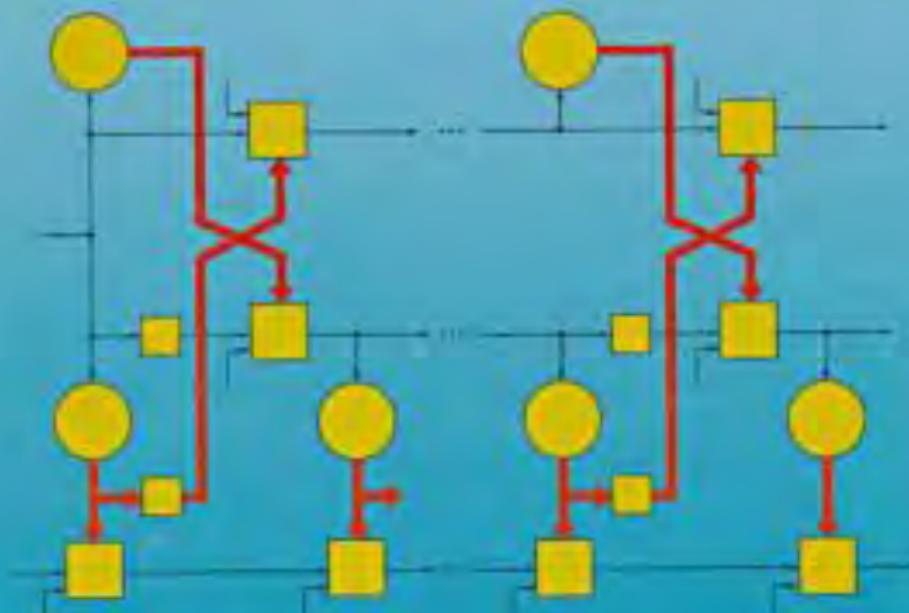


ADAPTIVE FILTER THEORY

THIRD EDITION



SIMON HAYKIN

WILEY-INTERSCIENCE INFORMATION AND SYSTEM SCIENCES SERIES

Advanced Research Computer Science

Contents

Preface xiii

Acknowledgments xvi

Introduction 1

1. The Filtering Problem 1
2. Adaptive Filters 2
3. Linear Filter Structures 4
4. Approaches to the Development of Linear Adaptive Filtering Algorithms 9
5. Real and Complex Forms of Adaptive Filters 14
6. Nonlinear Adaptive Filters 15
7. Applications 18
8. Some Historical Notes 67

PART 1 BACKGROUND MATERIAL 78

Chapter 1 Discrete-Time Signal Processing 79

- 1.1 z-Transform 79
- 1.2 Linear Time-Invariant Filters 81
- 1.3 Minimum-Phase Filters 86
- 1.4 Discrete Fourier Transform 87
- 1.5 Implementing Convolutions Using the DFT 87
- 1.6 Discrete Cosine Transform 93
- 1.7 Summary and Discussion 94
- Problems 95

Chapter 2 Stationary Processes and Models 96

- 2.1 Partial Characterization of a Discrete-Time Stochastic Process 97

2.2	Mean Ergodic Theorem	98
2.3	Correlation Matrix	100
2.4	Correlation Matrix of Sine Wave Plus Noise	106
2.5	Stochastic Models	108
2.6	Wold Decomposition	115
2.7	Asymptotic Stationarity of an Autoregressive Process	116
2.8	Yule-Walker Equations	118
2.9	Computer Experiment: Autoregressive Process of Order 2	120
2.10	Selecting the Model Order	128
2.11	Complex Gaussian Processes	130
2.12	Summary and Discussion	132
	Problems	133

Chapter 3 Spectrum Analysis 136

3.1	Power Spectral Density	136
3.2	Properties of Power Spectral Density	138
3.3	Transmission of a Stationary Process Through a Linear Filter	140
3.4	Cramér Spectral Representation for a Stationary Process	144
3.5	Power Spectrum Estimation	146
3.6	Other Statistical Characteristics of a Stochastic Process	149
3.7	Polyspectra	150
3.8	Spectral-Correlation Density	154
3.9	Summary and Discussion	157
	Problems	158

Chapter 4 Eigenanalysis 160

4.1	The Eigenvalue Problem	160
4.2	Properties of Eigenvalues and Eigenvectors	162
4.3	Low-Rank Modeling	176
4.4	Eigenfilters	181
4.5	Eigenvalue Computations	184
4.6	Summary and Discussion	187
	Problems	188

PART 2 LINEAR OPTIMUM FILTERING 193

Chapter 5 Wiener Filters 194

5.1	Linear Optimum Filtering: Problem Statement	194
5.2	Principle of Orthogonality	197
5.3	Minimum Mean-Squared Error	201
5.4	Wiener-Hopf Equations	203
5.5	Error-Performance Surface	206
5.6	Numerical Example	210
5.7	Channel Equalization	217
5.8	Linearly Constrained Minimum Variance Filter	220
5.9	Generalized Sidelobe Cancelers	227
5.10	Summary and Discussion	235
	Problems	236

Chapter 6 Linear Prediction 241

- 6.1 Forward Linear Prediction 242
- 6.2 Backward Linear Prediction 248
- 6.3 Levinson-Durbin Algorithm 254
- 6.4 Properties of Prediction-Error Filters 262
- 6.5 Schur-Cohn Test 271
- 6.6 Autoregressive Modeling of a Stationary Stochastic Process 273
- 6.7 Cholesky Factorization 276
- 6.8 Lattice Predictors 280
- 6.9 Joint-Process Estimation 286
- 6.10 Block Estimation 290
- 6.11 Summary and Discussion 293
- Problems 295

Chapter 7 Kalman Filters 302

- 7.1 Recursive Minimum Mean-Square Estimation for Scalar Random Variables 303
- 7.2 Statement of the Kalman Filtering Problem 306
- 7.3 The Innovations Process 307
- 7.4 Estimation of the State using the Innovations Process 310
- 7.5 Filtering 317
- 7.6 Initial Conditions 320
- 7.7 Summary of the Kalman Filter 320
- 7.8 Variants of the Kalman Filter 322
- 7.9 The Extended Kalman Filter 328
- 7.10 Summary and Discussion 333
- Problems 334

PART 3 LINEAR ADAPTIVE FILTERING 338**Chapter 8 Method of Steepest Descent 339**

- 8.1 Some Preliminaries 339
- 8.2 Steepest-Descent Algorithm 341
- 8.3 Stability of the Steepest-Descent Algorithm 343
- 8.4 Example 350
- 8.5 Summary and Discussion 362
- Problems 362

Chapter 9 Least-Mean-Square Algorithm 365

- 9.1 Overview of the Structure and Operation of the Least-Mean-Square Algorithm 365
- 9.2 Least-Mean-Square Adaptation Algorithm 367
- 9.3 Examples 372
- 9.4 Stability and Performance Analysis of the LMS Algorithm 390
- 9.5 Summary of the LMS Algorithm 405
- 9.6 Computer Experiment on Adaptive Prediction 406
- 9.7 Computer Experiment on Adaptive Equalization 412
- 9.8 Computer Experiment on Minimum-Variance Distortionless Response Beamformer 421
- 9.9 Directionality of Convergence of the LMS Algorithm for Non-White Inputs 425
- 9.10 Robustness of the LMS Algorithm 427

- 9.11 Normalized LMS Algorithm 432
- 9.12 Summary and Discussion 438
- Problems 439

Chapter 10 Frequency-Domain Adaptive Filters 445

- 10.1 Block Adaptive Filters 446
- 10.2 Fast LMS Algorithm 451
- 10.3 Unconstrained Frequency-Domain Adaptive Filtering 457
- 10.4 Self-Orthogonalizing Adaptive Filters 458
- 10.5 Computer Experiment on Adaptive Equalization 469
- 10.6 Classification of Adaptive Filtering Algorithms 477
- 10.7 Summary and Discussion 478
- Problems 479

Chapter 11 Method of Least Squares 483

- 11.1 Statement of the Linear Least-Squares Estimation Problem 483
- 11.2 Data Windowing 486
- 11.3 Principle of Orthogonality (Revisited) 487
- 11.4 Minimum Sum of Error Squares 491
- 11.5 Normal Equations and Linear Least-Squares Filters 492
- 11.6 Time-Averaged Correlation Matrix 495
- 11.7 Reformulation of the Normal Equations in Terms of Data Matrices 497
- 11.8 Properties of Least-Squares Estimates 502
- 11.9 Parametric Spectrum Estimation 506
- 11.10 Singular Value Decomposition 516
- 11.11 Pseudoinverse 524
- 11.12 Interpretation of Singular Values and Singular Vectors 525
- 11.13 Minimum Norm Solution to the Linear Least-Squares Problem 526
- 11.14 Normalized LMS Algorithm Viewed as the Minimum-Norm Solution to an Underdetermined Least-Squares Estimation Problem 530
- 11.15 Summary and Discussion 532
- Problems 533

Chapter 12 Rotations and Reflections 536

- 12.1 Plane Rotations 537
- 12.2 Two-Sided Jacobi Algorithm 538
- 12.3 Cyclic Jacobi Algorithm 544
- 12.4 Householder Transformation 548
- 12.5 The QR Algorithm 551
- 12.6 Summary and Discussion 558
- Problems 560

Chapter 13 Recursive Least-Squares Algorithm 562

- 13.1 Some Preliminaries 563
- 13.2 The Matrix Inversion Lemma 565
- 13.3 The Exponentially Weighted Recursive Least-Squares Algorithm 566
- 13.4 Update Recursion for the Sum of Weighted Error Squares 571
- 13.5 Example: Single-Weight Adaptive Noise Canceler 572

- 13.6 Convergence Analysis of the RLS Algorithm 573
- 13.7 Computer Experiment on Adaptive Equalization 580
- 13.8 State-Space Formulation of the RLS Problem 583
- 13.9 Summary and Discussion 587
- Problems 587

Chapter 14 Square-Root Adaptive Filters 589

- 14.1 Square-Root Kalman Filters 589
- 14.2 Building Square-Root Adaptive Filtering Algorithms on their Kalman Filter Counterparts 597
- 14.3 QR-RLS Algorithm 598
- 14.4 Extended QR-RLS Algorithm 614
- 14.5 Adaptive Beamforming 617
- 14.6 Inverse QR-RLS Algorithm 624
- 14.7 Summary and Discussion 627
- Problems 628

Chapter 15 Order-Recursive Adaptive Filters 630

- 15.1 Adaptive Forward Linear Prediction 631
- 15.2 Adaptive Backward Linear Prediction 634
- 15.3 Conversion Factor 636
- 15.4 Least-Squares Lattice Predictor 640
- 15.5 Angle-Normalized Estimation Errors 653
- 15.6 First-Order State-Space Models for Lattice Filtering 655
- 15.7 QR-Decomposition-Based Least-Squares Lattice Filters 660
- 15.8 Fundamental Properties of the QRD-LSL Filter 667
- 15.9 Computer Experiment on Adaptive Equalization 672
- 15.10 Extended QRD-LSL Algorithm 677
- 15.11 Recursive Least-Squares Lattice Filters Using *A Posteriori* Estimation Errors 679
- 15.12 Recursive LSL Filters Using *A Priori* Estimation Errors with Error Feedback 683
- 15.13 Computation of the Least-Squares Weight Vector 686
- 15.14 Computer Experiment on Adaptive Prediction 691
- 15.15 Other Variants of Least-Squares Lattice Filters 693
- 15.16 Summary and Discussion 694
- Problems 696

Chapter 16 Tracking of Time-Varying Systems 701

- 16.1 Markov Model for System Identification 702
- 16.2 Degree of Nonstationarity 705
- 16.3 Criteria for Tracking Assessment 706
- 16.4 Tracking Performance of the LMS Algorithm 708
- 16.5 Tracking Performance of the RLS Algorithm 711
- 16.6 Comparison of the Tracking Performance of LMS and RLS Algorithms 716
- 16.7 Adaptive Recovery of a Chirped Sinusoid in Noise 719
- 16.8 How to Improve the Tracking Behavior of the RLS Algorithm 726
- 16.9 Computer Experiment on System Identification 729
- 16.10 Automatic Tuning of Adaptation Constants 731
- 16.11 Summary and Discussion 736
- Problems 737

Chapter 17 Finite-Precision Effects 738

- 17.1 Quantization Errors 739
- 17.2 Least-Mean-Square Algorithm 741
- 17.3 Recursive Least-Squares Algorithm 751
- 17.4 Square-Root Adaptive Filters 757
- 17.5 Order-Recursive Adaptive Filters 760
- 17.6 Fast Transversal Filters 763
- 17.7 Summary and Discussion 767
- Problems 769

PART 4 NONLINEAR ADAPTIVE FILTERING 771**Chapter 18 Blind Deconvolution 772**

- 18.1 Theoretical and Practical Considerations 773
- 18.2 Bussgang Algorithm for Blind Equalization of Real Baseband Channels 776
- 18.3 Extension of Bussgang Algorithms to Complex Baseband Channels 791
- 18.4 Special Cases of the Bussgang Algorithm 792
- 18.5 Blind Channel Identification and Equalization Using Polyspectra 796
- 18.6 Advantages and Disadvantages of HOS-Based Deconvolution Algorithms 802
- 18.7 Channel Identifiability Using Cyclostationary Statistics 803
- 18.8 Subspace Decomposition for Fractionally-Spaced Blind Identification 804
- 18.9 Summary and Discussion 813
- Problems 814

Chapter 19 Back-Propagation Learning 817

- 19.1 Models of a Neuron 818
- 19.2 Multilayer Perceptron 822
- 19.3 Complex Back-Propagation Algorithm 824
- 19.4 Back-Propagation Algorithm for Real Parameters 837
- 19.5 Universal Approximation Theorem 838
- 19.6 Network Complexity 840
- 19.7 Filtering Applications 842
- 19.8 Summary and Discussion 852
- Problems 854

Chapter 20 Radial Basis Function Networks 855

- 20.1 Structure of RBF Networks 856
- 20.2 Radial-Basis Functions 858
- 20.3 Fixed Centers Selected at Random 859
- 20.4 Recursive Hybrid Learning Procedure 862
- 20.5 Stochastic Gradient Approach 863
- 20.6 Universal Approximation Theorem (Revisited) 865
- 20.7 Filtering Applications 866
- 20.8 Summary and Discussion 871
- Problems 873

Appendix A Complex Variables 875**Appendix B Differentiation with Respect to a Vector 890****Appendix C Method of Lagrange Multipliers 895**

Appendix D Estimation Theory	899
Appendix E Maximum-Entropy Method	905
Appendix F Minimum-Variance Distortionless Response Spectrum	912
Appendix G Gradient Adaptive Lattice Algorithm	915
Appendix H Solution of the Difference Equation (9.75)	919
Appendix I Steady-State Analysis of the LMS Algorithm without Invoking the Independence Assumption	921
Appendix J The Complex Wishart Distribution	924
Glossary	928
Abbreviations	932
Principal Symbols	933
Bibliography	941
Index	978

Introduction

1. THE FILTERING PROBLEM

The term *filter* is often used to describe a device in the form of a piece of physical hardware or software that is applied to a set of noisy data in order to extract information about a prescribed quantity of interest. The noise may arise from a variety of sources. For example, the data may have been derived by means of noisy sensors or may represent a useful signal component that has been corrupted by transmission through a communication channel. In any event, we may use a filter to perform three basic information-processing tasks:

1. *Filtering*, which means the extraction of information about a quantity of interest at time t by using data measured up to and including time t .
2. *Smoothing*, which differs from filtering in that information about the quantity of interest need not be available at time t , and data measured later than time t can be used in obtaining this information. This means that in the case of smoothing there is a *delay* in producing the result of interest. Since in the smoothing process we are able to use data obtained not only up to time t but also data obtained after time t , we would expect smoothing to be more accurate in some sense than filtering.
3. *Prediction*, which is the forecasting side of information processing. The aim here is to derive information about what the quantity of interest will be like at some time $t + \tau$ in the future, for some $\tau > 0$, by using data measured up to and including time t .

We may classify filters into linear and nonlinear. A filter is said to be *linear* if the filtered, smoothed, or predicted quantity at the output of the device is a *linear function of the observations applied to the filter input*. Otherwise, the filter is *nonlinear*.

In the statistical approach to the solution of the *linear filtering problem* as classified above, we assume the availability of certain statistical parameters (i.e., *mean and correlation functions*) of the useful signal and *unwanted additive noise*, and the requirement is to design a linear filter with the noisy data as input so as to minimize the effects of noise at the filter output according to some statistical criterion. A useful approach to this filter-optimization problem is to minimize the mean-square value of the *error signal* that is defined as the difference between some desired response and the actual filter output. For stationary inputs, the resulting solution is *commonly known* as the *Wiener filter*, which is said to be *optimum in the mean-square sense*. A plot of the mean-square value of the error signal versus the adjustable parameters of a linear filter is referred to as the *error-performance surface*. The minimum point of this surface represents the Wiener solution.

The Wiener filter is inadequate for dealing with situations in which *nonstationarity* of the signal and/or noise is intrinsic to the problem. In such situations, the optimum filter has to assume a *time-varying form*. A highly successful solution to this more difficult problem is found in the *Kalman filter*, a powerful device with a wide variety of engineering applications.

Linear filter theory, encompassing both Wiener and Kalman filters, has been developed fully in the literature for *continuous-time* as well as *discrete-time* signals. However, for technical reasons influenced by the wide availability of digital computers and the ever-increasing use of digital signal-processing devices, we find in practice that the discrete-time representation is often the preferred method. Accordingly, in subsequent chapters, we only consider the discrete-time version of Wiener and Kalman filters. In this method of representation, the input and output signals, as well as the characteristics of the filters themselves, are all defined at discrete instants of time. In any case, a continuous-time signal may always be represented by a *sequence of samples* that are derived by observing the signal at uniformly spaced instants of time. No loss of information is incurred during this conversion process provided, of course, we satisfy the well-known *sampling theorem*, according to which the sampling rate has to be greater than twice the highest frequency component of the continuous-time signal. We may thus represent a continuous-time signal $u(t)$ by the sequence $u(n)$, $n = 0, \pm 1, \pm 2, \dots$, where for convenience we have normalized the sampling period to unity, a practice that we follow throughout the book.

2. ADAPTIVE FILTERS

The design of a Wiener filter requires *a priori* information about the statistics of the data to be processed. The filter is optimum only when the statistical characteristics of the input data match the *a priori* information on which the design of the filter is based. When this information is not known completely, however, it may not be possible to design the Wiener filter or else the design may no longer be optimum. A straightforward approach that we may use in such situations is the "estimate and plug" procedure. This is a two-stage process whereby the filter first "estimates" the statistical parameters of the relevant signals and then "plugs" the results so obtained into a *nonrecursive formula* for computing

the filter parameters. For *real-time* operation, this procedure has the disadvantage of requiring excessively elaborate and costly hardware. A more efficient method is to use an *adaptive filter*. By such a device we mean one that is *self-designing* in that the adaptive filter relies for its operation on a *recursive algorithm*, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. The algorithm starts from some predetermined set of *initial conditions*, representing whatever we know about the environment. Yet, in a stationary environment, we find that after successive iterations of the algorithm it *converges* to the optimum Wiener solution in some statistical sense. In a nonstationary environment, the algorithm offers a *tracking* capability, in that it can track time variations in the statistics of the input data, provided that the variations are sufficiently slow.

As a direct consequence of the application of a recursive algorithm whereby the parameters of an adaptive filter are updated from one iteration to the next, the parameters become *data dependent*. This, therefore, means that an adaptive filter is in reality a *nonlinear device*, *in the sense that it does not obey the principle of superposition*. Notwithstanding this property, adaptive filters are commonly classified as linear or nonlinear. An adaptive filter is said to be *linear* if the estimate of a quantity of interest is computed adaptively (at the output of the filter) as a *linear combination of the available set of observations applied to the filter input*. Otherwise, the adaptive filter is said to be *nonlinear*.

A wide variety of recursive algorithms have been developed in the literature for the operation of linear adaptive filters. In the final analysis, the choice of one algorithm over another is determined by one or more of the following factors:

- *Rate of convergence.* This is defined as the number of iterations required for the algorithm, in response to stationary inputs, to converge “close enough” to the optimum Wiener solution in the mean-square sense. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.
- *Misadjustment.* For an algorithm of interest, this parameter provides a quantitative measure of the amount by which the final value of the mean-squared error, averaged over an ensemble of adaptive filters, deviates from the minimum mean-squared error that is produced by the Wiener filter.
- *Tracking.* When an adaptive filtering algorithm operates in a nonstationary environment, the algorithm is required to *track* statistical variations in the environment. The tracking performance of the algorithm, however, is influenced by two contradictory features: (1) rate of convergence, and (b) steady-state fluctuation due to algorithm noise.
- *Robustness.* For an adaptive filter to be *robust*, small disturbances (i.e., disturbances with small energy) can only result in small estimation errors. The disturbances may arise from a variety of factors, internal or external to the filter.
- *Computational requirements.* Here the issues of concern include (a) the number of operations (i.e., multiplications, divisions, and additions/subtractions) required to make one complete iteration of the algorithm, (b) the size of memory locations

required to store the data and the program, and (c) the investment required to program the algorithm on a computer.

- *Structure.* This refers to the structure of information flow in the algorithm, determining the manner in which it is implemented in hardware form. For example, an algorithm whose structure exhibits high modularity, parallelism, or concurrency is well suited for implementation using very large-scale integration (VLSI).¹
- *Numerical properties.* When an algorithm is implemented numerically, inaccuracies are produced due to *quantization errors*. The quantization errors are due to analog-to-digital conversion of the input data and digital representation of internal calculations. Ordinarily, it is the latter source of quantization errors that poses a serious design problem. In particular, there are two basic issues of concern: numerical stability and numerical accuracy. *Numerical stability* is an inherent characteristic of an adaptive filtering algorithm. *Numerical accuracy*, on the other hand, is determined by the number of *bits* (i.e., *binary digits*) used in the numerical representation of data samples and filter coefficients. An adaptive filtering algorithm is said to be numerically robust when it is insensitive to variations in the wordlength used in its digital implementation.

These factors, in their own ways, also enter into the design of nonlinear adaptive filters, except for the fact that we now no longer have a well-defined frame of reference in the form of a Wiener filter. Rather, we speak of a *nonlinear filtering algorithm* that may converge to a local minimum or, hopefully, a global minimum on the error-performance surface.

In the sections that follow, we shall first discuss various aspects of linear adaptive filters. Discussion of nonlinear adaptive filters is deferred to a later section in the chapter.

3. LINEAR FILTER STRUCTURES

The operation of a linear adaptive filtering algorithm involves two basic processes: (1) a *filtering* process designed to produce an output in response to a sequence of input data, and (2) an *adaptive* process, the purpose of which is to provide a mechanism for the *adaptive control* of an *adjustable* set of parameters used in the filtering process. These two processes work interactively with each other. Naturally, the choice of a structure for the filtering process has a profound effect on the operation of the algorithm as a whole.

¹VLSI technology favors the implementation of algorithms that possess high modularity, parallelism, or concurrency. We say that a structure is *modular* when it consists of similar stages connected in cascade. By *parallelism* we mean a large number of operations being performed side by side. By *concurrency* we mean a large number of *similar* computations being performed at the same time.

For a discussion of VLSI implementation of adaptive filters, see Shanbhag and Parhi (1994). This book emphasizes the use of *pipelining*, an architectural technique used for increasing the throughput of an adaptive filtering algorithm.

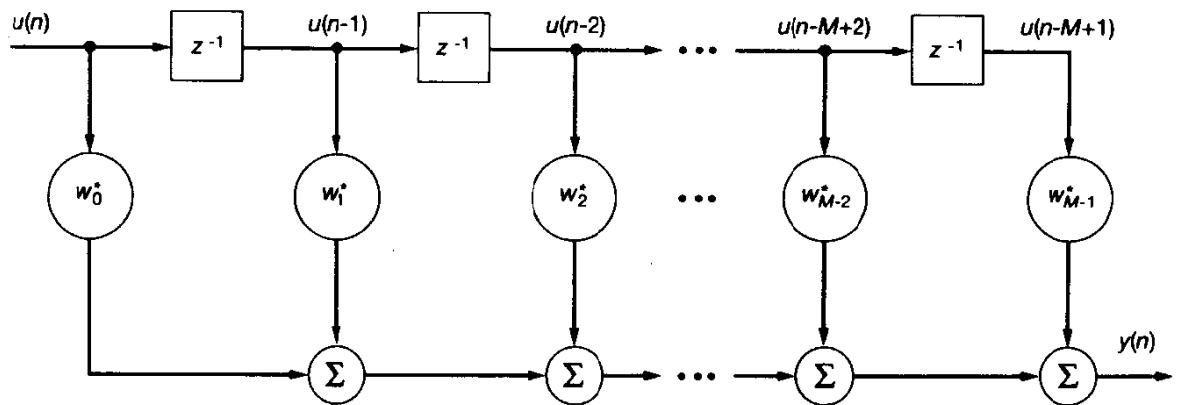


Figure 1 Transversal filter.

There are three types of filter structures that distinguish themselves in the context of an adaptive filter with *finite memory* or, equivalently, *finite-duration impulse response*. The three filter structures are as follows:

1. Transversal filter. The *transversal filter*,² also referred to as a *tapped-delay line filter*, consists of three basic elements, as depicted in Fig. 1: (a) *unit-delay element*, (b) *multiplier*, and (c) *adder*. The number of delay elements used in the filter determines the finite duration of its impulse response. The number of delay elements, shown as $M - 1$ in Fig. 1, is commonly referred to as the *filter order*. In this figure, the delay elements are each identified by the *unit-delay operator* z^{-1} . In particular, when z^{-1} operates on the input $u(n)$, the resulting output is $u(n - 1)$. The role of each multiplier in the filter is to multiply the *tap input* (to which it is connected) by a filter coefficient referred to as a *tap weight*. Thus a multiplier connected to the k th tap input $u(n - k)$ produces the scalar version of the *inner product*, $w_k^* u(n - k)$, where w_k is the respective tap weight and $k = 0, 1, \dots, M - 1$. The asterisk denotes *complex conjugation*, which assumes that the tap inputs and therefore the tap weights are all *complex valued*. The combined role of the adders in the filter is to sum the individual multiplier outputs and produce an overall filter output. For the transversal filter described in Fig. 1, the filter output is given by

$$y(n) = \sum_{k=0}^{M-1} w_k^* u(n - k) \quad (1)$$

²The transversal filter was first described by Kallmann as a continuous-time device whose output is formed as a linear combination of voltages taken from uniformly spaced taps in a nondispersive delay line (Kallmann, 1940). In recent years, the transversal filter has been implemented using digital circuitry, charge-coupled devices, or surface-acoustic wave devices. Owing to its versatility and ease of implementation, the transversal filter has emerged as an essential signal-processing structure in a wide variety of applications.

Equation (1) is called a finite *convolution sum* in the sense that it *convolves* the finite-duration impulse response of the filter, w_n^* , with the filter input $u(n)$ to produce the filter output $y(n)$.

2. Lattice predictor. A *lattice predictor*³ is *modular* in structure in that it consists of a number of individual stages, each of which has the appearance of a lattice, hence the name “lattice” as a structural descriptor. Figure 2 depicts a lattice predictor consisting of $M - 1$ stages; the number $M - 1$ is referred to as the *predictor order*. The m th stage of the lattice predictor in Fig. 2 is described by the pair of input–output relations (assuming the use of complex-valued, wide-sense stationary input data):

$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1) \quad (2)$$

$$b_m(n) = b_{m-1}(n-1) + \kappa_m f_{m-1}(n) \quad (3)$$

where $m = 1, 2, \dots, M - 1$, and $M - 1$ is the *final predictor order*. The variable $f_m(n)$ is the m th *forward prediction error*, and $b_m(n)$ is the m th *backward prediction error*. The coefficient κ_m is called the m th *reflection coefficient*. The forward prediction error $f_m(n)$ is defined as the difference between the input $u(n)$ and its *one-step predicted* value; the latter is based on the set of m *past inputs* $u(n-1), \dots, u(n-m)$. Correspondingly, the backward prediction error $b_m(n)$ is defined as the difference between the input $u(n-m)$ and its “backward” prediction based on the set of m “future” inputs $u(n), \dots, u(n-m+1)$. Considering the conditions at the input of stage 1 in Fig. 2, we have

$$f_0(n) = b_0(n) = u(n) \quad (4)$$

where $u(n)$ is the lattice predictor input at time n . Thus, starting with the *initial conditions* of Eq. (4) and given the set of reflection coefficients $\kappa_1, \kappa_2, \dots, \kappa_{M-1}$, we may determine the final pair of outputs $f_{M-1}(n)$ and $b_{M-1}(n)$ by moving through the lattice predictor, stage by stage.

For a *correlated* input sequence $u(n), u(n-1), \dots, u(n-M+1)$ drawn from a stationary process, the backward prediction errors $b_0, b_1(n), \dots, b_{M-1}(n)$ form a sequence of *uncorrelated* random variables. Moreover, there is a one-to-one correspondence between these two sequences of random variables in the sense that if we are given one of them, we may uniquely determine the other, and vice versa. Accordingly, a linear combination of the backward prediction errors $b_0(n), b_1(n), \dots, b_{M-1}(n)$ may be used to provide an *estimate* of some desired response $d(n)$, as depicted in the lower half of Fig. 2. The arithmetic difference between $d(n)$ and the estimate so produced represents the estimation error $e(n)$. The process described herein is referred to as a *joint-process estimation*. Naturally, we may use the original input sequence $u(n), u(n-1), \dots, u(n-M+1)$ to produce an estimate of the desired response $d(n)$ directly. The indirect method depicted in Fig. 2, however, has the advantage of simplifying the computation of the tap weights h_0, h_1, \dots, h_{M-1}

³The development of the lattice predictor is credited to Itakura and Saito (1972).

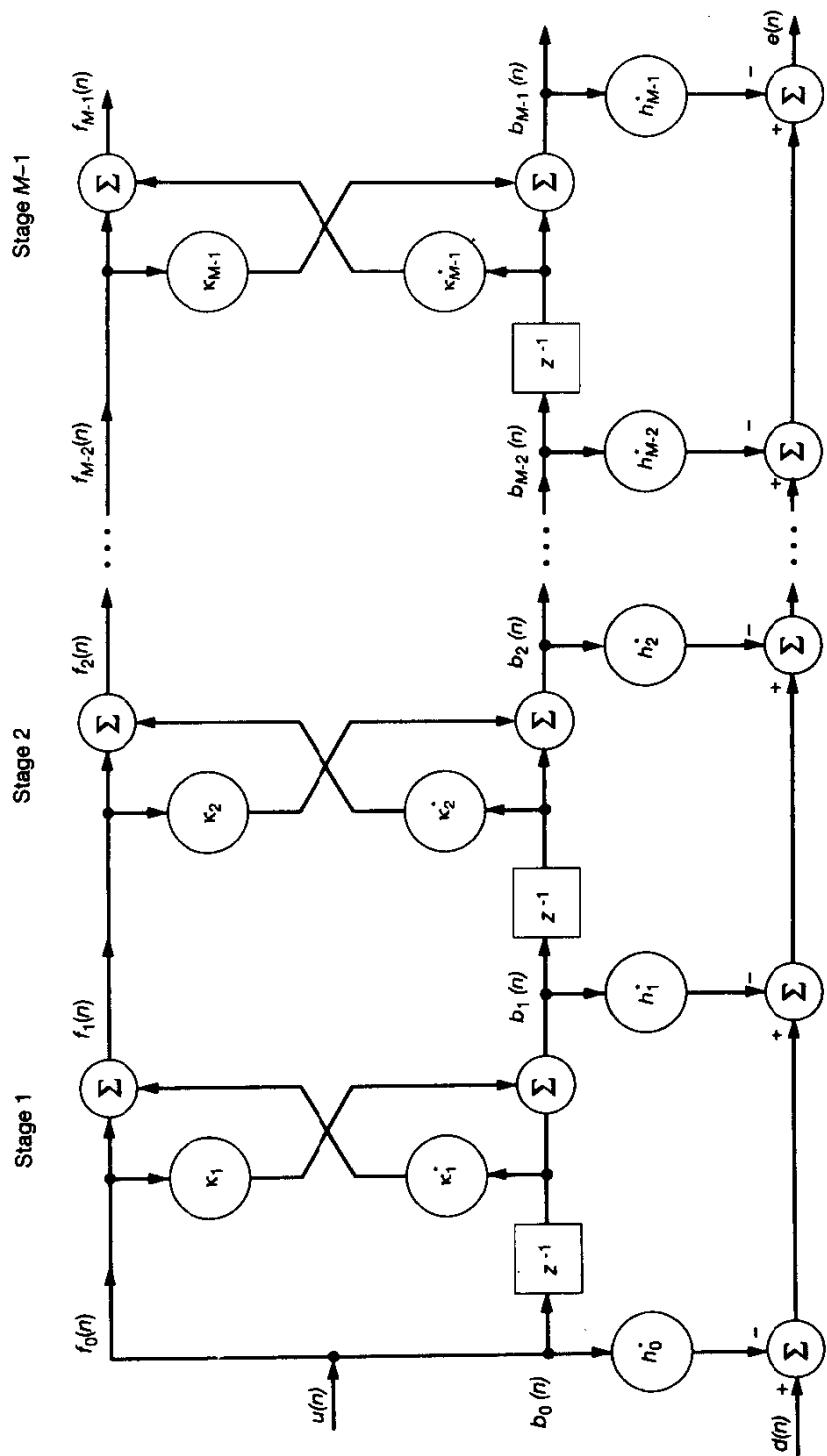


Figure 2 Multistage lattice filter.

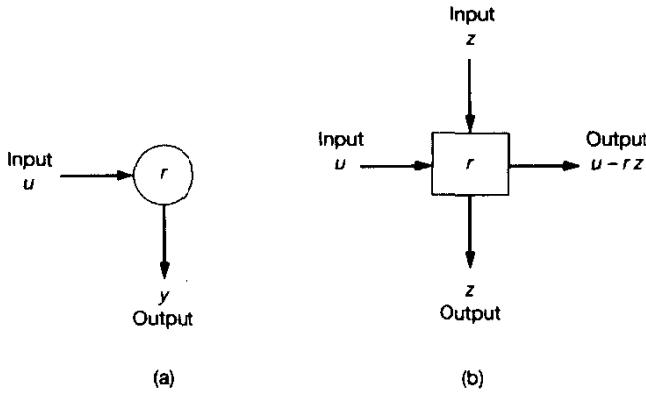


Figure 3 Two basic cells of a systolic array; (a) boundary cell; (b) internal cell.

by exploiting the uncorrelated nature of the corresponding backward prediction errors used in the estimation.

3. Systolic array. A *systolic array*⁴ represents a *parallel computing* network ideally suited for *mapping* a number of important linear algebra computations, such as *matrix multiplication*, *triangularization*, and *back substitution*. Two basic types of processing elements may be distinguished in a systolic array: *boundary cells* and *internal cells*. Their functions are depicted in Figs. 3(a) and 3(b), respectively. In each case, the parameter r represents a value *stored* within the cell. The function of the boundary cell is to produce an output equal to the input u divided by the number r stored in the cell. The function of the internal cell is twofold: (a) to multiply the input z (coming in from the top) by the number r stored in the cell, subtract the product rz from the second input (coming in from the left), and thereby produce the difference $u - rz$ as an output from the right-hand side of the cell, and (b) to transmit the first input z downward without alteration.

Consider, for example, the 3-by-3 triangular array shown in Fig. 4. This systolic array involves a combination of boundary and internal cells. In this case, the triangular array computes an output vector \mathbf{y} related to the input vector \mathbf{u} as follows:

$$\mathbf{y} = \mathbf{R}^{-T} \mathbf{u} \quad (5)$$

where the \mathbf{R}^{-T} is the *inverse* of the transposed matrix \mathbf{R}^T . The elements of \mathbf{R}^T are the respective cell contents of the triangular array. The zeros added to the inputs of the array in Fig. 4 are intended to provide the delays necessary for pipelining the computation described in Eq. (5).

A systolic array architecture, as described herein, offers the desirable features of modularity, local interconnections, and highly pipelined and synchronized parallel processing; the synchronization is achieved by means of a global clock.

We note that the transversal filter of Fig. 1, the joint-process estimator of Fig. 2 based on a lattice predictor, and the triangular systolic array of Fig. 4 have a common

⁴The systolic array was pioneered by Kung and Leiserson (1978). In particular, the use of systolic arrays has made it possible to achieve a high throughput, which is required for many advanced signal processing algorithms to operate in *real time*.

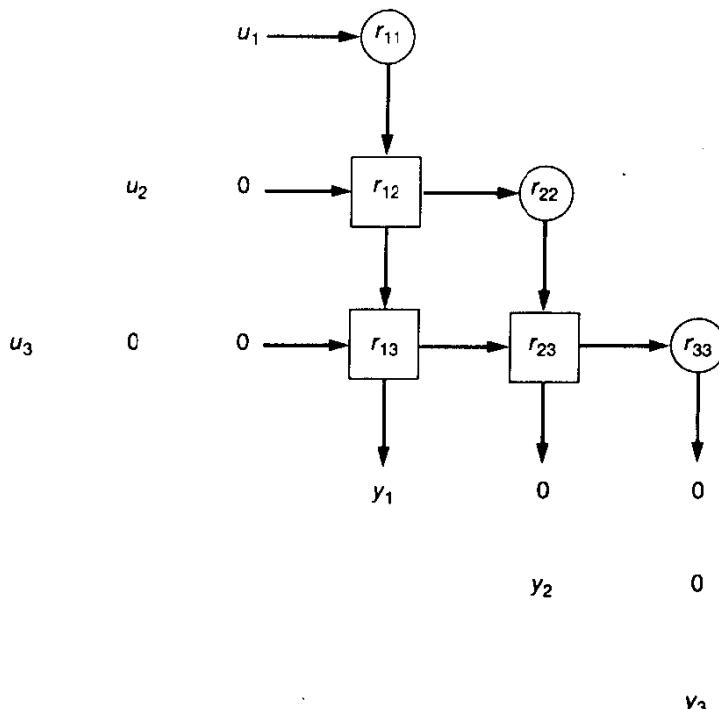


Figure 4 Triangular systolic array.

property: all three of them are characterized by an impulse response of finite duration. In other words, they are examples of a *finite-duration impulse response (FIR) filter*, whose structures contain *feedforward* paths only. On the other hand, the filter structure shown in Fig. 5 is an example of an *infinite-duration impulse response (IIR) filter*. The feature that distinguishes an IIR filter from an FIR filter is the inclusion of *feedback* paths. Indeed, it is the presence of feedback that makes the duration of the impulse response of an IIR filter infinitely long. Furthermore, the presence of feedback introduces a new problem, namely, that of *stability*. In particular, it is possible for an IIR filter to become unstable (i.e., break into oscillation), unless special precaution is taken in the choice of feedback coefficients. By contrast, an FIR filter is inherently *stable*. This explains the reason for the popular use of FIR filters, in one form or another, as the structural basis for the design of linear adaptive filters.

4. APPROACHES TO THE DEVELOPMENT OF LINEAR ADAPTIVE FILTERING ALGORITHMS

There is no unique solution to the linear adaptive filtering problem. Rather, we have a “kit of tools” represented by a variety of recursive algorithms, each of which offers desirable features of its own. The challenge facing the user of adaptive filtering is, first, to understand the capabilities and limitations of various adaptive filtering algorithms and, second,

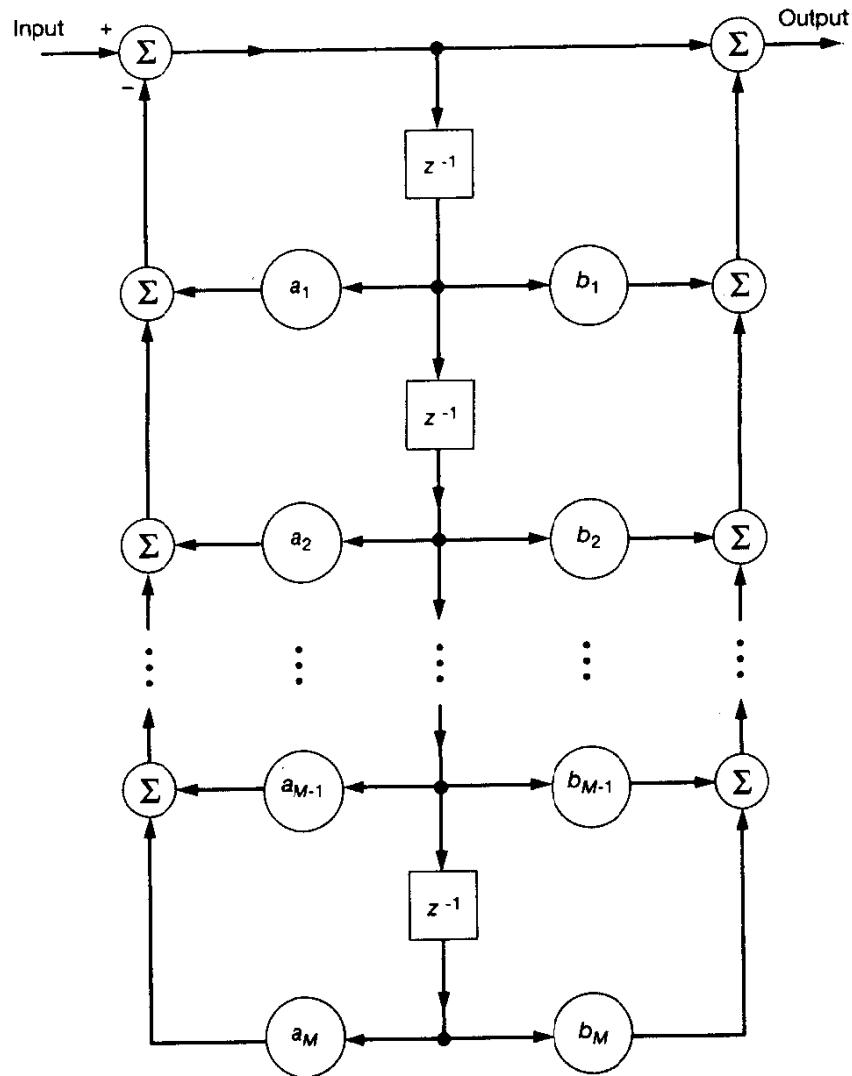


Figure 5 IIR filter.

to use this understanding in the selection of the appropriate algorithm for the application at hand.

Basically, we may identify two distinct approaches for deriving recursive algorithms for the operation of linear adaptive filters, as discussed next.

Stochastic Gradient Approach

Here we may use a tapped-delay line or transversal filter as the structural basis for implementing the linear adaptive filter. For the case of stationary inputs, the *cost function*,⁵ also referred to as the *index of performance*, is defined as the *mean-squared error* (i.e., the mean-square value of the difference between the desired response and the transversal filter output). This cost function is precisely a second-order function of the tap weights in the transversal filter. The dependence of the mean-squared error on the unknown tap weights may be viewed to be in the form of a *multidimensional paraboloid* (i.e., punch bowl) with a uniquely defined bottom or *minimum point*. As mentioned previously, we refer to this paraboloid as the *error-performance surface*; the tap weights corresponding to the minimum point of the surface define the optimum Wiener solution.

To develop a recursive algorithm for updating the tap weights of the adaptive transversal filter, we proceed in two stages. We first modify the system of *Wiener—Hopf equations* (i.e., the matrix equation defining the optimum Wiener solution) through the use of the *method of steepest descent*, a well-known technique in optimization theory. This modification requires the use of a *gradient vector*, the value of which depends on two parameters: the *correlation matrix* of the tap inputs in the transversal filter, and the *cross-correlation vector* between the desired response and the same tap inputs. Next, we use instantaneous values for these correlations so as to derive an *estimate* for the gradient vector, making it assume a *stochastic* character in general. The resulting algorithm is widely known as the *least-mean-square (LMS) algorithm*, the essence of which may be described in words as follows for the case of a transversal filter operating on real-valued data:

$$\begin{pmatrix} \text{updated value} \\ \text{of tap-weight} \\ \text{vector} \end{pmatrix} = \begin{pmatrix} \text{old value} \\ \text{of tap-weight} \\ \text{vector} \end{pmatrix} + \begin{pmatrix} \text{learning-} \\ \text{rate} \\ \text{parameter} \end{pmatrix} \begin{pmatrix} \text{tap-} \\ \text{input} \\ \text{vector} \end{pmatrix} \begin{pmatrix} \text{error} \\ \text{signal} \end{pmatrix}$$

where the error signal is defined as the difference between some desired response and the actual response of the transversal filter produced by the tap-input vector.

The LMS algorithm is simple and yet capable of achieving satisfactory performance under the right conditions. Its major limitations are a relatively slow rate of convergence and a sensitivity to variations in the condition number of the correlation matrix of the tap inputs; the *condition number* of a Hermitian matrix is defined as the ratio of its largest

⁵In the general definition of a function, we speak of a transformation from a vector space into the space of real (or complex) scalars (Luenberger, 1969; Dorn, 1975). A cost function provides a quantitative measure for assessing the quality of performance; hence the restriction of it to a real scalar.

eigenvalue to its smallest eigenvalue. Nevertheless, the LMS algorithm is highly popular and widely used in a variety of applications.

In a nonstationary environment, the orientation of the error-performance surface varies continuously with time. In this case, the LMS algorithm has the added task of continually *tracking* the bottom of the error-performance surface. Indeed, tracking will occur provided that the input data vary slowly compared to the *learning rate* of the LMS algorithm.

The stochastic gradient approach may also be pursued in the context of a lattice structure. The resulting adaptive filtering algorithm is called the *gradient adaptive lattice (GAL) algorithm*. In their own individual ways, the LMS and GAL algorithms are just two members of the *stochastic gradient family* of linear adaptive filters, although it must be said that the LMS algorithm is by far the most popular member of this family.

Least-squares Estimation

The second approach to the development of linear adaptive filtering algorithms is based on the *method of least squares*. According to this method we minimize a cost function or index of performance that is defined as the *sum of weighted error squares*, where the *error* or *residual* is itself defined as the difference between some desired response and the actual filter output. The method of least squares may be formulated with *block estimation* or *recursive estimation* in mind. In block estimation the input data stream is arranged in the form of blocks of equal length (duration), and the filtering of input data proceeds on a block-by-block basis. In recursive estimation, on the other hand, the estimates of interest (e.g., tap weights of a transversal filter) are *updated* on a sample-by-sample basis. Ordinarily, a recursive estimator requires less storage than a block estimator, which is the reason for its much wider use in practice.

Recursive least-squares (RLS) estimation may be viewed as a special case of Kalman filtering. A distinguishing feature of the Kalman filter is the notion of *state*, which provides a measure of all the inputs applied to the filter up to a specific instant of time. Thus, at the heart of the Kalman filtering algorithm we have a recursion that may be described in words as follows:

$$\begin{pmatrix} \text{updated value} \\ \text{of the} \\ \text{state} \end{pmatrix} = \begin{pmatrix} \text{old value} \\ \text{of the} \\ \text{state} \end{pmatrix} + \begin{pmatrix} \text{Kalman} \\ \text{gain} \end{pmatrix} \begin{pmatrix} \text{innovation} \\ \text{vector} \end{pmatrix}$$

where the *innovation vector* represents new information put into the filtering process at the time of the computation. For the present, it suffices to say that there is indeed a one-to-one correspondence between the Kalman variables and RLS variables. This correspondence means that we can tap the vast literature on Kalman filters for the design of linear adaptive filters based on recursive least-squares estimation. Moreover, we may classify the *recursive least-squares family* of linear adaptive filtering algorithms into three distinct categories, depending on the approach taken:

1. *Standard RLS algorithm*, which assumes the use of a transversal filter as the structural basis of the linear adaptive filter. Derivation of the standard RLS algorithm relies on a basic result in linear algebra known as the *matrix inversion lemma*. Most importantly, it enjoys the same virtues and suffers from the same limitations as the standard Kalman filtering algorithm. The limitations include lack of numerical robustness and excessive computational complexity. Indeed, it is these two limitations that have prompted the development of the other two categories of RLS algorithms, described next.
2. *Square-root RLS algorithms*, which are based on *QR-decomposition* of the incoming data matrix. Two well-known techniques for performing this decomposition are the *Householder transformation* and the *Givens rotation*, both of which are data-adaptive transformations. At this point in the discussion, we need to merely say that RLS algorithms based on the Householder transformation or Givens rotation are numerically stable and robust. The resulting linear adaptive filters are referred to as *square-root adaptive filters*, because in a matrix sense they represent the square-root forms of the standard RLS algorithm.
3. *Fast RLS algorithms*. The standard RLS algorithm and square-root RLS algorithms have a computational complexity that increases as the square of M , where M is the number of adjustable weights (i.e., the number of degrees of freedom) in the algorithm. Such algorithms are often referred to as $O(M^2)$ algorithms, where $O(\cdot)$ denotes “order of.” By contrast, the LMS algorithm is an $O(M)$ algorithm, in that its computational complexity increases linearly with M . When M is large, the computational complexity of $O(M^2)$ algorithms may become objectionable from a hardware implementation point of view. There is therefore a strong motivation to modify the formulation of the RLS algorithm in such a way that the computational complexity assumes an $O(M)$ form. This objective is indeed achievable, in the case of temporal processing, first by virtue of the inherent *redundancy* in the *Toeplitz structure* of the input data matrix and, second, by exploiting this redundancy through the use of *linear least-squares prediction in both the forward and backward directions*. The resulting algorithms are known collectively as *fast RLS algorithms*; they combine the desirable characteristics of recursive linear least-squares estimation with an $O(M)$ computational complexity. Two types of fast RLS algorithms may be identified, depending on the filtering structure employed:
 - *Order-recursive adaptive filters*, which are based on a latticelike structure for making linear forward and backward predictions.
 - *Fast transversal filters*, in which the linear forward and backward predictions are performed using separate transversal filters.

