectors also provide an important representation for matrices known as the *eigenvalue decomposition*. This decomposition, developed below, will be useful in our discussions of spectrum estimation as well as in our study of the convergence properties of adaptive filtering algorithms. This section begins with the definition of the eigenvalues and eigenvectors of a matrix and then proceeds to review some properties of eigenvalues and eigenvectors that will be useful in later chapters.

Let $\mathbf{A}$ be an $n \times n$ matrix and consider the following set of linear equations,

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{2.44}$$

where λ is a constant. Equation (2.44) may equivalently be expressed as a set of homogeneous linear equations of the form

$$(\mathbf{A} - \lambda\mathbf{I})\,\mathbf{v} = 0 \tag{2.45}$$

In order for a nonzero vector to be a solution to this equation, it is necessary for the matrix $\mathbf{A} - \lambda\mathbf{I}$ to be singular. Therefore, the determinant of $\mathbf{A} - \lambda\mathbf{I}$ must be zero,

$$p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0 \tag{2.46}$$

Note that $p(\lambda)$ is an nth-order polynomial in λ. This polynomial is called the characteristic polynomial of the matrix $\mathbf{A}$ and the n roots, λ_i for $i = 1, 2, \ldots, n$, are the *eigenvalues* of $\mathbf{A}$. For each eigenvalue, λ_i, the matrix $(\mathbf{A} - \lambda_i\mathbf{I})$ will be singular and there will be at least one nonzero vector, $\mathbf{v}_i$, that solves Eq. (2.44), i.e.,

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i \tag{2.47}$$

These vectors, $\mathbf{v}_i$, are called the *eigenvectors* of $\mathbf{A}$. For any eigenvector $\mathbf{v}_i$, it is clear that $\alpha\mathbf{v}_i$ will also be an eigenvector for any constant α. Therefore, eigenvectors are often normalized so that they have unit norm, $\|\mathbf{v}_i\| = 1$. The following property establishes the linear independence of eigenvectors that correspond to distinct eigenvalues.

> ***Property 1.*** The nonzero eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ corresponding to distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ are linearly independent.

If $\mathbf{A}$ is an $n \times n$ singular matrix, then there are nonzero solutions to the homogeneous equation

$$\mathbf{A}\mathbf{v}_i = \mathbf{0} \tag{2.48}$$

and it follows that $\lambda = 0$ is an eigenvalue of $\mathbf{A}$. Furthermore, if the rank of $\mathbf{A}$ is $\rho(\mathbf{A})$, then there will be $k = n - \rho(\mathbf{A})$ linearly independent solutions to Eq. (2.48). Therefore, it follows that $\mathbf{A}$ will have $\rho(\mathbf{A})$ nonzero eigenvalues and $n - \rho(\mathbf{A})$ eigenvalues that are equal to zero.

Example 2.3.8 ***Eigenvalues of a 2 x 2 Symmetric Matrix***

Consider the following 2×2 symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$$

The characteristic polynomial is

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \det\begin{bmatrix} 4-\lambda & 1 \\ 1 & 4-\lambda \end{bmatrix} = (4-\lambda)(4-\lambda) - 1 = \lambda^2 - 8\lambda + 15$$

Therefore, the eigenvalues of $\mathbf{A}$ are the roots of

$$\lambda^2 - 8\lambda + 15 = 0$$

which are

$$\lambda_1 = 5 \qquad \lambda_2 = 3$$

The eigenvectors are found by solving Eq. (2.47) with $\lambda_1 = 5$ and $\lambda_2 = 3$. Thus, for $\lambda_1 = 5$, Eq. (2.47) becomes

$$\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = 5\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix}$$

and it follows that

$$4v_{11} + v_{12} = 5v_{11}$$

or

$$v_{11} = v_{12}$$

Similarly, for $\lambda_2 = 3$ we find that

$$v_{11} = -v_{12}$$

Thus, the normalized eigenvectors of $\mathbf{A}$ are

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

In general, it is not possible to say much about the eigenvalues and eigenvectors of a matrix without knowing something about its properties. However, in the case of symmetric or Hermitian matrices, the eigenvalues and eigenvectors have several useful and important properties. For example,

Property 2. The eigenvalues of a Hermitian matrix are real.

This property is easily established as follows. Let $\mathbf{A}$ be a Hermitian matrix with eigenvalue λ_i and eigenvector $\mathbf{v}_i$,

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i$$

Multiplying this equation on the left by $\mathbf{v}_i^H$

$$\mathbf{v}_i^H\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i^H\mathbf{v}_i \tag{2.49}$$

and taking the Hermitian transpose gives

$$\mathbf{v}_i^H\mathbf{A}^H\mathbf{v}_i = \lambda_i^*\mathbf{v}_i^H\mathbf{v}_i \tag{2.50}$$

Since $\mathbf{A}$ is Hermitian, then Eq. (2.50) becomes

$$\mathbf{v}_i^H \mathbf{A} \mathbf{v}_i = \lambda_i^* \mathbf{v}_i^H \mathbf{v}_i \tag{2.51}$$

Comparing Eq. (2.49) with Eq. (2.51) it follows that λ_i must be real, i.e., $\lambda_i = \lambda_i^*$. ■

The next property establishes a fundamental result linking the positive definiteness of a matrix to the positivity of its eigenvalues.

Property 3. A Hermitian matrix is positive definite, $\mathbf{A} > 0$, if and only if the eigenvalues of $\mathbf{A}$ are positive, $\lambda_k > 0$.

Similar properties hold for positive semidefinite, negative definite, and negative semidefinite matrices. For example, if $\mathbf{A}$ is Hermitian and $\mathbf{A} \geq 0$ then $\lambda_k \geq 0$.

The determinant of a matrix is related to its eigenvalues by the following relationship,

$$\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i \tag{2.52}$$

Therefore, a matrix is nonsingular (invertible) if and only if all of its eigenvalues are nonzero. Conversely, if a matrix has one or more zero eigenvalues then it will be singular (noninvertible). As a result, it follows that any positive definite matrix is nonsingular.

In Property 2 we saw that the eigenvalues of a Hermitian matrix are real. The next property establishes the orthogonality of the eigenvectors corresponding to distinct eigenvalues.

Property 4. The eigenvectors of a Hermitian matrix corresponding to distinct eigenvalues are orthogonal, i.e., if $\lambda_i \neq \lambda_j$, then $< \mathbf{v}_i, \mathbf{v}_j >= 0$.

To establish this property, let λ_i and λ_j be two distinct eigenvalues of a Hermitian matrix with eigenvectors $\mathbf{v}_i$ and $\mathbf{v}_j$, respectively,

$$\begin{aligned} \mathbf{A}\mathbf{v}_i &= \lambda_i \mathbf{v}_i \\ \mathbf{A}\mathbf{v}_j &= \lambda_j \mathbf{v}_j \end{aligned} \tag{2.53}$$

Multiplying the first equation on the left by $\mathbf{v}_j^H$ and the second by $\mathbf{v}_i^H$ gives

$$\mathbf{v}_j^H \mathbf{A} \mathbf{v}_i = \lambda_i \mathbf{v}_j^H \mathbf{v}_i \tag{2.54}$$

$$\mathbf{v}_i^H \mathbf{A} \mathbf{v}_j = \lambda_j \mathbf{v}_i^H \mathbf{v}_j \tag{2.55}$$

Taking the Hermitian transpose of Eq. (2.55) yields

$$\mathbf{v}_j^H \mathbf{A}^H \mathbf{v}_i = \lambda_j^* \mathbf{v}_j^H \mathbf{v}_i \tag{2.56}$$

If $\mathbf{A}$ is Hermitian, then $\mathbf{A}^H = \mathbf{A}$ and it follows from Property 2 that $\lambda_j^* = \lambda_j$. Therefore, Eq. (2.56) becomes

$$\mathbf{v}_j^H \mathbf{A} \mathbf{v}_i = \lambda_j \mathbf{v}_j^H \mathbf{v}_i \tag{2.57}$$

Subtracting Eq. (2.57) from (2.54) leads to

$$(\lambda_i - \lambda_j) \mathbf{v}_j^H \mathbf{v}_i = 0$$

Therefore, if $\lambda_i \neq \lambda_j$, then $\mathbf{v}_j^H \mathbf{v}_i = 0$ and it follows that $\mathbf{v}_i$ and $\mathbf{v}_j$ are orthogonal. ■

Although stated and verified only for the case of distinct eigenvalues, it is also true that for any $n \times n$ Hermitian matrix there exists a set of n orthonormal eigenvectors [10]. For example, consider the 2×2 identity matrix,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

which has two eigenvalues that are equal to one. Since any vector is an eigenvector of $\mathbf{I}$, then $\mathbf{v}_1 = [1, 0]^T$ and $\mathbf{v}_2 = [0, 1]^T$ is one possible set of orthonormal eigenvectors.

For any $n \times n$ matrix $\mathbf{A}$ having a set of n linearly independent eigenvectors we may perform an *eigenvalue decomposition* that expresses $\mathbf{A}$ in the form

$$\boxed{\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}} \tag{2.58}$$

where $\mathbf{V}$ is a matrix that contains the eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda}$ is a diagonal matrix that contains the eigenvalues. This decomposition is performed as follows. Let $\mathbf{A}$ be an $n \times n$ matrix with eigenvalues λ_k and eigenvectors $\mathbf{v}_k$,

$$\mathbf{A}\mathbf{v}_k = \lambda_k \mathbf{v}_k \quad ; \quad k = 1, 2, \ldots, n$$

These n equations may be written in matrix form as

$$\mathbf{A}[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] = [\lambda_1 \mathbf{v}_1, \lambda_2 \mathbf{v}_2, \ldots, \lambda_n \mathbf{v}_n] \tag{2.59}$$

Therefore, with

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$$

and

$$\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$$

Eq. (2.59) may be written as

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{2.60}$$

If the eigenvectors $\mathbf{v}_i$ are independent, then $\mathbf{V}$ is invertible and the decomposition follows by multiplying both sides of Eq. (2.60) on the right by $\mathbf{V}^{-1}$.

For a Hermitian matrix, the harmonic decomposition assumes a special form. In particular, recall that for a Hermitian matrix we may always find an orthonormal set of eigenvectors. Therefore, if $\mathbf{A}$ is Hermitian, then $\mathbf{V}$ is unitary and the eigenvalue decomposition becomes

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^H \tag{2.61}$$

This result is known as the *spectral theorem*.

> ***Spectral Theorem.*** Any Hermitian matrix $\mathbf{A}$ may be decomposed as
>
> $$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^H + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^H + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^H$$
>
> where λ_i are the eigenvalues of $\mathbf{A}$ and $\mathbf{v}_i$ are a set of orthonormal eigenvectors.

As an application of the spectral theorem, suppose that $\mathbf{A}$ is a nonsingular Hermitian matrix. Using the spectral theorem we may find the inverse of $\mathbf{A}$ as follows,

$$\mathbf{A}^{-1} = \left(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^H\right)^{-1} = \left(\mathbf{V}^H\right)^{-1}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^H = \sum_{i=1}^{n}\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^H$$

Note that the invertibility of $\mathbf{A}$ guarantees that $\lambda_i \neq 0$ so this sum is always well defined.

In many signal processing applications one finds that a matrix $\mathbf{B}$ is singular or ill conditioned (one or more eigenvalues are close to zero). In these cases, we may sometimes stabilize the problem by adding a constant to each term along the diagonal,

$$\mathbf{A} = \mathbf{B} + \alpha\mathbf{I}$$

The effect of this operation is to leave the eigenvectors of $\mathbf{B}$ unchanged while changing the eigenvalues of $\mathbf{B}$ from λ_k to $\lambda_k + \alpha$. To see this, note that if $\mathbf{v}_k$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ_k, then

$$\mathbf{A}\mathbf{v}_k = \mathbf{B}\mathbf{v}_k + \alpha\mathbf{v}_k = \left(\lambda_k + \alpha\right)\mathbf{v}_k$$

Therefore, $\mathbf{v}_k$ is also an eigenvector of $\mathbf{A}$ with eigenvalue $\lambda_k + \alpha$. This result is summarized in the following property for future reference.

Property 5. Let $\mathbf{B}$ be an $n \times n$ matrix with eigenvalues λ_i and let $\mathbf{A}$ be a matrix that is related to $\mathbf{B}$ as follows

$$\mathbf{A} = \mathbf{B} + \alpha\mathbf{I}$$

Then $\mathbf{A}$ and $\mathbf{B}$ have the same eigenvectors and the eigenvalues of $\mathbf{A}$ are $\lambda_i + \alpha$.

The following example considers finding the inverse of a matrix of the form $\mathbf{A} = \mathbf{B} + \alpha\mathbf{I}$ where $\mathbf{B}$ is a noninvertible Hermitian matrix of rank one.

Example 2.3.9 ***Using the Spectral Theorem to Invert a Matrix***

Let $\mathbf{A}$ be an $n \times n$ Hermitian matrix of the form

$$\mathbf{A} = \mathbf{B} + \alpha\mathbf{I}$$

where α is a nonzero constant and where

$$\mathbf{B} = \lambda\mathbf{u}_1\mathbf{u}_1^H$$

with $\mathbf{u}_1$ a unit norm vector, $\mathbf{u}_1^H\mathbf{u}_1 = 1$, and $\lambda \neq 0$.[6] Since $\mathbf{B}$ is an $n \times n$ matrix of rank one, it has only one nonzero eigenvalue, the rest are equal to zero. Since

$$\mathbf{B}\mathbf{u}_1 = \lambda\left(\mathbf{u}_1\mathbf{u}_1^H\right)\mathbf{u}_1 = \lambda\mathbf{u}_1\left(\mathbf{u}_1^H\mathbf{u}_1\right) = \lambda\mathbf{u}_1$$

we see that $\mathbf{u}_1$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ. This vector is also an eigenvector of $\mathbf{A}$ with eigenvalue $\lambda + \alpha$. Since $\mathbf{A}$ is a Hermitian matrix, there exists an orthonormal set of eigenvectors, call them $\mathbf{v}_i$. One of these eigenvectors is $\mathbf{u}_1$, so we will set $\mathbf{v}_1 = \mathbf{u}_1$. The eigenvalues corresponding to the eigenvectors $\mathbf{v}_i$ for $i \geq 2$ are equal to α. Using the

[6] We will encounter matrices of this form in Chapter 8 when we consider the autocorrelation matrix of a random process consisting of a complex exponential in white noise.

spectral theorem, we find for the inverse of **A**,

$$\mathbf{A}^{-1} = \frac{1}{\alpha + \lambda}\mathbf{u}_1\mathbf{u}_1^H + \sum_{i=2}^{n}\frac{1}{\alpha}\mathbf{v}_i\mathbf{v}_i^H \tag{2.62}$$

Since the n orthonormal eigenvectors of **A** also form an orthonormal set of eigenvectors for **I**, then the spectral theorem applied to **I** gives

$$\mathbf{I} = \sum_{i=1}^{n}\mathbf{v}_i\mathbf{v}_i^H$$

Therefore, the second term in Eq. (2.62) may be written as

$$\sum_{i=2}^{n}\frac{1}{\alpha}\mathbf{v}_i\mathbf{v}_i^H = \frac{1}{\alpha}\mathbf{I} - \frac{1}{\alpha}\mathbf{u}_1\mathbf{u}_1^H$$

and the inverse of **A** becomes

$$\mathbf{A}^{-1} = \frac{1}{\alpha + \lambda}\mathbf{u}_1\mathbf{u}_1^H + \frac{1}{\alpha}\mathbf{I} - \frac{1}{\alpha}\mathbf{u}_1\mathbf{u}_1^H = \frac{1}{\alpha}\mathbf{I} - \frac{\lambda}{\alpha(\alpha + \lambda)}\mathbf{u}_1\mathbf{u}_1^H$$

which is the desired solution.

It will be useful in our development of adaptive filtering algorithms in Chapter 9 to be able to characterize the geometry of the surface described by the equation

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = 1 \tag{2.63}$$

where **A** is symmetric and positive definite. Using the spectral theorem this becomes

$$\mathbf{x}^T(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T)\mathbf{x} = 1 \tag{2.64}$$

With the change of variables

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T\mathbf{x} \\ \mathbf{v}_2^T\mathbf{x} \\ \vdots \\ \mathbf{v}_n^T\mathbf{x} \end{bmatrix} = \mathbf{V}^T\mathbf{x} \tag{2.65}$$

Eq. (2.64) simplifies to

$$\mathbf{y}^T\mathbf{\Lambda}\mathbf{y} = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2 = 1 \tag{2.66}$$

which is an equation for an ellipse in n dimensions that is centered at the origin. Note that Eq. (2.65) performs a rotation of the coordinate system with the new axes being aligned with the eigenvectors of **A**. Assuming that the eigenvalues have been ordered so that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ we see that $\mathbf{y} = [1/\sqrt{\lambda_1}, 0, \ldots, 0]^T$ is the point on the ellipse that is the farthest from the origin, lying at the end of the major axis of the ellipse (see Fig. 2.6). Furthermore, this point lies along the direction of the eigenvector $\mathbf{v}_1$, i.e.,

$$\mathbf{x} = \frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1$$

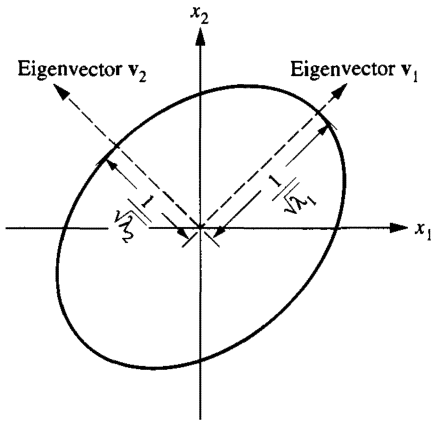


Figure 2.6 *The locus of points described by the equation* $\mathbf{x}^T\mathbf{A}\mathbf{x} = 1$ *where* $\mathbf{A}$ *is a positive definite symmetric matrix with eigenvalues* λ_1 *and* λ_2 *and eigenvectors* $\mathbf{v}_1$ *and* $\mathbf{v}_2$.

Similarly, the point closest to the origin is located at

$$\mathbf{x} = \frac{1}{\sqrt{\lambda_n}}\mathbf{v}_n$$

which is at the end of the minor axis of the ellipse (recall that since $\mathbf{A}$ is symmetric, then the eigenvectors $\mathbf{v}_i$ form an orthonormal set and, since $\mathbf{A} > 0$, then the eigenvalues λ_i are positive). The intermediate eigenvectors correspond to the intermediate axes. Thus, we have the following property of the surface described by Eq. (2.63).

Property 6. For a symmetric positive definite matrix $\mathbf{A}$, the equation

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = 1$$

defines an ellipse in n dimensions whose axes are in the direction of the eigenvectors $\mathbf{v}_j$ of $\mathbf{A}$ with the half-length of these axes equal to $1/\sqrt{\lambda_j}$.

In our treatment of adaptive filtering algorithms in Chapter 9, it will be useful to be able to place an upper bound on the largest eigenvalue of a matrix. A loose upper bound follows from the following property of the eigenvalues of a matrix,

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i$$

Denoting the maximum eigenvalue of a matrix $\mathbf{A}$ by $\lambda_{\max}$ we have the following upper bound

$$\lambda_{\max} \leq \sum_{i=1}^{n} \lambda_i = \text{tr}(\mathbf{A})$$

A tighter bound that follows from *Gershgorin's Circle theorem* [3, 10] is the following

Property 7. The largest eigenvalue of an $n \times n$ matrix $\mathbf{A} = \{a_{ij}\}$ is upper bounded by

$$\lambda_{\max} \leq \max_i \sum_{j=1}^{n} a_{ij}$$

Thus, the maximum eigenvalue is upper bounded by the maximum *row sum* of the matrix **A**. Since the eigenvalues of a matrix and its transpose are the same, a similar statement may be made for *column sums*. Clearly, the upper bound will be satisfied with equality for any diagonal matrix.

Finally, we conclude with a truly remarkable result known as the *Bordering theorem*, which is an interlacing theorem for the eigenvalues of a matrix [3].

Bordering Theorem. Let **A** be an $n \times n$ Hermitian matrix with ordered eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Let **y** be an arbitrary complex vector and a an arbitrary complex number. If $\widetilde{\mathbf{A}}$ is the $(n+1) \times (n+1)$ Hermitian matrix formed from **A**, **y**, and a as follows,

$$\widetilde{\mathbf{A}} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{y} \\ \hline \mathbf{y}^H & a \end{array}\right]$$

then the ordered eigenvalues of $\widetilde{\mathbf{A}}$, denoted by $\widetilde{\lambda}_1 \leq \widetilde{\lambda}_2 \leq \cdots \leq \widetilde{\lambda}_{n+1}$ are interlaced with those of **A** as follows

$$\widetilde{\lambda}_1 \leq \lambda_1 \leq \widetilde{\lambda}_2 \leq \lambda_2 \leq \cdots \leq \widetilde{\lambda}_n \leq \lambda_n \leq \widetilde{\lambda}_{n+1}$$

As a consequence of this theorem we see that, as a matrix is enlarged, the maximum eigenvalue does not decrease and the minimum eigenvalue does not increase. This result is useful in describing how the eigenvalues of an autocorrelation matrix vary as the dimension is increased.

2.3.10 Optimization Theory

Many of the problems that we will be formulating in the following chapters will involve the minimization of a function of one or more variables. Such problems fall in the domain of *optimization theory* [9]. The simplest form of optimization problem is that of finding the minimum of a scalar function $f(x)$ of a single real variable x. Assuming that this *objective function* $f(x)$ is differentiable, the stationary points of $f(x)$, the local and global minima, must satisfy the following conditions,

$$\frac{d\,f(x)}{dx} = 0 \tag{2.67}$$

$$\frac{d^2\,f(x)}{dx^2} > 0 \tag{2.68}$$

If $f(x)$ is a strictly convex function[7] then there is only one solution to Eq. (2.67) and this solution corresponds to the *global minimum* of $f(x)$. If the objective function is not convex, then each stationary point must be checked to see if it is the global minimum.

When the objective function depends upon a complex variable z and is differentiable (analytic), finding the stationary points of $f(z)$ is the same as in the real case. Unfortunately, however, many of the functions that we want to minimize are not differentiable. For example, consider the simple function $f(z) = |z|^2$. Although it is clear that $f(z)$ has a unique minimum that occurs at $z = 0$, this function is not differentiable [3]. The problem is that $f(z) = |z|^2$ is a function of z and z^* and any function that depends upon the complex conjugate z^* is not be differentiable with respect to z. There are two ways that we may get around this problem [4]. The first is to express $f(z)$ in terms of its real and imaginary parts, $z = x + jy$, and minimize $f(x, y)$ with respect to these two variables. A more elegant solution is to treat z and z^* as independent variables and minimize $f(z, z^*)$ with respect to both z and z^* [1]. For example, if we write $f(z) = |z|^2$ as $f(z, z^*) = zz^*$ and treat $f(z, z^*)$ as a function of z with z^* constant, then

$$\frac{d}{dz}|z|^2 = z^* \tag{2.69}$$

whereas if we treat $f(z, z^*)$ as a function of z^* with z constant, then we have

$$\frac{d}{dz^*}|z|^2 = z \tag{2.70}$$

Setting both derivatives equal to zero and solving this pair of equations we see that the solution is $z = 0$. Minimizing a real-valued function of z and z^* is actually a bit easier than this as stated in the following theorem [1].

> ***Theorem.*** If $f(z, z^*)$ is a *real-valued* function of z and z^* and if $f(z, z^*)$ is analytic with respect to both z and z^*, then the stationary points of $f(z, z^*)$ may be found by setting the derivative of $f(z, z^*)$ with respect to either z or z^* equal to zero and solving for z.

Thus, for $f(z, z^*) = |z|^2$, it is sufficient to set either Eq. (2.69) or (2.70) equal to zero and solve for z.

Another problem that we will be concerned with in this book is finding the minimum of a function of two or more variables. For a scalar function of n real variables, $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$, this involves computing the gradient, which is the vector of partial derivatives defined by

$$\nabla_x f(\mathbf{x}) = \frac{d}{d\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \dfrac{\partial}{\partial x_n} f(\mathbf{x}) \end{bmatrix}$$

The gradient points in the direction of the maximum rate of change of $f(\mathbf{x})$ and is equal to zero at the stationary points of $f(\mathbf{x})$. Thus, a necessary condition for a point $\mathbf{x}$ to be a

[7] A function $f(x)$ is strictly convex over a closed interval $[a, b]$ if, for any two point x_1 and x_2 in $[a, b]$, and for any scalar α such that $0 \leq \alpha \leq 1$, then $f\left(\alpha x_1 + (1 - \alpha)x_2\right) < \alpha f(x_1) + (1 - \alpha) f(x_2)$.

stationary point of $f(\mathbf{x})$ is

$$\boxed{\nabla_x f(\mathbf{x}) = 0} \tag{2.71}$$

In order for this stationary point to be a minimum the Hessian matrix, $\mathbf{H}_x$, must be positive definite, i.e.,

$$\mathbf{H}_x > 0$$

where $\mathbf{H}_x$ is the $n \times n$ matrix of second-order partial derivatives with the (i, j)th element given by

$$\{\mathbf{H}_x\}_{i,j} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$$

As with a function of a single real variable, if $f(\mathbf{x})$ is strictly convex, then the solution to Eq. (2.71) is unique and is equal to the global minimum of $f(\mathbf{x})$.

When the function to be minimized is a real-valued function of the complex vectors $\mathbf{z}$ and $\mathbf{z}^*$, finding the minimum of $f(\mathbf{z}, \mathbf{z}^*)$ is complicated by the fact that $f(\mathbf{z}, \mathbf{z}^*)$ is not differentiable. However, as in the scalar case, treating $\mathbf{z}$ and $\mathbf{z}^*$ as independent variables we may solve for the stationary points using the following theorem [1].

Theorem. If $f(\mathbf{z}, \mathbf{z}^*)$ is a *real-valued* function of the complex vectors $\mathbf{z}$ and $\mathbf{z}^*$, then the vector pointing in the direction of the maximum rate of change of $f(\mathbf{z}, \mathbf{z}^*)$ is $\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*)$, which is the derivative of $f(\mathbf{z}, \mathbf{z}^*)$ with respect to $\mathbf{z}^*$.

As a result of this theorem it follows that the stationary points of $f(\mathbf{z}, \mathbf{z}^*)$ are solutions to the equation

$$\boxed{\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*) = 0} \tag{2.72}$$

Some examples of gradient vectors are given in Table 2.6.

Finally, we consider a special type of constrained minimization problem. Let $\mathbf{z} = [z_1, z_2, \ldots, z_n]^T$ be a complex vector and $\mathbf{R}$ a positive-definite Hermitian matrix. Suppose that we want to find the vector $\mathbf{z}$ that minimizes the quadratic form $\mathbf{z}^H\mathbf{R}\mathbf{z}$ subject to the constraint that the solution must satisfy the linear equality $\mathbf{z}^H\mathbf{a} = 1$, where $\mathbf{a}$ is a given complex vector. This is a problem that arises in array processing [4] and, as we will see in Chapter 8, is the problem that must be solved to find the minimum variance spectrum estimate of a random process. Geometrically, this constrained minimization problem is as illustrated in Fig. 2.7 for the case of a two-dimensional vector. The solution may be derived in a number of different ways [9]. One approach, presented here, is to introduce a Lagrange

Table 2.6 ***The Complex Gradient of a Complex Function****

	$\mathbf{a}^H\mathbf{z}$	$\mathbf{z}^H\mathbf{a}$	$\mathbf{z}^H\mathbf{A}\mathbf{z}$
$\nabla_{\mathbf{z}}$	$\mathbf{a}^*$	0	$(\mathbf{A}\mathbf{z})^*$
$\nabla_{\mathbf{z}^*}$	0	$\mathbf{a}$	$\mathbf{A}\mathbf{z}$

*A is assumed to be Hermitian.

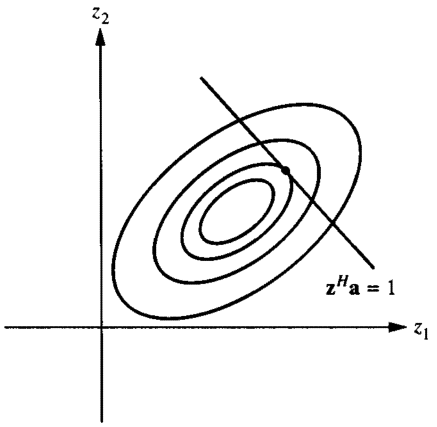


Figure 2.7 *Geometric illustration of the minimization of* $\mathbf{z}^H\mathbf{R}\mathbf{z}$ *subject to the constraint* $\mathbf{z}^H\mathbf{a} = 1$. *Each ellipse represents the locus of points for which the quadratic form is constant,* $\mathbf{z}^H\mathbf{R}\mathbf{z} = c$, *for some value of* c. *The solution occurs at the point where the line* $\mathbf{z}^H\mathbf{a} = 1$ *is tangent to a contour.*

multiplier, λ, and minimize the unconstrained objective function

$$Q_R(\mathbf{z}, \lambda) = \frac{1}{2}\mathbf{z}^H\mathbf{R}\mathbf{z} + \lambda(1 - \mathbf{z}^H\mathbf{a}) \tag{2.73}$$

As we have seen, minimizing $Q_R(\mathbf{z}, \lambda)$ may be accomplished by setting the gradient of $Q_R(\mathbf{z}, \lambda)$ with respect to $\mathbf{z}^*$ equal to zero. Since

$$\nabla_{\mathbf{z}^*} Q_R(\mathbf{z}, \lambda) = \mathbf{R}\mathbf{z} - \lambda\mathbf{a}$$

then

$$\mathbf{z} = \lambda\mathbf{R}^{-1}\mathbf{a} \tag{2.74}$$

The value of the Lagrange multiplier is then found by setting the derivative of $Q_R(\mathbf{z}, \lambda)$ with respect to λ equal to zero as follows

$$\frac{\partial Q_R(\mathbf{z}, \lambda)}{\partial \lambda} = 1 - \mathbf{z}^H\mathbf{a} = 0 \tag{2.75}$$

Substituting Eq. (2.74) for $\mathbf{z}$ into Eq. (2.75) and solving for λ gives

$$\lambda = \frac{1}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}} \tag{2.76}$$

Finally, substituting Eq. (2.76) into Eq. (2.74) yields

$$\boxed{\mathbf{z} = \frac{\mathbf{R}^{-1}\mathbf{a}}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}} \tag{2.77}$$

as the vector $\mathbf{z}$ that minimizes the objective function $Q_R(\mathbf{z}, \lambda)$.[8] Substituting this vector into the quadratic form $Q_R(\mathbf{z}) = \mathbf{z}^H\mathbf{R}\mathbf{z}$ gives, for the minimum value of $\mathbf{z}^H\mathbf{R}\mathbf{z}$,

$$\min_{\mathbf{z}}\left\{\mathbf{z}^H\mathbf{R}\mathbf{z}\right\} = \frac{\mathbf{z}^H\mathbf{a}}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}} = \frac{1}{\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}}$$

where the last equality follows from Eq. (2.75).

2.4 SUMMARY

In this chapter we have reviewed the fundamentals of digital signal processing and established the notation that will be used to describe discrete-time signals and systems. In addition, we have introduced the key concepts from linear algebra that will be used throughout this book. Since many of the problems that we will be solving require that we find the solution to a set of linear equations, of particular importance will be the discussions in Section 2.3.6 related to underdetermined and overdetermined sets of linear equations. For a more complete and rigorous treatment of linear equations, the reader may consult any one of a number of excellent references [3, 6, 10].

References

1. D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc., Parts F and H*, vol. 130, pp. 11–16, February 1983.
2. M. A. Clements and J. Pease, "On causal linear phase IIR digital filters", *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-3, no. 4, pp. 479–484, March 1989.
3. R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
4. D. H. Johnson and D. E. Dudgeon, *Array Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
5. S. Kay, *Modern Spectrum Estimation: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
6. B. Nobel and J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
7. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
8. A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
9. D. A. Pierce, *Optimization Theory with Applications*, John Wiley & Sons, New York, 1969.
10. G. Strang, *Linear Algebra and its Applications*, Academic Press, New York, 1980.
11. R. D. Strum and D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley, MA, 1988.

2.5 PROBLEMS

2.1. The DFT of a sequence $x(n)$ of length N may be expressed in matrix form as follows

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

where $\mathbf{x} = \left[x(0),\ x(1),\ \ldots,\ x(N-1)\right]^T$ is a vector containing the signal values and $\mathbf{X}$ is a vector containing the DFT coefficients $X(k)$,

[8] The assumption that $\mathbf{R}$ is positive definite guarantees that the matrix inverse exists and that this solution is the global minimum.

(a) Find the matrix $\mathbf{W}$.

(b) What properties does the matrix $\mathbf{W}$ have?

(c) What is the inverse of $\mathbf{W}$?

2.2. Prove or disprove each of the following statements:

(a) The product of two upper triangular matrices is upper triangular.

(b) The product of two Toeplitz matrices is Toeplitz.

(c) The product of two centrosymmetric matrices is centrosymmetric.

2.3. Find the minimum norm solution to the following set of underdetermined linear equations,

$$\begin{bmatrix} 1 & 0 & 2 & -1 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

2.4. Consider the set of inconsistent linear equations $\mathbf{Ax} = \mathbf{b}$ given by

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

(a) Find the least squares solution to these equations.

(b) Find the projection matrix $\mathbf{P}_A$.

(c) Find the best approximation $\hat{\mathbf{b}} = \mathbf{P}_A\mathbf{b}$ to $\mathbf{b}$.

(d) Consider the matrix

$$\mathbf{P}_A^{\perp} = \mathbf{I} - \mathbf{P}_A$$

Find the vector $\mathbf{b}^{\perp} = \mathbf{P}_A^{\perp}\mathbf{b}$ and show that it is orthogonal to $\hat{\mathbf{b}}$. What does the matrix $\mathbf{P}_A^{\perp}$ represent?

2.5. Consider the problem of trying to model a sequence $x(n)$ as the sum of a constant plus a complex exponential of frequency ω_0 [5]

$$\hat{x}(n) = c + ae^{jn\omega_0} \quad ; \quad n = 0, 1, \ldots, N-1$$

where c and a are unknown. We may express the problem of finding the values for c and a as one of solving a set of overdetermined linear equations

$$\begin{bmatrix} 1 & 1 \\ 1 & e^{j\omega_0} \\ \vdots & \vdots \\ 1 & e^{j(N-1)\omega_0} \end{bmatrix} \begin{bmatrix} c \\ a \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

(a) Find the least squares solution for c and a.

(b) If N is even and $\omega_0 = 2\pi k/N$ for some integer k, find the least squares solution for c and a.

2.6. It is known that the sum of the squares of n from $n = 1$ to $N - 1$ has a closed form solution of the following form

$$\sum_{n=0}^{N-1} n^2 = a_0 + a_1 N + a_2 N^2 + a_3 N^3$$

Given that a third-order polynomial is uniquely determined in terms of the values of the polynomial at four distinct points, derive a closed form expression for this sum by setting up a set of linear equations and solving these equations for a_0, a_1, a_2, a_3. Compare your solution to that given in Table 2.3.

2.7. Show that a projection matrix $\mathbf{P}_A$ has the following two properties,

1. It is *idempotent*, $\mathbf{P}_A^2 = \mathbf{P}_A$.
2. It is Hermitian.

2.8. Let $\mathbf{A} > 0$ and $\mathbf{B} > 0$ be positive definite matrices. Prove or disprove the following statements.

(a) $\mathbf{A}^2 > 0$.

(b) $\mathbf{A}^{-1} > 0$.

(c) $\mathbf{A} + \mathbf{B} > 0$.

2.9. (a) Prove that each eigenvector of a symmetric Toeplitz matrix is either symmetric or antisymmetric, i.e., $\mathbf{v}_k = \pm \mathbf{J}\mathbf{v}_k$.

(b) What property can you state about the eigenvalues of a Hermitian Toeplitz matrix?

2.10. (a) Find the eigenvalues and eigenvectors of the 2×2 symmetric Toeplitz matrix

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$$

(b) Find the eigenvalues and eigenvectors of the 2×2 Hermitian matrix

$$\mathbf{A} = \begin{bmatrix} a & b^* \\ b & a \end{bmatrix}$$

2.11. Establish Property 5 on p. 45.

2.12. A necessary and sufficient condition for a Hermitian matrix $\mathbf{A}$ to be positive definite is that there exists a nonsingular matrix $\mathbf{W}$ such that

$$\mathbf{A} = \mathbf{W}^H \mathbf{W}$$

(a) Prove this result.

(b) Find a factorization of the form $\mathbf{A} = \mathbf{W}^H \mathbf{W}$ for the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

2.13. Consider the 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

(a) Find the eigenvalues and eigenvectors of **A**.

(b) Are the eigenvectors unique? Are they linearly independent? Are they orthogonal?

(c) Diagonalize **A**, i.e., find **V** and **D** such that

$$\mathbf{V}^H\mathbf{A}\mathbf{V} = \mathbf{D}$$

where **D** is a diagonal matrix.

2.14. Find the eigenvalues and eigenvectors of the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 2 & 4 \end{bmatrix}$$

2.15. Consider the following 3×3 symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

(a) Find the eigenvalues and eigenvectors of **A**.

(b) Find the determinant of **A**.

(c) Find the spectral decomposition of **A**.

(d) What are the eigenvalues of $\mathbf{A} + \mathbf{I}$ and how are the eigenvectors related to those of **A**?

2.16. Suppose that an $n \times n$ matrix **A** has eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$.

(a) What are the eigenvalues and eigenvectors of $\mathbf{A}^2$?

(b) What are the eigenvalues and eigenvectors of $\mathbf{A}^{-1}$?

2.17. Find a matrix whose eigenvalues are $\lambda_1 = 1$ and $\lambda_2 = 4$ with eigenvectors $\mathbf{v}_1 = [3, 1]^T$ and $\mathbf{v}_2 = [2, 1]^T$.

2.18. Gerschgorin's circle theorem states that every eigenvalue of a matrix **A** lies in at least one of the circles $C_1, \ldots, C_N$ in the complex plane where C_i has center at the diagonal entry a_{ii} and its radius is $r_i = \sum_{j \neq i} |a_{ij}|$.

1. Prove this theorem by using the eigenvalue equation $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ to write

$$(\lambda - a_{ii})x_i = \sum_{j \neq i} a_{ij}x_j$$

and then use the triangular inequality,

$$\left| \sum_{j \neq i} a_{ij}x_j \right| \leq \sum_{j \neq i} |a_{ij}x_j|$$

2. Use this theorem to establish the bound on $\lambda_{\max}$ given in Property 7.

3. The matrix

$$\begin{bmatrix} 4 & 1 & 2 \\ 2 & 3 & 0 \\ 3 & 2 & 6 \end{bmatrix}$$

is said to be *diagonally dominant* since $a_{ii} > r_i$. Use Gerschgorin's circle theorem to show that this matrix is nonsingular.

2.19. Consider the following quadratic function of two real-valued variables z_1 and z_2,

$$f(z_1, z_2) = 3z_1^2 + 3z_2^2 + 4z_1z_2 + 8$$

Find the values of z_1 and z_2 that minimize $f(z_1, z_2)$ subject to the constraint that $z_1 + z_2 = 1$ and determine the minimum value of $f(z_1, z_2)$.

DISCRETE-TIME RANDOM PROCESSES

3

3.1 INTRODUCTION

Signals may be broadly classified into one of two types—deterministic and random. A *deterministic* signal is one that may be reproduced exactly with repeated measurements. The unit sample response of a linear shift-invariant filter is an example of a deterministic signal. A *random* signal, or random process, on the other hand, is a signal that is not repeatable in a predictable manner. Quantization noise produced by an A/D converter or a fixed-point digital filter is an example of a random process. Other examples include tape hiss or turntable rumble in audio signals, background clutter in radar images, speckle noise in synthetic aperture radar (SAR) images, and engine noise in speech transmissions from the cockpit of an airplane. Some signals may be considered to be either deterministic or random, depending upon the application. Speech, for example, may be considered to be a deterministic signal if a specific speech waveform is to be processed or analyzed. However, speech may also be viewed as a random process if one is considering the ensemble of all possible speech waveforms in order to design a system that will optimally process speech signals, in general.

In this book, many of the signals that we will be characterizing and analyzing will be random processes. Since a random process may only be described probabilistically or in terms of its average behavior, in this chapter we develop the background that is necessary to understand how a random process may be described and how its statistical properties are affected by a linear shift-invariant system. Although it is assumed that the reader is familiar with the basic theory of random variables, the chapter begins with a brief review of random variables in Section 3.2, focusing primarily on the concepts of statistical averages such as the mean, variance, and correlation. Discrete-time random processes are introduced in Section 3.3 as an indexed sequence of random variables. The characterization of a random process in terms of its ensemble averages such as the mean, autocorrelation, and autocovariance are defined, and properties of these averages are explored. The notion of stationarity and the concept of ergodicity are introduced next. Then, the power spectral density, or power spectrum, of a discrete random process is defined. The power spectrum, which is the Fourier transform of the autocorrelation function, provides an important characterization of a random process and will appear frequently in this book. In Chapter 8, for example, we consider techniques for estimating the power spectrum of a random process

from a single observation of the process. In Section 3.4, we consider the effect of filtering a random process with a linear shift-invariant filter. Specifically, the relationship between the autocorrelation of the input process and the autocorrelation of the output process is established. Finally, Section 3.6 provides a description of a useful and important class of random processes: those that may be generated by filtering white noise with a linear shift-invariant filter. These include autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) random processes.

3.2 RANDOM VARIABLES

In this section, we introduce the concept of a *random variable* and develop some basic properties and characteristics of random variables. These results are important for two reasons. First, since random variables are pervasive and may be found in almost any practical application or natural setting ranging from the theory of games to detection and estimation theory and the restoration and enhancement of distorted signals, it is important to be able to understand and manipulate random variables. The second reason is that the characterization of random variables will serve as the starting point for our development of random processes in Section 3.3.

3.2.1 Definitions

The concept of a random variable may best be illustrated by means of the following simple example. Given a fair coin that is equally likely to result in *Heads* or *Tails* when flipped, one would expect that if the coin were to be flipped repeatedly, then the outcome would be *Heads* approximately half of the time and *Tails* the other half. For example, if flipping the coin N_T times results in n_H *Heads* and n_T *Tails*, then, for N_T large, we should find that

$$\frac{n_H}{N_T} \approx 0.5 \quad ; \quad \frac{n_T}{N_T} \approx 0.5$$

Therefore, with a fair coin there is an equal probability of getting either *Heads* or *Tails* and the following *probability assignment* is made for the two possible experimental outcomes $H = \{\text{Heads}\}$ and $T = \{\text{Tails}\}$,

$$\Pr\{H\} = 0.5 \quad ; \quad \Pr\{T\} = 0.5 \tag{3.1}$$

The set of all possible experimental outcomes is called the *sample space* or the *certain event* and the sample space, denoted by Ω, is always assigned a probability of one

$$\Pr\{\Omega\} = 1$$

For the coin flipping experiment

$$\Omega = \{H, T\}$$

and

$$\Pr\{H, T\} = 1$$

Subsets of the sample space are called *events*, and events consisting of a single element are called *elementary events*. For the coin tossing experiment there are only two elementary

events,

$$\omega_1 = \{H\} \quad ; \quad \omega_2 = \{T\}$$

Given the coin flipping experiment, suppose that a real-valued variable, x, is defined in such a way that a value of 1 is assigned to x if the outcome of a coin flip is *Heads* and a value of -1 is assigned if the outcome is *Tails*. With this definition, a mapping has been defined between the set of experimental outcomes, $\omega_i \in \Omega$, and the set of real numbers, R, i.e., $f : \Omega \longrightarrow R$. This mapping is given by

$$\omega_1 = \{H\} \Longrightarrow x = 1$$
$$\omega_2 = \{T\} \Longrightarrow x = -1$$

Based on the probability assignments defined for the elementary events *Heads* and *Tails* in Eq. (3.1), it follows that there is an equal probability that the variable x will be assigned a value of 1 or -1

$$\begin{aligned} \Pr\{x = 1\} &= 0.5 \\ \Pr\{x = -1\} &= 0.5 \end{aligned} \tag{3.2}$$

Since the only two possible outcomes of the coin flip are *Heads* and *Tails*, then the only possible values that may be assigned to the variable x are $x = 1$ and $x = -1$. Therefore, for any number α that is different from 1 or -1, the probability that $x = \alpha$ is equal to zero

$$\Pr\{x = \alpha\} = 0 \quad \text{if} \quad \alpha \neq \pm 1$$

and the probability that x assumes one of the two values, $x = 1$ or $x = -1$, is equal to one

$$\Pr\{\Omega\} = \Pr\{x = \pm 1\} = 1$$

With this probability assignment on each of the elementary events in the sample space Ω, and with the mapping of each elementary event $\omega_i \in \Omega$ to a value of x, a *random variable* has been defined that is specified in terms of the likelihood (probability) that it assumes a particular value. This random variable has an equal probability of having a value of 1 or -1. More generally, a random variable x that may assume only one of two values, $x = 1$ and $x = -1$, with

$$\Pr\{x = 1\} = p \qquad \text{and} \qquad \Pr\{x = -1\} = 1 - p$$

is referred to as a *Bernoulli random variable*.

A slightly more complicated random variable may be defined by replacing the coin tossing experiment with the roll of a fair die. Specifically, if the number that is obtained with the roll of a fair die is assigned to the variable x, then x will be a random variable that is equally likely to take on any integer value from one to six. In a similar fashion, a *complex random variable* may be defined by assigning complex numbers to elementary events in the sample space. For example, with an experiment consisting of a roll of two fair die, one black and one white, a complex random variable z may be defined with the assignment

$$z = m + jn$$

where m is the number showing on the black die and n is the number showing on the white die.

Each of the random variables considered so far are examples of *discrete random variables* since the sample space Ω consists of a discrete set of events, ω_i. Unlike a discrete

random variable, a random variable of the *continuous type* may assume a continuum of values. An example of a random variable of the *continuous type* is the following. Consider an infinite resolution roulette wheel for which any number in the interval [0, 1] may result from the spin of the wheel. The sample space for this experiment is

$$\Omega = \{\omega : 0 \leq \omega \leq 1\}$$

If the wheel is fair so that any number in the interval [0, 1] is equally likely to occur, then a probability assignment on Ω may be made as follows. For any interval $I = (\alpha_1, \alpha_2]$ that is a subset of [0, 1], define the probability of the event $\omega \in I$ as follows,

$$\Pr\{\omega \in I\} = \Pr\{\alpha_1 < \omega \leq \alpha_2\} = \alpha_2 - \alpha_1$$

In addition, suppose we postulate that, for any two *disjoint intervals*, I_1 and I_2, the probability that an outcome will lie either in I_1 or in I_2 is equal to the sum

$$\Pr\{\omega \in I_1 \text{ or } \omega \in I_2\} = \Pr\{\omega \in I_1\} + \Pr\{\omega \in I_2\}.$$

With this postulate or axiom for the probability of the sum of disjoint events, we have established a probability assignment on the sample space that may be used to determine the probability of any event defined on Ω. Finally, if the value that results from the spin of the wheel is assigned to the variable x, then we have specified a random variable of the continuous type.