Certain (but not all) realizations of order-recursive adaptive filters are known to be numerically stable, whereas fast transversal filters suffer from a numerical sta-

bility problem and therefore require some form of stabilization for them to be of practical use.

An introductory discussion of linear adaptive filters would be incomplete without saying something about their tracking behavior. In this context, we note that stochastic gradient algorithms such as the LMS algorithm are *model-independent*; generally speaking, we would expect them to exhibit good tracking behavior, which indeed they do. In contrast, RLS algorithms are *model-dependent*; this, in turn, means that their tracking behavior may be inferior to that of a member of the stochastic gradient family, unless care is taken to minimize the mismatch between the mathematical model on which they are based and the underlying physical process responsible for generating the input data.

How to Choose an Adaptive Filter

Given the wide variety of adaptive filters available to a system designer, how can a choice be made for an application of interest? Clearly, whatever the choice, it has to be *cost-effective*. With this goal in mind, we may identify three important issues that require attention: *computational cost*, *performance*, and *robustness*. The use of computer simulation provides a good first step in undertaking a detailed investigation of these issues. We may begin by using the LMS algorithm as an adaptive filtering tool for the study. The LMS algorithm is relatively simple to implement. Yet it is powerful enough to evaluate the practical benefits that may result from the application of adaptivity to the problem at hand. Moreover, it provides a practical frame of reference for assessing any further improvement that may be attained through the use of more sophisticated adaptive filtering algorithms. Finally, the study must include tests with real-life data, for which there is no substitute.

Practical applications of adaptive filtering are very diverse, with each application having peculiarities of its own. The solution for one application may not be suitable for another. Nevertheless, to be successful we have to develop a physical understanding of the environment in which the filter has to operate and thereby relate to the realities of the application of interest.

5. REAL AND COMPLEX FORMS OF ADAPTIVE FILTERS

In the development of adaptive filtering algorithms, regardless of their origin, it is customary to assume that the input data are in baseband form. The term "baseband" is used to designate the band of frequencies representing the original (message) signal as generated by the source of information.

In such applications as communications, radar, and sonar, the information-bearing signal component of the receiver input typically consists of a message signal *modulated* onto a carrier wave. The bandwidth of the message signal is usually small compared to the carrier frequency, which means that the modulated signal is a *narrow-band signal*. To obtain the baseband representation of a narrow-band signal, the signal is translated down

in frequency in such a way that the effect of the carrier wave is completely removed, yet the information content of the message signal is fully preserved. In general, the baseband signal so obtained is *complex*. In other words, a sample $u(n)$ of the signal may be written as

$$u(n) = u_I(n) + j u_Q(n) \quad (6)$$

where $u_I(n)$ is the *in-phase* (real) component, and $u_Q(n)$ is the *quadrature* (imaginary) component. Equivalently, we may express $u(n)$ as

$$u(n) = |u(n)| e^{j\phi(n)} \quad (7)$$

where $|u(n)|$ is the *magnitude* and $\phi(n)$ is the *phase angle*.

Accordingly, the theory of adaptive filters (both linear and nonlinear) developed in subsequent chapters of the book assumes the use of complex signals. An adaptive filtering algorithm so developed is said to be in *complex form*. The important virtue of complex adaptive filters is that they preserve the mathematical formulation and elegant structure of complex signals encountered in the aforementioned areas of application.

If the signals to be processed are *real*, we naturally use the *real form* of the adaptive-filtering algorithm of interest. Given the complex form of an adaptive filtering algorithm, it is straightforward to deduce the corresponding real form of the algorithm. Specifically, we do two things:

1. The operation of *complex conjugation*, wherever in the algorithm, is simply removed.
2. The operation of *Hermitian transposition* (i.e., conjugate transposition) of a matrix, wherever in the algorithm, is replaced by ordinary transposition.

Simply put, complex adaptive filters include real adaptive filters as special cases.

6. NONLINEAR ADAPTIVE FILTERS

The theory of linear optimum filters is based on the mean-square error criterion. The Wiener filter that results from the minimization of such a criterion, and which represents the goal of linear adaptive filtering for a stationary environment, can only relate to second-order statistics of the input data and no higher. This constraint limits the ability of a linear adaptive filter to extract information from input data that are non-Gaussian. Despite its theoretical importance, the existence of Gaussian noise is open to question (Johnson and Rao, 1990). Moreover, non-Gaussian processes are quite common in many signal processing applications encountered in practice. The use of a Wiener filter or a linear adaptive filter to extract signals of interest in the presence of such non-Gaussian processes will therefore yield suboptimal solutions. We may overcome this limitation by incorporating some form of *nonlinearity* in the structure of the adaptive filter to take care of higher-order statistics. Although by so doing, we no longer have the Wiener filter as a frame of refer-

ence and so complicate the mathematical analysis, we would expect to benefit in two significant ways: improving learning efficiency and a broadening of application areas.

Fundamentally, there are two types of nonlinear adaptive filters, as described next.

Volterra-based Nonlinear Adaptive Filters

In this type of a nonlinear adaptive filter, the nonlinearity is localized at the front end of the filter. It relies on the use of a *Volterra series*⁶ that provides an attractive method for describing the input–output relationship of a nonlinear device with memory. This special form of a series derives its name from the fact that it was first studied by Vito Volterra around 1880 as a generalization of the Taylor series of a function. But Norbert Wiener (1958) was the first to use the Volterra series to model the input–output relationship of a nonlinear system.

Let the time series x_n denote the input of a nonlinear discrete-time system. We may then combine these input samples to define a set of *discrete Volterra kernels* as follows:

$$H_0 = \text{zero-order (dc) term}$$

$$H_1[x_n] = \text{first-order (linear) term}$$

$$= \sum_i h_i x_i$$

$$H_2[x_n] = \text{second-order (quadratic) term}$$

$$= \sum_i \sum_j h_{ij} x_i x_j$$

$$H_3[x_n] = \text{third-order (cubic) term}$$

$$= \sum_i \sum_j \sum_k h_{ijk} x_i x_j x_k$$

and so on for higher-order terms. Ordinarily, the nonlinear model coefficients, the h 's, are fixed by analytical methods. We may thus decompose a nonlinear adaptive filter as follows:⁷

- A *nonlinear Volterra state expander* that combines the set of input values x_0, x_1, \dots, x_n to produce a larger set of outputs u_0, u_1, \dots, u_q for which q is larger than n . For example, the extension vector for a (3,2) system has the form

$$\mathbf{u} = [1, x_0, x_1, x_2, x_0^2, x_0 x_1, x_0 x_2, x_1 x_0, x_1^2, x_1 x_2, x_2 x_0, x_2 x_1, x_2^2]^T$$

- A *linear FIR adaptive filter* that operates on the u_k (i.e., elements of \mathbf{u}) as inputs to produce an estimate \hat{d}_n of some desired response d_n .

⁶For a discussion of Volterra series, see the book by Schetzen (1981).

⁷The idea described herein is discussed in Rayner and Lynch (1989) and Lynch and Rayner (1989).

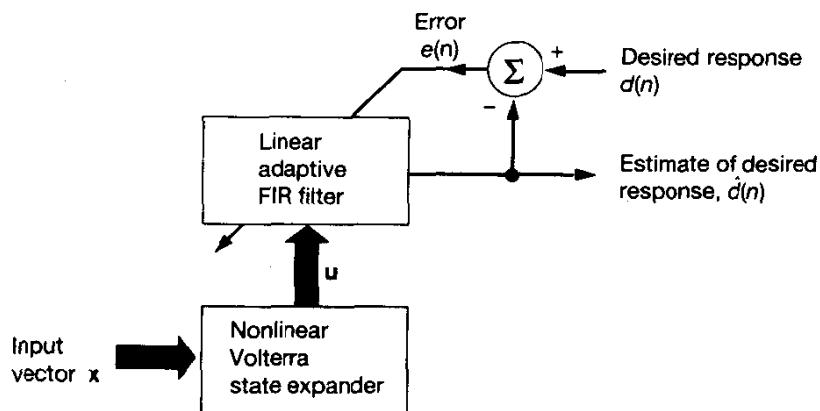


Figure 6 Volterra-based nonlinear adaptive filter.

The important thing to note here is that by using a scheme similar to that described in Fig. 6, we may expand the use of linear adaptive filters to include Volterra filters.

Neural Networks

An *artificial neural network*, or a *neural network* as it is commonly called, consists of the interconnection of a large number of nonlinear processing units called *neurons*; that is, the nonlinearity is distributed throughout the network. The development of neural networks, right from their inception, has been motivated by the way the human brain performs its operations; hence their name.

In this book, we are interested in a particular class of neural networks that *learn* about their environment in a *supervised manner*. In other words, as with the conventional form of a linear adaptive filter, we have a desired response that provides a target signal, which the neural network tries to approximate during the learning process. The approximation is achieved by adjusting a set of free parameters, called *synaptic weights*, in a systematic manner. In effect, the synaptic weights provide a mechanism for storing the information content of the input data.

In the context of adaptive signal processing applications, neural networks offer the following advantages:

- *Nonlinearity*, which makes it possible to account for the nonlinear behavior of physical phenomena responsible for generating the input data
- The ability to *approximate any prescribed input-output mapping* of a continuous nature
- *Weak statistical assumptions* about the environment, in which the network is embedded
- *Learning capability*, which is accomplished by undertaking a training session with input-output examples that are representative of the environment

- *Generalization*, which refers to the ability of the neural network to provide a satisfactory performance in response to *test data* never seen by the network before
- *Fault tolerance*, which means that the network continues to provide an acceptable performance despite the failure of some neurons in the network
- *VLSI implementability*, which exploits the massive parallelism built into the design of a neural network.

This is indeed an impressive list of attributes, which accounts for the widespread interest in the use of neural networks to solve signal-processing tasks that are too difficult for conventional (linear) adaptive filters.

7. APPLICATIONS

The ability of an adaptive filter to operate satisfactorily in an unknown environment and track time variations of input statistics make the adaptive filter a powerful device for signal-processing and control applications. Indeed, adaptive filters have been successfully applied in such diverse fields as communications, radar, sonar, seismology, and biomedical engineering. Although these applications are indeed quite different in nature, nevertheless, they have one basic common feature: an input vector and a desired response are used to compute an estimation error, which is in turn used to control the values of a set of adjustable filter coefficients. The adjustable coefficients may take the form of tap weights, reflection coefficients, rotation parameters, or synaptic weights, depending on the filter structure employed. However, the essential difference between the various applications of adaptive filtering arises in the manner in which the desired response is extracted. In this context, we may distinguish four basic classes of adaptive filtering applications, as depicted in Fig. 7. For convenience of presentation, the following notations are used in this figure:

\mathbf{x} = input applied to the adaptive filter

y = output of the adaptive filter

d = desired response

$e = d - y$ = estimation error.

The functions of the four basic classes of adaptive filtering applications depicted herein are as follows:

- I. *Identification* [Fig. 7(a)]. The notion of a *mathematical model* is fundamental to sciences and engineering. In the class of applications dealing with identification, an adaptive filter is used to provide a linear model that represents the best fit (in some sense) to an *unknown plant*. The plant and the adaptive filter are driven by the same input. The plant output supplies the desired response for the adaptive filter. If the plant is dynamic in nature, the model will be time varying.

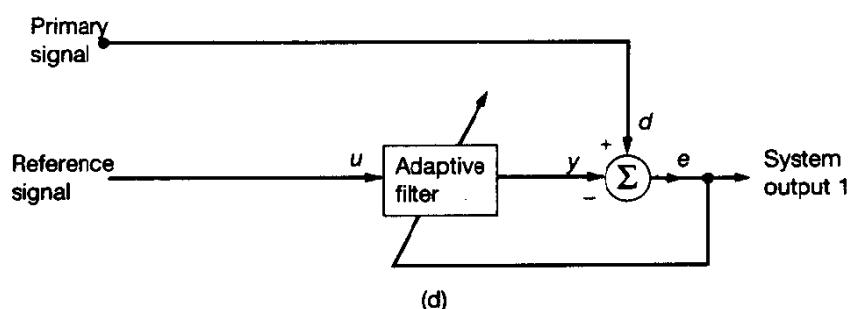
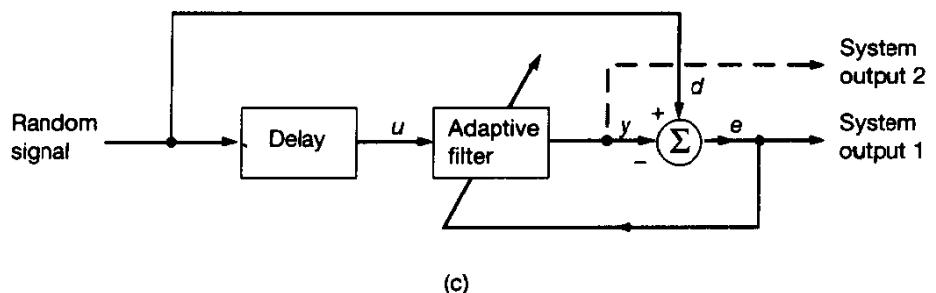
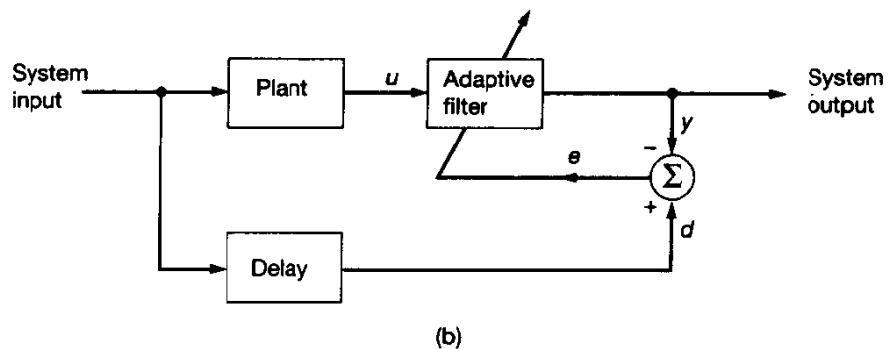
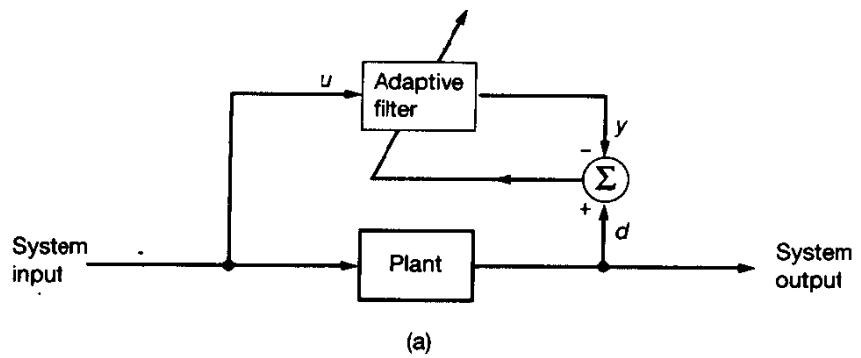


Figure 7 Four basic classes of adaptive filtering applications: (a) class I: identification; (b) class II: inverse modeling; (c) class III: prediction; (d) class IV: interference canceling.

- II.** *Inverse modeling* [Fig. 7(b)]. In this second class of applications, the function of the adaptive filter is to provide an *inverse model* that represents the best fit (in some sense) to an *unknown noisy plant*. Ideally, in the case of a linear system, the inverse model has a transfer function equal to the *reciprocal (inverse)* of the plant's transfer function, such that the combination of the two constitutes an ideal transmission medium. A delayed version of the plant (system) input constitutes the desired response for the adaptive filter. In some applications, the plant input is used without delay as the desired response.
- III.** *Prediction* [Fig. 7(c)]. Here the function of the adaptive filter is to provide the best *prediction* (in some sense) of the present value of a random signal. The present value of the signal thus serves the purpose of a desired response for the adaptive filter. Past values of the signal supply the input applied to the adaptive filter. Depending on the application of interest, the adaptive filter output or the estimation (prediction) error may serve as the system output. In the first case, the system operates as a *predictor*; in the latter case, it operates as a *prediction-error filter*.
- IV.** *Interference canceling* [Fig. 7(d)]. In this final class of applications, the adaptive filter is used to cancel *unknown interference* contained (alongside an information-bearing signal component) in a *primary signal*, with the cancellation being optimized in some sense. The primary signal serves as the desired response for the adaptive filter. A *reference (auxiliary) signal* is employed as the input to the adaptive filter. The reference signal is derived from a sensor or set of sensors located in relation to the sensor(s) supplying the primary signal in such a way that the information-bearing signal component is weak or essentially undetectable.

In Table 1 we have listed some applications that are illustrative of the four basic classes of adaptive filtering applications. These applications, totaling twelve, are drawn from the fields of control systems, seismology, electrocardiography, communications, and radar. They are described individually in the remainder of this section.

System Identification

System identification is the experimental approach to the modeling of a process or a plant (Goodwin and Payne, 1977; Ljung and Söderström, 1983; Ljung, 1987; Söderström and Stoica, 1988; Åström and Wittenmark, 1990). It involves the following steps: experimental planning, the selection of a model structure, parameter estimation, and model validation. The procedure of system identification, as pursued in practice, is iterative in nature in that we may have to go back and forth between these steps until a satisfactory model is built. Here we discuss briefly the idea of adaptive filtering algorithms for estimating the parameters of an unknown plant modeled as a transversal filter.

Suppose we have an unknown dynamic plant that is linear and time varying. The plant is characterized by a *real-valued* set of discrete-time measurements that describe the

TABLE 1 APPLICATIONS OF ADAPTIVE FILTERS

Class of adaptive filtering	Application
I. Identification	System identification Layered earth modeling
II. Inverse modeling	Predictive deconvolution Adaptive equalization Blind equalization
III. Prediction	Linear predictive coding Adaptive differential pulse-code modulation Autoregressive spectrum analysis Signal detection
IV. Interference canceling	Adaptive noise canceling Echo cancellation Adaptive beamforming

variation of the plant output in response to a known stationary input. The requirement is to develop an *on-line transversal filter model* for this plant, as illustrated in Fig. 8. The model consists of a finite number of unit-delay elements and a corresponding set of adjustable parameters (tap weights).

Let the available input signal at time n be denoted by the set of samples: $u(n)$, $u(n - 1)$, \dots , $u(n - M + 1)$, where M is the number of adjustable parameters in the model. This input signal is applied simultaneously to the plant and the model. Let their respective outputs be denoted by $d(n)$ and $y(n)$. The plant output $d(n)$ serves the purpose of a desired response for the adaptive filtering algorithm employed to adjust the model parameters. The model output is given by

$$y(n) = \sum_{k=0}^{M-1} \hat{w}_k(n) u(n - k) \quad (8)$$

where $\hat{w}_0(n)$, $\hat{w}_1(n)$, \dots , and $\hat{w}_{M-1}(n)$ are the estimated model parameters. The model output $y(n)$ is compared with the plant output $d(n)$. The difference between them, $d(n) - y(n)$, defines the *modeling (estimation) error*. Let this error be denoted by $e(n)$.

Typically, at time n , the modeling error $e(n)$ is nonzero, implying that the model deviates from the plant. In an attempt to account for this deviation, the error $e(n)$ is applied to an *adaptive control algorithm*. The samples of the input signal, $u(n)$, $u(n - 1)$, \dots , $u(n - M + 1)$, are also applied to the algorithm. The combination of the transversal filter and the adaptive control algorithm constitutes the adaptive filtering algorithm. The algorithm is designed to control the adjustments made in the values of the model parameters. As a result, the model parameters assume a new set of values for use on the next iteration. Thus, at time $n + 1$, a new model output is computed, and with it a new value for the modeling error. The operation described is then repeated. This process is continued for a sufficiently large number of iterations (starting from time $n = 0$), until the deviation of the

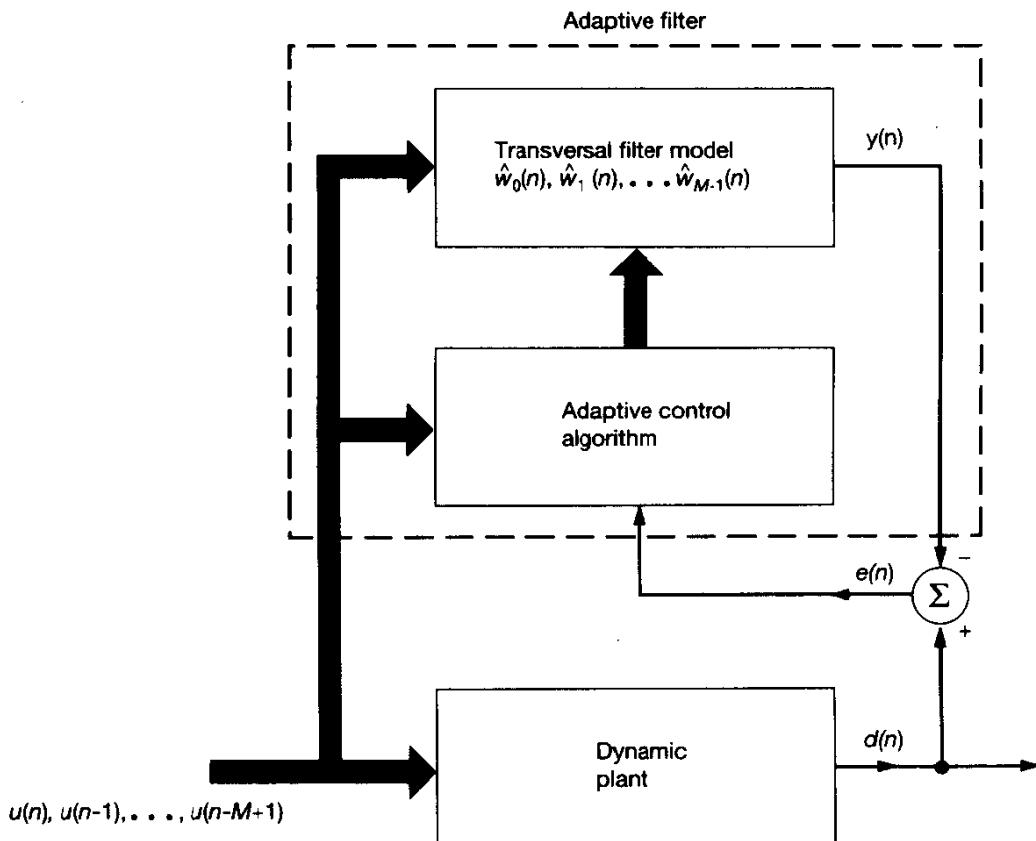


Figure 8 System identification.

model from the plant, measured by the magnitude of the modeling error $e(n)$, becomes sufficiently small in some statistical sense.

When the plant is time varying, the plant output is *nonstationary*, and so is the desired response presented to the adaptive filtering algorithm. In such a situation, the adaptive filtering algorithm has the task of not only keeping the modeling error small but also continually tracking the time variations in the dynamics of the plant.

Layered Earth Modeling

In *exploration seismology*, we usually think of a layered model of the earth (Robinson and Treitel, 1980; Justice, 1985; Mendel, 1986; Robinson and Durrani, 1986). In order to collect (record) seismic data for the purpose of characterizing such a model and thereby unraveling the complexities of the earth's surface, it is customary to use the *method of reflection seismology* that involves the following:

1. A source of seismic energy (e.g., dynamite, air gun) that is typically activated on the surface of the earth.

2. *Propagation* of the seismic signal away from the source and deep into the earth's crust.
3. *Reflection* of seismic waves from the interfaces between the earth's geological layers.
4. *Picking up and recording* the seismic returns (i.e., reflections of seismic waves from the interfaces) that carry information about the subsurface structure. On land, *geophones* (consisting of small sensors implanted into the earth) are used to pick up the seismic returns.

The method of reflection seismology, combined with a lot of signal processing, is capable of supplying a two- or three-dimensional "picture" of the earth's subsurface, down to about 20,000 to 30,000 feet and with high enough accuracy and resolution. This picture is then examined by an "interpreter" to see if it is likely that the part of the earth's subsurface (under exploration) contains hydrocarbon (petroleum) reservoirs. Accordingly, a decision is made whether or not to drill a well, which (in the final analysis) is the only way of knowing if petroleum is actually present.

A seismic wave is similar in nature to an acoustic wave, except that the earth permits the propagation of shear waves as well as compressional waves. (In an acoustic medium, only compressional waves are supported.) The earth tends to act like an *elastic medium* for the propagation of seismic waves. The property of elasticity means that a fluid or solid body resists changes in size and shape due to the applications of an external force, and that the body is restored to its original size and shape upon removal of the force. It is this property that permits the propagation of seismic waves through the earth.

An important issue in exploration seismology is the interpretation of seismic returns from the different geological layers of the earth. This interpretation is fundamental to the *identification* of crusted regions such as depth rocks, sand layers, or sedimentary layers. The sedimentary layers are of particular interest because they may contain hydrocarbon reservoirs. The idea of a layered earth model plays a key role here.

The *layered-earth model* is based on the physical fact that seismic-wave motion in each layer is characterized by two components propagating in opposite directions (Robinson and Durrani, 1986). This phenomenon is illustrated in Fig. 9. To understand the interaction between downgoing and upgoing waves, we have reproduced a portion of this diagram in Fig. 10(a), which pertains to the k th interface. The picture shown in Fig. 10(a) is decomposed into two parts, as depicted in Fig. 10(b) and 10(c). We thus observe the following:

- In layer k , there is an *upgoing* wave that consists of the superposition of the reflection of a downgoing wave incident on the k th interface (i.e., boundary) and the transmission of an upgoing (incident) wave from layer $k + 1$.
- In layer $k + 1$, there is a *downgoing* wave that consists of the superposition of the transmission of a downgoing (incident) wave from layer k and the reflection of an upgoing wave incident on the k th interface.

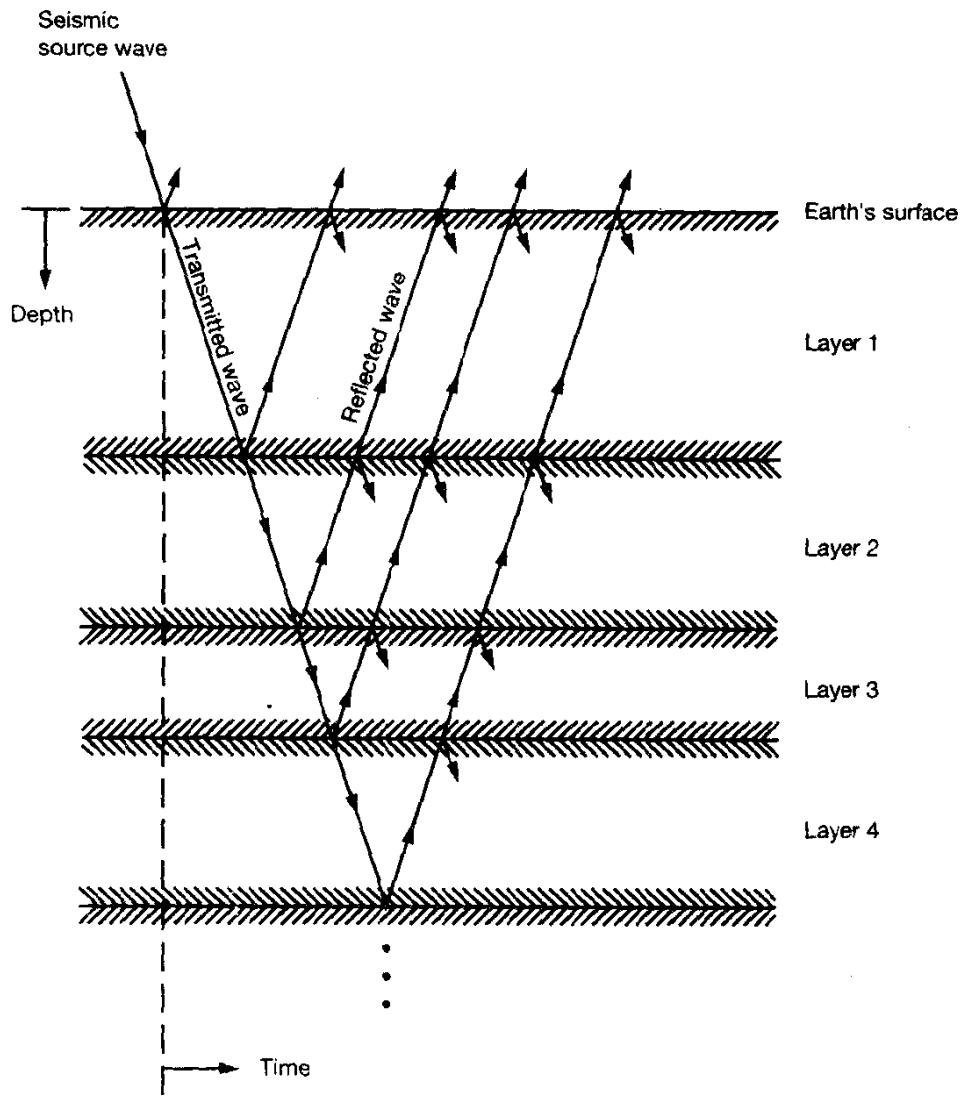


Figure 9 Upgoing and downgoing waves in different layers of the layered earth model.
Note: The layers are unevenly spaced to add a sense of realism.

Lattice Model. Let c_k denote the upward reflection coefficient of the k th interface [see Fig. 10(b)]. Let $d_k(n)$ and $u_k(n)$ denote the downgoing and upgoing waves, respectively, at the top of layer k , and let $d'_k(n)$ and $u'_k(n)$ denote the downgoing and upgoing waves, respectively, at the bottom of layer k , as depicted in Fig. 10(a). The index n denotes discrete time. Ideally, the waves propagate through the medium without distortion, or absorption. Accordingly, we have from Fig. 10(a),

$$d'_k(n) = d_k(n - \frac{1}{2}) \quad (9)$$

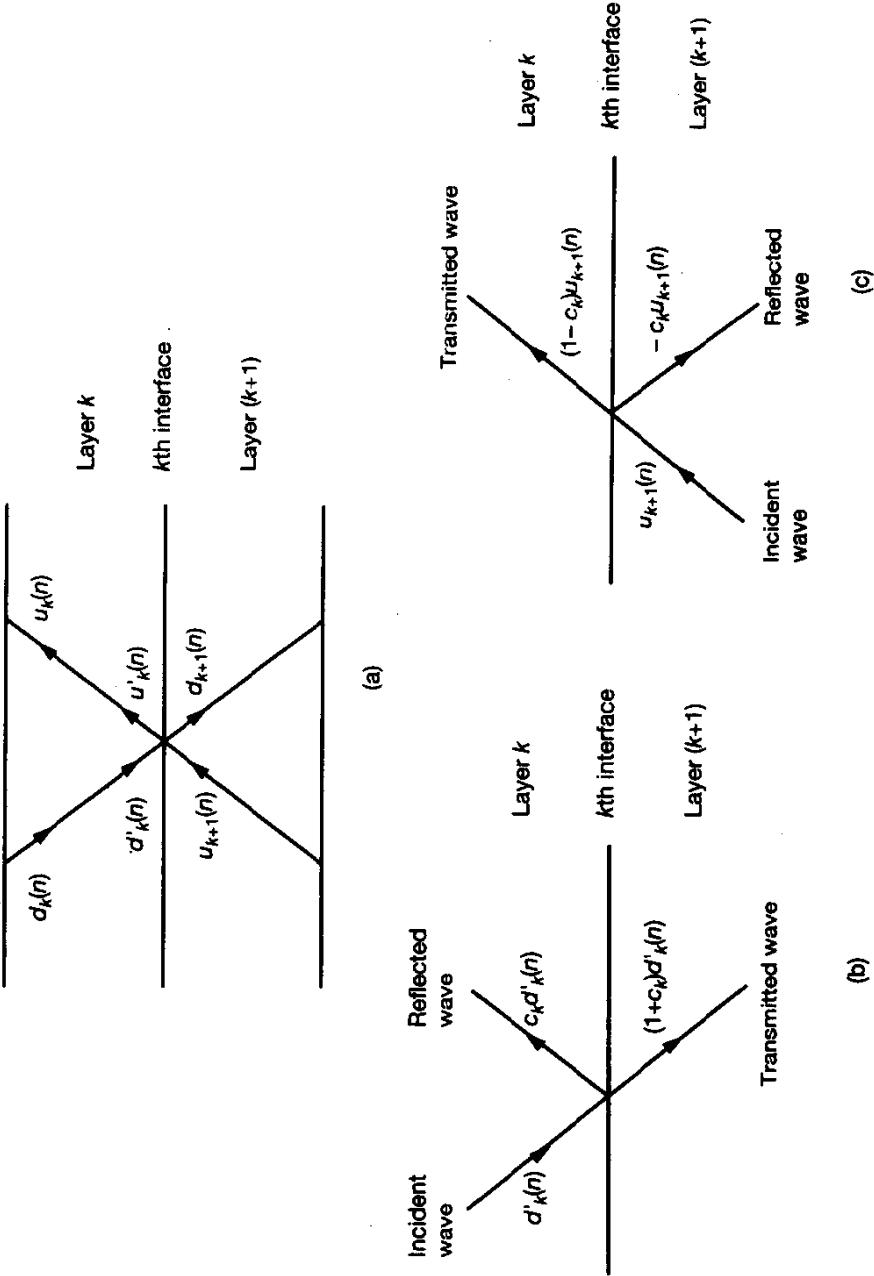


Figure 10 (a) Propagation of seismic waves through a pair of adjacent layers; (b) effects of downgoing incident wave; (c) effects of upgoing incident wave.

and

$$u'_k(n) = u_k(n + \frac{1}{2}) \quad (10)$$

where the travel time from the top of a layer to its bottom (or vice versa) is assumed to be one-half of a time unit. The superposition of the pictures depicted in parts (b) and (c) of Fig. 10 and comparison with that of part (a) yields the following interactions between the downgoing and upgoing waves:

$$d'_{k+1}(n) = -c_k u_{k+1}(n) + (1 + c_k) d'_k(n) \quad (11)$$

and

$$u'_k(n) = c_k d'_k(n) + (1 - c_k) u_{k+1}(n) \quad (12)$$

The upward *transmission coefficient*⁸ of the k th interface is defined by [see Fig. 10(c)]

$$\tau'_k = 1 - c_k \quad (13)$$

Thus, using this definition in Eq. (12), and also using this equation to eliminate $u_{k+1}(n)$ from Eq. (11), we obtain

$$u'_k(n) = c_k d'_k(n) + \tau'_k u_{k+1}(n) \quad (14)$$

and

$$d'_{k+1}(n) = \frac{1}{\tau'_k} d'_k(n) - \frac{c_k}{\tau'_k} u'_k(n) \quad (15)$$

Using this pair of equations, we may construct a *lattice model* for layer k , as shown in Fig. 11(a) (Robinson and Durrani, 1986). Moreover, we may extend this idea to develop a *multistage lattice model*, shown in block diagram form in Fig. 11(b), which depicts the propagation of waves through several layers of the medium. The lattice model for each layer has the details given in Fig. 11(a). The combined use of these two figures provides a great deal of physical insight into the interaction of downgoing and upgoing waves as they propagate from one layer to the next.

Examination of Eq. (14) reveals that the evaluation of $u'_k(n)$ at the bottom of layer k requires knowledge of $u_{k+1}(n)$ at the top of layer $k + 1$. But $u_{k+1}(n)$ is not available until the layer $k + 1$ has been dealt with. The lattice model of Fig. 11 is therefore of limited practical use. To overcome this limitation, we may use the z -transform to modify this model. Specifically, applying the z -transform to Eqs. (9), (10), (14), and (15) and manipulating them into matrix form, we get the so-called *scattering equation*:

$$\begin{bmatrix} D_{k+1}(z) \\ U_{k+1}(z) \end{bmatrix} = \frac{z^{1/2}}{\tau'_k} \begin{bmatrix} z^{-1} & -c_k \\ -c_k z^{-1} & 1 \end{bmatrix} \begin{bmatrix} D_k(z) \\ U_k(z) \end{bmatrix} \quad (16)$$

⁸The prime in the upward transmission coefficient τ'_k is used to distinguish it from the *downward* transmission coefficient [see Fig. 11(a)], given by

$$\tau_k = 1 + c_k$$

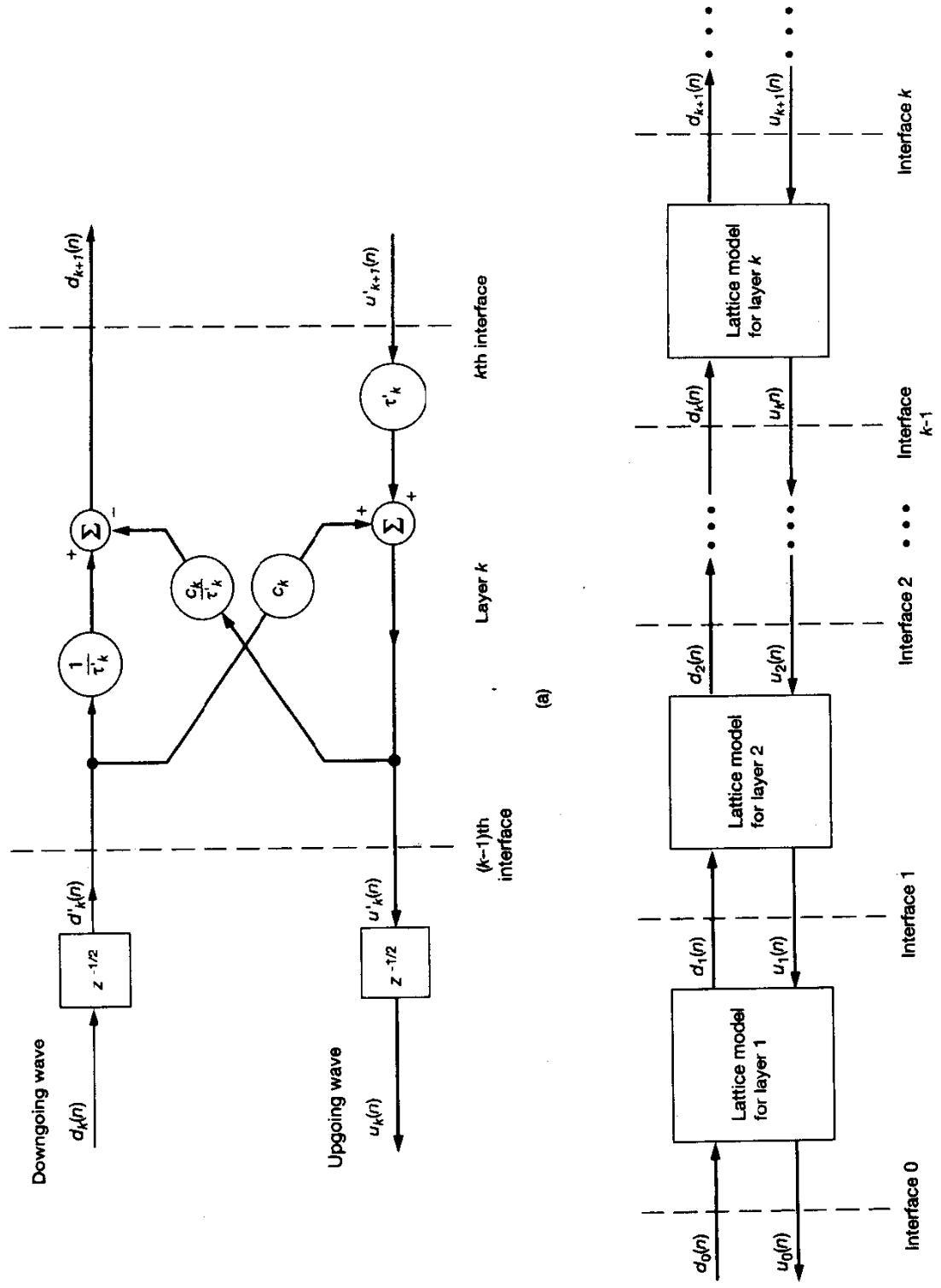


Figure 11 (a) Lattice model for layer k ; (b) multistage lattice model of the layered earth model, with each lattice configuration having the form given in part (a).

where z^{-1} is the unit-delay operator. The 2-by-2 matrix on the right-hand side of Eq. (16) is called the *scattering matrix*. Thus, on the basis of Eq. (16), we may construct the *modified lattice model* for layer k , shown in Fig. 12(a) (Robinson and Durrani, 1986). Correspondingly, the multistage version of the modified lattice model is as shown in Fig. 12(b).

The following points are noteworthy in the context of the modified lattice model of Fig. 12 for the propagation of compressional seismic waves in the subsurface of the earth:

1. The lattice structure of the model has *physical* significance, since it follows naturally from the notion of a layered earth.
2. The structure for each layer (state) of the model is *symmetric*.
3. The reciprocal of the transmission coefficient for each layer merely plays the role of a *scaling factor* insofar as input-output relations are concerned. Specifically, for layer k , we may remove $1/\tau'_k$ from the top path of the model in Fig. 12(a) simply by absorbing it in $D_{k+1}(z)$. Similarly, we may remove $1/\tau'_k$ from the bottom path by absorbing it in $U_{k+1}(z)$. Moreover, the values of the transmission coefficients $\tau'_1, \tau'_2, \dots, \tau'_k, \dots$ are determined from the respective values of the reflection coefficients $c_1, c_2, \dots, c_k, \dots$ by using Eq. (13).
4. The overall model for layers $1, 2, \dots, k, \dots$ is *uniquely determined by the sequence of reflection coefficients $c_1, c_2, \dots, c_k, \dots$*

A case of special interest arises when

$$u_{k+1}(n) = 0, \quad k \text{ is the deepest layer} \quad (17)$$

This case corresponds to the case when the *final* interface [i.e., the $(k + 1)$ th interface] acts as a *perfect absorber*. In other words, there is no outgoing wave from the deepest layer, so Eq. (17) follows. This equation thus represents the *boundary condition* on the lattice model of Fig. 11. The corresponding boundary condition for the modified lattice model of Fig. 12 is

$$U_{k+1}(z) = 0, \quad k \text{ is the deepest layer} \quad (18)$$

Given this boundary condition and the sequence of reflection coefficients $c_1, c_2, \dots, c_k, \dots$, we may then use the modified lattice model of Fig. 12(b) (in a stage-by-stage fashion) to determine $U_0(z)$, the z -transform of the output (outgoing) seismic wave $u_0(n)$ at the earth's surface, in terms of $D_0(z)$, the z -transform of the input (downgoing) seismic wave $d_0(n)$.

Tapped-Delay-Line (Transversal Model). Figure 13 depicts a *tapped-delay-line model* for a layered earth. It provides a local parameterization of the propagation (scattering) phenomenon in the earth's subsurface. According to the alternative model, the input (downgoing) seismic wave $d_0(n)$ and the output (upgoing) seismic wave $u_0(n)$ are, in general, linearly related by the *infinite convolution sum*

$$u_0(n) = \sum_{k=0}^{\infty} w_k d_0(n - k) \quad (19)$$

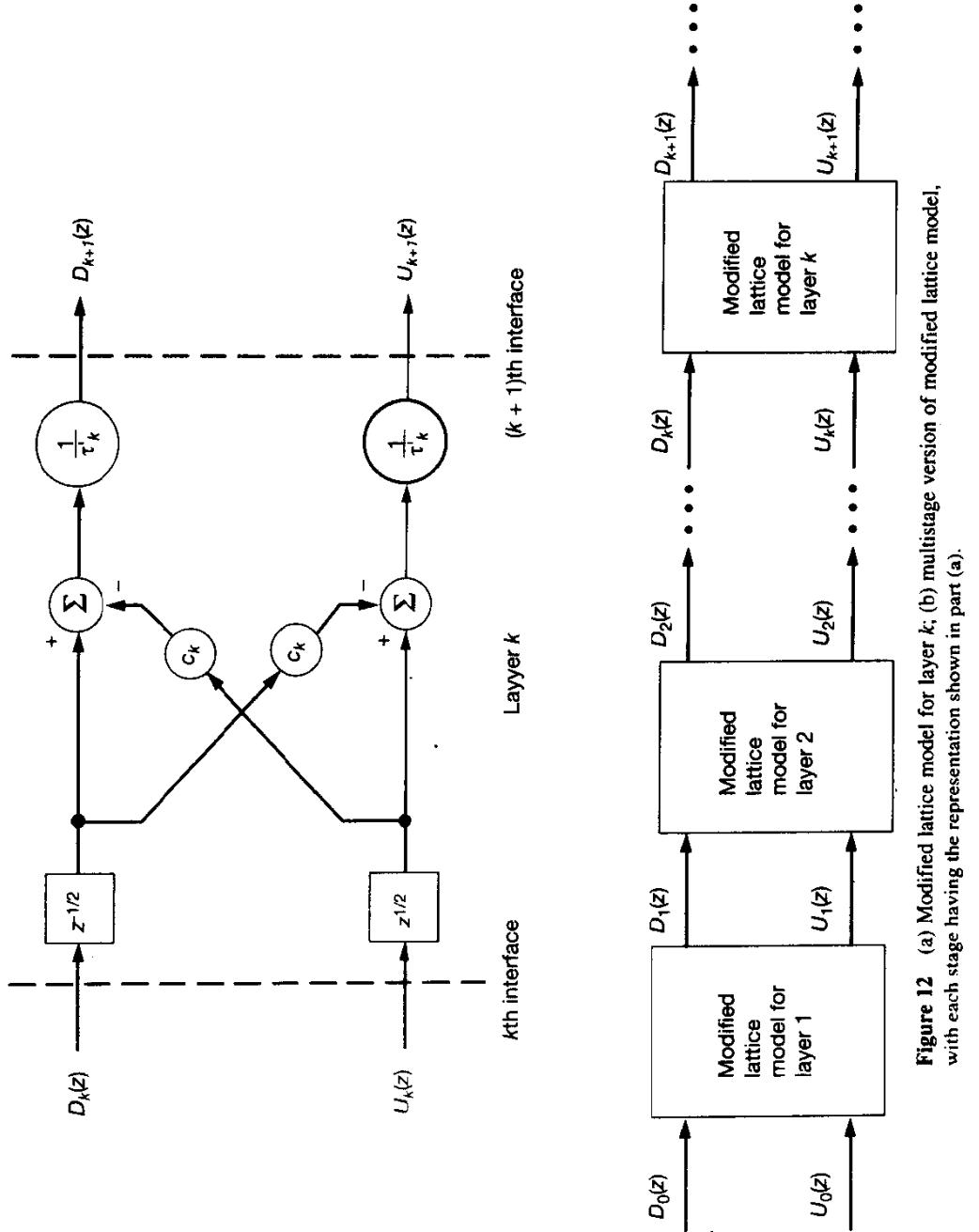


Figure 12 (a) Modified lattice model for layer k ; (b) multistage version of modified lattice model, with each stage having the representation shown in part (a).

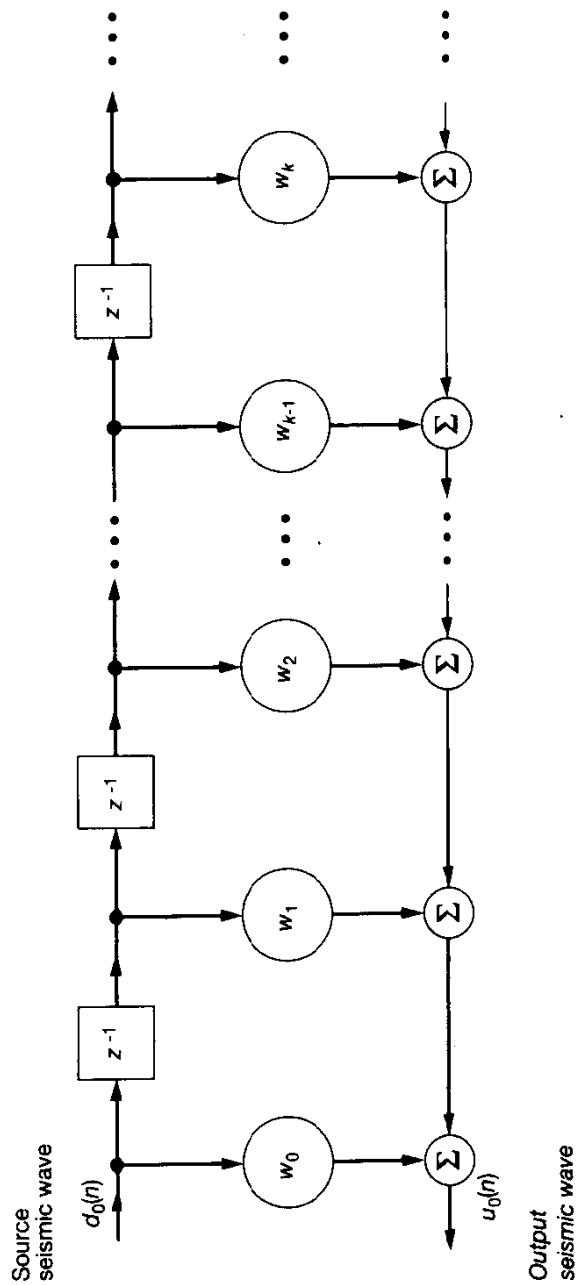


Figure 13 Tapped-delay-line model of layered earth.

where the infinite sequence of tap weights w_n represents the *spatial mapping* of the medium's weighting or the *impulse response* of the medium. Equation (19) states that the output $u_0(n)$ is an infinite series of time-delayed and scaled replicas of the input $d_0(n)$.

There is a *one-to-one correspondence* between the impulse response w_n that characterizes the tapped-delay-line model of Fig. 13 and the sequence of reflection coefficients c_n that characterizes the lattice model of Fig. 11:

$$\{w_n\} \Leftrightarrow \{c_n\} \quad (20)$$

In other words, given the c_n we may uniquely determine the w_n , and vice versa.

In reflection seismology, the model of Fig. 13 is referred to as the *convolutional model*, in view of the convolution of the impulse response of the medium with the input. This model is the starting point of seismic deconvolution (described in the next application).

Parameter Estimation.⁹ The seismic wave $d_0(n)$ generated by the source of energy acts as a "probing" wave that is transmitted into the earth. Correspondingly, the seismic wave $u_0(n)$ is the output evoked by the propagation of $d_0(n)$ in the earth's subsurface. A *recorded* trace of the output $u_0(n)$ for varying time n is called a *seismogram*. Thus, given digital recordings of the probing wave $d_0(n)$ and the resulting seismogram $u_0(n)$, we may apply an adaptive filtering algorithm to estimate the impulse response w_n of the layered earth. This computation is performed *off-line*, with the probing wave $d_0(n)$ used as input to the adaptive filtering algorithm and the seismogram $u_0(n)$ serving the role of desired response for the algorithm.

Predictive Deconvolution

Convolution is fundamental to the analysis of linear time-invariant systems. Specifically, the output of a linear time-invariant system is the convolution of the input with the impulse response of the system. Convolution is *commutative*. We may therefore also say that the output of the system is the convolution of the impulse response of the system with the input. Moreover, convolution is a linear operation; it therefore holds regardless of the type of signal used as the system input.

Consider the convolutional model for reflection seismology depicted in Fig. 13. We may express the input-output relation of this model simply as

$$u_0(n) = w_n * d_0(n) \quad (21)$$

⁹For a survey of different parameter estimation procedures applicable to reflection seismology, see Mendel (1986). This paper also discusses other related issues, namely, *representation* (i.e., how something should be modeled), *measurement* (which physical parameters should be measured and how they should be measured), and *validation* (i.e., demonstrating confidence in the model). For a deterministic approach applicable to reflection seismology, see Bruckstein and Kailath (1987). The approach taken here is based on an *inverse scattering* framework for determining the parameters of a layered wave propagation medium from measurements taken at the boundary.

where $d_0(n)$ is the input, w_n is the impulse response, and $u_0(n)$ is the output. The symbol * is shorthand for convolution. The important point to note here is that given the values of w_n and $d_0(n)$ for varying n , we may determine the corresponding values of $u_0(n)$.

Deconvolution is a linear operation that removes the effect of some previous convolution performed on a given data record (time series). Suppose that we are given the input $d_0(n)$ and the output $u_0(n)$. We may then use deconvolution to determine the impulse response w_n . In symbolic form we may thus write

$$w_n = u_0(n) * d_0^{-1}(n) \quad (22)$$

where $d_0^{-1}(n)$ denotes the *inverse* of $d_0(n)$. Note, however, that $d_0^{-1}(n)$ is *not* the reciprocal of $d_0(n)$; rather, the use of the superscript -1 is merely a flag indicating “inverse.”

In *seismic deconvolution*, we are given the seismogram $u_0(n)$ and the requirement is to unravel it so as to obtain an estimate of the impulse response w_n of a layered earth model. The problem, however, is complicated by the fact that in the general case of reflection seismology we do not have an estimate of the input seismic wave (also referred to as the seismic wavelet) $d_0(n)$. To overcome this practical uncertainty, we may use an elegant statistical procedure known as *predictive deconvolution* (Robinson, 1954; Robinson and Durrani, 1986). The term “predictive” arises from the fact that the procedure relies on the use of linear prediction. The derivation of predictive deconvolution rests on two simplifying hypotheses for seismic wave propagation with normal incidence:

1. The input wave $d_0(n)$, generated by the source of seismic energy, is the *impulse response of an all-pole feedback system*, and is thus minimum phase.
2. The impulse response w_n of the layered earth model has the properties of a *white-noise process*.

Condition 1 is referred to as the *feedback hypothesis*, and condition 2 is referred to as the *random hypothesis*. Geophysical experience over three decades has shown that it is indeed possible to satisfy these two hypotheses (Robinson, 1984). As a result, predictive deconvolution is used routinely on all seismic records in every exploration program.

The implication of the feedback hypothesis is that we may express the present value $d_0(n)$ of the input wave as a *linear combination of the past values*, as shown by

$$d_0(n) = -\sum_{k=1}^M a_k d_0(n-k) \quad (23)$$

where the a_k are the *feedback coefficients*, and M is the *order* of the all-pole feedback system. The order M may be fixed in advance; alternatively, it may be determined by a mean-square-error criterion.

According to the random hypothesis, the impulse response w_n has the properties of a white-noise process. We therefore expect the estimate \hat{w}_n produced by the deconvolution filter in Fig. 14 to have similar properties. In other words, the deconvolution filter acts as a *whitening filter*. Furthermore, the deconvolution filter is an *all-zero* filter with a transfer

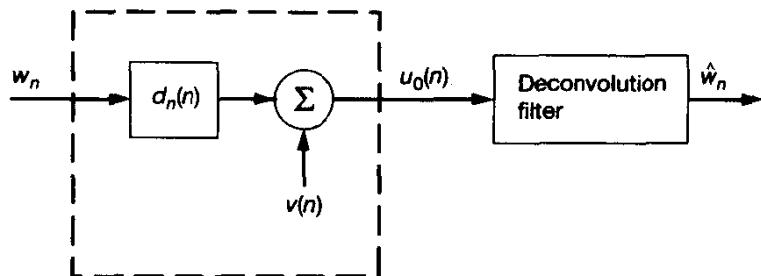


Figure 14 Block diagram illustrating seismic deconvolution.

function equal to the reciprocal of the transfer function of the all-pole feedback system used to model $d_0(n)$. This means that if we express the transfer function of the feedback system [i.e., the z -transform of $d_0(n)$] as:

$$D_0(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}} \quad (24)$$

where a_1, a_2, \dots, a_M are the *feedback coefficients*, and ignore the additive noise $v(n)$ in the model of Fig. 14, then the transfer function of the deconvolution filter is

$$\begin{aligned} A(z) &= \frac{1}{D_0(z)} \\ &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M} \end{aligned} \quad (25)$$

To evaluate $A(z)$, we may use a block processing approach based on the *augmented matrix form of the Wiener-Hopf equations for linear prediction*. This relation consists of a system of $(M + 1)$ simultaneous equations that involve the following:

1. A set of $(M + 1)$ known quantities represented by the *estimates* $\hat{r}(0), \hat{r}(1), \dots, \hat{r}(M)$ of the autocorrelation function of the seismogram $u_0(n)$ for varying lags $0, 1, \dots, M$, respectively. To get these values, we may use the formula for a *biased estimate* of the autocorrelation function:

$$\hat{r}(l) = \frac{1}{N} \sum_{n=l+1}^N d_0(n)d_0(n-l), \quad l = 0, 1, \dots, M \quad (26)$$

where N is the *record length* of the seismogram. Typically, N is very large compared to M .

2. A set of $(M + 1)$ unknowns, made up of the feedback coefficients a_1, a_2, \dots, a_M and the variance σ^2 of the white-noise process assumed to model w_n .

Given the seismogram $u_0(n)$, we may therefore uniquely determine the feedback coefficients a_1, a_2, \dots, a_M and the variance σ^2 by solving this system of equations.

From Eq. (25), we see that the impulse response of the deconvolution filter consists of the sequence a_k , $k = 1, 2, \dots, M$. Accordingly, the convolution of this impulse response with $u_0(n)$ yields the desired estimate \hat{w}_n , as shown by (see Fig. 14)

$$\hat{w}_n = \sum_{k=0}^M a_k u_0(n-k) \quad (27)$$

where $a_0 = 1$. Equation (27) is a description of the deconvolution process. Note, however, the wave $d_0(n)$ generated by the source of seismic energy does not enter this description directly as in the idealized representation of Eq. (23). Rather, the physical nature of $d_0(n)$ influences the deconvolution process by modeling $d_0(n)$ as the impulse response of an all-pole feedback system.

An alternative procedure for constructing the deconvolution filter is to use an adaptive filtering algorithm, as illustrated in Fig. 15. In this application, the present value $u_0(n)$ of the seismic output serves the purpose of a desired response for the algorithm, and the past values $u_0(n-1), u_0(n-2), \dots, u_0(n-M)$ are used as elements of the input vector. The prediction error controls the adaptation of the M tap weights of the transversal filter component of the algorithm. When the algorithm has converged, the tap weights of the transversal filter provide estimates of the feedback coefficients a_1, a_2, \dots, a_M .

Adaptive Equalization

In digital communications a considerable effort has been devoted to the study of data-transmission systems that utilize the available channel bandwidth efficiently. The objective here is to design a system that accommodates the highest possible rate of data transmission, subject to a specified reliability that is usually measured in terms of the error rate or average probability of symbol error. The transmission of digital data through a linear communication channel is limited by two factors:

1. *Intersymbol interference (ISI)*. This is caused by dispersion in the transmit filter, the transmission medium, and the receive filter.
2. *Thermal noise*. This is generated by the receiver at its front end.

For bandwidth-limited channels (e.g., voice-grade telephone channels), we usually find that intersymbol interference is the chief determining factor in the design of high-data-rate transmission systems.

Figure 16 shows the equivalent baseband model of a binary *pulse-amplitude modulation (PAM)* system. The signal applied to the input of the transmitter part of the system consists of a *binary data sequence* b_k , in which each symbol consists of 1 or 0. This sequence is applied to a pulse generator, the output of which is filtered first in the transmitter, then by the medium, and finally in the receiver. Let $u(k)$ denote the sampled output of the receive filter in Fig. 16; the sampling is performed in synchronism with the pulse generator in the transmitter. This output is compared to a *threshold* by means of a *decision device*. If the threshold is exceeded, the receiver makes a decision in favor of symbol 1.

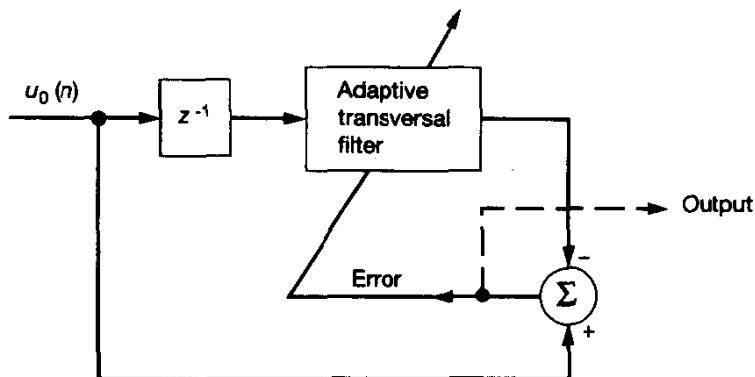


Figure 15 Adaptive filtering scheme for estimating the impulse response of the deconvolution filter.

Otherwise, it decides in favor of symbol 0.

Let a scaling factor a_k be defined by

$$a_k = \begin{cases} +1 & \text{if the input bit } b_k \text{ consists of symbol 1} \\ -1 & \text{if the input bit } b_k \text{ consists of symbol 0} \end{cases} \quad (28)$$

Then, in the absence of thermal noise, we may express $u(k)$ as

$$\begin{aligned} u(k) &= \sum_n a_n p(k-n) \\ &= a_k p(0) + \sum_{n \neq k} a_n p(k-n) \end{aligned} \quad (29)$$

where $p(n)$ is the sampled version of the impulse response of the cascade connection of the transmit filter, the transmission medium, and the receive filter. The first term on the right-hand side of Eq. (29) defines the desired symbol, whereas the remaining series represents the intersymbol interference caused by the *channel* (i.e., the combination of the transmit filter, the medium, and the receive filter). This intersymbol interference, if left unchecked, can result in erroneous decisions when the sampled signal at the channel output is compared with some preassigned threshold by means of a decision device.

To overcome the intersymbol interference problem, control of the time-sampled function $p(n)$ is required. In principle, if the characteristics of the transmission medium are known precisely, then it is virtually always possible to design a pair of transmit and receive filters that will make the effect of intersymbol interference (at sampling times) arbitrarily small. This is achieved by proper shaping of the overall response of the channel in accordance with Nyquist's classic work on telegraph transmission theory. The overall frequency response consists of a *flat portion* and a *roll-off portion* that has a cosine form (Haykin, 1994). Correspondingly, the overall impulse response attains its maximum value at time $n = 0$ and is zero at all other sampling instants; the intersymbol interference is therefore zero. In practice we find that the channel is *time varying*, due to variations in the transmission medium, which makes the received signal *nonstationary*. Accordingly, the

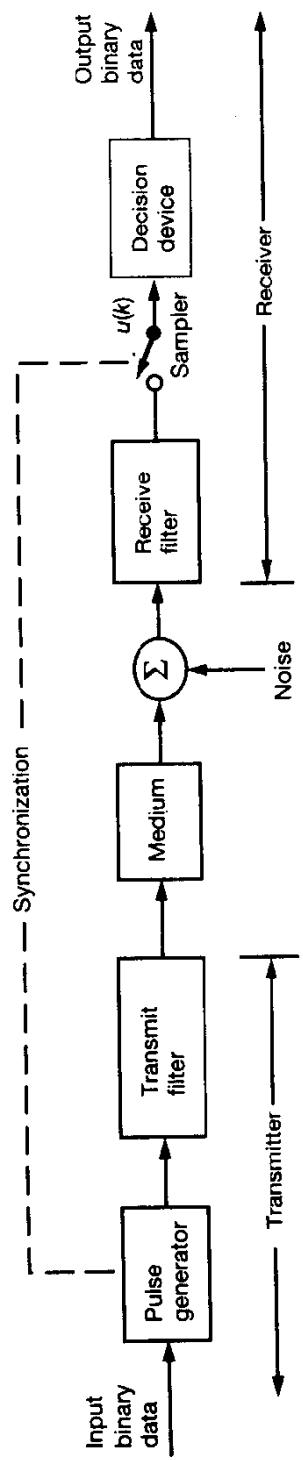


Figure 16 Block diagram of a baseband data transmission system (without equalization).

use of a fixed pair of transmit and receive filters, designed on the basis of average channel characteristics, may not adequately reduce intersymbol interference. This suggests the need for an *adaptive equalizer* that provides precise control over the time response of the channel (Lucky, 1965, 1966; Lucky et al., 1968; Proakis, 1975; Qureshi, 1985).

Among the basic philosophies for equalization of datatransmission systems are pre-equalization at the transmitter and postequalization at the receiver. Since the former technique requires the use of a feedback path, we will only consider equalization at the receiver, where the adaptive equalizer is placed after the receive filter-sampler combination in Fig. 16. In theory, the effect of intersymbol interference may be made arbitrarily small by making the number of adjustable coefficients (tap weights) in the adaptive equalizer infinitely large.

An adaptive filtering algorithm requires knowledge of the "desired" response so as to form the error signal needed for the adaptive process to function. In theory, the transmitted sequence (originating at the transmitter output) is the "desired" response for adaptive equalization. In practice, however, with the adaptive equalizer located in the receiver, the equalizer is physically separated from the origin of its ideal desired response. There are two methods in which a *replica (facsimile)* of the desired response may be generated locally in the receiver:

1. *Training method.* In the first method, a replica of the desired response is *stored* in the receiver. Naturally, the generator of this stored reference has to be electronically *synchronized* with the known transmitted sequence. A widely used *test (probing) signal* consists of a *pseudonoise (PN) sequence* (also known as a *maximal-length sequence*) with a broad and even power spectrum. The PN sequence has noiselike properties. Yet it has a deterministic waveform that repeats periodically. For the generation of a PN sequence, we may use a *linear feedback shift register* that consists of a number of consecutive two-state memory stages (flip-flops) regulated by a single timing clock (Golomb, 1964). A *feedback term*, consisting of the modulo-2 sum of the outputs of various memory stages, is applied to the first memory stage of the shift register and thereby prevents it from emptying.
2. *Decision-directed method.* Under normal operating conditions, a good facsimile of the transmitted sequence is being produced at the output of the *decision device* in the receiver. Accordingly, if this output were the correct transmitted sequence, it may be used as the "desired" response for the purpose of adaptive equalization. Such a method of learning is said to be *decision directed*, because the receiver attempts to learn by employing its own decisions (Lucky et al., 1968). If the average probability of symbol error is small (less than 10 percent, say), the decisions made by the receiver are correct enough for the *estimates of the error signal* (used in the adaptive process) to be *accurate most of the time*. This means that, in general, the adaptive equalizer is able to improve the tap-weight settings by virtue of the correlation procedure built into its feedback control loop. The improved tap-weight settings will, in turn, result in a lower average probability of

symbol error and therefore more accurate estimates of the error signal for adaptation, and so it goes on. However, it is also possible for the reverse effect to occur, in which case the tap-weight settings of the equalizer lose acquisition of the channel.

With a known training sequence, as in the first method, the adaptive filtering algorithm used to adjust the equalizer coefficients corresponds mathematically to searching for the unique minimum of a quadratic error-performance surface. The *unimodal* nature of this surface assures convergence of the algorithm. In the decision-directed method, on the other hand, the use of estimated and unreliable data modifies the error performance into a *multimodal* one, in which case complex behavior may result (Mazo, 1980). Specifically, the error performance surface now exhibits two types of local minima:

1. *Desired local minima*, whose positions correspond to coefficient (tap-weight) settings that yield the same performance as that obtained with a known training sequence
2. *Undesired (extraneous) local minima*, whose positions correspond to coefficient settings that yield inferior equalizer performance.

A poor choice of the initial coefficient settings may cause the adaptive equalizer to converge to an undesirable local minimum and stay there. The most significant point to note from this discussion is that, in general, a *linear* adaptive equalizer must be trained before it is switched to the decision-directed mode of operation if we are to be sure of delivering high performance.

A final comment pertaining to performance evaluation is in order. A popular experimental technique for assessing the performance of a data transmission system involves the use of an *eye pattern*. This pattern is obtained by applying (1) the received wave to the vertical deflection plates of an oscilloscope, and (2) a sawtooth wave at the transmitted symbol rate to the horizontal deflection plates. The resulting display is called an *eye pattern* because of its resemblance to the human eye for binary data. Thus, in a system using adaptive equalization, the equalizer attempts to correct for intersymbol interference in the system and thereby open the eye pattern as far as possible.

Thus far we have only discussed adaptive equalizers for baseband PAM systems. However, voice-band data transmission systems employ modulation-demodulation schemes that are commonly known as *modems*. Depending on the speed of operation, we may categorize modems as follows (Qureshi, 1985):

1. *Low-speed* (2400 to 4800 b/s) modems that use *phase-shift keying (PSK)*; PSK is a digital modulation scheme in which the phase of a sinusoidal carrier wave is shifted by $2\pi k/M$ radians in accordance with the input data, where M is the number of phase levels used and $k = 0, 1, \dots, M - 1$. Specific values of M used in practice are $M = 2$ and 4 , representing *binary phase-shift keying (BPSK)* and *quadrature-phase-shift keying (QPSK)*, respectively.

2. *High-speed* (4800 to 16,800 b/s or possibly even higher) modems that use combined *amplitude and phase modulation* or, equivalently, *quadrature amplitude modulation (QAM)*.

The important point to note is that the baseband model for BPSK is real, whereas the baseband models for QPSK and QAM are complex, involving both in-phase and quadrature channels. Hence, the baseband adaptive equalizer for data transmission systems using BPSK (or its variation) is *real*, whereas the baseband adaptive equalizers for QPSK and QAM are *complex* (i.e., the tap weights of the transversal filter are complex). Note also that a real equalizer processes real inputs to produce a real equalized output, whereas a complex equalizer processes complex inputs to produce complex equalized outputs.

Blind Equalization

In the case of a highly nonstationary communications environment (e.g., digital mobile communications), it is impractical to consider the use of a training sequence. In such a situation, the adaptive filter has to equalize the communication channel in a self-organized (unsupervised) manner, and the resulting operation is referred to as *blind equalization*. Clearly, the design of a blind equalizer is a more challenging task than a conventional adaptive equalizer, because it has to make up for the absence of a training sequence by some practical means. Whereas a conventional adaptive equalizer relies on second-order statistics of the input data, a blind equalizer relies on additional information about the environment.

This additional information may take one of two basic forms:

- *Higher-order statistics (HOS)*, the extraction of which is implicitly or explicitly built into the design of the blind equalizer. For this to be possible, the input data must be non-Gaussian, and the equalizer must include some form of nonlinearity.
- *Cyclostationarity*; which arises when the amplitude, phase, or frequency of a sinusoidal carrier is varied in accordance with an information-bearing signal. In this case, design of the blind equalizer is based on second-order cyclostationary statistics of the input data, and the use of nonlinearity is no longer a requirement.

An advantage of the latter type of blind equalizer is that it exhibits better convergence properties than an HOS-based blind equalizer.

Linear Predictive Coding

The coders used for the digital representation of speech signals fall into two broad classes: *source coders* and *waveform coders*. Source coders are *model dependent*, in that they use *a priori* knowledge about how the speech signal is generated at the source. Source coders for speech are generally referred to as *vocoders* (a contraction of voice coders). They can operate at low coding rates; however, they provide a synthetic quality, with the speech signal having lost substantial naturalness. Waveform coders, on the other hand, essentially

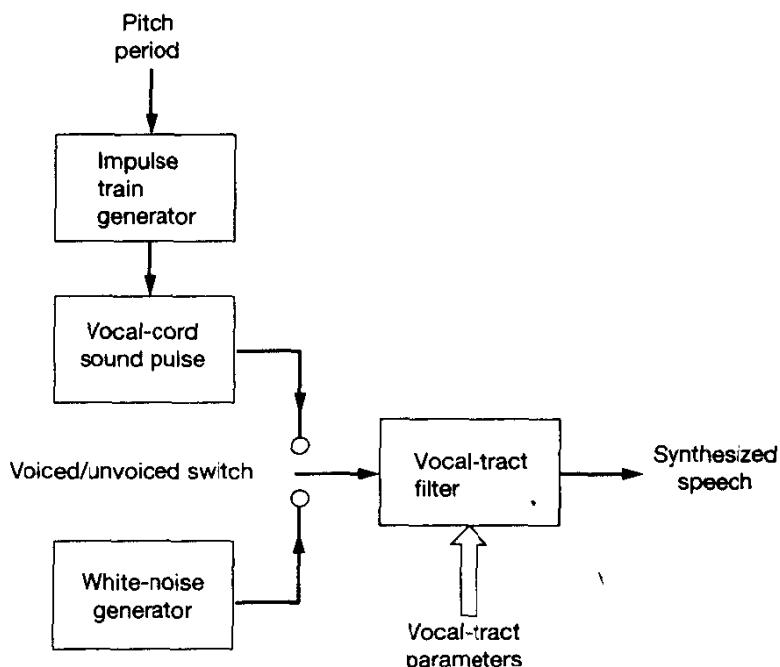


Figure 17 Block diagram of simplified model for the speech production process.

strive for facsimile reproduction of the speech waveform. In principle, these coders are *signal independent*. They may be designed to provide telephone-toll quality for speech at relatively high coding rates. In this subsection we describe a special form of source coder known as a linear predictive coder. Waveform coders are considered in the next subsection.

In the context of speech, *linear predictive coding* (LPC) strives to produce digitized voice data at low bit rates (as low as 2.4 kb/s), with two important motivations in mind. First, the use of linear predictive coding permits the transmission of digitized voice over a *narrow-band channel* (having a bandwidth of approximately 3 kHz). Second, the realization of a low-bit rate makes the *encryption* of voice signals easier and more reliable than would be the case otherwise; encryption is an essential requirement for *secure communications* (as in a military environment). Note that a bit rate of 2.4 kb/s is less than 5 percent of the 64 kb/s used typically for the standard pulse-code modulation (PCM); see the next subsection.

Linear predictive coding achieves a low bit rate for the digital representation of speech by exploiting the special properties of a classical model of the speech production process, which is described next.

Figure 17 shows a simplified block diagram of the classical model for the speech production process. It assumes that the sound-generating mechanism (i.e., the source of excitation) is linearly separable from the intelligence-modulating vocal-tract filter. The precise form of the excitation depends on whether the speech sound is voiced or unvoiced:

1. A *voiced* speech sound (such as ¹⁰/i/ in *eve*) is generated from quasi-periodic vocal-cord sound. In the model of Fig. 17 the impulse-train generator produces a sequence of impulses (i.e., very short pulses), which are spaced by a fundamental period equal to the *pitch period*. This signal, in turn, excites a linear filter whose impulse response equals the vocal-cord sound pulse.
2. An *unvoiced* speech sound (such as /f/ in *fish*) is generated from random sound produced by turbulent airflow. In this case the excitation consists simply of a *white* (i.e., broad spectrum) noise source. The probability distribution of the noise samples does not appear to be critical.

The frequency response of the vocal-tract filter for unvoiced speech or that of the vocal tract multiplied by the spectrum of the vocal-cord sound pulses determines the short-time spectral envelope of the speech signal.

At first sight, it may appear that the speech production model falls under class I of adaptive filtering application (i.e., identification). In reality, however, this is not so. As may be seen in Fig. 17, there is *no* access to the input signal of the vocal tract.

The method of *linear predictive coding (LPC)* is an example of source coding. This method is important, because it provides not only a powerful technique for the digital transmission of speech at low bit rates but also accurate estimates of basic speech parameters.

The development of LPC relies on the model of Fig. 17 for the speech-production process. The frequency response of the vocal tract for unvoiced speech or that of the vocal tract multiplied by the spectrum of the vocal sound pulse for voiced speech is described by the *transfer function*

$$H(z) = \frac{G}{1 + \sum_{k=1}^M a_k z^{-k}} \quad (30)$$

where G is a gain parameter and z^{-1} is the unit-delay operator. The form of excitation applied to this filter is changed by switching between voiced and unvoiced sounds. Thus, the filter with transfer function $H(z)$ is excited by a sequence of impulses to generate voiced sounds or a white-noise sequence to generate unvoiced sounds. In this application, the input data are real valued; hence the filter coefficients, a_k , are likewise real valued.

In linear predictive coding, as the name implies, linear prediction is used to estimate the speech parameters. Given a set of past samples of a speech signal, $u(n-1)$, $u(n-2)$, ..., $u(n-M)$, a linear prediction of $u(n)$, the present sample value of the signal, is defined by

$$\hat{u}(n) = \sum_{k=1}^M \hat{w}_k u(n-k) \quad (31)$$

¹⁰The symbol // is used to denote the *phoneme*, a basic linguistic unit.

The predictor coefficients, $\hat{w}_1, \hat{w}_2, \dots, \hat{w}_M$, are optimized by minimizing the mean-square value of the prediction error, $e(n)$, defined as the difference between $u(n)$ and $\hat{u}(n)$. The use of the minimum-mean-squared-error criterion for optimizing the predictor may be justified for two basic reasons:

1. If the speech signal satisfies the model described by Eq. (30) and if the mean-square value of the error signal $e(n)$ is minimized, then we find that $e(n)$ equals the excitation $u(n)$ multiplied by the gain parameter G in the model of Fig. 18 and $a_k = -\hat{w}_k$, $k = 1, 2, \dots, M$. Thus, the estimation error $e(n)$ consists of quasi-periodic pulses in the case of voiced sounds or a white-noise sequence in the case of unvoiced sounds. In either case, the estimation error $e(n)$ would be small most of the time.
2. The use of the minimum-mean-squared-error criterion leads to tractable mathematics.

Figure 18 shows the block diagram of an LPC vocoder. It consists of a transmitter and a receiver. The transmitter first applies a *window* (typically 10 to 30 ms long) to the input speech signal, thereby identifying a block of speech samples for processing. This window is short enough for the vocal-tract shape to be nearly stationary, so the parameters of the speech-production model in Fig. 18 may be treated as essentially constant for the duration of the window. The transmitter then analyzes the input speech signal in an adaptive manner, block by block, by performing a linear prediction and pitch detection. Finally, it codes the parameters made up of (1) the set of predictor coefficients, (2) the pitch period, (3) the gain parameter, and (4) the voiced-unvoiced parameter, for transmission over the channel. The receiver performs the inverse operations, by first decoding the incoming parameters. In particular, it computes the values of the predictor coefficients, the pitch period, and the gain parameter, and determines whether the segment of interest represents voiced or unvoiced sound. Finally, the receiver uses these parameters to synthesize the speech signal by utilizing the model of Fig. 17.

Adaptive Differential Pulse-Code Modulation

In *pulse-code modulation*, which is the standard technique for waveform coding, three basic operations are performed on the speech signal. The three operations are *sampling* (time discretization), *quantization* (amplitude discretization), and *coding* (digital representation of discrete amplitudes). The operations of sampling and quantization are designed to preserve the shape of the speech signal. As for coding, it is merely a method of translating a discrete sequence of sample values into a more appropriate form of signal representation.

The rationale for sampling follows from a basic property of all speech signals: they are bandlimited. This means that a speech signal can be sampled in time at a finite rate in accordance with the sampling theorem. For example, commercial telephone networks designed to transmit speech signals occupy a bandwidth from 200 to 3200 Hz. To satisfy

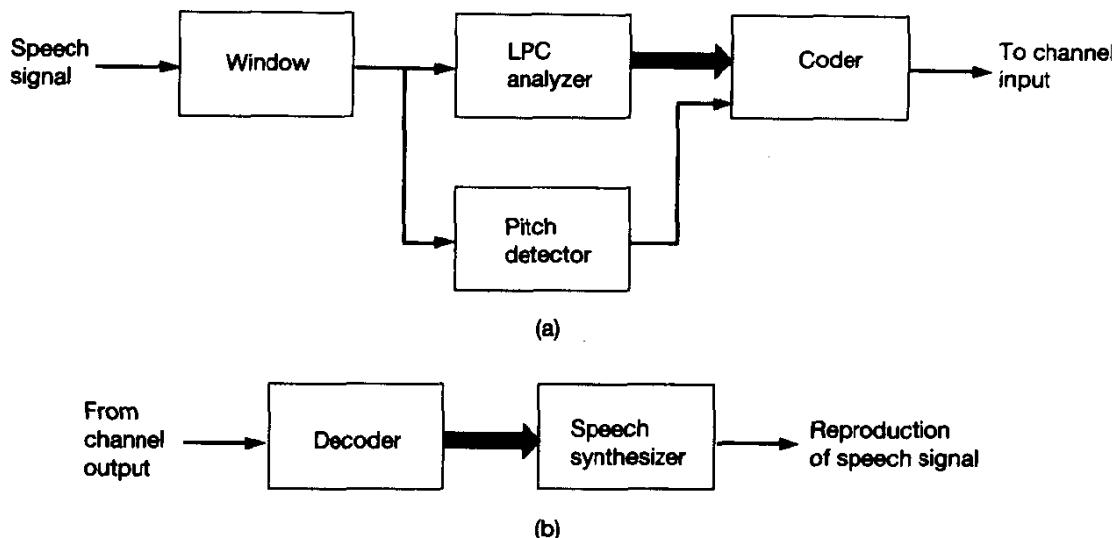


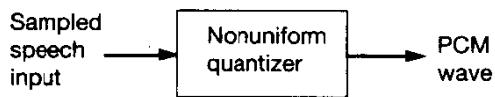
Figure 18 Block diagram of LPC vocoder: (a) transmitter, (b) receiver.

the sampling theorem, a conservative sampling rate of 8 kHz is commonly used in practice.

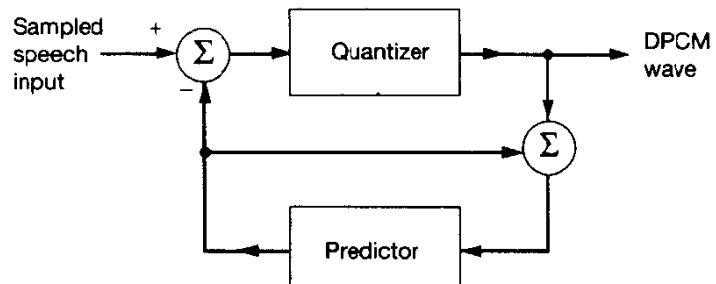
Quantization is justified on the following grounds. Although a speech signal has a continuous range of amplitudes (and therefore its samples also have a continuous amplitude range), it is not necessary to transmit the exact amplitudes of the samples. Basically, the human ear (as ultimate receiver) can only detect finite amplitude differences.

In PCM, as used in telephony, the speech signal (after low-pass filtering) is sampled at the rate of 8 kHz, nonlinearly (e.g., logarithmically) quantized, and then coded into 8-bit words; see Fig. 19(a). The result is a good signal-to-quantization-noise ratio over a wide dynamic range of input signal levels. This method requires a bit rate of 64 kb/s.

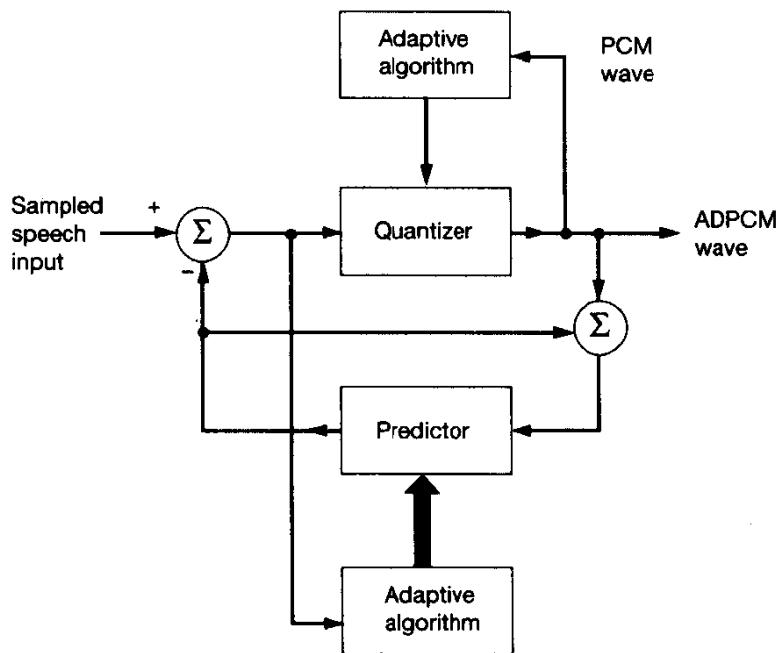
Differential pulse-code modulation (DPCM), another example of waveform coding, involves the use of a predictor as in Fig. 19(b). The predictor is designed to exploit the correlation that exists between adjacent samples of the speech signal, in order to realize a reduction in the number of bits required for the transmission of each sample of the speech signal and yet maintain a prescribed quality of performance. This is achieved by quantizing and then coding the prediction error that results from the subtraction of the predictor output from the input signal. If the prediction is optimized, the variance of the prediction error will be significantly smaller than that of the input signal, so a quantizer with a given number of levels can be adjusted to produce a quantizing error with a smaller variance than would be possible if the input signal were quantized directly as in a standard PCM system. Equivalently, for a quantizing error of prescribed variance, DPCM requires a smaller number of quantizing levels (and therefore a smaller bit rate) than PCM. Differential pulse-code modulation uses a fixed quantizer and a fixed predictor. A further reduction in the transmission rate can be achieved by using an adaptive quantizer together with an adaptive predictor of sufficiently high order, as in Fig. 19(c). This type of waveform coding is called *adaptive differential pulse-code modulation (ADPCM)*, where A denotes



(a)



(b)



(b)

Figure 19 Waveform coders: (a) PCM, (b) DPCM, (c) ADPCM.

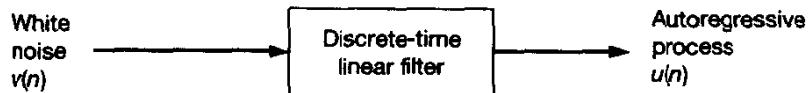


Figure 20 Black box representation of a stochastic model.

adaptation of both quantizer and predictor algorithms. An adaptive predictor is used in order to account for the nonstationary nature of speech signals. ADPCM can digitize speech with toll quality (8-bit PCM quality) at 32 kb/s. It can realize this level of quality with a 4-bit quantizer.¹¹

Adaptive Spectrum Estimation

The *power spectrum* provides a quantitative measure of the second-order statistics of a discrete-time stochastic process as a function of frequency. In *parametric spectrum analysis*, we evaluate the power spectrum of the process by assuming a *model* for the process. In particular, the process is modeled as the output of a linear filter that is excited by a *white-noise process*, as in Fig. 20. By definition, a white-noise process has a constant power spectrum. A model that is of practical utility is the *autoregressive (AR) model*, in which the transfer function of the filter is assumed to consist of poles only. Let this transfer function be denoted by

$$\begin{aligned}
 H(e^{j\omega}) &= \frac{1}{1 + a_1 e^{-j\omega} + \dots + a_M e^{-jM\omega}} \\
 &= \frac{1}{1 + \sum_{k=1}^M a_k e^{-jk\omega}}
 \end{aligned} \tag{32}$$

where the a_k are called the *autoregressive (AR) parameters*, and M is the *model order*. Let σ_v^2 denote the constant power spectrum of the white-noise process $v(n)$ applied to the filter input. Accordingly, the power spectrum of the filter output $u(n)$ equals

$$S_{\text{AR}}(\omega) = \sigma_v^2 |H(e^{j\omega})|^2 \tag{33}$$

We refer to $S_{\text{AR}}(\omega)$ as the *autoregressive (AR) power spectrum*. Equation (32) assumes that the AR process $u(n)$ is real, in which case the AR parameters themselves assume real values.

¹¹The International Telephone and Telegraph Consultative Committee (CCITT) has adopted the 32-kb/s ADPCM as an international standard. The adaptive predictor used herein has a transfer function consisting of two poles and six zeros. A two-pole configuration was chosen, because it permits control of decoder stability in the presence of transmission errors. Six zeros were combined with the two poles in order to improve performance. The eight coefficients of the predictor are adapted by using a simplified version of the LMS algorithm; for details, see Benvenuto et al. (1986) and Nishitani et al. (1987).

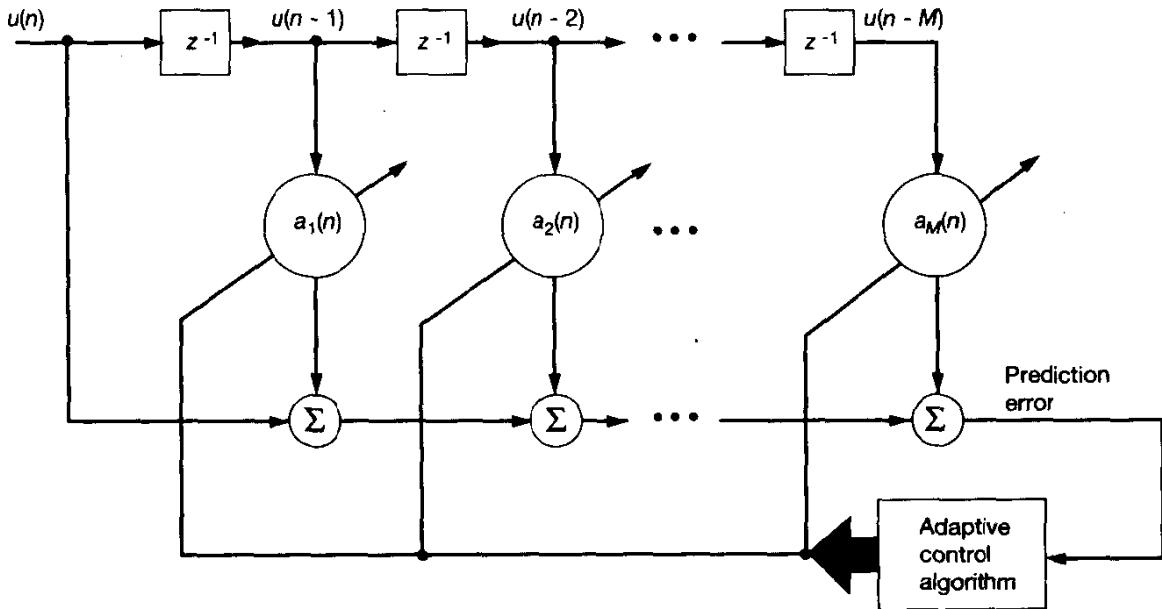


Figure 21 Adaptive prediction-error filter for real-valued data.

When the AR model is time varying, the model parameters become time dependent, as shown by $a_1(n)$, $a_2(n)$, \dots , $a_M(n)$. In this case, we express the power spectrum of the time-varying AR process as

$$S_{\text{AR}}(\omega, n) = \frac{\sigma_v^2}{\left| 1 + \sum_{k=1}^M a_k(n) e^{-j k \omega} \right|^2} \quad (34)$$

We may determine the AR parameters of the time-varying model by applying $u(n)$ to an *adaptive prediction-error filter*, as indicated in Fig. 21. The filter consists of a transversal filter with adjustable tap weights. In the adaptive scheme of Fig. 21, the prediction error produced at the output of the filter is used to control the adjustments applied to the tap weights of the filter.

The *adaptive AR model* provides a practical means for measuring the *instantaneous frequency* of a frequency-modulated process. In particular, we may do this by measuring the frequency at which the AR power spectrum $S_{\text{AR}}(\omega, n)$ attains its peak value for varying time n .

Signal Detection

The *detection problem*, that is, the problem of detecting an information-bearing signal in noise, may be viewed as one of *hypothesis testing* with deep roots in *statistical decision*

theory (Van Trees, 1968): In the statistical formulation of hypothesis testing, there are two criteria of most interest: the *Bayes criterion* and the *Neyman–Pearson criterion*. In the Bayes test, we minimize the *average cost* or *risk* of the experiment of interest, which incorporates two sets of parameters: (1) *a priori probabilities* that represent the observer's information about the source of information before the experiment is conducted, and (2) a set of *costs* assigned to the various possible courses of action. As such, the Bayes criterion is directly applicable to digital communications. In the Neyman–Pearson test, on the other hand, we maximize the *probability of detection* subject to the constraint that the *probability of false alarm* does *not* exceed some preassigned value. Accordingly, the Neyman–Pearson criterion is directly applicable to radar or sonar. An idea of fundamental importance that emerges in hypothesis testing is that, for a Bayes criterion or Neyman–Pearson criterion, the optimum test consists of two distinct operations: (1) processing the observed data to compute a test statistic called the *likelihood ratio*, and (2) computing the likelihood ratio with a *threshold* to make a *decision* in favor of one of the two hypotheses. The choice of one criterion or the other merely affects the value assigned to the threshold. Let H_1 denote the hypothesis that the observed data consist of noise alone, and H_2 denote the hypothesis that the data consist of signal plus noise. The likelihood ratio is defined as the ratio of two maximum likelihood functions, the numerator assuming that hypothesis H_2 is true and the denominator assuming that hypothesis H_1 is true. If the likelihood ratio exceeds the threshold, the decision is made in favor of hypothesis H_2 ; otherwise, the decision is made in favor of hypothesis H_1 .

In simple binary hypothesis testing, it is assumed that the signal is known, and the noise is both white and Gaussian. In this case, the likelihood ratio test yields a *matched filter* (matched in the sense that its impulse response equals the time-reversed version of the known signal). When the additive noise is a *colored Gaussian noise* of known mean and correlation matrix, the likelihood ratio test yields a filter that consists of two sections: a *whitening filter* that transforms the colored noise component at the input into a white Gaussian noise process, and a *matched filter* that is matched to the new version of the known signal as modified by the whitening filter.

However, in some important operational environments such as *communications*, *radar*, and *active sonar*, there may be inadequate information on the signal and noise statistics to design a fixed optimum detector. For example, in a sonar environment it may be difficult to develop a precise *model* for the received sonar signal, one that would account for the following factors completely:

- Loss in the signal strength of a *target echo* from an object of interest (e.g., enemy vessel), due to oceanic propagation effects and reflection loss at the target
- Statistical variations in the additive *reverberation* component, produced by reflections of the transmitted signal from scatterers such as the ocean surface, ocean floor, biologies, and inhomogeneities within the ocean volume
- Potential sources of *noise* such as biological, shipping, oil drilling, seismic, and oceanographic phenomena.

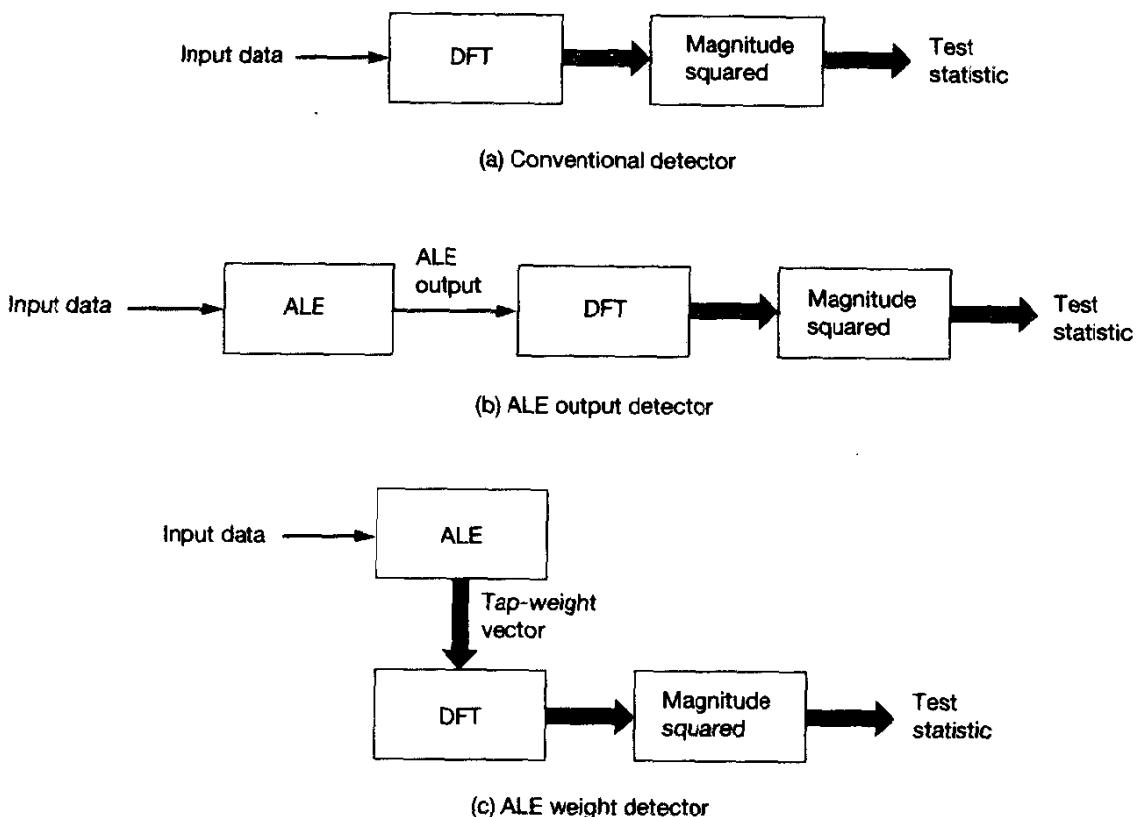


Figure 22 Fixed and adaptive detection schemes: (a) conventional detector. (b) ALE output detector. (c) ALE weight detector.

In situations of this kind, the use of adaptivity offers an attractive approach to solve the target (signal) detection problem. Typically, the design of an *adaptive detector* proceeds by exploiting some knowledge of general characteristics of the signal and noise, and designing the detector in such a way that its internal structure is adjustable in response to changes in the received signal. In general, the incorporation of this adjustment makes the performance analysis of an adaptive detector much more difficult to undertake than that of a fixed detector.

Fixed and adaptive detectors. Figure 22(a) shows the block diagram of a conventional detector based on the *discrete Fourier transform (DFT)* for the detection of narrow-band signals in white Gaussian noise (Williams and Ricker, 1972). The DFT may be viewed as a bank of nonoverlapping narrow-band filters whose passbands span the frequency range of interest. In the detector of Fig. 22(a) the magnitude of each complex output of the DFT is squared to form a *sufficient statistic*. This statistic is optimum (in the Neyman–Pearson sense) for detecting a sinusoid of known frequency (centered in the pertinent passband of the DFT) but unknown phase, and in the presence of white Gaussian noise. The detector output is compared to a threshold. If the threshold is exceeded, the

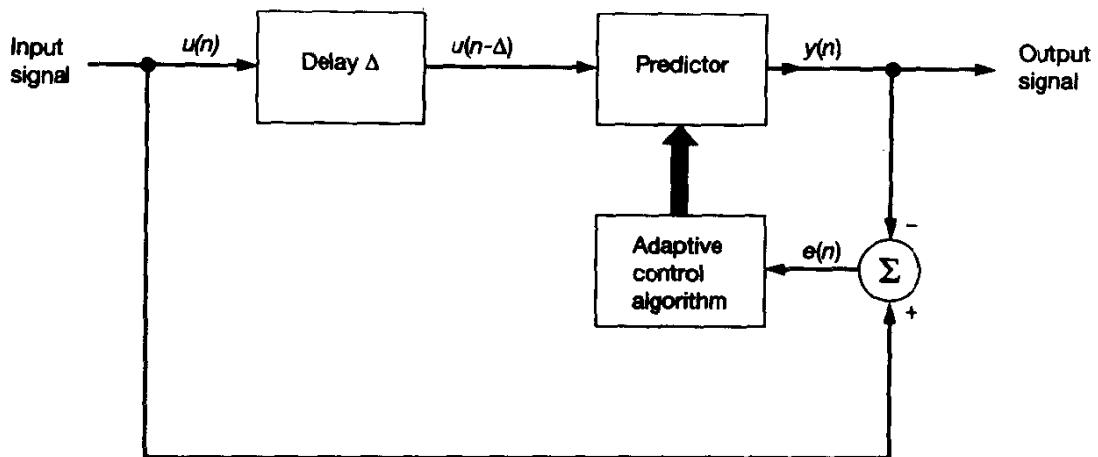


Figure 23 Adaptive line enhancer.

detector decides in favor of the narrow-band signal; otherwise, the detector declares the signal to be absent.