Looking back at the random variables that have been considered so far, we see that for each random variable there is an underlying experiment—the flip of a coin, the roll of a die, or the spin of a roulette wheel. In addition, for each experimental outcome ω_i in the sample space Ω, a real or complex number is assigned to the variable x. Thus, as illustrated in Fig. 3.1, a random variable is a function in which the domain is the sample space Ω and the range is a subset of the real or complex numbers. Furthermore, note that the characterization of a random variable x is given statistically in terms of a probability assignment, or *probability law*, that is defined on events in the sample space Ω. This probability law is a rule that assigns a number, called the probability, to each event A in the sample space. In order to be a valid probability assignment, the following three axioms must be satisfied:

1. $\Pr(A) \geq 0$ for every event $A \in \Omega$.
2. $\Pr(\Omega) = 1$ for the certain event Ω.
3. For any two mutually exclusive events A_1 and A_2,

$$\Pr(A_1 \cup A_2) = \Pr(A_1) + \Pr(A_2)$$

Once a probability assignment is defined on events in the sample space, it is possible to develop a probabilistic description for the random variable x. This description is often expressed in terms of the probability that x assumes a value within a given range of values such as

$$\Pr\{x = 1\} \quad \text{or} \quad \Pr\{x \leq 0\} \quad \text{or} \quad \Pr\{0 < x \leq 1\}$$

In signal processing applications, it is the probabilistic description of the random variable, rather than the statistical characterization of events in the sample space, that is generally of interest. Therefore, it is more convenient to have a probability law that is assigned directly to the random variable itself rather than to events in the sample space. For a real-valued random variable, x, one such statistical characterization is the *probability distribution function*,

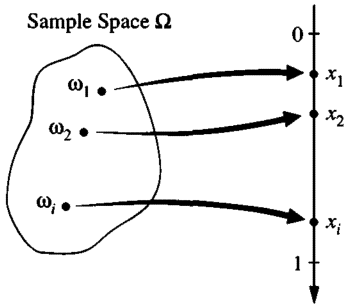


Figure 3.1 *The definition of a random variable in terms of a mapping from elementary events in a sample space Ω to points on the real line.*

$F_x(\alpha)$, given by

$$\boxed{F_x(\alpha) = \Pr\{x \leq \alpha\}} \tag{3.3}$$

For the random variable defined on the coin tossing experiment, the probability distribution function is

$$F_x(\alpha) = \begin{cases} 0 & ; \quad \alpha < -1 \\ 0.5 & ; \quad -1 \leq \alpha < 1 \\ 1 & ; \quad 1 \leq \alpha \end{cases} \tag{3.4}$$

which is plotted in Fig. 3.2*a*. Note that there are two step changes in $F_x(\alpha)$, one at $x = -1$ and the other at $x = 1$. These discontinuities in $F_x(\alpha)$ are due to the discrete *probability masses* at these points.

Another useful statistical characterization of a random variable is the *probability density function*, $f_x(\alpha)$, which is the derivative of the distribution function

$$\boxed{f_x(\alpha) = \frac{d}{d\alpha} F_x(\alpha)} \tag{3.5}$$

For the random variable having the distribution function given in Eq. (3.4), the probability density function, plotted in Fig. 3.2*b*, is

$$f_x(\alpha) = \frac{1}{2}u_0(\alpha + 1) + \frac{1}{2}u_0(\alpha - 1) \tag{3.6}$$

where $u_0(\alpha)$ is the unit impulse function. Density functions containing impulses are characteristic of random variables of the discrete type.

For the continuous random variable defined on the experiment of spinning a roulette wheel, the probability distribution function is

$$F_x(\alpha) = \begin{cases} 0 & ; \quad \alpha < 0 \\ \alpha & ; \quad 0 \leq \alpha < 1 \\ 1 & ; \quad 1 \leq \alpha \end{cases} \tag{3.7}$$

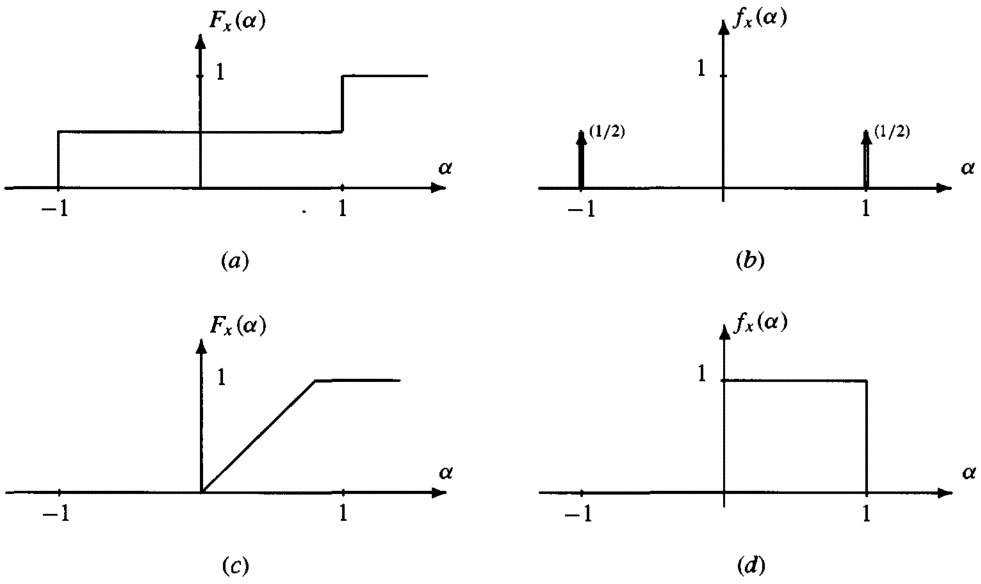


Figure 3.2 *Probability distribution and density functions: (a) Distribution function for the random variable defined on the coin tossing experiment and (b) the density function. (c) Distribution function for the random variable defined on the roulette wheel experiment and (d) the density function.*

and the probability density function is

$$f_x(\alpha) = \begin{cases} 1 & ; \quad 0 \leq \alpha \leq 1 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

These distribution and density functions are shown in Figs. 3.2*c* and 3.2*d*, respectively. Note that, in contrast to a discrete random variable, the probability distribution function of a random variable of the continuous type is a continuous function of α and the density function is piece-wise continuous.

For complex random variables, the distribution function as defined in Eq. (3.3) is not appropriate since the inequality $z \leq \alpha$ is meaningless when z is a complex number. Since a complex random variable

$$z = x + jy$$

is, in essence, a pair of random variables, x and y, the probability assignments on z are made in terms of the joint distribution and joint density functions of x and y, as discussed in Section 3.2.3.

3.2.2 Ensemble Averages

A complete statistical characterization of a random variable requires that it be possible to determine the probability of any event that may be defined on the sample space. In many applications, however, a complete statistical characterization of a random variable may not be necessary if the average behavior of the random variable is known. For example, in making the decision whether or not to play blackjack, it is the expected rate of return on a hand that is of interest rather than a complete statistical characterization of the game. Since

the primary focus of this book will be with statistical averages rather than with probabilistic descriptions of random variables, in this section we review what is meant by the *expected value* of a random variable and the expected value of a function of a random variable.

Let x be the random variable that was defined on the die experiment. Suppose that the die is rolled N_T times and that the number k appears n_k times. Computing the average value that is rolled by summing all of the numbers and dividing by the total number of times that the die is rolled yields the *sample mean*

$$\langle x \rangle_{N_T} = \frac{n_1 + 2n_2 + 3n_3 + 4n_4 + 5n_5 + 6n_6}{N_T} \tag{3.8}$$

If $N_T \gg 1$, then $n_k/N_T \approx \Pr\{x = k\}$ and Eq. (3.8) becomes

$$\langle x \rangle_{N_T} \approx \sum_{k=1}^{6} k \Pr\{x = k\} \tag{3.9}$$

The mean or *expected value* of a discrete random variable x that assumes a value of α_k with probability $\Pr\{x = \alpha_k\}$ is therefore defined as

$$\boxed{E\{x\} = \sum_k \alpha_k \Pr\{x = \alpha_k\}} \tag{3.10}$$

In terms of the probability density function $f_x(\alpha)$, this expectation may be written as

$$\boxed{E\{x\} = \int_{-\infty}^{\infty} \alpha \, f_x(\alpha) d\alpha} \tag{3.11}$$

For random variables of the continuous type, Eq. (3.11) is also used as the definition for the expected value.

Example 3.2.1 ***Computing the Mean of a Random Variable***

For the random variable defined on the coin flipping experiment, the expected value is

$$E\{x\} = \Pr\{x = 1\} - \Pr\{x = -1\} = 0$$

For the random variable defined on the die experiment, on the other hand,

$$E\{x\} = \sum_{k=1}^{6} k \Pr\{x = k\} = \frac{1}{6} \sum_{k=1}^{6} k = 3.5$$

Finally, for the continuous random variable defined in terms of the roulette wheel experiment,

$$E\{x\} = \int_0^1 \alpha f_x(\alpha) d\alpha = \int_0^1 \alpha \, d\alpha = 0.5$$

There are many examples for which it is necessary to be able to compute the expected value of a function of a random variable. For example, in finding the average power that is dissipated in a one ohm resistor when the voltage, x, is a random variable, it is necessary to find the expected value of $y = x^2$. If x is a random variable with a probability density

function $f_x(\alpha)$ and if $y = g(x)$, then the expected value of y is

$$E\{y\} = E\{g(x)\} = \int_{-\infty}^{\infty} g(\alpha) f_x(\alpha) d\alpha \tag{3.12}$$

For example, the expected value of $y = x^2$ is

$$E\{x^2\} = \int_{-\infty}^{\infty} \alpha^2 f_x(\alpha) d\alpha \tag{3.13}$$

and the expected value of $y = |x|$ is

$$E\{|x|\} = \int_{-\infty}^{\infty} |\alpha| f_x(\alpha) d\alpha = \int_{0}^{\infty} \alpha \left[f_x(\alpha) + f_x(-\alpha) \right] d\alpha$$

In order to motivate Eq. (3.12), consider the problem of computing the expected value of x^2 where x is the random variable defined in terms of the roll of a die. Assuming that the die is rolled N_T times and that the number k is rolled n_k times, it follows that the average squared value of the number rolled is approximately equal to the sample mean-square value,

$$\langle x^2 \rangle_{N_T} = \sum_{k=1}^{6} k^2 \frac{n_k}{N_T} \approx \sum_{k=1}^{6} k^2 \Pr\{x = k\} = E\{x^2\} \tag{3.14}$$

In terms of the probability density function, Eq. (3.14) becomes

$$\langle x^2 \rangle_{N_T} \approx \int_{-\infty}^{\infty} \alpha^2 f_x(\alpha) d\alpha$$

which is the same as Eq. (3.13).

The expected value of x^2 is an important statistical average referred to as the *mean-square value*. The mean-square value is frequently used as a measure for the quality of an estimate (e.g., see Section 3.2.6). For example, in developing an estimator for a random variable x, it is common to find the estimator $\hat{x}$ that minimizes the mean-square error,

$$\xi = E\left\{[x - \hat{x}]^2\right\}$$

A related statistical average is the *variance* which is the mean-square value of the random variable $y = x - E\{x\}$. Denoted either by $\text{Var}\{x\}$ or σ_x^2, the variance is

$$\sigma_x^2 = E\left\{\left[x - E\{x\}\right]^2\right\} = \int_{-\infty}^{\infty} \left[\alpha - E\{x\}\right]^2 f_x(\alpha) d\alpha \tag{3.15}$$

The square root of the variance, σ_x, is the *standard deviation*. For complex random variables, the mean-square value is $E\{zz^*\} = E\{|z|^2\}$ and the variance is

$$\sigma_z^2 = E\left\{\left[z - E\{z\}\right]\left[z^* - E\{z^*\}\right]\right\} = E\left\{|z - E\{z\}|^2\right\}$$

It is clear from Eq. (3.12) that the expectation is a linear operator since, for any two random variables x and y and for any constants a and b,

$$E\{ax + by\} = aE\{x\} + bE\{y\}$$

Using the linearity of the expectation operator, the variance may be expressed as

$$\text{Var}\{x\} = E\left\{[x - E\{x\}]^2\right\} = E\{x^2\} - E^2\{x\} \tag{3.16}$$

Therefore, it follows that if the mean of x is zero, then $\text{Var}\{x\} = E\left\{x^2\right\}$.

3.2.3 Jointly Distributed Random Variables

Having considered the basic properties of a single random variable, we now consider what is involved in working with two or more random variables. Unlike the case for a single random variable, with two or more random variables it is necessary to consider the statistical dependencies that may exist between the random variables. For example, consider once again the simple experiment of flipping a fair coin. Suppose that the random variable x is assigned a value of one if the flip results in *Heads* and a value of minus one if the flip results is *Tails*. If two fair coins are tossed and the outcome of the toss of the first coin determines the value of a random variable, $x(1)$, and the outcome of the toss of the second coin determines the value of a second random variable, $x(2)$, then $\{x(1), x(2)\}$ represents a *pair* of random variables with

$$\Pr\{x(i) = 1\} = 0.5$$
$$\Pr\{x(i) = -1\} = 0.5$$

for $i = 1, 2$. If the outcome of the first coin toss does not affect the outcome of the second, then any one of the four possible pairings $\{0, 0\}$, $\{0, 1\}$, $\{1, 0\}$, and $\{1, 1\}$ are equally likely to occur. Compare this, however, with the following experiment. Suppose that we have three coins: one coin which is fair, another coin which is unfair and is much more likely to result in *Heads*, and a third coin that is also unfair but much more likely to result in *Tails*. Just as in the first experiment, let $x(1)$ be a random variable whose value is determined by the flip of the fair coin. Let $x(2)$ be a second random variable whose value is also determined by the flip of a coin, but suppose that the coin that is used depends upon the outcome of the flip of the fair coin. Specifically, if the flip of the fair coin results in *Heads*, then the unfair coin that is more likely to show *Heads* is flipped. Otherwise the coin that is more likely to show *Tails* is flipped. With this definition of $x(1)$ and $x(2)$, we observe that there is a dependence between the two random variables in the sense that it is more likely for $x(1)$ and $x(2)$ to have the same value than it is for them to have different values. As a result, observing the value of $x(1)$ increases the probability of being able to correctly "predict" the value of $x(2)$. This example may be carried to the extreme by using a two-headed coin for the second coin and a two-tailed coin for the third coin. In this case, $x(1)$ and $x(2)$ will always have the same value. What this example illustrates is that the probabilistic description necessary to completely characterize two or more random variables requires that the statistical relationship between them be specified. This relationship is contained in the joint probability distribution and joint probability density functions, which are defined as follows. Given two random variables, $x(1)$ and $x(2)$, the *joint distribution function* is

$$F_{x(1),x(2)}(\alpha_1, \alpha_2) = \Pr\{x(1) \leq \alpha_1,\ x(2) \leq \alpha_2\} \tag{3.17}$$

which defines the probability that $x(1)$ is less than α_1 and $x(2)$ is less than α_2. As with the marginal density function, the *joint density function* for two random variables is the derivative of the distribution function

$$f_{x(1),x(2)}(\alpha_1, \alpha_2) = \frac{\partial^2}{\partial\alpha_1 \partial\alpha_2} F_{x(1),x(2)}(\alpha_1, \alpha_2) \tag{3.18}$$

The joint distribution and density functions are also used for the statistical characterization of complex random variables. For example, if $z = x + jy$ is a complex random variable and $\alpha = a + jb$ is a complex number, then the distribution function for z is given by the

joint distribution function

$$F_z(\alpha) = \Pr\{x \leq a, y \leq b\} \tag{3.19}$$

For more than two random variables, the joint distribution and joint density functions are defined in a similar manner. For example, the joint distribution function for n random variables $x(1), x(2), \ldots, x(n)$ is

$$F_{x(1),\ldots,x(n)}(\alpha_1, \ldots, \alpha_n) = \Pr\{x(1) \leq \alpha_1, \ldots, x(n) \leq \alpha_n\}$$

and the joint density function is

$$f_{x(1),\ldots,x(n)}(\alpha_1, \ldots, \alpha_n) = \frac{\partial^n}{\partial\alpha_1 \cdots \partial\alpha_n} F_{x(1),\ldots,x(n)}(\alpha_1, \ldots, \alpha_n)$$

3.2.4 Joint Moments

Just as with a single random variable, ensemble averages provide an important and useful characterization of jointly distributed random variables. The two averages of primary importance are the *correlation* and the *covariance*. The *correlation*, denoted by r_{xy}, is the second-order joint moment

$$\boxed{r_{xy} = E\{xy^*\}} \tag{3.20}$$

where, in the case of complex random variables, y^* is used to denote the complex conjugate of y. An ensemble average related to the correlation is the covariance, c_{xy}, which is

$$\boxed{c_{xy} = \text{Cov}(x, y) = E\{[x - m_x][y - m_y]^*\} = E\{xy^*\} - m_x m_y^*} \tag{3.21}$$

where $m_x = E\{x\}$ and $m_y = E\{y\}$ are the means (expected values) of the random variables x and y, respectively. Clearly, if either x or y have zero mean, then the covariance is equal to the correlation.

Frequently, it is useful to normalize the covariance so that it is invariant to scaling. Such a normalized covariance is the *correlation coefficient*, which is

$$\boxed{\rho_{xy} = \frac{E\{[x - m_x][y - m_y]^*\}}{\sigma_x \sigma_y} = \frac{E\{xy^*\} - m_x m_y^*}{\sigma_x \sigma_y}} \tag{3.22}$$

For zero mean random variables the correlation coefficient becomes

$$\rho_{xy} = \frac{E\{xy^*\}}{\sigma_x \sigma_y}$$

Due to the normalization by $\sigma_x \sigma_y$, the correlation coefficient is bounded by one in magnitude,

$$\boxed{|\rho_{xy}| \leq 1} \tag{3.23}$$

This property may be easily shown for the case of real random variables as follows. Assuming, without any loss in generality, that x and y have zero mean, it follows that if a is

any real number, then $(ax - y)^2 \geq 0$ and,

$$E\{[ax - y]^2\} = a^2 E\{x^2\} - 2aE\{xy\} + E\{y^2\} \geq 0 \tag{3.24}$$

Since Eq. (3.24) is a quadratic that is nonnegative for all values of a, then the roots of this equation must either be complex or, if they are real, they must be equal to each other. In other words, the discriminant must be nonpositive

$$4E^2\{xy\} - 4E\{x^2\}E\{y^2\} \leq 0$$

or

$$E^2\{xy\} \leq E\{x^2\}E\{y^2\} \tag{3.25}$$

which is the *cosine inequality* [5]. Therefore, it follows that $\rho_{xy}^2 \leq 1$. Note that if there is a value of a for which Eq. (3.24) holds with equality, then $E\{[ax - y]^2\} = 0$ which implies that $y = ax$ (with probability one).

3.2.5 Independent, Uncorrelated, and Orthogonal Random Variables

There are many examples of random variables that arise in applications in which the value of one random variable does not depend on the value of another. For example, if $x(1)$ is a random variable that is defined to be equal to the number of sunspots that occur on the last day of the year and if $x(2)$ is the random variable that is defined to be equal to the amount of the U.S. national debt (in dollars) on the same day, then, in accordance with traditional economic theory, there should not be any relationship or dependence between the values of $x(1)$ and $x(2)$. As a result, the random variables are said to be statistically independent. A more precise formulation of the meaning of statistical independence is given in the following definition.

Definition. Two random variables x and y are said to be *statistically independent* if the joint probability density function is separable,

$$f_{x,y}(\alpha, \beta) = f_x(\alpha) f_y(\beta)$$

A weaker form of independence occurs when the joint second-order moment $E\{xy^*\}$ is separable,

$$E\{xy^*\} = E\{x\}E\{y^*\} \tag{3.26}$$

or

$$r_{xy} = m_x m_y^*$$

Two random variables that satisfy Eq. (3.26) are said to be *uncorrelated.* Note that since

$$c_{xy} = r_{xy} - m_x m_y^*$$

then two random variables x and y will be uncorrelated if their covariance is zero, $c_{xy} = 0$. Clearly, statistically independent random variables are always uncorrelated. The converse, however, is not true in general [5]. A useful property of uncorrelated random variables is the following.

Property. The variance of a sum of uncorrelated random variables is equal to the sum of the variances,

$$\text{Var}\{x + y\} = \text{Var}\{x\} + \text{Var}\{y\}$$

The correlation between random variables provides an important characterization of the statistical dependence that exists between them and it will play an important role in our study of random processes as well as in our discussions of signal modeling, Wiener filtering, and spectrum estimation. As a result, it will be important to understand what the correlation means, beyond simply how it is defined. In the following section we will establish, for example, the relationship between correlation and *linear predictability* when we look at the problem of linear mean-square estimation.

A property related to uncorrelatedness is *orthogonality*. Specifically, two random variables are said to be orthogonal if their correlation is zero,

$$r_{xy} = 0$$

Although orthogonal random variables need not necessarily be uncorrelated, zero mean random variables that are uncorrelated will always be orthogonal.

Since the primary interest in this book is with statistical averages rather than with probabilistic descriptions of random variables, our concern will be primarily with whether or not random variables are uncorrelated rather than whether or not they are statistically independent.

3.2.6 Linear Mean-Square Estimation

In this section we consider briefly the problem of estimating a random variable y in terms of an observation of another random variable x. This problem generally arises when y cannot be directly measured or observed so a related random variable is measured and used to estimate y. For example, we may wish to estimate the IQ (Intelligence Quotient) of an individual in terms of his or her performance on a standardized test. If we represent the IQ by the random variable y and the test performance by the random variable x, assuming that there is some relation (correlation) between x and y, the goal is to find the best estimate of y in terms of x. In mean-square estimation, an estimate $\hat{y}$ is to be found that minimizes the mean-square error

$$\xi = E\left\{(y - \hat{y})^2\right\}$$

Although the solution to this problem generally leads to a nonlinear estimator,[1] in many cases a linear estimator is preferred. In linear mean-square estimation, the estimator is constrained to be of the form

$$\hat{y} = ax + b$$

and the goal is to find the values for a and b that minimize the mean-square error

$$\xi = E\left\{(y - \hat{y})^2\right\} = E\left\{(y - ax - b)^2\right\} \tag{3.27}$$

[1]The optimum estimate is the conditional mean, $\hat{y} = E\{y|x\}$.

There are several advantages to using a linear estimator. The first is that the parameters a and b depend only on the second-order moments of x and y and not on the joint density functions. Second, the equations that must be solved for a and b are linear. Finally, for Gaussian random variables (see Section 3.2.7 for a discussion of Gaussian random variables) the optimum nonlinear mean-square estimate is linear. As we will see in later chapters, this problem is a special case of a more general mean-square estimation problem that arises in many different signal processing applications.

Solving the linear mean-square estimation problem may be accomplished by differentiating ξ with respect to a and b and setting the derivatives equal to zero as follows,

$$\frac{\partial \xi}{\partial a} = -2E\{(y - ax - b)x\} = -2E\{xy\} + 2aE\{x^2\} + 2bm_x = 0 \tag{3.28}$$

$$\frac{\partial \xi}{\partial b} = -2E\{y - ax - b\} = -2m_y + 2am_x + 2b = 0. \tag{3.29}$$

Before solving these equations for a and b, note that Eq. (3.28) says that

$$\boxed{E\{(y - \hat{y})x\} = E\{ex\} = 0} \tag{3.30}$$

where $e = y - \hat{y}$ is the *estimation error*. This relationship, known as the *orthogonality principle*, states that for the optimum linear predictor the estimation error will be orthogonal to the data x. The orthogonality principle is fundamental in mean-square estimation problems and will reappear many times in our discussions of signal modeling and Wiener filtering.

Solving Eqs. (3.28) and (3.29) for a and b we find

$$a = \frac{E\{xy\} - m_x m_y}{\sigma_x^2} \tag{3.31}$$

$$b = \frac{E\{x^2\}m_y - E\{xy\}m_x}{\sigma_x^2} \tag{3.32}$$

From Eq. (3.31) it follows that

$$E\{xy\} = a\sigma_x^2 + m_x m_y \tag{3.33}$$

which, when substituted into Eq. (3.32), gives the following expression for b

$$b = \frac{E\{x^2\}m_y - [a\sigma_x^2 + m_x m_y]m_x}{\sigma_x^2} = m_y - am_x$$

where we have used the relation $\sigma_x^2 = E\{x^2\} - m_x^2$. As a result, the estimate for y may be written as

$$\hat{y} = ax + (m_y - am_x) = a(x - m_x) + m_y \tag{3.34}$$

where

$$a = \frac{E\{xy\} - m_x m_y}{\sigma_x^2} = \frac{\rho_{xy}\sigma_x\sigma_y}{\sigma_x^2} = \rho_{xy}\frac{\sigma_y}{\sigma_x} \tag{3.35}$$

Substituting Eq. (3.35) into Eq. (3.34) we have

$$\boxed{\hat{y} = \rho_{xy}\frac{\sigma_y}{\sigma_x}(x - m_x) + m_y} \tag{3.36}$$

Having found the optimum linear estimator for y, we may now evaluate the minimum mean-square error. With

$$\begin{aligned} E\{[y-\hat{y}]^2\} &= E\{[(y-m_y)-a(x-m_x)]^2\} \\ &= \sigma_y^2 + a^2\sigma_x^2 - 2aE\{(x-m_x)(y-m_y)\} \\ &= \sigma_y^2 + a^2\sigma_x^2 - 2a[E\{xy\} - m_x m_y] \end{aligned} \tag{3.37}$$

substituting Eq. (3.33) into Eq. (3.37) we have

$$E\left\{[y-\hat{y}]^2\right\} = \sigma_y^2 - a^2\sigma_x^2$$

Finally, using expression (3.35) for a, the minimum mean-square error becomes

$$\boxed{\xi_{\min} = \sigma_y^2(1-\rho_{xy}^2)} \tag{3.38}$$

which is plotted in Fig. 3.3 as a function of ρ_{xy}. Note that since the minimum mean-square error $\xi_{\min}$ must be nonnegative, Eq. (3.38) establishes again the property that the correlation coefficient can be no larger than one in absolute value.

We now look at some special cases of the linear mean-square estimation problem. First note that if x and y are uncorrelated, then $a = 0$ and $b = E\{y\}$. Therefore, the estimate for y is

$$\hat{y} = E\{y\} \tag{3.39}$$

and the minimum mean-square error is

$$\xi_{\min} = E\left\{(y-\hat{y})^2\right\} = E\left\{(y-E(y))^2\right\} = \sigma_y^2$$

Thus, x is not used in the estimation of y, so that knowing the value of the random variable x does not improve the accuracy of the estimate of y. Another special case occurs when $|\rho_{xy}| = 1$. In this case, the minimum mean-square error is zero,

$$\xi_{\min} = E\left\{(y-\hat{y})^2\right\} = 0$$

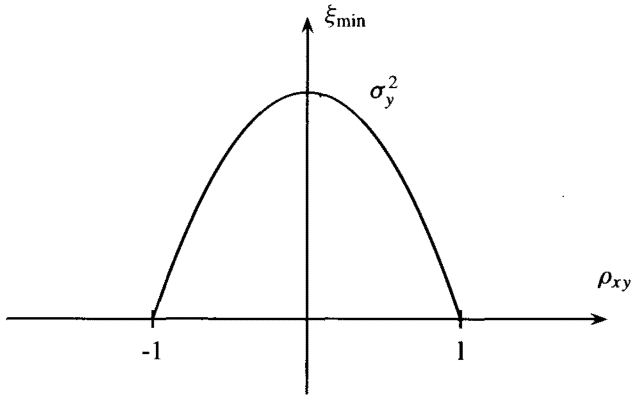


Figure 3.3 *The minimum mean-square error in the estimate of y from x, plotted as a function of the correlation coefficient, ρ_{xy}.*

and it follows that[2]

$$y = ax + b$$

Thus, when the magnitude of the correlation coefficient is equal to one, the random variables x and y are linearly related to each other. Based on these observations we see that the correlation coefficient provides a measure of the *linear predictability* between random variables. The closer $|\rho_{xy}|$ is to 1, the smaller the mean-square error in the estimation of y using a linear estimator. This property will be seen again in later chapters when we consider the more general problem of estimating a random variable, y, in terms of a linear combination of two or more random variables.

3.2.7 Gaussian Random Variables

Gaussian random variables play a central role in probability theory. A random variable x is said to be Gaussian if its density function is of the form

$$f_x(\alpha) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left\{-\frac{(\alpha - m_x)^2}{2\sigma_x^2}\right\}$$

where m_x and σ_x^2 are the mean and the variance of x, respectively. Note that the density function of a Gaussian random variable is completely defined once the mean and the variance are specified. Two random variables x and y are said to be *jointly Gaussian* if their joint density function is

$$f_{x,y}(\alpha, \beta) = A \exp\left\{-\frac{1}{2(1-\rho_{xy}^2)}\left[\frac{(\alpha - m_x)^2}{\sigma_x^2} - 2\rho_{xy}\frac{(\alpha - m_x)(\beta - m_y)}{\sigma_x\sigma_y} + \frac{(\beta - m_y)^2}{\sigma_y^2}\right]\right\}$$

where

$$A = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}}$$

Here, m_x and m_y are the means and σ_x^2 and σ_y^2 the variances of the Gaussian random variables x and y, respectively. The correlation between x and y is ρ_{xy}. As with a single Gaussian random variable, the joint density function is completely specified once the means, variances, and correlation are known.

Gaussian random variables have a number of interesting and important properties. A few of these properties are as follows.

Property 1. If x and y are jointly Gaussian, then for any constants a and b the random variable

$$z = ax + by$$

is Gaussian with mean

$$m_z = am_x + bm_y$$

[2]Strictly speaking, y is said to be equal to $ax + b$ in a *mean-square sense* since y may be different from $ax + b$ as long as the probability of this happening is equal to zero [5].

and variance

$$\sigma_z^2 = a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\sigma_x\sigma_y\rho_{xy}$$

Property 2. If two jointly Gaussian random variables are uncorrelated, $\rho_{xy} = 0$, then they are statistically independent, $f_{xy}(\alpha, \beta) = f_x(\alpha)f_y(\beta)$.

Property 3. If x and y are jointly Gaussian random variables then the optimum nonlinear estimator for y

$$\hat{y} = g(x)$$

that minimizes the mean-square error

$$\xi = E\left\{[y - g(x)]^2\right\}$$

is a linear estimator

$$\hat{y} = ax + b$$

Property 4. If x is Gaussian with zero mean then

$$E\{x^n\} = \begin{cases} 1 \times 3 \times 5 \times \cdots \times (n-1)\sigma_x^n & ; \quad n \text{ even} \\ 0 & ; \quad n \text{ odd} \end{cases}$$

For other interesting and useful properties of Gaussian random variable, the reader may consult any one of a number of standard probability textbooks [3, 5].

3.2.8 Parameter Estimation: Bias and Consistency

There are many examples in signal processing and other scientific fields where it is necessary to estimate the value of an unknown parameter from a set of observations of a random variable. For example, given a set of observations from a Gaussian distribution, estimating the mean or variance from these observations is a problem of parameter estimation. As a specific application, recall that in linear mean-square estimation, estimating the value of a random variable y from an observation of a related random variable, x, the coefficients a and b in the estimator $\hat{y} = ax + b$ depend upon the mean and variance of x and y as well as on the correlation. If these statistical averages are unknown, then it is necessary to estimate these parameters from a set of observations of x and y. Since any estimate will be a function of the observations, the estimates themselves will be random variables. Therefore, in classifying the effectiveness of a particular estimator, it is important to be able to characterize its statistical properties. The statistical properties of interest include the bias and the variance which are described as follows.

Consider the problem of estimating the value of a parameter, θ, from a sequence of random variables, x_n, for $n = 1, 2, \ldots, N$. Since the estimate is a function of N random variables, we will denote it by $\hat{\theta}_N$. In general, we would like the estimate to be equal, *on the average*, to the true value. The difference between the expected value of the estimate and the actual value, θ, is called the *bias* and will be denoted by B,

$$B = \theta - E\{\hat{\theta}_N\}$$

If the bias is zero, then the expected value of the estimate is equal to the true value

$$E\{\hat{\theta}_N\} = \theta$$

and the estimate is said to be *unbiased*. If $B \neq 0$, then $\hat{\theta}$ is said to be *biased*. If an estimate is biased but the bias goes to zero as the number of observations, N, goes to infinity,

$$\lim_{N\to\infty} E\{\hat{\theta}_N\} = \theta$$

then the estimate is said to be *asymptotically unbiased*. In general, it is desirable that an estimator be either unbiased or asymptotically unbiased. However, as the following example illustrates, the bias is not the only statistical measure of importance.

Example 3.2.2 *An Unbiased Estimator*

Let x be the random variable defined on the coin flipping experiment, with $x = 1$ if the outcome is heads and $x = -1$ if the outcome is tails. If the coin is unfair so that the probability of flipping *Heads* is p and the probability of flipping *Tails* is $1 - p$, then the mean of x is

$$m_x = E\{x\} = p - (1 - p) = 2p - 1$$

However, suppose the value for p is unknown and that the mean of x is to be estimated. Flipping the coin N times and denoting the resulting values for x by x_i, consider the following estimator for m_x,

$$\hat{m}_x = x_N$$

Since the expected value of $\hat{m}_x$ is

$$E\{\hat{m}_x\} = E\{x_N\} = 2p - 1$$

then this estimator is *unbiased*. However, it is clear that $\hat{m}_x = x_N$ is not a very good estimator of the mean. The reason is that $\hat{m}_x$ will either be equal to one, with a probability of p, or it will be equal to minus one, with a probability of $1 - p$. Therefore, the accuracy of the estimate does not improve as the number of observations N increases. In fact, note that the variance of the estimate,

$$\text{Var}\{\hat{m}_x\} = \text{Var}\{x_N\} = 4p(1 - p)$$

does not decrease with N.

In order for the estimate of a parameter to converge, in some sense, to its true value, it is necessary that the variance of the estimate go to zero as the number of observations goes to infinity,

$$\lim_{N\to\infty} \text{Var}\{\hat{\theta}_N\} = \lim_{N\to\infty} E\left\{\left|\hat{\theta}_N - E\{\hat{\theta}_N\}\right|^2\right\} = 0 \tag{3.40}$$

If $\hat{\theta}_N$ is unbiased, $E\{\hat{\theta}_N\} = \theta$, it follows from the Tchebycheff inequality [5] that, for any $\epsilon > 0$,

$$\Pr\{|\hat{\theta}_N - \theta| \geq \epsilon\} \leq \frac{\text{Var}\{\hat{\theta}_N\}}{\epsilon^2}$$

Therefore, if the variance goes to zero as $N \to \infty$, then the probability that $\hat{\theta}_N$ differs by more than ϵ from the true value will go to zero. In this case, $\hat{\theta}_N$ is said to *converge to θ with probability one*.

Another form of convergence, stronger than convergence with probability one, is *mean-square convergence*. An estimate $\hat{\theta}_N$ is said to converge to θ in the mean-square sense if

$$\lim_{N\to\infty} E\left\{\left|\hat{\theta}_N - \theta\right|^2\right\} = 0 \tag{3.41}$$

Note that for an unbiased estimator this is equivalent to the condition given in Eq. (3.40) that the variance of the estimate goes to zero.

Finally, an estimate is said to be *consistent* if it converges, in some sense, to the true value of the parameter. Depending upon the form of convergence that is used one has different definitions of consistency [8]. Here we will say that an estimate is consistent if it is asymptotically unbiased and has a variance that goes to zero as N goes to infinity.

Example 3.2.3 *The Sample Mean*

Let x be a random variable with a mean m_x and variance σ_x^2. Given N uncorrelated observations of x, denoted by x_n, suppose that an estimator for m_x is formed as follows,

$$\hat{m}_x = \frac{1}{N}\sum_{n=1}^{N} x_n$$

This estimate, known as the *sample mean*, has an expected value of

$$E\{\hat{m}_x\} = \frac{1}{N}\sum_{n=1}^{N} E\{x_n\} = m_x$$

Therefore, the sample mean is an *unbiased* estimator. In addition, the variance of the estimate is

$$\text{Var}\{\hat{m}_x\} = \frac{1}{N^2}\sum_{n=1}^{N} \text{Var}\{x\} = \frac{\sigma_x^2}{N}$$

Since the variance goes to zero as $N \to \infty$, the sample mean is a *consistent* estimator.

3.3 RANDOM PROCESSES

In this section, we consider the characterization and analysis of *discrete-time random processes*. Since a discrete-time random process is simply an indexed sequence of random variables, with a basic understanding of random variables the extension of the concepts from Section 3.2 to discrete-time random processes is straightforward.

3.3.1 Definitions

Just as a random variable may be thought of as a mapping from a sample space of a given experiment into the set of real or complex numbers, a discrete-time random process is a mapping from the sample space Ω into the set of discrete-time signals $x(n)$. Thus, a discrete-time random process is a collection, or *ensemble*, of discrete-time signals. A simple example of a discrete-time random process is the following. Consider the experiment of rolling a

fair die and let the outcome of a roll of the die be assigned to the random variable A. Thus, A is allowed to assume any integer value between one and six, each with equal probability. With

$$x(n) = A\cos(n\omega_0) \tag{3.42}$$

a random process has been created that consists of an ensemble of six different and equally probable discrete-time signals. A more complicated process may be constructed by considering the experiment of repeatedly flipping a fair coin. By setting the value of $x(n)$ at time n equal to one if *Heads* is flipped at time n and setting $x(n)$ equal to minus one if *Tails* is flipped, then $x(n)$ becomes a discrete-time random process consisting of a random sequence of 1's and -1's. If the flip of the coin at time n in no way affects the outcome of the flip of the coin at any other time (the coin flips are independent), then $x(n)$ is known as a *Bernoulli process*. A sample function of a Bernoulli process is shown in Fig. 3.4*a*.

Given a random process $x(n)$, other processes may be generated by transforming $x(n)$ by means of some mathematical operation. A particularly important and useful transformation is that of linear filtering. For example, given a Bernoulli process $x(n)$, a new process may be generated by filtering $x(n)$ with the first-order recursive filter defined by the difference equation

$$y(n) = 0.5y(n-1) + x(n)$$

An example of a Bernoulli process filtered in this way is shown in Fig. 3.4*b*.

As a final example of a discrete-time random process, consider the experiment of spinning an infinite resolution roulette wheel for which any number in the interval [0, 1] is equally likely to occur. If the number obtained with the spin of the wheel is assigned to the random variable x, let the infinite binary expansion of x be

$$x = \sum_{n=0}^{\infty} x(n)2^{-n}$$

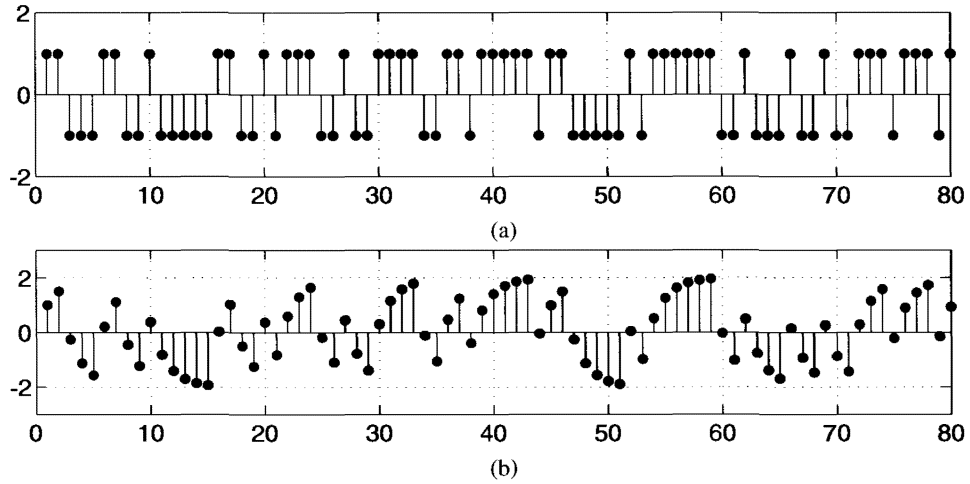


Figure 3.4 *Sample realization of two different random processes: (a) a Bernoulli process $x(n)$ with $\Pr\{x(n) = 1\} = 0.5$ and (b) a filtered Bernoulli process $y(n)$ generated by the difference equation $y(n) = 0.5y(n-1) + x(n)$.*

where $x(n)$ is either equal to zero or one for all $n \geq 0$. The sequence of binary coefficients, $x(n)$, forms a discrete-time random process.

As with random variables, a discrete-time random process is a mapping from a sample space of experimental outcomes to a set (ensemble) of discrete-time signals. For example, as shown in Fig. 3.5, to each experimental outcome ω_i in the sample space Ω, there is a corresponding discrete-time signal, $x_i(n)$. However, in the description and analysis of random processes, another viewpoint is often more convenient. Specifically, note that for a particular value of n, say $n = n_0$, the signal value $x(n_0)$ is a random variable that is defined on the sample space Ω. That is to say, for each $\omega \in \Omega$ there is a corresponding value of $x(n_0)$. Therefore, a random process may also be viewed as an indexed sequence of random variables,

$$\ldots, x(-2), x(-1), x(0), x(1), x(2), \ldots$$

where each random variable in the sequence has an underlying probability distribution function

$$F_{x(n)}(\alpha) = \Pr\{x(n) \leq \alpha\}$$

and probability density function

$$f_{x(n)}(\alpha) = \frac{d}{d\alpha} F_{x(n)}(\alpha) \tag{3.43}$$

However, in order to form a complete statistical characterization of a random process, in addition to the first-order density function, the joint probability density or distribution functions that define how collections of random variables are related to each other must be defined. Specifically, what is required are the joint distribution functions

$$F_{x(n_1),\ldots,x(n_k)}(\alpha_1, \ldots, \alpha_k) = \Pr\{x(n_1) \leq \alpha_1, \ldots, x(n_k) \leq \alpha_k\}$$

(or the corresponding joint density functions) for any collection of random variables $x(n_i)$. Depending upon the specific form of these joint distribution functions, significantly different

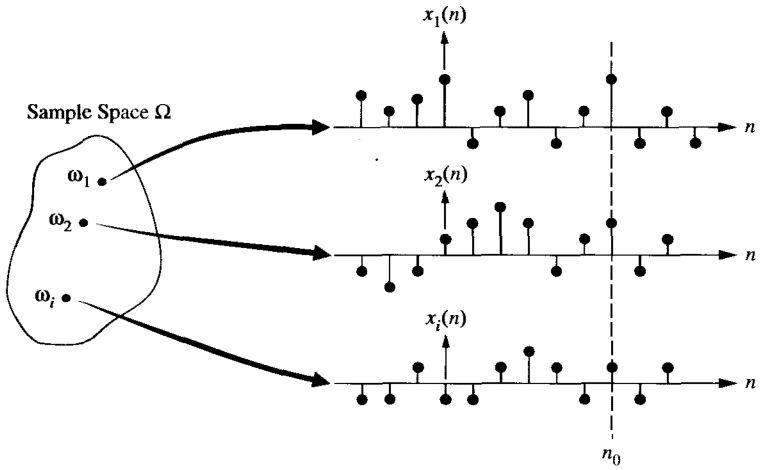


Figure 3.5 *The generation of a random process from a sample space point of view.*

types of random process may be generated. Consider, for example, a random process formed from a sequence of Gaussian random variables $x(n)$. If the random variables $x(n)$ are uncorrelated, then the process, known as white Gaussian noise, is a very random, noiselike sequence. If, on the other hand, $x(n) = \alpha$ where α is a Gaussian random variable, then each random process in the ensemble is equal to a constant for all n. Thus, although both processes have the same first-order statistics, they are significantly different as a result of the differences in their higher-order density functions.

3.3.2 Ensemble Averages

Since a discrete-time random process is an indexed sequence of random variables, we may calculate the mean of each of these random variables and generate the (deterministic) sequence

$$\boxed{m_x(n) = E\{x(n)\}} \tag{3.44}$$

known as the *mean* of the process. Similarly, computing the variance of each random variable in the sequence

$$\boxed{\sigma_x^2(n) = E\left\{|x(n) - m_x(n)|^2\right\}} \tag{3.45}$$

defines the *variance* of the process. These first-order statistics represent ensemble averages and both, in general, depend upon n. Whereas the mean defines the average value of the process as a function of n, the variance represents the average squared deviation of the process away from the mean.

Two additional ensemble averages that are important in the study of random processes are the *autocovariance*

$$\boxed{c_x(k,l) = E\left\{[x(k) - m_x(k)]\,[x(l) - m_x(l)]^*\right\}} \tag{3.46}$$

and the *autocorrelation*

$$\boxed{r_x(k,l) = E\left\{x(k)x^*(l)\right\}} \tag{3.47}$$

relating the random variables $x(k)$ and $x(l)$. Note that if $k = l$ then the autocovariance function reduces to the variance,

$$c_x(k,k) = \sigma_x^2(k)$$

Note also that if the product in Eq. (3.46) is expanded, then it follows that the autocovariance and autocorrelation sequences and related by

$$c_x(k,l) = r_x(k,l) - m_x(k)m_x^*(l)$$

Thus, for zero mean random processes the autocovariance and autocorrelation are equal. In this book, for convenience, unless stated otherwise, **random processes will always be assumed to have zero mean** so that the autocovariance and autocorrelation sequences may be used interchangeably. This assumption results in no loss of generality since, for any process $x(n)$ that has a nonzero mean, a zero mean process $y(n)$ may always be formed by

subtracting the mean from $x(n)$ as follows

$$y(n) = x(n) - m_x(n)$$

As in the case of random variables, the autocorrelation and autocovariance functions provide information about the degree of linear dependence between two random variables. For example, if $c_x(k,l) = 0$ for $k \neq l$, then the random variables $x(k)$ and $x(l)$ are uncorrelated and knowledge of one does not help in the estimation of the other using a linear estimator.

Example 3.3.1 *The Harmonic Process*

An important random process that is found in applications such as radar and sonar signal processing is the harmonic process. An example of a real-valued harmonic process is the random phase sinusoid, which is defined by

$$x(n) = A\sin(n\omega_0 + \phi)$$

where A and ω_0 are fixed constants and ϕ is a random variable that is uniformly distributed over the interval $-\pi$ to π, i.e., the probability density function for ϕ is

$$f_\phi(\alpha) = \begin{cases} (2\pi)^{-1} & ; \quad -\pi \leq \alpha < \pi \\ 0 & ; \quad \text{otherwise} \end{cases}$$

The mean of this process,

$$m_x(n) = E\{x(n)\} = E\{A\sin(n\omega_0 + \phi)\}$$

may be found using Eq. (3.12) as follows

$$m_x(n) = \int_{-\infty}^{\infty} A\sin(n\omega_0 + \alpha) f_\phi(\alpha) d\alpha = \int_{-\pi}^{\pi} \frac{1}{2\pi} A\sin(n\omega_0 + \alpha) d\alpha = 0$$

Thus, $x(n)$ is a zero mean process. The autocorrelation of $x(n)$ may be determined in a similar fashion. Specifically, with

$$r_x(k,l) = E\{x(k)x^*(l)\} = E\{A\sin(k\omega_0 + \phi)A\sin(l\omega_0 + \phi)\}$$

using the trigonometric identity

$$2\sin A \sin B = \cos(A - B) - \cos(A + B)$$

gives

$$r_x(k,l) = \frac{1}{2}A^2 E\{\cos[(k-l)\omega_0]\} - \frac{1}{2}A^2 E\{\cos[(k+l)\omega_0 + 2\phi]\}$$

Note that the first term is the expected value of a constant and the second term is equal to zero. Therefore,

$$r_x(k,l) = \frac{1}{2}A^2\cos[(k-l)\omega_0]$$

As another example, consider the complex harmonic process

$$x(n) = Ae^{j(n\omega_0+\phi)}$$

where, as with the random phase sinusoid, ϕ is a random variable that is uniformly distributed between $-\pi$ and π. The mean of this process is zero

$$m_x(n) = E\{Ae^{j(n\omega_0+\phi)}\} = 0$$

and the autocorrelation is

$$\begin{aligned} r_x(k,l) &= E\{x(k)x^*(l)\} = E\{Ae^{j(k\omega_0+\phi)}A^*e^{-j(l\omega_0+\phi)}\} \\ &= |A|^2 E\{e^{j(k-l)\omega_0}\} = |A|^2 e^{j(k-l)\omega_0} \end{aligned}$$

Note that, for both processes, the mean is a constant and the autocorrelation $r_x(k,l)$ is a function only of the difference between k and l

$$r_x(k,l) = r_x(k-l,0)$$

Thus, the first- and second-order statistics do not depend upon an absolute time origin. That is, the mean and autocorrelation do not change if the process is shifted in time. As discussed in Section 3.3.4, the harmonic process is an example of a *wide-sense stationary process*.

The autocovariance and the autocorrelation sequences provide information about the statistical relationship between two random variables that are derived from the same process, e.g., $x(k)$ and $x(l)$. In applications involving more than one random process it is often of interest to determine the covariance or the correlation between a random variable in one process, $x(k)$, and a random variable in another, $y(l)$. Given two random processes, $x(n)$ and $y(n)$, the *cross-covariance* is defined by

$$c_{xy}(k,l) = E\{[x(k) - m_x(k)][y(l) - m_y(l)]^*\} \tag{3.48}$$

and the *cross-correlation* by

$$r_{xy}(k,l) = E\{x(k)y^*(l)\} \tag{3.49}$$

These two functions satisfy the relation

$$c_{xy}(k,l) = r_{xy}(k,l) - m_x(k)m_y^*(l)$$

Just as a pair of random variables are said to be uncorrelated if $c_{xy} = 0$, two random processes $x(n)$ and $y(n)$ are said to be *uncorrelated* if

$$c_{xy}(k,l) = 0$$

for all k and l or, equivalently, if

$$r_{xy}(k,l) = m_x(k)m_y^*(l) \tag{3.50}$$

Two random processes $x(n)$ and $y(n)$ are said to be *orthogonal* if their cross-correlation is zero

$$r_{xy}(k,l) = 0$$

Although orthogonal random processes are not necessarily uncorrelated, zero mean processes that are uncorrelated are orthogonal.

Example 3.3.2 *Cross-correlation*

Consider the pair of processes, $x(n)$ and $y(n)$, where $y(n)$ is related to $x(n)$ as follows

$$y(n) = x(n-1)$$

The cross-correlation between $x(n)$ and $y(n)$ is

$$r_{xy}(k,l) = E\{x(k)y^*(l)\} = E\{x(k)x^*(l-1)\} = r_x(k,l-1)$$

On the other hand, if $y(n)$ is equal to the convolution of $x(n)$ with a deterministic sequence $h(n)$, such as the unit sample response of a linear shift-invariant filter,

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m)$$

then the cross-correlation between $x(n)$ and $y(n)$ is

$$r_{xy}(k,l) = E\{x(k)y^*(l)\} = E\left\{x(k)\sum_{m=-\infty}^{\infty} h^*(m)x^*(l-m)\right\} = \sum_{m=-\infty}^{\infty} h^*(m)r_x(k,l-m)$$

In any practical data acquisition device, noise or measurement errors are invariably introduced into the recorded data. In many applications, this noise is modeled as additive so that if $x(n)$ denotes the "signal" and $w(n)$ the "noise," the recorded signal is

$$y(n) = x(n) + w(n)$$

Often, this additive noise is assumed to have zero mean and to be uncorrelated with the signal. In this case, the autocorrelation of the measured data, $y(n)$, is the sum of the autocorrelations of $x(n)$ and $w(n)$. Specifically, note that since

$$\begin{aligned} r_y(k,l) &= E\{y(k)y^*(l)\} = E\{[x(k)+w(k)][x(l)+w(l)]^*\} \\ &= E\{x(k)x^*(l)\} + E\{w(k)w^*(l)\} + E\{x(k)w^*(l)\} + E\{w(k)x^*(l)\} \end{aligned} \quad (3.51)$$

if $x(n)$ and $w(n)$ are uncorrelated, then

$$E\{x(k)w^*(l)\} = E\{w(k)x^*(l)\} = 0$$

and it follows that

$$r_y(k,l) = r_x(k,l) + r_w(k,l)$$

This fundamental result is summarized in the following property.

Property. If two random processes $x(n)$ and $y(n)$ are uncorrelated, then the autocorrelation of the sum

$$z(n) = x(n) + y(n)$$

is equal to the sum of the autocorrelations of $x(n)$ and $y(n)$,

$$r_z(k,l) = r_x(k,l) + r_y(k,l)$$

Example 3.3.3 ***Autocorrelation of a Sum of Processes***

Example 3.3.1 considered a harmonic process consisting of a single sinusoid. Now consider a process consisting of a sum of M sinusoids in additive noise

$$x(n) = \sum_{m=1}^{M} A_m \sin(n\omega_m + \phi_m) + v(n)$$

where A_m and ω_m are constants and ϕ_m are uncorrelated random variables that are uniformly distributed between $-\pi$ and π. Since the random variables ϕ_m are uncorrelated it follows that each sinusoidal process is uncorrelated with the others. Assuming that the noise is uncorrelated with the sinusoids, using the results of Example 3.3.1 it follows that

$$r_x(k,l) = \frac{1}{2}\sum_{m=1}^{M} A_m^2 \cos\left[(k-l)\omega_m\right] + r_v(k,l)$$

where $r_v(k,l)$ is the autocorrelation of the additive noise.

3.3.3 Gaussian Processes

In Section 3.2.7, we defined what it means for a pair of random variables to be jointly Gaussian. This definition may be extended to a collection of n random variables as follows. With $\mathbf{x} = \left[x_1, x_2, \ldots, x_n\right]^T$ a vector of n real-valued random variables, $\mathbf{x}$ is said to be a Gaussian random vector and the random variables x_i are said to be jointly Gaussian if the joint probability density function of the n random variables x_i is

$$f_x(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_x|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_x)^T\mathbf{C}_x^{-1}(\mathbf{x}-\mathbf{m}_x)\right\}$$

where $\mathbf{m}_x = \left[m_1, m_2, \ldots, m_n\right]^T$ is a vector containing the means of x_i,

$$m_i = E\left\{x_i\right\}$$

$\mathbf{C}_x$ is a symmetric positive definite matrix with elements c_{ij} that are the covariances between x_i and x_j,

$$c_{ij} = E\left\{(x_i - m_i)(x_j - m_j)\right\}$$

and $|\mathbf{C}_x|$ is the determinant of the covariance matrix.

A discrete-time random process $x(n)$ is said to be *Gaussian* if every finite collection of samples of $x(n)$ are jointly Gaussian. Note that a Gaussian random process is completely defined once the mean vector and covariance matrix are known. Gaussian processes are of great interest both from a theoretical as well as a practical point of view. Many of the processes that are found in applications, for example, are Gaussian, or approximately Gaussian as a result of the Central Limit theorem [5].

3.3.4 Stationary Processes

In signal processing applications, the statistics or ensemble averages of a random process are often independent of time. For example, quantization noise that results from roundoff errors in a fixed point digital signal processor typically has a constant mean and a constant variance whenever the input signal is "sufficiently complex." In addition, it is often assumed that the quantization noise has first- and second-order probability density functions that are independent of time. These conditions are examples of "statistical time-invariance" or *stationarity*. In this section several different types of stationarity are defined. As we will see in Section 3.3.6, a stationarity assumption is important for the estimation of ensemble averages.

If the first-order density function of a random process $x(n)$ is independent of time, i.e.,

$$f_{x(n)}(\alpha) = f_{x(n+k)}(\alpha)$$

for all k, then the process is said to be *first-order stationary*. For a first-order stationary process, the first-order statistics will be independent of time. For example, the mean of the process will be constant

$$m_x(n) = m_x$$

and the same will be true of the variance, $\sigma_x^2(n) = \sigma_x^2$. In a similar manner, a process is said to be *second-order stationary* if the second-order joint density function $f_{x(n_1),x(n_2)}(\alpha_1, \alpha_2)$ depends only on the difference, $n_2 - n_1$, and not on the individual times n_1 and n_2. Equivalently, the process $x(n)$ is second-order stationary if, for any k, the processes $x(n)$ and $x(n+k)$ have the same second-order joint density function

$$f_{x(n_1),x(n_2)}(\alpha_1, \alpha_2) = f_{x(n_1+k),x(n_2+k)}(\alpha_1, \alpha_2)$$

If a process is second-order stationary, then it will also be first-order stationary. In addition, second-order stationary processes have second-order statistics that are invariant to a time shift of the process. For example, it follows that the autocorrelation sequence has the property

$$\begin{aligned} r_x(k,l) &= \int_{-\infty}^{\infty} \alpha\beta f_{x(k),x(l)}(\alpha,\beta)d\alpha\, d\beta \\ &= \int_{-\infty}^{\infty} \alpha\beta f_{x(k+n),x(l+n)}(\alpha,\beta)d\alpha\, d\beta = r_x(k+n, l+n) \end{aligned} \tag{3.52}$$

Therefore, the correlation between the random variables $x(k)$ and $x(l)$ depends only on the difference, $k - l$, separating the two random variables in time

$$r_x(k,l) = r_x(k-l, 0)$$

This difference, $k - l$, is called the *lag*, and with a slight abuse in notation, we will drop the second argument and write $r_x(k,l)$ simply as a function of the lag,

$$r_x(k,l) \equiv r_x(k-l)$$

Continuing to higher-order joint density functions, a process is said to be *stationary of order L* if the processes $x(n)$ and $x(n+k)$ have the same Lth-order joint density functions. Finally, a process that is stationary for all orders $L > 0$ is said to be *stationary in the strict sense*.

Since it is primarily the mean and autocorrelation of a process that will be of interest in this book, and not probability density functions, we will be concerned with another form of stationarity known as *wide-sense stationary* (WSS), which is defined as follows.

Wide Sense Stationarity. A random process $x(n)$ is said to be *wide-sense stationary* if the following three conditions are satisfied:

1. The mean of the process is a constant, $m_x(n) = m_x$.
2. The autocorrelation $r_x(k,l)$ depends only on the difference, $k - l$.
3. The variance of the process is finite, $c_x(0) < \infty$.

Since constraints are placed on ensemble averages rather than on density functions, wide-sense stationarity is a weaker constraint than second-order stationarity. However, in the case of a Gaussian process, wide-sense stationarity is equivalent to strict-sense stationarity.

This is a consequence of the fact that a Gaussian random process is completely defined in terms of the mean and covariance. Some examples of WSS random processes include the Bernoulli process (Fig. 3.4*a*) and the random phase sinusoid considered in Example 3.3.1. An example of a process that is not WSS is the sinusoidal process defined in Eq. (3.42).

In the case of two or more processes, similar definitions exist for *joint stationarity*. For example, two processes $x(n)$ and $y(n)$ are said to be *jointly wide-sense stationary* if $x(n)$ and $y(n)$ are wide-sense stationary and if the cross-correlation $r_{xy}(k, l)$ depends only on the difference, $k - l$,

$$r_{xy}(k, l) = r_{xy}(k + n, l + n)$$

Again, for jointly WSS processes, we will write the cross-correlation as a function only of the lag, $k - l$, as follows

$$r_{xy}(k - l) = E\{x(k)y^*(l)\}$$

The autocorrelation sequence of a WSS process has a number of useful and important properties, some of which are presented as follows.

Property 1—Symmetry. The autocorrelation sequence of a WSS random process is a conjugate symmetric function of k,

$$r_x(k) = r_x^*(-k)$$

For a real process, the autocorrelation sequence is symmetric

$$r_x(k) = r_x(-k)$$

This property follows directly from the definition of the autocorrelation function,

$$r_x(k) = E\{x(n + k)x^*(n)\} = E\{x^*(n)x(n + k)\} = r_x^*(-k)$$

The next property relates the value of the autocorrelation sequence at lag $k = 0$ to the mean-square value of the process.

Property 2—Mean-square value. The autocorrelation sequence of a WSS process at lag $k = 0$ is equal to the mean-square value of the process

$$r_x(0) = E\{|x(n)|^2\} \geq 0$$

As with property 1, this property follows directly from the definition of the autocorrelation sequence. The next property places an upper bound on the autocorrelation sequence in terms of its value at lag $k = 0$.

Property 3—Maximum value. The magnitude of the autocorrelation sequence of a WSS random process at lag k is upper bounded by its value at lag $k = 0$,

$$r_x(0) \geq |r_x(k)|$$

This property may be established as follows. Let a be an arbitrary complex number. Since

$$E\left\{\left|x(n + k) - ax(n)\right|^2\right\} \geq 0$$

expanding the square we have

$$r_x(0) - a^* r_x(k) - a r_x(-k) + |a|^2 r_x(0) \geq 0$$

Using the conjugate symmetry of the autocorrelation sequence, $r_x(-k) = r_x^*(k)$, this becomes

$$(1 + |a|^2) r_x(0) - a^* r_x(k) - a r_x^*(k) \geq 0 \tag{3.53}$$

Since $r_x(k)$ is, in general, complex, let

$$r_x(k) = |r_x(k)| e^{j\phi(k)}$$

where $\phi(k)$ is the phase of $r_x(k)$. If we set $a = e^{j\phi(k)}$, then Eq. (3.53) becomes

$$2r_x(0) - 2|r_x(k)| \geq 0$$

and the property follows. ■

The next property is concerned with periodic random processes.

Property 4—Periodicity. If the autocorrelation sequence of a WSS random process is such that

$$r_x(k_0) = r_x(0)$$

for some k_0, then $r_x(k)$ is periodic with period k_0. Furthermore,

$$E\{|x(n) - x(n - k_0)|^2\} = 0$$

and $x(n)$ is said to be *mean-square periodic*.

We will establish this property in the case of a real-valued random process using the cosine inequality, Eq. (3.25), as follows. With

$$E^2\{[x(n+k+k_0) - x(n+k)][x(n)]\} \leq E\{[x(n+k+k_0) - x(n+k)]^2\} E\{x^2(n)\}$$

expanding the products and using the linearity of the expectation we have

$$[r_x(k + k_0) - r_x(k)]^2 \leq 2[r_x(0) - r_x(k_0)] r_x(0)$$

for all k. Thus, if $r_x(k_0) = r_x(0)$, then $r_x(k + k_0) = r_x(k)$ and $r_x(k)$ is periodic. To show that $x(n)$ is mean-square periodic, note that

$$E\{[x(n) - x(n - k_0)]^2\} = 2r_x(0) - 2r_x(k_0)$$

Therefore, if $r_x(k_0) = r_x(0)$, then

$$E\{[x(n) - x(n - k_0)]^2\} = 0$$

and the result follows. ■

An example of a periodic process is the random phase sinusoid considered in Example 3.3.1. With $x(n) = A\cos(n\omega_0 + \phi)$ the autocorrelation sequence is $r_x(k) = \frac{1}{2}A^2 \cos(k\omega_0)$ Therefore, if $\omega_0 = 2\pi/N$, then $r_x(k)$ is periodic with period N and $x(n)$ is mean-square periodic.

3.3.5 The Autocovariance and Autocorrelation Matrices

The autocovariance and autocorrelation sequences are important second-order statistical characterizations of discrete-time random processes that are often represented in matrix form. For example, if

$$\mathbf{x} = [x(0), x(1), \ldots, x(p)]^T$$

is a vector of $p + 1$ values of a process $x(n)$, then the outer product

$$\mathbf{x}\mathbf{x}^H = \begin{bmatrix} x(0)x^*(0) & x(0)x^*(1) & \cdots & x(0)x^*(p) \\ x(1)x^*(0) & x(1)x^*(1) & \cdots & x(1)x^*(p) \\ \vdots & \vdots & & \vdots \\ x(p)x^*(0) & x(p)x^*(1) & \cdots & x(p)x^*(p) \end{bmatrix} \tag{3.54}$$

is a $(p + 1) \times (p + 1)$ matrix. If $x(n)$ is wide-sense stationary, taking the expected value and using the Hermitian symmetry of the autocorrelation sequence, $r_x(k) = r_x^*(-k)$, leads to the $(p + 1) \times (p + 1)$ matrix of autocorrelation values

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^H\} = \begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2) & \cdots & r_x^*(p) \\ r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix} \tag{3.55}$$

referred to as the *autocorrelation matrix*. In a similar fashion, forming an outer product of the vector $(\mathbf{x} - \mathbf{m}_x)$ with itself and taking the expected value leads to a $(p + 1) \times (p + 1)$ matrix referred to as the *autocovariance matrix*

$$\mathbf{C}_x = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^H\}$$

The relationship between $\mathbf{R}_x$ and $\mathbf{C}_x$ is

$$\mathbf{C}_x = \mathbf{R}_x - \mathbf{m}_x\mathbf{m}_x^H$$

where

$$\mathbf{m}_x = [m_x, m_x, \ldots, m_x]^T$$

is a vector of length $(p + 1)$ containing the mean value of the process.[3] For zero mean processes the autocovariance and autocorrelation matrices are equal. As we stated earlier, it will generally be assumed that all random processes have zero mean. Therefore, only rarely will we encounter the autocovariance matrix. However, since the autocorrelation matrix will arise frequently, in the following paragraphs we explore some of the important properties of this matrix.

The first thing to note about the autocorrelation matrix of a WSS process is that it has a great deal of structure. In addition to being Hermitian, all of the terms along each of the diagonals are equal. Thus, $\mathbf{R}_x$ is a Hermitian Toeplitz matrix. In the case of a real-valued random process, the autocorrelation matrix is a symmetric Toeplitz matrix. Thus, we have the following property of $\mathbf{R}_x$.

[3]Recall that since $x(n)$ is WSS, the mean is a constant.

Property 1. The autocorrelation matrix of a WSS random process $x(n)$ is a Hermitian Toeplitz matrix,

$$\mathbf{R}_x = \text{Toep}\left\{r_x(0),\ r_x(1), \ldots, r_x(p)\right\}$$

The converse, however, is not true—not all Hermitian Toeplitz matrices represent a valid autocorrelation matrix. Note, for example, that since $r_x(0) = E\left\{|x(n)|^2\right\}$, then the terms along the main diagonal of $\mathbf{R}_x$ must be nonnegative. Therefore,

$$\mathbf{R}_x = \begin{bmatrix} -2 & 3 \\ 3 & -2 \end{bmatrix}$$

cannot be the autocorrelation matrix of a WSS process. However, positivity of the terms along the main diagonal is not sufficient to guarantee that a Hermitian Toeplitz matrix is a valid autocorrelation matrix. For example,

$$\mathbf{R}_x = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}$$

does not correspond to a valid autocorrelation matrix. What is required is that $\mathbf{R}_x$ be a nonnegative definite matrix.

Property 2. The autocorrelation matrix of a WSS random process is nonnegative definite, $\mathbf{R}_x > 0$.

To establish this property it must be shown that if $\mathbf{R}_x$ is an autocorrelation matrix, then

$$\mathbf{a}^H \mathbf{R}_x \mathbf{a} \geq 0 \tag{3.56}$$

for any vector $\mathbf{a}$. Since $\mathbf{R}_x = E\left\{\mathbf{x}\mathbf{x}^H\right\}$, we may write Eq. (3.56) as follows

$$\begin{aligned} \mathbf{a}^H \mathbf{R}_x \mathbf{a} &= \mathbf{a}^H E\left\{\mathbf{x}\mathbf{x}^H\right\} \mathbf{a} = E\left\{\mathbf{a}^H \left(\mathbf{x}\mathbf{x}^H\right) \mathbf{a}\right\} \\ &= E\left\{\left(\mathbf{a}^H \mathbf{x}\right)\left(\mathbf{x}^H \mathbf{a}\right)\right\} \end{aligned} \tag{3.57}$$

Therefore,

$$\mathbf{a}^H \mathbf{R}_x \mathbf{a} = E\left\{|\mathbf{a}^H \mathbf{x}|^2\right\}$$

and, since $|\mathbf{a}^H \mathbf{x}|^2 \geq 0$ for any $\mathbf{a}$, it follows that

$$E\left\{|\mathbf{a}^H \mathbf{x}|^2\right\} \geq 0$$

and the property is established. ■

Property 2 provides a necessary condition for a given sequence $r_x(k)$ for $k = 0, 1, \ldots, p$, to represent the autocorrelation values of a WSS random process. A more difficult question concerns how to find a random process having an autocorrelation whose first $p + 1$ values match the given sequence. Since the autocorrelation sequence of a random process is, in general, infinite in length, it is necessary to find a valid extrapolation of $r_x(k)$ for all $|k| > p$. Unfortunately, simply extending $r_x(k)$ with zeros will not always produce a valid autocorrelation sequence. This problem will be considered in more detail in Section 5.2.8 of Chapter 5.

The next property is a result of the fact that the autocorrelation matrix is Hermitian and nonnegative definite. Specifically, recall that the eigenvalues of a Hermitian matrix are real and, for a nonnegative definite matrix, they are nonnegative (p. 43). This leads to the following property.

Property 3. The eigenvalues, λ_k, of the autocorrelation matrix of a WSS random process are real-valued and nonnegative.

Example 3.3.4 *Autocorrelation Matrix*

As we saw in Example 3.3.1 (p. 78), the autocorrelation sequence of a random phase sinusoid is

$$r_x(k) = \frac{1}{2}A^2 \cos(k\omega_0)$$

Therefore, the 2×2 autocorrelation matrix is

$$\mathbf{R}_x = \frac{1}{2}A^2 \begin{bmatrix} 1 & \cos\omega_0 \\ \cos\omega_0 & 1 \end{bmatrix}$$

The eigenvalues of $\mathbf{R}_x$ are

$$\begin{aligned} \lambda_1 &= 1 + \cos\omega_0 \geq 0 \\ \lambda_2 &= 1 - \cos\omega_0 \geq 0 \end{aligned}$$

and the determinant is

$$\det(\mathbf{R}_x) = 1 - \cos^2\omega_0 = \sin^2\omega_0 \geq 0$$

Clearly, $\mathbf{R}_x$ is nonnegative definite, and, if $\omega_0 \neq 0, \pi$, then $\mathbf{R}_x$ is positive definite.

As another example, consider the complex-valued process consisting of a sum of two complex exponentials

$$y(n) = Ae^{j(n\omega_1+\phi_1)} + Ae^{j(n\omega_2+\phi_2)}$$

where A, ω_1, and ω_2 are constants and ϕ_1 and ϕ_2 are uncorrelated random variables that are uniformly distributed between $-\pi$ and π. As we saw in Example 3.3.1, the autocorrelation sequence of a single complex exponential $x(n) = Ae^{j(n\omega_0+\phi)}$ is

$$r_x(k) = |A|^2 e^{jk\omega_0}$$

Since $y(n)$ is the sum of two uncorrelated processes, the autocorrelation sequence of $y(n)$ is

$$r_y(k) = |A|^2 e^{jk\omega_1} + |A|^2 e^{jk\omega_2}$$

and the 2×2 autocorrelation matrix is

$$\mathbf{R}_y = |A|^2 \begin{bmatrix} 2 & e^{-j\omega_1} + e^{-j\omega_2} \\ e^{j\omega_1} + e^{j\omega_2} & 2 \end{bmatrix}$$

The eigenvalues of $\mathbf{R}_y$ are

$$\lambda_1 = 2 + 2\cos\left(\frac{\omega_1 - \omega_2}{2}\right) \geq 0$$

$$\lambda_2 = 2 - 2\cos\left(\frac{\omega_1 - \omega_2}{2}\right) \geq 0$$

Note that if $\omega_2 = -\omega_1$, then this reduces to the random phase sinusoid discussed above.

3.3.6 Ergodicity

The mean and autocorrelation of a random process are examples of ensemble averages that describe the statistical averages of the process over the ensemble of all possible discrete-time signals. Although these ensemble averages are often required in problems such as signal modeling, optimum filtering, and spectrum estimation, they are not generally known a priori. Therefore, being able to estimate these averages from a realization of a discrete-time random process becomes an important problem. In this section, we consider the estimation of the mean and autocorrelation of a random process and present some conditions for which it is possible to estimate these averages using an appropriate *time average*.

Let us begin by considering the problem of estimating the mean, $m_x(n)$, of a random process $x(n)$. If a large number of sample realizations of the process were available, e.g., $x_i(n)$, $i = 1, \ldots, L$, then an average of the form

$$\hat{m}_x(n) = \frac{1}{L}\sum_{i=1}^{L} x_i(n)$$

could be used to estimate the mean $m_x(n)$. In most situations, however, such a collection of sample realizations is not generally available and it is necessary to consider methods for estimating the ensemble average from a *single* realization of the process. Given only a single realization of $x(n)$, we may consider estimating the ensemble average $E\{x(n)\}$ with a *sample mean* that is taken as the average, over time, of $x(n)$ as follows

$$\hat{m}_x(N) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) \tag{3.58}$$

In order for this to be a meaningful estimator for the mean, however, it is necessary that some constraints be imposed on the process. Consider, for example, the random process

$$x(n) = A\cos(n\omega_0)$$

where A is a random variable that is equally likely to assume a value of 1 or 2. The mean of this process is

$$E\{x(n)\} = E\{A\}\cos(n\omega_0) = 1.5\cos(n\omega_0)$$

However, given a single realization of this process, for large N the sample mean will be approximately zero

$$\hat{m}_x(N) \approx 0 \quad ; \quad N \gg 1$$

It should be clear from this example that since the sample mean involves an average over time, the ensemble mean must be a constant, $E\{x(n)\} = m_x$, at least over the interval over

which the time average is being taken. Even with a constant mean, however, the sample mean may not converge to m_x as $N \rightarrow \infty$ without further restrictions on $x(n)$. An illustration of what may happen is given in the following example.

Example 3.3.5 *The Sample Mean*

Consider the random process

$$x(n) = A$$

where A is a random variable whose value is determined by the flip of a fair coin. Specifically, if the toss of a coin results in *Heads*, then $A = 1$; if the result is *Tails*, then $A = -1$. Since the coin is fair, $\Pr\{A = 1\} = \Pr\{A = -1\} = 0.5$, then the mean of the process is

$$m_x(n) = E\{x(n)\} = E\{A\} = 0$$

However, with the sample mean,

$$\hat{m}_x(N) = 1$$

with a probability of one half and

$$\hat{m}_x(N) = -1$$

with a probability of one half. Therefore, $\hat{m}_x(N)$ will not converge, in any sense, to the true mean, $m_x = 0$, as $N \rightarrow \infty$. The difficulty is that the random variables $x(n)$ are perfectly correlated, $r_x(k) = 1$, and no new information about the process is obtained as more sample values are used.

At the other extreme, consider the Bernoulli process that is generated by repeatedly flipping a fair coin. With $x(n) = 1$ if *Heads* is flipped at time n and $x(n) = -1$ if *Tails* is flipped, then, as in the previous case,

$$m_x(n) = E\{x(n)\} = 0$$

However, given a single realization of the process the sample mean is

$$\hat{m}_x(N) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) = \frac{n_1}{N} - \frac{n_2}{N}$$

where n_1 is the number of times that *Heads* is flipped and n_2 is the number of times that *Tails* is flipped. As $N \rightarrow \infty$, since n_1/N converges to $\Pr\{H\} = 1/2$ and n_2/N converges to $\Pr\{T\} = 1/2$, then $\hat{m}_x(N)$ converges to zero, the mean of the process. Unlike the first example, the Bernoulli process consists of a sequence of uncorrelated random variables. Therefore, each new value of $x(n)$ brings additional information about the statistics of the process.

We now consider the convergence of the sample mean to the ensemble mean, m_x, of a WSS process $x(n)$. However, note that since the sample mean is the average of the random variables, $x(0), \ldots, x(N)$, then $\hat{m}_x(N)$ is also a random variable. In fact, viewed as a sequence that is indexed by N, the sample mean is a sequence of random variables. Therefore, in discussing the convergence of the sample mean it is necessary to consider convergence within a statistical framework. Although there are a number of different ways

in which to formulate conditions for the convergence of a sequence of random variables (see Section 3.2.8), the condition of interest here is mean-square convergence [3, 5],

$$\lim_{N\to\infty} E\left\{\left|\hat{m}_x(N) - m_x\right|^2\right\} = 0 \tag{3.59}$$

If Eq. (3.59) is satisfied then $x(n)$ is said to be ergodic in the mean.

Definition. If the sample mean $\hat{m}_x(N)$ of a wide-sense stationary process converges to m_x in the mean-square sense, then the process is said to be *ergodic in the mean* and we write

$$\lim_{N\to\infty} \hat{m}_x(N) = m_x$$

In order for the sample mean to converge in the mean-square sense it is necessary and sufficient that [2, 5].

1. The sample mean be asymptotically unbiased,

$$\lim_{N\to\infty} E\left\{\hat{m}_x(N)\right\} = m_x \tag{3.60}$$

2. The variance of the estimate go to zero as $N \to \infty$,

$$\lim_{N\to\infty} \text{Var}\left\{\hat{m}_x(N)\right\} = 0 \tag{3.61}$$

From the definition of the sample mean it follows easily that the sample mean is unbiased for any wide-sense stationary process,

$$E\left\{\hat{m}_x(N)\right\} = \frac{1}{N}\sum_{n=0}^{N-1} E\{x(n)\} = m_x$$

In order for the variance to go to zero, however, some constraints must be placed on the process $x(n)$. Evaluating the variance of $\hat{m}_x(N)$ we have

$$\begin{aligned}
\text{Var}\{\hat{m}_x(N)\} &= E\left\{\left|\hat{m}_x(N) - m_x\right|^2\right\} = E\left\{\left|\frac{1}{N}\sum_{n=0}^{N-1}[x(n) - m_x]\right|^2\right\} \\
&= \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} E\left\{[x(m) - m_x][x(n) - m_x]^*\right\} \\
&= \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} c_x(m-n)
\end{aligned} \tag{3.62}$$

where $c_x(m-n)$ is the autocovariance of $x(n)$. Grouping together common terms we may write the variance as

$$\begin{aligned}
\text{Var}\{\hat{m}_x(N)\} &= \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} c_x(m-n) = \frac{1}{N^2}\sum_{k=-N+1}^{N-1}(N - |k|)\,c_x(k) \\
&= \frac{1}{N}\sum_{k=-N+1}^{N-1}\left(1 - \frac{|k|}{N}\right)c_x(k)
\end{aligned} \tag{3.63}$$

Therefore, $x(n)$ will be ergodic in the mean if and only if

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=-N+1}^{N-1} \left(1 - \frac{|k|}{N}\right) c_x(k) = 0 \tag{3.64}$$

An equivalent condition that is necessary and sufficient for $x(n)$ to be ergodic in the mean is given in the following theorem [5, 8]

Mean Ergodic Theorem 1. Let $x(n)$ be a WSS random process with autocovariance sequence $c_x(k)$. A necessary and sufficient condition for $x(n)$ to be ergodic in the mean is

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} c_x(k) = 0 \tag{3.65}$$

Equation (3.65) places a necessary and sufficient constraint on the asymptotic decay of the autocorrelation sequence. A sufficient condition that is much easier to apply is given in the following theorem [5]

Mean Ergodic Theorem 2. Let $x(n)$ be a WSS random process with autocovariance sequence $c_x(k)$. Sufficient conditions for $x(n)$ to be ergodic in the mean are that $c_x(0) < \infty$ and

$$\lim_{k\to\infty} c_x(k) = 0 \tag{3.66}$$

Thus, according to Eq. (3.66) a WSS process will be ergodic in the mean if it is asymptotically uncorrelated.

We now establish the sufficiency of (3.66). Due to the conjugate symmetry of $c_x(k)$, the magnitude of $c_x(k)$ is symmetric, $|c_x(k)| = |c_x(-k)|$, and the variance of the sample mean may be upper bounded using Eq.(3.63) as follows:

$$\text{Var}\{\hat{m}_x(N)\} \leq \frac{1}{N} c_x(0) + \frac{2}{N} \sum_{k=1}^{N-1} \left(1 - \frac{|k|}{N}\right) |c_x(k)| \tag{3.67}$$

If $c_x(k) \to 0$ as $k \to \infty$ then, for any $\epsilon > 0$, we may find a value of k, say k_0, such that

$$|c_x(k)| < \epsilon \qquad \text{for all } |k| > k_0$$

Thus, for $N > k_0$ we may write Eq. (3.67) as follows

$$\text{Var}\{\hat{m}_x(N)\} \leq \frac{1}{N} c_x(0) + \frac{2}{N} \sum_{k=1}^{k_0} \left(1 - \frac{|k|}{N}\right) |c_x(k)| + \frac{2}{N} \sum_{k=k_0+1}^{N-1} \left(1 - \frac{|k|}{N}\right) \epsilon \tag{3.68}$$

Since[4] $|c_x(k)| \leq c_x(0)$ and

$$\sum_{k=1}^{k_0} k = \frac{1}{2} k_0(k_0 + 1)$$

[4]This is a straightforward extension of the maximum value property given on p. 83 for the autocorrelation.

we may write (3.68) as

$$\text{Var}\{\hat{m}_x(N)\} \leq \frac{1}{N}c_x(0)\left[1 + 2k_0 - \frac{1}{N}k_0(k_0+1)\right] + \frac{2}{N}\epsilon(N - k_0 - 1)(1 - \frac{k_0}{N})$$

If we fix k_0 and let $N \to \infty$ then

$$\lim_{N\to\infty} \text{Var}\{\hat{m}_x(N)\} \leq 2\epsilon$$

Since the value of ϵ is arbitrary, it follows that the variance goes to zero as $N \to \infty$ and the theorem is established. ■

Example 3.3.6 ***Ergodicity in the Mean***

In Example 3.3.5 we considered the random process

$$x(n) = A$$

where A is a random variable with $\Pr\{A = 1\} = 0.5$ and $\Pr\{A = -1\} = 0.5$. Since the variance of A is equal to one then the autocovariance of $x(n)$ is

$$c_x(k) = 1$$

Therefore,

$$\frac{1}{N}\sum_{k=0}^{N-1} c_x(k) = 1$$

and it follows that this process is not ergodic in the mean (the variance of the sample mean is, in fact, equal to one). The Bernoulli process, on the other hand, consisting of a sequence of independent random variables, each with a variance of one, has an autocovariance of

$$c_x(k) = \delta(k)$$

Therefore, by the second mean ergodic theorem, it follows that the Bernoulli process is ergodic in the mean. Finally, consider the random phase sinusoid $x(n) = A\sin(n\omega_0 + \phi)$ which, assuming $\omega_0 \neq 0$, has an autocovariance of

$$c_x(k) = \frac{1}{2}A^2\cos(k\omega_0)$$

With

$$\sum_{k=0}^{N-1}\cos k\omega_0 = \text{Re}\left\{\sum_{k=0}^{N-1} e^{jk\omega_0}\right\} = \text{Re}\left\{\frac{1 - e^{jN\omega_0}}{1 - e^{j\omega_0}}\right\} = \frac{\sin(N\omega_0/2)}{\sin(\omega_0/2)}\cos[(N-1)\omega_0/2]$$

we have

$$\frac{1}{N}\sum_{k=0}^{N-1} c_x(k) = \frac{A^2}{2N}\frac{\sin(N\omega_0/2)}{\sin(\omega_0/2)}\cos[(N-1)\omega_0/2]$$

which goes to zero as $N \to \infty$ provided $\omega_0 \neq 0$. For $\omega_0 = 0$

$$x(n) = A\sin\phi$$

which is not ergodic in the mean since the covariance is

$$c_x(k) = \frac{1}{2}A^2$$

Therefore, the random phase sinusoid is ergodic in the mean provided $\omega_0 \neq 0$.

The mean ergodic theorems may be generalized to the estimation of other ensemble averages. For example, consider the estimation of the autocorrelation sequence

$$r_x(k) = E\{x(n)x^*(n-k)\}$$

from a single realization of a process $x(n)$. Since, for each k, the autocorrelation is the expected value of the process

$$y_k(n) = x(n)x^*(n-k)$$

we may estimate the autocorrelation from the sample mean of $y_k(n)$ as follows:

$$\hat{r}_x(k, N) = \frac{1}{N}\sum_{n=0}^{N-1} y_k(n) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

If $\hat{r}_x(k, N)$ converges in the mean-square sense to $r_x(k)$ as $N \to \infty$,

$$\lim_{N\to\infty} E\left\{\left|\hat{r}_x(k, N) - r_x(k)\right|^2\right\} = 0$$

then the process is said to be *autocorrelation ergodic* and we write

$$\lim_{N\to\infty} \hat{r}_x(k, N) = r_x(k)$$

Since $\hat{r}_x(k, N)$ is the sample mean of $y_k(n)$, it follows that $x(n)$ will be autocorrelation ergodic if $y_k(n)$ is ergodic in the mean. Applying the first mean ergodic theorem to the sample mean of $y_k(n)$ places a constraint on the covariance of $y_k(n)$, which is equivalent to a constraint on the fourth-order moment of $x(n)$. Another result of interest that is applicable to Gaussian processes is the following [8].

Autocorrelation Ergodic Theorem. A necessary and sufficient condition for a wide-sense stationary Gaussian process with covariance $c_x(k)$ to be autocorrelation ergodic is

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} c_x^2(k) = 0$$

Clearly, in most applications, determining whether or not a given process is ergodic is not practical. Therefore, whenever the solution to a problem requires knowledge of the mean, the autocorrelation, or some other ensemble average, the process is typically assumed to be ergodic and time averages are used to estimate these ensemble averages. Whether or not this assumption is appropriate will be determined by the performance of the algorithm that uses these estimates.

3.3.7 White Noise

An important and fundamental discrete-time process that will be frequently encountered in our treatment of discrete-time random processes is white noise. A wide-sense stationary process $v(n)$, either real or complex, is said to be *white* if the autocovariance function is zero for all $k \neq 0$,

$$c_v(k) = \sigma_v^2\delta(k)$$

Thus, white noise is simply a sequence of uncorrelated random variables, each having a variance of σ_v^2. Since white noise is defined only in terms of the form of its second-order

moment, there is an infinite variety of white noise random processes. For example, a random process consisting of a sequence of uncorrelated real-valued Gaussian random variables is a white noise process referred to as white Gaussian noise (WGN). A sample realization of zero mean unit variance white Gaussian noise is shown in Fig. 3.6*a*. A significantly different type of white noise process is the *Bernoulli process* shown in Fig. 3.6*b*, which consists of a sequence of uncorrelated Bernoulli random variables.

For complex white noise, note that if

$$v(n) = v_1(n) + jv_2(n)$$

then

$$E\{|v(n)|^2\} = E\{|v_1(n)|^2\} + E\{|v_2(n)|^2\}$$

Therefore, it is important to note that the variance of $v(n)$ is the sum of the variances of the real and imaginary components, $v_1(n)$ and $v_2(n)$, respectively.

As we will see in Section 3.4, a wide variety of different and important random processes may be generated by filtering white noise with a linear shift-invariant filter.

3.3.8 The Power Spectrum

Just as Fourier analysis is an important tool in the description and analysis of deterministic discrete-time signals, it also plays an important role in the study of random processes. However, since a random process is an ensemble of discrete-time signals, we cannot compute the Fourier transform of the process itself. Nevertheless, as we will see below, it is possible to develop a frequency domain representation of the process if we express the Fourier transform in terms of an ensemble average.

Recall that the autocorrelation sequence of a WSS process provides a time domain description of the second-order moment of the process. Since $r_x(k)$ is a deterministic sequence

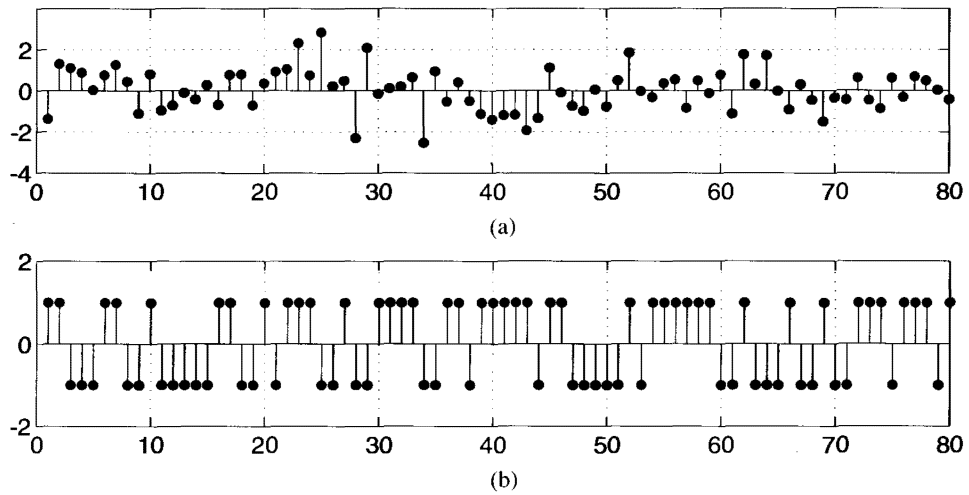


Figure 3.6 *Examples of white noise.* (*a*) *White Gaussian noise.* (*b*) *White Bernoulli noise.*

we may compute its discrete-time Fourier transform,

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_x(k)e^{-jk\omega} \tag{3.69}$$

which is called the *power spectrum* or *power spectral density* of the process.[5] Given the power spectrum, the autocorrelation sequence may be determined by taking the inverse discrete-time Fourier transform of $P_x(e^{j\omega})$,

$$r_x(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})e^{jk\omega}d\omega \tag{3.70}$$

Thus, the power spectrum provides a frequency domain description of the second-order moment of the process. In some cases it may be more convenient to use the z-transform instead of the discrete-time Fourier transform, in which case

$$P_x(z) = \sum_{k=-\infty}^{\infty} r_x(k)z^{-k} \tag{3.71}$$

will also be referred to as the power spectrum of $x(n)$.

Just as with the autocorrelation sequence, there are a number of properties of the power spectrum that will be useful. First of all, since the autocorrelation of a WSS random process is conjugate symmetric, it follows that the power spectrum will be a real-valued function of ω. In the case of a *real-valued* random process, the autocorrelation sequence is real and even, which implies that the power spectrum is real and *even*. Thus, we have the following symmetry property of the power spectrum.

Property 1—Symmetry. The power spectrum of a WSS random process $x(n)$ is real-valued, $P_x(e^{j\omega}) = P_x^*(e^{j\omega})$, and $P_x(z)$ satisfies the symmetry condition

$$P_x(z) = P_x^*(1/z^*)$$

In addition, if $x(n)$ is real then the power spectrum is *even*, $P_x(e^{j\omega}) = P_x(e^{-j\omega})$, which implies that

$$P_x(z) = P_x^*(z^*)$$

In addition to being real-valued, the power spectrum is nonnegative. Specifically,

Property 2—Positivity. The power spectrum of a WSS random process is nonnegative

$$P_x(e^{j\omega}) \geq 0$$

[5] If $x(n)$ does not have zero mean, then the power spectrum as it is defined here will have an impulse at $\omega = 0$. For nonzero mean random processes, the power spectrum is normally defined to be the discrete-time Fourier transform of the autocovariance.

This property follows from the constraint that the autocorrelation matrix is nonnegative definite and will be established in the next section. Finally, a property that relates the average power in a random process to the power spectrum is as follows.

Property 3—Total power. The power in a zero mean WSS random process is proportional to the area under the power spectral density curve

$$E\{|x(n)|^2\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})d\omega$$

This property follows from Eq. (3.70) with $k = 0$ and the fact that $r_x(0) = E\{|x(n)|^2\}$.

Example 3.3.7 ***The Power Spectrum***

The autocorrelation sequence of a zero mean white noise process is $r_v(k) = \sigma_v^2\delta(k)$ where σ_v^2 is the variance of the process. The power spectrum, therefore, is equal to a constant

$$P_v(e^{j\omega}) = \sigma_v^2$$

The random phase sinusoid, on the other hand, has an autocorrelation sequence that is sinusoidal,

$$r_x(k) = \frac{1}{2}A^2\cos(k\omega_0)$$

and has a power spectrum given by

$$P_x(e^{j\omega}) = \frac{1}{2}\pi A^2[u_0(\omega - \omega_0) + u_0(\omega + \omega_0)]$$

As a final example, consider the autocorrelation sequence

$$r_x(k) = \alpha^{|k|}$$

where $|\alpha| < 1$, which, as we will see in Section 3.6.2, corresponds to a first-order autoregressive process. The power spectrum is

$$\begin{aligned} P_x(e^{j\omega}) &= \sum_{k=-\infty}^{\infty} r_x(k)e^{-jk\omega} = \sum_{k=0}^{\infty}\alpha^k e^{-jk\omega} + \sum_{k=0}^{\infty}\alpha^k e^{jk\omega} - 1 \\ &= \frac{1}{1-\alpha e^{-j\omega}} + \frac{1}{1-\alpha e^{j\omega}} - 1 = \frac{1-\alpha^2}{1-2\alpha\cos\omega + \alpha^2} \end{aligned}$$

which is clearly real, even, and nonnegative.

The next property relates the extremal eigenvalues of the autocorrelation matrix of a WSS process to the maximum and minimum values of the power spectrum.