The performance of this conventional noncoherent detector may be improved by using an *adaptive line enhancer* (*ALE*) as a *prefilter* (preprocessor) to the detector (Widrow et al., 1975b). The ALE is a special form of adaptive noise canceler that is designed to suppress the wide-band noise component of the input, while passing the narrow-band signal component with little attenuation. Figure 23 depicts the block diagram of an ALE. It consists of the interconnection of a delay element and a linear predictor. The predictor output $y(n)$ is subtracted from the input signal $u(n)$ to produce the estimation error $e(n)$. This estimation error is, in turn, used to adaptively control the tap weights of the predictor. The predictor input equals $u(n - \Delta)$, where the delay Δ is equal to or greater than the sampling period. The main function of the *prediction depth* Δ is to remove the correlation between the noise component in the original input signal $u(n)$ and the delayed predictor input $u(n - \Delta)$. It is for this reason that the delay Δ is also called the *decorrelation parameter* of the ALE.

Two types of ALE detection structures have been proposed in the literature (Zeidler, 1990):

1. *ALE output detector.* In this adaptive detector shown in Fig. 22(b), the output of an ALE is applied to a DFT. The magnitude of the resulting DFT output is squared to produce the sufficient statistic for the detector.
2. *ALE weight detector.* In this second adaptive detector, shown in Fig. 22(c), the tap-weight vector of an ALE is applied to a DFT. The magnitude of the DFT output is squared as before to produce the sufficient statistic.

In both cases, the ALE processes N input data points, with the ALE length small compared to N . The real benefit of the ALE is realized in a nonstationary noise background (Zeidler, 1990).

The practical value of an ALE as a preprocessor to a conventional matched filter has been demonstrated by Nielson and Thomas (1988) as a means of improving the performance of the detector in the presence of Arctic ocean noise. This type of noise is known to have highly non-Gaussian and nonstationary characteristics; hence the benefit to be gained from the use of an ALE.

Adaptive Noise Canceling

As the name implies, adaptive noise canceling relies on the use of *noise canceling* by subtracting noise from a received signal, an operation controlled in an *adaptive* manner for the purpose of improved signal-to-noise ratio. Ordinarily, it is inadvisable to subtract noise from a received signal, because such an operation could produce disastrous results by causing an increase in the average power of the output noise. However, when proper provisions are made, and filtering and subtraction are controlled by an adaptive process, it is possible to achieve a superior system performance compared to direct filtering of the received signal (Widrow et al., 1975b; Widrow and Stearns, 1985).

Basically, an adaptive noise canceler is a *dual-input, closed-loop adaptive feedback system* as illustrated in Fig. 24. The two inputs of the system are derived from a pair of sensors: a *primary sensor* and a *reference (auxiliary) sensor*. Specifically, we have the following:

1. The primary sensor receives an *information-bearing signal* $s(n)$ corrupted by *additive noise* $v_0(n)$, as shown by

$$d(n) = s(n) + v_0(n) \quad (35)$$

The signal $s(n)$ and the noise $v_0(n)$ are uncorrelated with each other; that is,

$$E[s(n)v_0(n-k)] = 0 \quad \text{for all } k \quad (36)$$

where $s(n)$ and $v_0(n)$ are assumed to be real valued.

2. The reference sensor receives a noise $v_1(n)$ that is *uncorrelated* with the signal $s(n)$ but *correlated* with the noise $v_0(n)$ in the primary sensor output in an *unknown* way; that is,

$$E[s(n)v_1(n-k)] = 0 \quad \text{for all } k \quad (37)$$

and

$$E[v_0(n)v_1(n-k)] = p(k) \quad (38)$$

where, as before, the signals are real valued and $p(k)$ is an *unknown* cross-correlation for lag k .

The reference signal $v_1(n)$ is processed by an adaptive filter to produce the output signal:

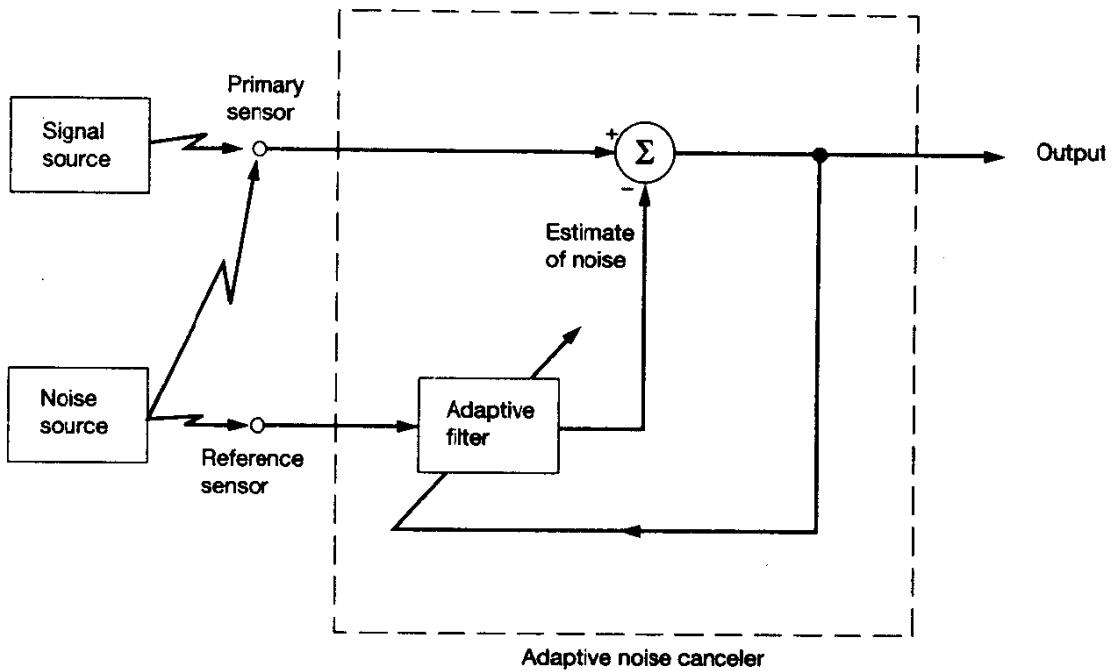


Figure 24 Adaptive noise canceler.

$$y(n) = \sum_{k=0}^{M-1} \hat{w}_k(n) v_1(n-k) \quad (39)$$

where the $\hat{w}_k(n)$ are the adjustable (real) tap weights of the adaptive filter. The filter output $y(n)$ is subtracted from the primary signal $d(n)$, serving as the “desired” response for the adaptive filter. The error signal is defined by

$$e(n) = d(n) - y(n) \quad (40)$$

Thus, substituting Eq. (35) in (40), we get

$$e(n) = s(n) + v_0(n) - y(n) \quad (41)$$

The error signal is, in turn, used to adjust the tap weights of the adaptive filter, and the control loop around the operations of filtering and subtraction is thereby closed. Note that the information-bearing signal $s(n)$ is indeed part of the error signal $e(n)$, as indicated in Eq. (41).

The error signal $e(n)$ constitutes the overall *system output*. From Eq. (41) we see that the noise component in the system output is $v_0(n) - y(n)$. Now, the adaptive filter attempts to minimize the mean-square value (i.e., average power) of the error signal $e(n)$. The information-bearing signal $s(n)$ is essentially unaffected by the adaptive noise canceler.

Hence, minimizing the mean-square value of the error signal $e(n)$ is equivalent to minimizing the mean-square value of the output noise $v_0(n) - y(n)$. With the signal $s(n)$ remaining essentially constant, it follows that *the minimization of the mean-square value of the error signal is indeed the same as the maximization of the output signal-to-noise ratio of the system.*

The signal-processing operation described herein has two limiting cases that are noteworthy:

1. The adaptive filtering operation is *perfect* in the sense that

$$y(n) = v_0(n)$$

In this case, the system output is *noise free* and the noise cancelation is perfect. Correspondingly, the output signal-to-noise ratio is infinitely large.

2. The reference signal $v_1(n)$ is *completely uncorrelated* with both the signal and noise components of the primary signal $d(n)$; that is,

$$E[d(n)v_1(n - k)] = 0 \quad \text{for all } k$$

In this case, the adaptive filter “switches itself off,” resulting in a zero value for the output $y(n)$. Hence, the adaptive noise canceler has *no* effect on the primary signal $d(n)$, and the output signal-to-noise ratio remains unaltered.

The effective use of adaptive noise canceling therefore requires that we place the reference sensor in the noise field of the primary sensor with two specific objectives in mind. First, the information-bearing signal component of the primary sensor output is *undetectable* in the reference sensor output. Second, the reference sensor output is *highly correlated* with the noise component of the primary sensor output. Moreover, the adaptation of the adjustable filter coefficients must be near optimum.

In the remainder of this subsection, we describe three useful applications of the adaptive noise-canceling operation:

1. *Canceling 60-Hz interference in electrocardiography.* In *electrocardiography (ECG)*, commonly used to monitor heart patients, an *electrical discharge* radiates energy through a human *tissue* and the resulting output is received by an *electrode*. The electrode is usually positioned in such a way that the received energy is maximized. Typically, however, the electrical discharge involves very low potentials. Correspondingly, the received energy is very small. Hence extra care has to be exercised in minimizing signal degradation due to external *interference*. By far, the strongest form of interference is that of a 60-Hz periodic waveform picked up by the receiving electrode (acting like an antenna) from nearby electrical equipment (Huhta and Webster, 1973). Needless to say, this interference has undesirable effects in the interpretation of electrocardiograms. Widrow et al. ((1975b)) have demonstrated the use of adaptive noise canceling (based on the LMS algorithm) as a method for reducing this form of interference. Specifically, the primary signal is taken from the ECG preamplifier, and the reference signal is taken from a wall outlet with proper attenuation. Figure 25 shows a block diagram of the adaptive noise canceler used

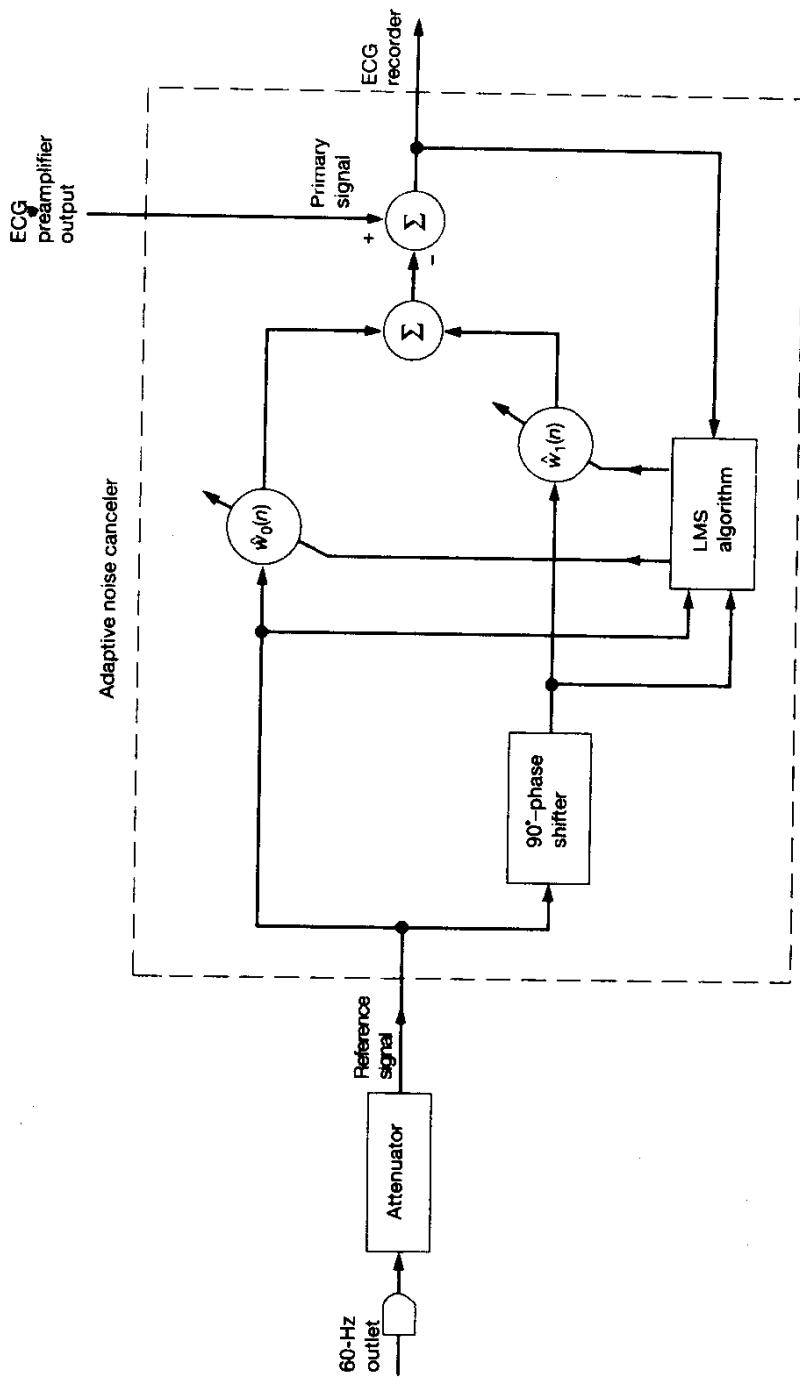


Figure 25 Adaptive noise canceler for suppressing 60-Hz interference in electrocardiography
(After Widrow et al., 1975b).

by Widrow et al. (1975b). The adaptive filter has two adjustable weights, $\hat{w}_0(n)$ and $\hat{w}_1(n)$. One weight, $\hat{w}_0(n)$, is fed directly from the reference point. The second weight, $\hat{w}_1(n)$, is fed from a 90°-phase-shifted version of the reference input. The sum of the two weighted versions of the reference signal is then subtracted from the ECG output to produce an error signal. This error signal together with the weighted inputs are applied to the LMS algorithm, which, in turn, controls the adjustments applied to the two weights. In this application, the adaptive noise canceler acts as a variable “notch filter.” The frequency of the sinusoidal interference in the ECG output is presumably the same as that of the sinusoidal reference signal. However, the amplitude and phase of the sinusoidal interference in the ECG output are unknown. The two weights $\hat{w}_0(n)$ and $\hat{w}_1(n)$ provide the two *degrees of freedom* required to control the amplitude and phase of the sinusoidal reference signal so as to cancel the 60-Hz interference contained in the ECG output.

2. Reduction of acoustic noise in speech. At a noisy site (e.g., the cockpit of a military aircraft), voice communication is affected by the presence of *acoustic noise*. This effect is particularly serious when linear predictive coding (LPC) is used for the digital representation of voice signals at low bit rates; LPC was discussed earlier. To be specific, high-frequency acoustic noise severely affects the estimated LPC spectrum in both the low- and high-frequency regions. Consequently, the intelligibility of digitized speech using LPC often falls below the minimum acceptable level. Kang and Fransen (1987) describe the use of an adaptive noise canceler, based on the LMS algorithm, for reducing acoustic noise in speech. The noise-corrupted speech is used as the primary signal. To provide the reference signal (noise only), a reference microphone is placed in a location where there is sufficient isolation from the source of speech (i.e., the known location of the speaker’s mouth). In the experiments described by Kang and Fransen, a reduction of 10 to 15 dB in the acoustic noise floor is achieved, without degrading voice quality. Such a level of noise reduction is significant in improving voice quality, which may be unacceptable otherwise.

3. Adaptive speech enhancement. Consider the situation depicted in Fig. 26. The requirement is to listen to the voice of the desired speaker in the presence of background noise, which may be satisfied through the use of adaptive noise canceling. Specifically, *reference microphones* are added at locations far enough away from the desired speaker such that their outputs contain *only* noise. As indicated in Fig. 26, a weighted sum of the auxiliary microphone outputs is subtracted from the output of the desired speech-containing microphone, and an adaptive filtering algorithm (e.g., the LMS algorithm) is used to adjust the weights so as to minimize the average output power. A useful application of the idea described herein is in the adaptive noise cancellation for hearing aids¹² (Chazan et al., 1988). The so-called “cocktail party effect” severely limits the usefulness of hearing aids. The cocktail party phenomenon refers to the ability of a person with normal hearing to focus on a conversation taking place at a distant location in a crowded room. This ability

¹²This idea is similar to that of adaptive spatial filtering in the context of antennas, which is considered later in this section.

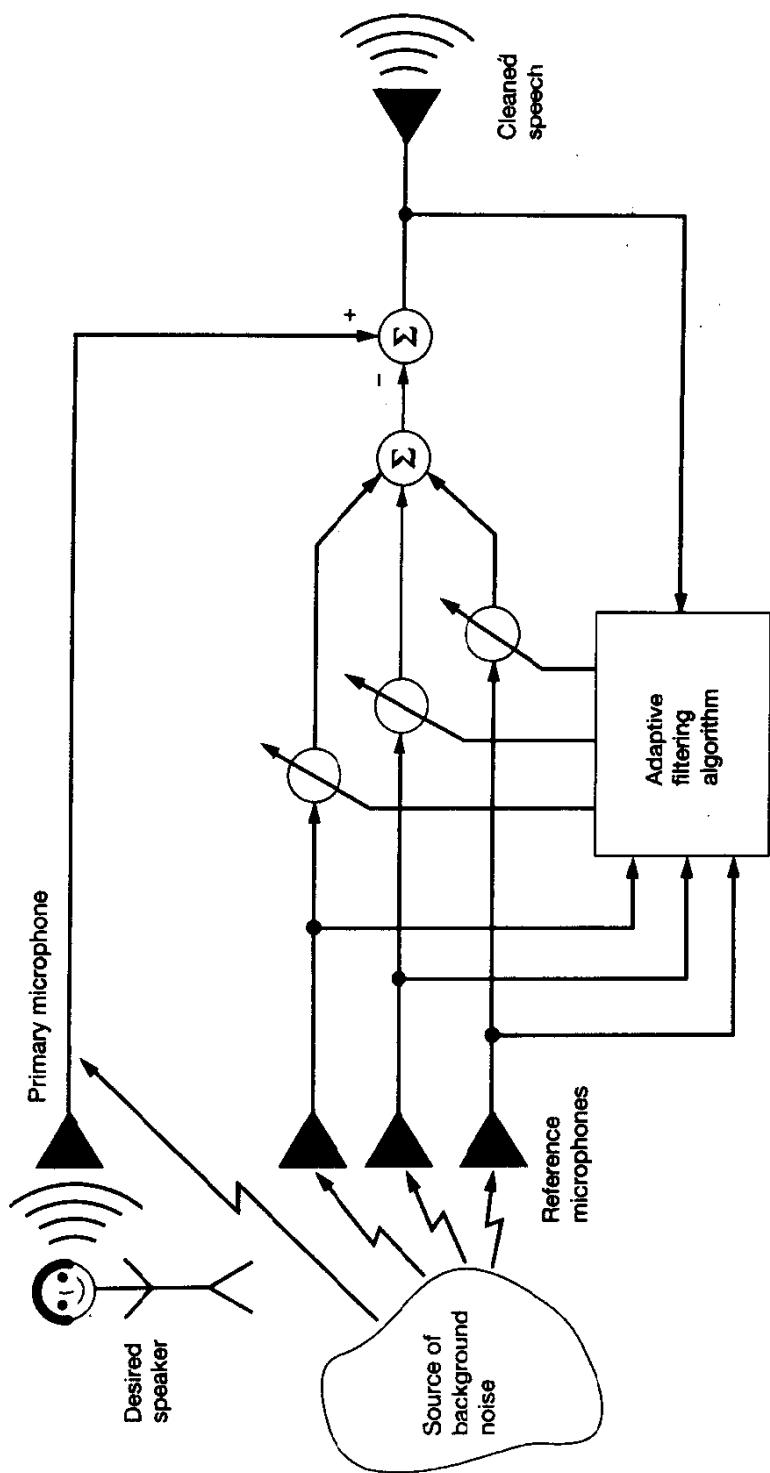


Figure 26 Block diagram of an adaptive noise canceler for speech.

is lacking in a person who wears hearing aids, because of extreme sensitivity to the presence of *background noise*. This sensitivity is attributed to two factors: (a) the loss of directional cues, and (b) the limited channel capacity of the ear caused by the reduction in both dynamic range and frequency response. Chazan et al. (1988) describe an adaptive noise canceling technique aimed at overcoming this problem. The technique involves the use of an *array of microphones* that exploit the difference in spatial characteristics between the desired signal and the noise in a crowded room. The approach taken by Chazan et al. is based on the fact that each microphone output may be viewed as the sum of the signals produced by the individual speakers engaged in conversations in the room. Each signal contribution in a particular microphone output is essentially the result of a speaker's speech signal having passed through the *room filter*. In other words, each speaker (including the desired speaker) produces a signal at the microphone output that is the sum of the direct transmission of his or her speech signal and its reflections from the walls of the room. The requirement is to reconstruct the desired speaker signal, including its room reverberations, while canceling out the source of noise. In general, the transformation undergone by the speech signal from the desired speaker is not known. Also, the characteristics of the background noise are variable. We thus have a signal-processing problem for which adaptive noise canceling offers a feasible solution.

Echo Cancellation

Almost all conversations are conducted in the presence of *echoes*. An echo may be nonnoticeable or distinct, depending on the time delay involved. If the delay between the speech and the echo is short, the echo is not noticeable but perceived as a form of spectral distortion or reverberation. If, on the other hand, the delay exceeds a few tens of milliseconds, the echo is distinctly noticeable. Distinct echoes are annoying.

Echoes may also be experienced on a telephone circuit (Sondhi and Berkley, 1980). When a speech signal encounters an *impedance mismatch* at any point on a telephone circuit, a portion of that signal is reflected (returned) as an echo. An echo represents an *impairment* that can be annoying subjectively as the more obvious impairments of low volume and noise.

To see how echoes occur, consider a long-distance telephone circuit depicted in Fig. 27. Every telephone set in a given geographical area is connected to a central office by a *two-wire line* called the *customer loop*; the two-wire line serves the need for communications in either direction. However, for circuits longer than about 35 miles, a separate path is necessary for each direction of transmission. Accordingly, there has to be provision for connecting the two-wire circuit to the four-wire circuit. This connection is accomplished by means of a *hybrid transformer*, commonly referred to as a *hybrid*. Basically, a hybrid is a bridge circuit with three ports (terminal pairs), as depicted in Fig. 28. If the bridge is *not* perfectly balanced, the "in" port of the hybrid becomes coupled to the "out" port, thereby giving rise to an echo.

Echoes are noticeable when a long-distance call is made on a telephone circuit, particularly one that includes a *geostationary satellite*. Due to the high altitude of such a satellite, there is a one-way travel time of about 300 ms between a ground station and the

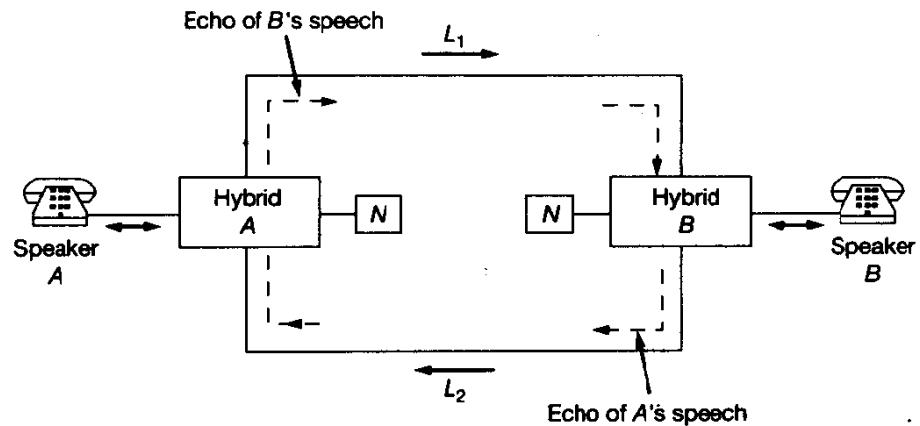


Figure 27 Long-distance telephone circuit; the boxes marked N are balancing impedances.

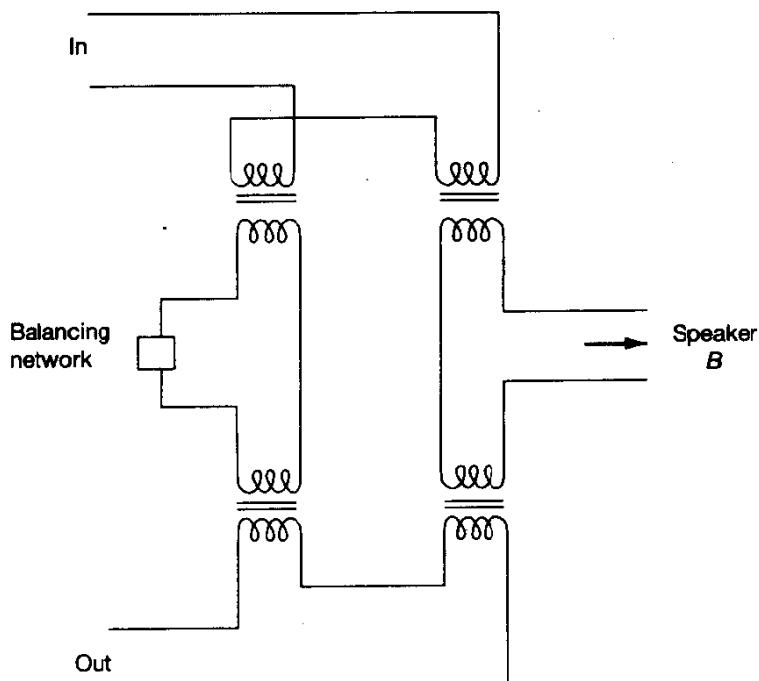


Figure 28 Hybrid circuit.

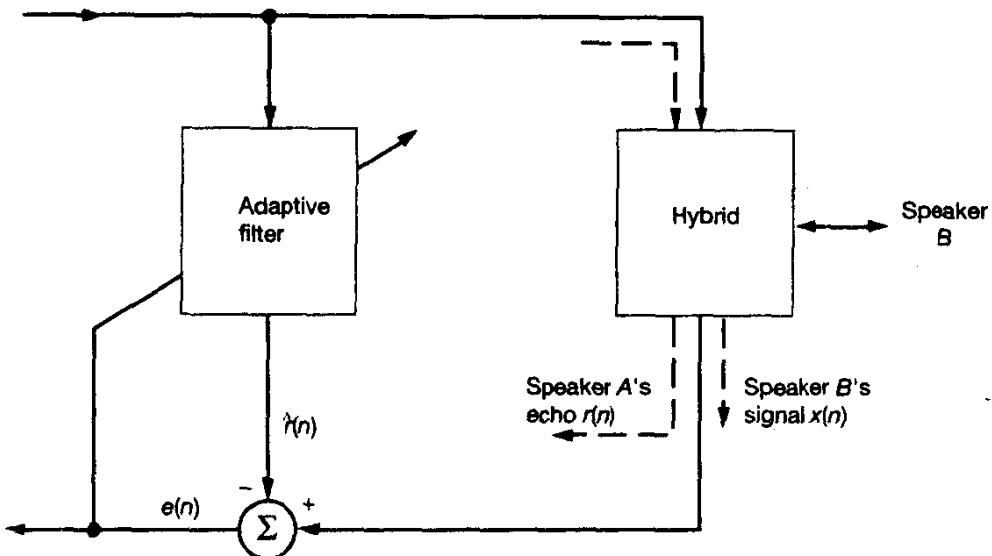


Figure 29 Signal definitions for echo cancellation.

satellite. Thus, the *round-trip delay* in a satellite link (including telephone circuits) can be as long as 600 ms. Generally speaking, the longer the echo delay, the more it must be attenuated before it becomes noticeable.

The question to be answered is: How do we exercise echo control? It appears that the idea with the greatest potential for echo control is that of *adaptive echo cancellation* (Sondhi and Prasti, 1966; Sondhi, 1967; Sondhi and Berkley, 1980; Messerschmitt, 1984; Murano et al., 1990). The basic principle of echo cancellation is to *synthesize a replica of the echo and subtract it from the returned signal*. This principle is illustrated in Fig. 29 for only one direction of transmission (from speaker A on the far left of the hybrid to speaker B on the right). The adaptive canceler is placed in the four-wire path *near the origin of the echo*. The synthetic echo, denoted by $\hat{r}(n)$, is generated by passing the speech signal from speaker A (i.e., the "reference" signal for the adaptive canceler) through an adaptive filter that ideally matches the transfer function of the echo path. The reference signal, passing through the hybrid, results in the echo signal $r(n)$. This echo, together with a near-end talker signal $x(n)$ (i.e., the speech signal from speaker B) constitutes the "desired" response for the adaptive canceler. The synthetic echo $\hat{r}(n)$ is subtracted from the desired response $r(n) + x(n)$ to yield the canceler error signal

$$e(n) = r(n) - \hat{r}(n) + x(n) \quad (42)$$

Note that the error signal $e(n)$ also contains the near-end talker signal $x(n)$. In any event, the error signal $e(n)$ is used to control the adjustments made in the coefficients (tap weights) of the adaptive filter. In practice, the echo path is highly variable, depending on the distance to the hybrid, the characteristics of the two-wire circuit, and so on. These variations are taken care of by the adaptive control loop built into the canceler. The control loop continuously adapts the filter coefficients to take care of fluctuations in the echo path.

For the adaptive echo cancelation circuit to operate satisfactorily, the impulse response of the adaptive filter should have a length greater than the longest echo path that needs to be accommodated. Let T_s be the sampling period of the digitized speech signal, M be the number of adjustable coefficients (tap weights) in the adaptive filter, and τ be the longest echo delay to be accommodated. We must then choose

$$MT_s > \tau \quad (43)$$

As mentioned previously (when discussing adaptive differential pulse-code modulation), the sampling rate for speech signals on the telephone network is conservatively chosen as 8 kHz, that is,

$$T_s = 125 \mu\text{s}$$

Suppose, for example, that the echo delay $\tau = 30$ ms. Then we must choose

$$M > 240 \text{ taps}$$

Thus, the use of an echo canceler with $M = 256$ taps, say, is satisfactory for this situation.

Adaptive Beamforming

For our last application, we describe a *spatial* form of adaptive signal processing that finds practical use in radar, sonar, communications, geophysical exploration, astrophysical exploration, and biomedical signal processing.

In the particular type of spatial filtering of interest to us in this book, a number of independent *sensors* are placed at different points in space to "listen" to the received signal. In effect, the sensors provide a means of *sampling* the received signal *in space*. The set of sensor outputs collected at a particular instant of time constitutes a *snapshot*. Thus, a snapshot of data in spatial filtering (for the case when the sensors lie uniformly on a straight line) plays a role analogous to that of a set of consecutive tap inputs that exist in a transversal filter at a particular instant of time.¹³

In radar, the sensors consist of antenna elements (e.g., dipoles, horns, slotted waveguides) that respond to incident electromagnetic waves. In sonar, the sensors consist of hydrophones designed to respond to acoustic waves. In any event, spatial filtering, known as *beamforming*, is used in these systems to distinguish between the spatial properties of signal and noise. The device used to do the beamforming is called a *beamformer*. The term "beamformer" is derived from the fact that the early forms of antennas (spatial filters) were designed to form *pencil beams*, so as to receive a signal radiating from a specific direction and attenuate signals radiating from other directions of no interest (Van Veen and Buckley, 1988). Note that the beamforming applies to the radiation (transmission) or reception of energy.

¹³For a discussion of the analogies between time- and space-domain forms of signal processing, see Bracewell (1986) and Van Veen and Buckley (1988).

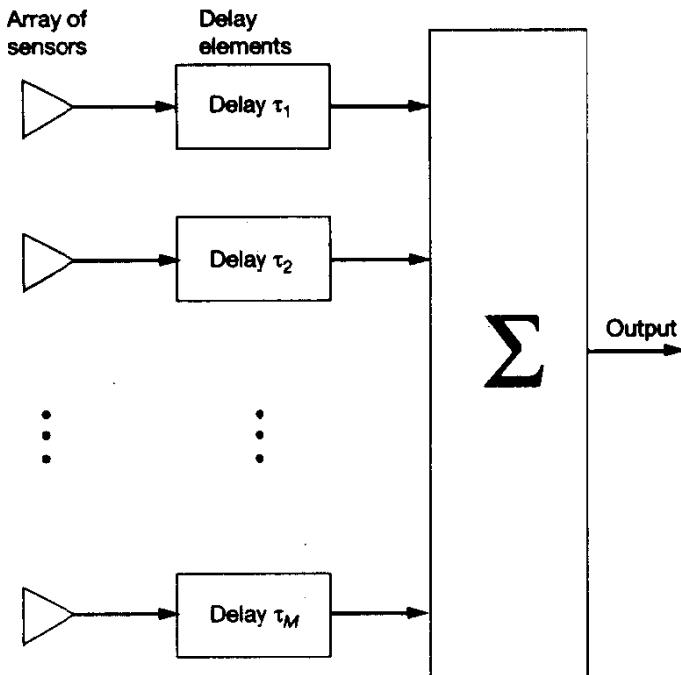


Figure 30 Delay-and-sum beamformer.

In a primitive type of spatial filtering, known as the *delay-and-sum-beamformer*, the various sensor outputs are delayed (by appropriate amounts to align signal components coming from the direction of a target) and then summed, as in Fig. 30. Thus, for a single target, the average power at the output of the delay-and-sum beamformer is maximized when it is steered toward the target. A major limitation of the delay-and-sum beamformer, however, is that it has no provisions for dealing with sources of *interference*.

In order to enable a beamformer to respond to an unknown interference environment, it has to be made *adaptive* in such a way that it places *nulls* in the direction(s) of the source(s) of interference automatically and in real time. By so doing, the output signal-to-noise ratio of the system is increased, and the *directional response* of the system is thereby improved. Below, we consider two examples of *adaptive beamformers* that are well suited for use with narrow-band signals in radar and sonar systems.

Adaptive beamformer with minimum-variance distortionless response. Consider an adaptive beamformer that uses a linear array of M identical sensors, as in Fig. 31. The individual sensor outputs, assumed to be in *baseband* form, are weighted and then summed. The beamformer has to satisfy two requirements: (1) a *steering* capability whereby the target signal is always protected, and (2) the effects of sources of interference are minimized. One method of providing for these two requirements is to

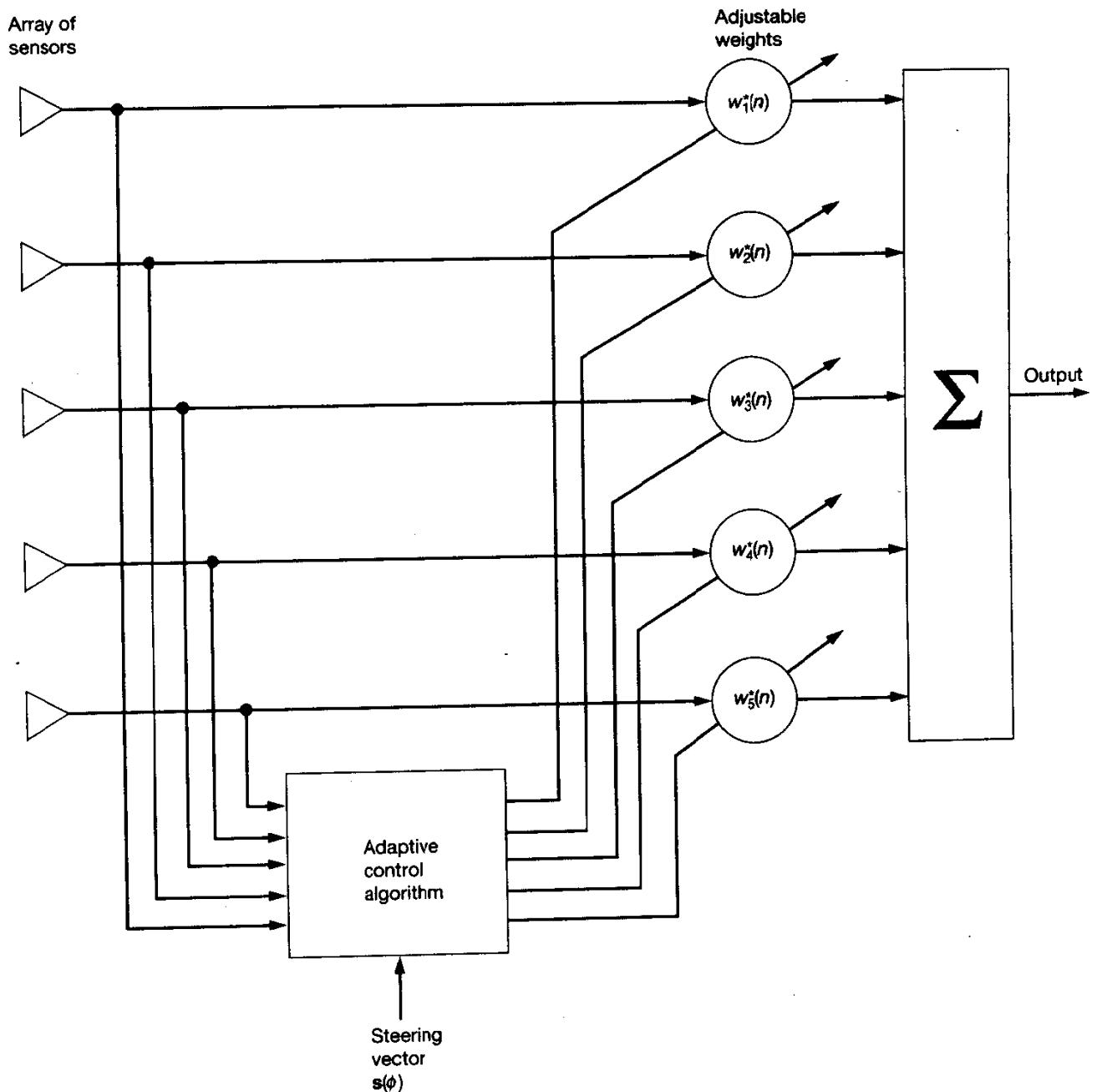


Figure 31 Adaptive beamformer for an array of 5 sensors. The sensor outputs (in baseband form) are complex valued; hence the weights are complex valued.

minimize the variance (i.e., average power) of the beamformer output, subject to the *constraint* that, during the process of adaptation, the weights satisfy the condition:

$$\mathbf{w}^H(n)\mathbf{s}(\phi) = 1 \quad \text{for all } n, \text{ and } \phi = \phi_i \quad (44)$$

where $\mathbf{w}(n)$ is the M -by-1 weight vector and $\mathbf{s}(\phi)$ is an M -by-1 *steering vector*. The superscript H denotes Hermitian transposition (i.e., transposition combined with complex conjugation). In this application, the baseband data are complex valued; hence the need for complex conjugation. The value of *electrical angle* $\phi = \phi_i$ is determined by the direction of the target. The angle ϕ is itself measured with sensor 1 (at the top end of the array) treated as the point of reference.

The dependence of vector $\mathbf{s}(\phi)$ on the angle ϕ is defined by

$$\mathbf{s}(\phi) = [1, e^{-j\phi}, \dots, e^{-j(M-1)\phi}]^T$$

The angle ϕ is itself related to incidence angle θ of a plane wave, measured with respect to the normal to the linear array, as follows¹⁴

$$\phi = \frac{2\pi d}{\lambda} \sin \theta \quad (45)$$

where d is the spacing between adjacent sensors of the array, and λ is the wavelength (see Fig. 32). The incidence angle θ lies inside the range $-\pi/2$ to $\pi/2$. The permissible values that the angle ϕ may assume lie inside the range $-\pi$ to π . This means that we must choose the spacing $d < \lambda/2$, so that there is a one-to-one correspondence between the values of θ and ϕ without ambiguity. The condition $d < \lambda/2$ may be viewed as the spatial analog of the sampling theorem.

The imposition of the *signal-protection constraint* in Eq. (44) ensures that, for a prescribed look direction, the response of the array is maintained constant (i.e., equal to 1), no matter what values are assigned to the weights. An algorithm that minimizes the variance of the beamformer output, subject to this constraint, is therefore referred to as the *minimum-variance distortionless response (MVDR) beamforming algorithm* (Capon, 1969; Owsley, 1985). The imposition of the constraint described in Eq. (44) reduces the number of “degrees of freedom” available to the MVDR algorithm to $M - 2$, where M is the number of sensors in the array. This means that the number of independent nulls produced by the MVDR algorithm (i.e., the number of independent interferences that can be canceled) is $M - 2$.

The MVDR beamforming is a special case of *linearly constrained minimum variance (LCMV) beamforming*. In the latter case, we minimize the variance of the beamformer output, subject to the constraint

$$\mathbf{w}^H(n)\mathbf{s}(\phi) = g \quad \text{for all } n, \text{ and } \phi = \phi_i \quad (46)$$

¹⁴When a plane wave impinges on a linear array as in Fig. 32 there is a spatial delay of $d \sin \theta$ between the signals received at any pair of adjacent sensors. With a wavelength of λ , this spatial delay is translated into an electrical angular difference defined by $\phi = 2\pi(d \sin \theta/\lambda)$.

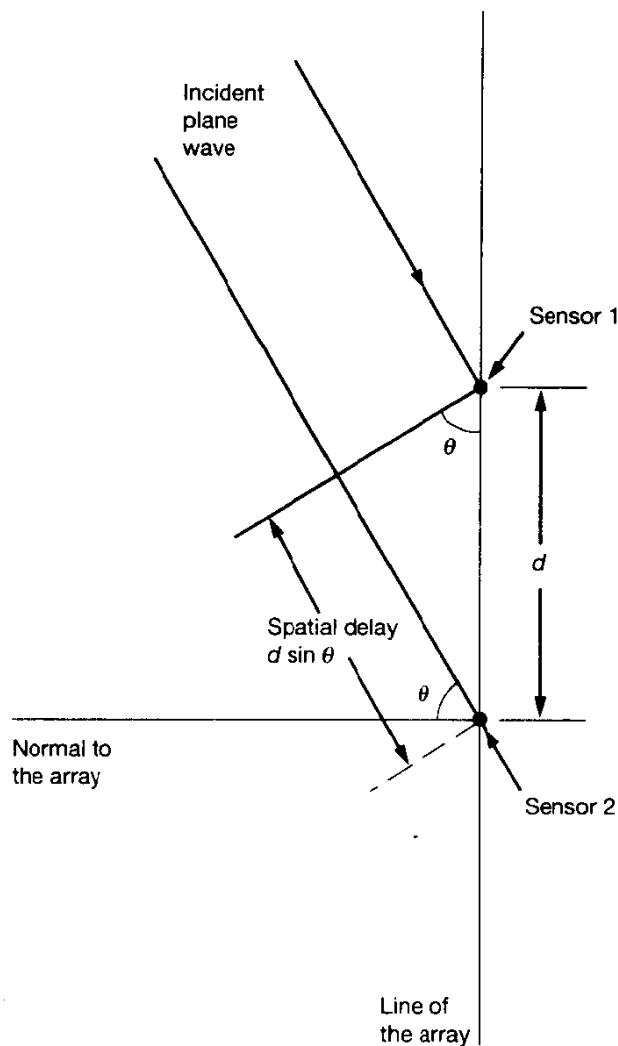


Figure 32 Spatial delay incurred when a plane wave impinges on a linear array.

where g is a complex constant. The LCMV beamformer linearly constrains the weights, such that any signal coming from electrical angle ϕ_i is passed to the output with response (gain) g . Comparing the constraint of Eq. (44) with that of Eq. (46), we see that the MVDR beamformer is indeed a special case of the LCMV beamformer for $g = 1$.

Adaptation in beam space. The MVDR beamformer performs adaptation directly in the *data space*. The adaptation process for interference cancelation may also be performed in *beam space*. To do so, the input data (received by the array of sensors) are transformed into the beam space by means of an *orthogonal multiple-beamforming network*, as illustrated in the block diagram of Fig. 33. The resulting output is processed by a *multiple sidelobe canceler* so as to cancel interference(s) from unknown directions.

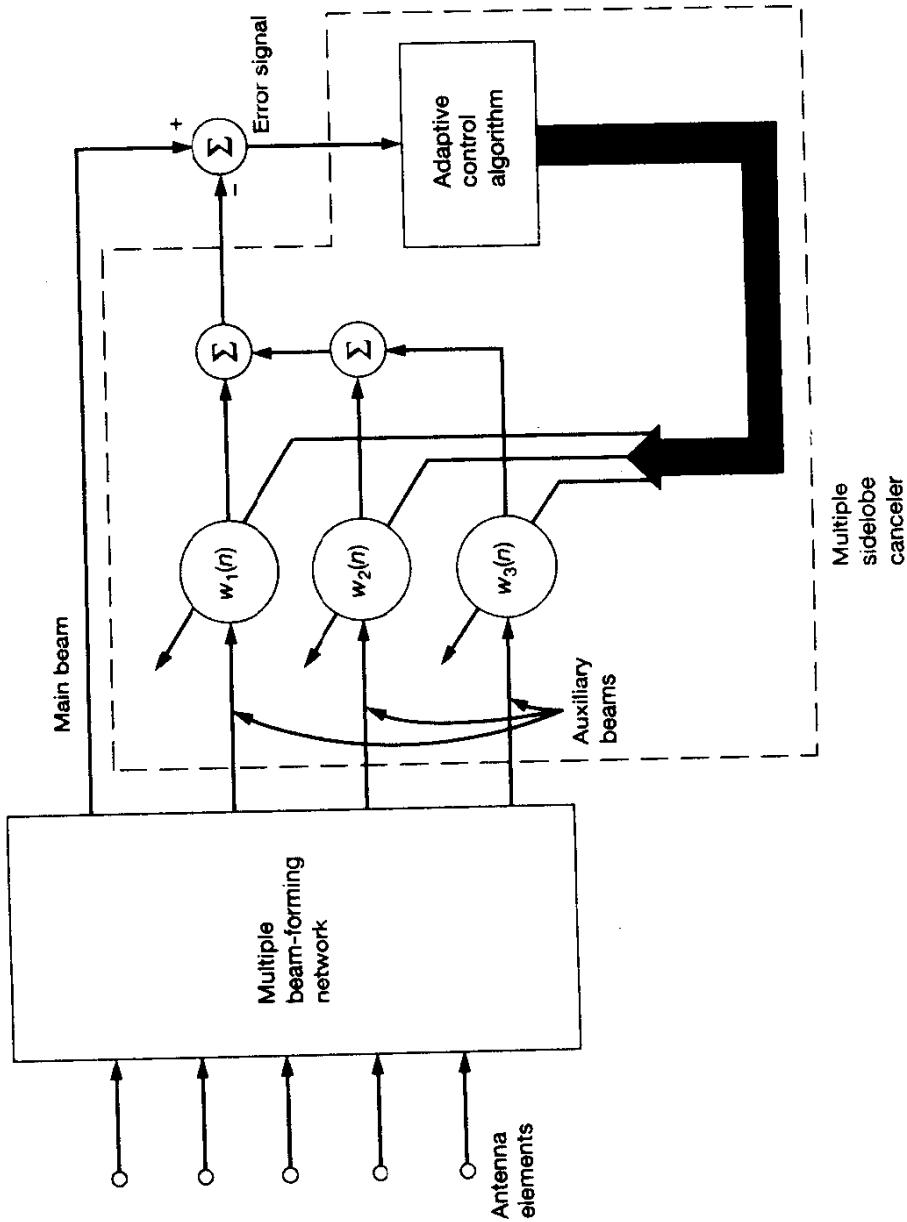


Figure 3.3 Block diagram of adaptive combiner with fixed beams; owing to the symmetric nature of the multiple beamforming network, final values of the weights are real valued.