Property 4—Eigenvalue Extremal Property. The eigenvalues of the $n \times n$ autocorrelation matrix of a zero mean WSS random process are upper and lower bounded by the maximum and minimum values, respectively, of the power spectrum,

$$\min_{\omega} P_x(e^{j\omega}) \leq \lambda_i \leq \max_{\omega} P_x(e^{j\omega})$$

To establish this property, let λ_i and $\mathbf{q}_i$ be the eigenvalues and eigenvectors, respectively, of the $n \times n$ autocorrelation matrix $\mathbf{R}_x$,

$$\mathbf{R}_x \mathbf{q}_i = \lambda_i \mathbf{q}_i \quad ; \quad i = 1, 2, \ldots, n$$

Since

$$\mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i = \lambda_i \mathbf{q}_i^H \mathbf{q}_i$$

it follows that

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i} \tag{3.72}$$

Expanding the Hermitian form in the numerator and denoting the coefficients of $\mathbf{q}_i$ by $q_i(k)$,

$$\mathbf{q}_i = [q_i(0),\ q_i(1), \ldots,\ q_i(n-1)]^T$$

we have

$$\mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i = \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} q_i^*(k) r_x(k-l) q_i(l) \tag{3.73}$$

With

$$r_x(k-l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_x(e^{j\omega}) e^{j\omega(k-l)} d\omega$$

we may rewrite Eq. (3.3.7) as follows

$$\begin{aligned} \mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i &= \frac{1}{2\pi} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} q_i^*(k) q_i(l) \int_{\pi}^{\pi} P_x(e^{j\omega}) e^{j\omega(k-l)} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} P_x(e^{j\omega}) \left[\sum_{k=0}^{n-1} q_i^*(k) e^{jk\omega} \sum_{l=0}^{n-1} q_i(l) e^{-jl\omega} \right] d\omega \end{aligned} \tag{3.74}$$

With

$$Q_i(e^{j\omega}) = \sum_{k=0}^{n-1} q_i(k) e^{-jk\omega}$$

which is the Fourier transform of the eigenvector $\mathbf{q}_i$, Eq. (3.74) may be written as

$$\mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_x(e^{j\omega}) |Q(e^{j\omega})|^2 d\omega$$

Repeating these steps for the denominator of Eq. (3.72) we find

$$\mathbf{q}_i^H \mathbf{q}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} |Q(e^{j\omega})|^2 d\omega$$

Therefore,

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R}_x \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i} = \frac{\frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})|Q(e^{j\omega})|^2 d\omega}{\frac{1}{2\pi}\int_{-\pi}^{\pi} |Q(e^{j\omega})|^2 d\omega}$$

Finally, since

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})|Q(e^{j\omega})|^2 d\omega \le \left[\max_{\omega} P_x(e^{j\omega})\right] \frac{1}{2\pi}\int_{-\pi}^{\pi} |Q(e^{j\omega})|^2 d\omega$$

and

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})|Q(e^{j\omega})|^2 d\omega \ge \left[\min_{\omega} P_x(e^{j\omega})\right] \frac{1}{2\pi}\int_{-\pi}^{\pi} |Q(e^{j\omega})|^2 d\omega$$

then it follows that

$$\min_{\omega} P_x(e^{j\omega}) \le \lambda_i \le \max_{\omega} P_x(e^{j\omega})$$

as was to be demonstrated. ■

In addition to providing a frequency domain representation of the second-order moment, the power spectrum may also be related to the ensemble average of the squared Fourier magnitude, $|X(e^{j\omega})|^2$. In particular, consider

$$\begin{aligned} P_N(e^{j\omega}) &= \frac{1}{2N+1}\left|\sum_{n=-N}^{N} x(n)e^{-jn\omega}\right|^2 \\ &= \frac{1}{2N+1}\sum_{n=-N}^{N}\sum_{m=-N}^{N} x(n)x^*(m)e^{-j(n-m)\omega} \end{aligned} \tag{3.75}$$

which is proportional to the squared magnitude of the discrete-time Fourier transform of $2N+1$ samples of a given realization of the random process. Since, for each frequency ω, $P_N(e^{j\omega})$ is a random variable, taking the expected value it follows that

$$E\left\{P_N(e^{j\omega})\right\} = \frac{1}{2N+1}\sum_{n=-N}^{N}\sum_{m=-N}^{N} r_x(n-m)e^{-j(n-m)\omega} \tag{3.76}$$

With the substitution $k = n - m$, Eq. (3.76) becomes, after some rearrangement of terms,

$$\begin{aligned} E\left\{P_N(e^{j\omega})\right\} &= \frac{1}{2N+1}\sum_{k=-2N}^{2N}(2N+1-|k|)r_x(k)e^{-jk\omega} \\ &= \sum_{k=-2N}^{2N}\left(1-\frac{|k|}{2N+1}\right)r_x(k)e^{-jk\omega} \end{aligned} \tag{3.77}$$

Assuming that the autocorrelation sequence decays to zero fast enough so that

$$\sum_{k=-\infty}^{\infty} |k| r_x(k) < \infty$$

taking the limit of Eq. (3.77) by letting $N \to \infty$ it follows that

$$\lim_{N\to\infty} E\left\{P_N(e^{j\omega})\right\} = \sum_{k=-\infty}^{\infty} r_x(k)e^{-jk\omega} = P_x(e^{j\omega}) \tag{3.78}$$

Therefore, combining Eqs. (3.75) and (3.78) we have

$$P_x(e^{j\omega}) = \lim_{N\to\infty} \frac{1}{2N+1} E\left\{ \left| \sum_{n=-N}^{N} x(n)e^{-jn\omega} \right|^2 \right\} \tag{3.79}$$

Thus, the power spectrum may be viewed as the expected value of $P_N(e^{j\omega})$ in the limit as $N \to \infty$.

3.4 FILTERING RANDOM PROCESSES

Linear shift-invariant filters are frequently used to perform a variety of different signal processing tasks in applications that range from signal detection and estimation, to deconvolution, signal representation and synthesis. Since the inputs to these filters are often random processes, it is important to be able to determine how the statistics of these processes change as a result of filtering. In this section we derive the relationship between the mean and autocorrelation of the input process to the mean and autocorrelation of the output process.

Let $x(n)$ be a WSS random process with mean m_x and autocorrelation $r_x(k)$. If $x(n)$ is filtered with a *stable* linear shift-invariant filter having a unit sample response $h(n)$, then the output, $y(n)$, is a random process that is related to $x(n)$ by the convolution sum

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{3.80}$$

The mean of $y(n)$ may be found by taking the expected value of Eq. (3.80) as follows,

$$\begin{aligned} E\{y(n)\} &= E\left\{ \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right\} = \sum_{k=-\infty}^{\infty} h(k)E\{x(n-k)\} \\ &= m_x \sum_{k=-\infty}^{\infty} h(k) = m_x H(e^{j0}) \end{aligned} \tag{3.81}$$

Thus, the mean of $y(n)$ is constant and it is related to the mean of $x(n)$ by a scale factor that is equal to the frequency response of the filter at $\omega = 0$.

We may also use Eq. (3.80) to relate the autocorrelation of $y(n)$ to the autocorrelation of $x(n)$. This is done by first computing the cross-correlation between $x(n)$ and $y(n)$ as follows,

$$\begin{aligned} r_{yx}(n+k,n) &= E\{y(n+k)x^*(n)\} = E\left\{ \sum_{l=-\infty}^{\infty} h(l)x(n+k-l)x^*(n) \right\} \\ &= \sum_{l=-\infty}^{\infty} h(l)E\{x(n+k-l)x^*(n)\} \\ &= \sum_{l=-\infty}^{\infty} h(l)r_x(k-l) \end{aligned} \tag{3.82}$$

Thus, for a wide-sense stationary process $x(n)$, the cross-correlation $r_{yx}(n+k,n)$ depends

only on the difference between $n + k$ and n and Eq. (3.82) may be written as

$$r_{yx}(k) = r_x(k) * h(k) \tag{3.83}$$

The autocorrelation of $y(n)$ may now be determined as follows,

$$\begin{aligned} r_y(n+k,n) &= E\{y(n+k)y^*(n)\} = E\left\{y(n+k)\sum_{l=-\infty}^{\infty} x^*(l)h^*(n-l)\right\} \\ &= \sum_{l=-\infty}^{\infty} h^*(n-l)E\{y(n+k)x^*(l)\} \\ &= \sum_{l=-\infty}^{\infty} h^*(n-l)r_{yx}(n+k-l) \end{aligned} \tag{3.84}$$

Changing the index of summation by setting $m = n - l$ we have

$$r_y(n+k,n) = \sum_{m=-\infty}^{\infty} h^*(m)r_{yx}(m+k) = r_{yx}(k) * h^*(-k) \tag{3.85}$$

Therefore, the autocorrelation sequence $r_y(n + k, n)$ depends only on k, the difference between the indices $n + k$ and n, i.e.,

$$r_y(k) = r_{yx}(k) * h^*(-k) \tag{3.86}$$

Combining Eqs. (3.83) and (3.86) we have

$$r_y(k) = \sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(l)r_x(m-l+k)h^*(m) \tag{3.87}$$

or,

$$\boxed{r_y(k) = r_x(k) * h(k) * h^*(-k)} \tag{3.88}$$

These relationships are illustrated in Fig. 3.7. Thus, it follows from Eqs. (3.81) and (3.88) that if $x(n)$ is WSS, then $y(n)$ will be WSS provided $\sigma_y^2 < \infty$ (this condition requires that the filter be stable). In addition, it follows from Eq. (3.82) that $x(n)$ and $y(n)$ will be jointly WSS.

Another interpretation of Eq. (3.88) is as follows. Defining $r_h(k)$ to be the (deterministic) autocorrelation of the unit sample response, $h(n)$,

$$r_h(k) = h(k) * h^*(-k) = \sum_{n=-\infty}^{\infty} h(n)h^*(n+k)$$

we see that $r_y(k)$ is the convolution of the autocorrelation of the input process with the

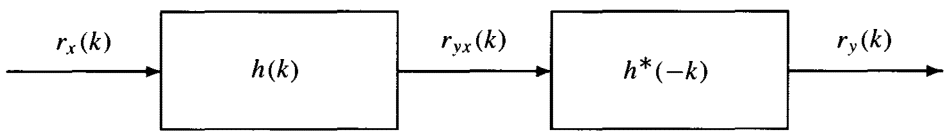


Figure 3.7 *Computing the autocorrelation sequence of a filtered process* $y(n) = x(n) * h(n)$.

deterministic autocorrelation of the filter

$$r_y(k) = r_x(k) * r_h(k) \tag{3.89}$$

For the variance of the output process, $y(n)$, we see from Eq. (3.87) that

$$\sigma_y^2 = r_y(0) = \sum_{l=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h(l) r_x(m-l) h^*(m)$$

As a special case, note that if $h(n)$ is finite in length and zero outside the interval $[0, N-1]$ then the variance (power) of $y(n)$ may be expressed in terms of the autocorrelation matrix $\mathbf{R}_x$ of $x(n)$ and the vector of filter coefficients $\mathbf{h}$ as follows:

$$\boxed{\sigma_y^2 = E\left\{|y(n)|^2\right\} = \mathbf{h}^H \mathbf{R}_x \mathbf{h}} \tag{3.90}$$

Having found the relationship between the autocorrelation of $x(n)$ and the autocorrelation of $y(n)$, we may relate the power spectrum of $y(n)$ to that of $x(n)$ as follows:

$$\boxed{P_y(e^{j\omega}) = P_x(e^{j\omega}) |H(e^{j\omega})|^2} \tag{3.91}$$

Therefore, if a WSS process is filtered with a linear shift-invariant filter, then the power spectrum of the input signal is multiplied by the squared magnitude of the frequency response of the filter. In terms of z-transforms, Eq. (3.88) becomes

$$\boxed{P_y(z) = P_x(z) H(z) H^*(1/z^*)} \tag{3.92}$$

If $h(n)$ is real, then $H(z) = H^*(z^*)$ and Eq. (3.92) becomes

$$P_y(z) = P_x(z) H(z) H(1/z) \tag{3.93}$$

Thus, if $h(n)$ is complex and the system function $H(z)$ has a pole at $z = z_0$, then the power spectrum $P_y(z)$ will have a pole at $z = z_0$ and another at the conjugate reciprocal location $z = 1/z_0^*$.[6] Similarly, if $H(z)$ has a zero at $z = z_0$, then the power spectrum of $y(n)$ will have a zero at $z = z_0$ as well as a zero at $z = 1/z_0^*$. This is a special case of *spectral factorization* as discussed in the following section.

Example 3.4.1 ***Filtering White Noise***

Let $x(n)$ be the random process that is generated by filtering white noise $w(n)$ with a first-order linear shift-invariant filter having a system function

$$H(z) = \frac{1}{1 - 0.25z^{-1}}$$

If the variance of the white noise is equal to one, $\sigma_w^2 = 1$, then the power spectrum of $x(n)$ is

$$P_x(z) = \sigma_w^2 H(z) H(z^{-1}) = \frac{1}{(1 - 0.25z^{-1})(1 - 0.25z)}$$

[6] It is assumed that there are no pole/zero cancelations between $P_x(z)$ and $H(z)$.

Therefore, the power spectrum has a pair of poles, one at $z = 0.25$ and one at the reciprocal location $z = 4$.

The autocorrelation of $x(n)$ may be found from $P_x(z)$ as follows. Performing a partial fraction expansion of $P_x(z)$,

$$P_x(z) = \frac{z^{-1}}{(1 - 0.25z^{-1})(z^{-1} - 0.25)} = \frac{16/15}{1 - 0.25z^{-1}} + \frac{4/15}{z^{-1} - 0.25}$$
$$= \frac{16/15}{1 - 0.25z^{-1}} - \frac{16/15}{1 - 4z^{-1}}$$

and taking the inverse z-transform yields

$$r_x(k) = \tfrac{16}{15}\left(\tfrac{1}{4}\right)^k u(k) + \tfrac{16}{15}4^k u(-k-1) = \tfrac{16}{15}\left(\tfrac{1}{4}\right)^{|k|}$$

The following example shows that it is possible to work backwards from a rational power spectral density function to find a filter that may be used to generate the process.

Example 3.4.2 *Spectral Shaping Filter*

Suppose that we would like to generate a random process having a power spectrum of the form

$$P_x(e^{j\omega}) = \frac{5 + 4\cos 2\omega}{10 + 6\cos\omega} \tag{3.94}$$

by filtering unit variance white noise with a linear shift-invariant filter. Writing $P_x(e^{j\omega})$ in terms of complex exponentials we have

$$P_x(e^{j\omega}) = \frac{5 + 2e^{j2\omega} + 2e^{-j2\omega}}{10 + 3e^{j\omega} + 3e^{-j\omega}}$$

Replacing $e^{j\omega}$ by z gives

$$P_x(z) = \frac{5 + 2(z^2 + z^{-2})}{10 + 3(z + z^{-1})} = \frac{(2z^2 + 1)(2z^{-2} + 1)}{(3z + 1)(3z^{-1} + 1)}$$

Performing the factorization

$$P_x(z) = H(z)H(z^{-1}) \tag{3.95}$$

where

$$H(z) = \frac{2z^2 + 1}{3z + 1} = z\frac{2}{3}\frac{1 + \frac{1}{2}z^{-2}}{1 + \frac{1}{3}z^{-1}}$$

we see that $H(z)$ is a stable filter. Therefore, it follows from Eq. (3.93) that if unit variance white noise is filtered with $H(z)$, then the filtered process will have the power spectrum given in Eq. (3.94). Since introducing a delay into $H(z)$ will not alter the power spectrum of the filtered process, we may equivalently use the filter

$$H(z) = \frac{2}{3}\frac{1 + \frac{1}{2}z^{-2}}{1 + \frac{1}{3}z^{-1}}$$

which is *causal* and has a unit sample response given by

$$h(n) = \tfrac{2}{3}\left(-\tfrac{1}{3}\right)^n u(n) + \tfrac{1}{3}\left(-\tfrac{1}{3}\right)^{n-2} u(n-2)$$

We will have more to say about factorizations of the form given in Eq. (3.95) in the following section.

Thus far, we have defined the power spectrum to be the Fourier transform of the autocorrelation sequence of a WSS process. Physically, the power spectrum describes how the signal power is distributed as a function of frequency. To see this, let $H(e^{j\omega})$ be a narrow-band bandpass filter with center frequency ω_0 and bandwidth $\Delta\omega$, as shown in Fig. 3.8. If $x(n)$ is a zero mean WSS process that is filtered with $H(e^{j\omega})$, then the output process, $y(n)$, will have a power spectrum given by Eq. (3.91). Since the average power in $y(n)$ is

$$E\{|y(n)|^2\} = r_y(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_y(e^{j\omega})d\omega$$

it follows from Eq. (3.91) that

$$\begin{aligned} E\{|y(n)|^2\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 P_x(e^{j\omega})d\omega \\ &= \frac{1}{2\pi}\int_{\omega_0-\Delta\omega/2}^{\omega_0+\Delta\omega/2} P_x(e^{j\omega})d\omega \\ &\approx \frac{\Delta\omega}{2\pi} P_x(e^{j\omega_0}) \end{aligned} \tag{3.96}$$

Therefore, $P_x(e^{j\omega})$ may be viewed as a density function that describes how the power in $x(n)$ varies with ω.

Equation (3.96) may be used to show that the power spectrum is a nonnegative function of ω. To do this, note that since $E\{|y(n)|^2\}$ is always nonnegative, then

$$\frac{1}{2\pi}\int_{\omega_0-\Delta\omega/2}^{\omega_0+\Delta\omega/2} P_x(e^{j\omega})d\omega \geq 0 \tag{3.97}$$

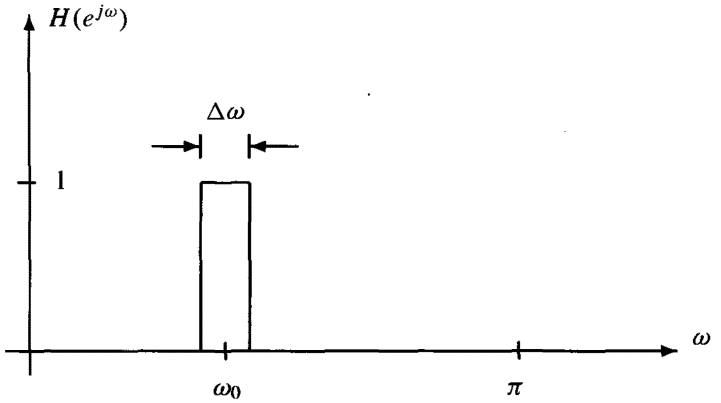


Figure 3.8 *A narrow-band bandpass filter with a center frequency of* ω_0 *and a passband width of* $\Delta\omega$.

However, the only way that Eq. (3.97) may hold for any ω_0 and for all $\Delta\omega$ is for $P_x(e^{j\omega})$ to be nonnegative.

3.5 SPECTRAL FACTORIZATION

The power spectrum $P_x(e^{j\omega})$ of a wide-sense stationary process is a real-valued, positive, and periodic function of ω. In this section, we show that if $P_x(e^{j\omega})$ is a continuous function of ω, then these constraints imply that $P_x(z)$ may be factored into a product of the form

$$P_x(z) = \sigma_0^2 Q(z) Q^*(1/z^*)$$

This is known as the *spectral factorization* of $P_x(z)$.

Let $x(n)$ be a WSS process with autocorrelation $r_x(k)$ and power spectrum $P_x(e^{j\omega})$. We will assume that $P_x(e^{j\omega})$ is a continuous function of ω, which implies that $x(n)$ contains no periodic components. With

$$P_x(z) = \sum_{k=-\infty}^{\infty} r_x(k) z^{-k}$$

let us further assume that $\ln[P_x(z)]$ is *analytic* in an annulus $\rho < |z| < 1/\rho$ that contains the unit circle. Analyticity of the logarithm of $P_x(z)$ implies that $\ln[P_x(z)]$ and all of its derivatives are continuous functions of z and that $\ln[P_x(z)]$ can be expanded in a Laurent series of the form [1]

$$\ln P_x(z) = \sum_{k=-\infty}^{\infty} c(k) z^{-k}$$

where $c(k)$ are the coefficients of the expansion. Thus, $c(k)$ is the sequence that has $\ln[P_x(z)]$ as its z-transform.[7] Alternatively, evaluating $\ln[P_x(z)]$ on the unit circle

$$\ln P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} c(k) e^{-jk\omega} \tag{3.98}$$

we see that $c(k)$ may also be viewed as the Fourier coefficients of the periodic function $\ln[P_x(e^{j\omega})]$. Therefore,

$$c(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln P_x(e^{j\omega}) e^{jk\omega} d\omega$$

Since $P_x(e^{j\omega})$ is real, it follows that the coefficients $c(k)$ are conjugate symmetric, $c(-k) = c^*(k)$. In addition, note that $c(0)$ is proportional to the area under the logarithm of the power spectrum,

$$\boxed{c(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln P_x(e^{j\omega}) d\omega} \tag{3.99}$$

Using the expansion given in Eq. (3.98), we may write the power spectrum in factored form

[7]The sequence $c(k)$ is the *cepstrum* of the sequence $r_x(k)$ [4].

as follows:

$$\begin{aligned} P_x(z) &= \exp\left\{\sum_{k=-\infty}^{\infty} c(k)z^{-k}\right\} \\ &= \exp\{c(0)\}\exp\left\{\sum_{k=1}^{\infty} c(k)z^{-k}\right\}\exp\left\{\sum_{k=-\infty}^{-1} c(k)z^{-k}\right\} \end{aligned} \tag{3.100}$$

Let us define

$$Q(z) = \exp\left\{\sum_{k=1}^{\infty} c(k)z^{-k}\right\} \quad ; \quad |z| > \rho \tag{3.101}$$

which is the z-transform of a causal and stable sequence, $q(k)$. Therefore, $Q(z)$ may be expanded in a power series of the form

$$Q(z) = 1 + q(1)z^{-1} + q(2)z^{-2} + \cdots$$

where $q(0) = 1$ due to the fact that $Q(\infty) = 1$. Since $Q(z)$ and $\ln[Q(z)]$ are both analytic for $|z| > \rho$, $Q(z)$ is a *minimum phase* filter. For a rational function of z this implies that $Q(z)$ has no poles or zeros outside the unit circle. As a result, $Q(z)$ has a stable and causal inverse $1/Q(z)$. Using the conjugate symmetry of $c(k)$, we may express the second factor in Eq. (3.100) in terms of $Q(z)$ as follows:

$$\exp\left\{\sum_{k=-\infty}^{-1} c(k)z^{-k}\right\} = \exp\left\{\sum_{k=1}^{\infty} c^*(k)z^{k}\right\} = \exp\left\{\sum_{k=1}^{\infty} c(k)(1/z^*)^{-k}\right\}^* = Q^*(1/z^*)$$

This leads to the following *spectral factorization* of the power spectrum $P_x(z)$,

$$\boxed{P_x(z) = \sigma_0^2\, Q(z)\, Q^*(1/z^*)} \tag{3.102}$$

where

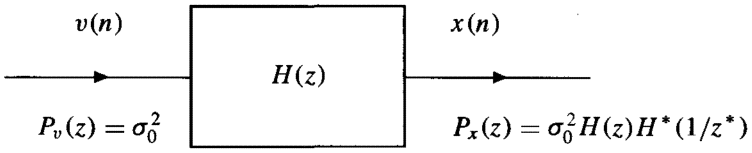
$$\boxed{\sigma_0^2 = \exp\{c(0)\} = \exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln P_x(e^{j\omega})d\omega\right\}} \tag{3.103}$$

is real and nonnegative. For a real-valued process the spectral factorization takes the form

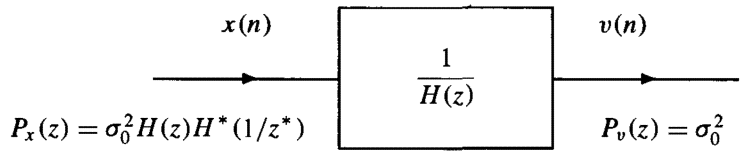
$$P_x(z) = \sigma_0^2 Q(z)Q(z^{-1}) \tag{3.104}$$

Any process that can be factored as in Eq. (3.102) is called a *regular* process [8, 10]. From this factorization we have the following properties of regular processes.

1. Any regular process may be realized as the output of a causal and stable filter that is driven by white noise having a variance of σ_0^2. This is known as the *innovations representation* of the process and is illustrated in Fig. 3.9*a*.
2. The inverse filter $1/H(z)$ is a *whitening filter*. That is, if $x(n)$ is filtered with $1/H(z)$, then the output is white noise with a variance of σ_0^2. The formation of this white noise process, called the *innovations process*, is illustrated in Fig. 3.9*b*,
3. Since $v(n)$ and $x(n)$ are related by an invertible transformation, either process may be derived from the other. Therefore, they both contain the same *information*.



(a) A random process $x(n)$ represented as the response of a minimum phase filter to white noise.



(b) A whitening filter for the process $x(n)$.

Figure 3.9 *Innovations representation of a WSS process.*

For the special case in which $P_x(z)$ is a rational function

$$P_x(z) = \frac{N(z)}{D(z)}$$

the spectral factorization (3.102) states that $P_x(z)$ may be written in factored form as

$$\boxed{P_x(z) = \sigma_0^2 Q(z)Q^*(1/z^*) = \sigma_0^2 \left[\frac{B(z)}{A(z)}\right]\left[\frac{B^*(1/z^*)}{A^*(1/z^*)}\right]} \tag{3.105}$$

where σ_0^2 is a constant, $B(z)$ is a monic polynomial [8]

$$B(z) = 1 + b(1)z^{-1} + \cdots + b(q)z^{-q}$$

having all of its roots inside the unit circle, and $A(z)$ is a monic polynomial

$$A(z) = 1 + a(1)z^{-1} + \cdots + a(p)z^{-p}$$

with all of its roots inside the unit circle. The fact that any rational power spectrum has a factorization of this form is easy to show. As stated in the symmetry property for the power spectrum (Property 1 on p. 95), since $P_x(e^{j\omega})$ is real, then $P_x(z) = P_x^*(1/z^*)$. Therefore, for each pole (or zero) in $P_x(z)$ there will be a matching pole (or zero) at the conjugate reciprocal location. These symmetry constraints are illustrated in Fig. 3.10 for the zeros of the power spectrum.

Before concluding this discussion, we should point out that the power spectrum given in Eq. (3.102) is not the most general representation for a WSS random process. According to the Wold decomposition theorem, any WSS random process may be decomposed into a sum of two orthogonal processes, a regular process $x_r(n)$ and a predictable process $x_p(n)$ [6, 11]. A process $x_p(n)$ is said to be predictable (deterministic) if there exists a set of coefficients,

[8] A monic polynomial is one for which the coefficient of the zeroth-order term is equal to one. For example, $f(z) = 1 + 2z - 3z^2$ is a second-order monic polynomial whereas $f(z) = 3 + 2z - 3z^2$ is not.

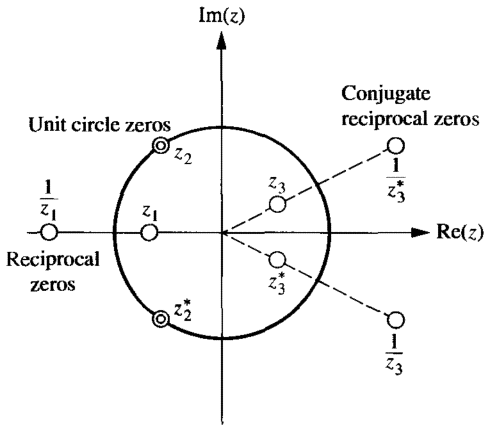


Figure 3.10 *The symmetry constraints that are placed on the zeros of the power spectrum of a real-valued random process when the power spectrum is a rational function of z.*

$a(k)$, such that

$$x_p(n) = \sum_{k=1}^{\infty} a(k)x_p(n-k)$$

In other words, $x_p(n)$ may be predicted without error in terms of a linear combination of its previous values. It may be shown [6] that a process is predictable if and only if its spectrum consists of impulses,

$$P_{x_p}(e^{j\omega}) = \sum_{k=1}^{N} \alpha_k u_0(\omega - \omega_k)$$

Examples of predictable processes include the random phase sinusoid and the complex harmonic process given in Example 3.3.1.

Wold Decomposition Theorem. A general random process can be written as a sum of two processes,

$$x(n) = x_p(n) + x_r(n)$$

where $x_r(n)$ is a regular random process and $x_p(n)$ is a predictable process, with $x_r(n)$ being *orthogonal* to $x_p(n)$,

$$E\{x_r(m)x_p^*(n)\} = 0$$

As a corollary to the Wold decomposition theorem, it follows that the general form for the power spectrum of a wide sense stationary process is

$$P_x(e^{j\omega}) = P_{x_r}(e^{j\omega}) + \sum_{k=1}^{N} \alpha_k u_0(\omega - \omega_k) \tag{3.106}$$

where $P_{x_r}(e^{j\omega})$ is the continuous part of the spectrum corresponding to a regular process and where the sum of weighted impulses that give rise to the line spectrum represent the predictable part of the process [6].

3.6 SPECIAL TYPES OF RANDOM PROCESSES

In this section, we look at the characteristics and properties of some of the random processes that we will be encountering in the following chapters. We begin by looking at those processes that may be generated by filtering white noise with a linear shift-invariant filter that has a rational system function. These include the autoregressive, moving average, and autoregressive moving average processes. Of interest will be the form of the autocorrelation sequence and the power spectrum of these processes. We will show, for example, that the autocorrelation sequences of these processes satisfy a set of equations, known as the Yule-Walker equations, relating $r_x(k)$ to the parameters of the filter. In addition to the filtered white noise processes, we will also look at harmonic processes that consist of a sum of random phase sinusoids or complex exponentials. These processes are important in applications where signals have periodic components.

3.6.1 Autoregressive Moving Average Processes

Suppose that we filter white noise $v(n)$ with a causal linear shift-invariant filter having a rational system function with p poles and q zeros

$$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k) z^{-k}}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} \tag{3.107}$$

Assuming that the filter is stable, the output process $x(n)$ will be wide-sense stationary and, with $P_v(z) = \sigma_v^2$, the power spectrum will be

$$P_x(z) = \sigma_v^2 \frac{B_q(z) B_q^*(1/z^*)}{A_p(z) A_p^*(1/z^*)}$$

or, in terms of ω,

$$\boxed{P_x(e^{j\omega}) = \sigma_v^2 \frac{|B_q(e^{j\omega})|^2}{|A_p(e^{j\omega})|^2}} \tag{3.108}$$

A process having a power spectrum of this form is known as an *autoregressive moving average process* of order (p, q) and is referred to as an ARMA(p, q) process. Note that the power spectrum of an ARMA(p, q) process contains $2p$ poles and $2q$ zeros with the reciprocal symmetry discussed in Section 3.4.

Example 3.6.1 *An ARMA(2,2) Random Process*

The power spectrum of an ARMA(2,2) process is shown in Fig. 3.11. This process is generated by filtering white noise $v(n)$ with a linear shift-invariant system having two poles

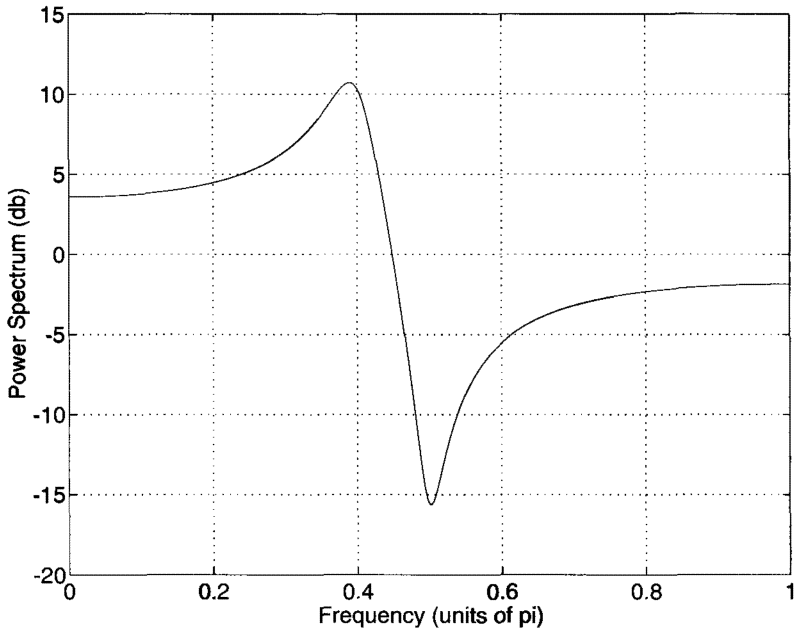


Figure 3.11 *The power spectrum of an ARMA(2,2) random process with zeros at* $z = 0.95e^{\pm j\pi/2}$ *and poles at* $z = 0.9e^{\pm j2\pi/5}$.

and two zeros. The zeros of $H(z)$ are at $z = 0.95e^{\pm j\pi/2}$, and the poles are at $z = 0.9e^{\pm j2\pi/5}$. Thus,

$$H(z) = \frac{1 + 0.9025z^{-2}}{1 - 0.5562z^{-1} + .81z^{-2}}$$

Note that there is a peak in the power spectrum close to $\omega = 0.4\pi$ due to the poles in the denominator of $H(z)$ and a null in the spectrum close to $\omega = 0.5\pi$ due to the zeros of $H(z)$.

Since $x(n)$ and $v(n)$ are related by the linear constant coefficient difference equation

$$x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) = \sum_{l=0}^{q} b_q(l)v(n-l) \tag{3.109}$$

it follows that the autocorrelation of $x(n)$ and the cross-correlation between $x(n)$ and $v(n)$ satisfy the same difference equation. To see this, note that if we multiply both sides of Eq. (3.109) by $x^*(n-k)$ and take the expected value, then we have

$$r_x(k) + \sum_{l=1}^{p} a_p(l)r_x(k-l) = \sum_{l=0}^{q} b_q(l)E\left\{v(n-l)x^*(n-k)\right\} \tag{3.110}$$

Since $v(n)$ is WSS, it follows that $v(n)$ and $x(n)$ are jointly WSS (see Section 3.4, p. 100) and

$$E\left\{v(n-l)x^*(n-k)\right\} = r_{vx}(k-l)$$

Therefore, Eq. (3.110) becomes

$$\boxed{r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \sum_{l=0}^{q} b_q(l) r_{vx}(k-l)} \tag{3.111}$$

which is a difference equation of the same form as Eq. (3.109). In its present form, this difference equation is not very useful. However, by writing the cross-correlation $r_{vx}(k)$ in terms of autocorrelation $r_x(k)$ and the unit sample response of the filter, we may derive a set of equations relating the autocorrelation of $x(n)$ to the filter. Since

$$x(n) = h(n) * v(n) = \sum_{m=-\infty}^{\infty} v(m) h(n-m)$$

the cross-correlation may be written as

$$\begin{aligned} E\{v(n-l)x^*(n-k)\} &= E\left\{ \sum_{m=-\infty}^{\infty} v(n-l)v^*(m)h^*(n-k-m) \right\} \\ &= \sum_{m=-\infty}^{\infty} E\{v(n-l)v^*(m)\} h^*(n-k-m) \\ &= \sigma_v^2 h^*(l-k) \end{aligned} \tag{3.112}$$

where the last equality follows from the fact that $E\{v(n-l)v^*(m)\} = \sigma_v^2 \delta(n-l-m)$, i.e., $v(n)$ is white noise. Substituting Eq. (3.112) into (3.111) gives

$$r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \sigma_v^2 \sum_{l=0}^{q} b_q(l) h^*(l-k) \tag{3.113}$$

Assuming that $h(n)$ is causal, the sum on the right side of Eq. (3.113), which we denote by $c_q(k)$, may be written as

$$\boxed{c_q(k) = \sum_{l=k}^{q} b_q(l) h^*(l-k) = \sum_{l=0}^{q-k} b_q(l+k) h^*(l)} \tag{3.114}$$

Since $c_q(k) = 0$ for $k > q$, we may write Eq. (3.113) for $k \geq 0$, as follows

$$\boxed{r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \begin{cases} \sigma_v^2 c_q(k) & ; \quad 0 \leq k \leq q \\ 0 & ; \quad k > q \end{cases}} \tag{3.115}$$

which are the *Yule-Walker equations*. Writing these equations for $k = 0, 1, \ldots, p+q$ in

matrix form we have

$$\left[\begin{array}{cccc} r_x(0) & r_x(-1) & \cdots & r_x(-p) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\ \vdots & \vdots & & \vdots \\ r_x(q) & r_x(q-1) & \cdots & r_x(q-p) \\ \hline r_x(q+1) & r_x(q) & \cdots & r_x(q-p+1) \\ \vdots & \vdots & & \vdots \\ r_x(q+p) & r_x(q+p-1) & \cdots & r_x(q) \end{array}\right] \left[\begin{array}{c} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \sigma_v^2 \left[\begin{array}{c} c_q(0) \\ c_q(1) \\ \vdots \\ c_q(q) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] \tag{3.116}$$

Note that Eq. (3.115) defines a recursion for the autocorrelation sequence in terms of the filter coefficients, $a_p(k)$ and $b_q(k)$.[9] Therefore, the Yule-Walker equations may be used to extrapolate the autocorrelation sequence from a finite set of values of $r_x(k)$. For example, if $p \geq q$ and if $r_x(0), \ldots, r_x(p-1)$ are known, then $r_x(k)$ for $k \geq p$ may be computed recursively using the difference equation

$$r_x(k) = -\sum_{l=1}^{p} a_p(l) r_x(k-l)$$

Note that the Yule-Walker equations also provide a relationship between the filter coefficients and the autocorrelation sequence. Thus, Eq. (3.115) may be used to estimate the filter coefficients, $a_p(k)$ and $b_q(k)$, from the autocorrelation sequence $r_x(k)$. However, due to the product $h^*(l)b_q(k+l)$ that appears in Eq. (3.114), the Yule-Walker equations are *nonlinear* in the filter coefficients and solving them for the filter coefficients is, in general, difficult. As we will see in later chapters, the Yule-Walker equations are important in problems such as signal modeling and spectrum estimation.

In the next two sections, we look at two special cases of ARMA processes. The first is when $q = 0$ (autoregressive process), and the second is when $p = 0$ (moving average process).

3.6.2 Autoregressive Processes

A special type of ARMA(p, q) process results when $q = 0$ in Eq. (3.107), i.e., when $B_q(z) = b(0)$. In this case $x(n)$ is generated by filtering white noise with an all-pole filter of the form

$$H(z) = \frac{b(0)}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} \tag{3.117}$$

An ARMA$(p, 0)$ process is called an *autoregressive process* of order p and will be referred to as an AR(p) process. From Eq. (3.92) it follows that if $P_v(z) = \sigma_v^2$, then the power spectrum of $x(n)$ is

$$P_x(z) = \sigma_v^2 \frac{|b(0)|^2}{A_p(z) A_p^*(1/z^*)} \tag{3.118}$$

[9]The unit sample response $h(k)$ may be computed if the filter coefficients $a_p(k)$ and $b_q(k)$ are known.

or, in terms of the frequency variable ω,

$$\boxed{P_x(e^{j\omega}) = \sigma_v^2 \frac{|b(0)|^2}{|A_p(e^{j\omega})|^2}} \tag{3.119}$$

Thus, the power spectrum of an AR(p) process contains $2p$ poles and no zeros (except those located at $z = 0$ and $z = \infty$). The Yule-Walker equations for an AR(p) process may be found from Eq. (3.115) by setting $q = 0$. With $c_0(0) = b(0)h^*(0) = |b(0)|^2$ these equations are

$$\boxed{r_x(k) + \sum_{l=1}^{p} a_p(l) r_x(k-l) = \sigma_v^2 |b(0)|^2 \delta(k) \quad ; \quad k \geq 0} \tag{3.120}$$

which, in matrix form become

$$\begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\ \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \sigma_v^2 |b(0)|^2 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.121}$$

Note that since the Yule-Walker equations are *linear* in the coefficients $a_p(k)$, it is a simple matter to find the coefficients $a_p(k)$ from the autocorrelation sequence $r_x(k)$. For example, suppose that we are given the first two autocorrelation values of a real-valued AR(1) process. Assuming that $\sigma_v^2 = 1$, using the property that $r_x(k) = r_x(-k)$ for real processes the Yule-Walker equations for $k = 0, 1$ are as follows:

$$\begin{aligned} r_x(0) + r_x(1)a(1) &= b^2(0) \\ r_x(0)a(1) &= -r_x(1) \end{aligned}$$

From the second equation, we find for $a(1)$

$$a(1) = -\frac{r_x(1)}{r_x(0)}$$

and, from the first equation, we may solve for $b(0)$,

$$b^2(0) = r_x(0) + r_x(1)a(1) = \frac{r_x^2(0) - r_x^2(1)}{r_x(0)}$$

Expressed in terms of $r_x(0)$ and $r_x(1)$, a first-order filter that generates an AR(1) process with the given autocorrelation values is

$$H(z) = \frac{\sqrt{r_x(0)\left[r_x^2(0) - r_x^2(1)\right]}}{r_x(0) - r_x(1)z^{-1}}$$

In a similar fashion, we may use the Yule-Walker equations to generate the autocorrelation sequence from a given set of filter coefficients. For example, again let $x(n)$ be a first-order autoregressive process. Writing the first two Yule-Walker equations in matrix form in terms of the unknowns $r_x(0)$ and $r_x(1)$ we have

$$\begin{bmatrix} 1 & a(1) \\ a(1) & 1 \end{bmatrix} \begin{bmatrix} r_x(0) \\ r_x(1) \end{bmatrix} = \begin{bmatrix} b^2(0) \\ 0 \end{bmatrix}$$

Solving for $r_x(0)$ and $r_x(1)$ we find

$$r_x(0) = \frac{b^2(0)}{1 - a^2(1)}$$

$$r_x(1) = -a(1)r_x(0) = -a(1)\frac{b^2(0)}{1 - a^2(1)}$$

Now observe from Eq. (3.120) that for all $k > 0$

$$r_x(k) = -a(1)r_x(k-1).$$

Therefore, the autocorrelation sequence for $k \geq 0$ may be written as

$$r_x(k) = \frac{b^2(0)}{1 - a^2(1)}\Big(-a(1)\Big)^k \quad ; \quad k \geq 0$$

Finally, using the symmetry of $r_x(k)$ gives

$$\boxed{r_x(k) = \frac{b^2(0)}{1 - a^2(1)}\big[-a(1)\big]^{|k|}} \tag{3.122}$$

We now look at specific examples of AR(1) and AR(2) processes.

Example 3.6.2 ***AR(1) and AR(2) Random Processes***

Consider the real AR(1) process that is formed by filtering unit variance white noise with the first-order all-pole filter

$$H(z) = \frac{b(0)}{1 - a(1)z^{-1}}$$

where $b(0)$ and $a(1)$ are real-valued coefficients. The power spectral density of this process is

$$P_x(z) = \frac{b^2(0)}{[1 - a(1)z^{-1}][1 - a(1)z]} = \frac{b^2(0)}{1 + a^2(1) - a(1)[z + z^{-1}]}$$

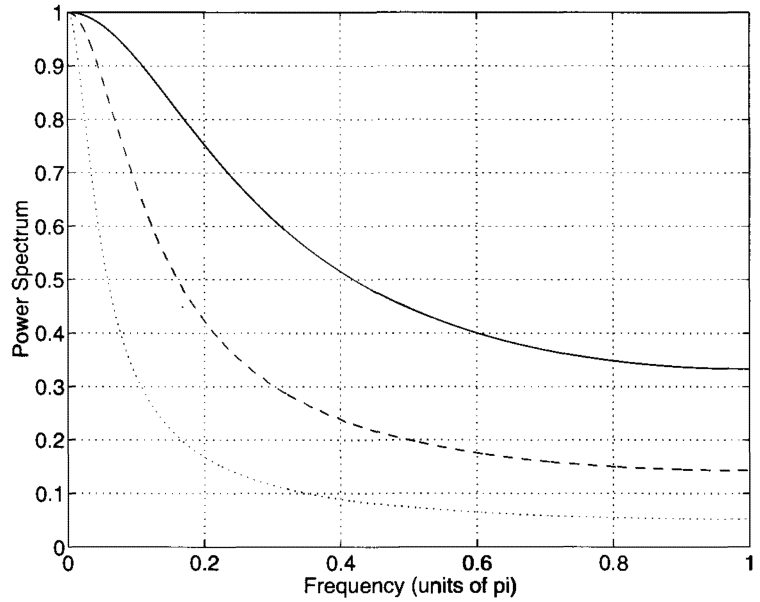
or, in terms of Fourier transforms,

$$P_x(e^{j\omega}) = \frac{b^2(0)}{[1 - a(1)e^{-j\omega}][1 - a(1)e^{j\omega}]} = \frac{b^2(0)}{1 + a^2(1) - 2a(1)\cos\omega}$$

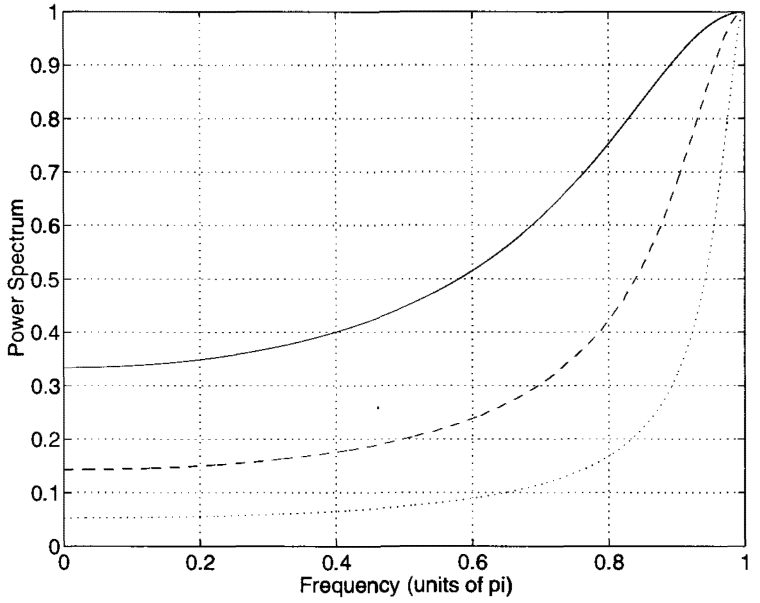
The power spectrum of this process is shown in Fig. 3.12*a* for $a(1) = 0.5$, 0.75, and 0.9 with $b(0)$ chosen so that the peak at $\omega = 0$ is equal to one. Note that the power in $x(n)$ is concentrated in the low frequencies and, as a result, $x(n)$ is referred to as a low-pass process. Also note that as the pole moves closer to the unit circle the process bandwidth decreases. Now consider what happens if $a(1) < 0$. The power spectrum in this case is shown in Fig. 3.12*b* for $a(1) = -0.5$, -0.75, and -0.9 with the value of $b(0)$ again chosen so that the peak at $\omega = \pi$ is equal to one. Since the power in $x(n)$ is now concentrated in the high frequencies, $x(n)$ is referred to as a high-pass process.

Now consider the AR(2) process that is generated by filtering white noise with the second-order filter

$$H(z) = \frac{b(0)}{[1 - a(1)z^{-1}][1 - a^*(1)z^{-1}]}$$



(*a*) The power spectrum of a low-pass AR(1) random process with a pole at $z = 0.5$ (solid line), a pole at $z = 0.75$ (dashed line), and a pole at $z = 0.9$ (dotted line).