The beamforming network is designed to generate a set of *orthogonal beams*. The multiple outputs of the beamforming network are referred to as *beam ports*. Assume that the sensor outputs are equally weighted and have a *uniform phase*. Under this condition, the response of the array produced by an incident plane wave arriving at the array along direction θ , measured with respect to the normal to the array, is given by

$$A(\phi, \alpha) = \sum_{n=-N}^N e^{jn\phi} e^{-jna} \quad (47)$$

where $M = (2N + 1)$ is the total number of sensors in the array, with the sensor at the mid-point of the array treated as the point of reference. The electrical angle ϕ is related to θ by Eq. (45), and α is a constant called the *uniform phase factor*. The quantity $A(\phi, \alpha)$ is called the *array pattern*. For $d = \lambda/2$, we find from Eq. (45) that

$$\phi = \pi \sin \theta$$

Summing the geometric series in Eq. (47), we may express the array pattern as

$$A(\phi, \alpha) = \frac{\sin[\frac{1}{2}(2N+1)(\phi - \alpha)]}{\sin[\frac{1}{2}(\phi - \alpha)]} \quad (48)$$

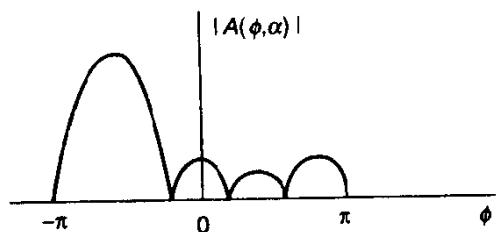
By assigning different values to α , the main beam of the antenna is thus scanned across the range $-\pi < \phi \leq \pi$. To generate an orthogonal set of beams, equal to $2N$ in number, we assign the following discrete values to the uniform phase factor

$$\alpha = \frac{\pi}{2N+1}k, \quad k = \pm 1, \pm 3, \dots, \pm 2N-1 \quad (49)$$

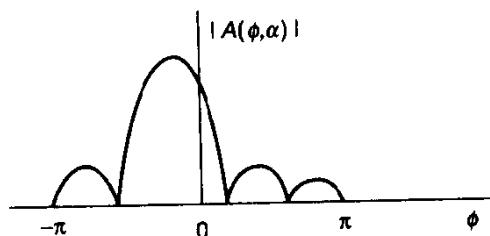
Figure 34 illustrates the variations of the magnitude of the array pattern $A(\phi, \alpha)$ with ϕ for the case of $2N + 1 = 5$ elements and $\alpha = \pm\pi/5, \pm 3\pi/5$. Note that owing to the symmetric nature of the beamformer, the final values of the weights are real valued.

The orthogonal beams generated by the beamforming network represent $2N$ independent *look directions*, one per beam. Depending on the target direction of interest, a particular beam in the set is identified as the *main beam* and the remainder are viewed as *auxiliary beams*. We note from Fig. 34 that each of the auxiliary beams has a *null in the look direction of the main beam*. The auxiliary beams are adaptively weighted by the multiple sidelobe canceler so as to form a cancellation beam that is subtracted from the main beam. The resulting estimation error is fed back to the multiple sidelobe canceler so as to control the corrections applied to its adjustable weights.

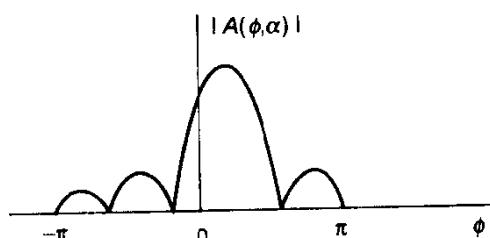
Since all the auxiliary beams have nulls in the look direction of the main beam, and the main beam is excluded from the multiple sidelobe canceler, the overall output of the adaptive beamformer is constrained to have a constant response in the look direction of the main beam (i.e., along the direction of the target). Moreover, with $(2N - 1)$ degrees of freedom (i.e., the number of available auxiliary beams) the system is capable of placing up to $(2N - 1)$ nulls along the (unknown) directions of independent interferences.



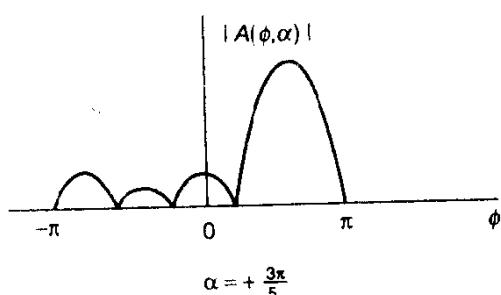
$$\alpha = -\frac{3\pi}{5}$$



$$\alpha = -\frac{\pi}{5}$$



$$\alpha = +\frac{\pi}{5}$$



$$\alpha = +\frac{3\pi}{5}$$

Figure 34 Variations of the magnitude of the array pattern $A(\phi, \alpha)$ with ϕ and α .

Note that with an array of $(2N + 1)$ sensors, we may produce a beamforming network with $(2N + 1)$ orthogonal beam ports by assigning the uniform phase factor the following set of values:

$$\alpha = \frac{k\pi}{2N+1}, \quad k = 0, \pm 2, \dots, \pm 2N \quad (50)$$

In this case, a small fraction of the main lobe of the beam port at either end lies in the non-visible region. Nevertheless, with one of the beam ports providing the main beam and the remaining $2N$ ports providing the auxiliary beams, the adaptive beamformer is now capable of producing up to $2N$ independent nulls.

8. SOME HISTORICAL NOTES

To understand a science it is necessary to know its history.
Auguste Comte (1798—1857)

We complete this introductory chapter by presenting a brief historical review of developments in four areas that are closely related insofar as the subject matter of this book is concerned. The areas are: linear estimation theory, linear adaptive filters, neural networks, and adaptive signal-processing applications.

Linear Estimation Theory

The earliest stimulus for the development of linear estimation theory¹⁵ was apparently provided by astronomical studies in which the motion of planets and comets was studied using telescopic measurement data. The beginnings of a “theory” of estimation in which attempts are made to minimize various functions of errors can be attributed to Galileo Galilei in 1632. However, the origin of linear estimation theory is credited to Gauss who, at the age of 18 in 1795, invented the *method of least squares* to study the motion of heavenly bodies (Gauss, 1809). Nevertheless, in the early nineteenth century, there was considerable controversy regarding the actual inventor of the method of least squares. The controversy arose because Gauss did not publish his discovery in 1795. Rather, it was first published by Legendre in 1805, who independently invented the method (Legendre, 1810).

The first studies of minimum mean-square estimation in stochastic processes were made by Kolmogorov, Krein, and Wiener during the late 1930s and early 1940s (Kolmogorov, 1939; Krein, 1945; Wiener, 1949). The works of Kolmogorov and Krein were independent of Wiener’s, and while there was some overlap in the results, their aims were rather different. There were many conceptual differences (as one would expect after 140 years) between Gauss’s problem and the problem treated by Kolmogorov, Krein, and Wiener.

¹⁵The notes presented on linear estimation are influenced by the following review papers: Sorenson (1970), Kailath (1974), and Makhoul (1975).

Kolmogorov, inspired by some early work of Wold on discrete-time stationary processes (Wold, 1938), developed a comprehensive treatment of the linear prediction problem for discrete-time stochastic processes. Krein noted the relationship of Kolmogorov's results to some early work by Szegö on orthogonal polynomials (Szegö, 1939; Grenander and Szegö, 1958) and extended the results to continuous time by clever use of a bilinear transformation.

Wiener, independently, formulated the continuous-time linear prediction problem and derived an explicit formula for the optimum predictor. Wiener also considered the "filtering" problem of estimating a process corrupted by an additive "noise" process. The explicit formula for the optimum estimate required the solution of an integral equation known as the *Wiener–Hopf equation* (Wiener and Hopf, 1931).

In 1947, Levinson formulated the Wiener filtering problem in discrete time. In the case of discrete-time signals, the Wiener–Hopf equation takes on a matrix form described by¹⁶

$$\mathbf{R}\mathbf{w}_0 = \mathbf{p} \quad (51)$$

where \mathbf{w}_0 is the tap-weight vector of the optimum Wiener filter structured in the form of a transversal filter, \mathbf{R} is the correlation matrix of the tap inputs, and \mathbf{p} is the cross-correlation vector between the tap inputs and the desired response. For stationary inputs, the correlation matrix \mathbf{R} assumes a special structure known as *Toeplitz*, so named after the mathematician O. Toeplitz. By exploiting the properties of a Toeplitz matrix, Levinson derived an elegant recursive procedure for solving the matrix form of the Wiener–Hopf equation (Levinson, 1947). In 1960, Durbin rediscovered Levinson's recursive procedure as a scheme for recursive fitting of autoregressive models to scalar time-series data (Durbin, 1960). The problem considered by Durbin is a special case of Eq. (51) in that the column vector \mathbf{p} comprises the same elements found in the correlation matrix \mathbf{R} . In 1963, Whittle showed there is a close relationship between the Levinson–Durbin recursion and that for Szegö's orthogonal polynomials, and also derived a multivariate generalization of the Levinson–Durbin recursion (Whittle, 1963).

Wiener and Kolmogorov assumed an infinite amount of data and assumed the stochastic processes to be stationary. During the 1950s, some generalizations of the Wiener–Kolmogorov filter theory were made by various authors to cover the estimation of stationary processes given only for a finite observation interval and to cover the estimation of nonstationary processes. However, there were dissatisfactions with the most significant of the results of this period because they were rather complicated, difficult to update with increases in the observations interval, and difficult to modify for the vector case. These last two difficulties became particularly evident in the late 1950s in the problem of determining satellite orbits. In this application, there were generally vector observations of

¹⁶The Wiener–Hopf equation, originally formulated as an integral equation, specifies the optimum solution of a continuous-time linear filter subject to the constraint of causality. This is a difficult-to-solve equation that has resulted in the development of a considerable amount of theory, including spectral factorization. For a tutorial treatment of this subject, see Gardner (1990).

some combinations of position and velocity, and there were also large amounts of data sequentially accumulated with each pass of the satellite over a tracking station. Swerling was one of the first to tackle this problem by presenting some useful recursive algorithms (Swerling, 1958). For different reasons, Kalman independently developed a somewhat more restricted algorithm than Swerling's, but it was an algorithm that seemed particularly matched to the dynamical estimation problems that were brought by the advent of the space age (Kalman, 1960). After Kalman had published his paper and it had attained considerable fame, Swerling wrote a letter claiming priority for the Kalman filter equations (Swerling, 1963). However, history shows that Swerling's plea has fallen on deaf ears. It is ironic that orbit determination problems provided the stimulus for both Gauss's method of least squares and the Kalman filter, and that there were squabbles concerning their inventors. Kalman's original formulation of the linear filtering problem was derived for discrete-time processes. The continuous-time filter was derived by Kalman in his subsequent collaboration with Bucy; this latter solution is sometimes referred to as the *Kalman-Bucy filter* (Kalman and Bucy, 1961).

In a series of stimulating papers, Kailath reformulated the solution to the linear filtering problem by using the *innovations approach* (Kailath, 1968, 1970; Kailath and Frost, 1968; Kailath and Geesey, 1973). In this approach, a stochastic process $u(n)$ is represented as the output of a causal and causally invertible filter driven by a white-noise process $v(n)$. The white noise process $v(n)$ is called the *innovations process*, with the term "innovation" denoting "newness." The reason for this terminology is that each sample of the process $v(n)$ provides entirely new information, in the sense that it is statistically independent of all past samples of the original process $u(n)$, assuming Gaussianity; otherwise, it is only uncorrelated with all past samples of $u(n)$. The idea of innovations approach was introduced by Kolmogorov (1941).

Linear Adaptive Filters

Stochastic gradient algorithms. The earliest work on adaptive filters may be traced back to the late 1950s, during which time a number of researchers were working independently on different applications of adaptive filters. From this early work, the *least-mean-square (LMS) algorithm* emerged as a simple and yet effective algorithm for the operation of adaptive transversal filters. The LMS algorithm was devised by Widrow and Hoff in 1959 in their study of a pattern recognition scheme known as the *adaptive linear (threshold logic) element*, commonly referred to in the literature as the *Adaline* (Widrow and Hoff, 1960; Widrow, 1970). The LMS algorithm is a stochastic gradient algorithm in that it iterates each tap weight of a transversal filter in the direction of the gradient of the squared magnitude of an error signal with respect to the tap weight. As such, the LMS algorithm is closely related to the concept of *stochastic approximation* developed by Robbins and Monro (1951) in statistics for solving certain sequential parameter estimation problems. The primary difference between them is that the LMS algorithm uses a fixed step-size parameter to control the correction applied to each tap weight from one iteration

to the next, whereas in stochastic approximation methods the step-size parameter is made inversely proportional to time n or to a power of n . Another stochastic gradient algorithm, closely related to the LMS algorithm, is the *gradient adaptive lattice (GAL) algorithm* (Griffiths, 1977, 1978); the difference between them is structural in that the GAL algorithm is lattice-based, whereas the LMS algorithm uses a transversal filter.

In 1981, Zames introduced the H^∞ norm (or *minimax criterion*) as a robust index of performance for solving problems in estimation and control, and with it the field of robust control took on a new research direction. In this context, it is particularly noteworthy that Sayed and Rupp (1994) have shown that the LMS algorithm is indeed optimal under the H^∞ criterion. Thus, for the first time, theoretical evidence was presented for the robust performance of the LMS algorithm. It is also of interest to note that the zero-forcing algorithm, which represents an alternative to the LMS algorithm for the adaptive equalization of communication channels, also uses a minimax type of performance criterion (Lucky, 1965).

Recursive least-squares algorithms. Turning next to the recursive least-squares (RLS) family of adaptive filtering algorithms, the original paper on the *standard RLS algorithm* appears to be that of Plackett (1950), though it must be said that many other investigators have derived and rederived the RLS algorithm. In 1974, Godard used Kalman filter theory to derive a variant of the RLS algorithm, which is sometimes referred to in the literature as the *Godard algorithm*. Although prior to this date, several investigators had applied Kalman filter theory to solve the adaptive filtering problem, Godard's approach was widely accepted as the most successful application of Kalman filter theory for a span of two decades. Then, Sayed and Kailath (1994) published an expository paper, in which the *exact* relationship between the RLS algorithm and Kalman filter theory was delineated for the first time, thereby laying the groundwork for how to exploit the vast literature on Kalman filters for solving linear adaptive filtering problems.

In 1981, Gentleman and Kung introduced a numerically robust method, based on the *QR-decomposition* of matrix algebra, for solving the recursive least-squares problem. The resulting adaptive filter structure, sometimes referred to as the *Gentleman-Kung (systolic) array*, was subsequently refined and extended in various ways by many other investigators.

In the 1970s and during subsequent years, a great deal of research effort was expended on the development of numerically stable *fast RLS algorithms*, with the aim of reducing computational complexity to a level comparable to that of the LMS algorithm. In one form or another, the development of these algorithms can be traced back to results derived by Morf in 1974 for solving the deterministic counterpart of the stochastic filtering problem solved efficiently by the Levinson-Durbin algorithm for stationary inputs.

Returning to the paper by Sayed and Kailath (1994), the one-to-one correspondences between RLS and Kalman variables was exploited in that paper to show that QR-decomposition-based RLS algorithms and fast RLS algorithms are all in fact special cases of the Kalman filter, thereby providing a unified treatment of the RLS family of linear adaptive filters in a rather elegant and compact fashion.

Neural Networks¹⁷

Research interest in neural networks began with the pioneering work of McCulloch and Pitts (1943), who described a logical calculus for neural networks. Then, in 1958, Rosenblatt introduced a new approach to the pattern-classification problem using a neural network known as the *perceptron*. Out of this early work on neural networks, the LMS algorithm was pioneered by Widrow and Hoff in 1959, which, as mentioned previously, was used to formulate the Adaline. In the 1960s, it seemed as if neural networks could solve any problem. But then came the book by Minsky and Papert (1969), who used elegant mathematics to demonstrate that there are fundamental limits on what single-layer perceptrons can compute, and with it interest in neural networks took a sharp downturn.

In 1986, successful development of the *back-propagation algorithm* was reported by Rumelhart, Hinton, and Williams as a device for the training of multilayer perceptrons; the back-propagation algorithm is a generalization of the LMS algorithm. In that same year, the two-volume seminal book, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, with Rumelhart and McClelland as editors, was published. This book has been a major influence in reviving interest in the use of neural networks. After the publication of this book, however, it became known that the back-propagation algorithm had actually been described earlier by Werbos in his Ph.D. thesis at Harvard University in 1974.

The multilayer perceptron represents one important type of feedforward layered network that is well suited for adaptive signal processing. Another equally important feedforward layered network is the *radial-basis function (RBF) network*, which was described by Broomhead and Lowe in 1988. However, the basic idea of RBF networks may be traced back to earlier work by Bashkirov, Braverman, and Muchnick in 1964 on the method of potential functions.

The field of neural networks encompasses many other types of network structures and learning algorithms. Indeed, they have been established as an interdisciplinary subject with deep roots in the neurosciences, psychology, mathematics, the physical sciences, and engineering. Needless to say, they have a major impact on adaptive signal processing, particularly in those applications that require the use of nonlinearity.

Adaptive Signal-Processing Applications

Adaptive Equalization. Until the early 1960s, the equalization of telephone channels to combat the degrading effects of intersymbol interference on data transmission was performed by using either fixed equalizers (resulting in a performance loss) or equalizers whose parameters were adjusted manually (a rather cumbersome procedure). In 1965, Lucky made a major breakthrough in the equalization problem by proposing a *zero-forcing algorithm* for automatically adjusting the tap weights of a transversal equalizer. A distinguishing feature of the work by Lucky was the use of a *minimax* type of performance

¹⁷For a more complete historical account of neural networks, see Cowan (1990) and Haykin (1994).

criterion. In particular, he used a performance index called *peak distortion*, which is directly related to the maximum value of intersymbol interference that can occur. The tap weights in the equalizer are adjusted to minimize the peak distortion. This has the effect of *forcing* the intersymbol interference due to those adjacent pulses that are contained in the transversal equalizer to become *zero*; hence the name of the algorithm. A sufficient, but not necessary, condition for optimality of the zero-forcing algorithm is that the *initial distortion* (the distortion that exists at the equalizer input) be less than unity. In a subsequent paper published in 1966, Lucky extended the use of the zero-forcing algorithm to the tracking mode of operation. In 1965, DiToro independently used adaptive equalization for combatting the effect of intersymbol interference on data transmitted over high-frequency links.

The pioneering work by Lucky inspired many other significant contributions to different aspects of the adaptive equalization problem in one way or another. Gersho (1969) and Proakis and Miller (1969) independently reformulated the adaptive equalization problem using a mean-square-error criterion. In 1972, Ungerboeck presented a detailed mathematical analysis of the convergence properties of an adaptive transversal equalizer using the LMS algorithm. In 1974, as mentioned previously, Godard used Kalman filter theory to derive a powerful algorithm for adjusting the tap weights of a transversal equalizer. In 1978, Falconer and Ljung presented a modification of this algorithm that simplified its computational complexity to a level comparable to that of the simple LMS algorithm. Satorius and Alexander (1979) and Satorius and Pack (1981) demonstrated the usefulness of lattice-based algorithms for adaptive equalization of dispersive channels.

This brief historical review pertains to the use of adaptive equalizers for *linear synchronous receivers*; by "synchronous" we mean that the equalizer in the receiver has its taps spaced at the reciprocal of the symbol rate. Even though our interest in adaptive equalizers is largely restricted to this class of receivers, nevertheless, such a historical review would be incomplete without some mention of fractionally spaced equalizers and decision-feedback equalizers.

In a *fractionally spaced equalizer (FSE)*, the equalizer taps are spaced closer than the reciprocal of the symbol rate. An FSE has the capability of compensating for delay distortion much more effectively than a conventional synchronous equalizer. Another advantage of the FSE is the fact that data transmission may begin with an arbitrary sampling phase. However, mathematical analysis of the FSE is much more complicated than for a conventional synchronous equalizer. It appears that early work on the FSE was initiated by Brady (1970). Other contributions to the subject include subsequent work by Ungerboeck (1976) and Gitlin and Weinstein (1981).

A *decision-feedback equalizer* consists of a feedforward section and a feedback section connected as shown in Fig. 35. The feedforward section itself consists of a transversal filter whose taps are spaced at the reciprocal of the symbol rate. The data sequence to be equalized is applied to the input of this section. The feedback section consists of another transversal filter whose taps are also spaced at the reciprocal of the symbol rate. The input applied to the feedback section is made up of decisions on previously detected symbols. The function of the feedback section is to subtract out that portion of intersymbol interference produced by previously detected symbols from the estimates of future symbols. This

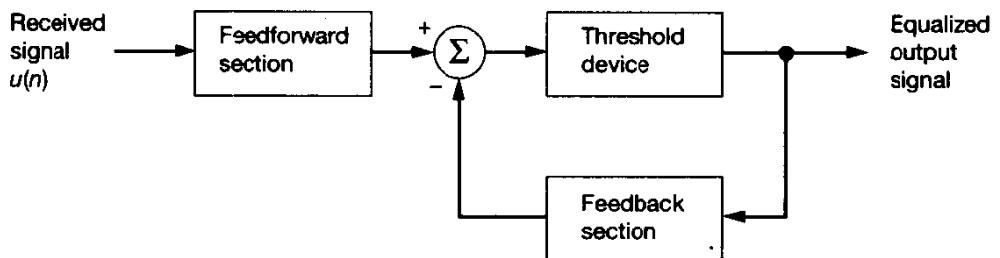


Figure 35 Block diagram of decision-feedback equalizer.

cancellation is an old idea known as the *bootstrap technique*. A decision-feedback equalizer yields good performance in the presence of severe intersymbol interference as experienced in fading radio channels, for example. The first report on decision-feedback equalization was published by Austin (1967), and the optimization of the decision-feedback receiver for minimum mean-squared error was first accomplished by Monsen (1971).

Coding of speech. In 1966, Saito and Itakura used a *maximum likelihood* approach for the application of prediction to speech. A standard assumption in the application of the maximum likelihood principle is that the input process is Gaussian. Under this condition, the exact application of the maximum likelihood principle yields a set of non-linear equations for the parameters of the predictor. To overcome this difficulty, Itakura and Saito utilized approximations based on the assumption that the number of available data points greatly exceeds the prediction order. The use of this assumption makes the result obtained from the maximum likelihood principle assume an approximate form that is the same as the *autocorrelation method* of linear prediction. The application of the maximum likelihood principle is justified on the assumption that speech is a stationary Gaussian process, which seems reasonable in the case of unvoiced sounds.

In 1970, Atal presented the first use of the term "linear prediction" for speech analysis. Details of this new approach, linear predictive coding (LPC), to speech analysis and synthesis were published by Atal and Hanauer in 1971, in which the speech waveform is represented directly in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the excitation. The predictor coefficients are determined by minimizing the mean-squared error, with the error defined as the difference between the actual and predicted values of the speech samples. In the work by Atal and Hanauer, the speech wave was sampled at 10 kHz and then analyzed by predicting the present speech sample as a linear combination of the 12 previous samples. Thus 15 parameters [the 12 parameters of the predictor, the pitch period, a binary parameter indicating whether the speech is voiced or unvoiced, and the root-mean-square (rms) value of the speech samples] were used to describe the speech analyzer. For the speech synthesizer, an all-pole filter was used, with a sequence of quasi-periodic pulses or a white-noise source providing the excitation.

Another significant contribution to the linear prediction of speech was made in 1972 by Itakura and Saito; they used partial correlation techniques to develop a new structure,

the lattice, for formulating the linear prediction problem.¹⁸ The parameters that characterize the lattice predictor are called *reflection coefficients* or *partial correlation (PARCOR) coefficients*, depending on the algebraic sign used in the definition. Although by that time the essence of the lattice structure had been considered by several other investigators, the invention of the lattice predictor is credited to Saito and Itakura. In 1973, Wakita showed that the filtering actions of the lattice predictor model and an acoustic tube model of speech are identical, with the reflection coefficients in the acoustic tube model as common factors. This discovery made possible the extraction of the reflection coefficients by the use of a lattice predictor.

Early designs of a lattice predictor were based on a *block processing* approach (Burg, 1967). In 1981, Makhoul and Cossell used an *adaptive* approach for designing the lattice predictor for applications in speech analysis and synthesis. They showed that the convergence of the adaptive lattice predictor is fast enough for its performance to equal that of the optimal (but more expensive) adaptive autocorrelation method.

This historical review on speech coding relates to LPC vocoders. We next present a historical review of the adaptive predictive coding of speech, starting with ordinary pulse-code modulation (PCM).

PCM was invented in 1937 by Reeves (1975). This was followed by the invention of differential pulse-code modulation (DPCM) by Cutler (1952). The early use of DPCM for the predictive coding of speech signals was limited to linear predictors with *fixed* parameters (McDonald, 1966). However, due to the nonstationary nature of speech signals, a fixed predictor cannot predict the signal values efficiently at all times. In order to respond to the nonstationary characteristics of speech signals, the predictor has to be adaptive (Atal and Schroeder, 1967). In 1970, Atal and Schroeder described a sophisticated scheme for adaptive predictive coding of speech. The scheme recognizes that there are two main causes of redundancy in speech (Schroeder, 1966): (1) quasi-periodicity during voiced segments, and (2) lack of flatness of the short-time spectral envelope. Thus, the predictor is designed to remove signal redundancy in two stages. The first stage of the predictor removes the quasi-periodic nature of the signal. The second stage removes formant information from the spectral envelope. The scheme achieves dramatic reductions in bit rate at the expense of a significant increase in circuit complexity. Atal and Schroeder (1970) report that the scheme can transmit speech at 10 kb/s, which is several times less than the bit rate required for logarithmic-PCM encoding with comparable speech quality.

Spectrum analysis. At the turn of the twentieth century, Schuster introduced the *periodogram* for analyzing the power spectrum¹⁹ of a time series (Schuster, 1898). The periodogram is defined as the squared amplitude of the discrete Fourier transform of the time series. The periodogram was originally used by Schuster to detect and estimate the amplitude of a sine wave of known frequency that is buried in noise. Until the work of

¹⁸According to Markel and Gray (1976), the work of Itakura and Saito in Japan on the PARCOR formulation of linear prediction had been presented in 1969.

¹⁹For a fascinating historical account of the concept of power spectrum, its origin and its estimation, see Robinson (1982).

Yule in 1927, the periodogram was the only numerical method available for spectrum analysis. However, the periodogram suffers from the limitation that when it is applied to empirical time series observed in nature the results obtained are very erratic. This led Yule to introduce a new approach based on the concept of a *finite parameter model* for a stationary stochastic process in his investigation of the periodicities in time series with special reference to Wolfer's sunspot number (Yule, 1927). Yule, in effect, created a stochastic feedback model in which the present sample value of the time series is assumed to consist of a linear combination of past sample values plus an error term. This model is called an autoregressive model in that a sample of the time series regresses on its own past values, and the method of spectrum analysis based on such a model is accordingly called autoregressive spectrum analysis. The name "autoregressive" was coined by Wold in his doctoral thesis (Wold, 1938).

Interest in the autoregressive method was reinitiated by Burg (1967, 1975). Burg introduced the term *maximum-entropy method* to describe an algorithmic approach for estimating the power spectrum directly from the available time series. The idea behind the maximum-entropy method is to extrapolate the autocorrelation function of the time series in such a way that the *entropy* of the corresponding probability density function is maximized at each step of the extrapolation. In 1971, Van den Bos showed that the maximum-entropy method is equivalent to least-squares fitting of an autoregressive model to the known autocorrelation sequence.

Another important contribution made to the literature on spectrum analysis is that by Thomson (1982). His *method of multiple windows*, based on the prolate spheroidal wave functions, represents a nonparametric method for spectrum estimation that overcomes many of the limitations of the above-mentioned techniques.

Adaptive Noise Cancelation. The initial work on adaptive echo cancelers started around 1965. It appears that Kelly of Bell Telephone Laboratories was the first to propose the use of an adaptive filter for echo cancellation, with the speech signal itself utilized in performing the adaptation; Kelly's contribution is recognized in the paper by Sondhi (1967). This invention and its refinement are described in the patents by Kelly and Logan (1970) and Sondhi (1970).

The adaptive line enhancer was originated by Widrow and his co-workers at Stanford University. An early version of this device was built in 1965 to cancel 60-Hz interference at the output of an electrocardiographic amplifier and recorder. This work is described in the paper by Widrow et al. (1975b). The adaptive line enhancer and its application as an adaptive detector are patented by McCool et al. (1980).

The adaptive echo canceler and the adaptive linear enhancer, although intended for different applications, may be viewed as examples of the *adaptive noise canceler* discussed by Widrow et al. (1975). This scheme operates on the outputs of two sensors: a *primary sensor* that supplies a desired signal of interest buried in noise, and a *reference sensor* that supplies noise alone, as illustrated in Fig. 24. It is assumed that (1) the signal and noise at the output of the primary sensor are uncorrelated, and (2) the noise at the output of the reference sensor is correlated with the noise component of the primary sensor output.

The adaptive noise canceler consists of an adaptive filter that operates on the reference sensor output to produce an *estimate* of the noise, which is subtracted from the primary sensor output. The overall output of the canceler is used to control the adjustments applied to the tap weights in the adaptive filter. The adaptive canceler tends to minimize the mean-square value of the overall output, thereby causing the output to be the best estimate of the desired signal in the minimum-mean-square error sense.

Adaptive beamforming. The development of adaptive beamforming technology may be traced back to the invention of the *intermediate frequency (IF) sidelobe canceler* by Howells in the late 1950s. In a paper published in the 1976 Special Issue of the IEEE Transactions on Antennas and Propagation, Howells describes his personal observations on early work on adaptive antennas at the General Electric and Syracuse University Research Corporation (Howells, 1976). According to this historic report, Howells had developed by mid-1957 a sidelobe canceler capable of automatically nulling out the effect of one jammer. The sidelobe canceler uses a *primary* (high-gain) antenna and a *reference omni-directional* (low-gain) antenna to form a two-element array with one degree of freedom that makes it possible to steer a deep null anywhere in the sidelobe region of the combined antenna pattern. In particular, a null is placed in the direction of the jammer, with only a minor perturbation of the main lobe. Subsequently, Howells (1965) patented the sidelobe canceler.

The second major contribution to adaptive array antennas was made by Applebaum in 1966. In a classic report, he derived the *control law* governing the operation of an adaptive array antenna, with a control loop for each element of the array (Applebaum, 1966). The algorithm derived by Applebaum was based on maximizing the signal-to-noise ratio (SNR) at the array antenna output for any type of noise environment. Applebaum's theory included the sidelobe canceler as a special case. His 1966 classic report was reprinted in the 1976 Special Issue of IEEE Transactions on Antennas and Propagation.

Another algorithm for the weight adjustment in adaptive array antennas was advanced independently in 1967 by Widrow and his co-workers at Stanford University. They based their theory on the simple and yet effective LMS algorithm. The 1967 paper by Widrow et al. was not only the first publication in the open literature on adaptive array antenna systems, but also it is considered to be another classic of that era.

It is noteworthy that the maximum SNR algorithm (used by Applebaum) and the LMS algorithm (used by Widrow and his co-workers) for adaptive array antennas are rather similar. Both algorithms derive the control law for adaptive adjustment of the weights in the array antenna by sensing the correlation between element signals. Indeed, they both converge toward the optimum Wiener solution for stationary inputs (Gabriel, 1976).

A different method for solving the adaptive beamforming problem was proposed by Capon (1969). Capon realized that the poor performance of the delay-and-sum beamformer is due to the fact that its response along a direction of interest depends not only on the power of the incoming target signal but also undesirable contributions received from other sources of interference. To overcome this limitation of the delay-and-sum beam-

former, Capon proposed a new beamformer in which the weight vector $\mathbf{w}(n)$ is chosen so as to *minimize the variance* (i.e., average power) of the beamformer output, subject to the constraint $\mathbf{w}^H(n)\mathbf{s}(\phi) = 1$ for all n , where $\mathbf{s}(\phi)$ is a prescribed *steering vector*. This constrained minimization yields an adaptive beamformer with *minimum-variance distortionless response (MVDR)*.

In 1983, McWhirter proposed a simplification of the Gentleman–Kung (systolic) array for recursive least-squares estimation. The resulting filtering structure, often referred to as the *McWhirter (systolic) array*, is particularly well suited for adaptive beamforming applications.

The historical notes presented in this last section of the chapter on adaptive filter theory and applications are not claimed to be complete. Rather, they are intended to highlight many of the significant contributions made to this important part of the ever-expanding field of signal processing. Above all, it is hoped that they provide a source of inspiration to the reader.

PART 1

Background Material

Part I consists of Chapters 1 through 4. In this part of the book we present background material on discrete-time signals and systems and thereby lay a foundation for the rest of the book, as summarized here:

- Chapter 1 reviews fundamentals of discrete-time signal processing, with emphasis on the z -transform, the discrete Fourier transform, and the discrete cosine transform.
- Chapter 2 covers the time-domain characteristics of discrete-time stochastic processes.
- Chapter 3 covers the frequency-domain characteristics of discrete-time stochastic processes, with particular emphasis on the notion of a power spectrum or power spectral density. Higher-order statistics and cyclostationary properties of stochastic processes are also discussed here.
- In Chapter 4 we study the eigenvalue problem, which is central to a detailed mathematical description of discrete-time wide-sense stationary processes.

CHAPTER

1

Discrete-time Signal Processing

Typically, a signal of interest is described as a function of time. The transformation of a signal from the time domain into the frequency domain plays a key role in the study of signal processing. The particular transformation used in practice depends on the type of signal being considered. Given the pervasive nature of digital processing and the benefits (flexibility and accuracy of computation) offered by its use, our interest in this book is confined to discrete-time signals. Specifically, the signal is described as a *time series*, consisting of a sequence of uniformly spaced samples whose varying amplitudes carry the useful information content of the signal. In such a situation, the transforms that immediately come to mind are two closely related transforms, namely, the z-transform and the Fourier transform. The Fourier transform is defined in terms of a real variable (frequency), whereas the z-transform is defined in terms of a complex variable.

In this chapter we present a brief review of discrete-time signal processing,¹ beginning with a definition of the z-transform and its properties.

1.1 z-TRANSFORM

Consider a time series (sequence) denoted by the samples $u(n)$, $u(n - 1)$, $u(n - 2)$, . . . , where n denotes *discrete time*. For convenience of presentation, it is assumed that the

¹For a detailed treatment of the many facets of discrete-time signal processing, see Oppenheim and Schafer (1989).

spacing between adjacent samples of the sequence is unity. The sequence is written as $\{u(n)\}$ or simply $u(n)$. The two-sided *z-transform* of $u(n)$ is defined as

$$\begin{aligned} U(z) &= z[u(n)] \\ &= \sum_{n=-\infty}^{\infty} u(n)z^{-n} \end{aligned} \quad (1.1)$$

where z is a *complex variable*. The first line of Eq. (1.1) describes the *z-transform* as an “operator,” and the second line defines it as an infinite power series in z . The sequence $u(n)$ and its *z-transform* form a *z-transform pair*, described by

$$u(n) \rightleftharpoons U(z) \quad (1.2)$$

The power series defined in Eq. (1.1) is a *Laurent series*, which features prominently in the functional theory of complex variables; a brief review of complex variable theory is presented in Appendix A. The important point to note here is that for the *z-transform* $U(z)$ to be meaningful, the power series defined in Eq. (1.1) must be absolutely summable; that is, $U(z)$ is uniformly convergent. For any given time series $u(n)$, the set of values of the complex variable z for which the *z-transform* $U(z)$ is uniformly convergent is referred to as the *region of convergence* (ROC).

Let the region of convergence of the *z-transform* $U(z)$ be denoted by the annular domain $R_1 < |z| < R_2$. Let \mathcal{C} be a closed contour that encloses the origin and is contained in this region of convergence. Then, given the *z-transform* $U(z)$, the original time series $u(n)$ may be uniquely recovered using the *z-transform inversion integral formula* (see Appendix A)

$$u(n) = \frac{1}{2\pi j} \oint_{\mathcal{C}} U(z)z^n \frac{dz}{z} \quad (1.3)$$

where the contour integration is performed by transversing the contour \mathcal{C} in the counter-clockwise direction.

Properties of the *z*-Transform

The *z-transform* is a *linear transform* in that it satisfies the principle of superposition. Specifically, given two sequences $u_1(n)$ and $u_2(n)$ whose *z-transforms* are denoted by $U_1(z)$ and $U_2(z)$, respectively, we may write

$$a u_1(n) + b u_2(n) \rightleftharpoons a U_1(z) + b U_2(z) \quad (1.4)$$

where a and b are scaling factors. The region of convergence, for which Eq. (1.4) holds, contains the intersection of the regions of convergence of $U_1(z)$ and $U_2(z)$.

Another important property of the *z-transform* is the *time-shifting property*. Let $U(z)$ denote the *z-transform* of the sequence $u(n)$. The *z-transform* of $u(n - n_0)$ is described by the relation

$$u(n - n_0) \rightleftharpoons z^{-n_0}U(z) \quad (1.5)$$

where n_0 is an integer. Equation (1.5) holds for the same region of convergence as the original time series $u(n)$, except for a possible addition or deletion of $z = 0$ or $z = \infty$. For the special case of $n_0 = 1$, we see that such a time shift has the effect of multiplying the z -transform $U(z)$ by the factor z^{-1} . It is for this reason that z^{-1} is commonly referred to as a *unit-delay element*.

One other property of the z -transform of particular interest to us is the *convolution theorem*. Let $U_1(z)$ and $U_2(z)$ denote the z -transforms of the time series $u_1(n)$ and $u_2(n)$, respectively. According to the convolution theorem, we have

$$\sum_{i=-\infty}^{\infty} u_1(i)u_2(n-i) \rightleftharpoons U_1(z)U_2(z) \quad (1.6)$$

where the region of convergence includes the intersection of the regions of convergence of $U_1(z)$ and $U_2(z)$. The proof of Eq. (1.6) follows directly from the defining equation (1.1). In other words, convolution of two sequences in the time domain is transformed into multiplication of their z -transforms in the frequency domain.

1.2 LINEAR TIME-INVARIANT FILTERS

The z -transform plays a key role in the study of a particular class of filters known as *linear time-invariant filters*, which are characterized by the following two properties: linearity and time invariance. The *linearity* property means that the filter satisfies the principle of superposition. Specifically, if $v_1(n)$ and $v_2(n)$ are two different *excitations* applied to the filter and $u_1(n)$ and $u_2(n)$ are the *responses* produced by the filter, respectively, then the response of the filter to the composite excitation $a v_1(n) + b v_2(n)$ is equal to $a u_1(n) + b u_2(n)$, where a and b are arbitrary constants. The *time-invariance* property means that if $u(n)$ is the response of the filter due to the excitation $v(n)$, then the response of the filter to the new excitation $v(n - k)$ is equal to $u(n - k)$, where k is an arbitrary time shift.

One useful way of describing a linear time-invariant filter is in terms of its *impulse response*, defined as the response of the filter to a unit impulse or delta function applied to the filter at zero time. Let $h(n)$ denote the impulse response of the filter. The response $u(n)$ of the filter produced by an arbitrary excitation $v(n)$ is defined by the *convolution sum*

$$u(n) = \sum_{i=-\infty}^{\infty} h(i)v(n-i) \quad (1.7)$$

Applying the z -transform to both sides of Eq. (1.7) and invoking the convolution theorem, we may write

$$U(z) = H(z)V(z) \quad (1.8)$$

where $U(z)$, $V(z)$, and $H(z)$ are the z -transforms of $u(n)$, $v(n)$, and $h(n)$, respectively.

The z -transform $H(z)$ [i.e., the z -transform of the impulse response $h(n)$] is called the *transfer function* of the filter; it provides the basis of another way of describing a linear time-invariant filter. According to Eq. (1.8), we have

$$H(z) = \frac{U(z)}{V(z)} \quad (1.9)$$

Thus, the transfer function $H(z)$ is equal to the ratio of the z -transform of the filter's response to the z -transform of the excitation applied to the filter.

In an important subclass of linear time-invariant filters, the input sequence (excitation) $v(n)$ and the output sequence (response) $u(n)$ are related by a difference equation of order N as follows:

$$\sum_{j=0}^N a_j u(n-j) = \sum_{j=0}^N b_j v(n-j) \quad (1.10)$$

where the a_j and the b_j are constant coefficients. Applying the z -transform to both sides of Eq. (1.10) and using the time-shifting property of the z -transform, we may readily express the transfer function of the filter as

$$\begin{aligned} H(z) &= \frac{U(z)}{V(z)} \\ &= \frac{\sum_{j=0}^N a_j z^{-j}}{\sum_{j=0}^N b_j z^{-j}} \end{aligned} \quad (1.11)$$

Equivalently, we may express the rational transfer function of Eq. (1.11) in the factored form

$$H(z) = \frac{a_0}{b_0} \frac{\prod_{k=1}^N (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (1.12)$$

Each factor $(1 - c_k z^{-1})$ in the numerator on the right-hand side of Eq. (1.12) contributes a zero at $z = c_k$ and a pole at $z = 0$, whereas each factor $(1 - d_k z^{-1})$ in the denominator contributes a pole at $z = d_k$ and a zero at $z = 0$. Thus, except for the scaling factor a_0/b_0 , the transfer function $H(z)$ of the filter is uniquely defined in terms of its poles and zeros. Note that with the time-domain behavior of the filter defined by a constant-coefficient difference equation of the form given in Eq. (1.10), the poles and zeros of the transfer function $H(z)$ are real or else appear in complex-conjugate pairs.

Based on the representation given in Eq. (1.12), we may distinguish between two distinct types of linear time-invariant filters:

1. *Finite-duration impulse response (FIR) filters.* For this type of filter, d_k is zero for all k , which means that the filter is an *all-zero filter* in that the poles of its transfer function $H(z)$ are all confined to $z = 0$. Correspondingly, the impulse response $h(n)$ of the filter has a finite duration; hence the descriptor “finite-duration impulse response.”
2. *Infinite-duration impulse response (IIR) filters.* In this second type of filter, the transfer function $H(z)$ has at least one nonzero pole that is not canceled by a zero. Correspondingly, the impulse response $h(n)$ of the filter has an infinite duration; hence the descriptor “infinite-duration impulse response.” When c_k is zero for all k , the IIR filter is said to be an *all pole filter*, in that the zeros of its transfer function $H(z)$ are all confined to $z = 0$.

Figures 1.1(a) and 1.1(b) show examples of FIR and IIR filters, respectively. The boxes labeled z^{-1} represent unit-delay elements, and the circles labeled a_1, a_2, \dots, a_N represent filter coefficients. Note that the FIR filter of Fig. 1.1(a) involves feedforward paths only, whereas the IIR filter of Fig. 1.1(b) involves both feedforward and feedback paths. In both cases, the basic functional blocks needed to build the filters consist of unit-delay elements, multipliers, and adders.

Causality and Stability

A linear time-invariant filter is said to be *causal* if its impulse response $h(n)$ is zero for negative time, as shown by

$$h(n) = 0 \quad \text{for } n < 0 \quad (1.13)$$

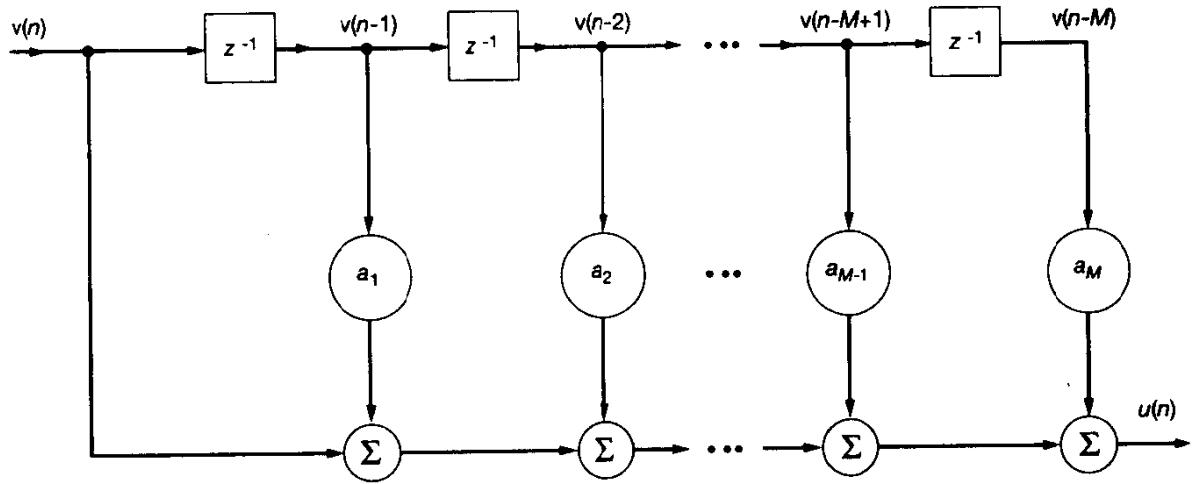
Clearly, for a filter to operate in real time, it would have to be causal. However, causality is not a necessary requirement for physical realizability. There are many applications in which the signal to be processed is available in stored form; in these situations, the filter can be noncausal and yet physically realizable.

The filter is said to be *stable* if the output sequence (response) of the filter is bounded for all bounded input sequences (excitations). This requirement is called the *bounded input–bounded output (BIBO) stability criterion*, the application of which is well suited for linear time-invariant filters. From Eq. (1.7) we readily see that the necessary and sufficient condition for BIBO stability is

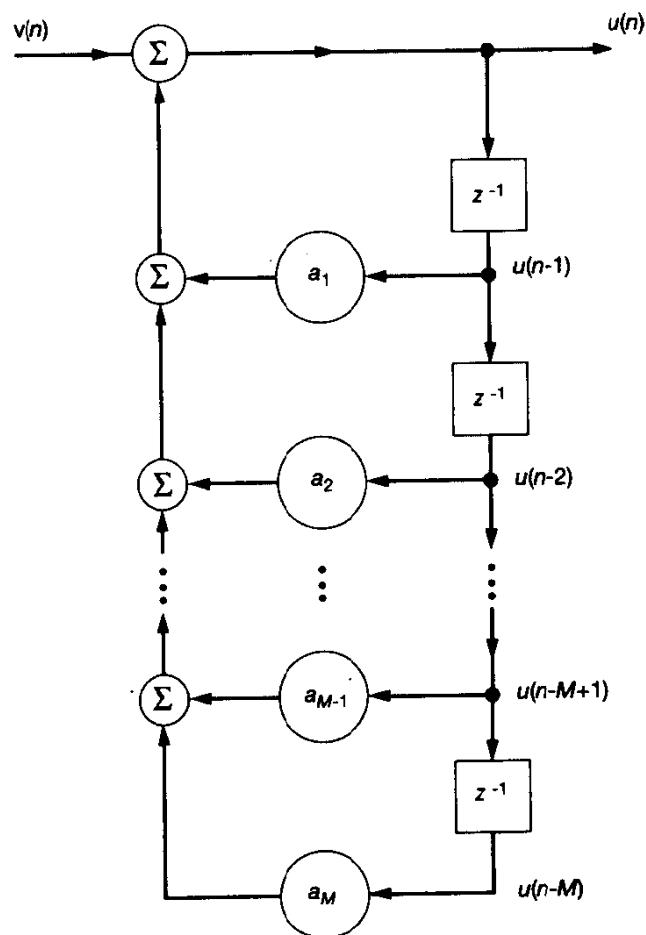
$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \quad (1.14)$$

That is, the impulse response of the filter must be absolutely summable.

Causality and stability are not necessarily compatible requirements. For a linear time-invariant filter defined by the difference equation (1.10) to be both causal and stable,



(a) FIR filter



(b) IIR filter

Figure 1.1 Two basic types of filters.

the region of convergence of the filter's transfer function $H(z)$ must satisfy two requirements (Oppenheim and Schafer, 1989):

1. It must lie outside the outermost poles of $H(z)$.
2. It must include the unit circle in the z -plane.

Clearly, these requirements can only be satisfied if all the poles of $H(z)$ lie inside the unit circle, as indicated in Fig. 1.2. We may thus make the following important statement on the issue of stability: *A causal, linear time-invariant filter is stable if and only if all of the*

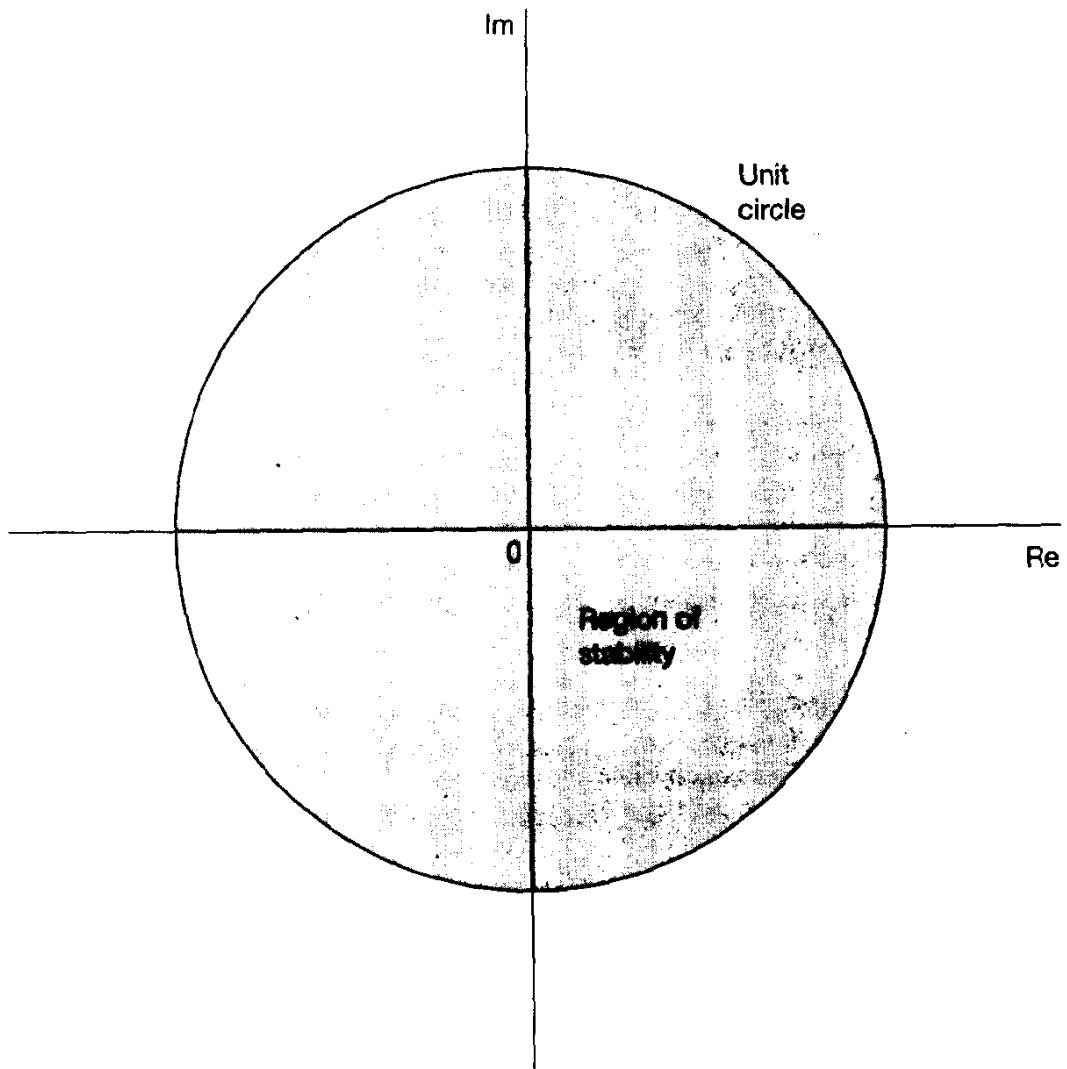


Figure 1.2 z -plane.

poles of the filter's transfer function lie inside the unit circle in the z-plane. Note that this statement says nothing about the zeros of the filter's transfer function $H(z)$. Insofar as causality and stability are concerned, the zeros of $H(z)$ can indeed lie anywhere in the z-plane.

1.3 MINIMUM-PHASE FILTERS

The unit circle plays a critical role not only in the stability criterion of a causal filter, but also in the evaluation of its frequency response. Specifically, setting

$$z = e^{j2\pi f}$$

in the expression for the transfer function $H(z)$, we get the filter's *frequency response* denoted by $H(e^{j2\pi f})$, where f denotes the frequency in Hertz. Expressing $H(e^{j2\pi f})$ in its polar form, we may define the frequency response of the filter in terms of two components:

- The *magnitude (amplitude) response*, denoted by $|H(e^{j2\pi f})|$
- The *phase response*, denoted by $\text{ang}(H(e^{j2\pi f}))$

In the case of a special class of filters known as *minimum-phase filters*, the magnitude response and phase response of the filter are uniquely related to each other, in that if we are given one of them, we can compute the other component uniquely (Oppenheim and Schafer, 1989). A minimum-phase filter derives its name from the fact that, for a specified magnitude response, it has the minimum phase response possible for all values of z on the unit circle.

The minimum-phase property of a linear time-invariant filter places restrictions of its own on possible locations of the zeros of the filter's transfer function $H(z)$. Specifically, the zeros of $H(z)$ must satisfy the following requirements:

- The zeros of $H(z)$ may lie anywhere inside the unit circle in the z-plane.
- Zeros are permitted to lie on the unit circle, provided that they are *simple* (i.e., they are of order one).

A minimum-phase filter has the following interesting property: given a minimum-phase filter of transfer function $H(z)$, we may define an *inverse filter* with transfer function $1/H(z)$ that is both causal and stable, provided that $H(z)$ does not have zeros on the unit circle. The cascade connection of such a pair of filters has a transfer function equal to unity.

Finally, we note that a *nonminimum-phase filter*, whose transfer function $H(z)$ has zeros outside the unit circle, can always be treated as the cascade connection of a minimum-phase filter and an all-pass filter. An *all-pass filter* is defined as a filter whose transfer function has poles and zeros that are the reciprocals of each other with respect to the unit circle; naturally, the poles are confined to the interior of the unit circle, in which case all the zeros are confined to the exterior of the unit circle. Consequently, the magnitude response of an all-pass filter is equal to unity, which means that it passes all the frequency

components of the input signal with no change in amplitude. When the nonminimum-phase filter has all of its zeros located outside the unit circle, it is said to be a *maximum-phase filter*.

1.4 DISCRETE FOURIER TRANSFORM

The *Fourier transform* of a sequence is readily obtained from its *z-transform* simply by setting the complex variable z equal to $\exp(j2\pi f)$, where f is the real frequency variable. When the sequence of interest has a finite duration, we may go one step further and develop a Fourier representation for it by defining the *discrete Fourier transform (DFT)*. The DFT is itself made up of a sequence of samples, uniformly spaced in frequency. The DFT has established itself as a powerful tool in digital signal processing by virtue of the fact that there exist efficient algorithms for its numerical computation; these algorithms are known collectively as *fast Fourier transform (FFT) algorithms* (Oppenheim and Schafer, 1989).

Consider a finite-duration sequence $u(n)$, assumed to be of length N . The DFT of $u(n)$ is defined by

$$U(k) = \sum_{n=0}^{N-1} u(n) \exp\left(-\frac{j2\pi kn}{N}\right), \quad k = 0, \dots, N-1 \quad (1.15)$$

The *inverse discrete Fourier transform (IDFT)* of $U(k)$ is defined by

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} U(k) \exp\left(\frac{j2\pi kn}{N}\right), \quad n = 0, 1, \dots, N-1 \quad (1.16)$$

Note that both the original sequence $u(n)$ and its DFT $U(k)$ are of the same length, N . We thus speak of the discrete Fourier transform as an “ N -point DFT.”

The discrete Fourier transform has an interesting interpretation in terms of the *z-transform*, as described here: the DFT of a finite-duration sequence may be obtained by evaluating the *z-transform* of that same sequence at N points uniformly spaced on the unit circle in the *z-plane*. This “sampling” process is illustrated in Fig. 1.3 for $N = 8$.

Though the sequence $u(n)$ and its DFT $U(k)$ are defined as “finite-length” sequences, in reality they both represent a single period of their respective periodic sequences. This double periodicity is the direct consequence of sampling a continuous-time signal as well as its continuous Fourier transform.

1.5 IMPLEMENTING CONVOLUTIONS USING THE DFT

The underlying “double-periodic” nature of the discrete Fourier transform just mentioned imparts to it certain properties that distinguish it from the continuous Fourier transform. In particular, the *linear convolution* of two sequences, $h(n)$ and $v(n)$, say, involves multiplying one sequence by a time-reversed and linearly shifted version of the other sequence and

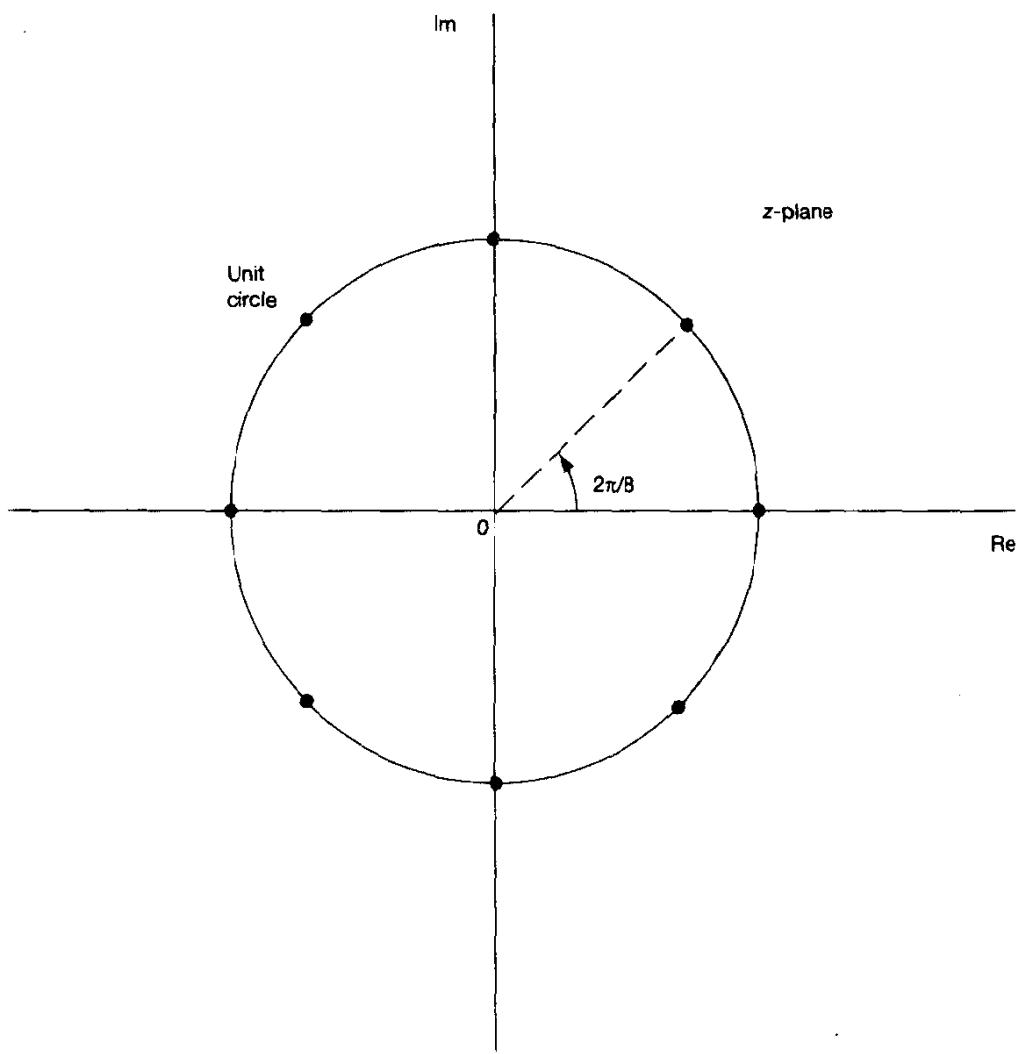


Figure 1.3 A set of $N (= 8)$ uniformly spaced points on the unit circle in the *z*-plane.

then summing the product $h(i)v(n - i)$ over all i , as described in Eq. (1.7). In contrast, in the case of DFT we have a *circular convolution* in which the second sequence is circularly time-reversed and circularly shifted with respect to the first sequence. In other words, in circular convolution both sequences have length N (or less) and the sequences are shifted modulo N . It is only when convolution is defined in this way that the convolution of two sequences in the time domain is transformed into the product of their DFTs in the frequency domain (Oppenheim and Schafer, 1989). Stating this property in another way, if we multiply the DFTs of two finite-duration sequences and then evaluate the IDFT of the product, the result so obtained is equivalent to a circular convolution of the original sequences.

With circular convolution being markedly different from linear convolution, the key issue is how to use the DFT to perform linear convolution. To illustrate how we may do this, consider two sequences $v(n)$ and $h(n)$, assuming that they are of lengths L and P , respectively. The linear convolution of these two sequences is a finite-duration sequence of length $L + P - 1$. Recognizing that the convolution of two periodic sequences is another periodic sequence of the same period, we may proceed as follows:

- Append an appropriate number of zero-valued samples to $v(n)$ and $h(n)$ to make them both N -point sequences, where $N = L + P - 1$; this process is referred to as *zero padding*.
- Compute the N -point DFTs of the appended versions of the sequences $v(n)$ and $h(n)$, multiply the DFTs, and then compute the IDFT of the product.
- Use one period of the circular convolution so computed as the linear convolution of the original sequences $v(n)$ and $h(n)$.

The procedure described here works perfectly well for finite-duration sequences. But, what about linear filtering applications where the input signal is, for all practical purposes, of infinite duration? In situations of this kind, we may use two widely used techniques known as the *overlap-add* and *overlap-save* sectioning methods, which are described next.

Overlap-Add Method

The best way to explain the overlap-add method is by way of an example. Consider the sequences $v(n)$ and $h(n)$ shown in Fig. 1.4; it is assumed that the sequence $v(n)$ is effectively of “infinite” length, and the sequence $h(n)$ is of some finite length P . The sequence $v(n)$ is first sectioned into nonoverlapping blocks, each of length $Q = N - P$ for some predetermined N , as illustrated in Fig. 1.5(a). It may therefore be represented as the sum of shifted finite-duration sequences, as shown by

$$v(n) = \sum_{r=0}^{\infty} v_r(n) \quad (1.17)$$

where

$$v_r(n) = \begin{cases} v(n + rQ), & n = 0, 1, \dots, Q - 1 \\ 0, & \text{otherwise} \end{cases} \quad (1.18)$$

Next, each section is padded with $P - 1$ zero-valued samples to form one period of a periodic sequence, as illustrated in Fig. 1.5(a). We may thus describe the first section by writing

$$v_0(n) = \begin{cases} v(n), & n = 0, 1, \dots, N - P \\ 0, & n = N - P + 1, \dots, N - 1 \end{cases} \quad (1.19)$$

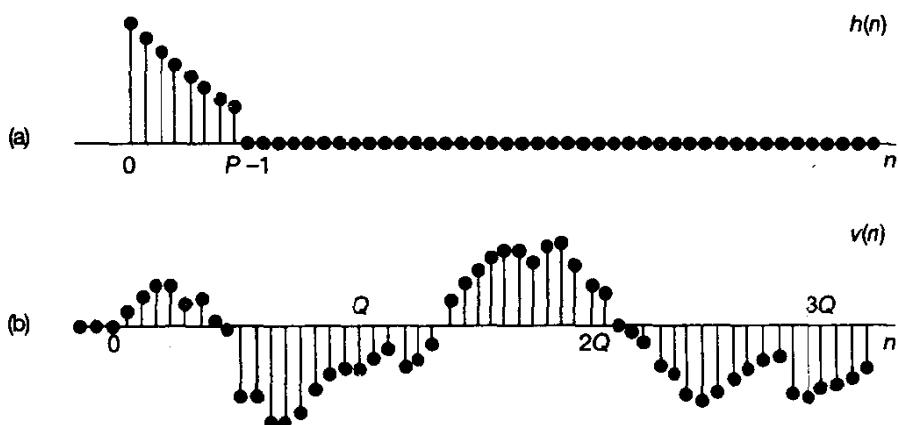


Figure 1.4 Finite-length impulse response $h(n)$ and indefinite-length signal $v(n)$ to be filtered by $h(n)$ (reproduced, with permission, from Oppenheim and Schafer, 1989).

The circular convolution of $v_0(n)$ with $h(n)$ yields the output sequence $u_0(n)$ shown in the first trace of Fig. 1.5(b).

The second section $v_1(n)$ and all other sections of the “infinitely” long sequence $v(n)$ are treated in a similar manner. The resulting output sequences $u_1(n)$, and $u_2(n)$ are also illustrated in Fig. 1.5(b) for the input sections $v_1(n)$ and $v_2(n)$, respectively. Finally, the output sequences $u_0(n)$, $u_1(n)$, $u_2(n)$, . . . are combined to yield the overall output sequence $u(n)$. Note that $u_1(n)$, $u_2(n)$, . . . are shifted by the appropriate values, namely, N , $2N$, . . . , before they are added to $u_0(n)$. The sectioned convolution technique described here is called the *overlap-add method* for two reasons: the output sequences tend to overlap each other, and they are added together to produce the correct result.

Overlap-Save Method

The overlap-save method differs from the overlap-add method in that it involves overlapping input sections rather than output sections. Specifically, the “infinitely” long sequence is sectioned into N -point blocks that overlap by $P - 1$ samples, where P is the length of the “short” sequence $h(n)$, as illustrated in Fig. 1.6(a). The N -point circular convolution of $h(n)$ and $v_r(n)$ is computed for $r = 0, 1, 2, \dots$. The resulting output sequences $u_0(n)$, $u_1(n)$, and $u_2(n)$ for the sections $v_0(n)$, $v_1(n)$, and $v_2(n)$ are illustrated in Fig. 1.6(b). The first $P - 1$ samples of each output sequence $u_r(n)$, $r = 0, 1, 2, \dots$ are ignored, because they are due to the wraparound (end) effect of the circular convolution. Finally, the remaining samples of the output sequences $u_0(n)$, $u_1(n)$, $u_2(n)$, . . . are added after they have been shifted by appropriate values, yielding the correct output sequence $u(n)$. For obvious reasons, this second sectioning technique is referred to as the *overlap-save method*.

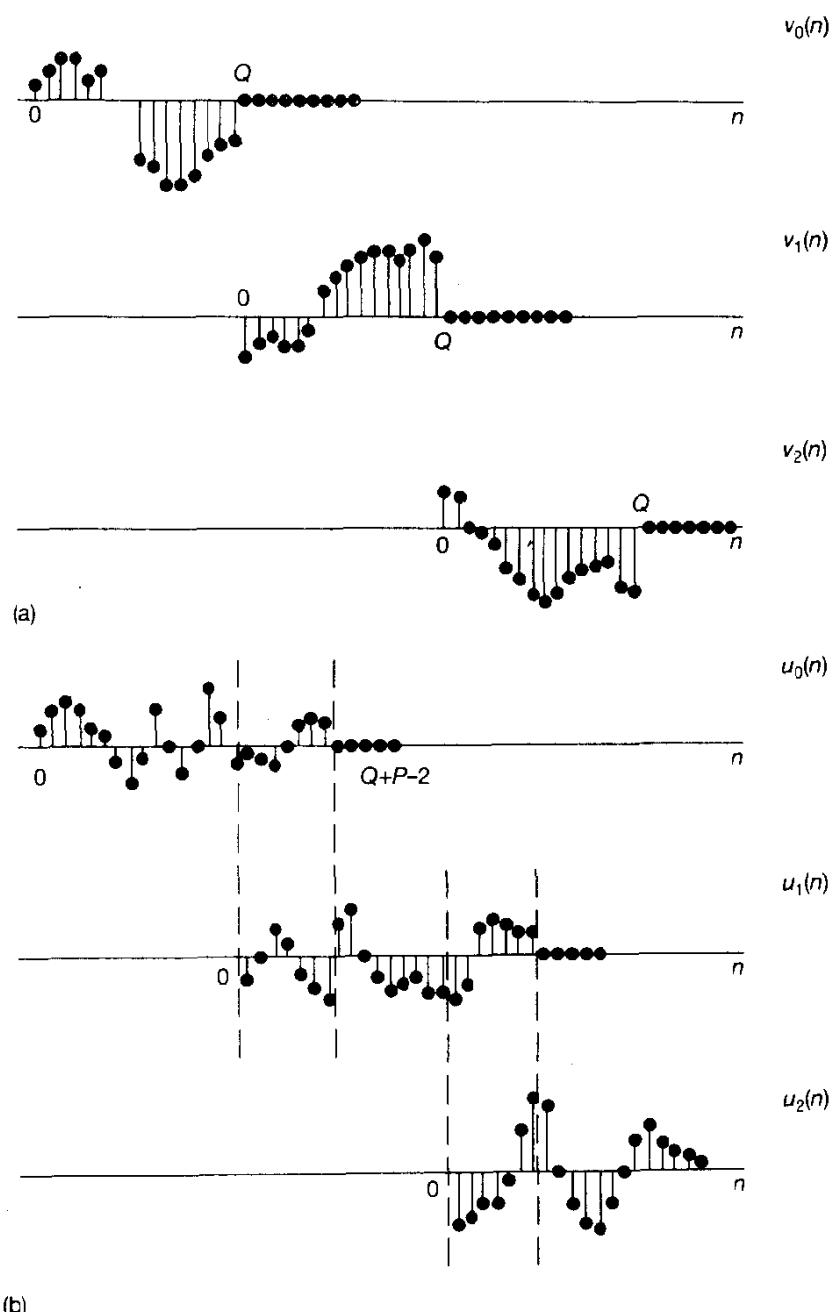
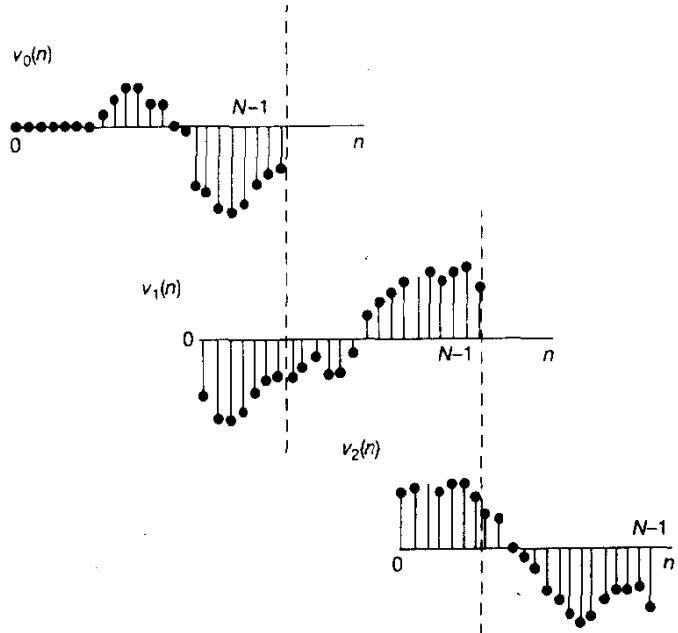
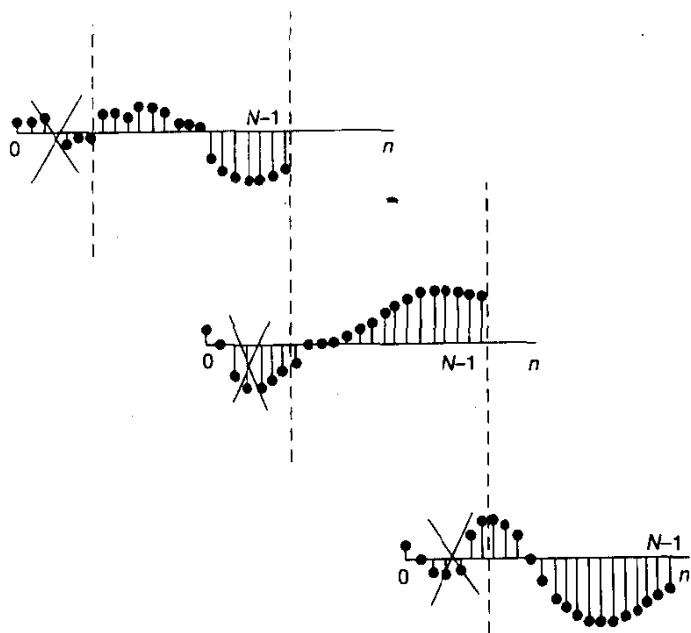


Figure 1.5 (a) Decomposition of the input signal $v(n)$ in Fig. 1.4 into nonoverlapping sections, each of length Q . (b) Result of convolving each such section with $h(n)$ (reproduced, with permission, from Oppenheim and Schafer, 1989).



(a)



(b)

Figure 1.6 (a) Decomposition of the input signal $v(n)$ in Fig. 1.4 into overlapping sections, each of length N . (b) Result of convolving each section with $h(n)$; the portions of each filtered section to be discarded in forming the linear convolution are indicated (reproduced, with permission, from Oppenheim and Schafer, 1989).

Thus, we may use the overlap-add method or the overlap-save method to compute the linear convolution of a short sequence $h(n)$ with a much longer sequence $v(n)$ by first sectioning the latter sequence into small blocks, then indirectly computing the circular convolution of each such block with the short sequence $h(n)$ via the DFT, and finally piecing the individual results together in an appropriate fashion. The utility of the overlap-add and overlap-save methods is made a practical reality by virtue of the availability of highly efficient algorithms (i.e., FFT algorithms) for computing the DFT. The indirect computation of convolution using the overlap-add method or overlap-save method via the FFT is referred to as *fast convolution*, as it is faster than its direct computation.

1.6 DISCRETE COSINE TRANSFORM

Another transform that features in certain applications of digital signal processing is the *discrete cosine transform* (DCT). Unlike the DFT, the DCT may be defined in several different ways (Rao and Yip, 1990). For the purpose of our present discussion, the DCT of an N -point sequence $u(n)$ is defined by

$$U(m) = k_m \sum_{n=0}^{N-1} u(n) \cos\left(\frac{(2n+1)m\pi}{2N}\right), \quad m = 0, 1, \dots, N-1 \quad (1.20)$$

and the *inverse discrete cosine transform* (IDCT) of $U(m)$ is defined by

$$u(n) = \frac{2}{N} \sum_{m=0}^{N-1} k_m U(m) \cos\left(\frac{(2n+1)m\pi}{2N}\right), \quad n = 0, 1, \dots, N-1 \quad (1.21)$$

The constant k_m in Eqs. (1.20) and (1.21) is itself defined by

$$k_m = \begin{cases} 1/\sqrt{2}, & m = 0 \\ 1, & m = 1, \dots, N-1 \end{cases} \quad (1.22)$$

The DCT is related to the DFT, as one would expect. Specifically, we first construct a $2N$ -point sequence $\tilde{u}(n)$ related to the original sequence $u(n)$ as follows:

$$\tilde{u}(n) = \begin{cases} u(n), & n = 0, 1, \dots, N-1 \\ u(2N-n-1), & n = N, N+1, \dots, 2N-1 \end{cases} \quad (1.23)$$

Thus, the $\tilde{u}(n)$ is an even extension of $u(n)$. The $2N$ -point DFT of the sequence $\tilde{u}(n)$ is given by

$$\begin{aligned} \tilde{U}(m) &= \sum_{n=0}^{2N-1} \tilde{u}(n) \exp\left(-\frac{j2\pi mn}{2N}\right) \\ &= \sum_{n=0}^{N-1} \tilde{u}(n) \exp\left(-\frac{j2\pi mn}{2N}\right) + \sum_{n=N}^{2N-1} \tilde{u}(n) \exp\left(-\frac{j2\pi mn}{2N}\right) \end{aligned} \quad (1.24)$$

Substituting Eq. (1.23) into (1.24), we get

$$\begin{aligned}\tilde{U}(m) &= \sum_{n=0}^{N-1} u(n) \exp\left(-\frac{j2\pi mn}{2N}\right) + \sum_{n=N}^{2N-1} u(2N-n-1) \exp\left(-\frac{j2\pi mn}{2N}\right) \\ &= \sum_{n=0}^{N-1} u(n) \left[\exp\left(-\frac{j2\pi mn}{2N}\right) + \exp\left(\frac{2j\pi m(n+1)}{2N}\right) \right]\end{aligned}\quad (1.25)$$

Introducing the phase shift $m\pi/2N$ and the weighting factor $k_m/2$ into Eq. (1.25), we may write

$$\begin{aligned}\frac{1}{2}k_m \exp\left(-\frac{jm\pi}{2N}\right)\tilde{U}(m) &= \frac{1}{2}k_m \sum_{n=0}^{N-1} u(n) \left[\exp\left(-\frac{j(2n+1)m\pi}{2N}\right) + \exp\left(\frac{j(2n+1)m\pi}{2N}\right) \right] \\ &= k_m \sum_{n=0}^{N-1} u(n) \cos\left(\frac{(2n+1)m\pi}{2N}\right)\end{aligned}\quad (1.26)$$

The right-hand side of Eq. (1.26) is recognized as the definition for the DCT of the original sequence $u(n)$. It follows, therefore, that the discrete cosine transform $U(m)$ of the sequence $u(n)$ and the discrete Fourier transform $\tilde{U}(m)$ of its extended version $\tilde{u}(n)$ are related as follows:

$$U(m) = \frac{1}{2}k_m \exp\left(-\frac{jm\pi}{2N}\right)\tilde{U}(m), \quad m = 0, 1, \dots, N-1 \quad (1.27)$$

This relation shows that, whereas the DFT is periodic with period N , the DCT is periodic with period $2N$.

1.7 SUMMARY AND DISCUSSION

In this chapter we reviewed the z -transform, the discrete Fourier transform, and the discrete cosine transform; these transforms are all related to each other. The discrete Fourier transform represents an important example of a general class of finite-length orthogonal transforms, which may be defined by the following pair of relations:

$$\begin{aligned}U(k) &= \sum_{n=0}^{N-1} u(n) \varphi_k^*(n), \quad k = 0, 1, \dots, N-1 \\ u(n) &= \frac{1}{N} \sum_{k=0}^{N-1} U(k) \varphi_k(n), \quad n = 0, 1, \dots, N-1\end{aligned}$$

where $u(n)$ is the given sequence and $U(k)$ is its discrete transform. The sequences $\varphi_k(n)$ for different k constitute an *orthogonal set*, as shown by

$$\sum_{n=0}^{N-1} \varphi_k(n) \varphi_l^*(n) = \begin{cases} N, & l = k \\ 0, & \text{otherwise} \end{cases}$$

Our interest in the z -transform is motivated by the fact it provides a basic tool for the characterization of linear time-invariant systems, which constitute an important class of systems of particular interest in the study of linear adaptive filtering. As for the discrete Fourier transform and the discrete cosine transform, they provide the necessary tools for the implementation of adaptive filtering in the frequency domain, which is sometimes found to be preferable to adaptive filtering in the time domain, an issue that is discussed later in this book.

PROBLEMS

1. An all-pole filter is characterized by the second-order difference equation:

$$u(n) - 0.1 u(n-1) - 0.8 u(n-2) = v(n)$$

- (a) Determine the transfer function $H(z)$ of the filter.
 - (b) Plot the pole-zero map of $H(z)$.
 - (c) Find the impulse response of the filter.
2. The inverse of the filter described in Problem 1 consists of an all-zero filter.
- (a) Plot the pole-zero map of the transfer function for this inverse filter.
 - (b) Find the difference equation that describes the time-domain behavior of the inverse filter.
 - (c) Find the impulse response of the inverse filter.
3. A second-order nonminimum phase system has the transfer function

$$H(z) = \frac{2(1+z^{-1}-2z^{-2})}{1-0.2828z^{-1}+z^{-2}}$$

- (a) Plot a pole-zero map for $H(z)$.
 - (b) The system described here may be considered to be equivalent to the cascade connection of a minimum phase system and an all-pass system. Determine the transfer functions of these two systems, and plot their individual pole-zero maps.
4. When considering the inverse of a minimum-phase system characterized by the transfer function $H(z)$, it is not permissible for $H(z)$ to have any zero on the unit circle in the z -plane. Why?
5. Convolution, be it linear or circular, is a commutative operation. Demonstrate this property.
6. If $U(e^{j2\pi f})$ is the Fourier transform of a finite-duration sequence $u(n)$, the Fourier transform of the time-shifted sequence $u(n-m)$ is $e^{-jm\pi f} U(e^{j2\pi f})$. How is the corresponding time-shifting property of the discrete Fourier transform for the sequence $u(n)$ described?

CHAPTER

2

Stationary Processes and Models

The term *stochastic process* or *random process* is used to describe the time evolution of a statistical phenomenon according to probabilistic laws. The time evolution of the phenomenon means that the stochastic process is a function of time, defined on some observation interval. The statistical nature of the phenomenon means that, before conducting an experiment, it is not possible to define exactly the way it evolves in time. Examples of a stochastic process include speech signals, television signals, radar signals, digital computer data, the output of a communication channel, seismological data, and noise.

The form of a stochastic process that is of interest to us is one that is defined at *discrete and uniformly spaced instants of time* (Box and Jenkins, 1976; Priestley, 1981). Such a restriction may arise naturally in practice, as in the case of radar signals or digital computer data. Alternatively, the stochastic process may be defined originally for a continuous range of real values of time; however, before processing, it is *sampled uniformly* in time, with the sampling rate chosen to be greater than twice the highest frequency component of the process (Haykin, 1994).

A stochastic process is *not* just a single function of time; rather, it represents, in theory, an infinite number of *different* realizations of the process. One particular realization of a discrete-time stochastic process is called a *discrete-time series* or simply *time series*. For convenience of notation, we *normalize time with respect to the sampling period*. For example, the sequence $u(n), u(n - 1), \dots, u(n - M)$ represents a time series that consists of the *present* observation $u(n)$ made at time n and M past observations of the process made at times $n - 1, \dots, n - M$.

We say that a stochastic process is *strictly stationary* if its statistical properties are invariant to a shift of time. Specifically, for a discrete-time stochastic process represented by the time series $u(n), u(n - 1), \dots, u(n - M)$ to be strictly stationary, the *joint probability density function* of these observations made at times $n, n - 1, \dots, n - M$ must remain the same no matter what values we assign to n for fixed M .

2.1 PARTIAL CHARACTERIZATION OF A DISCRETE-TIME STOCHASTIC PROCESS

In practice, we usually find that it is not possible to determine (by means of suitable measurements) the joint probability density function for an arbitrary set of observations made on a stochastic process. Accordingly, we must content ourselves with a partial characterization of the process by specifying its first and second moments.

Consider a discrete-time stochastic process represented by the time series $u(n), u(n - 1), \dots, u(n - M)$, which may be complex valued. We define the *mean-value function* of the process as

$$\mu(n) = E[u(n)] \quad (2.1)$$

where E denotes the *statistical expectation operator*. We define the *autocorrelation function* of the process as

$$r(n, n - k) = E[u(n)u^*(n - k)], \quad k = 0, \pm 1, \pm 2, \dots, \quad (2.2)$$

where the asterisk denotes *complex conjugation*. We define the *autocovariance function* of the process as

$$c(n, n - k) = E[(u(n) - \mu(n))(u(n - k) - \mu(n - k))^*], \quad k = 0, \pm 1, \pm 2, \dots \quad (2.3)$$

From Eqs. (2.1) to (2.3), we see that the mean-value, autocorrelation and autocovariance functions of the process are related by

$$c(n, n - k) = r(n, n - k) - \mu(n)\mu^*(n - k) \quad (2.4)$$

For a partial characterization of the process, we therefore need to specify (1) the mean-value function $\mu(n)$ and (2) the autocorrelation function $r(n, n - k)$ or the autocovariance function $c(n, n - k)$ for various values of n and k that are of interest. Note also the autocorrelation and autocovariance functions have the same value when the mean $\mu(n)$ is zero for all n .

This form of partial characterization offers two important advantages:

1. It lends itself to practical measurements.
2. It is well suited to *linear* operations on stochastic processes.

For a discrete-time stochastic process that is strictly stationary, all three quantities defined in Eqs. (2.1) to (2.3) assume simpler forms. In particular, we find that the mean-value

function of the process is a constant μ (say), so we may write

$$\mu(n) = \mu \quad \text{for all } n \quad (2.5)$$

We also find that both the autocorrelation and autocovariance functions depend only on the *difference* between the observation times n and $n - k$, that is, k , as shown by

$$r(n, n - k) = r(k) \quad (2.6)$$

and

$$c(n, n - k) = c(k) \quad (2.7)$$

Note that when $k = 0$, corresponding to a time difference or *lag* of zero, $r(0)$ equals the *mean-square value* of $u(n)$:

$$r(0) = E[|u(n)|^2] \quad (2.8)$$

and $c(0)$ equals the *variance* of $u(n)$:

$$c(0) = \sigma_u^2 \quad (2.9)$$

The conditions of Eqs. (2.5) to (2.7) are *not* sufficient to guarantee that the discrete-time stochastic process is strictly stationary. However, a discrete-time stochastic process that is not strictly stationary, but for which these conditions hold, is said to be *wide-sense stationary*, or *stationary to the second order*. A strictly stationary process $\{u(n)\}$, or $u(n)$ for short, is stationary in the wide sense if and only if (Doob, 1953)

$$E[|u(n)|^2] < \infty \quad \text{for all } n$$

This condition is ordinarily satisfied by stochastic processes encountered in the physical sciences and engineering.

2.2 MEAN ERGODIC THEOREM

The *expectations* or *ensemble averages* of a stochastic process are averages "across the process." Clearly, we may also define *long-term sample averages* or *time averages* that are averages "along the process." Indeed, time averages may be used to build a *stochastic model* of a physical process by *estimating* unknown parameters of the model. For such an approach to be rigorous, however, we have to show that time averages converge to corresponding ensemble averages of the process in some statistical sense. A popular criterion for convergence is that of mean square-error, as described next.

To be specific, consider a discrete-time stochastic process $u(n)$ that is wide-sense stationary. Let a constant μ denote the mean of the process, and $c(k)$ denote its autocovariance function for lag k . For an estimate of the mean μ , we may use the time average

$$\hat{\mu}(N) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) \quad (2.10)$$

where N is the total number of samples used in the estimation. Note that the estimate $\hat{\mu}(N)$ is a random variable with a mean and variance of its own. In particular, we readily find from Eq. (2.10) that the mean (expectation) of $\hat{\mu}(N)$ is

$$E[\hat{\mu}(N)] = \mu \quad \text{for all } N \quad (2.11)$$

It is in the sense of Eq. (2.11) that we say the time average $\hat{\mu}(N)$ is an *unbiased estimator* of the ensemble average (mean) of the process.

Moreover, we say that the process $u(n)$ is *mean ergodic in the mean-square error sense* if the mean-square value of the error between the ensemble average μ and the time average $\hat{\mu}(N)$ approaches zero as the number of samples N approaches infinity; that is,

$$\lim_{N \rightarrow \infty} [(\mu - \hat{\mu}(N))^2] = 0$$

Using the time average formula of Eq. (2.10), we may write

$$\begin{aligned} E[|\mu - \hat{\mu}(N)|^2] &= E\left[\left|\mu - \frac{1}{N} \sum_{n=0}^{N-1} u(n)\right|^2\right] \\ &= \frac{1}{N^2} E\left[\left|\sum_{n=0}^{N-1} (u(n) - \mu)\right|^2\right] \\ &= \frac{1}{N^2} E\left[\sum_{n=0}^{N-1} \sum_{k=0}^{N-1} (u(n) - \mu)(u(k) - \mu)^*\right] \quad (2.12) \\ &= \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} E[(u(n) - \mu)(u(k) - \mu)^*] \\ &= \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} c(n-k) \end{aligned}$$

Let $l = n - k$. We may then simplify the double summation in Eq. (2.12) as follows:

$$E[|\mu - \hat{\mu}(N)|^2] = \frac{1}{N} \sum_{l=-N+1}^{N-1} \left(1 - \frac{|l|}{N}\right) c(l)$$

Accordingly, we may state that the necessary and sufficient condition for the process $u(n)$ to be mean ergodic in the mean-square error sense is that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=-N+1}^{N-1} \left(1 - \frac{|l|}{N}\right) c(l) = 0 \quad (2.13)$$

In other words, if the process $u(n)$ is asymptotically uncorrelated in the sense of Eq. (2.13), then the time average $\hat{\mu}(N)$ of the process converges to the ensemble average μ in

the mean-square error sense. This is the statement of a particular form of the *mean ergodic theorem* (Gray and Davisson, 1986).

The use of the mean ergodic theorem may be extended to other time averages of the process. Consider, for example, the following time average used to estimate the autocorrelation function of a wide-sense stationary process:

$$\hat{r}(k, N) = \frac{1}{N} \sum_{n=0}^{N-1} u(n)u(n-k), \quad 0 \leq k \leq N-1 \quad (2.14)$$

The process $u(n)$ is said to be *correlation ergodic* in the mean-square error sense if the mean-square value of the difference between the true value $r(k)$ and the estimate $\hat{r}(k, N)$ approaches zero as the number of samples N approaches infinity. Let $z(n, k)$ denote a new discrete-time stochastic process related to the original process $u(n)$ as follows:

$$z(n, k) = u(n)u(n-k) \quad (2.15)$$

Hence, by substituting $z(n, k)$ for $u(n)$, we may use the mean ergodic theorem to establish the conditions for $z(n, k)$ to be mean ergodic or, equivalently, for $u(n)$ to be correlation ergodic.

2.3 CORRELATION MATRIX

Let the M -by-1 *observation vector* $\mathbf{u}(n)$ represent the elements of the time series $u(n)$, $u(n-1), \dots, u(n-M+1)$. To show the composition of the vector $\mathbf{u}(n)$ explicitly, we write

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T \quad (2.16)$$

where the superscript T denotes *transposition*. We define the *correlation matrix* of a stationary discrete-time stochastic process represented by this time series as the *expectation of the outer product of the observation vector $\mathbf{u}(n)$ with itself*. Let \mathbf{R} denote the M -by- M correlation matrix defined in this way. We thus write

$$\mathbf{R} = E[\mathbf{u}(n) \mathbf{u}^H(n)] \quad (2.17)$$

where the superscript H denotes *Hermitian transposition* (i.e., the operation of transposition combined with complex conjugation). By substituting Eq. (2.16) in (2.17) and using the condition of wide-sense stationarity, we may express the correlation matrix \mathbf{R} in the expanded form:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(-1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ r(-M+1) & r(-M+2) & \cdots & r(0) \end{bmatrix} \quad (2.18)$$

The element $r(0)$ on the main diagonal is always real valued. For complex-valued data, the remaining elements of \mathbf{R} assume complex values.

Properties of the Correlation Matrix

The correlation matrix \mathbf{R} plays a key role in the statistical analysis and design of discrete-time filters. It is therefore important that we understand its various properties and their implications. In particular, using the definition of Eq. (2.17), we find that the correlation matrix of a stationary discrete-time stochastic process has the following properties.

Property 1. *The correlation matrix of a stationary discrete-time stochastic process is Hermitian.*

We say that a *complex-valued* matrix is *Hermitian* if it is equal to its *conjugate transpose*. We may thus express the Hermitian property of the correlation matrix \mathbf{R} by writing

$$\mathbf{R}^H = \mathbf{R} \quad (2.19)$$

This property follows directly from the definition of Eq. (2.17).

Another way of stating the Hermitian property of the correlation matrix \mathbf{R} is to write

$$r(-k) = r^*(k) \quad (2.20)$$

where $r(k)$ is the autocorrelation function of the stochastic process $u(n)$ for a lag of k . Accordingly, for a wide-sense stationary process we only need M values of the autocorrelation function $r(k)$ for $k = 0, 1, \dots, M - 1$ in order to completely define the correlation matrix \mathbf{R} . We may thus rewrite Eq. (2.18) as follows:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix} \quad (2.21)$$

From here on, we will use this representation for the expanded matrix form of the correlation matrix of a wide-sense stationary discrete-time stochastic process. Note that for the special case of *real-valued data*, the autocorrelation function $r(k)$ is real for all k , and the correlation matrix \mathbf{R} is *symmetric*.

Property 2. *The correlation matrix of a stationary discrete-time stochastic process is Toeplitz.*

We say that a square matrix is *Toeplitz* if all the elements on its main diagonal are equal, and if the elements on any other diagonal parallel to the main diagonal are also equal. From the expanded form of the correlation matrix \mathbf{R} given in Eq. (2.21), we see that all the elements on the main diagonal are equal to $r(0)$, all the elements on the first diagonal above the main diagonal are equal to $r(1)$, all the elements along the first diagonal

below the main diagonal are equal to $r^*(1)$, and so on for the other diagonals. We conclude therefore that the correlation matrix \mathbf{R} is Toeplitz.

It is important to recognize, however, that the Toeplitz property of the correlation matrix \mathbf{R} is a direct consequence of the assumption that the discrete-time stochastic process represented by the observation vector $\mathbf{u}(n)$ is wide-sense stationary. Indeed, we may state that if the discrete-time stochastic process is wide-sense stationary, then its correlation matrix \mathbf{R} must be Toeplitz; and, conversely, if the correlation matrix \mathbf{R} is Toeplitz, then the discrete-time stochastic process must be wide-sense stationary.

Property 3. *The correlation matrix of a discrete-time stochastic process is always nonnegative definite and almost always positive definite.*

Let \mathbf{x} be an arbitrary (nonzero) M -by-1 complex-valued vector. Define the scalar random variable y as the *inner product* of \mathbf{x} and the observation vector $\mathbf{u}(n)$, as shown by

$$y = \mathbf{x}^H \mathbf{u}(n)$$

Taking the Hermitian transpose of both sides and recognizing that y is a scalar, we get

$$y^* = \mathbf{u}^H(n) \mathbf{x}$$

where the asterisk denotes *complex conjugation*. The mean-square value of the random variable y equals

$$\begin{aligned} E[|y|^2] &= E[yy^*] \\ &= E[\mathbf{x}^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{x}] \\ &= \mathbf{x}^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{x} \\ &= \mathbf{x}^H \mathbf{R} \mathbf{x} \end{aligned}$$

where \mathbf{R} is the correlation matrix defined in Eq. (2.17). The expression $\mathbf{x}^H \mathbf{R} \mathbf{x}$ is called a Hermitian form. Since

$$E[|y|^2] \geq 0$$

it follows that

$$\mathbf{x}^H \mathbf{R} \mathbf{x} \geq 0 \quad (2.22)$$

A Hermitian form that satisfies this condition for every nonzero \mathbf{x} is said to be *nonnegative definite* or *positive semidefinite*. Accordingly, we may state that the correlation matrix of a wide-sense stationary process is always nonnegative definite.

If the Hermitian form $\mathbf{x}^H \mathbf{R} \mathbf{x}$ satisfies the condition

$$\mathbf{x}^H \mathbf{R} \mathbf{x} > 0$$

for every nonzero \mathbf{x} , we say that the correlation matrix \mathbf{R} is *positive definite*. This condition is satisfied for a wide-sense stationary process unless there are linear dependencies between the random variables that constitute the M elements of the observation vector

$\mathbf{u}(n)$. Such a situation arises essentially only when the process $\mathbf{u}(n)$ consists of the sum of K sinusoids with $K \leq M$; see Section 2.4 for more details. In practice, we find that this idealized situation is so rare in occurrence that the correlation matrix \mathbf{R} is almost always positive definite.

The positive definiteness of a correlation matrix implies that its determinant and all principal minors are greater than zero. For example, for $M = 2$, we must have

$$\begin{vmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{vmatrix} > 0$$

Similarly, for $M = 3$, we must have

$$\begin{vmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{vmatrix} > 0$$

$$\begin{vmatrix} r(0) & r(2) \\ r^*(2) & r(0) \end{vmatrix} > 0$$

$$\begin{vmatrix} r(0) & r(1) & r(2) \\ r^*(1) & r(0) & r(1) \\ r^*(2) & r^*(1) & r(0) \end{vmatrix} > 0$$

and so on for higher values of M . These conditions, in turn, imply that the correlation matrix is nonsingular. We say that a matrix is *nonsingular* if its inverse exists; otherwise, it is singular. Accordingly, we may state that a correlation matrix is almost always nonsingular.

Property 4. *When the elements that constitute the observation vector of a stationary discrete-time stochastic process are rearranged backward, the effect is equivalent to the transposition of the correlation matrix of the process.*

Let $\mathbf{u}^B(n)$ denote the M -by-1 vector obtained by rearranging the elements that constitute the observation vector $\mathbf{u}(n)$ *backward*. We illustrate this operation by writing

$$\mathbf{u}^{BT}(n) = [u(n - M + 1), u(n - M + 2), \dots, u(n)] \quad (2.23)$$

where the superscript B denotes the backward rearrangement of a vector. The correlation matrix of the vector $\mathbf{u}^B(n)$ equals, by definition,

$$E[\mathbf{u}^B(n)\mathbf{u}^{BH}(n)] = \begin{bmatrix} r(0) & r^*(1) & \cdots & r^*(M-1) \\ r(1) & r(0) & \cdots & r^*(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & r(M-2) & \cdots & r(0) \end{bmatrix} \quad (2.24)$$

Hence, comparing the expanded correlation matrix of Eq. (2.24) with that of Eq. (2.21), we see that

$$E [\mathbf{u}^B(n) \mathbf{u}^{BH}(n)] = \mathbf{R}^T \quad (2.25)$$

which is the desired result.

Property 5. *The correlation matrices \mathbf{R}_M and \mathbf{R}_{M+1} of a stationary discrete-time stochastic process, pertaining to M and $M + 1$ observations of the process, respectively, are related by*

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & \mathbf{r}^H \\ \vdots & \ddots \\ \mathbf{r} & \mathbf{R}_M \end{bmatrix} \quad (2.26)$$

or equivalently,

$$\mathbf{R}_{M+1} = \begin{bmatrix} \mathbf{R}_M & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \quad (2.27)$$

where $r(0)$ is the autocorrelation of the process for a lag of zero, and

$$\mathbf{r}^H = [r(1), r(2), \dots, r(M)] \quad (2.28)$$

and

$$\mathbf{r}^{BT} = [r(-M), r(-M + 1), \dots, r(-1)] \quad (2.29)$$

Note that in describing Property 5 we have added a subscript, M or $M + 1$, to the symbol for the correlation matrix in order to display dependence on the number of observations used to define this matrix. We follow such a practice (in the context of the correlation matrix and other vector quantities) *only* when the issue at hand involves dependence on the number of observations or dimensions of the matrix.

To prove the relation of Eq. (2.26), we express the correlation matrix \mathbf{R}_{M+1} in its expanded form, partitioned as follows:

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(M) \\ r^*(1) & r(0) & r(1) & \cdots & r(M - 1) \\ r^*(2) & r^*(1) & r(0) & \cdots & r(M - 2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r^*(M) & r^*(M - 1) & r^*(M - 2) & \cdots & r(0) \end{bmatrix} \quad (2.30)$$

Using Eqs. (2.18), (2.20), and (2.28) in (2.30), we get the result given in Eq. (2.26). Note that according to this relation, the observation vector $\mathbf{u}_{M+1}(n)$ is *partitioned* in the form

$$\begin{aligned}\mathbf{u}_{M+1}(n) &= \begin{bmatrix} u(n) \\ \vdots \\ u(n-1) \\ u(n-2) \\ \vdots \\ \vdots \\ u(n-M) \end{bmatrix} \\ &= \begin{bmatrix} u(n) \\ \vdots \\ \mathbf{u}_M(n-1) \end{bmatrix} \quad (2.31)\end{aligned}$$

where the subscript $M + 1$ is intended to denote the fact that the vector $\mathbf{u}_{M+1}(n)$ has $M + 1$ elements, and likewise for $\mathbf{u}_M(n)$.