(*b*) The power spectrum for a high-pass AR(1) random process, pole at $z = -0.5$ (solid line), pole at $z = -0.75$ (dashed line), and pole at $z = -0.9$ (dotted line).

Figure 3.12 *The power spectrum of first-order autoregressive processes. In each case, the numerator coefficient, $b(0)$, is chosen so that the power spectrum is normalized to one at $\omega = 0$.*

With $a(1) = re^{j\omega_0}$ and $\omega_0 = \pi/2$, the power spectrum is shown in Fig. 3.13 for $r = 0.5$, 0.75, and 0.9 with the value of $b(0)$ chosen so that the peak value of the power spectrum is equal to one. Note that the power in $x(n)$ is now concentrated within a band of frequencies centered around ω_0 and that as the pole moves closer to the unit circle the process bandwidth decreases. A process of this form is referred to as a *bandpass* process.

3.6.3 Moving Average Processes

Another special case of ARMA(p, q) process results when $p = 0$. With $p = 0$, $x(n)$ is generated by filtering white noise with an FIR filter that has a system function of the form

$$H(z) = \sum_{k=0}^{q} b_q(k) z^{-k}$$

i.e., $A_p(z) = 1$ in Eq. (3.107). An ARMA$(0, q)$ process is known as a *moving average process* of order q and will be referred to as an MA(q) process. From Eq. (3.92) note that if $P_v(z) = \sigma_v^2$, then the power spectrum is

$$P_x(z) = \sigma_v^2 B_q(z) B_q^*(1/z^*) \tag{3.123}$$

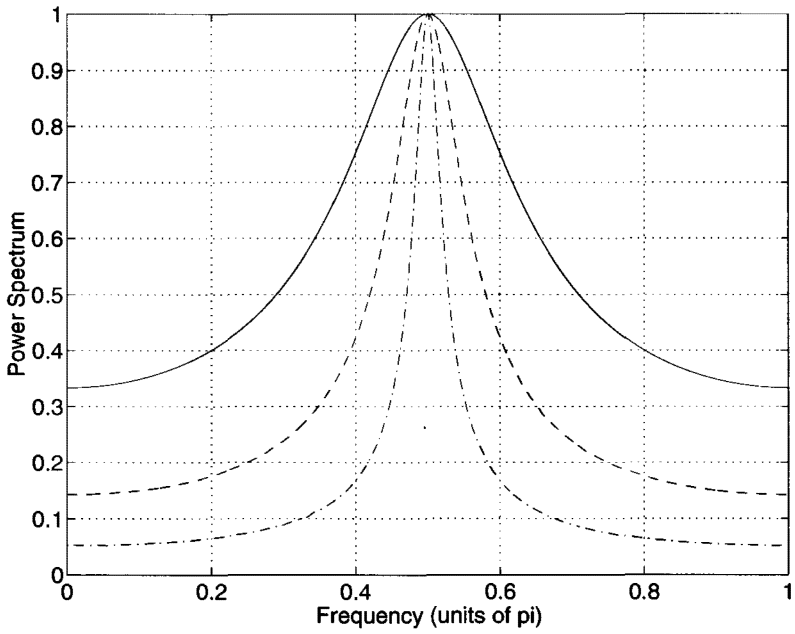


Figure 3.13 *The power spectrum of an AR(2) random process where the filter generating the process has a complex pole pair at $z = re^{\pm j\omega_0}$ where $\omega_0 = \pi/2$ and $r = 0.5$ (solid line), $r = 0.75$ (dashed line), and $r = 0.9$ (dashdot line). The numerator $b(0)$ is chosen so that the peak value of the power spectrum is normalized to one.*

or, in terms of ω,

$$\boxed{P_x(e^{j\omega}) = \sigma_v^2 \left| B_q(e^{j\omega}) \right|^2} \tag{3.124}$$

Therefore, $P_x(z)$ contains $2q$ zeros and no poles (except those that lie at $z = 0$ and $z = \infty$). The Yule-Walker equations for an MA(q) process may be found by simply taking the inverse z-transform of Eq. (3.123). Alternatively, they may be found from Eq. (3.115) by setting $a_p(k) = 0$ and noting that since $h(n) = b(n)$, then

$$c_q(k) = \sum_{l=0}^{q-k} b_q(l+k)b^*(l)$$

In either case we have

$$\boxed{r_x(k) = \sigma_v^2 b_q(k) * b_q^*(-k) = \sigma_v^2 \sum_{l=0}^{q-|k|} b_q(l+|k|)b_q^*(l)} \tag{3.125}$$

where we have used the absolute value on k so that the expression for $r_x(k)$ is valid for all k. Thus, we see that the autocorrelation sequence of an MA(q) process is equal to zero for all values of k that lie outside the interval $[-q, q]$. In addition, note that $r_x(k)$ depends nonlinearly on the moving average parameters, $b_q(k)$. For example, for $q = 2$

$$\begin{aligned} r_x(0) &= |b(0)|^2 + |b(1)|^2 + |b(2)|^2 \\ r_x(1) &= b^*(0)b(1) + b^*(1)b(2) \\ r_x(2) &= b^*(0)b(2) \end{aligned} \tag{3.126}$$

Therefore, unlike the case for an autoregressive process, estimating the moving average parameters for a moving average process is a nontrivial problem.

Moving average processes are characterized by slowly changing functions of frequency that will have sharp nulls in the spectrum if $P_x(z)$ contains zeros that are close to the unit circle. The following example provides an illustration of this behavior for an MA(4) process.

Example 3.6.3 *An MA(4) Process*

Shown in Fig. 3.14 is the power spectrum of an MA(4) process that is generated by filtering white noise with a filter having a pair of complex zeros at $z = e^{\pm j\pi/2}$ and a pair of complex zeros at $z = 0.8e^{\pm j3\pi/4}$. Thus, the autocorrelation sequence is equal to zero for $|k| > 4$. Due to the zeros on the unit circle, the power spectrum goes to zero at $\omega = \pm\pi/2$. In addition, due to the zeros at a radius of 0.8 there is an additional null in the spectrum close to $\omega = 3\pi/4$.

3.6.4 Harmonic Processes

Harmonic processes provide useful representations for random processes that arise in applications such as array processing, when the signals contain periodic components. An example of a wide sense stationary harmonic process is the random phase sinusoid

$$x(n) = A\sin(n\omega_0 + \phi)$$

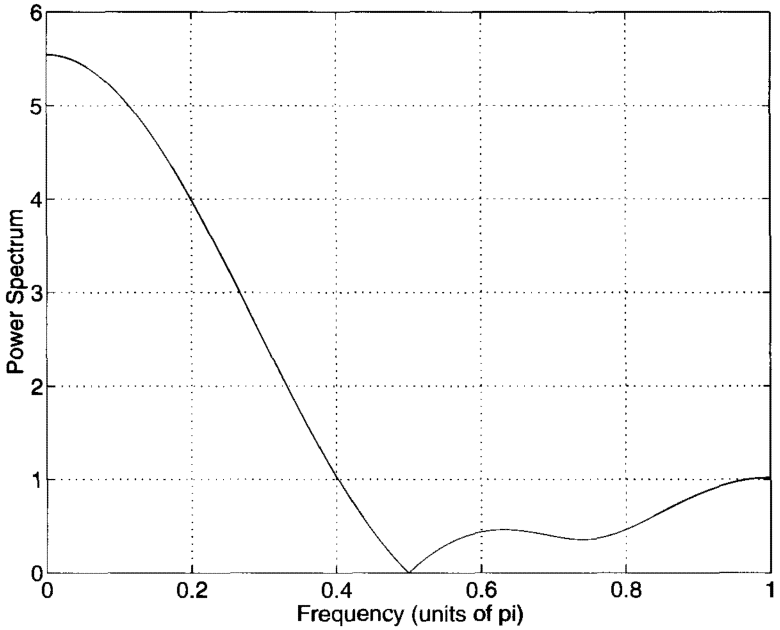


Figure 3.14 *The power spectrum of an MA(4) process having zeros at* $z = e^{\pm j\pi/2}$ *and* $z = 0.8e^{\pm j3\pi/4}$.

where A and ω_0 are constants and ϕ is a random variable that is uniformly distributed between $-\pi$ and π. As shown in Example 3.3.1, the autocorrelation sequence of $x(n)$ is periodic with frequency ω_0,

$$r_x(k) = \tfrac{1}{2}A^2 \cos(k\omega_0) \tag{3.127}$$

Therefore, the power spectrum is

$$P_x(e^{j\omega}) = \frac{\pi}{2}A^2\Big[u_0(\omega - \omega_0) + u_0(\omega + \omega_0)\Big]$$

where $u_0(\omega - \omega_0)$ is an impulse at frequency ω_0 and $u_0(\omega + \omega_0)$ is an impulse at $-\omega_0$. If the amplitude is also a random variable that is uncorrelated with ϕ, then the autocorrelation sequence will be the same as Eq. (3.127) with A^2 replaced with $E\left\{A^2\right\}$,

$$r_x(k) = \tfrac{1}{2}E\left\{A^2\right\}\cos(k\omega_0)$$

Higher-order harmonic processes may be formed from a sum of random phase sinusoids as follows:

$$x(n) = \sum_{l=1}^{L} A_l \sin(n\omega_l + \phi_l)$$

Assuming the random variables ϕ_l and A_l are uncorrelated, the autocorrelation sequence is

$$r_x(k) = \sum_{l=1}^{L} \tfrac{1}{2}E\left\{A_l^2\right\}\cos(k\omega_l)$$

and the power spectrum is

$$P_x(e^{j\omega}) = \sum_{l=1}^{L} \tfrac{1}{2}\pi E\{A_l^2\}\left[u_0(\omega - \omega_l) + u_0(\omega + \omega_l)\right]$$

In the case of complex signals, a first order harmonic process has the form

$$x(n) = |A|e^{j(n\omega_0+\phi)}$$

where ϕ, and possibly $|A|$, are random variables. Similar to what we found in the case of a sinusoid, if ϕ and A are uncorrelated random variables, then the autocorrelation sequence is a complex exponential with frequency ω_0,

$$r_x(k) = E\{|A|^2\}\exp(jk\omega_0)$$

and the power spectrum is

$$P_x(e^{j\omega}) = 2\pi E\{|A|^2\}u_0(\omega - \omega_0)$$

With a sum of L uncorrelated harmonic processes

$$x(n) = \sum_{l=1}^{L} A_l \exp[j(n\omega_l + \phi_l)]$$

the autocorrelation sequence is

$$r_x(k) = \sum_{l=1}^{L} E\{|A_l|^2\}\exp[jk\omega_l]$$

and the power spectrum is

$$P_x(e^{j\omega}) = \sum_{l=1}^{L} 2\pi E\{|A_l|^2\}u_0(\omega - \omega_l)$$

3.7 SUMMARY

This chapter provided an introduction to the basic theory of discrete-time random processes. Beginning with a review of random variables, we then introduced a discrete-time random process as an indexed sequence of random variables. Although a complete description of a random process requires the specification of the joint distribution or density functions, our concern in this book is primarily with ensemble averages. Therefore, we proceeded to define the mean, the variance, and the autocorrelation of a discrete-time random process. Next, we introduced the notion of stationarity and defined a process to be wide-sense stationary (WSS) if its mean is constant and autocorrelation $E\{x(k)x^*(l)\}$ depends only on the difference, $k - l$, between the times of the samples $x(k)$ and $x(l)$. Although it is not generally possible to assume, in practice, that a process is wide-sense stationary, in some cases it is safe to assume that it is wide-sense stationary over a sufficiently short time interval. Throughout most of this book, the processes that we will be considering will be assumed to be wide-sense stationary. However, in Chapter 9 we will take a broader view of statistical signal processing and look at the problem of filtering nonstationary random processes.

In many of the problems that we will be considering in this book it will be necessary to know the first- and second-order statistics of a process. Since these statistics are not generally known a priori, we will find it necessary to estimate these statistics from a single realization of a process. This leads us to the notion of ergodicity, which is concerned with the problem of estimating the statistical properties of a random process from time averages of a single realization. Although a proper treatment of ergodicity requires a deep knowledge of probability theory, we introduced two useful and interesting mean ergodic theorems and an autocorrelation ergodic theorem for Gaussian processes. Instead of concerning ourselves with whether or not the conditions of the ergodic theorems are satisfied (interestingly, these theorems cannot be applied without some a priori knowledge of higher order statistics), in this book it will be assumed that when an ensemble average is required it may be estimated by simply taking an appropriate time average. The ultimate justification for this assumption, in practice, will be in the performance of the algorithm that uses these estimates.

Another important statistical quantity introduced in this chapter is the power spectrum, which is the discrete-time Fourier transform of a wide-sense stationary process. Just as with deterministic signals, the power spectrum is an important tool in the description, modeling, and analysis of wide-sense stationary processes. So important and useful is the power spectrum, in fact, that an entire chapter will be devoted to the problem of power spectrum estimation. We also developed the very important *spectral factorization theorem*, which gives us insight into how to design filters to whiten a wide-sense stationary process or to generate a process with a given rational power spectrum by filtering white noise.

Next we looked at the process of filtering a wide-sense stationary process with a discrete-time filter. Of particular importance is how the mean, autocorrelation, and power spectrum of the input process are modified by the filter. The fundamental result here was that the power spectrum of the input process is multiplied by the magnitude squared of the frequency response of the filter. Finally, we concluded the chapter with a listing of some of the important types of processes that will be encountered in this book. These processes include those that result from filtering white noise with a linear shift-invariant filter. Depending upon whether the filter is FIR, all-pole, or a general pole-zero filter, we saw that it was possible to generate moving average, autoregressive, and autoregressive moving average processes, respectively.

References

1. R. V. Churchill and J. W. Brown, *Introduction to Complex Variables and Applications*, 4th ed., McGraw-Hill, New York, 1984.
2. S. Haykin, *Modern Filters*, MacMillan, New York, 1989.
3. H. L. Larson and B. O. Shubert, *Probabilistic Models in Engineering Sciences: Vol. I, Random Variables and Stochastic Processes*, John Wiley & Sons, New York, 1979.
4. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
5. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, New York, 1991.
6. A. Papoulis, "Predictable processes and Wold's decomposition: A review," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. 33, no. 4, pp. 933–938, August, 1985.
7. B. Picinbono, *Random Signals and Noise*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

8. B. Porat, *Digital Processing of Random Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
9. I. S. Reed, "On a moment factoring theorem for complex Gaussian processes," *IRE Trans. Inf. Theory*, vol. IT-8, pp. 194–195.
10. C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
11. H. Wold, *The Analysis of Stationary Time Series*, 2nd ed., Almquist and Wicksell, Uppsala, Sweden, 1954 (originally published in 1934).

3.8 PROBLEMS

3.1. Let x be a random variable with a mean m_x and variance σ_x^2. Let x_i for $i = 1, 2, \ldots, N$ be N independent measurements of the random variable x.

(a) With $\hat{m}_x$ the sample mean defined by

$$\hat{m}_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

determine whether or not the sample variance

$$\hat{\sigma}_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{m}_x)^2$$

is unbiased, i.e., is $E\{\hat{\sigma}_x^2\} = \sigma_x^2$?

(b) If x is a Gaussian random variable, find the variance of the sample variance,

$$E\{(\hat{\sigma}_x^2 - E\{\hat{\sigma}_x^2\})^2\}.$$

3.2. Let $x(n)$ be a stationary random process with zero mean and autocorrelation $r_x(k)$. We form the process, $y(n)$, as follows:

$$y(n) = x(n) + f(n)$$

where $f(n)$ is a known deterministic sequence. Find the mean $m_y(n)$ and the autocorrelation $r_y(k, l)$ of the process $y(n)$.

3.3. A discrete-time random process $x(n)$ is generated as follows:

$$x(n) = \sum_{k=1}^{p} a(k)x(n-k) + w(n)$$

where $w(n)$ is a white noise process with variance σ_w^2. Another process, $z(n)$, is formed by adding noise to $x(n)$,

$$z(n) = x(n) + v(n)$$

where $v(n)$ is white noise with a variance of σ_v^2 that is uncorrelated with $w(n)$.

(a) Find the power spectrum of $x(n)$.

(b) Find the power spectrum of $z(n)$.

3.4. Suppose we are given a linear shift-invariant system having a system function

$$H(z) = \frac{1 - \frac{1}{2}z^{-1}}{1 - \frac{1}{3}z^{-1}}$$

that is excited by zero mean exponentially correlated noise $x(n)$ with an autocorrelation sequence

$$r_x(k) = \left(\tfrac{1}{2}\right)^{|k|}$$

Let $y(n)$ be the output process, $y(n) = x(n) * h(n)$.

(a) Find the power spectrum, $P_y(z)$, of $y(n)$.

(b) Find the autocorrelation sequence, $r_y(k)$, of $y(n)$.

(c) Find the cross-correlation, $r_{xy}(k)$, between $x(n)$ and $y(n)$.

(d) Find the *cross-power spectral density*, $P_{xy}(z)$, which is the z transform of the cross-correlation $r_{xy}(k)$.

3.5. Find the power spectrum for each of the following wide-sense stationary random processes that have the given autocorrelation sequences.

(a) $r_x(k) = 2\delta(k) + j\delta(k-1) - j\delta(k+1)$.

(b) $r_x(k) = \delta(k) + 2(0.5)^{|k|}$.

(c) $r_x(k) = 2\delta(k) + \cos(\pi k/4)$.

(d) $r_x(k) = \begin{cases} 10 - |k| & ; \quad |k| < 10 \\ 0 & ; \quad \text{otherwise} \end{cases}$

3.6. Find the autocorrelation sequence corresponding to each of the following power spectral densities.

(a) $P_x(e^{j\omega}) = 3 + 2\cos\omega$.

(b) $P_x(e^{j\omega}) = \dfrac{1}{5 + 3\cos\omega}$.

(c) $P_x(z) = \dfrac{-2z^2 + 5z - 2}{3z^2 + 10z + 3}$.

3.7. Let $x(n)$ be a zero mean WSS process with an $N \times N$ autocorrelation matrix $\mathbf{R}_x$. Determine whether each of the following statements are *True* or *False*.

(a) If the eigenvalues of $\mathbf{R}_x$ are equal, $\lambda_1 = \lambda_2 = \cdots = \lambda_N$, then $r_x(k) = 0$ for $k = 1, 2, \ldots, N$.

(b) If $\lambda_1 > 0$ and $\lambda_k = 0$ for $k = 2, 3, \ldots, N$, then $r_x(k) = Ae^{jk\omega_0}$.

3.8. Consider the random process

$$x(n) = A\cos(n\omega_0 + \phi) + w(n)$$

where $w(n)$ is zero mean white Gaussian noise with a variance σ_w^2. For each of the following cases, find the autocorelation sequence and power spectrum of $x(n)$.

(a) A is a Gaussian random variable with zero mean and variance σ_A^2 and both ω_0 and ϕ are constants.

(b) ϕ is uniformly distributed over the interval $[-\pi, \pi]$ and both A and ω_0 are constants.

(c) ω_0 is a random variable that is uniformly distributed over the interval $[\omega_0 - \Delta, \omega_0 + \Delta]$ and both A and ϕ are constants.

3.9. Determine whether or not each of the following are valid autocorrelation matrices. If they are not, explain why not.

(a) $\mathbf{R}_1 = \begin{bmatrix} 4 & 1 & 1 \\ -1 & 4 & 1 \\ -1 & -1 & 4 \end{bmatrix}$

(b) $\mathbf{R}_2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

(c) $\mathbf{R}_3 = \begin{bmatrix} 1 & 1+j \\ 1-j & 1 \end{bmatrix}$

(d) $\mathbf{R}_4 = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix}$

(e) $\mathbf{R}_5 = \begin{bmatrix} 2 & j & 1 \\ -j & 4j & -j \\ 1 & j & 2 \end{bmatrix}$

3.10. The input to a linear shift-invariant filter with unit sample response

$$h(n) = \delta(n) + \tfrac{1}{2}\delta(n-1) + \tfrac{1}{4}\delta(n-2)$$

is a zero mean wide-sense stationary process with autocorrelation

$$r_x(k) = \left(\tfrac{1}{2}\right)^{|k|}$$

(a) What is the variance of the output process?

(b) Find the autocorrelation of the output process, $r_y(k)$, for all k.

3.11. Consider a first-order AR process that is generated by the difference equation

$$y(n) = ay(n-1) + w(n)$$

where $|a| < 1$ and $w(n)$ is a zero mean white noise random process with variance σ_w^2.

(a) Find the unit sample response of the filter that generates $y(n)$ from $w(n)$.

(b) Find the autocorrelation of $y(n)$.

(c) Find the power spectrum of $y(n)$.

3.12. Consider an MA(q) process that is generated by the difference equation

$$y(n) = \sum_{k=0}^{q} b(k)w(n-k)$$

where $w(n)$ is zero mean white noise with variance σ_w^2.

(a) Find the unit sample response of the filter that generates $y(n)$ from $w(n)$.

(b) Find the autocorrelation of $y(n)$.

(c) Find the power spectrum of $y(n)$.

3.13. Suppose we are given a zero mean process $x(n)$ with autocorrelation

$$r_x(k) = 10\left(\tfrac{1}{2}\right)^{|k|} + 3\left(\tfrac{1}{2}\right)^{|k-1|} + 3\left(\tfrac{1}{2}\right)^{|k+1|}$$

(a) Find a filter which, when driven by unit variance white noise, will yield a random process with this autocorrelation.

(b) Find a stable and causal filter which, when excited by $x(n)$, will produce zero mean, unit variance, white noise.

3.14. For each of the following, determine whether or not the random process is

i. Wide-sense stationary.

ii. Mean ergodic.

(a) $x(n) = A$ where A is a random variable with probability density function $f_A(\alpha)$.

(b) $x(n) = A\cos(n\omega_0)$ where A is a Gaussian random variable with mean m_A and variance σ_A^2.

(c) $x(n) = A\cos(n\omega_0 + \phi)$ where ϕ is a random variable that is uniformly distributed between $-\pi$ and π.

(d) $x(n) = A\cos(n\omega_0) + B\sin(n\omega_0)$ where A and B are uncorrelated zero mean random variables with variance σ^2.

(e) A Bernoulli process with $\Pr\{x(n) = 1\} = p$ and $\Pr\{x(n) = -1\} = 1 - p$.

(f) $y(n) = x(n) - x(n-1)$ where $x(n)$ is the Bernoulli process defined in part (e).

3.15. Determine which of the following correspond to a valid autocorrelation sequence for a WSS random process. For those that are not valid, state why not. For those that are valid, describe a way for generating a process with the given autocorrelation.

(a) $r_x(k) = \delta(k-1) + \delta(k+1)$

(b) $r_x(k) = 3\delta(k) + 2\delta(k-1) + 2\delta(k+1)$

(c) $r_x(k) = \exp(jk\pi/4)$

(d) $r_x(k) = \begin{cases} 1 & ; \quad |k| < N \\ 0 & ; \quad \text{else} \end{cases}$

(e) $r_x(k) = \begin{cases} \dfrac{N - |k|}{N} & ; \quad |k| < N \\ 0 & ; \quad \text{else} \end{cases}$

(f) $r_x(k) = 2^{-k^2}$

3.16. Show that the cross-correlation, $r_{xy}(k)$, between two jointly wide-sense stationary processes $x(n)$ and $y(n)$ satisfies the following inequalities,

(a) $|r_{xy}(k)| \leq \left[r_x(0)r_y(0)\right]^{1/2}$

(b) $|r_{xy}(k)| \leq \frac{1}{2}\left[r_x(0) + r_y(0)\right]$

3.17. Given a wide-sense stationary random process $x(n)$, we would like to design a "linear predictor" that will predict the value of $x(n+1)$ using a linear combination of $x(n)$ and $x(n-1)$. Thus, our predictor for $x(n+1)$ is of the form

$$\hat{x}(n+1) = ax(n) + bx(n-1)$$

where a and b are constants. Assume that the process has zero mean

$$E\{x(n)\} = 0$$

and that we want to minimize the mean-square error

$$\xi = E\left\{\left[x(n+1) - \hat{x}(n+1)\right]^2\right\}$$

(a) With $r_x(k)$ the autocorrelation of $x(n)$, determine the optimum predictor of $x(n)$ by finding the values of a and b that minimize the mean-square error.

(b) What is the minimum mean-square error of the predictor? Express your answer in terms of the autocorrelation $r_x(k)$.

(c) If $x(n+1)$ is uncorrelated with $x(n)$, what form does your predictor take?

(d) If $x(n+1)$ is uncorrelated with both $x(n)$ and $x(n-1)$, what form does your predictor take?

3.18. *True* or *False:* If $x(n)$ is a WSS process and $y(n)$ is the process that is formed by filtering $x(n)$ with a stable, linear shift-invariant filter $h(n)$, then

$$\sigma_y^2 = \sigma_x^2 \sum_{n=-\infty}^{\infty} |h(n)|^2$$

where σ_x^2 and σ_y^2 are the variances of the processes $x(n)$ and $y(n)$, respectively.

3.19. Show that a sufficient condition for a wide-sense stationary process to be ergodic in the mean is that the autocovariance be absolutely summable,

$$\sum_{k=-\infty}^{\infty} |c_x(k)| < \infty$$

3.20. For each of the following, determine whether the statement is *True* or *False*.

(a) All wide-sense stationary moving average processes are ergodic in the mean.

(b) All wide-sense stationary autoregressive processes are ergodic in the mean.

3.21. Let $x(n)$ be a real WSS Gaussian random process with autocovariance function $c_x(k)$. Show that $x(n)$ will be *correlation ergodic* if and only if

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} c_x^2(k) = 0$$

Hint: Use the *moment factoring theorem* for real Gaussian random variables which states that

$$E\{x_1x_2x_3x_4\} = E\{x_1x_2\}E\{x_3x_4\} + E\{x_1x_3\}E\{x_2x_4\} + E\{x_1x_4\}E\{x_2x_3\}$$

3.22. Ergodicity in the mean depends on the asymptotic behavior of the autocovariance of a process, $c_x(k)$. The asymptotic behavior of $c_x(k)$, however, is related to the behavior of the power spectrum $P_x(e^{j\omega})$ at $\omega = 0$. Show that $x(n)$ is ergodic in the mean if and only if $P_x(e^{j\omega})$ is continuous at the origin, $\omega = 0$. *Hint:* Express Eq. (3.65) as a limit, as $N \to \infty$, of the convolution of $c_x(k)$ with a pulse

$$p_N(k) = \begin{cases} 1/N & ; \quad 0 \le k < N \\ 0 & ; \quad \text{otherwise} \end{cases}$$

with the convolution being evaluated at $k = N$.

3.23. In Section 3.2.4 it was shown that for real-valued zero mean random variables the autocorrelation is bounded by one in magnitude

$$|\rho_{xy}| \leq 1$$

Show that this bound also applies when x and y are complex random variables with nonzero mean. Determine what relationship must hold between x and y if

$$|\rho_{xy}| = 1$$

3.24. Let $P_x(e^{j\omega})$ be the power spectrum of a wide-sense stationary process $x(n)$ and let λ_k be the eigenvalues of the $M \times M$ autocorrelation matrix $\mathbf{R}_x$. *Szegö's theorem* states that if $g(\cdot)$ is a continuous function then

$$\lim_{M\to\infty} \frac{g(\lambda_1) + g(\lambda_2) + \cdots + g(\lambda_M)}{M} = \frac{1}{2\pi}\int_{-\pi}^{\pi} g\left[P_x(e^{j\omega})\right]d\omega$$

Using this theorem, show that

$$\lim_{M\to\infty}\left[\det \mathbf{R}_x\right]^{1/M} = \exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left[P_x(e^{j\omega})\right]d\omega\right\}$$

3.25. In some applications, the data collection process may be flawed so that there are either missing data values or outliers that should be discarded. Suppose that we are given N samples of a WSS process $x(n)$ with one value, $x(n_0)$, missing. Let $\mathbf{x}$ be the vector containing the given sample values,

$$\mathbf{x} = \left[x(0),\ x(1), \ldots, x(n_0 - 1),\ x(n_0 + 1), \ldots, x(N)\right]^T$$

(a) Let $\mathbf{R}_x$ be the autocorrelation matrix for the vector $\mathbf{x}$,

$$\mathbf{R}_x = E\left\{\mathbf{x}\mathbf{x}^H\right\}$$

Which of the following statements are true:

1. $\mathbf{R}_x$ is Toeplitz.
2. $\mathbf{R}_x$ is Hermitian.
3. $\mathbf{R}_x$ is positive semidefinite.

(b) Given the autocorrelation matrix for $\mathbf{x}$, is it possible to find the autocorrelation matrix for the vector

$$\mathbf{x} = \left[x(0),\ x(1), \ldots, x(N)\right]^T$$

that does not have $x(n_0)$ missing? If so, how would you find it? If not, explain why not.

3.26. The power spectrum of a wide-sense stationary process $x(n)$ is

$$P_x(e^{j\omega}) = \frac{25 - 24\cos\omega}{26 - 10\cos\omega}$$

Find the whitening filter $H(z)$ that produces unit variance white noise when the input is $x(n)$.

3.27. We have seen that the autocorrelation matrix of a WSS process is positive semidefinite,

$$\mathbf{R}_x \geq 0$$

The spectral factorization theorem states that if $P_x(e^{j\omega})$ is continuous then the power spectrum may be factored as

$$P_x(e^{j\omega}) = \sigma_0^2 |Q(e^{j\omega})|^2$$

where $Q(e^{j\omega})$ corresonds to a causal and stable filter.

(a) If $\sigma_0^2 \neq 0$ and $Q(e^{j\omega})$ is nonzero for all ω, show that the autocorrelation matrix is positive definite.

(b) Give an example of a nontrivial process for which $\mathbf{R}_x$ is *not* positive definite.

Computer Exercises

C3.1. If u is a random variable with a uniform probability density function,

$$f_u(\alpha) = \begin{cases} 1 & ; \quad 0 \leq \alpha \leq 1 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

then a random variable x with a distribution function $F_x(\alpha)$ may be formed from u using the transformation [5]

$$x = F_x^{-1}(u)$$

(a) Let u be uniformly distributed between 0 and 1. How would you create an exponentially distributed random variable with a density function

$$f_x(\alpha) = \frac{1}{\mu} e^{-\alpha/\mu} \quad ; \quad \alpha \geq 0$$

(b) Find the mean and variance of the exponentially distributed random variable x given in (a).

(c) How many samples of an exponentially distributed random variable x are required in order for the sample mean

$$\hat{m}_x = \frac{1}{N} \sum_{i=1}^{N} x_i$$

to be within 1% of the true mean with a probability of 0.01, i.e.,

$$\text{Prob}\{|\hat{m}_x - m_x| > 0.01 m_x\} < 0.01$$

(d) Using the MATLAB m-file `rand`, which generates random numbers that are uniformly distributed between 0 and 1, generate a sufficient number of exponentially distributed random numbers having a density function

$$f_x(\alpha) = \frac{1}{2} e^{-\alpha/2} \quad ; \quad \alpha \geq 0$$

so that the sample mean will be within 1% of the true mean with a probability of 0.01. Find the sample mean and compare it to the true mean. Repeat this experiment for several different sets of random samples.

C3.2. In many applications it is necessary to be able to efficiently estimate the autocorrelation sequence of a random process from a finite number of samples, e.g. $x(n)$ for

$n = 0, 1, \ldots, N-1$. The autocorrelation may be estimated using the sample autocorrelation

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

Since $x(n)$ is only given for values of n within the interval $[0, N-1]$, in evaluating this sum $x(n)$ is assumed to be equal to zero for values of n that are outside the interval $[0, N-1]$. Alternatively, the sample mean may be expressed as

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=k}^{N-1} x(n)x^*(n-k) = \frac{1}{N}\sum_{n=0}^{N-1-k} x(n+k)x^*(n) \quad ; \quad 0 \le k < N$$

with $\hat{r}_x(k) = \hat{r}_x^*(-k)$ for $k < 0$, and $\hat{r}_x(k) = 0$ for $|k| \ge N$.

(a) Show that $\hat{r}_x(k)$ may be written as a convolution

$$\hat{r}_x(k) = \frac{1}{N}x(k) * x^*(-k)$$

if it is assumed that the values of $x(n)$ outside the interval $[0, N-1]$ are zero.

(b) Show that the discrete-time Fourier transform of $\hat{r}_x(k)$ is equal to the magnitude squared of the discrete-time Fourier transform of $x(n)$,

$$\sum_{k=-N+1}^{N-1} \hat{r}_x(k)e^{-jk\omega} = \frac{1}{N}|X(e^{j\omega})|^2$$

where

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x(n)e^{-jn\omega}$$

In other words, the sample autocorrelation may be computed by taking the inverse DTFT of the magnitude squared of the DTFT of $x(n)$, scaled by $1/N$.

(c) With $x(n) = 1$ for $n = 0, 1, \ldots, 7$, plot the sample autocorrelation sequence $\hat{r}_x(k)$. Using MATLAB, find the sample autocorrelation of $x(n)$ using DFT's of length 8, 16, and 32. Comment on what you observe. (Note: the sample autocorrelation may be computed using a single MATLAB command line.)

C3.3. In this exercise we will look at how accurately the sample autocorrelation is in estimating the autocorrelation of white noise.

(a) Generate 1000 samples of zero mean unit variance white Gaussian noise.

(b) Estimate the first 100 lags of the autocorrelation sequence using the sample autocorrelation

$$\hat{r}_x(k) = \frac{1}{1000}\sum_{n=0}^{999} x(n)x(n-k)$$

How close is your estimate to the *true* autocorrelation sequence $r_x(k) = \delta(k)$?

(c) Segment your white noise sequence into ten different white noise sequences each having a length of 100 samples, and estimate the autocorrelation by averaging the sample

autocorrelations of each subsequence, i.e.,

$$\hat{r}_x(k) = \frac{1}{1000}\sum_{m=0}^{9}\sum_{n=0}^{99} x(n+100m)x(n-k+100m) \quad ; \quad k = 0, 1, \ldots, 99$$

How does your estimate compare to that in part (b)? How does it compare to the true autocorrelation sequence $r_x(k)$?

(d) Generate 10,000 samples of a zero mean unit variance white Gaussian noise sequence and estimate the first 100 lags of the autocorrelation sequence as in part (b). How does your estimate compare to that in part (b)? What conclusions can you draw from these experiments?

C3.4. Consider the autoregressive random process

$$x(n) = a(1)x(n-1) + a(2)x(n-2) + b(0)v(n)$$

where $v(n)$ is unit variance white noise.

(a) With $a(1) = 0$, $a(2) = -0.81$, and $b(0) = 1$, generate 24 samples of the random process $x(n)$.

(b) Estimate the autocorrelation sequence using the sample autocorrelation and compare it to the true autocorrelation sequence.

(c) Using your estimated autocorrelation sequence, estimate the power spectrum of $x(n)$ by computing the Fourier transform of $\hat{r}_x(k)$. (Hint: this may be done easily using the results derived in Prob. C3.3.)

(d) Using the estimated autocorrelation values found in (b), use the Yule-Walker equations to estimate the values of $a(1)$, $a(2)$, and $b(0)$ and comment on the accuracy of your estimates.

(e) Estimate the power spectrum using your estimates of $a(k)$ and $b(0)$ derived in (d) as follows

$$\hat{P}_x(e^{j\omega}) = \frac{b^2(0)}{\left|1 + a(1)e^{-j\omega} + a(2)e^{-2j\omega}\right|^2}$$

(f) Compare your power spectrum estimates in parts (c) and (e) with the true power spectrum.

SIGNAL MODELING

4

4.1 INTRODUCTION

Signal modeling is an important problem that arises in a variety of applications. In data compression, for example, a waveform $x(n)$ consisting of N data values, $x(0), \ldots, x(N-1)$, is to be transmitted across a communication channel or archived on tape or disk. One method that may be used for transmission or storage is to process the signal on a point-by-point basis, i.e., transmit $x(0)$ followed by $x(1)$ and so on. However, if it is possible to accurately model the signal with a small number of parameters k, where $k \ll N$, then it would be more efficient to transmit or store these parameters instead of the signal values. The reconstruction of the signal from the model parameters would then be performed either by the communications receiver or the data retrieval system. As a more concrete example, consider the signal $x(n) = \alpha \cos(n\omega_0 + \phi)$. If this signal is to be saved on disk, then, clearly, the signal values themselves could be stored. A more efficient approach, however, would be to record the amplitude α, the frequency ω_0, and the phase ϕ. At any later time the signal $x(n)$ could be reconstructed exactly from these parameters. In a more practical setting, suppose that a signal to be stored or transmitted can be accurately modeled as a sum of L sinusoids,

$$x(n) \approx \sum_{k=1}^{L} \alpha_k \cos(n\omega_k + \phi_k) \tag{4.1}$$

Again, either the signal values $x(n)$ or the model parameters $\{\alpha_k, \omega_k, \phi_k : k = 1, \ldots, L\}$ could be recorded or transmitted. The tradeoff between the two representations is one of accuracy versus efficiency.

Another application of signal modeling is in the area of signal prediction (extrapolation) and signal interpolation. In both of these problems a signal $x(n)$ is known over a given interval of time and the goal is to determine $x(n)$ over some other interval. In the prediction problem, for example, $x(n)$ may be known for $n = 0, 1, \ldots, N-1$ and the goal is to predict the next signal value, $x(N)$. In the case of interpolation, one or more consecutive values may be missing or severely distorted over some interval, say $[N_1, N_2]$, and the problem is to use the data outside this interval to recover or estimate the values of $x(n)$ inside the

interval. In both cases, if a model can be found that provides an accurate representation for $x(n)$, then the model may be used to estimate the unknown values of $x(n)$. For example, suppose that $x(n)$ may be modeled with a recursion of the form

$$x(n) \approx \sum_{k=1}^{p} a_p(k)x(n-k) \tag{4.2}$$

Given $x(n)$ for $n \in [0, N-1]$, the next signal value, $x(N)$, could be estimated using Eq. (4.2) as follows[1]:

$$\hat{x}(N) = \sum_{k=1}^{p} a_p(k)x(N-k)$$

Signal modeling is concerned with the representation of signals in an efficient manner. In general, there are two steps in the modeling process. The first is to choose an appropriate parametric form for the model. One possibility, as described above, is to model the signal as a sum of sinusoids. Another is to use a sum of damped sinusoids

$$x(n) \approx \sum_{k=1}^{L} \alpha_k \lambda_k^n \cos(n\omega_k + \phi_k)$$

In this chapter, the model that will be used is one that represents a signal as the output of a causal linear shift-invariant filter that has a rational system function of the form

$$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k)z^{-k}}{1 + \sum_{k=1}^{p} a_p(k)z^{-k}} \tag{4.3}$$

as shown in Fig. 4.1. Thus, the *signal model* includes the filter coefficients, $a_p(k)$ and $b_q(k)$, along with a description of the input signal $v(n)$. For the most part, the signals that we will be trying to model will be deterministic. Therefore, in order to keep the model as efficient as possible, the filter input will typically either be a fixed signal, such as a unit sample $\delta(n)$, or it will be a signal that may be represented parametrically using only a few parameters, such as

$$v(n) = \sum_{k=0}^{p} \delta(n - kn_0)$$

which is used in modeling voiced speech where n_0 represents the pitch period [11] or

$$v(n) = \sum_{k=0}^{p} \alpha_k \delta(n - n_k)$$

which is used in multipulse linear predictive coding [1, 4]. In Section 4.7, where we consider models for random processes, the input to the filter will be a stochastic process, which, generally, will be taken to be white noise.

Once the form of the model has been selected, the next step is to find the model parameters that provide the *best* approximation to the given signal. There are, however, many different ways to define what is meant by "*the best*" approximation and, depending upon the definition that is used, there will be different solutions to the modeling problem along

[1] In general, a "hat" above any signal or function will be used to indicate an estimate or approximation to the given signal or function.

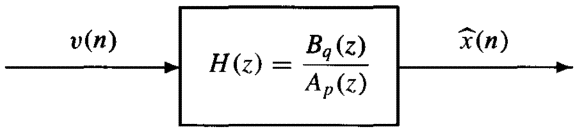


Figure 4.1 *Modeling a signal $x(n)$ as the response of a linear shift-invariant filter to an input $v(n)$. The goal is to find the filter $H(z)$ and the input $v(n)$ that make $\hat{x}(n)$ "as close as possible" to $x(n)$.*

with different techniques for finding the model parameters. Therefore, in developing an approach to signal modeling, it will be important not only to find a model that is useful, i.e., works well, but also one that has a computationally efficient procedure for deriving the model parameters from the given data.

In this chapter, a number of different approaches to signal modeling will be developed. The first, presented in Section 4.2, is a *least squares method* that minimizes the sum of the squares of the error

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

which is the difference between the signal $x(n)$ and the output of the filter. As we will see, this approach requires finding the solution to a set of nonlinear equations. Therefore, it is rarely used in practice. In Section 4.3, a more practical solution is developed that forces the model to be exact, $e'(n) = 0$, over a fixed finite interval such as $[0, N-1]$. This approach, known as the Padé approximation method, has the advantage of only requiring the solution to a set of linear equations. However, this method is limited in that there is no guarantee on how accurate the model will be for values of n outside the interval $[0, N-1]$. Therefore, in Section 4.4 we develop Prony's method which is a blend between the least squares approach and the Padé approximation method and has the advantage of only requiring the solution to a set of linear equations, while, at the same time, producing a model that is more accurate, on the average, than the Padé method. We then derive Shanks' method, which modifies the Prony approach for finding the zeros of the system function. The special case of all-pole modeling using Prony's method is then considered, and the relationship between all-pole modeling and linear prediction is discussed. In Section 4.5, we return to the least squares method of Section 4.2 and develop an iterative algorithm for solving the nonlinear modeling equations. In Section 4.6, we develop the autocorrelation and covariance methods of all-pole modeling, which are modifications to the all-pole Prony method for the case in which only a finite length data record is available. Then, in Section 4.7, we consider the problem of modeling random processes. Our discussion of signal modeling will not end, however, with this chapter. In Chapter 6, for example, models that are based on the lattice filter will be considered; in Chapter 9, we will present some methods that may be used to model nonstationary signals.

4.2 THE LEAST SQUARES (DIRECT) METHOD

In this section, we consider the problem of modeling a deterministic signal, $x(n)$, as the unit sample response of a linear shift-invariant filter, $h(n)$, having a rational system function of the form given in Eq. (4.3). Thus, $v(n)$ in Fig. 4.1 is taken to be the unit sample, $\delta(n)$. We will assume, without any loss in generality, that $x(n) = 0$ for $n < 0$ and that the filter $h(n)$

is causal. Denoting the modeling error by $e'(n)$,

$$e'(n) = x(n) - h(n)$$

the problem is to find the filter coefficients, $a_p(k)$ and $b_q(k)$, that make $e'(n)$ *as small as possible*. Before $e'(n)$ may be minimized, however, it is necessary to define what is meant by *small*. For example, does *small* mean that the maximum value of $|e'(n)|$ is to be made as small as possible? Does it mean that we should minimize the mean square value? Or does it mean something else? In the *least squares* or *direct method* of signal modeling, the error measure that is to be minimized is the squared error [6]

$$\mathcal{E}_{LS} = \sum_{n=0}^{\infty} |e'(n)|^2$$

(Note that since $h(n)$ and $x(n)$ are both assumed to be zero for $n < 0$, then $e'(n) = 0$ for $n < 0$ and the summation begins at $n = 0$.) A block diagram of the direct method of signal modeling is shown in Fig. 4.2. A necessary condition for the filter coefficients $a_p(k)$ and $b_q(k)$ to minimize the squared error is that the partial derivative of $\mathcal{E}_{LS}$ with respect to each of the coefficients vanish, i.e.,

$$\frac{\partial \mathcal{E}_{LS}}{\partial a_p^*(k)} = 0 \quad ; \quad k = 1, 2, \ldots, p$$

and

$$\frac{\partial \mathcal{E}_{LS}}{\partial b_q^*(k)} = 0 \quad ; \quad k = 0, 1, \ldots, q$$

Using Parseval's theorem, the least squares error may be written in the frequency domain in terms of $E'(e^{j\omega})$, the Fourier transform of $e'(n)$, as follows:

$$\mathcal{E}_{LS} = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E'(e^{j\omega})|^2 d\omega \tag{4.4}$$

This representation allows us to express $\mathcal{E}_{LS}$ explicitly in terms of the model parameters $a_p(k)$ and $b_q(k)$. Thus, setting the partial derivatives with respect to $a_p^*(k)$ equal to zero we have

$$\frac{\partial \mathcal{E}_{LS}}{\partial a_p^*(k)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\partial}{\partial a_p^*(k)} [E'(e^{j\omega}) E'^*(e^{j\omega})] d\omega = 0 \tag{4.5}$$

Since

$$E'(e^{j\omega}) = X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}$$

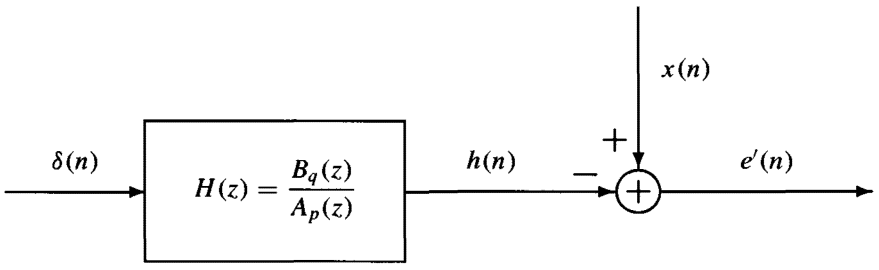


Figure 4.2 *The direct method of signal modeling for approximating a signal $x(n)$ as the unit sample response of a linear shift-invariant system having p poles and q zeros.*

treating $a_p(k)$ and $a_p^*(k)$ as independent variables, as discussed in Section 2.3.10, Eq. (4.5) becomes

$$\boxed{\frac{\partial \mathcal{E}_{LS}}{\partial a_p^*(k)} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}\right]\frac{B_q^*(e^{j\omega})}{[A_p^*(e^{j\omega})]^2}e^{jk\omega}d\omega = 0} \tag{4.6}$$

for $k = 1, 2, \ldots, p$. Similarly, differentiating Eq. (4.4) with respect to $b_q(k)$ we have

$$\boxed{\frac{\partial \mathcal{E}_{LS}}{\partial b_q^*(k)} = -\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}\right]\frac{e^{jk\omega}}{A_p^*(e^{j\omega})}d\omega = 0} \tag{4.7}$$

for $k = 0, 1, \ldots, q$. What is clear from Eqs. (4.6) and (4.7) is that the optimum set of model parameters are defined implicitly in terms of a set of $p+q+1$ nonlinear equations. Although iterative techniques such as the method of steepest descent, Newton's method, or iterative prefiltering (see Section 4.5) could be used to solve these equations, the least squares approach is not mathematically tractable and not amenable to real-time signal processing applications. It is for this reason that we now turn our attention to some indirect methods of signal modeling. In these methods, the modeling problem is changed slightly so that the model parameters may be found more easily. In the following sections it is important to keep in mind, however, that the problems that are being solved are different from the true least squares solution presented above.

4.3 THE PADÉ APPROXIMATION

Since a signal model that is based on a least squares error criterion leads to a mathematically intractable solution, in this section we will reformulate the signal modeling problem and derive what has become known as the Padé approximation method [6, 9]. Unlike the least squares solution, the Padé approximation only requires solving a set of linear equations. In order to motivate the Padé approximation, let $x(n)$ be a signal that is to be modeled as the unit sample response of a causal linear shift-invariant filter. With a system function having p poles and q zeros as in Eq. (4.3), note that there are $p+q+1$ *degrees of freedom* in the model, i.e., the p denominator coefficients and the $q+1$ numerator coefficients. Therefore, with the appropriate choice for the coefficients $a_p(k)$ and $b_q(k)$ it would seem plausible that we could force the filter output, $h(n)$, to be equal to the given signal $x(n)$ for $p+q+1$ values of n. In order to explore this idea a little further, let us consider a simple example.

Example 4.3.1 *Exact Matching of a Signal*

Let $x(n)$ be a signal that is to be modeled using a causal first-order all-pole filter of the form

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1}}$$

Since the unit sample response of this filter is

$$h(n) = b(0)[-a(1)]^n u(n)$$

the constraint that $h(n) = x(n)$ for $n = 0, 1$ requires that values for $a(1)$ and $b(0)$ be found such that

$$\begin{aligned} x(0) &= b(0) \\ x(1) &= -b(0)a(1) \end{aligned}$$

Assuming that $x(0) \neq 0$, the solution to these two equations is $b(0) = x(0)$ and $a(1) = -x(1)/x(0)$ and the model is[2]

$$H(z) = \frac{x^2(0)}{x(0) - x(1)z^{-1}}$$

Therefore, with a first-order model it is always possible to match the first two signal values exactly if $x(0) \neq 0$.

If we increase the order of the model to include one pole and one zero,

$$H(z) = \frac{b(0) + b(1)z^{-1}}{1 + a(1)z^{-1}}$$

the problem becomes a bit more complicated. Since the unit sample response in this case is

$$h(n) = b(0)\left[-a(1)\right]^n u(n) + b(1)\left[-a(1)\right]^{n-1} u(n-1)$$

the equations that must be solved so that $h(n) = x(n)$ for $n = 0, 1, 2$ are

$$\begin{aligned} x(0) &= b(0) \\ x(1) &= -b(0)a(1) + b(1) \\ x(2) &= b(0)a^2(1) - b(1)a(1) = a(1)\left[b(0)a(1) - b(1)\right] \end{aligned}$$

Although these equations are nonlinear, if we substitute the second equation into the third, then we have

$$x(2) = -a(1)x(1)$$

Therefore,

$$a(1) = -\frac{x(2)}{x(1)}$$

and it follows from the first two equations that

$$\begin{aligned} b(0) &= x(0) \\ b(1) &= x(1) - x(0)\frac{x(2)}{x(1)} \end{aligned}$$

Thus, if $x(1) \neq 0$, then we may match the first three values of $x(n)$ with a model consisting of one pole and one zero.

In the previous example, it is important to note that, in spite of the fact that we were able to find the filter coefficients that make $h(n) = x(n)$ for $n = 0, 1$ in the first case and $n = 0, 1, 2$ in the second, the equations that we needed to solve were *nonlinear*. Although this will be the case in general, we will see in the following discussion that it is possible to

[2]The form of the model presupposes that $h(0) \neq 0$ and, therefore, that $x(0) \neq 0$. For those cases in which $x(0) = 0$, the signal may simply be shifted until $x(0) \neq 0$.

reformulate the problem in such a way that the filter coefficients may be found by solving a set of *linear* equations. We will also establish that, for a model containing p poles and q zeros, it is almost always possible to find a set of filter coefficients that force $h(n) = x(n)$ for $p + q + 1$ values of n. In the discussions that follow, $x(n)$ may be either a real- or complex-valued sequence.

The development of the Padé method begins by expressing the system function,

$$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_q(k)z^{-k}}{1 + \sum_{k=1}^{p} a_p(k)z^{-k}}$$

as follows:

$$H(z)A_p(z) = B_q(z) \tag{4.8}$$

In the time domain, Eq. (4.8) becomes a convolution

$$h(n) + \sum_{k=1}^{p} a_p(k)h(n-k) = b_q(n) \tag{4.9}$$

where $h(n) = 0$ for $n < 0$ and $b_q(n) = 0$ for $n < 0$ and $n > q$. To find the coefficients $a_p(k)$ and $b_q(k)$ that give an exact fit of the data to the model over the interval $[0, p + q]$ we set $h(n) = x(n)$ for $n = 0, 1, \ldots, p + q$ in Eq. (4.9). This leads to the following set of $p + q + 1$ linear equations in $p + q + 1$ unknowns,

$$x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) = \begin{cases} b_q(n) & ; \quad n = 0, 1, \ldots, q \\ 0 & ; \quad n = q+1, \ldots, q+p \end{cases} \tag{4.10}$$

In matrix form these equations may be written as follows:

$$\left[\begin{array}{cccc} x(0) & 0 & \cdots & 0 \\ x(1) & x(0) & \cdots & 0 \\ x(2) & x(1) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ x(q) & x(q-1) & \cdots & x(q-p) \\ \hline x(q+1) & x(q) & \cdots & x(q-p+1) \\ \vdots & \vdots & & \vdots \\ x(q+p) & x(q+p-1) & \cdots & x(q) \end{array}\right] \left[\begin{array}{c} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.11}$$

To solve Eq. (4.11) for $a_p(k)$ and $b_q(k)$ we use a two-step approach, first solving for the denominator coefficients $a_p(k)$ and then solving for the numerator coefficients $b_q(k)$.

In the first step of the Padé approximation method, solving for the coefficients $a_p(k)$, we use the last p equations of Eq. (4.11) as indicated by the partitioning, i.e.,

$$\left[\begin{array}{cccc} x(q+1) & x(q) & \cdots & x(q-p+1) \\ x(q+2) & x(q+1) & \cdots & x(q-p+2) \\ \vdots & \vdots & & \vdots \\ x(q+p) & x(q+p-1) & \cdots & x(q) \end{array}\right] \left[\begin{array}{c} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.12}$$

Expanding Eq. (4.12) as follows

$$\begin{bmatrix} x(q+1) \\ x(q+2) \\ \vdots \\ x(q+p) \end{bmatrix} + \begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ \vdots & \vdots & & \vdots \\ x(q+p-1) & x(q+p-2) & \cdots & x(q) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and then bringing the vector on the left to the right-hand side leads to the following set of p equations in the p unknowns $a_p(1), a_p(2), \ldots, a_p(p)$,

$$\begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ \vdots & \vdots & & \vdots \\ x(q+p-1) & x(q+p-2) & \cdots & x(q) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} x(q+1) \\ x(q+2) \\ \vdots \\ x(q+p) \end{bmatrix} \tag{4.13}$$

Using matrix notation Eq. (4.13) may be expressed concisely as

$$\boxed{\mathbf{X}_q \bar{\mathbf{a}}_p = -\mathbf{x}_{q+1}} \tag{4.14}$$

where

$$\bar{\mathbf{a}}_p = \left[a_p(1), a_p(2), \ldots, a_p(p)\right]^T$$

is the vector of denominator coefficients,

$$\mathbf{x}_{q+1} = \left[x(q+1),\ x(q+2), \ldots, x(q+p)\right]^T$$

is a column vector of length p, and

$$\mathbf{X}_q = \begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ \vdots & \vdots & & \vdots \\ x(q+p-1) & x(q+p-2) & \cdots & x(q) \end{bmatrix}$$

is a $p \times p$ nonsymmetric Toeplitz matrix.[3]

With respect to solving Eq. (4.13), there are three cases of interest, depending upon whether or not the matrix $\mathbf{X}_q$ is invertible. Each case is considered separately below.

- **Case I. $\mathbf{X}_q$ is nonsingular.** If $\mathbf{X}_q$ is nonsingular then $\mathbf{X}_q^{-1}$ exists and the coefficients of $A_p(z)$ are unique and are given by

$$\bar{\mathbf{a}}_p = -\mathbf{X}_q^{-1}\mathbf{x}_{q+1}$$

  Since $\mathbf{X}_q$ is a nonsymmetric Toeplitz matrix, these equations may be solved efficiently using a recursion known as the Trench algorithm [18].

- **Case II. $\mathbf{X}_q$ is singular and a solution to Eq. (4.14) exists.** If there is a vector $\bar{\mathbf{a}}_p$ that solves Eq. (4.14) and if $\mathbf{X}_q$ is singular, then the solution is not unique. Specifically, when $\mathbf{X}_q$ is singular there are nonzero solutions to the homogeneous equation

$$\mathbf{X}_q \mathbf{z} = 0 \tag{4.15}$$

[3]The subscript q on the data matrix $\mathbf{X}_q$ indicates the index of $x(n)$ in the element in the upper-left corner. Similarly, the subscript $q+1$ on the data vector $\mathbf{x}_{q+1}$ indicates the index of $x(n)$ in the first element.

Therefore, for any solution $\bar{\mathbf{a}}_p$ to Eq. (4.14) and for any $\mathbf{z}$ that satisfies Eq. (4.15),

$$\tilde{\mathbf{a}}_p = \bar{\mathbf{a}}_p + \mathbf{z}$$

is also a solution. In this case, the best choice for $\bar{\mathbf{a}}_p$ may be the one that has the fewest number of nonzero coefficients since this solution would provide the most efficient parametric representation for the signal.

- **Case III.** $\mathbf{X}_q$ is singular and no solution exists. In the formulation of Padé's method, it was assumed that the denominator polynomial in $H(z)$ could be written in the form

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k) z^{-k}$$

In other words, it was assumed that the leading coefficient, $a_p(0)$, was nonzero and that $A_p(z)$ could be normalized so that $a_p(0) = 1$. However, if $\mathbf{X}_q$ is singular and no vector can be found that satisfies Eq. (4.14), then this nonzero assumption on $a_p(0)$ is incorrect. Therefore, if we set $a_p(0) = 0$, then the Padé equations that are derived from Eq. (4.12) become

$$\mathbf{X}_q \bar{\mathbf{a}}_p = 0 \tag{4.16}$$

Since $\mathbf{X}_q$ is singular, there is a nonzero solution to these homogeneous equations. Thus, as we will see in Example 4.3.3, the denominator coefficients may be found by solving Eq. (4.16).

Having solved Eq. (4.14) for $\bar{\mathbf{a}}_p$, the second step in the Padé approximation is to find the numerator coefficients $b_q(k)$. These coefficients are found from the first $(q+1)$ equations in Eq. (4.11),

$$\begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ x(q) & x(q-1) & x(q-2) & \cdots & x(q-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix} \tag{4.17}$$

With

$$\mathbf{a}_p = \left[1,\ a_p(1),\ a_p(2),\ \cdots,\ a_p(p)\right]^T$$

and

$$\mathbf{b}_q = \left[b_q(0),\ b_q(1),\ b_q(2),\ \cdots,\ b_q(p)\right]^T$$

these equations may be written in matrix form as follows

$$\boxed{\mathbf{X}_0 \mathbf{a}_p = \mathbf{b}_q} \tag{4.18}$$

Therefore, $\mathbf{b}_q$ may be found by simply multiplying $\mathbf{a}_p$ by the matrix $\mathbf{X}_0$. Equivalently, the coefficients $b_q(k)$ may be evaluated using Eq. (4.10) as follows

$$b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k) x(n-k) \quad ; \quad n = 0, 1, \ldots, q \tag{4.19}$$

Table 4.1 ***The Padé Approximation for Modeling a Signal as the Unit Sample Response of Linear Shift-Invariant System Having p Poles and q Zeros***

Denominator coefficients

$$\begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ \vdots & \vdots & & \vdots \\ x(q+p-1) & x(q+p-2) & \cdots & x(q) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} x(q+1) \\ x(q+2) \\ \vdots \\ x(q+p) \end{bmatrix}$$

Numerator coefficients

$$\begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ x(q) & x(q-1) & x(q-2) & \cdots & x(q-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix}$$

Thus, as summarized in Table 4.1, the Padé approximation method is a two-step approach for finding the model parameters. First, Eq. (4.14) is solved for the denominator coefficients $a_p(k)$. Then either Eq. (4.17) or Eq. (4.19) is used to find the numerator coefficients $b_q(k)$. A MATLAB program for the Padé approximation is given in Fig. 4.3.

Before looking at some examples, we consider the special case of an *all pole-model*,

$$H(z) = \frac{b(0)}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} \tag{4.20}$$

With $q = 0$, the Padé equations for the coefficients $a_p(k)$ become

$$\begin{bmatrix} x(0) & 0 & \cdots & 0 \\ x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ x(p-1) & x(p-2) & \cdots & x(0) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(p) \end{bmatrix} \tag{4.21}$$

The Padé Approximation

```
function [a,b] = pade(x,p,q)
%
   x   = x(:);
   if p+q>=length(x), error('Model order too large'), end
      X   = convm(x,p+1);
      Xq  = X(q+2:q+p+1,2:p+1);
      a   = [1;-Xq\X(q+2:q+p+1,1)];
      b   = X(1:q+1,1:p+1)*a;
      end;
```

Figure 4.3 *A* MATLAB *program for signal modeling using the Padé approximation. Note that this program calls* `convm.m` (*see Appendix*).

or,

$$\mathbf{X}_0\bar{\mathbf{a}}_p = -\mathbf{x}_1$$

Since $\mathbf{X}_0$ is a lower triangular Toeplitz matrix the denominator coefficients may be found easily by back substitution using the recursion

$$a_p(k) = -\frac{1}{x(0)}\left[x(k) + \sum_{l=1}^{k-1} a_p(l)x(k-l)\right]$$

In order to gain some familiarity with the Padé approximation method and its properties, we now look at a few examples.

Example 4.3.2 *Padé Approximation*

Given a signal whose first six values are

$$\mathbf{x} = [1,\ 1.500,\ 0.750,\ 0.375,\ 0.1875,\ 0.0938]^T$$

let us use the Padé approximation to find a second-order all-pole model ($p = 2$ and $q = 0$), a second-order moving-average model ($p = 0$ and $q = 2$), and a model containing one pole and one zero ($p = q = 1$).

1. To find a second-order all-pole model, $p = 2$ and $q = 0$, the equations that must be solved are

$$\begin{bmatrix} x(0) & 0 & 0 \\ x(1) & x(0) & 0 \\ x(2) & x(1) & x(0) \end{bmatrix}\begin{bmatrix} 1 \\ a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} b(0) \\ 0 \\ 0 \end{bmatrix} \tag{4.22}$$

From the last two equations we have

$$\begin{bmatrix} x(0) & 0 \\ x(1) & x(0) \end{bmatrix}\begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = -\begin{bmatrix} x(1) \\ x(2) \end{bmatrix} \tag{4.23}$$

Substituting the given values of $x(n)$ into Eq. (4.23)

$$\begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix}\begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = -\begin{bmatrix} 1.50 \\ 0.75 \end{bmatrix}$$

and solving for $a(1)$ and $a(2)$ gives

$$a(1) = -1.50 \quad \text{and} \quad a(2) = 1.50$$

Finally, using the first equation in Eq. (4.22) it follows that $b(0) = x(0) = 1$. Therefore, the model for $x(n)$ is

$$H(z) = \frac{1}{1 - 1.50z^{-1} + 1.50z^{-2}}$$

Note that this model is unstable, having a pair of complex poles that lie outside the unit circle. Evaluating the unit sample response we see that the model produces the following approximation for $x(n)$,

$$\hat{\mathbf{x}} = [1,\ 1.500,\ 0.750,\ -1.125,\ -2.8125,\ -2.5312]$$

Although $\hat{x}(n) = x(n)$ for $n = 0, 1, 2$, as it should, the model does not produce an accurate representation of $x(n)$ for $n = 3, 4, 5$.

2. For a second-order all-zero model, $p = 0$ and $q = 2$, the denominator polynomial is a constant, $A(z) = 1$, and the equations that need to be solved are trivial,

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ b(2) \end{bmatrix}$$

Therefore, the model is simply

$$H(z) = 1 + 1.50z^{-1} + 0.75z^{-2}$$

and $\hat{x}(n) = 0$ for $n > p + q = 2$.

3. Finally, let us consider a model having one pole and one zero, $p = q = 1$. In this case the model is of the form

$$H(z) = \frac{b(0) + b(1)z^{-1}}{1 + a(1)z^{-1}}$$

and the Padé equations are

$$\begin{bmatrix} x(0) & 0 \\ x(1) & x(0) \\ x(2) & x(1) \end{bmatrix} \begin{bmatrix} 1 \\ a(1) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ 0 \end{bmatrix} \tag{4.24}$$

The denominator coefficient $a(1)$ may be found from the last equation in Eq. (4.24),

$$x(2) + a(1)x(1) = 0$$

which gives

$$a(1) = -\frac{x(2)}{x(1)} = -0.5$$

Given $a(1)$, the coefficients $b(0)$ and $b(1)$ may be found from the first two equations in Eq. (4.24) as follows

$$\begin{bmatrix} b(0) \\ b(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Therefore, the model is

$$H(z) = \frac{1 + z^{-1}}{1 - 0.5z^{-1}}$$

Since the unit sample response is

$$h(n) = (0.5)^n u(n) + (0.5)^{n-1} u(n-1) = \delta(n) + 1.5\,(0.5)^{n-1} u(n-1)$$

we see that this model matches all of the signal values from $n = 0$ to $n = 5$. This will not be true in general, however, unless the signal that is being modeled is the unit sample response of a linear shift-invariant filter and a sufficient number of poles and zeros are used in the Padé model.

If $\mathbf{X}_q$ is nonsingular, as in the previous example, then there are no difficulties in solving the Padé equations. The following example, however, illustrates what must be done when $\mathbf{X}_q$ is singular.

Example 4.3.3 *Singular Padé Approximation*

Let $x(n)$ be a signal that is to be approximated using a second-order model with $p = q = 2$, where the first five values of $x(n)$ are

$$\mathbf{x} = [1, 4, 2, 1, 3]^T$$

The Padé equations are

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 4 & 1 \\ 1 & 2 & 4 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ 0 \\ 0 \end{bmatrix} \tag{4.25}$$

From the last two equations in Eq. (4.25) we have

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = -\begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

which is a singular set of equations. In addition, since there are no values for $a(1)$ and $a(2)$ that satisfy these equations, it follows that the assumption that $a(0) = 1$ is incorrect. Therefore, setting $a(0) = 0$ gives

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which has the nontrivial solution

$$a(1) = 2 \;\; ; \;\; a(2) = -1$$

Using the first three equations in Eq. (4.25) to determine the coefficients $b(k)$, we have

$$\begin{bmatrix} b(0) \\ b(1) \\ b(2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 7 \end{bmatrix}$$

and the model becomes

$$H(z) = \frac{2z^{-1} + 7z^{-2}}{2z^{-1} - z^{-2}} = \frac{2 + 7z^{-1}}{2 - z^{-1}}$$

Note that a lower order model then originally proposed has been found, i.e., $p = 1$ and $q = 1$ instead of $p = q = 2$. Computing the inverse z-transform of $H(z)$ we find

$$h(n) = (\tfrac{1}{2})^n u(n) + \tfrac{7}{2}(\tfrac{1}{2})^{n-1} u(n-1) = \delta(n) + 8(\tfrac{1}{2})^n u(n-1)$$

Therefore, the first five values of $\hat{x}(n) = h(n)$ are

$$\hat{\mathbf{x}} = [1, \; 4, \; 2, \; 1, \; 0.5]^T$$

and the model produces a signal that only matches the data for $n = 0, 1, 2, 3$, i.e., $\hat{x}(4) \neq x(4)$. Thus, if $\mathbf{X}_q$ is singular and no solution to the Padé equations exists, then it is not always possible to match all of the signal values for $n = 0, 1, \ldots p + q$.

In the next example, we consider the application of the Padé approximation method to the design of digital filters [2]. The approach that is to be used is as follows. Given the

unit sample response of an ideal digital filter, $i(n)$, we will find the system function of a realizable linear shift-invariant filter, $H(z) = B_q(z)/A_p(z)$, that has a unit sample response $h(n)$ that is equal to $i(n)$ for $n = 0, 1, \ldots, (p + q)$.[4] The hope is that if $h(n)$ "looks like" $i(n)$, then $H(e^{j\omega})$ will "look like" $I(e^{j\omega})$.

Example 4.3.4 *Filter Design Using the Padé Approximation*

Suppose that we would like to design a linear phase lowpass filter having a cutoff frequency of $\pi/2$. The frequency response of the ideal lowpass filter is

$$I(e^{j\omega}) = \begin{cases} e^{-jn_d\omega} & ; \quad |\omega| < \pi/2 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

where n_d is the filter delay, which, for the moment, is left unspecified, and the unit sample response is

$$i(n) = \frac{\sin\left[(n - n_d)\pi/2\right]}{(n - n_d)\pi}$$

We will consider two designs, one with $p = 0$ and $q = 10$ and the other with $p = q = 5$. In both cases, $p + q + 1 = 11$ values of $i(n)$ may be matched exactly. Since $i(n)$ has its largest value at $n = n_d$ and decays as $1/n$ away from $n = n_d$, we will let $n_d = 5$ so that the model forces $h(n) = i(n)$ over the interval for which the energy in $i(n)$ is maximum. With $n_d = 5$, the first eleven values of $i(n)$ are

$$\mathbf{i} = [0.0637,\ 0,\ -0.1061,\ 0,\ 0.3183,\ 0.5,\ 0.3183,\ 0,\ -0.1061,\ 0,\ 0.0637]^T$$

With $p = 0$ and $q = 10$, the Padé approximation is

$$h(n) = \begin{cases} i(n) & ; \quad 0 \le n \le 10 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

In other words, the filter is designed by multiplying the ideal unit sample response by a rectangular window. This is equivalent to the well-known and commonly used filter design technique known as windowing [8]. The frequency response of this filter is shown in Fig. 4.4 (solid line).

With $p = q = 5$, the coefficients $a(k)$ are found by solving Eq. (4.13). The solution is

$$\mathbf{a} = [1.0,\ -2.5256,\ 3.6744,\ -3.4853,\ 2.1307,\ -0.7034]^T$$

Next, the coefficients $b(k)$ are found from Eq. (4.17). The result is

$$\mathbf{b} = [0.0637,\ -0.1608,\ 0.1280,\ 0.0461,\ 0.0638,\ 0.0211]^T$$

The frequency response of this filter is shown in Fig. 4.4 (dotted line), and the error, $e'(n) = i(n) - h(n)$, is shown in Fig. 4.5. Note that this filter does not compare favorably with the filter designed by windowing. Also note that the extrapolation performed by the Padé approximation for $n > 11$ is not very accurate. It is this large difference between $i(n)$ and $h(n)$ that accounts for the poor frequency response characteristics.

[4]Note that in this application, $i(n)$ is used instead of $x(n)$ to represent the given data and $h(n)$ is used to denote the approximation to $i(n)$.

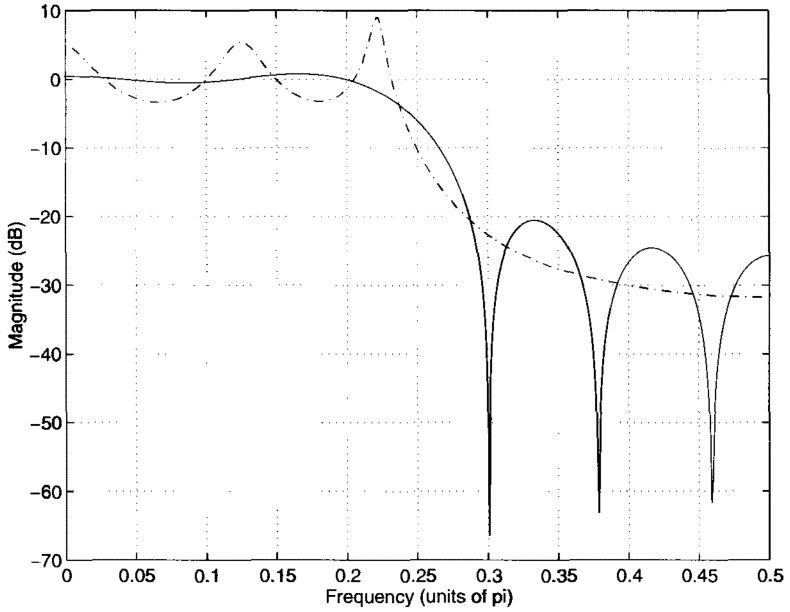


Figure 4.4 *Lowpass filter design using the Padé approximation. The frequency response of the filter having* $p = 0$ *poles and* $q = 10$ *zeros (solid line) and the frequency response of the filter having* $p = 5$ *poles and* $q = 5$ *zeros (dotted line).*

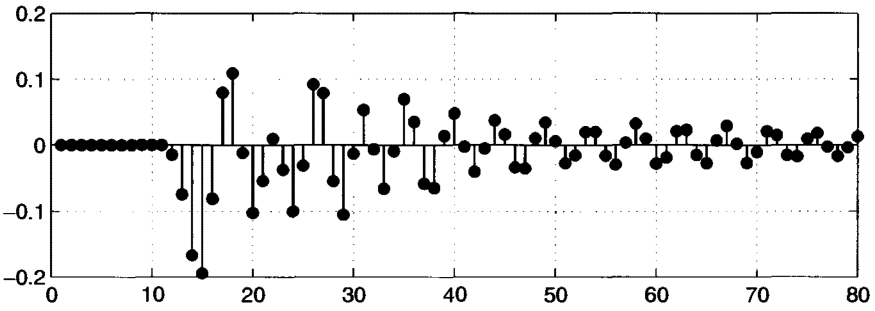


Figure 4.5 *The difference,* $e'(n)$*, between the unit sample response of the ideal lowpass filter,* $i(n)$*, and the unit sample response of the filter designed using the Padé approximation with* $p = q = 5$.

In conclusion, we may summarize the properties of the Padé approximation as follows. First, the model that is formed from the Padé approximation will always produce an exact fit to the data over the interval $[0, p + q]$, provided that $\mathbf{X}_q$ is nonsingular. However, since the data outside the interval $[0, p + q]$ is never considered in the modeling process, there is no guarantee on how accurate the model will be for $n > p + q$. In some cases, the match may be very good, while in others it may be poor. Second, for those cases in which $x(n)$ has a rational z-transform, the Padé approximation will give the correct model parameters, provided the model order is chosen to be large enough. Finally, since the Padé approximation forces the model to match the signal only over a limited range of values, the model that

is generated is not guaranteed to be stable. This may be a serious problem in applications that rely on the model to synthesize or extrapolate the signal. For example, consider a telecommunications system in which a signal is segmented into subsequences of length N and the model parameters of each subsequence are transmitted across the channel, instead of the signal values themselves. The receiver then resynthesizes the signal by piecing together the unit sample responses of each of the transmitted filters. Although such a system is capable of realizing a significant reduction in the data transmission rate if $p + q \ll N$, if the model is unstable, then there is a risk of unacceptably large errors.

4.4 PRONY'S METHOD

The limitation with the Padé approximation is that it only uses values of the signal $x(n)$ over the interval $[0, p + q]$ to determine the model parameters and, over this interval, it models the signal without error. Since all of the degrees of freedom are used to force the error to zero for $n = 0, 1, \ldots, p+q$, there is no guarantee on how well the model will approximate the signal for $n > p + q$. An alternative is to relax the requirement that the model be exact over the interval $[0, p + q]$ in order to produce a better approximation to the signal for all n. In this section, we develop such an approach known as Prony's method. In Section 4.4.1 we consider the general problem of pole-zero modeling using Prony's method. As with the Padé approximation, Prony's method only requires solving a set of linear equations. In Section 4.4.2, we then introduce a modification to Prony's method known as Shanks' method, which differs from Prony's method in the approach that is used to find the zeros of the system function. Prony's method for the special case of all-pole signal modeling is then considered in Section 4.4.3. As we will see, all-pole models have a computational advantage over the more general pole-zero models. In Section 4.4.4 we look at the relationship between all-pole modeling and linear prediction and, finally, in Section 4.4.5 we look at the application of the all-pole Prony method to the design of FIR least squares inverse filters.

4.4.1 Pole-Zero Modeling

Suppose that a signal $x(n)$ is to be modeled as the unit sample response of a linear shift-invariant filter having a rational system function $H(z)$ with p poles and q zeros, i.e., $H(z) = B_q(z)/A_p(z)$, as in Eq. (4.3). For complete generality, let us assume that $x(n)$ is complex and allow the coefficients of the filter to be complex (the analysis in the case of real signals will be the same except that the complex conjugates will disappear). Thus, assuming that $x(n) = 0$ for $n < 0$ and that $x(n)$ is known for all values of $n \geq 0$, we would like to find the filter coefficients $a_p(k)$ and $b_q(k)$ that make $h(n)$, the unit sample response of the filter, as close as possible to $x(n)$ in the sense of minimizing the error

$$e'(n) = x(n) - h(n)$$

As we saw in Section 4.2, a least squares minimization of $e'(n)$ or, equivalently, a least squares minimization of

$$E'(z) = X(z) - \frac{B_q(z)}{A_p(z)} \tag{4.26}$$

results in a set of nonlinear equations for the filter coefficients. However, if we borrow the approach used in the Padé approximation and multiply both sides of Eq. (4.26) by $A_p(z)$,

then we have a new error,

$$E(z) = A_p(z)E'(z) = A_p(z)X(z) - B_q(z)$$

that is *linear* in the filter coefficients. As illustrated in Fig. 4.6, in the time domain this error is the difference between $b_q(n)$ and the filtered signal $\hat{b}_q(n) = a_p(n) * x(n)$,

$$e(n) = a_p(n) * x(n) - b_q(n) = \hat{b}_q(n) - b_q(n) \tag{4.27}$$

Since $b_q(n) = 0$ for $n > q$, we may write this error explicitly for each value of n as follows

$$e(n) = \begin{cases} x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) - b_q(n) & ; \quad n = 0, 1, \ldots, q \\ x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) & ; \quad n > q \end{cases} \tag{4.28}$$

(recall that $A_p(z)$ is normalized so that $a_p(0) = 1$). Instead of setting $e(n) = 0$ for $n = 0, 1, \ldots, p + q$ as in the Padé approximation, Prony's method begins by finding the coefficients $a_p(k)$ that minimize the squared error[5]

$$\mathcal{E}_{p,q} = \sum_{n=q+1}^{\infty} |e(n)|^2 = \sum_{n=q+1}^{\infty} \left| x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) \right|^2 \tag{4.29}$$

Note that $\mathcal{E}_{p,q}$ depends only on the coefficients, $a_p(k)$, and not on $b_q(k)$. Therefore, the coefficients that minimize this squared error may be found by setting the partial derivatives of $\mathcal{E}_{p,q}$ with respect to $a_p^*(k)$ equal to zero, as follows[6]

$$\frac{\partial \mathcal{E}_{p,q}}{\partial a_p^*(k)} = \sum_{n=q+1}^{\infty} \frac{\partial [e(n)e^*(n)]}{\partial a_p^*(k)} = \sum_{n=q+1}^{\infty} e(n) \frac{\partial e^*(n)}{\partial a_p^*(k)} = 0 \quad ; \quad k = 1, 2, \ldots, p \tag{4.30}$$

Since the partial derivative of $e^*(n)$ with respect to $a_p^*(k)$ is $x^*(n-k)$, Eq. (4.30) becomes

$$\boxed{\sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0 \quad ; \quad k = 1, 2, \ldots, p} \tag{4.31}$$

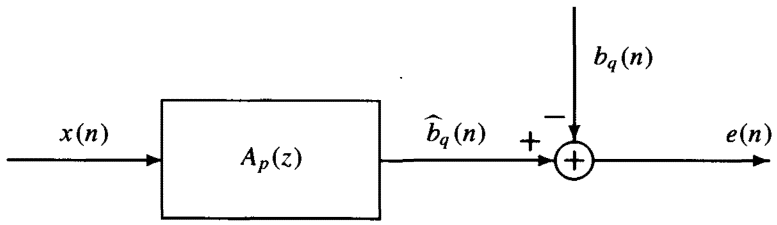


Figure 4.6 *System interpretation of Prony's method for signal modeling.*

[5]The subscripts on $\mathcal{E}_{p,q}$ are used to denote the order of the model, i.e., the number of poles and zeros. To be consistent with this notation, we should probably also denote the error sequence by $e_{p,q}(n)$. However, since this would significantly complicate the notation and add little in terms of clarity, we will continue writing the error simply as $e(n)$. In later chapters, however, most notably Chapters 5 and 6, we will use the notation $e_p(n)$ when discussing all-pole models ($q = 0$).

[6]See Section 2.3.10 for a discussion of how to find the minimum of a real-valued function of a complex vector and its conjugate.

which is known as the *orthogonality principle*.[7] Substituting Eq. (4.28) for $e(n)$ into Eq. (4.31) yields

$$\sum_{n=q+1}^{\infty} \left\{ x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) \right\} x^*(n-k) = 0$$

or equivalently,

$$\sum_{l=1}^{p} a_p(l) \left[\sum_{n=q+1}^{\infty} x(n-l)x^*(n-k) \right] = - \sum_{n=q+1}^{\infty} x(n)x^*(n-k) \tag{4.32}$$

In order to simplify Eq. (4.32) notationally, let us define

$$\boxed{r_x(k,l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k)} \tag{4.33}$$

which is a conjugate symmetric function, $r_x(k,l) = r_x^*(l,k)$, that is similar to the sample autocorrelation sequence defined in Section 3.3.6. In terms of this (deterministic) autocorrelation sequence, Eq. (4.32) becomes

$$\boxed{\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0) \quad ; \quad k = 1, 2, \ldots, p} \tag{4.34}$$

which is a set of p linear equations in the p unknowns $a_p(1), \ldots, a_p(p)$ referred to as the *Prony normal equations*. In matrix form, the normal equations are

$$\begin{bmatrix} r_x(1,1) & r_x(1,2) & r_x(1,3) & \cdots & r_x(1,p) \\ r_x(2,1) & r_x(2,2) & r_x(2,3) & \cdots & r_x(2,p) \\ r_x(3,1) & r_x(3,2) & r_x(3,3) & \cdots & r_x(3,p) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p,1) & r_x(p,2) & r_x(p,3) & \cdots & r_x(p,p) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} r_x(1,0) \\ r_x(2,0) \\ r_x(3,0) \\ \vdots \\ r_x(p,0) \end{bmatrix} \tag{4.35}$$

which may be written concisely as follows

$$\boxed{\mathbf{R}_x \bar{\mathbf{a}}_p = -\mathbf{r}_x} \tag{4.36}$$

where $\mathbf{R}_x$ is a $p \times p$ Hermitian matrix of autocorrelations and $\mathbf{r}_x$ is the column vector

$$\mathbf{r}_x = [r_x(1,0),\ r_x(2,0), \ldots, r_x(p,0)]^T \tag{4.37}$$

[7]The origin of this terminology is as follows. Defining the (infinite-dimensional) vector $\mathbf{e}$ by

$$\mathbf{e} = [e(q+1),\ e(q+2), \ldots]^T$$

and the set of (infinite-dimensional) vectors $\mathbf{x}(k)$ by

$$\mathbf{x}(k) = [x(q+1-k),\ x(q+2-k), \ldots]^T$$

Eq. (4.31) says that the inner product of $\mathbf{e}$ and $\mathbf{x}(k)$ must be zero. Therefore, the orthogonality principle states that the error vector, $\mathbf{e}$, is *orthogonal* to the data vector, $\mathbf{x}(k)$.

We may express the Prony normal equations in a slightly different form as follows. With $\mathbf{X}_q$ the data matrix consisting of p infinite-dimensional column vectors,

$$\mathbf{X}_q = \begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ x(q+2) & x(q+1) & \cdots & x(q-p+3) \\ \vdots & \vdots & & \vdots \end{bmatrix} \tag{4.38}$$

note that the autocorrelation matrix $\mathbf{R}_x$ may be written in terms of $\mathbf{X}_q$ as follows

$$\mathbf{R}_x = \mathbf{X}_q^H \mathbf{X}_q \tag{4.39}$$

The vector of autocorrelations, $\mathbf{r}_x$, may also be expressed in terms of $\mathbf{X}_q$ as follows

$$\mathbf{r}_x = \mathbf{X}_q^H \mathbf{x}_{q+1}$$

where $\mathbf{x}_{q+1} = [x(q+1),\ x(q+2),\ x(q+3), \ldots]^T$. Thus, we have the following equivalent form for the Prony normal equations

$$\boxed{(\mathbf{X}_q^H \mathbf{X}_q)\bar{\mathbf{a}}_p = -\mathbf{X}_q^H \mathbf{x}_{q+1}} \tag{4.40}$$

If $\mathbf{R}_x$ is nonsingular[8] then the coefficients $a_p(k)$ that minimize $\mathcal{E}_{p,q}$ are

$$\bar{\mathbf{a}}_p = -\mathbf{R}_x^{-1} \mathbf{r}_x$$

or, equivalently,

$$\bar{\mathbf{a}}_p = -(\mathbf{X}_q^H \mathbf{X}_q)^{-1} \mathbf{X}_q^H \mathbf{x}_{q+1} \tag{4.41}$$

Let us now evaluate the minimum value for the modeling error $\mathcal{E}_{p,q}$. With

$$\begin{aligned} \mathcal{E}_{p,q} &= \sum_{n=q+1}^{\infty} |e(n)|^2 = \sum_{n=q+1}^{\infty} e(n) \left[x(n) + \sum_{k=1}^{p} a_p(k) x(n-k) \right]^* \\ &= \sum_{n=q+1}^{\infty} e(n) x^*(n) + \sum_{n=q+1}^{\infty} e(n) \left[\sum_{k=1}^{p} a_p(k) x(n-k) \right]^* \end{aligned} \tag{4.42}$$

it follows from the orthogonality principle that the second term in Eq. (4.42) is zero, i.e.,

$$\sum_{n=q+1}^{\infty} e(n) \left[\sum_{k=1}^{p} a_p(k) x(n-k) \right]^* = \sum_{k=1}^{p} a_p^*(k) \left[\sum_{n=q+1}^{\infty} e(n) x^*(n-k) \right] = 0$$

Thus, the minimum value of $\mathcal{E}_{p,q}$, which we will denote by $\epsilon_{p,q}$, is

$$\epsilon_{p,q} = \sum_{n=q+1}^{\infty} e(n) x^*(n) = \sum_{n=q+1}^{\infty} \left[x(n) + \sum_{k=1}^{p} a_p(k) x(n-k) \right] x^*(n) \tag{4.43}$$

[8]From the factorization of $\mathbf{R}_x$ given in Eq. (4.39), it follows that $\mathbf{R}_x$ is positive semidefinite. Although this does not guarantee that $\mathbf{R}_x$ is nonsingular, as we will see in Chapter 5, it does guarantee that the roots of the filter $A_p(z)$ will lie on or inside the unit circle (see, for example, Property 3 on p. 228).

which, in terms of the autocorrelation sequence $r_x(k,l)$, may be written as

$$\boxed{\epsilon_{p,q} = r_x(0,0) + \sum_{k=1}^{p} a_p(k) r_x(0,k)} \tag{4.44}$$

It is important to keep in mind that since the orthogonality principle is used in the derivation of Eq. (4.44), this expression is only valid for those coefficients $a_p(k)$ that satisfy the normal equations. To compute the modeling error for an arbitrary set of coefficients, it is necessary to use Eq. (4.42), which has no restrictions on $a_p(k)$.

Having derived an expression for the minimum error, the normal equations may be written in a slightly different form that will be useful later. If we bring the vector on the right-hand side of Eq. (4.35) to the left and absorb it into the autocorrelation matrix by adding a leading coefficient of 1 to the vector of coefficients $a_p(k)$ we have

$$\left[\begin{array}{c|cccc} r_x(1,0) & r_x(1,1) & r_x(1,2) & \cdots & r_x(1,p) \\ r_x(2,0) & r_x(2,1) & r_x(2,2) & \cdots & r_x(2,p) \\ r_x(3,0) & r_x(3,1) & r_x(3,2) & \cdots & r_x(3,p) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p,0) & r_x(p,1) & r_x(p,2) & \cdots & r_x(p,p) \end{array}\right] \left[\begin{array}{c} 1 \\ \hline a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.45}$$

Since the matrix on the left side of Eq. (4.45) has p rows and $p+1$ columns, we may make it square by augmenting these equations with the addition of Eq. (4.44) as follows:

$$\left[\begin{array}{cccc} r_x(0,0) & r_x(0,1) & r_x(0,2) & \cdots & r_x(0,p) \\ \hline r_x(1,0) & r_x(1,1) & r_x(1,2) & \cdots & r_x(1,p) \\ r_x(2,0) & r_x(2,1) & r_x(2,2) & \cdots & r_x(2,p) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p,0) & r_x(p,1) & r_x(p,2) & \cdots & r_x(p,p) \end{array}\right] \left[\begin{array}{c} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} \epsilon_{p,q} \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.46}$$

Using matrix notation, Eq. (4.46) may be written as

$$\boxed{\mathbf{R}_x \mathbf{a}_p = \epsilon_{p,q} \mathbf{u}_1} \tag{4.47}$$

where $\mathbf{R}_x$ is a Hermitian matrix having $p+1$ rows and $p+1$ columns, $\mathbf{u}_1$ is the unit vector

$$\mathbf{u}_1 = [1, 0, \ldots, 0]^T$$

and

$$\mathbf{a}_p = [1, a_p(1), \ldots, a_p(p)]^T$$

is the augmented vector of filter coefficients. Equation (4.47) is sometimes referred to as the *augmented normal equations*.

Once the coefficients $a_p(k)$ have been found, either by solving Eq. (4.36) or Eq. (4.47), the second step in Prony's method is to find the numerator coefficients. There are several different ways that this may be done. In what is referred to as Prony's method, these coefficients are found by setting the error

$$e(n) = a_p(n) * x(n) - b_q(n)$$

equal to zero for $n = 0, \ldots, q$. In other words, the coefficients are computed in exactly the

Table 4.2 ***Prony's Method of Modeling a Signal as the Unit Sample Response of a Linear Shift-Invariant System Having p Poles and q Zeros***

Normal equations

$$\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0) \quad ; \quad k = 1, 2, \ldots, p$$

$$r_x(k,l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k) \quad ; \quad k, l \geq 0$$

Numerator

$$b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \quad ; \quad n = 0, 1, \ldots, q$$

Minimum error

$$\epsilon_{p,q} = r_x(0,0) + \sum_{k=1}^{p} a_p(k) r_x(0,k)$$

same way as they are in the Padé approximation method,

$$b_q(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \tag{4.48}$$

Another approach, described in Section 4.4.2, is to use Shanks' method, which produces a set of coefficients $b_q(k)$ that minimize a least squares error. Yet another method is to formulate the MA parameter estimation problem as a pair of AR parameter estimation problems using Durbin's method, which is described in Section 4.7.3. Prony's method is summarized in Table 4.2.

Example 4.4.1 ***Prony's Method***

Given the signal $x(n)$ consisting of a single pulse of length N,

$$x(n) = \begin{cases} 1 & ; \; n = 0, 1, \ldots, N-1 \\ 0 & ; \; \text{else} \end{cases}$$

let us use Prony's method to model $x(n)$ as the unit sample response of a linear shift-invariant filter having one pole and one zero,

$$H(z) = \frac{b(0) + b(1)z^{-1}}{1 + a(1)z^{-1}}$$

With $p = 1$, the normal equations are

$$a(1)r_x(1,1) = -r_x(1,0)$$

Therefore, with

$$r_x(1,1) = \sum_{n=2}^{\infty} x^2(n-1) = N - 1$$

$$r_x(1,0) = \sum_{n=2}^{\infty} x(n)x(n-1) = N - 2$$

then

$$a(1) = -\frac{r_x(1,0)}{r_x(1,1)} = -\frac{N-2}{N-1}$$

and the denominator of $H(z)$ becomes

$$A(z) = 1 - \frac{N-2}{N-1}z^{-1}$$

For the numerator coefficients, we have

$$\begin{aligned} b(0) &= x(0) = 1 \\ b(1) &= x(1) + a(1)x(0) = 1 - \frac{N-2}{N-1} = \frac{1}{N-1} \end{aligned}$$

Thus, the model for $x(n)$ is

$$H(z) = \frac{1 + \frac{1}{N-1}z^{-1}}{1 - \frac{N-2}{N-1}z^{-1}}$$

Finally, for the minimum squared error we have

$$\epsilon_{1,1} = r_x(0,0) + a(1)r_x(0,1)$$

Since

$$r_x(0,0) = \sum_{n=2}^{\infty} x^2(n) = N-2$$

then

$$\epsilon_{1,1} = (N-2) - \frac{N-2}{N-1}(N-2) = \frac{N-2}{N-1}$$

For example, if $N = 21$, then the model is

$$H(z) = \frac{1 + 0.05z^{-1}}{1 - 0.95z^{-1}}$$

and the unit sample response that is used to approximate $x(n)$ is

$$h(n) = \delta(n) + (0.95)^{n-1}u(n-1)$$

The error,

$$e(n) = a(n) * x(n) - b(n)$$

is shown in Fig. 4.7*a* and corresponds to a minimum squared error of

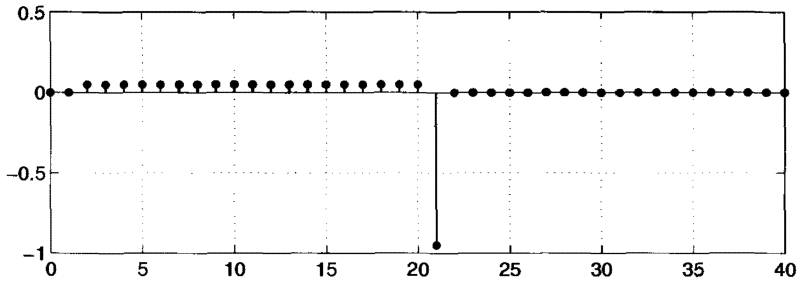
$$\epsilon_{1,1} = 0.95$$

By comparison, the error

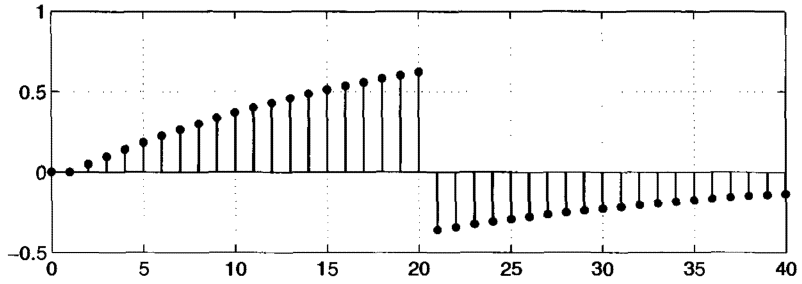
$$e'(n) = x(n) - h(n)$$

which is the difference between the signal and the model, is shown in Fig. 4.7*b*. The sum of the squares of $e'(n)$ is

$$\sum_{n=0}^{\infty}[e'(n)]^2 \approx 4.5954$$



(a) The error sequence, $e(n)$.