To prove the relation of Eq. (2.27), we express the correlation matrix \mathbf{R}_{M+1} in its expanded form, partitioned in the alternative form

$$\mathbf{R}_{M+1} = \left[\begin{array}{ccccc} r(0) & r(1) & \cdots & r(M-1) & r(M) \\ r^*(1) & r(0) & \cdots & r(M-2) & r(M-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) & r(1) \\ \hline r^*(M) & r^*(M-1) & \cdots & r^*(1) & r(0) \end{array} \right] \quad (2.32)$$

Here again, using Eqs. (2.18), (2.20), and (2.29) in (2.32), we get the result given in Eq. (2.27). Note that according to this second relation the observation vector $\mathbf{u}_{M+1}(n)$ is partitioned in the alternative form

$$\begin{aligned}\mathbf{u}_{M+1}(n) &= \begin{bmatrix} u(n) \\ u(n-1) \\ \vdots \\ \vdots \\ u(n-M+1) \\ \vdots \\ u(n-M) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{u}_M(n) \\ \vdots \\ u(n-M) \end{bmatrix} \quad (2.33)\end{aligned}$$

2.4 CORRELATION MATRIX OF SINE WAVE PLUS NOISE

A time series of special interest is one that consists of a *complex sinusoid corrupted by additive noise*. Such a time series is representative of several important signal-processing applications. In the *temporal context*, for example, this time series represents the composite signal at the input of a receiver, with the complex sinusoid representing a *target signal* and the noise representing thermal noise generated at the front end of the receiver. In the *spatial context*, it represents the received signal in a linear array of sensors, with the complex sinusoid representing a *plane wave* produced by a remote source (emitter) and the noise representing *sensor noise*.

Let α denote the amplitude of the complex sinusoid, and ω denote its angular frequency. Let $v(n)$ denote a sample of the noise, assumed to have zero mean. We may then write a corresponding sample of the time series that consists of the complex sinusoid plus noise as follows:

$$u(n) = \alpha \exp(j\omega n) + v(n), \quad n = 0, 1, \dots, N - 1 \quad (2.34)$$

The sources of the complex sinusoid and the noise are independent of each other. Since the noise component $v(n)$ has zero mean, by assumption, we see from Eq. (2.34) that the mean of $u(n)$ is equal to $\alpha \exp(j\omega n)$.

To calculate the autocorrelation function of the process $u(n)$, we clearly need to know the autocorrelation function of the noise process $v(n)$. To proceed then, we assume a special form of noise characterized by the autocorrelation function

$$E[v(n)v^*(n - k)] = \begin{cases} \sigma_v^2, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.35)$$

Such a form of noise is commonly referred to as *white noise*; more will be said about it in Chapter 3. Since the sources responsible for the generation of the complex sinusoid and the noise are independent and, therefore, uncorrelated, it follows that the autocorrelation function of the process $u(n)$ equals the sum of the autocorrelation functions of its two individual components. Accordingly, using Eqs. (2.34) and (2.35), we find that the autocorrelation function of the process $u(n)$ for a lag k is given by

$$\begin{aligned} r(k) &= E[u(n)u^*(n - k)] \\ &= \begin{cases} |\alpha|^2 + \sigma_v^2, & k = 0 \\ |\alpha|^2 \exp(j\omega k), & k \neq 0 \end{cases} \end{aligned} \quad (2.36)$$

where $|\alpha|$ is the magnitude of the complex amplitude α . Note that for a lag $k \neq 0$, the autocorrelation function $r(k)$ varies with k in the same sinusoidal fashion as the sample $u(n)$ varies with n , except for a change in amplitude. Given the series of samples $u(n), u(n - 1), \dots, u(n - M + 1)$, we may thus express the correlation matrix of $u(n)$ as

$$\mathbf{R} = |\alpha|^2 \begin{bmatrix} 1 + \frac{1}{\rho} & \exp(j\omega) & \cdots & \exp(j\omega(M-1)) \\ \exp(-j\omega) & 1 + \frac{1}{\rho} & \cdots & \exp(j\omega(M-2)) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \exp(-j\omega(M-1)) & \exp(-j\omega(M-2)) & \cdots & 1 + \frac{1}{\rho} \end{bmatrix} \quad (2.37)$$

where ρ is the *signal-to-noise ratio*, defined by

$$\rho = \frac{|\alpha|^2}{\sigma_v^2} \quad (2.38)$$

The correlation matrix \mathbf{R} of Eq. (2.37) has all of the properties described in Section 2.3; the reader is invited to verify them.

Equation (2.36) provides the mathematical basis of a two-step practical procedure for estimating the parameters of a complex sinusoid in the presence of additive noise:

1. Measure the mean-square value $r(0)$ of the process $u(n)$. Hence, given the noise variance σ_v^2 , determine the magnitude $|\alpha|$.
2. Measure the autocorrelation function $r(k)$ of the process $u(n)$ for a lag $k \neq 0$. Hence, given $|\alpha|^2$ from step 1, determine the angular frequency ω .

Note that this estimation procedure is *invariant to the phase of α* , which is a direct consequence of the definition of the autocorrelation function $r(k)$.

Example 1

Consider the idealized case of a noiseless sinusoid of angular frequency ω . For the purpose of illustration, we assume that the time series of interest consists of three uniformly spaced samples drawn from this sinusoid. Hence, setting the signal-to-noise ratio $\rho = \infty$ and the number of samples $M = 3$, we find from Eq. (2.37) that the correlation matrix of the time series so obtained has the following value:

$$\mathbf{R} = |\alpha|^2 \begin{bmatrix} 1 & \exp(j\omega) & \exp(j2\omega) \\ \exp(-j\omega) & 1 & \exp(j\omega) \\ \exp(-j2\omega) & \exp(-j\omega) & 1 \end{bmatrix}$$

From this expression we readily see that the determinant of \mathbf{R} and all principal minors are identically zero. Hence, this correlation matrix is singular.

We may generalize the result of this example by stating that when a process $u(n)$ consists of M samples drawn from the sum of K sinusoids with $K < M$ and there is *no* additive noise, then the correlation matrix of that process is singular.

2.5 STOCHASTIC MODELS

The term *model* is used for any hypothesis that may be applied to explain or describe the hidden laws that are supposed to govern or constrain the generation of physical data of interest. The representation of a stochastic process by a model dates back to an idea by Yule (1927). The idea is that a time series $u(n)$, consisting of highly correlated observations, may be generated by applying a series of statistically independent "shocks" to a linear filter, as in Fig. 2.1. The shocks are random variables drawn from a fixed distribution that is usually assumed to be *Gaussian* with zero mean and constant variance. Such a series of random variables constitutes a purely random process, commonly referred to as *white Gaussian noise*. Specifically, we may describe the input $v(n)$ in Figure 2.1 in statistical terms as follows:

$$E[v(n)] = 0 \quad \text{for all } n \quad (2.39)$$

and

$$E[v(n)v^*(k)] = \begin{cases} \sigma_v^2, & k = n \\ 0, & \text{otherwise} \end{cases} \quad (2.40)$$

where σ_v^2 is the noise variance. Equation (2.39) follows from the zero-mean assumption, and Eq. (2.40) follows from the white assumption. The implication of the Gaussian assumption is discussed in Section 2.11.

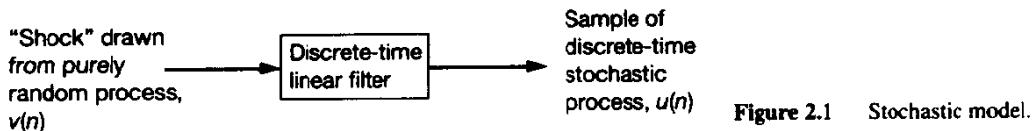
In general, the time-domain description of the input-output relation for the stochastic model of Fig. 2.1 may be described as follows:

$$\left(\begin{array}{c} \text{present value} \\ \text{of model output} \end{array} \right) + \left(\begin{array}{c} \text{linear combination} \\ \text{of past values} \\ \text{of model output} \end{array} \right) = \left(\begin{array}{c} \text{linear combination of} \\ \text{present and past values} \\ \text{of model input} \end{array} \right) \quad (2.41)$$

A stochastic process so described is referred to as a *linear process*.

The structure of the linear filter in Fig. 2.1 is determined by the manner in which the two linear combinations indicated in Eq. (2.41) are formulated. We may thus identify three popular types of linear stochastic models:

1. Autoregressive models, in which no past values of the model input are used.
2. Moving average models, in which no past values of the model output are used.
3. Mixed autoregressive-moving average models, in which the description of Eq. (2.41) applies in its entire form. Hence, this class of stochastic models includes autoregressive and moving average models as special cases.



These models are described next, in that order.

Autoregressive Models

We say that the time series $u(n), u(n - 1), \dots, u(n - M)$ represents the realization of an *autoregressive process (AR) of order M* if it satisfies the difference equation

$$u(n) + a_1^* u(n - 1) + \dots + a_M^* u(n - M) = v(n) \quad (2.42)$$

where a_1, a_2, \dots, a_M are constants called the AR *parameters*, and $v(n)$ is a white-noise process. The term $a_k^* u(n - k)$ is the scalar version of *inner product* of a_k and $u(n - k)$, where $k = 1, \dots, M$.

To explain the reason for the term “autoregressive,” we rewrite Eq. (2.42) in the form

$$u(n) = w_1^* u(n - 1) + w_2^* u(n - 2) + \dots + w_M^* u(n - M) + v(n) \quad (2.43)$$

where $w_k = -a_k$. We thus see that the present value of the process, that is, $u(n)$, equals a *finite linear combination of past values* of the process, $u(n - 1), \dots, u(n - M)$, plus an *error term* $v(n)$. We now see the reason for the term “autoregressive.” Specifically, a linear model

$$y = \sum_{k=1}^M w_k^* x_k + v$$

relating a *dependent variable* y to a set of *independent variables* x_1, x_2, \dots, x_M plus an error term v is often referred to as a *regression model*, and y is said to be “regressed” on x_1, x_2, \dots, x_M . In Eq. (2.43), the variable $u(n)$ is *regressed* on previous values of *itself*; hence the term “autoregressive.”

The left-hand side of Eq. (2.42) represents the *convolution* of the input sequence $u(n)$ and the sequence of parameters a_n^* . To highlight this point, we rewrite Eq. (2.42) in the form of a convolution sum:

$$\sum_{k=0}^M a_k^* u(n - k) = v(n) \quad (2.44)$$

where $a_0 = 1$. By taking the *z-transform* of both sides of Eq. (2.44), we transform the convolution sum on the left-hand side of the equation into a multiplication of the z-transforms of the two sequences $u(n)$ and a_n^* . Let $H_A(z)$ denote the z-transform of the sequence a_n^* :

$$H_A(z) = \sum_{n=0}^M a_n^* z^{-n} \quad (2.45)$$

Let $U(z)$ denote the z-transform of the input sequence $u(n)$:

$$U(z) = \sum_{n=0}^{\infty} u(n) z^{-n} \quad (2.46)$$

where z is a *complex variable*. We may thus transform the difference equation (2.42) into the equivalent form

$$H_A(z)U(z) = V(z) \quad (2.47)$$

where

$$V(z) = \sum_{n=0}^{\infty} v(n)z^{-n} \quad (2.48)$$

The z -transform of Eq. (2.47) offers two interpretations, depending on whether the AR process $u(n)$ is viewed as the input or output of interest:

1. Given the AR process $u(n)$, we may use the filter shown in Fig. 2.2(a) to produce the white noise process $v(n)$ as output. The parameters of this filter bear a one-to-one correspondence with those of the AR process $u(n)$. Accordingly, this filter represents a *process analyzer* with discrete transfer function $H_A(z) = V(z)/U(z)$. The impulse response of the AR process analyzer, that is, the inverse z -transform of $H_A(z)$, has *finite duration*.
2. With the white noise $v(n)$ acting as input, we may use the filter shown in Fig. 2.2(b) to produce the AR process $u(n)$ as output. Accordingly, this second filter represents a *process generator*, whose transfer function equals

$$\begin{aligned} H_G(z) &= \frac{U(z)}{V(z)} \\ &= \frac{1}{H_A(z)} \\ &= \frac{1}{\sum_{n=0}^{M} a_n^* z^{-n}} \end{aligned} \quad (2.49)$$

The impulse response of the AR process generator, that is, the inverse z -transform of $H_G(z)$, has *infinite duration*.

The AR process analyzer of Fig. 2.2(a) is an *all-zero filter*. It is so called because its transfer function $H_A(z)$ is completely defined by specifying the locations of its *zeros*. This filter is inherently stable.

The AR process generator of Fig. 2.2(b) is an *all-pole filter*. It is so called because its transfer function $H_G(z)$ is completely defined by specifying the locations of its *poles*, as shown by

$$H_G(z) = \frac{1}{(1 - p_1 z^{-1})(1 - p_2 z^{-1}) \cdots (1 - p_M z^{-1})} \quad (2.50)$$

The parameters p_1, p_2, \dots, p_M are *poles* of $H_G(z)$; they are defined by the roots of the *characteristic equation*

$$1 + a_1^* z^{-1} + a_2^* z^{-2} + \cdots + a_M^* z^{-M} = 0 \quad (2.51)$$

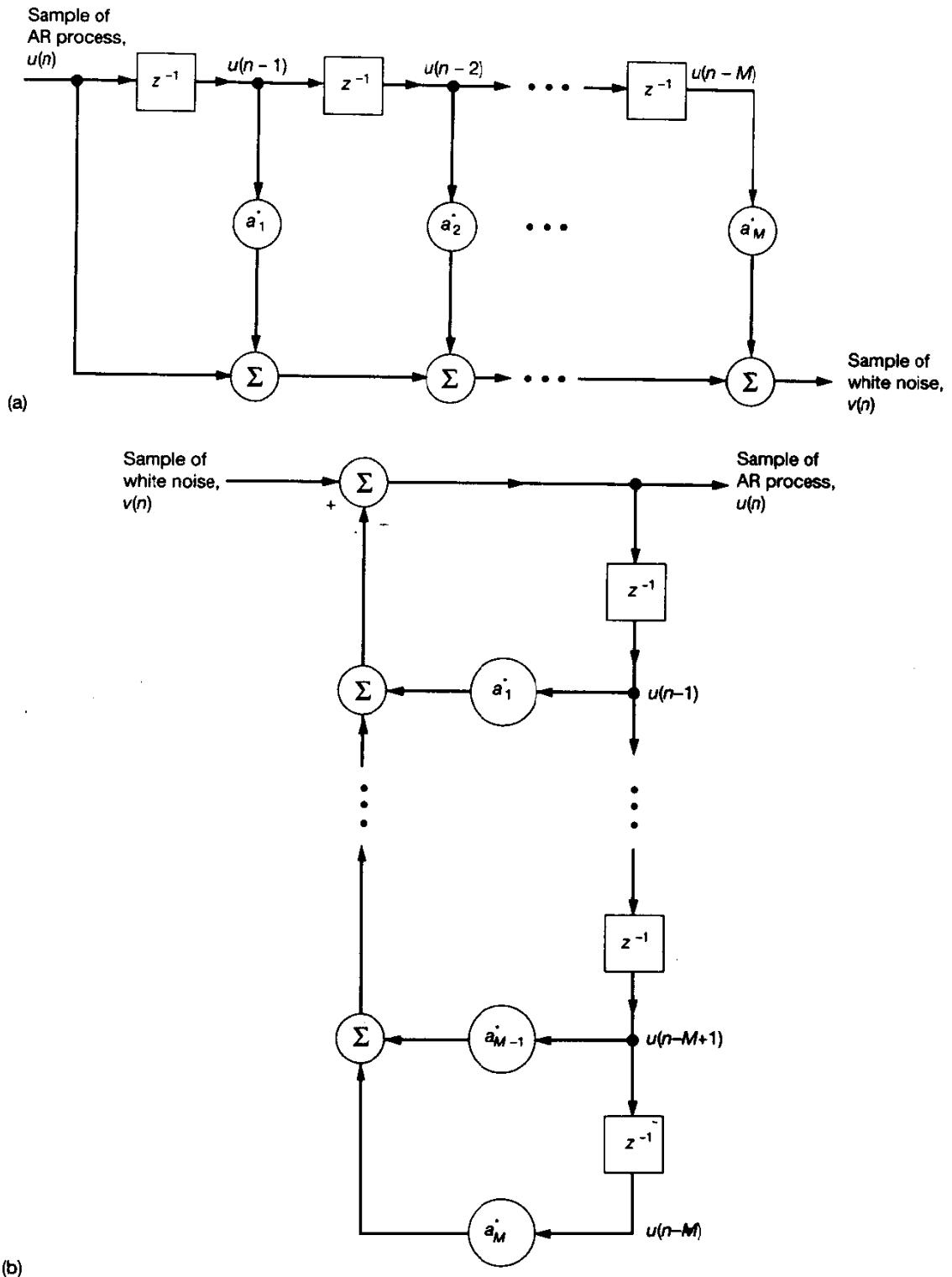


Figure 2.2 (a) AR process analyzer; (b) AR process generator.

For the all-pole AR process generator of Fig. 2.2(b) to be stable, the roots of the characteristic equation (2.51) must all lie inside the unit circle in the z -plane. This is also a necessary and sufficient condition for wide-sense stationarity of the AR process produced by the model of Fig. 2.2(b). We have more to say on the issue of stationarity in Section 2.7.

Moving Average Models

In a *moving average (MA) model*, the discrete-time linear filter of Fig. 2.1 consists of an *all-zero filter* driven by white noise. The resulting process $u(n)$, produced at the filter output, is described by the difference equation:

$$u(n) = v(n) + b_1^*v(n-1) + \cdots + b_K^*v(n-K) \quad (2.52)$$

where b_1, \dots, b_K are constants called the *MA parameters*, and $v(n)$ is a white-noise process of zero mean and variance σ_v^2 . Except for $v(n)$, each term on the right-hand side of Eq. (2.52) represents the scalar version of an inner product. The *order* of the MA process equals K . The term moving average is a rather quaint one; nevertheless, its use is firmly established in the literature. Its usage arose in the following way: If we are given a complete temporal realization of the white-noise process $v(n)$, we may compute $u(n)$ by constructing a *weighted average* of the sample values $v(n), v(n-1), \dots, v(n-K)$.

From Eq. (2.52), we readily obtain the MA model (i.e., process-generator) depicted in Fig. 2.3. Specifically, we start with a white-noise process $v(n)$ at the model input and generate an MA process $u(n)$ of order K at the model output. To proceed in the reverse manner, that is, to produce the white-noise process $v(n)$, given the MA process $u(n)$, we require the use of an *all-pole filter*. In other words, the filters used in the generation and analysis of an MA process are the *opposite* of those used in the case of an AR process.

Autoregressive–Moving Average Models

To generate a mixed *autoregressive–moving average (ARMA) process* $u(n)$, we use a discrete-time linear filter in Fig. 2.1 with a transfer function that contains *both poles and zeros*. Accordingly, given a white-noise process $v(n)$ as the filter input, the ARMA process $u(n)$ produced at the filter output is described by the difference equation

$$u(n) + a_1^*u(n-1) + \cdots + a_M^*u(n-M) = v(n) + b_1^*v(n-1) + \cdots + b_K^*v(n-K) \quad (2.53)$$

where a_1, \dots, a_M and b_1, \dots, b_K are called the *ARMA parameters*. Except for $u(n)$ on the left-hand side and $v(n)$ on the right-hand side of Eq. (2.53), all of the terms represent scalar versions of inner products. The *order* of the ARMA process equals (M, K) .

From Eq. (2.53), we readily deduce the ARMA model (i.e., process generator) depicted in Fig. 2.4. Comparing this figure with Figs. 2.2(b) and 2.3, we clearly see that AR and MA models are indeed special cases of an ARMA model.

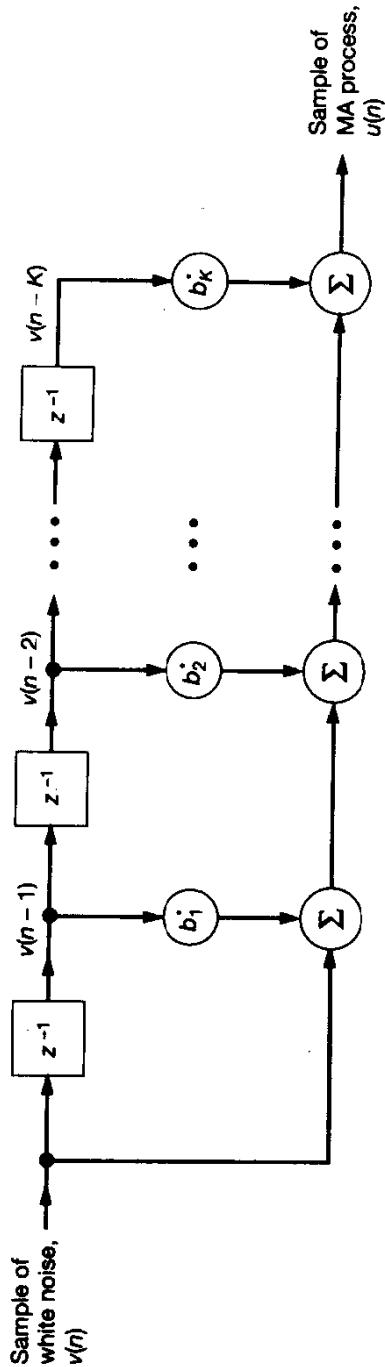


Figure 2.3 Moving average model (process generator).

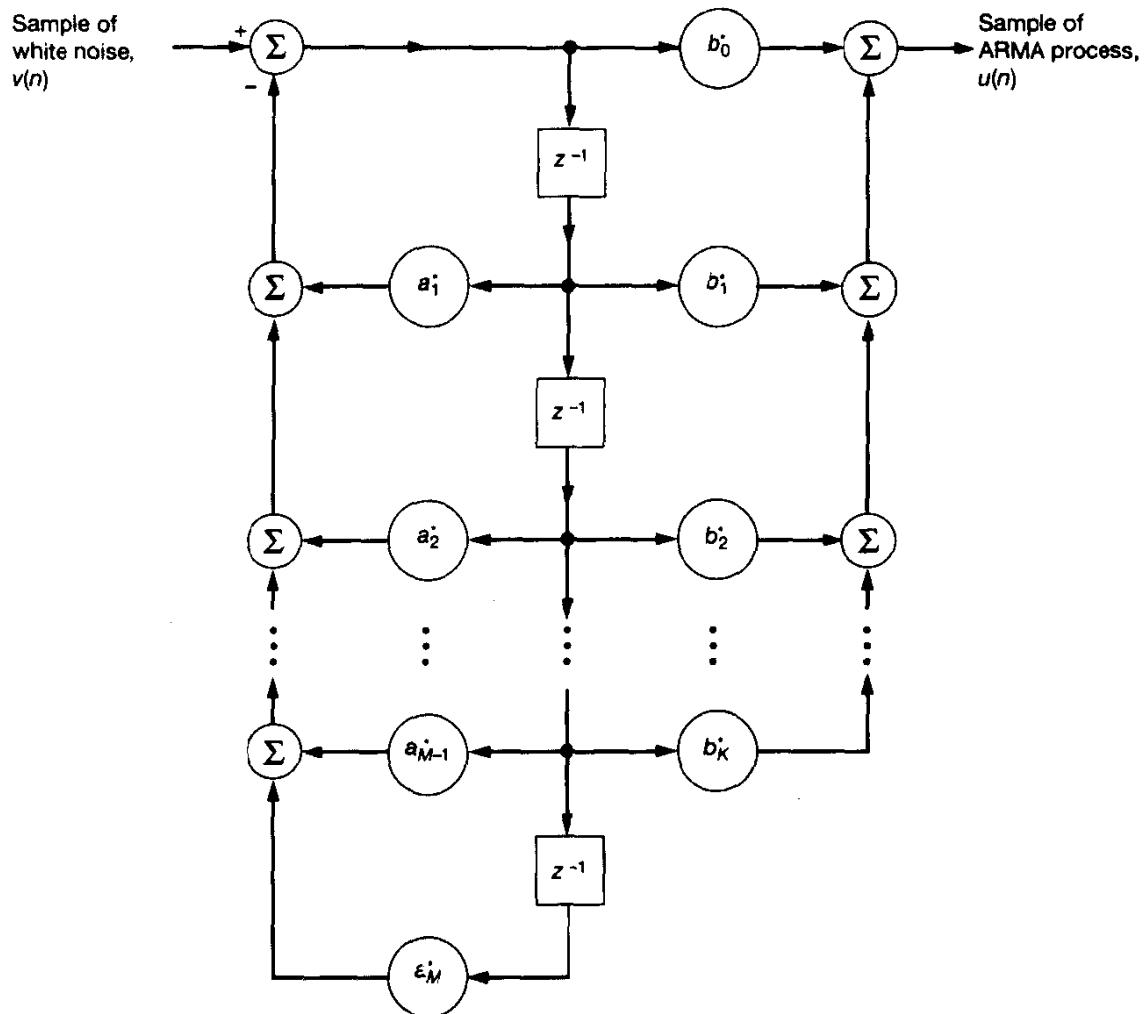


Figure 2.4 ARMA model (process generator) of order (M, K) , assuming that $M > K$.

The transfer function of the ARMA process generator in Fig. 2.4 has both poles and zeros. Similarly, the ARMA analyzer used to generate a white-noise process $v(n)$, given an ARMA process $u(n)$, is characterized by a transfer function containing both poles and zeros.

From a computational viewpoint, the AR model has an advantage over the MA and ARMA models. Specifically, the computation of the AR coefficients in the model of Fig. 2.2(a) involves a system of *linear equations* known as the Yule-Walker equations, details of which are given in Section 2.8. On the other hand, the computation of the MA coefficients in the model of Fig. 2.3 and the computation of the ARMA coefficients in the model of Fig. 2.4 are much more complicated. Both of these computations require solving systems of *nonlinear equations*. It is for this reason that, in practice, we find that the use of AR models is more popular than MA and ARMA models. The wide application of AR

models may also be justified by virtue of a fundamental theorem of time series analysis, which is discussed next.

2.6 WOLD DECOMPOSITION

Wold (1938) proved a fundamental theorem, which states that any stationary discrete-time stochastic process may be decomposed into the sum of a *general linear process* and a *predictable process*, with these two processes being uncorrelated with each other. More precisely, Wold proved the following result:

Any stationary discrete-time stochastic process $x(n)$ may be expressed in the form

$$x(n) = u(n) + s(n) \quad (2.54)$$

where

1. $u(n)$ and $s(n)$ are uncorrelated processes,
2. $u(n)$ is a general linear process represented by the MA model:

$$u(n) = \sum_{k=0}^{\infty} b_k^* v(n-k) \quad (2.55)$$

with $b_0 = 1$, and

$$\sum_{k=0}^{\infty} |b_k|^2 < \infty,$$

and where $v(n)$ is a white-noise process uncorrelated with $s(n)$; that is,

$$E[v(n)s^*(k)] = 0 \quad \text{for all } (n, k)$$

3. $s(n)$ is a predictable process; that is, the process can be predicted from its own past with zero prediction variance.

This result is known as *Wold's decomposition theorem*. A proof of this theorem is given in Priestley (1981).

According to Eq. (2.55), the general linear process $u(n)$ may be generated by feeding an *all-zero filter* with the white-noise process $v(n)$ as in Fig. 2.5(a). The zeros of the transfer function of this filter equal the roots of the equation:

$$B(z) = \sum_{n=0}^{\infty} b_n^* z^{-n} = 0$$

A solution of particular interest is an all-zero filter that is *minimum phase*, which means that all the zeros of the polynomial $B(z)$ lie inside the unit circle. In such a case, we may replace the all-zero filter with an *equivalent* all-pole filter that has the same impulse response $h_n = b_n^*$, as in Fig. 2.5(b). This means that except for a predictable component, a stationary discrete-time stochastic process may also be represented as an AR process of the appropriate order, subject to the above-mentioned restriction on $B(z)$. The basic difference between the MA and AR models is that $B(z)$ operates on the input $v(n)$ in the MA model, whereas the inverse $B^{-1}(z)$ operates on the output $u(n)$ in the AR model.

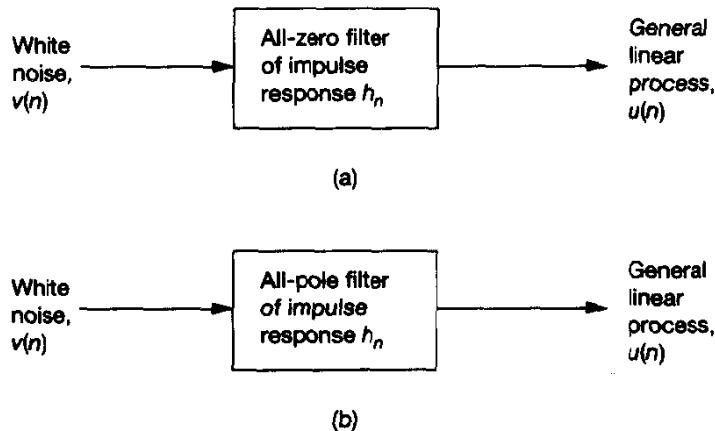


Figure 2.5 (a) Model, based on all-zero filter, for generating the linear process $u(n)$; (b) model, based on all-pole filter, for generating the general linear process $u(n)$. Both filters have exactly the same impulse response.

2.7 ASYMPTOTIC STATIONARITY OF AN AUTOREGRESSIVE PROCESS

Equation (2.42) represents a *linear, constant coefficient, difference equation of order M*, in which $v(n)$ plays the role of *input or driving function* and $u(n)$ that of *output or solution*. By using the *classical method*¹ for solving such an equation, we may formally express the solution $u(n)$ as the sum of a *complementary function*, $u_c(n)$, and a *particular solution*, $u_p(n)$, as follows:

$$u(n) = u_c(n) + u_p(n) \quad (2.56)$$

The evaluation of the solution $u(n)$ may thus proceed in two stages:

1. The complementary function $u_c(n)$ is the solution of the *homogeneous equation*

$$u(n) + a_1^* u(n-1) + a_2^* u(n-2) + \cdots + a_M^* u(n-M) = 0$$

In general, the complementary function $u_c(n)$ will therefore be of the form

$$u_c(n) = B_1 p_1^n + B_2 p_2^n + \cdots + B_M p_M^n \quad (2.57)$$

where B_1, B_2, \dots, B_M are arbitrary constants, and p_1, p_2, \dots, p_M are roots of the characteristic equation (2.51).

2. The particular solution $u_p(n)$ is defined by

$$u_p(n) = H_G(D)[v(n)] \quad (2.58)$$

¹We may also use the *z-transform method* to solve the difference equation (2.42). However, for the discussion presented here, we find it more informative to use the classical method.

where D is the *unit-delay operator*, and the operator $H_G(D)$ is obtained by substituting D for z^{-1} in the discrete-transfer function of Eq. (2.49). The unit-delay operator D has the property

$$D^k[u(n)] = u(n - k), \quad k = 0, 1, 2, \dots \quad (2.59)$$

The constants B_1, B_2, \dots, B_M are determined by the choice of *initial conditions* that equal M in number. It is customary to set

$$\begin{aligned} u(0) &= 0 \\ u(-1) &= 0 \\ \vdots & \\ u(-M + 1) &= 0 \end{aligned} \quad (2.60)$$

This is equivalent to setting the output of the model in Fig. 2.2(b) as well as the succeeding $(M - 1)$ tap inputs equal to zero at time $n = 0$. Thus, by substituting these initial conditions into Eqs. (2.56) – (2.58), we obtain a set of M simultaneous equations that can be solved for the constants B_1, B_2, \dots, B_M .

The result of imposing the initial conditions of Eq. (2.60) on the solution $u(n)$ is to make the discrete-time stochastic process represented by this solution nonstationary. On reflection, it is clear that this must be so, since we have given a “special status” to the time point $n = 0$, and the property of *invariance under a shift of time origin* cannot hold, even for second-order moments. If, however, the solution $u(n)$ is able to “forget” its initial conditions, the resulting process is asymptotically stationary in the sense that it settles down to a stationary behavior as n approaches infinity (Priestley, 1981). This requirement may be achieved by choosing the parameters of the AR model in Fig. 2.2(b) such that the complementary function $u_c(n)$ decays to zero as n approaches infinity. From Eq. (2.57) we see that, for arbitrary constants in the equation, this requirement can be met if and only if

$$|p_k| < 1 \quad \text{for all } k$$

Hence, for *asymptotic stationarity of the discrete-time stochastic process represented by the solution $u(n)$* , we require that *all the poles of the filter in the AR model lie inside the unit circle in the z -plane*. This is intuitively satisfying.

Correlation Function of an Asymptotically Stationary AR Process

Assuming that the condition for asymptotic stationarity is satisfied, we may derive an important recursive relation for the autocorrelation function of the resulting AR process $u(n)$ as follows. We first multiply both sides of Eq. (2.42) by $u^*(n - l)$ and then apply the expectation operator, thereby obtaining

$$E\left[\sum_{k=0}^M a_k^* u(n-k) u^*(n-l)\right] = E[v(n) u^*(n-l)] \quad (2.61)$$

Next, we simplify the left-hand side of Eq. (2.61) by interchanging the expectation and summation, and recognizing that the expectation $E[u(n-k)u^*(n-l)]$ equals the autocorrelation function of the AR process for a lag of $l-k$. We simplify the right-hand side by observing that the expectation $E[v(n)u^*(n-l)]$ is zero for $l > 0$, since $u(n-l)$ only involves samples of the white-noise process at the filter input in Fig. 2.2(b) up to time $n-l$, which are uncorrelated with the white-noise sample $v(n)$. Accordingly, we simplify Eq. (2.61) as follows:

$$\sum_{k=0}^{M-1} a_k^* r(l-k) = 0, \quad l > 0 \quad (2.62)$$

where $a_0 = 1$. We thus see that the autocorrelation function of the AR process satisfies the difference equation

$$r(l) = w_1^* r(l-1) + w_2^* r(l-2) + \dots + w_M^* r(l-M), \quad l > 0 \quad (2.63)$$

where $w_k = -a_k$, $k = 1, 2, \dots, M$. Note that Eq. (2.63) is analogous to the difference equation satisfied by the AR process $u(n)$ itself.

We may express the general solution of Eq. (2.63) as follows:

$$r(m) = \sum_{k=1}^{M-1} C_k p_k^m \quad (2.64)$$

where C_1, C_2, \dots, C_M are constants, and p_1, p_2, \dots, p_M are roots of the characteristic equation (2.51). Note that when the AR model of Fig. 2.2(b) satisfies the condition for asymptotic stationarity, $|p_k| < 1$ for all k , in which case the autocorrelation function $r(m)$ approaches zero as the lag m approaches infinity.

The exact form of the contribution made by a pole p_k in Eq. (2.64) depends on whether the pole is real or complex. When p_k is real, the corresponding contribution decays geometrically to zero as the lag m increases. We refer to such a contribution as a *damped exponential*. On the other hand, complex poles occur in conjugate pairs, and the contribution of a complex-conjugate pair of poles is in the form of a *damped sine wave*. We thus find that, in general, the autocorrelation function of an asymptotically stationary AR process consists of a mixture of damped exponentials and damped sine waves.

2.8 YULE-WALKER EQUATIONS

In order to uniquely define the AR model of order M , depicted in Fig. 2.2(b), we need to specify two sets of model parameters:

1. The AR coefficients a_1, a_2, \dots, a_M
2. The variance σ_v^2 of the white noise $v(n)$ used as excitation.

We now address these two issues in turn.

First, writing Eq. (2.63) for $l = 1, 2, \dots, M$, we get a set of M simultaneous equations with the values $r(0), r(1), \dots, r(M)$ of the autocorrelation function of the AR process as the known quantities and the AR parameters a_1, a_2, \dots, a_M as the unknowns. This set of equations may be expressed in the expanded matrix form

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ \vdots \\ r^*(M) \end{bmatrix} \quad (2.65)$$

where we have $w_k = -a_k$. The set of equations (2.65) is called the *Yule-Walker equations* (Yule, 1927; Walker, 1931).

We may express the Yule-Walker equations in the compact matrix form

$$\mathbf{R}\mathbf{w} = \mathbf{r} \quad (2.66)$$

and its solution as (assuming that the correlation matrix \mathbf{R} is nonsingular)

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{r} \quad (2.67)$$

where \mathbf{R}^{-1} is the inverse of matrix \mathbf{R} , and the vector \mathbf{w} is defined by

$$\mathbf{w} = [w_1, w_2, \dots, w_M]^T$$

The correlation matrix \mathbf{R} is defined by Eq. (2.21), and vector \mathbf{r} is defined by Eq. (2.28). From these two equations, we see that we may uniquely determine both the matrix \mathbf{R} and the vector \mathbf{r} , given the autocorrelation sequence $r(0), r(1), \dots, r(M)$. Hence, using Eq. (2.67) we may compute the coefficient vector \mathbf{w} and, therefore, the AR coefficients $a_k = -w_k, k = 1, 2, \dots, M$. In other words, there is a unique relationship between the coefficients a_1, a_2, \dots, a_M of the AR model and the *normalized* correlation coefficients $\rho_1, \rho_2, \dots, \rho_M$ of the AR process $u(n)$, as shown by

$$\{a_1, a_2, \dots, a_M\} \rightleftharpoons \{\rho_1, \rho_2, \dots, \rho_M\} \quad (2.68)$$

where the *correlation coefficient* ρ_k is defined by

$$\rho_k = \frac{r(k)}{r(0)}, \quad k = 1, 2, \dots, M \quad (2.69)$$

Variance of the White Noise

For $l = 0$, we find that the expectation on the right-hand side of Eq. (2.61) assumes the special form

$$\begin{aligned} E[v(n)u^*(n)] &= E[v(n)v^*(n)] \\ &= \sigma_v^2 \end{aligned} \quad (2.70)$$

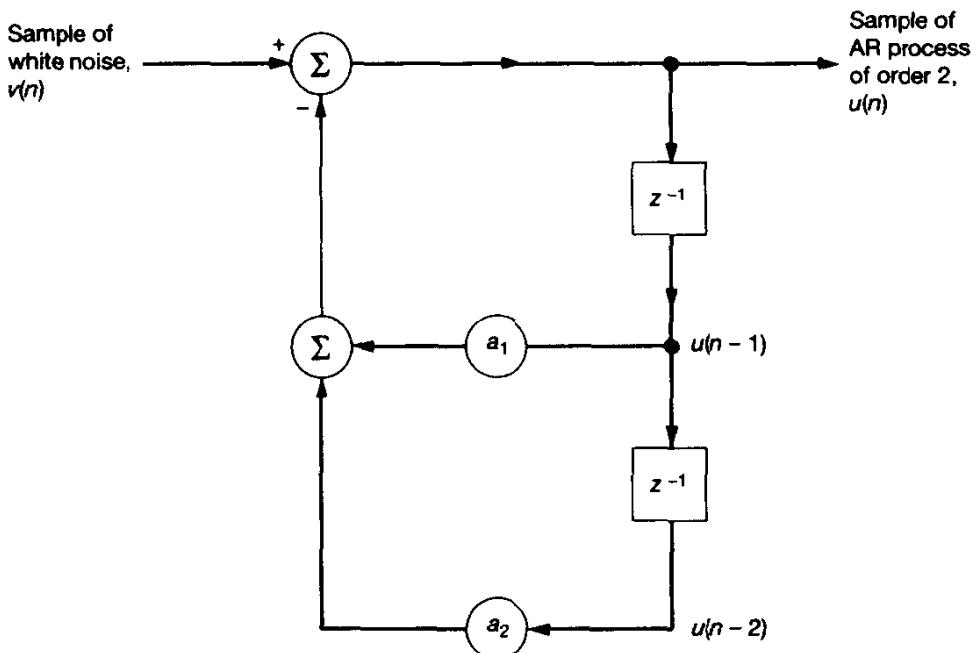


Figure 2.6 Model of (real-valued) autoregressive process of order 2.

where σ_v^2 is the variance of the zero-mean white noise $v(n)$. Accordingly, setting $l = 0$ in Eq. (2.61) and complex-conjugating both sides, we get the following formula for the variance of the white-noise process:

$$\sigma_v^2 = \sum_{k=0}^M a_k r(k) \quad (2.71)$$

where $a_0 = 1$. Hence, given the autocorrelations $r(0), r(1), \dots, r(M)$, we may determine the white-noise variance σ_v^2 .

2.9 COMPUTER EXPERIMENT: AUTOREGRESSIVE PROCESS OF ORDER 2

To illustrate the theory developed above for the modeling of an AR process, we consider the example of a second-order AR process that is real valued.² Figure 2.6 shows the block diagram of the model used to generate this process. Its time-domain description is governed by the second-order difference equation

$$u(n) + a_1 u(n-1) + a_2 u(n-2) = v(n) \quad (2.72)$$

²In this example, we follow the approach described by Box and Jenkins (1976).

where $v(n)$ is drawn from a white-noise process of zero mean and variance σ_v^2 . Figure 2.7(a) shows one realization of this white-noise process. The variance σ_v^2 is chosen to make the variance of $u(n)$ equal unity.

Conditions for Asymptotic Stationarity

The second-order AR process $u(n)$ has the characteristic equation

$$1 + a_1 z^{-1} + a_2 z^{-2} = 0 \quad (2.73)$$

Let p_1 and p_2 denote the two roots of this equation:

$$p_1, p_2 = \frac{1}{2}(-a_1 \pm \sqrt{a_1^2 - 4a_2}) \quad (2.74)$$

To ensure the asymptotic stationarity of the AR process $u(n)$, we require that these two roots lie inside the unit circle in the z -plane. That is, both p_1 and p_2 must have a magnitude less than 1. This, in turn, requires that the AR parameters a_1 and a_2 lie in the triangular region defined by

$$\begin{aligned} -1 &\leq a_2 + a_1 \\ -1 &\leq a_2 - a_1 \\ -1 &\leq a_2 \leq 1 \end{aligned} \quad (2.75)$$

as shown in Fig. 2.8.

Autocorrelation Function

The autocorrelation function $r(m)$ of an asymptotically stationary AR process for lag m satisfies the difference equation (2.62). Hence, using this equation, we obtain the following second-order difference equation for the autocorrelation function of a second-order AR process:

$$r(m) + a_1 r(m - 1) + a_2 r(m - 2) = 0, \quad m > 0 \quad (2.76)$$

For the initial values, we have (as will be explained later)

$$\begin{aligned} r(0) &= \sigma_u^2 \\ r(1) &= \frac{-a_1}{1 + a_2} \sigma_u^2 \end{aligned} \quad (2.77)$$

Thus, solving Eq. (2.76) for $r(m)$, we get (for $m > 0$)

$$r(m) = \sigma_u^2 \left[\frac{p_1(p_2^2 - 1)}{(p_2 - p_1)(p_1 p_2 + 1)} p_1^m - \frac{p_2(p_1^2 - 1)}{(p_2 - p_1)(p_1 p_2 + 1)} p_2^m \right] \quad (2.78)$$

where p_1 and p_2 are defined by Eq. (2.74).

There are two specific cases to be considered, depending on whether the roots p_1 and p_2 are real or complex valued, as described next.

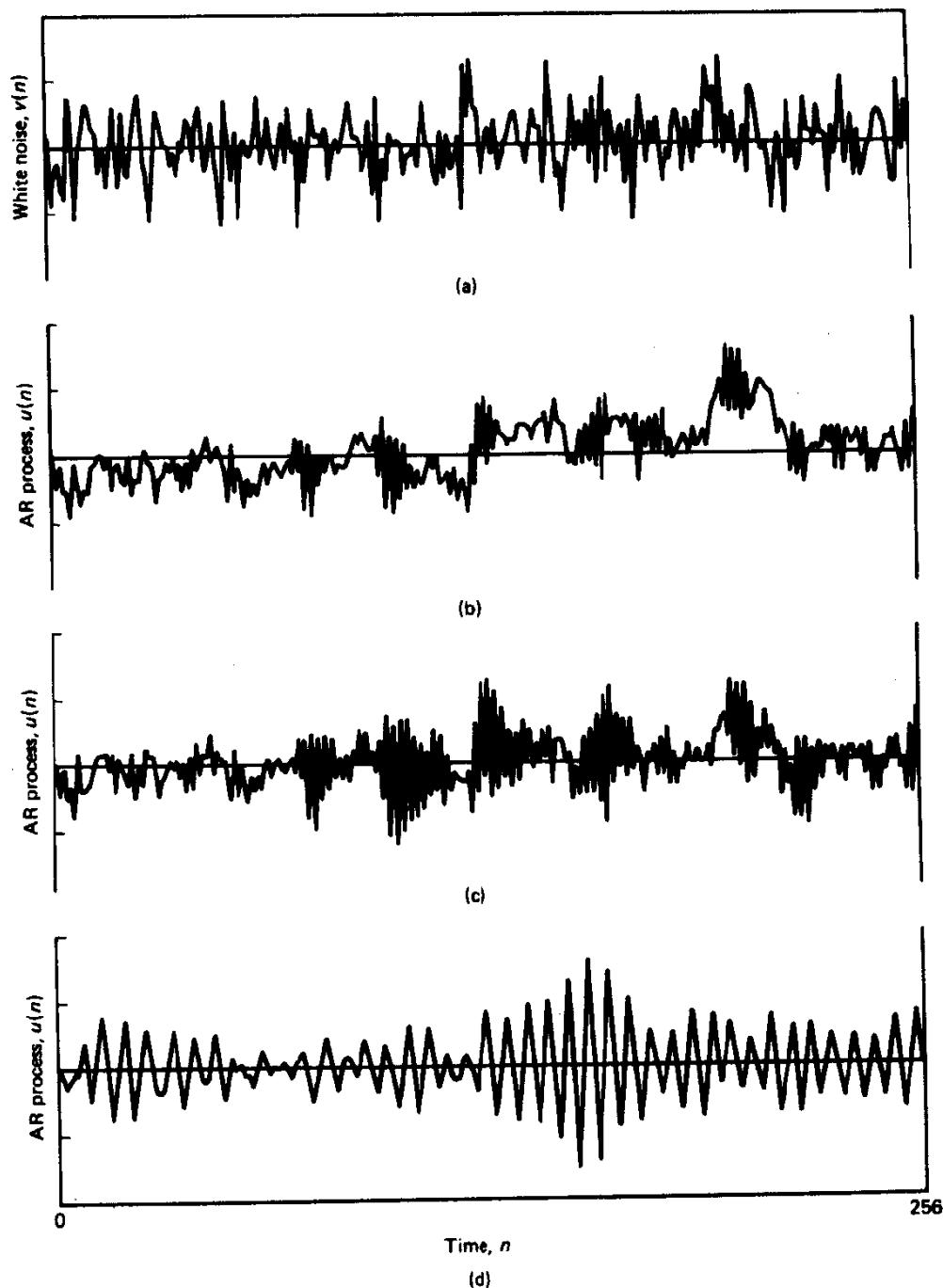


Figure 2.7 (a) One realization of white-noise input; (b), (c), (d) corresponding outputs of AR model of order 2 for parameters of Eqs. (2.79), (2.80), and (2.81), respectively.

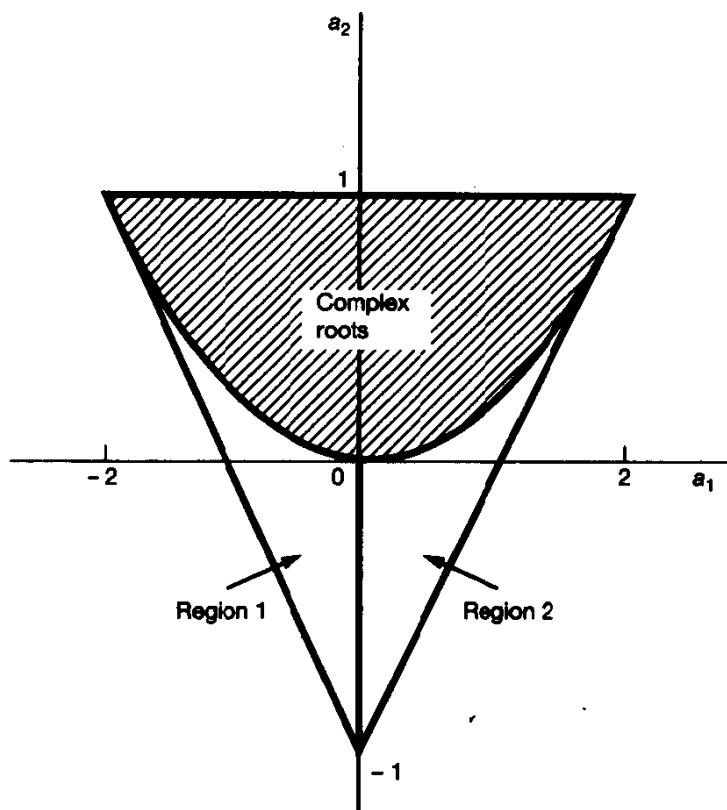


Figure 2.8 Permissible region for the AR parameters a_1 and a_2 .