(b) The error $e'(n) = x(n) - h(n)$.

Figure 4.7 *Prony's method to model a pulse, $x(n) = u(n) - u(n-21)$, as the unit sample response of a linear shift-invariant system having one pole and one zero ($p = q = 1$).*

Recall that it is this error that the direct method attempts to minimize. Finally, it is interesting to compare this model with that obtained using the Padé approximation. In Padé's method the model coefficients are found by solving Eq. (4.11), which, for our example, is

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ a(1) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ 0 \end{bmatrix}$$

Solving for $a(1)$, $b(0)$, and $b(1)$, we obtain the model

$$H(z) = \frac{1}{1 - z^{-1}}$$

which corresponds to a unit sample response of

$$h(n) = u(n)$$

Thus, the model error $e(n)$ is

$$e(n) = a(n) * x(n) - b(n) = x(n) - x(n-1) - \delta(n) = \begin{cases} 0 & ; \quad n < N \\ -1 & ; \quad n = N \\ 0 & ; \quad n > N \end{cases}$$

In the next example, we consider once again the design of a digital lowpass filter. This time, however, we will use Prony's method and compare it to the filter designed in the previous section using the Padé approximation.

Example 4.4.2 *Filter Design Using Prony's Method*

As in Example 4.3.4, let us design a linear phase lowpass filter having a cutoff frequency of $\pi/2$. The frequency response of the ideal lowpass filter is

$$I(e^{j\omega}) = \begin{cases} e^{-jn_d\omega} & ; \quad |\omega| < \pi/2 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

which has a unit sample response

$$i(n) = \frac{\sin\left[(n - n_d)\pi/2\right]}{(n - n_d)\pi}$$

We will use Prony's method to approximate $i(n)$ using a fifth-order model containing $p = 5$ poles and $q = 5$ zeros. As in the previous example, the delay will be $n_d = 5$.

With $p = q = 5$, the coefficients $a(k)$ may be found by solving Eq. (4.36). The solution is[9]

$$\mathbf{a} = [1.0, \ -1.9167, \ 2.3923, \ -1.9769, \ 1.0537, \ -0.2970]^T$$

Next, the coefficients $b(k)$ are found from Eq. (4.48). The result is

$$\mathbf{b} = [0.0637, \ -0.1220, \ 0.0462, \ 0.0775, \ 0.1316, \ 0.0807]^T$$

The magnitude of the frequency response of this filter is shown in Fig. 4.8 (dashed line). For comparison, also shown is the magnitude of the frequency response of a fifth-order digital elliptic filter that was designed to have a passband ripple of 0.1dB and a stopband attenuation of 40dB [8]. Although not as good as the elliptic filter, comparing this frequency response to that given in Fig. 4.4 we see that it is much better than the Padé approximation. Finally, shown in Fig. 4.9 is the difference, $e'(n) = i(n) - h(n)$ for $n \leq 80$, between the

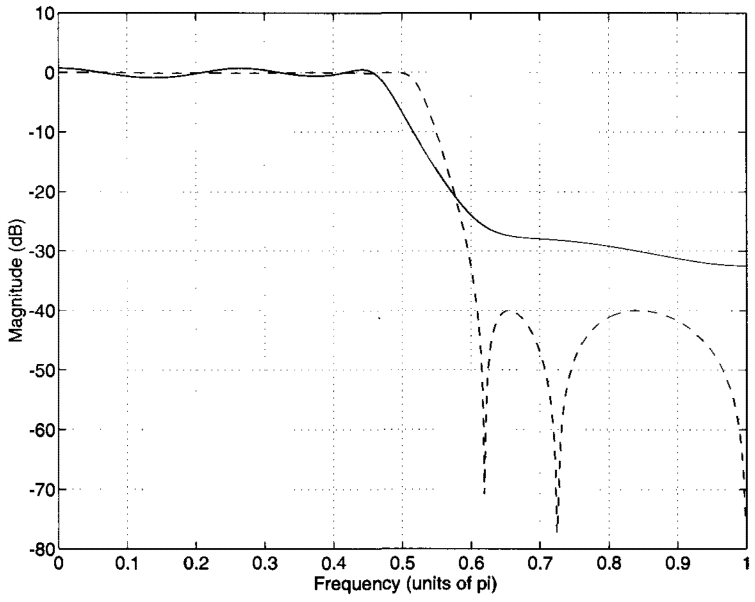


Figure 4.8 *Lowpass filter design using Prony's method. The frequency response of the filter having five poles and five zeros is shown (solid line) along with the frequency response of a fifth-order elliptic filter (dashed line).*

[9]This solution was obtained using the MATLAB program, `prony.m`, given in Fig. 4.10.

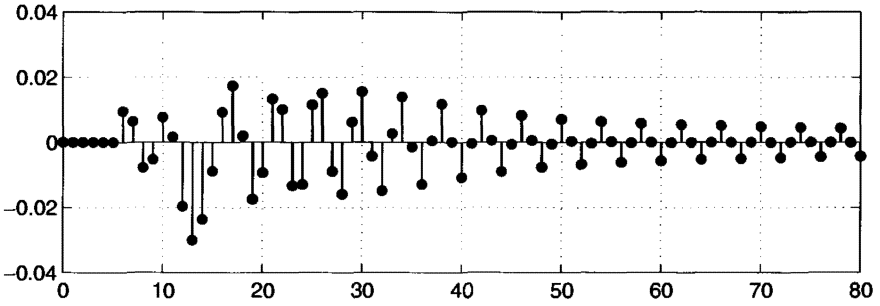


Figure 4.9 *The difference, $e'(n)$, between the unit sample response of the ideal lowpass filter, $i(n)$, and the unit sample response of the filter designed using Prony's method.*

ideal unit sample response and the unit sample response of the filter designed using Prony's method. The total squared error with this design is

$$\mathcal{E}_{LS} = \sum_n [e'(n)]^2 = 0.0838$$

compared to a squared error of 0.4426 for the Padé approximation. Comparing this design with Fig. 4.5 we see that the Prony design produces a much better approximation to the unit sample response of the ideal lowpass filter.

In the development of Prony's method given above, the normal equations were derived by differentiating the error $\mathcal{E}_{p,q}$ with respect to $a_p^*(k)$. We may also formulate Prony's method in terms of finding the least squares solution to a set of overdetermined linear equations. Specifically, note that in Prony's method we would like to find a set of coefficients $a_p(k)$ such that

$$x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) = 0 \quad ; \quad n > q \tag{4.49}$$

i.e., we would like $e(n)$ to be equal to zero for $n > q$. In matrix form, these equations may be expressed as

$$\begin{bmatrix} x(q) & x(q-1) & \cdots & x(q-p+1) \\ x(q+1) & x(q) & \cdots & x(q-p+2) \\ x(q+2) & x(q+1) & \cdots & x(q-p+3) \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} x(q+1) \\ x(q+2) \\ x(q+3) \\ \vdots \end{bmatrix} \tag{4.50}$$

or, equivalently, as

$$\boxed{\mathbf{X}_q \bar{\mathbf{a}}_p = -\mathbf{x}_{q+1}} \tag{4.51}$$

where $\mathbf{X}_q$ is a convolution matrix, identical to the one that was defined in Eq. (4.38), and $\mathbf{x}_{q+1}$ is the data vector on the right side of Eq. (4.50). Since Eq. (4.51) represents an infinite set of linear equations in p unknowns, in general it is not possible to find values for $a_p(k)$ that solve these equations exactly. Therefore, suppose that we find the least squares solution to these equations. As discussed in Section 2.3.6 (p. 33), the least squares solution is found

by solving the *normal equations*

$$(\mathbf{X}_q^H \mathbf{X}_q)\tilde{\mathbf{a}}_p = -\mathbf{X}_q^H \mathbf{x}_{q+1} \tag{4.52}$$

However, these equations are identical to those given in Eq. (4.40). Therefore, finding the least squares solution to Eq. (4.51) and solving the normal equations in Eq. (4.36) are equivalent formulations of the Prony approximation problem. A MATLAB program for Prony's method that is based on finding the least squares solution to Eq. (4.51) with the numerator coefficients computed using Eq. (4.48) is given in Fig. 4.10.

4.4.2 Shanks' Method

In Prony's method, the moving average (numerator) coefficients are found by setting the error

$$e(n) = a_p(n) * x(n) - b_q(n)$$

equal to zero for $n = 0, 1, \ldots, q$. Although this forces the model to be exact over the interval $[0, q]$, it does not take into account the data for $n > q$. A better approach, suggested by Shanks [14], is to perform a least squares minimization of the model error

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

over the entire length of the data record. To develop this idea, we begin by noting that the filter used to model $x(n)$ may be viewed as a cascade of two filters, $B_q(z)$ and $A_p(z)$,

$$H(z) = B_q(z)\left\{\frac{1}{A_p(z)}\right\}$$

As shown in Fig. 4.11, once $A_p(z)$ has been determined, we may compute $g(n)$, the unit sample response of the filter $1/A_p(z)$ using, for example, the recursion[10]

$$g(n) = \delta(n) - \sum_{k=1}^{p} a_p(k)g(n-k)$$

Prony's Method

```
function [a,b,err] = prony(x,p,q)
%
   x   = x(:);
   N   = length(x);
   if p+q>=length(x), error('Model order too large'), end
      X   = convm(x,p+1);
      Xq  = X(q+1:N+p-1,1:p);
      a   = [1;-Xq\X(q+2:N+p,1)];
      b   = X(1:q+1,1:p+1)*a;
      err = x(q+2:N)'*X(q+2:N,1:p+1)*a;
      end;
```

Figure 4.10 *A* MATLAB *program for Prony's method of modeling a signal using p poles and q zeros. Note that this program calls* convm.m. *(See Appendix.)*

[10] Although we should probably use the notation $g_p(n)$, this extra subscript makes the notation in the development of Shanks' method unnecessarily complicated. Therefore, we will drop the subscript and simply write $g(n)$ for the input to the filter $B_q(z)$.

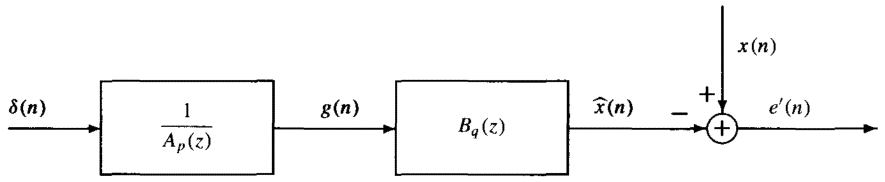


Figure 4.11 *Signal model for Shanks' method. The denominator $A_p(z)$ is found using Prony's method, and then the numerator $B_q(z)$ is found by minimizing the sum of the squares of the error $e'(n)$.*

with $g(n) = 0$ for $n < 0$. It then remains to find the coefficients of the filter $B_q(z)$ that produce the best approximation to $x(n)$ when the input to the filter is $g(n)$. Instead of forcing $e(n)$ to zero for the first $q + 1$ values of n as in Prony's method, Shanks' method minimizes the squared error

$$\mathcal{E}_S = \sum_{n=0}^{\infty} |e'(n)|^2 \tag{4.53}$$

where

$$e'(n) = x(n) - \hat{x}(n) = x(n) - \sum_{l=0}^{q} b_q(l) g(n-l) \tag{4.54}$$

Note that although $\mathcal{E}_S$ is the same error used in the direct method, which led to a set of nonlinear equations, in Shanks' method the denominator coefficients are fixed and the minimization of $\mathcal{E}_S$ is carried out only with respect to the numerator coefficients $b_q(k)$. Since $e'(n)$ is *linear* in the coefficients $b_q(k)$, this problem is much easier to solve. To find the coefficients $b_q(k)$ that minimize $\mathcal{E}_S$, we set the partial derivative of $\mathcal{E}_S$ with respect to $b_q^*(k)$ equal to zero for $k = 0, 1, \ldots, q$ as follows

$$\frac{\partial \mathcal{E}_S}{\partial b_q^*(k)} = \sum_{n=0}^{\infty} e'(n) \frac{\partial [e'(n)]^*}{\partial b_q^*(k)} = -\sum_{n=0}^{\infty} e'(n) g^*(n-k) = 0 \quad ; \quad k = 0, 1, \ldots, q \tag{4.55}$$

Substituting Eq. (4.54) into Eq. (4.55) we have

$$-\sum_{n=0}^{\infty} \left\{ x(n) - \sum_{l=0}^{q} b_q(l) g(n-l) \right\} g^*(n-k) = 0$$

or

$$\sum_{l=0}^{q} b_q(l) \left[\sum_{n=0}^{\infty} g(n-l) g^*(n-k) \right] = \sum_{n=0}^{\infty} x(n) g^*(n-k) \quad ; \quad k = 0, 1, \ldots, q \tag{4.56}$$

As we did in Prony's method, if we let $r_g(k, l)$ denote the (deterministic) autocorrelation of $g(n)$,

$$r_g(k, l) = \sum_{n=0}^{\infty} g(n-l) g^*(n-k) \tag{4.57}$$

and if we define

$$r_{xg}(k) = \sum_{n=0}^{\infty} x(n) g^*(n-k) \tag{4.58}$$

to be the (deterministic) cross-correlation between $x(n)$ and $g(n)$, then Eq. (4.56) becomes

$$\boxed{\sum_{l=0}^{q} b_q(l) r_g(k,l) = r_{xg}(k) \quad ; \quad k = 0, 1, \ldots, q} \tag{4.59}$$

In matrix form, these equations are

$$\begin{bmatrix} r_g(0,0) & r_g(0,1) & r_g(0,2) & \cdots & r_g(0,q) \\ r_g(1,0) & r_g(1,1) & r_g(1,2) & \cdots & r_g(1,q) \\ r_g(2,0) & r_g(2,1) & r_g(2,2) & \cdots & r_g(2,q) \\ \vdots & \vdots & \vdots & & \vdots \\ r_g(q,0) & r_g(q,1) & r_g(q,2) & \cdots & r_g(q,q) \end{bmatrix} \begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix} = \begin{bmatrix} r_{xg}(0) \\ r_{xg}(1) \\ r_{xg}(2) \\ \vdots \\ r_{xg}(q) \end{bmatrix} \tag{4.60}$$

We may simplify these equations considerably, however, by observing that $r_g(k+1, l+1)$ may be expressed in terms of $r_g(k,l)$ as follows

$$\begin{aligned} r_g(k+1, l+1) &= \sum_{n=0}^{\infty} g(n - [l+1]) g^*(n - [k+1]) \\ &= \sum_{n=-1}^{\infty} g(n-l) g^*(n-k) = r_g(k,l) + g(-1-l) g^*(-1-k) \end{aligned} \tag{4.61}$$

Since $g(n) = 0$ for $n < 0$, the last term in Eq. (4.61) is zero when $k \geq 0$ or $l \geq 0$ and

$$r_g(k+1, l+1) = r_g(k,l)$$

Therefore, each term $r_g(k,l)$ in Eq. (4.60) depends only on the difference between the indices k and l. Consequently, with a slight abuse in notation, we will define

$$r_g(k-l) \equiv r_g(k,l) = \sum_{n=0}^{\infty} g(n-l) g^*(n-k) \tag{4.62}$$

Replacing $r_g(k,l)$ with $r_g(k-l)$ in Eq. (4.59), we have

$$\boxed{\sum_{l=0}^{q} b_q(l) r_g(k-l) = r_{xg}(k) \quad ; \quad k = 0, \ldots, q} \tag{4.63}$$

Since $r_x(k-l)$ is conjugate symmetric,

$$r_x(k-l) = r_x^*(l-k)$$

writing Eq. (4.63) in matrix form, we have

$$\begin{bmatrix} r_g(0) & r_g^*(1) & r_g^*(2) & \cdots & r_g^*(q) \\ r_g(1) & r_g(0) & r_g^*(1) & \cdots & r_g^*(q-1) \\ r_g(2) & r_g(1) & r_g(0) & \cdots & r_g^*(q-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_g(q) & r_g(q-1) & r_g(q-2) & \cdots & r_g(0) \end{bmatrix} \begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix} = \begin{bmatrix} r_{xg}(0) \\ r_{xg}(1) \\ r_{xg}(2) \\ \vdots \\ r_{xg}(q) \end{bmatrix}$$

or

$$\boxed{\mathbf{R}_g\mathbf{b}_q = \mathbf{r}_{xg}} \tag{4.64}$$

where $\mathbf{R}_g$ is a $(q+1) \times (q+1)$ Hermitian matrix.

To find the minimum squared error, we may proceed as in Prony's method and write

$$\begin{aligned}\mathcal{E}_S &= \sum_{n=0}^{\infty} |e'(n)|^2 = \sum_{n=0}^{\infty} e'(n)\left[x(n) - \sum_{k=0}^{q} b_q(k)g(n-k)\right]^* \\ &= \sum_{n=0}^{\infty} e'(n)x^*(n) - \sum_{k=0}^{q} b_q^*(k)\left[\sum_{n=0}^{\infty} e'(n)g^*(n-k)\right]\end{aligned} \tag{4.65}$$

Using the orthogonality of $e'(n)$ and $g(n)$ in Eq. (4.55), it follows that the second term in Eq. (4.65) is equal to zero. Therefore, the minimum error is

$$\begin{aligned}\{\mathcal{E}_S\}_{\min} &= \sum_{n=0}^{\infty} e'(n)x^*(n) = \sum_{n=0}^{\infty}\left[x(n) - \sum_{k=0}^{q} b_q(k)g(n-k)\right]x^*(n) \\ &= \sum_{n=0}^{\infty} |x(n)|^2 - \sum_{k=0}^{q} b_q(k)\left[\sum_{n=0}^{\infty} g(n-k)x^*(n)\right]\end{aligned} \tag{4.66}$$

In terms of $r_x(k)$ and $r_{xg}(k)$, the minimum error becomes

$$\{\mathcal{E}_S\}_{\min} = r_x(0) - \sum_{k=0}^{q} b_q(k)r_{gx}(-k)$$

or, since $r_{gx}(-k) = r_{xg}^*(k)$,

$$\boxed{\{\mathcal{E}_S\}_{\min} = r_x(0) - \sum_{k=0}^{q} b_q(k)r_{xg}^*(k)} \tag{4.67}$$

As we did with Prony's method, we may also formulate Shanks' method in terms of finding the least squares solution to a set of overdetermined linear equations. Specifically, setting $e'(n) = 0$ for $n \geq 0$ in Eq. (4.54) and writing these equations in matrix form we have

$$\begin{bmatrix} g(0) & 0 & 0 & \cdots & 0 \\ g(1) & g(0) & 0 & \cdots & 0 \\ g(2) & g(1) & g(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}\begin{bmatrix} b_q(0) \\ b_q(1) \\ b_q(2) \\ \vdots \\ b_q(q) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \end{bmatrix} \tag{4.68}$$

or, equivalently, as

$$\boxed{\mathbf{G}_0\mathbf{b}_q = \mathbf{x}_0} \tag{4.69}$$

The least squares solution is found by solving the linear equations

$$\boxed{(\mathbf{G}_0^H\mathbf{G}_0)\mathbf{b}_q = \mathbf{G}_0^H\mathbf{x}_0} \tag{4.70}$$

Table 4.3 *Shanks' Method for Finding the Zeros in a Pole-Zero Model*

Numerator

$$\sum_{l=0}^{q} b_q(l) r_g(k-l) = r_{xg}(k) \quad ; \quad k = 0, \ldots, q$$

Minimum error

$$\{\mathcal{E}_S\}_{\min} = r_x(0) - \sum_{k=0}^{q} b_q(k) r_{xg}^*(k)$$

Definitions

$$r_g(k-l) = \sum_{n=0}^{\infty} g(n-l) g^*(n-k)$$

$$r_{xg}(k) = \sum_{n=0}^{\infty} x(n) g^*(n-k)$$

$$g(n) = \delta(n) - \sum_{k=1}^{p} a_p(k) g(n-k)$$

From a computational point of view, Shanks' method is more involved than Prony's method. In addition to having to compute the sequence $g(n)$, it is necessary to evaluate the autocorrelation of $g(n)$ and the cross-correlation between $x(n)$ and $g(n)$. In spite of this extra computation, however, in some cases the reduction in the mean square error may be desirable, particularly in applications that do not require that the signal modeling be performed in real time.

Shanks' method is summarized in Table 4.3 and a MATLAB program is given in Fig. 4.12. This program is based on finding the least squares solution to the overdetermined set of linear equations given in Eq. (4.68).

Shanks' method

```
function [a,b,err] = shanks(x,p,q)
%
   x   = x(:);
   N   = length(x);
   if p+q>=length(x), error('Model order too large'), end
      a   = prony(x,p,q);
      u   = [1; zeros(N-1,1)];
      g   = filter(1,a,u);
      G   = convm(g,q+1);
      b   = G(1:N,:)\x;
      err = x'*x-x'*G(1:N,1:q+1)*b;
      end;
```

Figure 4.12 *A* MATLAB *program for Shanks' method of modeling a signal using p poles and q zeros. Note that this program calls* `convm.m` *and* `prony.m` *(see Appendix).*

Example 4.4.3 *Shanks' Method*

As an example illustrating the use of Shanks' method, let us reconsider the problem in Example 4.4.1 of modeling a unit pulse of length N,

$$x(n) = \begin{cases} 1 & ; \ n = 0, 1, \ldots, N-1 \\ 0 & ; \ \text{else} \end{cases}$$

With $p = q = 1$, the solution to the Prony normal equations for $A(z)$ was found to be

$$A(z) = 1 - \frac{N-2}{N-1} z^{-1}$$

Using Shanks' method to find the numerator polynomial, $B(z) = b(0) + b(1)z^{-1}$, we begin by evaluating $g(n)$. Since

$$G(z) = \frac{1}{A(z)} = \frac{1}{1 - \dfrac{N-2}{N-1} z^{-1}}$$

it follows that

$$g(n) = \left(\frac{N-2}{N-1}\right)^n u(n)$$

For the autocorrelation of $g(n)$ we have

$$r_g(0) = \sum_{n=0}^{\infty} g^2(n) = \sum_{n=0}^{\infty} \left(\frac{N-2}{N-1}\right)^{2n} = \frac{1}{1 - \left(\dfrac{N-2}{N-1}\right)^2}$$

$$r_g(1) = \sum_{n=0}^{\infty} g(n)g(n-1) = \frac{N-2}{N-1} r_g(0)$$

and for the cross-correlation between $g(n)$ and $x(n)$ we have

$$r_{xg}(0) = \sum_{n=0}^{\infty} x(n)g(n) = \sum_{n=0}^{N-1} g(n) = \frac{1 - \left(\dfrac{N-2}{N-1}\right)^N}{1 - \left(\dfrac{N-2}{N-1}\right)} = (N-1)\left[1 - \left(\frac{N-2}{N-1}\right)^N\right]$$

$$r_{xg}(1) = \sum_{n=0}^{\infty} x(n)g(n-1) = \sum_{n=0}^{N-1} g(n-1) = (N-1)\left[1 - \left(\frac{N-2}{N-1}\right)^{N-1}\right]$$

Therefore,

$$\mathbf{R}_g = \frac{1}{1 - \left(\dfrac{N-2}{N-1}\right)^2} \begin{bmatrix} 1 & \dfrac{N-2}{N-1} \\ \dfrac{N-2}{N-1} & 1 \end{bmatrix}$$

and

$$\mathbf{r}_{xg} = (N-1)\begin{bmatrix} 1-\left(\dfrac{N-2}{N-1}\right)^{N} \\ 1-\left(\dfrac{N-2}{N-1}\right)^{N-1} \end{bmatrix}$$

Solving Eq. (4.60) for $b(0)$ and $b(1)$ we find

$$b(0) = 1$$

$$b(1) = (N-1)\left[\frac{1}{N-1} - \left(\frac{N-2}{N-1}\right)^{N-1} + \left(\frac{N-2}{N-1}\right)^{N+1}\right]$$

For example, with $N = 21$ the model becomes

$$H(z) = \frac{1+0.301z^{-1}}{1-0.950z^{-1}}$$

Finally, we have for the modeling error,

$$\{\mathcal{E}_S\}_{\min} = r_x(0) - b(0)r_{xg}(0) - b(1)r_{xg}(1) = 3.95$$

Recall that with Prony's method the model coefficients are $b(0) = 1$, $b(1) = 0.05$, and $a(1) = -0.95$ and the sum of the squares of $e'(n)$ is $\mathcal{E} = 4.5954$. Thus, Shanks' method results in a small improvement.

Although the coefficients $b_q(k)$ using Shanks' method are found by minimizing the error in Eq. (4.53), there may be instances in which other errors may be more appropriate. For example, it may be preferable to consider using the error

$$\mathcal{E} = \sum_{n=q+1}^{\infty} |e'(n)|^2$$

In this case, the interval over which the error is minimized, $[q+1, \infty)$, is the same as that used to derive the denominator coefficients $a_p(k)$. Alternatively, it may be preferable to consider minimizing the squared error over a finite interval,

$$\mathcal{E} = \sum_{n=0}^{N-1} |e'(n)|^2$$

This may be appropriate for signals that have a significant percentage of their energy over the interval $[0, N-1]$ or if the model is only going to be used to represent the signal over a finite interval. If this is the case, however, then one may also want to modify Prony's method to minimize the error $\mathcal{E}_{p,q}$ in Eq. (4.29) over the same interval.

4.4.3 All-Pole Modeling

In this section, we consider the special case of all-pole signal modeling using Prony's method. All-pole models are important for several reasons. First of all, in some applications the physical process by which a signal is generated will result in an all-pole (autoregressive)

signal. In speech processing, for example, an acoustic tube model for speech production leads to an all-pole model for speech [11]. However, even in those applications for which it may not be possible to justify an all-pole model for the signal, one often finds an all-pole model being used. One reason for this is that all-pole models have been found to provide a sufficiently accurate representation for many different types of signals in many different applications. Another reason for the popularity of all-pole models is the special structure that is found in the all-pole Prony normal equations, which leads to fast and efficient algorithms for finding the all-pole parameters (see Chapter 5).

Let $x(n)$ be a signal, either real or complex, that is equal to zero for $n < 0$, and suppose that we would like to model $x(n)$ using an all-pole model of the form

$$H(z) = \frac{b(0)}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} \tag{4.71}$$

With Prony's method, the coefficients of the all-pole model, $a_p(k)$, are found by minimizing the error given in Eq. (4.29) with $q = 0$, i.e.,

$$\mathcal{E}_{p,0} = \sum_{n=1}^{\infty} |e(n)|^2 \tag{4.72}$$

where

$$e(n) = x(n) + \sum_{k=1}^{p} a_p(k) x(n-k)$$

Note that since $x(n) = 0$ for $n < 0$, then the error at time $n = 0$ is equal to $x(0)$, and, therefore, does not depend upon the coefficients $a_p(k)$. Thus, the coefficients that minimize $\mathcal{E}_{p,0}$ are the same as those that minimize the error[11]

$$\boxed{\mathcal{E}_p = \sum_{n=0}^{\infty} |e(n)|^2} \tag{4.73}$$

Since this error is more convenient to use than $\mathcal{E}_{p,0}$, we will formulate the all-pole modeling problem in terms of finding the coefficients $a_p(k)$ that minimize $\mathcal{E}_p$.

From our derivation of Prony's method, repeating the steps that led up to Eq. (4.34) we see that the all-pole normal equations for the coefficients $a_p(k)$ that minimize $\mathcal{E}_p$ are

$$\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0) \quad ; \quad k = 1, 2, \ldots, p \tag{4.74}$$

where

$$r_x(k,l) = \sum_{n=0}^{\infty} x(n-l) x^*(n-k)$$

Note that in this definition for $r_x(k,l)$, the sum begins at $n = 0$ instead of $n = 1$ as we would have if we used the definition given in Eq. (4.33) with $q = 0$. This is due to the change that

[11] Since $e(n)$ is equal to zero for $n < 0$ this error may, in fact, be defined so that the sum extends from $n = -\infty$ to $n = \infty$.

was made in the limit on the sum in Eq. (4.73) for the error $\mathcal{E}_p$ that is being minimized. As in Shanks' method, the all-pole normal equations may be simplified considerably due to the underlying structure that is found in $r_x(k,l)$. Specifically, note that $r_x(k+1,l+1)$ may be expressed in terms of $r_x(k,l)$ as follows:

$$\begin{aligned} r_x(k+1,l+1) &= \sum_{n=0}^{\infty} x\big(n-[l+1]\big)x^*\big(n-[k+1]\big) \\ &= \sum_{n=-1}^{\infty} x(n-l)x^*(n-k) = r_x(k,l) + x(-1-l)x^*(-1-k) \end{aligned} \tag{4.75}$$

Since $x(n)$ is assumed to be equal to zero for $n < 0$, then the last term in Eq. (4.75) is equal to zero whenever $k \geq 0$ or $l \geq 0$. Therefore,

$$r_x(k+1,l+1) = r_x(k,l) \quad ; \quad k \geq 0 \text{ or } l \geq 0 \tag{4.76}$$

so that $r_x(k,l)$ depends only on the difference between the indices k and l. Consequently, with a slight abuse in notation we will define

$$\boxed{r_x(k-l) \equiv r_x(k,l) = \sum_{n=0}^{\infty} x(n-l)x^*(n-k)} \tag{4.77}$$

which is a conjugate symmetric function of k and l,

$$r_x(k-l) = r_x^*(l-k)$$

Replacing $r_x(k,l)$ with $r_x(k-l)$ in Eq. (4.74) we have

$$\boxed{\sum_{l=1}^{p} a_p(l)r_x(k-l) = -r_x(k) \quad ; \quad k = 1, \ldots, p} \tag{4.78}$$

which are the *all-pole normal equations*. Using the conjugate symmetry of $r_x(k)$, these equations may be written in matrix form as follows:

$$\begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2) & \cdots & r_x^*(p-1) \\ r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-2) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p-1) & r_x(p-2) & r_x(p-3) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \\ qr_x(3) \\ \vdots \\ r_x(p) \end{bmatrix} \tag{4.79}$$

Note that the autocorrelation matrix multiplying the vector of coefficients $a_p(k)$ is a Hermitian Toeplitz matrix. As we will see in Chapter 5, this Toeplitz structure allows us to solve the all-pole normal equations efficiently using the Levinson-Durbin recursion. In addition to this, however, note that the Toeplitz structure implies that only $p+1$ values of the autocorrelation sequence need to be computed and stored, compared to $\frac{1}{2}p(p+3)$ values of $r_x(k,l)$ that are found on both sides of Eq. (4.35) when $q \neq 0$. Finally, the minimum

modeling error for the all-pole model follows from Eq. (4.44) and is given by

$$\boxed{\{\mathcal{E}_p\}_{\min} \equiv \epsilon_p = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k)} \tag{4.80}$$

From Eq. (4.50) note that the all-pole coefficients may also be determined by finding the least squares solution to the following set of overdetermined linear equations

$$\begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ x(3) & x(2) & x(1) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \\ \vdots \end{bmatrix} \tag{4.81}$$

Having found the denominator coefficients, we need only determine the numerator coefficient $b(0)$ to complete the model. One possibility, as in the pole-zero Prony method, is to set

$$b(0) = x(0) \tag{4.82}$$

so that $\hat{x}(0) = x(0)$, i.e., the signal generated by the model at time $n = 0$ is equal to the value of the given data at time $n = 0$. Another approach, generally preferable to Eq. (4.82), is to choose $b(0)$ so that the energy in $x(n)$ is equal to the energy in $\hat{x}(n) = h(n)$, i.e.,

$$r_x(0) = r_h(0)$$

This prevents an anomalous or bad data value for $x(0)$ from affecting the scale factor $b(0)$. In order to satisfy this energy matching constraint, $b(0)$ must be chosen as follows

$$b(0) = \sqrt{\epsilon_p} \tag{4.83}$$

We will postpone the proof of this until Section 5.2.3 of Chapter 5 when we have a few more tools. At that time we will also show that this value for $b(0)$ ensures that $r_x(k) = r_h(k)$ for all $|k| \leq p$. This is referred to as the *autocorrelation matching property*. The equations for all-pole modeling using Prony's method are summarized in Table 4.4.

Table 4.4 *All-Pole Modeling Using Prony's Method*

Normal equations

$$\sum_{l=1}^{p} a_p(l) r_x(k-l) = -r_x(k) \quad ; \quad k = 1, \ldots, p$$

$$r_x(k) = \sum_{n=0}^{\infty} x(n) x^*(n-k)$$

Numerator

$$b(0) = \sqrt{\epsilon_p}$$

Minimum error

$$\epsilon_p = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k)$$

Example 4.4.4 *All-Pole Modeling Using Prony's Method*

Let us find a first-order all-pole model of the form

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1}}$$

for the signal

$$x(n) = \delta(n) - \delta(n-1)$$

The autocorrelation sequence is

$$r_x(k) = \begin{cases} 2 & ; \quad k = 0 \\ -1 & ; \quad k = 1 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

The all-pole normal equations for a first-order model are

$$r_x(0)a(1) = -r_x(1)$$

Therefore,

$$a(1) = -\frac{r_x(1)}{r_x(0)} = \frac{1}{2}$$

and the modeling error is

$$\epsilon_1 = r_x(0) + a(1)r_x(1) = 1.5$$

With $b(0)$ chosen to satisfy the energy matching constraint,

$$b(0) = \sqrt{1.5} = 1.2247$$

the model for $x(n)$ becomes

$$H(z) = \frac{1.2247}{1 + 0.5z^{-1}}$$

Let us now find the second-order all-pole model. In this case, the normal equations are

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = -\begin{bmatrix} r_x(1) \\ r_x(2) \end{bmatrix}$$

Using the autocorrelation values computed above, the normal equations become

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Solving for $a(0)$ and $a(1)$ we have

$$a(1) = 2/3$$
$$a(2) = 1/3$$

For this second-order model, the modeling error is

$$\epsilon_2 = r_x(0) + a(1)r_x(1) + a(2)r_x(2) = 2 - 2/3 = 4/3$$

Again, with $b(0)$ chosen to satisfy the energy matching constraint,

$$b(0) = 2/\sqrt{3} = 1.1547$$

the model for $x(n)$ becomes

$$H(z) = \frac{1.1547}{1 + \frac{2}{3}z^{-1} + \frac{1}{3}z^{-2}}$$

Note that since $r_x(k) = 0$ for $k > 1$, for a pth-order all-pole model, the minimum modeling error is

$$\epsilon_p = r_x(0) + a_p(1)r_x(1)$$

In other words, ϵ_p depends only on the value of $a_p(1)$.

As we did in Section 4.4.1, the all-pole normal equations may be written in a different form that will be useful in later discussions. Specifically, bringing the vector on the right side of Eq. (4.79) to the left and absorbing it into the autocorrelation matrix by adding a coefficient of 1 to the vector of model parameters, the normal equations become

$$\left[\begin{array}{c|cccc} r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ r_x(3) & r_x(2) & r_x(1) & \cdots & r_x^*(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{array}\right] \left[\begin{array}{c} 1 \\ \hline a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.84}$$

Since the matrix in Eq. (4.84) has p rows and $(p+1)$ columns, we will add an additional equation to make the matrix square while preserving the Toeplitz structure. The equation that accomplishes this is Eq. (4.80),

$$r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k) = \epsilon_p \tag{4.85}$$

With Eq. (4.85) incorporated into Eq. (4.84) the normal equations become

$$\left[\begin{array}{ccccc} r_x(0) & r_x^*(1) & r_x^*(2) & \cdots & r_x^*(p) \\ r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{array}\right] \left[\begin{array}{c} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{array}\right] = \left[\begin{array}{c} \epsilon_p \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{4.86}$$

which, using vector notation, may be written as

$$\boxed{\mathbf{R}_x \mathbf{a}_p = \epsilon_p \mathbf{u}_1} \tag{4.87}$$

where, as before, $\mathbf{u}_1 = [1, 0, \ldots, 0]^T$ is a unit vector of length $p+1$.

4.4.4 Linear Prediction

In this section, we will establish the equivalence between all-pole signal modeling and a problem known as *linear prediction*. Recall that Prony's method finds the set of all-pole parameters that minimizes the squared error

$$\mathcal{E}_p = \sum_{n=0}^{\infty} |e(n)|^2$$

where

$$e(n) = x(n) + \sum_{k=1}^{p} a_p(k) x(n-k)$$

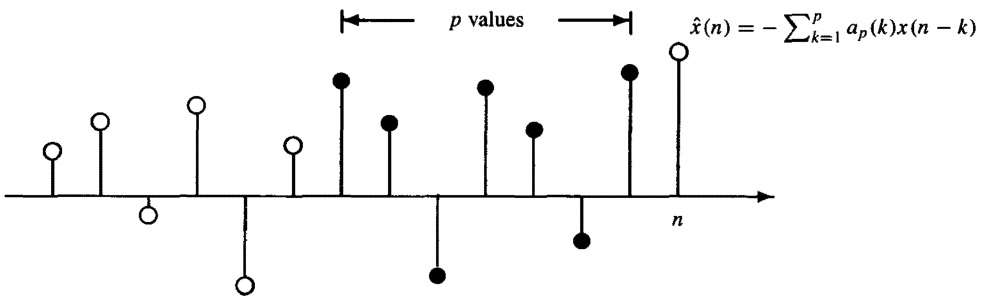


Figure 4.13 *Prony's method viewed as a problem of linear prediction in which a linear combination of the signal values* $x(n-1), x(n-2), \ldots, x(n-p)$ *are used to predict* $x(n)$.

If this error is expressed as follows

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

where

$$\hat{x}(n) = -\sum_{k=1}^{p} a_p(k)x(n-k)$$

then, as shown in Fig. 4.13, $\hat{x}(n)$ is a linear combination of the values of $x(n)$ over the interval $[n-p, n-1]$. Since $e(n)$ is the difference between $x(n)$ and $\hat{x}(n)$, minimizing the sum of the squares of $e(n)$ is equivalent to finding the coefficients $a_p(k)$ that make $\hat{x}(n)$ as close as possible to $x(n)$. Therefore, $\hat{x}(n)$ is an estimate, or prediction, of the signal $x(n)$ in terms of a linear combination of the p previous values of $x(n)$. As a result, the error $e(n)$ is referred to as the *linear prediction error* and the coefficients $a_p(k)$ are called the *linear prediction coefficients*. Furthermore, with

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k)z^{-k}$$

since $x(n) * a_p(n) = e(n)$, then $A_p(z)$ is called the *Prediction Error Filter (PEF)*. In Chapter 6, we will look at the problem of linear prediction within the context of lattice filters, and in Chapter 7 will consider the problem of linear prediction within the context of Wiener filtering.

4.4.5 Application: FIR Least Squares Inverse Filters

Before continuing with our discussion of signal modeling techniques, we turn our attention briefly to another problem that is closely related to signal modeling: the design of an FIR least squares inverse filter. Given a filter $g(n)$, the inverse filter, which we will denote by $g^{-1}(n)$, is one for which[12]

$$g(n) * g^{-1}(n) = \delta(n)$$

or

$$G(z)G^{-1}(z) = 1$$

[12]Although $G^{-1}(z) = 1/G(z)$, the reader is cautioned not to interpret $g^{-1}(n)$ as $1/g(n)$. The superscript -1 is used simply as notation to indicate that when $g^{-1}(n)$ is convolved with $g(n)$ the result is $\delta(n)$.

The design of an inverse filter is of interest in a number of important applications. In a digital communication system, for example, a signal is to be transmitted across a nonideal channel. Assuming that the channel is linear and has a system function $G(z)$, to minimize the chance of making errors at the output of the receiver, we would like to design a *channel equalization filter* whose frequency response is the inverse, or approximate inverse, of the channel response $G(z)$. Thus, the goal is to find an equalizer $H(z)$ such that

$$G(z)H(z) \approx 1$$

A similar problem arises in seismic signal processing in which a recorded seismic signal is the convolution of a source signal $g(n)$ (wavelet) with the impulse response of a layered earth model [13]. Due to the nonimpulsive nature of the wavelet that is generated by the source, such as an airgun, the resolution of the signal will be limited. Therefore, in order to increase the amount of detail that is available in the recorded signal, a *spiking filter* $h(n)$ is designed so that

$$g(n) * h(n) \approx \delta(n)$$

i.e., $h(n)$ attempts to convert (shape) the wavelet $g(n)$ into an impulse [13].

In most applications, the inverse system $H(z) = 1/G(z)$ is not a practical solution. One of the difficulties with this solution is that, unless $G(z)$ is minimum phase, the inverse filter cannot be both causal and stable (see p. 18). Another limitation with the solution is that, in some applications, it may be necessary to constrain $H(z)$ to be an FIR filter. Since the inverse filter will be infinite in length unless $g(n)$ is an all-pole filter, constraining $h(n)$ to be FIR requires that we find the best approximation to the inverse filter.

We may formulate the FIR least squares inverse filtering problem as follows. If $g(n)$ is a causal filter that is to be equalized, the problem is to find an FIR filter $h_N(n)$ of length N such that

$$h_N(n) * g(n) \approx d(n)$$

where $d(n) = \delta(n)$. If we let

$$e(n) = d(n) - h_N(n) * g(n) = d(n) - \sum_{l=0}^{N-1} h_N(l)g(n-l) \tag{4.88}$$

be the difference between $d(n)$ and the equalized signal, $\hat{d}(n) = h_N(n) * g(n)$, as illustrated in Fig. 4.14, then we may use a least squares approach to minimize the sum of the squares of $e(n)$,

$$\mathcal{E}_N = \sum_{n=0}^{\infty} |e(n)|^2 = \sum_{n=0}^{\infty} \left| d(n) - \sum_{l=0}^{N-1} h_N(l)g(n-l) \right|^2 \tag{4.89}$$

Note that this error is the same as that which is minimized in Shanks' method, Eq. (4.54). In fact, comparing Figs. 4.11 and 4.14 we see that the two problems are identical. Therefore, from the solution that we derived for Shanks' method, Eq. (4.63), it follows that the solution for the optimum least squares inverse filter is

$$\boxed{\sum_{l=0}^{N-1} h_N(l)r_g(k-l) = r_{dg}(k) \quad ; \quad k = 0, 1, \ldots, N-1} \tag{4.90}$$

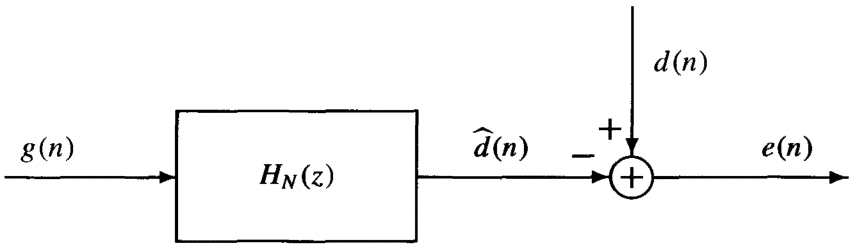


Figure 4.14 *A block diagram illustrating the design of an FIR least squares inverse filter for g(n).*

where

$$r_g(k-l) = \sum_{n=0}^{\infty} g(n-l)g^*(n-k) = r_g^*(l-k)$$

and

$$r_{dg}(k) = \sum_{n=0}^{\infty} d(n)g^*(n-k) \tag{4.91}$$

Since $d(n) = \delta(n)$ and $g(n) = 0$ for $n < 0$, then $r_{dg}(k) = g^*(0)\delta(k)$ and Eq. (4.90), written in matrix form, becomes

$$\begin{bmatrix} r_g(0) & r_g^*(1) & r_g^*(2) & \cdots & r_g^*(N-1) \\ r_g(1) & r_g(0) & r_g^*(1) & \cdots & r_g^*(N-2) \\ r_g(2) & r_g(1) & r_g(0) & \cdots & r_g^*(N-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_g(N-1) & r_g(N-2) & r_g(N-3) & \cdots & r_g(0) \end{bmatrix} \begin{bmatrix} h_N(0) \\ h_N(1) \\ h_N(2) \\ \vdots \\ h_N(N-1) \end{bmatrix} = \begin{bmatrix} g^*(0) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.92}$$

or

$$\boxed{\mathbf{R}_g\mathbf{h}_N = g^*(0)\mathbf{u}_1} \tag{4.93}$$

which is a Toeplitz system of N linear equations in the N unknowns $h_N(n)$. From Eq. (4.67) it follows that the minimum squared error is

$$\{\mathcal{E}_N\}_{\min} = r_d(0) - \sum_{k=0}^{N-1} h_N(k)r_{dg}^*(k)$$

which, since $r_{dg}(k) = g^*(0)\delta(k)$, is equal to

$$\boxed{\{\mathcal{E}_N\}_{\min} = 1 - h_N(0)g(0)} \tag{4.94}$$

As we did in Shanks' method, we may also formulate the FIR least squares inverse filter in terms of finding the least squares solution to a set of overdetermined linear equations. Specifically, setting $e(n) = 0$ for $n \geq 0$ in Eq. (4.88) and writing these equations in matrix

form we have

$$\begin{bmatrix} g(0) & 0 & 0 & \cdots & 0 \\ g(1) & g(0) & 0 & \cdots & 0 \\ g(2) & g(1) & g(0) & \cdots & 0 \\ g(3) & g(2) & g(1) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} h_N(0) \\ h_N(1) \\ h_N(2) \\ \vdots \\ h_N(N-1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{4.95}$$

or

$$\boxed{\mathbf{G}_0\mathbf{h}_N = \mathbf{u}_1} \tag{4.96}$$

where $\mathbf{G}_0$ is a matrix having N infinite-dimensional column vectors, and $\mathbf{u}_1$ is an infinite-dimensional column vector. The least squares solution is found by solving the linear equations

$$\boxed{(\mathbf{G}_0^H\mathbf{G}_0)\mathbf{h}_N = \mathbf{G}_0^H\mathbf{u}_1} \tag{4.97}$$

which is equivalent to Eq. (4.93).

Example 4.4.5 *FIR Least Squares Inverse*

Let us find the FIR least squares inverse for the system having a unit sample response

$$g(n) = \delta(n) - \alpha\delta(n-1)$$

where α is an arbitrary real number. Note that if $|\alpha| > 1$, then $G(z)$ has a zero that is outside the unit circle. In this case, $G(z)$ is not minimum phase and the inverse filter $1/G(z)$ cannot be both causal and stable. However, if $|\alpha| < 1$, then

$$G^{-1}(z) = \frac{1}{G(z)} = \frac{1}{1-\alpha z^{-1}}$$

and the inverse filter is

$$g^{-1}(n) = \alpha^n u(n)$$

We begin by finding the least squares inverse of length $N = 2$. The autocorrelation of $g(n)$ is

$$r_g(k) = \begin{cases} 1+\alpha^2 & ; \quad k = 0 \\ -\alpha & ; \quad k = \pm 1 \\ 0 & ; \quad \text{else} \end{cases}$$

Therefore, Eq. (4.92) is

$$\begin{bmatrix} 1+\alpha^2 & -\alpha \\ -\alpha & 1+\alpha^2 \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and the solution for $h(0)$ and $h(1)$ is

$$h(0) = \frac{1+\alpha^2}{1+\alpha^2+\alpha^4}$$

$$h(1) = \frac{\alpha}{1+\alpha^2+\alpha^4}$$

with a squared error of

$$\{\mathcal{E}\}_{\min} = 1 - h(0) = \frac{\alpha^4}{1+\alpha^2+\alpha^4}$$

The system function of this least squares inverse filter is

$$H(z) = \frac{1+\alpha^2}{1+\alpha^2+\alpha^4} + \frac{\alpha}{1+\alpha^2+\alpha^4}z^{-1} = \frac{1+\alpha^2}{1+\alpha^2+\alpha^4}\left(1 + \frac{\alpha}{1+\alpha^2}z^{-1}\right)$$

which has a zero at

$$z_0 = -\frac{\alpha}{1+\alpha^2}$$

Note that since

$$|z_0| = \left|\frac{\alpha}{1+\alpha^2}\right| = \left|\frac{1}{\alpha + 1/\alpha}\right| < 1$$

then the zero of $H(z)$ is inside the unit circle and $H(z)$ is minimum phase, regardless of whether the zero of $G(z)$ is inside or outside the unit circle.

Let us now look at the least squares inverse, $h_N(n)$, of length N. In this case Eq. (4.92) becomes

$$\begin{bmatrix} 1+\alpha^2 & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1+\alpha^2 & -\alpha & \cdots & 0 \\ 0 & -\alpha & 1+\alpha^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1+\alpha^2 \end{bmatrix} \begin{bmatrix} h_N(0) \\ h_N(1) \\ h_N(2) \\ \vdots \\ h_N(N-1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.98}$$

Solving these equations for arbitrary α and N may be accomplished as follows [12]. For $n = 1, 2, \ldots, N-2$ these equations may be represented by the homogeneous difference equation,

$$-\alpha h_N(n-1) + (1+\alpha^2)h_N(n) - \alpha h_N(n+1) = 0$$

The general solution to this equation is

$$h_N(n) = c_1\alpha^n + c_2\alpha^{-n} \tag{4.99}$$

where c_1 and c_2 are constants that are determined by the boundary conditions at $n = 0$ and $n = N-1$, i.e., the first and last equation in Eq. (4.98),

$$(1+\alpha^2)h_N(0) - \alpha h_N(1) = 1$$

and

$$-\alpha h_N(N-2) + (1+\alpha^2)h_N(N-1) = 0 \tag{4.100}$$

Substituting Eq. (4.99) into Eq. (4.100) we have

$$(1+\alpha^2)[c_1 + c_2] - \alpha[c_1\alpha + c_2\alpha^{-1}] = 1$$

and

$$-\alpha[c_1\alpha^{N-2} + c_2\alpha^{-(N-2)}] + (1+\alpha^2)[c_1\alpha^{N-1} + c_2\alpha^{-(N-1)}] = 0$$

which, after canceling common terms, may be simplified to

$$c_1 + \alpha^2 c_2 = 1$$

and

$$\alpha^{N+1}c_1 + \alpha^{-(N-1)}c_2 = 0$$

or,

$$\begin{bmatrix} 1 & \alpha^2 \\ \alpha^{N+1} & \alpha^{-(N-1)} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The solution for c_1 and c_2 is

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \frac{1}{\alpha^{-(N-1)} - \alpha^{N+3}} \begin{bmatrix} \alpha^{-(N-1)} \\ -\alpha^{N+1} \end{bmatrix}$$

Therefore, $h_N(n)$ is

$$h_N(n) = \begin{cases} \dfrac{\alpha^{n-N} - \alpha^{N-n}}{\alpha^{-N} - \alpha^{N+2}} & ; \quad 0 \le n \le N-1 \\ 0 & ; \quad \text{else} \end{cases}$$

Finally, with

$$h_N(0) = \frac{\alpha^{-N} - \alpha^{N}}{\alpha^{-N} - \alpha^{N+2}}$$

it follows that the squared error is

$$\{\mathcal{E}_N\}_{\min} = 1 - h_N(0)g^*(0) = 1 - \frac{\alpha^{-N} - \alpha^{N}}{\alpha^{-N} - \alpha^{N+2}} = \frac{\alpha^{N} - \alpha^{N+2}}{\alpha^{-N} - \alpha^{N+2}}$$

Let us now look at what happens asymptotically as $N \to \infty$. If $|\alpha| < 1$, then

$$\lim_{N\to\infty} h_N(n) = \frac{\alpha^{n-N}}{\alpha^{-N}} = \alpha^n \quad ; \quad n \ge 0$$

which is the inverse filter, i.e.,

$$\lim_{N\to\infty} h_N(n) = \alpha^n u(n) = g^{-1}(n)$$

and

$$\lim_{N\to\infty} H_N(z) = \frac{1}{1 - \alpha z^{-1}}$$

In addition,

$$\lim_{N\to\infty} \{\mathcal{E}_N\} = 0$$

However, if $|\alpha| > 1$, then

$$\lim_{N\to\infty} h_N(n) = \frac{\alpha^{N-n}}{\alpha^{N+2}} = \alpha^{-n-2} \quad ; \quad n \ge 0$$

and

$$\lim_{N\to\infty} H_N(z) = \frac{\alpha^{-2}}{1 - \alpha^{-1}z^{-1}}$$

which is *not* the inverse filter. The squared error in this case is

$$\lim_{N\to\infty} \{\mathcal{E}_N\} = 1 - \frac{1}{\alpha^2}$$

Note that although $\hat{d}(n) = h_N(n) * g(n)$ does not converge to $\delta(n)$ as $N \to \infty$, taking the limit of $\hat{D}_N(z)$ as $N \to \infty$ we have

$$\lim_{N\to\infty} \hat{D}_N(z) = \lim_{N\to\infty} \hat{H}_N(z)G(z) = \frac{1}{\alpha}\left(\frac{1 - \alpha z^{-1}}{\alpha - z^{-1}}\right)$$

which is an allpass filter, i.e.,

$$|\hat{D}_N(e^{j\omega})| = \frac{1}{\alpha}$$

In many cases, constraining the least squares inverse filter to minimize the difference between $h_N(n) * g(n)$ and $\delta(n)$ is overly restrictive. For example, if a delay may be tolerated,

then we may consider minimizing the error

$$e(n) = \delta(n - n_0) - h_N(n) * g(n)$$

i.e., $d(n) = \delta(n - n_0)$ in Eq. (4.88). In most cases, a nonzero delay will produce a better approximate inverse filter and, in many cases, the improvement will be substantial. The least squares inverse filter with delay is found by solving Eq. (4.90) with $d(n) = \delta(n - n_0)$. In this case, since

$$r_{dg}(k) = \sum_{n=0}^{\infty} d(n)g^*(n-k) = \sum_{n=0}^{\infty} \delta(n - n_0)g^*(n-k) = g^*(n_0 - k)$$

then the equations that define the coefficients of the FIR least squares inverse filter are

$$\mathbf{R}_g\mathbf{h}_N = \begin{bmatrix} g^*(n_0) \\ g^*(n_0 - 1) \\ \vdots \\ g^*(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.101}$$

and the minimum squared error is

$$\{\mathcal{E}_N\}_{\min} = 1 - \sum_{k=0}^{n_0} h_N(k)g^*(n_0 - k)$$

Generalizing Eq. (4.96) to the case of a least squares inverse with a delay, we have

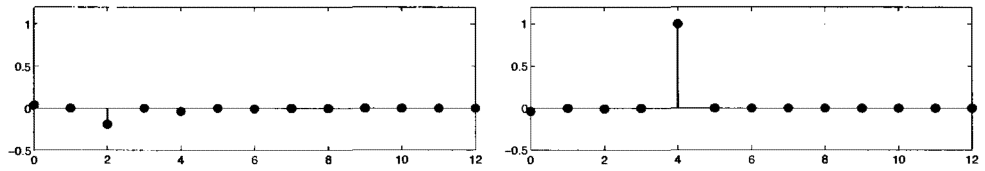
$$\boxed{\mathbf{G}_0\mathbf{h}_N = \mathbf{u}_{n_0+1}} \tag{4.102}$$

where $\mathbf{u}_{n_0+1}$ is a unit vector with a one in the $(n_0 + 1)$st position. A MATLAB program for the design of an FIR least squares inverse filter with delay is given in Fig. 4.15. This program is based on finding the least squares solution to Eq. (4.102).

FIR Least Squares Inverse

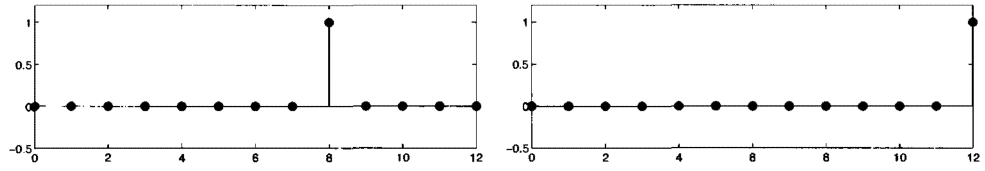
```
function [h,err] = spike(g,n0,n)
%
   g   = g(:);
   m   = length(g);
   if m+n-1<=n0, error('Delay too large'), end
      G   = convm(g,n);
      d   = zeros(m+n-1,1);
      d(n0+1) = 1;
      h   = G\d;
      err = 1 - G(n0+1,:)*h;
      end;
```

Figure 4.15 *A* MATLAB *program for finding the FIR least squares inverse filter for approximating a unit sample at time* $n = n_0$, *i.e.,* $\delta(n - n_0)$. *Note that this program calls* `convm.m` (*see Appendix*).



(a) Delay $n_0 = 0$.

(b) Delay $n_0 = 4$.



(c) Delay $n_0 = 8$.

(d) Delay $n_0 = 12$.

Figure 4.16 *The FIR least squares inverse for* $g(n) = -0.2\delta(n) + \delta(n-2)$. *Shown in each figure is the equalized signal,* $\hat{d}(n) = g(n) * h(n)$, *for different values of the delay* n_0.

Example 4.4.6 ***Least Squares Inverse with Delay***

Let us find the FIR least squares inverse filter of length $N = 11$ for the system

$$G(z) = -0.20 + z^{-2}$$

which has a unit sample response

$$g(n) = -0.20\delta(n) + \delta(n-2)$$

Note that $G(z)$ has two zeros that are outside the unit circle and, therefore, is not minimum phase. Since $h(n)$ is of length 11 and $g(n)$ is of length 3, then

$$\hat{d}(n) = h(n) * g(n)$$

is of length 13. Therefore, the delay n_0 is restricted to the values of $n_0 = 0, 1, \ldots, 12$. Using the MATLAB program in Fig. 4.15 we find that, without a delay ($n_0 = 0$), the squared error is $\mathcal{E} = 0.960$ and that the FIR least squares inverse filter is poor. Specifically, shown in Fig. 4.16*a* is the equalized signal, $\hat{d}(n) = g(n) * h(n)$, which ideally, should be equal to a unit sample, $\delta(n)$. Shown in Fig. 4.16*b–d*, on the other hand, are the equalized signals for delays of $n_0 = 4$, 8, and 12. The corresponding squared errors are $\mathcal{E} = 1.5 \times 10^{-3}$, 2.457×10^{-6}, and 3.9322×10^{-9}, respectively. For this particular example, a delay of $n_0 = 12$ results in the best least squares inverse filter in the sense of producing the smallest squared error.

So far we have only considered the problem of finding the least squares inverse filter $h_N(n)$ so that, for a given $g(n)$,

$$h_N(n) * g(n) \approx \delta(n - n_0)$$

However, this problem may be easily generalized to one of finding the least squares inverse filter $h_N(n)$ such that

$$h_N(n) * g(n) \approx d(n)$$

where $d(n)$ is an arbitrary sequence. In fact, note that since no assumptions were made about the form of $d(n)$ in the derivation leading up to Eq. (4.90), the solution to this generalized problem is

$$\mathbf{R}_g\mathbf{h}_N = \mathbf{r}_{dg}$$

where $\mathbf{r}_{dg}$ is a vector containing the cross-correlation $r_{dg}(k)$ as defined in Eq. (4.91). As in Eq. (4.102) for the special case of $d(n) = \delta(n - n_0)$, the FIR least squares inverse filter is the least squares solution to the set of overdetermined linear equations

$$\boxed{\mathbf{G}_0\mathbf{h}_N = \mathbf{d}} \tag{4.103}$$

where $\mathbf{d}$ is a vector containing the signal values $d(n)$. In Chapter 7 where we consider the problem of Wiener filtering, we will be looking at the stochastic version of this problem.

4.5 ITERATIVE PREFILTERING*

In Section 4.2 we saw how the direct method of performing a least squares minimization of the error

$$E'(z) = X(z) - \frac{B_q(z)}{A_p(z)} \tag{4.104}$$

led to a set of nonlinear equations in the coefficients $a_p(k)$ and $b_q(k)$. In this section we consider an iterative approach to minimizing Eq. (4.104) known as *iterative prefiltering*. Originally developed by Steiglitz and McBride for system identification and later applied to pole-zero modeling of speech [15, 16], this method is sometimes referred to as the Steiglitz-McBride method.

The basic idea of iterative prefiltering may be developed as an extension of Shanks' method as follows. Recall that in Shanks' method, once the coefficients of $A_p(z)$ have been found using Prony's method, the coefficients of $B_q(z)$ are determined by performing a least squares minimization of the error

$$E'(z) = X(z) - B_q(z)G(z) \tag{4.105}$$

where

$$G(z) = \frac{1}{A_p(z)}$$

Since $E'(z)$ is *linear* in the coefficients $b_q(k)$, this minimization is straightforward. Note, however, that if we rewrite Eq. (4.105) as

$$E'(z) = \Big[X(z)A_p(z) - B_q(z)\Big]G(z) \tag{4.106}$$

then, once $G(z)$ fixed, $E'(z)$ becomes *linear* in $A_p(z)$ as well as $B_q(z)$. Therefore, we may easily perform a least squares minimization of $E'(z)$ with respect to both $a_p(k)$ and $b_q(k)$, thereby refining the estimate of $a_p(k)$. Having found a new set of coefficients for $A_p(z)$, we may then update $G(z)$ and repeat the process, thus forming an iteration. Therefore, the iteration proceeds as follows. Let $B_q^{(i)}(z)$ and $A_p^{(i)}(z)$ be the estimates of the numerator and denominator of $H(z)$, respectively, at the ith iteration. With $A_p^{(i)}(z)$ fixed, new

estimates $B_q^{(i+1)}(z)$ and $A_p^{(i+1)}(z)$ are found by performing a least squares minimization of

$$E^{(i+1)}(z) = \frac{A_p^{(i+1)}(z)X(z) - B_q^{(i+1)}(z)}{A_p^{(i)}(z)} \tag{4.107}$$

Given the new estimate $A_p^{(i+1)}(z)$, the denominator of $E^{(i+1)}(z)$ is then updated and the process repeated.

In order to formalize this iteration, we begin by rewriting Eq. (4.107) as follows:

$$E^{(i+1)}(z) = \left[A_p^{(i+1)}(z)X(z) - B_q^{(i+1)}(z)\right]G^{(i)}(z) \tag{4.108}$$

where $G^{(i)}(z) = 1/A_p^{(i)}(z)$. If we define

$$F^{(i)}(z) = X(z)G^{(i)}(z)$$

which is the signal that is formed by filtering $x(n)$ with $g^{(i)}(n)$, then $E^{(i+1)}(z)$ becomes

$$E^{(i+1)}(z) = A_p^{(i+1)}(z)F^{(i)}(z) - B_q^{(i+1)}(z)G^{(i)}(z) \tag{4.109}$$

which, in the time domain, is

$$e^{(i+1)}(n) = a_p^{(i+1)}(n) * f^{(i)}(n) - b_q^{(i+1)}(n) * g^{(i)}(n) \tag{4.110}$$

Note that both sequences, $g^{(i)}(n)$ and $f^{(i)}(n)$, may be computed recursively as follows,

$$g^{(i)}(n) = \delta(n) - \sum_{k=1}^{p} a_p^{(i)}(k)g^{(i)}(n-k) \quad ; \quad n \geq 0$$

$$f^{(i)}(n) = x(n) - \sum_{k=1}^{p} a_p^{(i)}(k)f^{(i)}(n-k) \quad ; \quad n \geq 0$$

where $g^{(i)}(n) = f^{(i)}(n) = 0$ for $n < 0$. With $f^{(i)}(n)$ and $g^{(i)}(n)$ fixed, the coefficients $a_p^{(i+1)}(k)$ and $b_q^{(i+1)}(k)$ are found by minimizing the error

$$\mathcal{E}^{(i+1)} = \sum_{n=0}^{\infty} |e^{(i+1)}(n)|^2$$

As we have done before, a set of linear equations for the coefficients that minimize $\mathcal{E}^{(i+1)}$ may be derived by differentiating this error with respect to $[a_p^{(i+1)}(k)]^*$ and $[b_q^{(i+1)}(k)]^*$. Differentiating and setting the derivatives equal to zero results in the following *orthogonality conditions*:

$$\sum_{n=0}^{\infty} e^{(i+1)}(n)\left[f^{(i)}(n-k)\right]^* = 0 \quad ; \quad k = 1, 2, \ldots p$$

$$\sum_{n=0}^{\infty} e^{(i+1)}(n)\left[g^{(i)}(n-k)\right]^* = 0 \quad ; \quad k = 0, 1 \ldots q \tag{4.111}$$

Substituting Eq. (4.110) for $e^{(i+1)}(n)$ into Eq. (4.111) we obtain a set of linear equations that may be solved for $a_p^{(i+1)}(k)$ and $b_q^{(i+1)}(k)$. Instead of pursuing this approach, we will formulate the minimization of $\mathcal{E}^{(i+1)}$ as a problem of finding the least squares solution to a set of overdetermined linear equations. This may be done as follows. Note that in the iterative prefiltering method, we would like to find a set of coefficients, $a_p^{(i+1)}(k)$ and

$b_q^{(i+1)}(k)$, that force the error $e^{(i+1)}(n)$ to zero for $n \geq p$. From Eq. (4.110) this implies that

$$f^{(i)}(n) + \sum_{k=1}^{p} a_p^{(i+1)}(k) f^{(i)}(n-k) - \sum_{k=0}^{q} b_q^{(i+1)}(k) g^{(i)}(n-k) = 0 \quad ; \quad n \geq p \tag{4.112}$$

Writing these equations in matrix form we have

$$\boxed{\left[\begin{array}{c|c} \mathbf{F}^{(i)} & \mathbf{G}^{(i)} \end{array}\right] \left[\begin{array}{c} \bar{\mathbf{a}}_p^{(i+1)} \\ \hline -\mathbf{b}_q^{(i+1)} \end{array}\right] = -\mathbf{f}^{(i)}} \tag{4.113}$$

where

$$\mathbf{F}^{(i)} = \begin{bmatrix} f^{(i)}(p-1) & f^{(i)}(p-2) & \cdots & f^{(i)}(0) \\ f^{(i)}(p) & f^{(i)}(p-1) & \cdots & f^{(i)}(1) \\ f^{(i)}(p+1) & f^{(i)}(p) & \cdots & f^{(i)}(2) \\ \vdots & \vdots & & \vdots \end{bmatrix} \tag{4.114}$$

$$\mathbf{G}^{(i)} = \begin{bmatrix} g^{(i)}(p) & g^{(i)}(p-1) & \cdots & g^{(i)}(p-q) \\ g^{(i)}(p+1) & g^{(i)}(p) & \cdots & g^{(i)}(p-q+1) \\ g^{(i)}(p+2) & g^{(i)}(p+1) & \cdots & g^{(i)}(p-q+2) \\ \vdots & \vdots & & \vdots \end{bmatrix} \tag{4.115}$$

and

$$\mathbf{f}^{(i)} = \left[f^{(i)}(p),\ f^{(i)}(p+1),\ f^{(i)}(p+2)\cdots\right]^T \tag{4.116}$$

Since Eq. (4.113) is a set of overdetermined linear equations, we seek a least squares solution, which, as discussed in Section 2.3.6, is found by solving

$$\left[\begin{array}{c|c} \mathbf{F}^{(i)} & \mathbf{G}^{(i)} \end{array}\right]^H \left[\begin{array}{c|c} \mathbf{F}^{(i)} & \mathbf{G}^{(i)} \end{array}\right] \left[\begin{array}{c} \bar{\mathbf{a}}_p^{(i+1)} \\ \hline -\mathbf{b}_q^{(i+1)} \end{array}\right] = -\left[\begin{array}{c|c} \mathbf{F}^{(i)} & \mathbf{G}^{(i)} \end{array}\right]^H \mathbf{f}^{(i)} \tag{4.117}$$

Multiplying out the matrix product leads to

$$\left[\begin{array}{c|c} (\mathbf{F}^{(i)})^H \mathbf{F}^{(i)} & (\mathbf{F}^{(i)})^H \mathbf{G}^{(i)} \\ \hline (\mathbf{G}^{(i)})^H \mathbf{F}^{(i)} & (\mathbf{G}^{(i)})^H \mathbf{G}^{(i)} \end{array}\right] \left[\begin{array}{c} \bar{\mathbf{a}}_p^{(i+1)} \\ \hline -\mathbf{b}_q^{(i+1)} \end{array}\right] = -\left[\begin{array}{c|c} \mathbf{F}^{(i)} & \mathbf{G}^{(i)} \end{array}\right]^H \mathbf{f}^{(i)} \tag{4.118}$$

which is equivalent to the set of linear equations that would be derived by differentiation as discussed above. In fact, these equations have the form

$$\left[\begin{array}{c|c} \mathbf{R}_f^{(i)} & \mathbf{R}_{gf}^{(i)} \\ \hline \mathbf{R}_{fg}^{(i)} & \mathbf{R}_g^{(i)} \end{array}\right] \left[\begin{array}{c} \bar{\mathbf{a}}_p^{(i+1)} \\ \hline -\mathbf{b}_q^{(i+1)} \end{array}\right] = -\left[\begin{array}{c} \mathbf{r}_f^{(i)} \\ \hline \mathbf{r}_{fg}^{(i)} \end{array}\right] \tag{4.119}$$

where $\mathbf{R}_f^{(i)}$ and $\mathbf{R}_g^{(i)}$ contain the autocorrelations of $g^{(i)}(n)$ and $f^{(i)}(n)$, respectively, and $\mathbf{R}_{fg}^{(i)}$ is a vector of cross-correlations between $f^{(i)}(n)$ and $g^{(i)}(n)$. A MATLAB program for the iterative prefiltering method that is based on finding the least squares solution to Eq. (4.113) is given in Fig. 4.17.

Iterative Prefiltering

```
function [a,b,err] = ipf(x,p,q,n,a)
%
   x   = x(:);
   N   = length(x);
   if p+q>=length(x), error('Model order too large'), end
   if nargin < 5
      a   = prony(x,p,q);  end;
   delta = [1; zeros(N-1,1)];
   for i=1:n
      f   = filter(1,a,x);
      g   = filter(1,a,delta);
      u   = convm(f,p+1);
      v   = convm(g,q+1);
      ab  = -[u(1:N,2:p+1) -v(1:N,:) ]\u(1:N,1);
      a   = [1; ab(1:p)];
      b   = ab(p+1:p+q+1);
      err = norm( u(1:N,1) + [u(1:N,2:p+1) -v(1:N,:)]*ab);
      end;
```

Figure 4.17 *A* MATLAB *program for the Iterative Prefiltering (Steiglitz-McBride) method. Note that this program calls* `convm.m` *and* `prony.m` *(see Appendix).*

As with any iterative algorithm, an important issue concerns the conditions for which this iteration converges. Although no general conditions have been given to guarantee convergence, this technique has been observed to either converge or to reach an acceptable solution within five to ten iterations [6]. Since the rate of convergence will depend, in general, on how close the initial guess is to the optimum solution, it is important to initialize the iteration with a set of coefficients $a_p^{(0)}(k)$ that is as accurate as possible using, for example, Prony's method.

4.6 FINITE DATA RECORDS

In Section 4.4, we considered the problem of modeling a signal $x(n)$ using Prony's method. Since Prony's method assumes that $x(n)$ is known for all n, the coefficients $a_p(k)$ are found by minimizing the sum of the squares of $e(n)$ for all $n > q$. What happens, however, when $x(n)$ is only known or measured for values of n over a finite interval such as $[0, N]$? In this case, since $e(n)$ cannot be evaluated for $n > N$, it is not possible to minimize the sum of the squares of $e(n)$ over any interval that includes values of $n > N$, unless some assumptions are made about the values of $x(n)$ outside the interval $[0, N]$. A similar problem arises in applications for which we are only interested in modeling a signal over a finite interval. Suppose, for example, that the properties of the signal that we wish to model are slowly varying in time. It may be appropriate, in this case, to consider segmenting the signal into nonoverlapping subsequences, $x_i(n)$, and to find a model for each subsequence. Again, finding the denominator coefficients by minimizing the sum of the squares of the error for all $n > q$ would not be appropriate. In this section we look at the problem of signal modeling when only a finite data record is available. We consider two approaches: the *autocorrelation method* and the *covariance method.* In the autocorrelation method, the data

outside the interval $[0, N]$ is set to zero by applying a data window to $x(n)$ and then Prony's method is used to find a model for the windowed signal. With the covariance method, on the other hand, instead of using a data window, the denominator coefficients are found by minimizing an error that does not depend upon the values of $x(n)$ outside the interval $[0, N]$. As we will soon see, there are advantages and disadvantages with each method. Since the autocorrelation and covariance methods are typically used for all-pole signal modeling, our development will be in terms of all-pole models. At the end of the section we will indicate how these methods may be used in the context of a more general signal model.

4.6.1 The Autocorrelation Method

Suppose that a signal $x(n)$, possibly complex, is only known over the finite interval $[0, N]$ and that $x(n)$ is to be approximated using an all-pole model of the form given in Eq. (4.71). Using Prony's method to find an all-pole model for $x(n)$, the coefficients $a_p(k)$ are found that minimize the error

$$\mathcal{E}_p = \sum_{n=0}^{\infty} |e(n)|^2$$

where

$$e(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \tag{4.120}$$

However, if $x(n)$ is unknown or unspecified outside the interval $[0, N]$, then $e(n)$ cannot be evaluated for $n < p$ or for $n > N$. Therefore, it is not possible to minimize $\mathcal{E}_p$ without modifying Prony's method or making some assumptions about the values of $x(n)$ outside the interval $[0, N]$. One way around this difficulty is to assume that $x(n) = 0$ for $n < 0$ and for $n > N$ and then use Prony's method to find the coefficients $a_p(k)$. This approach, known as the *autocorrelation method*, effectively forms a new signal, $\tilde{x}(n)$, by applying a rectangular window to $x(n)$, i.e.,

$$\tilde{x}(n) = x(n)w_R(n)$$

where

$$w_R(n) = \begin{cases} 1 & ; \quad n = 0, 1, \ldots, N \\ 0 & ; \quad \text{otherwise} \end{cases}$$

and then uses Prony's method to find an all-pole model for $\tilde{x}(n)$. The *normal equations* for the coefficients $a_p(k)$ that minimize $\mathcal{E}_p$ are the same as those given in Eq. (4.79) for Prony's method except that $r_x(k)$ is computed using $\tilde{x}(n)$ instead of $x(n)$,

$$r_x(k) = \sum_{n=0}^{\infty} \tilde{x}(n)\tilde{x}^*(n-k) = \sum_{n=k}^{N} x(n)x^*(n-k) \quad ; \quad k = 0, 1, \ldots, p \tag{4.121}$$

Since the autocorrelation method simply involves the application of Prony's method to a windowed signal, the Toeplitz structure of the normal equations is preserved. Therefore, the Levinson-Durbin recursion presented in Chapter 5 may be used to solve the *autocorrelation normal equations*. The equations for all-pole signal modeling using the autocorrelation method are summarized in Table 4.5.

Table 4.5 ***All-Pole Modeling Using the Autocorrelation Method***

Normal equations

$$\sum_{l=1}^{p} a_p(l) r_x(k-l) = -r_x(k) \quad ; \quad k = 1, 2, \ldots, p$$

$$r_x(k) = \sum_{n=k}^{N} x(n) x^*(n-k) \quad ; \quad k \geq 0$$

Minimum error

$$\epsilon_p = r_x(0) + \sum_{k=1}^{p} a_p(k) r_x^*(k)$$

Example 4.6.1 *The Autocorrelation Method*

Consider the signal $x(n)$ whose first $N+1$ values are

$$\mathbf{x} = [1, \ \beta, \ \beta^2, \ldots, \beta^N]^T$$

where β is an arbitrary complex number. Assuming that $x(n)$ is either unknown or unspecified outside the interval $[0, N]$, let us use the autocorrelation method to find a first-order all-pole model of the form

$$H(z) = \frac{b(0)}{1 + a(1) z^{-1}}$$

With $p = 1$, it follows from the all-pole normal equations, Eq. (4.79), that

$$a(1) = -r_x(1)/r_x(0)$$

Since

$$r_x(k) = \sum_{n=k}^{N} x(n) x^*(n-k) = \sum_{n=k}^{N} \beta^n (\beta^*)^{n-k} = \beta^k \sum_{n=0}^{N-k} |\beta|^{2n} = \beta^k \frac{1 - |\beta|^{2(N-k+1)}}{1 - |\beta|^2}$$

and $r_x(k) = 0$ for $k > N$, then

$$r_x(1) = \beta \frac{1 - |\beta|^{2N}}{1 - |\beta|^2} \qquad \text{and} \qquad r_x(0) = \frac{1 - |\beta|^{2N+2}}{1 - |\beta|^2}$$

Therefore, for $a(1)$ we have

$$a(1) = -\beta \frac{1 - |\beta|^{2N}}{1 - |\beta|^{2N+2}}$$

Note that if $|\beta| < 1$, then $|a(1)| < 1$ and the model is stable. However, if β is replaced with $1/\beta^*$, then we have exactly the same model. Therefore, it follows that $|a(1)| < 1$ and the model is stable for any value of β. Furthermore, note that if $|\beta| < 1$, then

$$\lim_{N \to \infty} a(1) = -\beta$$

If, on the other hand, $|\beta| > 1$, then the model approaches the minimum phase solution,

$$\lim_{N\to\infty} a(1) = -1/\beta$$

Finally, after a bit of algebra, it follows that the model error is

$$\epsilon_1 = r_x(0) + a(1)r_x^*(1) = \frac{1 - |\beta|^{4N+2}}{1 - |\beta|^{2N+2}}$$

Note that if $|\beta| < 1$, then the error approaches one as $N \to \infty$. This is due to the term $e(0)$ that is included in the error ϵ_1. Thus, although $e(n) = 0$ for $n > 0$, since $e(0) = x(0)$, then the minimum value for the squared error is $\epsilon_1 = 1$.

Applying a window to $x(n)$ in the autocorrelation method forces $x(n)$ to zero for $n < 0$ and $n > N$, even if $x(n)$ is nonzero outside the interval $[0, N]$. As a result, the accuracy of the model is compromised. For example, although the signal $x(n) = \beta^n u(n)$ is the unit sample response of the all-pole filter

$$H(z) = \frac{1}{1 - \beta z^{-1}}$$

as we saw in the previous example, setting $x(n) = 0$ for $n > N$ modifies the signal and biases the solution found using the autocorrelation method. On the other hand, it may be shown that windowing the data has the advantage of ensuring that the model will be stable, i.e., the poles of $H(z)$ will be inside the unit circle. This property may be of importance in applications in which the model is used to synthesize or extrapolate a signal over long periods of time since an unstable model will result in the signal becoming unbounded. Although many different proofs of this property have been given, we will postpone the proof of this very important result until Chapter 5, after we have developed the Levinson-Durbin recursion and a few of the properties of this recursion.

As we did with Prony's method, the autocorrelation method may also be expressed in terms of finding the least squares solution to a set of overdetermined linear equations. Specifically, note that in the autocorrelation method we would like to find a set of coefficients $a_p(k)$ so that the error, $e(n)$, is equal to zero for $n > 0$, i.e.,

$$e(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) = 0 \quad ; \quad n > 0 \tag{4.122}$$

In matrix form, Eq. (4.122) may be expressed as

$$\boxed{\mathbf{X}_p\bar{\mathbf{a}}_p = -\mathbf{x}_1} \tag{4.123}$$

where

$$\mathbf{X}_p = \left[\begin{array}{ccccc} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \hline x(p-1) & x(p-2) & x(p-3) & \cdots & x(0) \\ x(p) & x(p-1) & x(p-2) & \cdots & x(1) \\ \vdots & \vdots & \vdots & & \vdots \\ x(N-2) & x(N-3) & x(N-4) & \cdots & x(N-p-1) \\ x(N-1) & x(N-2) & x(N-3) & \cdots & x(N-p) \\ \hline x(N) & x(N-1) & x(N-2) & \cdots & x(N-p+1) \\ 0 & x(N) & x(N-1) & \cdots & x(N-p+2) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & x(N) \end{array}\right] \tag{4.124}$$

is an $(N + p) \times p$ matrix and

$$\mathbf{x}_1 = [x(1),\ x(2), \ldots, x(N),\ 0, \ldots, 0]^T \tag{4.125}$$

is a vector of length $(N + p)$. The least squares solution is thus determined by solving the linear equations

$$(\mathbf{X}_p^H \mathbf{X}_p)\bar{\mathbf{a}}_p = \mathbf{X}_p \mathbf{x}_1$$

A MATLAB program for all-pole signal modeling using the autocorrelation method is given in Fig. 4.18.

Although typically associated with all-pole modeling, the approach used in the autocorrelation method may also be used for pole-zero modeling. The procedure simply requires solving the Prony normal equations, Eq. (4.34), with the autocorrelation $r_x(k, l)$ computed as in Eq. (4.33) using windowed data. Unlike the all-pole case, however, if $q > 0$, then the normal equations will no longer be Toeplitz.

The Autocorrelation Method

```
function [a,err] = acm(x,p)
%
   x   = x(:);
   N   = length(x);
   if p>=length(x), error('Model order too large'), end
      X   = convm(x,p+1)
      Xq  = X(1:N+p-1,1:p)
      a   = [1;-Xq\X(2:N+p,1)];
      err = abs(X(1:N+p,1)'*X*a);
      end;
```

Figure 4.18 *A* MATLAB *program for finding the coefficients of an all-pole model for a signal* $x(n)$ *using the autocorrelation method. Note that this program calls* `convm.m` *(see Appendix).*

Finally, it should be pointed out that windows other than a rectangular window may be applied to $x(n)$. The reason for considering a different window would be to minimize the edge effects in the autocorrelation method. Specifically, note that for $n = 0, 1, \ldots, p-1$ the prediction of $x(n)$ is based on fewer than p values of $x(n)$, i.e.,

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

where

$$\hat{x}(n) = -\sum_{k=1}^{n} a_p(k)x(n-k) \quad ; \quad n < p$$

Therefore, $e(n)$ will typically be larger for $n < p$ than it would be if the prediction were based on p values. The same will be true for $n > N - p$ since again the prediction is based on fewer than p values. As a result, $\mathcal{E}_p$ may be disproportionately influenced by the large errors at the edge of the window. To reduce the contribution of these error to $\mathcal{E}_p$, a window with a taper is often used. This is particularly true and important, for example, in all-pole modeling of speech [4, 11].

4.6.2 The Covariance Method

As we saw in the previous section, all-pole modeling with the autocorrelation method forces the signal to zero outside the interval $[0, N]$ by applying a window to $x(n)$. This may not always be the best approach to take, however, particularly in light of the fact that an all-pole model implies that the signal is infinite in length. In this section we consider another method for all-pole signal modeling, known as the *covariance method*, that does not make any assumptions about the data outside the given observation interval and does not apply a window to the data. As a result, the model obtained with the covariance method is typically more accurate than the autocorrelation method. With the covariance method, however, we lose the Toeplitz structure of the normal equations along with the guaranteed stability of the all-pole model.

To develop the covariance method, recall that all-pole modeling using Prony's method requires the minimization of $\mathcal{E}_p$ in Eq. (4.73), which is a sum of the squares of the error $e(n)$. However, as we see from the definition of $e(n)$ given in Eq. (4.120), in order to evaluate $e(n)$ at time n, it is necessary that the values of $x(n)$ be known at times $n, n-1, \ldots, n-p$. If we assume that $x(n)$ is only specified for $n = 0, 1, \ldots, N$ then, as illustrated in Fig. 4.19, the error may only be evaluated for values of n in the interval $[p, N]$. Therefore, suppose that we modify the error $\mathcal{E}_p$ so that the sum includes only those values of $e(n)$ that may be computed from the given data, i.e.,

$$\boxed{\mathcal{E}_p^C = \sum_{n=p}^{N} |e(n)|^2} \tag{4.126}$$

Minimizing $\mathcal{E}_p^C$ avoids having to make any assumptions about the data outside the interval $[0, N]$ and does not require that the data be windowed. Finding the all-pole model that minimizes $\mathcal{E}_p^C$ is known as the *covariance method*. Since the only difference between the covariance method and Prony's method is in the limits on the sum for the error, the *covariance normal equations* are identical to those in Prony's method,

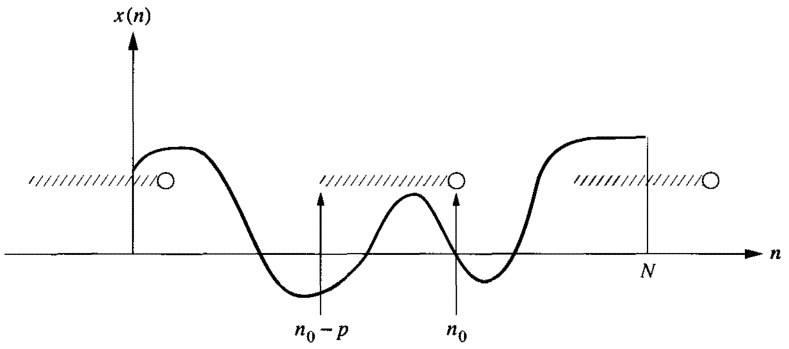


Figure 4.19 *Evaluation of the error,* $e(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k)$. *If* $x(n)$ *is only known over the interval* $[0, N]$, *then* $e(n)$ *may only be evaluated from* $n = p$ *to* $n = N$. *To evaluate the error for* $n < p$ *it is necessary that* $x(n)$ *be known for* $n < 0$, *whereas for* $n > N$ *it is necessary that* $x(n)$ *be known for* $n > N$.

$$\begin{bmatrix} r_x(1,1) & r_x(1,2) & r_x(1,3) & \cdots & r_x(1,p) \\ r_x(2,1) & r_x(2,2) & r_x(2,3) & \cdots & r_x(2,p) \\ r_x(3,1) & r_x(3,2) & r_x(3,3) & \cdots & r_x(3,p) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p,1) & r_x(p,2) & r_x(p,3) & \cdots & r_x(p,p) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} r_x(1,0) \\ r_x(2,0) \\ r_x(3,0) \\ \vdots \\ r_x(p,0) \end{bmatrix} \tag{4.127}$$

except that the computation of the autocorrelation sequence $r_x(k,l)$ is modified as follows:

$$\boxed{r_x(k,l) = \sum_{n=p}^{N} x(n-l)x^*(n-k)} \tag{4.128}$$

Unlike the autocorrelation normal equations, the covariance normal equations are not Toeplitz. This follows by noting that

$$\begin{aligned} r_x(k+1,l+1) &= \sum_{n=p}^{N} x(n-[l+1])x^*(n-[k+1]) = \sum_{m=p-1}^{N-1} x(m-l)x^*(m-k) \\ &= r_x(k,l) - x(N-l)x^*(N-k) + x(p-1-l)x^*(p-1-k) \end{aligned} \tag{4.129}$$

Therefore, the Levinson-Durbin recursion cannot be used to solve Eq. (4.127). Although there is a fast covariance algorithm that is more efficient than Gaussian elimination, it is more complicated and computationally less efficient than the Levinson-Durbin recursion [6, 7]. Finally, the covariance modeling error is

$$\{\mathcal{E}_p^C\}_{\min} = r_x(0,0) + \sum_{k=1}^{p} a_p(k)r_x(0,k)$$

The covariance method is summarized in Table 4.6.

As with the autocorrelation method, the covariance method may be formulated as a least squares approximation problem. Specifically, note that if we set the error in Eq. (4.122) equal

Table 4.6 *All-Pole Modeling Using the Covariance Method*

Normal equations

$$\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0) \quad ; \quad k = 1, 2, \ldots, p$$

$$r_x(k,l) = \sum_{n=p}^{N} x(n-l)x^*(n-k) \quad ; \quad k, l \geq 0$$

Minimum error

$$\left\{\mathcal{E}_p^c\right\}_{\min} = r_x(0,0) + \sum_{k=1}^{p} a_p(k) r_x(0,k)$$

The Covariance Method

```
function [a,err] = covm(x,p)
%
   x   = x(:);
   N   = length(x);
   if p>=length(x), error('Model order too large'), end
      X   = convm(x,p+1);
      Xq  = X(p:N-1,1:p);
      a   = [1;-Xq\X(p+1:N,1)];
      err = abs(X(p+1:N,1)'*X(p+1:N,:)*a);
      end;
```

Figure 4.20 *A* MATLAB *program for finding the coefficients of an all-pole model for a signal* $x(n)$ *using the covariance method. Note that this program calls* `convm.m` *(see Appendix).*

to zero for n in the interval $[p, N]$ we have a set of $(N - p + 1)$ linear equations of the form

$$\begin{bmatrix} x(p-1) & x(p-2) & \cdots & x(0) \\ x(p) & x(p-1) & \cdots & x(1) \\ \vdots & \vdots & & \vdots \\ x(N-2) & x(N-3) & \cdots & x(N-p-1) \\ x(N-1) & x(N-2) & \cdots & x(N-p) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} x(p) \\ x(p+1) \\ \vdots \\ x(N-1) \\ x(N) \end{bmatrix} \tag{4.130}$$

Note that these equations correspond to the subset of $N - p + 1$ equations in Eq. (4.123) that do not require information about $x(n)$ outside the range $[0, N]$. Since these equations are overdetermined, we may use the same approach described for the autocorrelation method and find the least squares best approximation. A MATLAB program for finding this least squares solution is given in Fig. 4.20.

Example 4.6.2 *The Covariance Method*

Let us again consider the problem of finding a first-order all-pole model for the signal $x(n)$ whose first $N + 1$ values are

$$\mathbf{x} = \left[1, \beta, \beta^2, \ldots, \beta^N\right]^T$$

where β is an arbitrary complex number. With

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1}}$$

the value for $a(1)$ using the covariance method is

$$a(1) = -r_x(1,0)/r_x(1,1)$$

where

$$r_x(k,l) = \sum_{n=1}^{N} x(n-l)x^*(n-k)$$

Since

$$r_x(1,1) = \sum_{n=1}^{N} x(n-1)x^*(n-1) = \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1-|\beta|^{2N}}{1-|\beta|^2}$$

$$r_x(1,0) = \sum_{n=1}^{N} x(n)x^*(n-1) = \beta\frac{1-|\beta|^{2N}}{1-|\beta|^2}$$

then $a(1) = -\beta$. With $b(0) = 1$ so that $x(0) = \hat{x}(0)$ the model becomes

$$H(z) = \frac{1}{1-\beta z^{-1}}$$

Note that the model will be stable if and only if $|\beta| < 1$.

In the previous example, the covariance method was applied to the first $N+1$ values of the signal $x(n) = \beta^n u(n)$, which is the unit sample response of the all-pole filter

$$H(z) = \frac{1}{1-\beta z^{-1}}$$

Unlike the autocorrelation method, the covariance method was able to find the pole location exactly. This will be true, in general, when modeling a signal having an all-pole z-transform

$$X(z) = \frac{b(0)}{1 + \sum_{k=1}^{p} a(k)z^{-k}}$$

provided a sufficient number of values of $x(n)$ are given. In the next example, we look at the models that are produced when the autocorrelation and covariance methods are applied to the signal $x(n) = (-1)^n$ for $n = 0, 1, \ldots, N$.

Example 4.6.3 ***Comparison of the Autocorrelation and Covariance Methods***

Suppose that a signal $x(n)$ is to be modeled as the unit sample response of a second-order all-pole filter,

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1} + a(2)z^{-2}}$$

The first 20 values of this signal are measured and found to be

$$\mathbf{x} = [1, -1, 1, -1, \ldots, 1, -1]$$

To solve for the coefficients $a(k)$ using the autocorrelation method, Prony's method is applied to the windowed sequence

$$\tilde{x}(n) = \begin{cases} x(n) & ; \quad n = 0, 1, \ldots, N \\ 0 & ; \quad \text{otherwise} \end{cases}$$

The normal equations are

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \end{bmatrix}$$

where

$$r_x(k) = \sum_{n=0}^{\infty} \tilde{x}(n)\tilde{x}(n-k) = \sum_{n=k}^{N} x(n)x(n-k)$$

Evaluating this sum for $k = 0$, 1, and 2, we find

$$r_x(0) = 20 \qquad r_x(1) = -19 \qquad r_x(2) = 18$$

Substituting these values into the normal equations, we have

$$\begin{bmatrix} 20 & -19 \\ -19 & 20 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = - \begin{bmatrix} -19 \\ 18 \end{bmatrix}$$

Solving for $a(1)$ and $a(2)$, we find

$$a(1) = 0.9744 \quad ; \quad a(2) = 0.0256$$

Therefore, the denominator polynomial is

$$A(z) = 1 + 0.9744z^{-1} + 0.0256z^{-2}$$

From Eq. (4.80) we see that the modeling error is

$$\epsilon_p = r_x(0) + a(1)r_x(1) + a(2)r_x(2) = 1.9487$$

Setting $b(0) = \sqrt{\epsilon_1}$ to satisfy the energy matching constraint the model becomes

$$H(z) = \frac{1.3960}{1 + 0.9744z^{-1} + 0.0256z^{-2}}$$

Suppose, now, that the covariance method is used to find the all-pole model parameters. In this case, the normal equations take the form

$$\begin{bmatrix} r_x(1,1) & r_x(1,2) \\ r_x(2,1) & r_x(2,2) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = - \begin{bmatrix} r_x(1,0) \\ r_x(2,0) \end{bmatrix} \tag{4.131}$$

where the autocorrelations are

$$r_x(k,l) = \sum_{n=p}^{N} x(n-l)x(n-k) = \sum_{n=2}^{19} x(n-l)x(n-k)$$

Evaluating the sum, we find

$$r_x(1,1) = r_x(2,2) = r_x(2,0) = 18 \qquad r_x(1,2) = r_x(2,1) = r_x(1,0) = -18$$

which, when substituted into the normal equations, gives

$$\begin{bmatrix} 18 & -18 \\ -18 & 18 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = - \begin{bmatrix} -18 \\ 18 \end{bmatrix}$$