Case 1: Real Roots. This case occurs when

$$a_1^2 - 4a_2 > 0$$

which corresponds to regions 1 and 2 below the parabolic boundary in Fig. 2.8. In region 1, the autocorrelation function remains positive as it damps out, corresponding to a positive dominant root. This situation is illustrated in Fig. 2.9(a) for the AR parameters

$$a_1 = -0.10 \quad (2.79)$$

$$a_2 = -0.8$$

In Fig. 2.7(b), we show the time variation of the output of the model in Fig. 2.6 [with a_1 and a_2 assigned the values given in Eq. (2.79)]. This output is produced by the white-noise input shown in Fig. 2.7(a).

In region 2 of Fig. 2.8, the autocorrelation function alternates in sign as it damps out, corresponding to a negative dominant root. This situation is illustrated in Fig. 2.9(b) for the AR parameters

$$a_1 = 0.1 \quad (2.80)$$

$$a_2 = -0.8$$

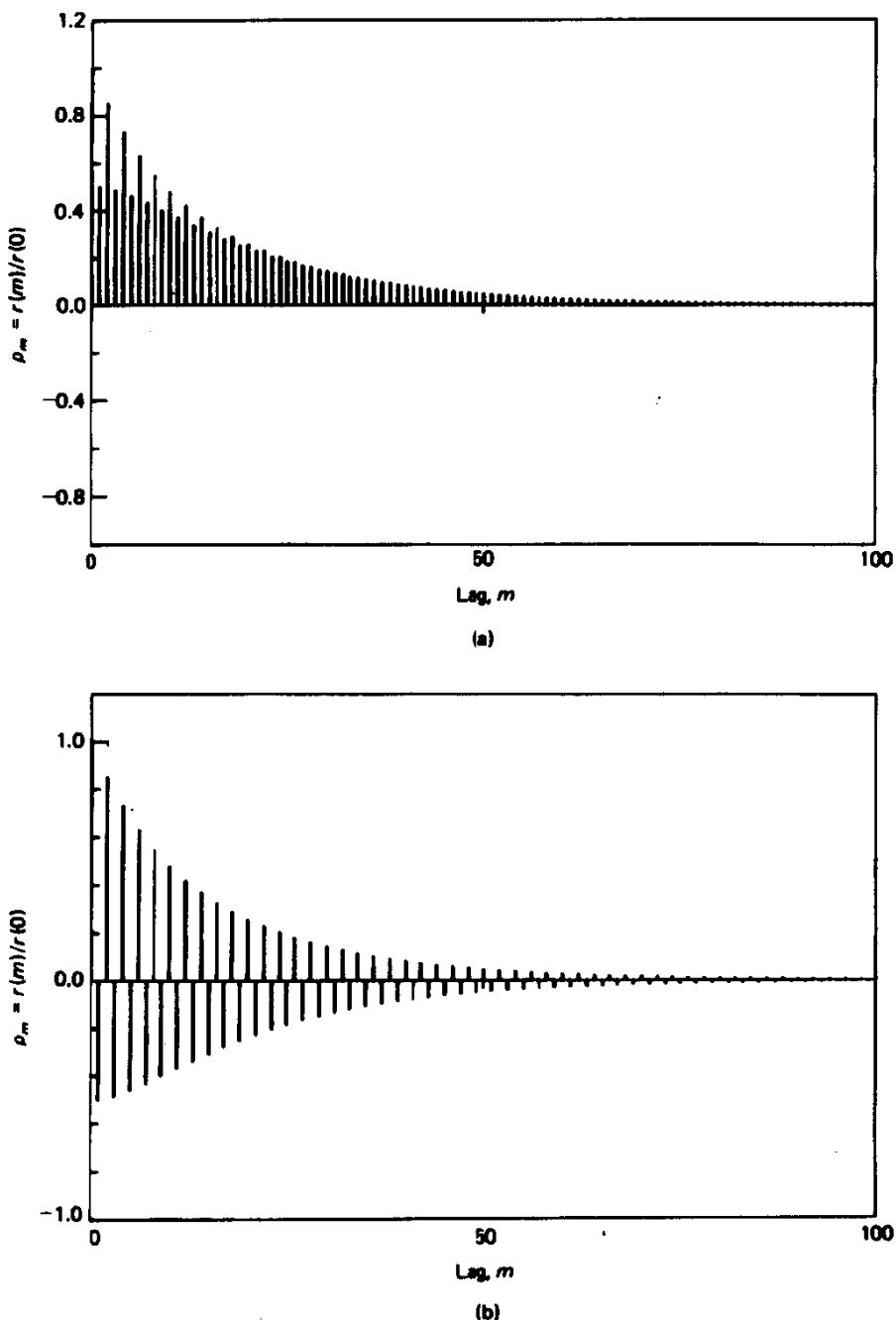


Figure 2.9 Plots of normalized autocorrelation function of real-valued AR(2) process;
(a) $r(1) > 0$; (b) $r(1) < 0$; (c) conjugate roots.

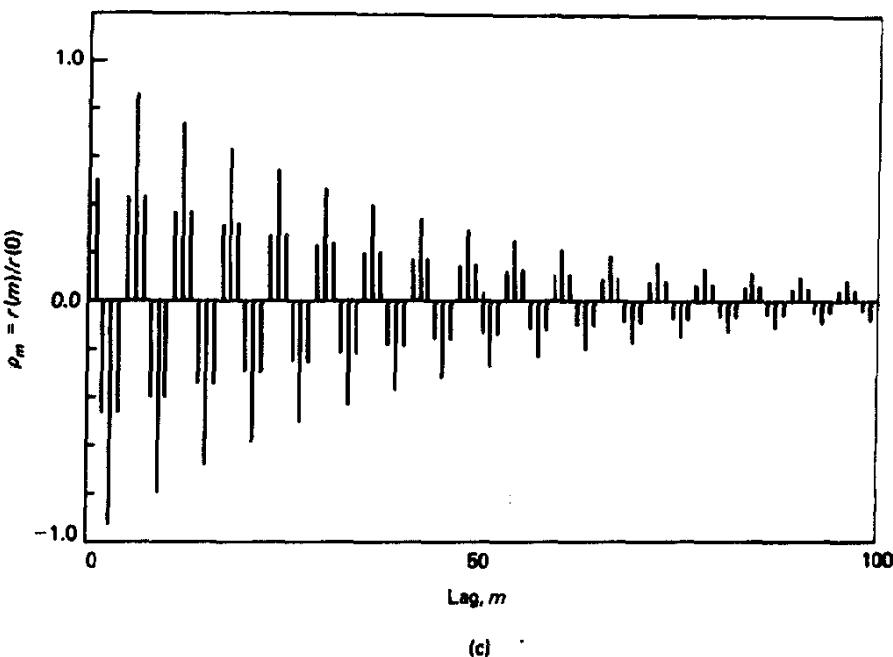


Figure 2.9 (continued)

In Fig. 2.7(c) we show the time variation of the output of the model in Fig. 2.6 [with a_1 and a_2 assigned the values given in Eq. (2.80)]. This output is also produced by the white-noise input shown in Fig. 2.7(a).

Case 2: Complex-Conjugate Roots. This occurs when

$$a_1^2 - 4a_2 < 0$$

which corresponds to the shaded region shown in Fig. 2.8 above the parabolic boundary. In this case, the autocorrelation function displays a pseudoperiodic behavior, as illustrated in Fig. 2.9(c) for the AR parameters

$$\begin{aligned} a_1 &= -0.975 \\ a_2 &= 0.95 \end{aligned} \tag{2.81}$$

In Fig. 2.7(d) we show the time variation of the output of the model in Fig. 2.6 [with a_1 and a_2 assigned the values given in Eq. (2.81)], which is produced by the white-noise input shown in Fig. 2.7(a).

Yule-Walker Equations

Substituting the value $M = 2$ for the AR model order in Eq. (2.65), we get the following Yule-Walker equations for the second-order AR process:

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \end{bmatrix} \quad (2.82)$$

where we have used the fact that $r(-1) = r(1)$ for a real-valued process. Solving Eq. (2.82) for w_1 and w_2 , we get

$$\begin{aligned} w_1 &= -a_1 = \frac{r(1)[r(0) - r(2)]}{r^2(0) - r^2(1)} \\ w_2 &= -a_2 = \frac{r(0)r(2) - r^2(1)}{r^2(0) - r^2(1)} \end{aligned} \quad (2.83)$$

We may also use Eq. (2.82) to express $r(1)$ and $r(2)$ in terms of the AR parameters a_1 and a_2 as follows:

$$\begin{aligned} r(1) &= \frac{-a_1}{1+a_2} \sigma_u^2 \\ r(2) &= \left(-a_2 + \frac{a_1^2}{1+a_2} \right) \sigma_u^2 \end{aligned} \quad (2.84)$$

where $\sigma_u^2 = r(0)$. This solution explains the initial values for $r(0)$ and $r(1)$ that were quoted in Eq. (2.77).

The conditions for asymptotic stationarity of the second-order AR process are given in terms of the AR parameters a_1 and a_2 in Eq. (2.75). Using the expressions for $r(1)$ and $r(2)$ in terms of a_1 and a_2 , given in Eq. (2.84), we may reformulate the conditions for asymptotic stationarity as follows:

$$\begin{aligned} -1 &< \rho_1 < 1 \\ -1 &< \rho_2 < 1 \\ \rho_1^2 &< \frac{1}{2}(1+\rho_2) \end{aligned} \quad (2.85)$$

where ρ_1 and ρ_2 are the normalized *correlation coefficients* defined by

$$\left. \begin{aligned} \rho_1 &= \frac{r(1)}{r(0)} \\ \rho_2 &= \frac{r(2)}{r(0)} \end{aligned} \right\} \quad (2.86)$$

and

Figure 2.10 shows the admissible region for ρ_1 and ρ_2 .

Variance of the White-Noise Process

Putting $M = 2$ in Eq. (2.71), we may express the variance of the white-noise process $v(n)$ as

$$\sigma_v^2 = r(0) + a_1 r(1) + a_2 r(2) \quad (2.87)$$

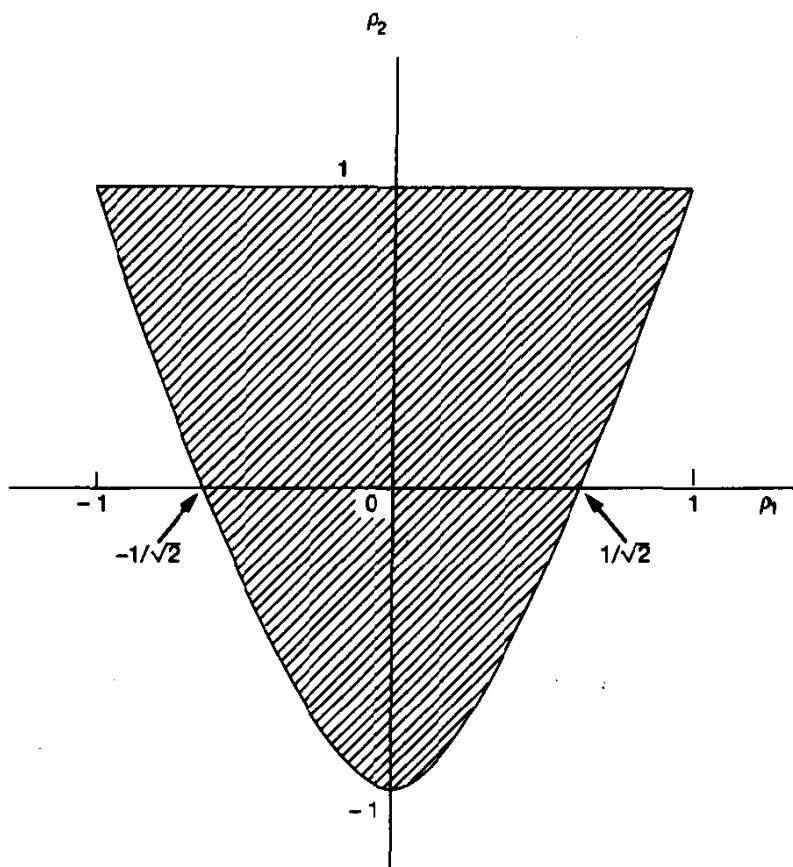


Figure 2.10 Permissible region for parameters of second-order AR process in terms of the normalized correlation coefficients ρ_1 and ρ_2 .

Next, substituting Eq. (2.84) in (2.87), and solving for $\sigma_u^2 = r(0)$, we get

$$\sigma_u^2 = \left(\frac{1 + a_2}{1 - a_2} \right) \frac{\sigma_v^2}{[(1 + a_2)^2 - a_1^2]} \quad (2.88)$$

For the three sets of AR parameters considered previously, we thus find that the variance of the white noise $v(n)$ has the values given in Table 2.1, assuming that $\sigma_u^2 = 1$.

TABLE 2.1 AR PARAMETERS AND NOISE VARIANCE

a_1	a_2	σ_v^2
-0.10	-0.8	0.27
0.1	-0.8	0.27
-0.975	0.95	0.0731

2.10 SELECTING THE MODEL ORDER

The representation of a stochastic process by a linear model may be used for synthesis or analysis. In *synthesis*, we generate a desired time series by assigning a prescribed set of values to the parameters of the model and feeding it with white noise of zero mean and prescribed variance. In *analysis*, on the other hand, we *estimate* the parameters of the model by processing a given time series of finite length. Insofar as the estimation is statistical, we need an appropriate measure of the fit between the model and the observed data. This implies that unless we have some prior information, the estimation procedure should include a criterion for selecting the *model order* (i.e., the number of independently adjusted parameters in the model). In the case of an AR process defined by Eq. (2.42), the model order equals M . In the case of an MA process defined by Eq. (2.52), the model order equals K . In the case of an ARMA process defined by Eq. (2.53), the model order equals (M, K) . Various criteria for model-order selection are described in the literature (Priestley, 1981; Kay, 1988). In this section we describe two important criteria for selecting the order of the model, one of which was pioneered by Akaike (1973, 1974) and the other by Rissanen (1978) and Schwartz (1978); both criteria result from the use of information-theoretic arguments, but in entirely different ways.

An Information-Theoretic Criterion

Let $u_i = u(i)$, $i = 1, 2, \dots, N$, denote the data obtained by N independent observations of a stationary discrete-time stochastic process, and $g(u_i)$ denote the probability density function of u_i . Let $f_U(u_i|\hat{\theta}_m)$ denote the conditional probability density function of u_i , given $\hat{\theta}_m$, where $\hat{\theta}_m$ is the *estimated* vector of parameters that model the process. Let m be the model order, so that we may write

$$\hat{\theta}_m = \begin{bmatrix} \hat{\theta}_{1m} \\ \hat{\theta}_{2m} \\ \vdots \\ \vdots \\ \hat{\theta}_{mm} \end{bmatrix} \quad (2.89)$$

We thus have several models that compete with each other to represent the process of interest. An *information-theoretic criterion* (AIC) proposed by Akaike selects the model for which the quantity

$$AIC(m) = -2L(\hat{\theta}_m) + 2m \quad (2.90)$$

is a minimum. The function $L(\hat{\theta}_m)$ is defined by

$$L(\hat{\theta}_m) = \max \sum_{i=1}^N \ln f_U(u_i|\hat{\theta}_m) \quad (2.91)$$

where \ln denotes the natural logarithm. The criterion of Eq. (2.91) is derived by minimizing the *Kullback–Leibler mean information*,³ which is used to provide a measure of the separation or distance between the “unknown” true probability density function $g(u)$ and the conditional probability density function $f_U(u_i|\hat{\theta}_m)$ given by the model in light of the observed data.

The function $L(\hat{\theta}_m)$, constituting the first term on the right-hand side of Eq. (2.90), except for a scalar, is recognized as a *logarithm* of the *maximum-likelihood estimates* of the parameters in the model; for a discussion of the method of maximum likelihood, see Appendix D. The second term, $2m$, represents a *model complexity penalty* that makes $AIC(m)$ an estimate of the Kullback–Leibler mean information.

The first term of Eq. (2.90) tends to decrease rapidly with model order m . On the other hand, the second term increases linearly with m . The result is that if we plot $AIC(m)$ versus model order m , the graph will, in general, show a definite minimum value, and the *optimum order* of the model is determined by that value of m at which $AIC(m)$ attains its minimum value. The minimum value of AIC is called **MAIC**(minimum AIC).

Minimum Description Length Criterion

Rissanen (1978, 1989) has used an entirely different approach to solve the statistical model identification problem. Specifically, he starts with the notion that a model may be viewed as a device for describing the regular features of a set of observed data, with the objective being that of searching for a model that best captures the regular features or constraints that give the data their special structure. Recognizing that the presence of constraints reduces uncertainty about the data, the objective may equally be that of *encoding* the data in the shortest or least redundant manner; the term “encoding” used here refers to an exact description of the observed data. Accordingly, the number of binary digits needed to encode both the observed data, when advantage is taken of the constraints offered by a model, and the model itself may be used as a criterion for *measuring the amount of the same constraints* and therefore the goodness of the model.

We may thus formally state Rissanen’s *minimum description length (MDL) criterion*⁴ as follows: Given a data set of interest and a family of competing statistical

³In Akaike (1973, 1974, 1977) and Ulrych and Ooe (1983), the criterion of Eq. (2.90) is derived from the principle of minimizing the expectation $E[I(g; f^*|\hat{\theta}_m)]$, where

$$I(g; f^*|\hat{\theta}_m) = \int_{-\infty}^{\infty} g(u) \ln g(u) du - \int_{-\infty}^{\infty} g(u) \ln f_U(u|\hat{\theta}_m) du$$

We refer to $I(g; f^*|\hat{\theta}_m)$ as the *Kullback–Leibler mean information* for discrimination between $g(u)$ and $f_U(u|\hat{\theta}_m)$ (Kullback and Leibler, 1951). The idea is to minimize the information added to the time series by modeling it as an AR, MA, or ARMA process of finite order, since any information added is virtually false information in a real-world situation. Since $g(u)$ is fixed and unknown, the problem reduces to one of maximizing the second term that makes up $I(g; f^*|\hat{\theta}_m)$.

⁴The idea of *minimum description length* of individual recursively definable objects may be traced to Kolmogorov (1968).

models, the best model is the particular one that provides the shortest description length for the data. In mathematical terms, it is defined by⁵ (Rissanen, 1978, 1989; Wax, 1995)

$$\text{MDL}(m) = -L(\hat{\theta}_m) + \frac{1}{2} m \ln N \quad (2.92)$$

where m is the number of independently adjusted parameters in the model, and N is the sample size (i.e., the number of observations). As with Akaike's information-theoretic criterion, $L(\hat{\theta}_m)$ is the logarithm of the maximum likelihood estimates of the model parameters. In comparing Eqs. (2.90) and (2.92), we see that the principal difference between the AIC and MDL criterion lies in the structure-dependent term.

According to Rissanen (1989), the MDL criterion offers the following attributes:

- The model permits the shortest encoding of the observed data and captures all the *learnable* properties of the observed data in the best possible manner.
- The MDL criterion is a *consistent* model-order estimator in the sense that it converges to the true model order as the sample size increases.
- The model is optimal in the context of linear regression problems as well as ARMA models.

Perhaps the most significant point to note is the fact that in all of the applications involving the MDL criterion, there has been no anomalous result or a model with undesirable properties reported in the literature.

2.11 COMPLEX GAUSSIAN PROCESSES

Gaussian stochastic processes, or simply *Gaussian processes*, are frequently encountered in both theoretical and applied analysis. In this section we present a summary of some important properties of Gaussian processes that are *complex valued*.⁶

Let $u(n)$ denote a complex Gaussian process consisting of N samples. For the first- and second-order statistics of this process, we assume the following:

1. A *mean* of zero as shown by

$$\mu = E[u(n)] = 0 \quad \text{for } n = 1, 2, \dots, N \quad (2.93)$$

⁵Schwartz (1989) has derived a similar result, using a Bayesian approach. In particular, he considers the asymptotic behavior of Bayes estimators under a special class of *priors*. These priors put positive probability on the subspaces that correspond to the competing models. The decision is made by selecting the model that yields the maximum *a posteriori probability*.

It turns out that, in the large sample limit, the two approaches taken by Schwartz and Rissanen yield essentially the same result. However, Rissanen's approach is much more general, whereas Schwartz's approach is restricted to the case that the observations are independent and come from an exponential distribution.

⁶For a detailed treatment of complex Gaussian processes, see the book by Miller (1974). Properties of complex Gaussian processes are also discussed in Kelly et al. (1960), Reed (1962), and McGee (1971).

2. An autocorrelation function denoted by

$$r(k) = E[u(n)u^*(n - k)], \quad k = 0, 1, \dots, N - 1 \quad (2.94)$$

The set of autocorrelation functions $\{r(k), k = 0, 1, \dots, N - 1\}$ defines the correlation matrix \mathbf{R} of the Gaussian process $u(n)$.

The shorthand notation $N(\mathbf{0}, \mathbf{R})$ is commonly used to refer to a Gaussian process with a mean vector of zero and correlation matrix \mathbf{R} .

Equations (2.93) and (2.94) imply wide-sense stationarity of the process. Knowledge of the mean μ and the autocorrelation function $r(k)$ for varying values of lag k is indeed sufficient for the complete characterization of the complex Gaussian process $u(n)$. In particular, it may be shown that the *joint probability density function* of N samples of the process so described is as follows (Kelly et al., 1960):

$$f_U(\mathbf{u}) = \frac{1}{(2\pi)^N \det(\Lambda)} \exp\left(-\frac{1}{2}\mathbf{u}^H \Lambda^{-1} \mathbf{u}\right) \quad (2.95)$$

where \mathbf{u} is the N -by-1 data vector; that is,

$$\mathbf{u} = [u(1), u(2), \dots, u(N)]^T \quad (2.96)$$

and Λ is the N -by- N Hermitian-symmetric *moment matrix* of the process, defined in terms of the correlation matrix $\mathbf{R} = \{r(k)\}$ as

$$\begin{aligned} \Lambda &= \frac{1}{2} E[\mathbf{u}\mathbf{u}^H] \\ &= \frac{1}{2}\mathbf{R} \end{aligned} \quad (2.97)$$

Note that the joint probability density function $f_U(\mathbf{u})$ is $2N$ -dimensional, where the factor 2 accounts for the fact that each of the N samples of the process has a real and an imaginary part. Note also that the probability density function of a single sample $u(n)$ of the process, which is a special case of Eq. (2.95), is given by

$$f_U(u) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{|u|^2}{\sigma^2}\right) \quad (2.98)$$

where $|u|$ is the magnitude of the sample $u(n)$ and σ^2 is its variance.

Based on the representation described herein, we may now summarize some important properties of a *zero-mean complex Gaussian process $u(n)$ that is wide-sense stationary* as follows:

1. The process $u(n)$ is *stationary in the strict sense*.
2. The process $u(n)$ is *circularly complex* in the sense that any two different samples $u(n)$ and $u(k)$ of the process satisfy the condition

$$E[u(n)u(k)] = 0 \quad \text{for } n \neq k \quad (2.99)$$

It is for this reason that the process $u(n)$ is often referred to as a *circularly complex Gaussian process*.

3. Suppose that $u_n = u(n)$, for $n = 1, 2, \dots, N$, are samples picked from a zero-mean, complex Gaussian process $u(n)$. We may thus state Property 3 in two parts (Reed, 1962):

- (a) If $k \neq l$, then

$$E[u_{s_1}^* u_{s_2}^* \dots u_{s_k}^* u_{t_1} u_{t_2} \dots u_{t_l}] = 0 \quad (2.100)$$

where s_i and t_j are integers selected from the available set $\{1, 2, \dots, N\}$.

- (b) If $k = l$, then

$$E[u_{s_1}^* u_{s_2}^* \dots u_{s_l}^* u_{t_1} u_{t_2} \dots u_{t_l}] = E[u_{s_{\pi(1)}}^* u_{t_1}] E[u_{s_{\pi(2)}}^* u_{t_2}] \dots E[u_{s_{\pi(l)}}^* u_{t_l}] \quad (2.101)$$

where π is a permutation of the set of integers $\{1, 2, \dots, l\}$, and $\pi(j)$ is the j th element of that permutation. For the set of integers $\{1, 2, \dots, l\}$ we have a total of $l!$ possible permutations. This means that the right-hand side of Eq. (2.101) consists of the product of $l!$ expectation product terms. Equation (2.101) is called the *Gaussian moment factoring theorem*.

Example 2

Consider the case of $N = 4$, for which the complex Gaussian process $u(n)$ consists of the four samples u_1, u_2, u_3 , and u_4 . Hence, the use of the Gaussian moment factoring theorem given in Eq. (2.101) yields the following useful identity:

$$E[u_1^* u_2^* u_3 u_4] = E[u_1^* u_3] E[u_2^* u_4] + E[u_2^* u_3] E[u_1^* u_4] \quad (2.102)$$

For other useful identities derived from the Gaussian moment factoring theorem, see Problem 11.

2.12 SUMMARY AND DISCUSSION

In this chapter we studied the partial characterization of a stationary discrete-time stochastic process. Such a characterization is uniquely described in terms of two statistical parameters:

1. The mean, which is a constant
2. The autocorrelation function, which depends only on the time difference between any two samples of the process

The mean of the process may naturally be zero, or it can always be subtracted from the process to yield a new process of zero mean. For this reason, in much of the discussion in subsequent chapters of this book, the mean of the process is assumed to be zero. Thus, given an M -by-1 observation vector $\mathbf{u}(n)$ known to belong to a complex, stationary, discrete-time stochastic process of zero mean, we may partially describe it by defining an M -by- M correlation matrix \mathbf{R} as the statistical expectation of the outer product of $\mathbf{u}(n)$ with

itself. The matrix \mathbf{R} is Hermitian, Toeplitz, and almost always positive definite; the latter property means that \mathbf{R} is almost always nonsingular, and therefore the inverse matrix \mathbf{R}^{-1} exists.

Another topic discussed in the chapter is the notion of a stochastic model, the need for which arises when we are given a set of experimental data known to be of a statistical nature, and the requirement is to analyze the data. In this context, we may mention two general requirements for a suitable model:

1. An *adequate number of adjustable parameters* for the model to capture the essential information content of the input data
2. *Mathematical tractability* of the model

The first requirement, in effect, means that the complexity of the model should closely match the complexity of the underlying physical mechanism responsible for generating the input data; in so doing, problems associated with underfitting or overfitting the input data are avoided. The second requirement is usually satisfied by the choice of a linear model.

Within the family of linear stochastic models, the autoregressive (AR) model is often preferred over the moving average (MA) model and the autoregressive-moving average (ARMA) model for an important reason: unlike an MA or ARMA model, computation of the AR coefficients is governed by a system of linear equations, namely, the Yule–Walker equations. Moreover, except for a predictable component, we may approximate a stationary discrete-time stochastic process by an AR model of sufficiently high order, subject to certain restrictions. To select a suitable value for the model order, we may use an information-theoretic criterion (AIC) according to Akaike or the minimum-description length (MDL) criterion according to Rissanen. A useful feature of the MDL criterion is that it is a consistent model-order estimator.

PROBLEMS

1. The sequences $y(n)$ and $u(n)$ are related by the difference equation

$$y(n) = u(n + a) - u(n - a)$$

where a is a constant. Evaluate the autocorrelation function of $y(n)$ in terms of that of $u(n)$.

2. Consider a correlation matrix \mathbf{R} for which the inverse matrix \mathbf{R}^{-1} exists. Show that \mathbf{R}^{-1} is Hermitian.
3. (a) Equation (2.26) relates the $(M + 1)$ -by- $(M + 1)$ correlation matrix \mathbf{R}_{M+1} , pertaining to the observation vector $\mathbf{u}_{M+1}(n)$ taken from a stationary stochastic process, to the M -by- M correlation matrix \mathbf{R}_M of the observation vector $\mathbf{u}_M(n)$ taken from the same process. Evaluate the inverse of the correlation matrix \mathbf{R}_{M+1} in terms of the inverse of the correlation matrix \mathbf{R}_M .
 (b) Repeat your evaluation using Eq. (2.27).
4. A first-order autoregressive (AR) process $u(n)$, which is real-valued, satisfies the real-valued difference equation

$$u(n) + a_1 u(n - 1) = v(n)$$

where a_1 is a constant, and $v(n)$ is a white-noise process of variance σ_v^2 .

- (a) Show that if $v(n)$ has a nonzero mean, the AR process $u(n)$ is nonstationary.
- (b) For the case when $v(n)$ has zero mean, and the constant a_1 satisfies the condition $|a_1| < 1$, show that the variance of $u(n)$ equals

$$\text{var}[u(n)] = \frac{\sigma_v^2}{1 - a_1^2}$$

- (c) For the conditions specified in part (b), find the autocorrelation function of the AR process $u(n)$. Sketch this autocorrelation function for the two cases $0 < a_1 < 1$ and $-1 < a_1 < 0$.

5. Consider an autoregressive process $u(n)$ of order 2, described by the difference equation

$$u(n) = u(n - 1) - 0.5u(n - 2) + v(n)$$

where $v(n)$ is a white-noise process of zero mean and variance 0.5.

- (a) Write the Yule-Walker equations for the process.
- (b) Solve these two equations for the autocorrelation function values $r(1)$ and $r(2)$.
- (c) Find the variance of $u(n)$.

6. Consider a wide-sense stationary process that is modeled as an AR process $u(n)$ of order M . The set of parameters made up of the average power P_0 and the AR coefficients a_1, a_2, \dots, a_M bear a one-to-one correspondence with the autocorrelation sequence $r(0), r(1), r(2), \dots, r(M)$, as shown by

$$\{r(0), r(1), r(2), \dots, r(M)\} \rightleftharpoons \{P_0, a_1, a_2, \dots, a_M\}$$

Justify the validity of this statement.

7. Evaluate the transfer functions of the following two stochastic models:

- (a) The MA model of Fig. 2.3
- (b) The ARMA model of Fig. 2.4.
- (c) Specify the conditions for which the transfer function of the ARMA model of Fig. 2.4 reduces (1) to that of an AR model, and (2) to that of an MA model.

8. Consider an MA process $x(n)$ of order 2 described by the difference equation

$$x(n) = v(n) + 0.75v(n - 1) + 0.25v(n - 2)$$

where $v(n)$ is a zero-mean white-noise process of unit variance. The requirement is to approximate this process by an AR process $u(n)$ of order M . Do this approximation for the following orders:

- (a) $M = 2$
- (b) $M = 5$
- (c) $M = 10$

Comment on your results. How big would the order M of the AR process $u(n)$ have to be for it to be equivalent to the MA process $x(n)$ exactly?

9. A time series $u(n)$ obtained from a wide-sense stationary stochastic process of zero mean and correlation matrix \mathbf{R} is applied to an FIR filter of impulse response w_n . This impulse response defines the coefficient vector \mathbf{w} .

- (a) Show that the average power of the filter output is equal to $\mathbf{w}^H \mathbf{R} \mathbf{w}$.
- (b) How is the result in part (a) modified if the stochastic process at the filter input is a white noise of variance σ^2 ?

10. A general linear complex-valued process $u(n)$ is described by

$$u(n) = \sum_{k=0}^{\infty} b_k^* v(n-k)$$

where $v(n)$ is a white noise process, and b_k is a complex coefficient. Justify the following statements:

- (a) If the process $v(n)$ is Gaussian, then the original process $u(n)$ is also Gaussian.
- (b) Conversely, a Gaussian process $u(n)$ implies that the process $v(n)$ is Gaussian.

11. Consider a complex Gaussian process $u(n)$. Let $u(n) = u_n$. Using the Gaussian moment factoring theorem, demonstrate the following identities:

- (a) $E[(u_1^* u_2)^k] = k! (E[u_1^* u_2])^k$
- (b) $E[|u|^k] = k! (E[|u|^2])^{\frac{k}{2}}$

CHAPTER

3

Spectrum Analysis

The autocorrelation function is a *time-domain description* of the second-order statistics of a stochastic process. The *frequency-domain description* of the second-order statistics of such a process is the *power spectral density*, which is also commonly referred to as the *power spectrum* or simply *spectrum*. Indeed, the power spectral density of a stochastic process is firmly established as the most useful description of the time series commonly encountered in engineering and physical sciences.

This chapter is devoted in part to the definition of the power spectral density of a wide-sense stationary discrete-time stochastic process, the properties of power spectral density, and methods for its estimation. We begin the discussion by establishing a mathematical definition of the power spectral density of a stationary process in terms of the Fourier transform of a single realization of the process.

3.1 POWER SPECTRAL DENSITY

Consider an infinitely long time series $u(n)$, $n = 0, \pm 1, \pm 2, \dots$, that represents a *single realization* of a wide-sense stationary discrete-time stochastic process. For convenience of presentation, we assume that the process has zero mean. Initially, we focus our attention on a *windowed* portion of this time series, written as

$$u_N(n) = \begin{cases} u(n), & n = 0, 1, \dots, N - 1 \\ 0, & n > N, n < 0 \end{cases} \quad (3.1)$$

where N is the *total length (duration) of the window*. By definition, the *discrete-time Fourier transform* of the windowed time series $u_N(n)$ is given by

$$U_N(\omega) = \sum_{n=0}^{N-1} u_N(n) e^{-j\omega n} \quad (3.2)$$

where ω is the *angular frequency*, lying in the interval $(-\pi, \pi]$. In general, $U_N(\omega)$ is complex valued; specifically, its *complex conjugate* is given by

$$U_N^*(\omega) = \sum_{k=0}^{N-1} u_N^*(k) e^{j\omega k} \quad (3.3)$$

where the asterisk denotes complex conjugation. In Eq. (3.3) we have used the variable k to denote discrete time for reasons that will become apparent immediately. In particular, we may multiply Eq. (3.2) by (3.3) to express the squared magnitude of $U_N(n)$ as follows:

$$|U_N(\omega)|^2 = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} u_N(n) u_N^*(k) e^{-j\omega(n-k)} \quad (3.4)$$

Each realization $U_N(n)$ produces such a result. The *expected* result is obtained by taking the statistical expectation of both sides of Eq. (3.4), and interchanging the order of expectation and double summation:

$$E[|U_N(\omega)|^2] = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} E[u_N(n) u_N^*(k)] e^{-j\omega(n-k)} \quad (3.5)$$

We now recognize that for the wide-sense stationary discrete-time stochastic process under discussion, the autocorrelation function of $u_N(n)$ for lag $n - k$ is

$$r_N(n - k) = E[u_N(n) u_N^*(k)] \quad (3.6)$$

which may be rewritten as follows, in light of the defining equation (3.1):

$$r_N(n - k) = \begin{cases} E[u(n)u^*(k)] = r(n - k) & \text{for } 0 \leq (n, k) \leq N - 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

Accordingly, Eq. (3.6) takes on the form

$$E[|U_N(\omega)|^2] = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} r(n - k) e^{-j\omega(n-k)} \quad (3.8)$$

Let $l = n - k$, and so rewrite Eq. (3.8) as follows:

$$\frac{1}{N} E[|U_N(\omega)|^2] = \sum_{l=-N+1}^{N-1} \left(1 - \frac{|l|}{N}\right) r(l) e^{-j\omega l} \quad (3.9)$$

Equation (3.9) may be interpreted as the discrete-time Fourier transform of the product of two time functions: the autocorrelation function $r_N(l)$ for lag l , and a triangular window

$w_B(l)$ known as the *Barlett window*. The latter function is defined by

$$w_B(l) = \begin{cases} 1 - \frac{|l|}{N}, & |l| \leq N - 1 \\ 0, & |l| \geq N \end{cases} \quad (3.10)$$

As N approaches infinity, the Barlett window $w_B(l)$ approaches unity for all l . Correspondingly, we may write

$$\lim_{N \rightarrow \infty} \frac{1}{N} E[|U_N(\omega)|^2] = \sum_{l=-\infty}^{\infty} r(l) e^{-j\omega l} \quad (3.11)$$

where $r(l)$ is the autocorrelation function of the original time series $u(n)$, assumed to have infinite length. The quantity $U_N(\omega)$ is the discrete-time Fourier transform of a rectangular *windowed* portion of this time series that has length N .

Equation (3.11) leads us to define the quantity

$$S(\omega) = \lim_{N \rightarrow \infty} \frac{1}{N} E[|U_N(\omega)|^2] \quad (3.12)$$

where the quantity $|U_N(\omega)|^2/N$ is called the *periodogram* of the windowed time series $u_N(n)$. Note that the order of expectation and limiting operations indicated in Eq. (3.12) cannot be changed. Note also that the periodogram converges to $S(\omega)$ only in the mean value, but *not* in mean square or any other meaningful way.

When the limit in Eq. (3.12) exists, the quantity $S(\omega)$ has the following interpretation (Priestley, 1981):

$$S(\omega) d\omega = \text{average of the contribution to the total power from components of a wide-sense stationary stochastic process with angular frequencies located between } \omega \text{ and } \omega + d\omega; \text{ the average is taken over all possible realizations of the process} \quad (3.13)$$

Accordingly, the quantity $S(\omega)$ is the "spectral density of expected power," which is abbreviated as the *power spectral density* of the process. Thus, equipped with the definition of power spectral density given in Eq. (3.12), we may now rewrite Eq. (3.11) as

$$S(\omega) = \sum_{l=-\infty}^{\infty} r(l) e^{-j\omega l} \quad (3.14)$$

In summary, Eq. (3.12) gives a basic definition of the power spectral density of a wide-sense stationary stochastic process, and Eq. (3.14) defines the mathematical relationship between the autocorrelation function and the power spectral density of such a process.

3.2 PROPERTIES OF POWER SPECTRAL DENSITY

Property 1. *The autocorrelation function and power spectral density of a wide-sense stationary stochastic process form a Fourier transform pair.*

Consider a wide-sense stationary stochastic process represented by the time series $u(n)$, assumed to be of infinite length. Let $r(l)$ denote the autocorrelation function of such a process for lag l , and let $S(\omega)$ denote its power spectral density. According to Property 1, these two quantities are related by the pair of relations:

$$S(\omega) = \sum_{l=-\infty}^{\infty} r(l)e^{-j\omega l}, \quad -\pi < \omega \leq \pi \quad (3.15)$$

and

$$r(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega)e^{j\omega l} d\omega, \quad l = 0, \pm 1, \pm 2, \dots \quad (3.16)$$

Equation (3.15) states that *the power spectral density is the discrete-time Fourier transform of the autocorrelation function*. On the other hand, Eq. (3.16) states that *the autocorrelation function is the inverse discrete-time Fourier transform of the power spectral density*. This fundamental pair of equations constitutes the *Einstein–Wiener–Khintchine relations*.

In a way, we already have a proof of this property. Specifically, Eq. (3.15) is merely a restatement of Eq. (3.14), previously established in Section 3.1. Equation (3.16) follows directly from this result by invoking the formula for the inverse discrete-time Fourier transform.

Property 2. *The frequency support of the power spectral density $S(\omega)$ is the Nyquist interval $-\pi < \omega \leq \pi$.*

Outside this interval, $S(\omega)$ is periodic as shown by

$$S(\omega + 2k\pi) = S(\omega) \quad \text{for integer } k \quad (3.17)$$

Property 3. *The power spectral density of a stationary discrete-time stochastic process is real.*

To prove this property, we rewrite Eq. (3.15) as

$$S(\omega) = r(0) + \sum_{k=1}^{\infty} r(k)e^{-j\omega k} + \sum_{k=-\infty}^{-1} r(k)e^{-j\omega k}$$

Replacing k with $-k$ in the third term on the right-hand side of this equation, and recognizing that $r(-k) = r^*(k)$, we get

$$\begin{aligned} S(\omega) &= r(0) + \sum_{k=1}^{\infty} [r(k)e^{-j\omega k} + r^*(k)e^{j\omega k}] \\ &= r(0) + 2 \sum_{k=1}^{\infty} \operatorname{Re}[r(k)e^{-j\omega k}] \end{aligned} \quad (3.18)$$

where Re denotes the *real part operator*. Equation (3.18) shows that the power spectral density $S(\omega)$ is a real-valued function of ω . It is because of this property that we have used the notation $S(\omega)$ rather than $S(e^{j\omega})$ for the power spectral density.

Property 4. *The power spectral density of a real-valued stationary discrete-time stochastic process is even (i.e., symmetric); if the process is complex-valued, its power spectral density is not necessarily even.*

For a real-valued stochastic process, we find that $S(-\omega) = S(\omega)$, indicating that $S(\omega)$ is an even function of ω ; that is, it is symmetric about the origin. If, however, the process is complex-valued, then $r(-k) = r^*(k)$, in which case we find that $S(-\omega) \neq S(\omega)$, and $S(\omega)$ is not an even function of ω .

Property 5. *The mean-square value of a stationary discrete-time stochastic process equals, except for the scaling factor $1/2\pi$, the area under the power spectral density curve for $-\pi < \omega \leq \pi$.*

This property follows directly from Eq. (3.16), evaluated for $l = 0$. For this condition, we may thus write

$$r(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega \quad (3.19)$$

Since $r(0)$ equals the mean-square value of the process, we see that Eq. (3.19) is a mathematical description of Property 5. The mean-square value of a process is equal to the *expected power* of the process developed across a load resistor of 1 ohm. On this basis, the terms "expected power" and "mean-square value" are used interchangeably in what follows.

Property 6. *The power spectral density of a stationary discrete-time stochastic process is nonnegative.*

That is,

$$S(\omega) \geq 0 \quad \text{for all } \omega \quad (3.20)$$

This property follows directly from the basic formula of Eq. (3.12), reproduced here for convenience of presentation:

$$S(\omega) = \lim_{N \rightarrow \infty} \frac{1}{N} E[|U_N(\omega)|^2]$$

We first note that $|U_N(\omega)|^2$, representing the squared magnitude of the discrete-time Fourier transform of a windowed portion of the time series $u(n)$, is nonnegative for all ω . The expectation $E[|U_N(\omega)|^2]$ is also nonnegative for all ω . Thus, using the basic definition of $S(\omega)$ in terms of $U_N(\omega)$, the property described by Eq. (3.20) follows immediately.

3.3 TRANSMISSION OF A STATIONARY PROCESS THROUGH A LINEAR FILTER

Consider a discrete-time filter that is *linear*, *time invariant*, and *stable*. Let the filter be characterized by the *discrete transfer function* $H(z)$, defined as the *ratio of the z-transform of the filter output to the z-transform of the filter input*. Suppose that we feed the filter with

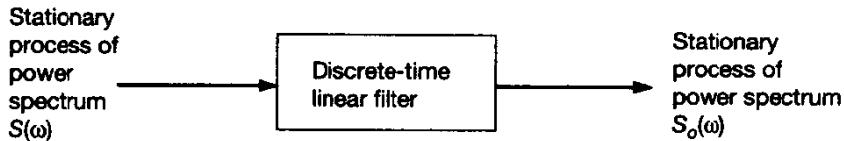


Figure 3.1 Transmission of stationary process through a discrete-time linear filter.

a stationary discrete-time stochastic process of power spectral density $S(\omega)$, as in Fig. 3.1. Let $S_o(\omega)$ denote the power spectral density of the filter output. We may then write

$$S_o(\omega) = |H(e^{j\omega})|^2 S(\omega) \quad (3.21)$$

where $H(e^{j\omega})$ is the *frequency response* of the filter. The frequency response $H(e^{j\omega})$ equals the discrete transfer function $H(z)$ evaluated on the unit circle in the z -plane. The important feature of this result is that the value of the output spectral density at angular frequency ω depends purely on the squared *amplitude response* of the filter and the input power spectral density at the same angular frequency ω .

Equation (3.21) is a fundamental relation in stochastic process theory. To prove it, we may proceed as follows. Let the time series $y(n)$ denote the filter output in Fig. 3.1, produced in response to the time series $u(n)$ applied to the filter input. Assuming that $u(n)$ represents a single realization of a wide-sense stationary discrete-time stochastic process, we find that $y(n)$ also represents a single realization of a wide-sense stationary discrete-time stochastic process modified by the filtering operation. Thus, given that the autocorrelation function of the filter input $u(n)$ is written as

$$r_u(l) = E[u(n)u^*(n - l)]$$

we may express the autocorrelation function of the filter output $y(n)$ in a corresponding way as

$$r_y(l) = E[y(n)y^*(n - l)] \quad (3.22)$$

where $y(n)$ is related to $u(n)$ by the convolution sum

$$y(n) = \sum_{i=-\infty}^{\infty} h(i)u(n-i) \quad (3.23)$$

Similarly, we may write

$$y^*(n-l) = \sum_{k=-\infty}^{\infty} h^*(k)u^*(n-l-k) \quad (3.24)$$

Substituting Eqs. (3.23) and (3.24) in (3.22), and interchanging the orders of expectation and summations, we find that the autocorrelation functions $r_y(l)$ and $r_u(l)$, for lag l , are related as follows:

$$r_y(l) = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(i)h^*(k)r_u(k-i+l) \quad (3.25)$$

Finally, taking the discrete-time Fourier transforms of both sides of Eq. (3.25), and invoking Property 1 of the power spectral density and the fact that the transfer function of a linear filter is equal to the Fourier transform of its impulse response, we get the result described in Eq. (3.21).

Power Spectrum Analyzer

Suppose that the discrete-time linear filter in Fig. 3.1 is designed to have a bandpass characteristic. That is, the amplitude response of the filter is defined by

$$|H(e^{j\omega})| = \begin{cases} 1, & |\omega - \omega_c| \leq \Delta\omega \\ 0, & \text{remainder of the interval } -\pi < \omega \leq \pi \end{cases} \quad (3.26)$$

This amplitude response is depicted in Fig. 3.2. We assume that the *angular bandwidth* of the filter, $2\Delta\omega$, is small enough for the spectrum inside this bandwidth to be essentially constant. Then using Eq. (3.21) we may write

$$S_o(\omega) = \begin{cases} S(\omega_c), & |\omega - \omega_c| \leq \Delta\omega \\ 0, & \text{remainder of the interval } -\pi < \omega \leq \pi \end{cases} \quad (3.27)$$

Next, using Properties 4 and 5 of the power spectral density, we may express the mean-square value of the filter output resulting from a real-valued stochastic input as

$$\begin{aligned} P_o &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_o(\omega) d\omega \\ &= \frac{2\Delta\omega}{2\pi} S(\omega_c) + \frac{2\Delta\omega}{2\pi} S(-\omega_c) \\ &= 2\frac{\Delta\omega}{\pi} S(\omega_c) \quad \text{for real data} \end{aligned}$$

Equivalently, we may write

$$S(\omega_c) = \frac{\pi P_o}{2\Delta\omega} \quad (3.28)$$

where $\Delta\omega/\pi$ is that fraction of the Nyquist interval that corresponds to the passband of the filter. Equation (3.28) states that the value of the power spectral density of the filter input $u(n)$, measured at the center frequency ω_c of the filter, is equal to the mean-square value P_o of the filter output, scaled by a constant factor. We may thus use Eq. (3.28) as the mathematical basis for building a *power spectrum analyzer*, as depicted in Fig. 3.3. Ideally, the discrete-time bandpass filter employed here should satisfy two requirements: *fixed bandwidth* and *adjustable center frequency*. Clearly, in a practical filter design, we can only approximate these two ideal requirements. Note also that the reading of the *average power meter* at the output end of Fig. 3.3 approximates (for finite averaging time) the expected power of an ergodic process $y(n)$.

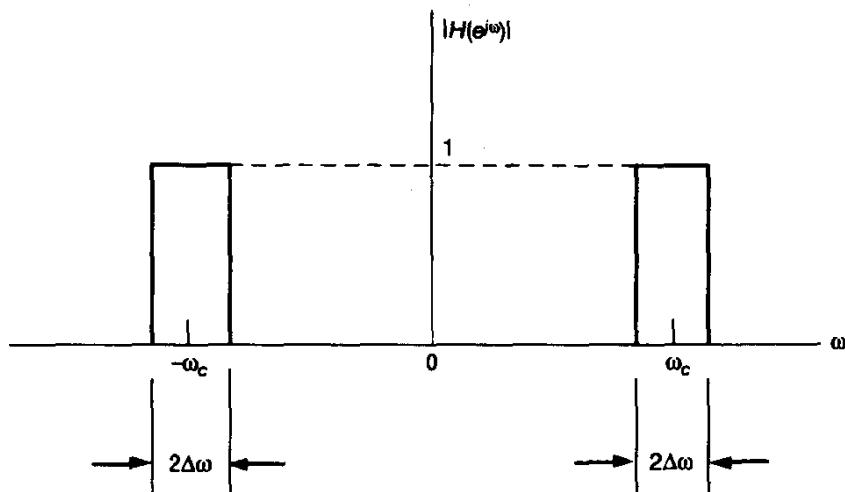


Figure 3.2 Ideal bandpass characteristic.

Example 1: White Noise

A stochastic process of zero mean is said to be *white* if its power spectral density $S(\omega)$ is constant for all frequencies, as shown by

$$S(\omega) = \sigma^2 \quad \text{for } -\pi < \omega \leq \pi$$

where σ^2 is the variance of a sample taken from the process. Suppose that this process is passed through a discrete-time bandpass filter, characterized as in Fig. 3.2. Hence, from Eq. (3.28), we find that the mean-square value of the filter output is

$$P_o = \frac{2\sigma^2\Delta\omega}{\pi}$$

White noise has the property that any two of its samples are uncorrelated, as shown by the autocorrelation function

$$r(\tau) = \sigma^2\delta_{\tau,0}$$

where $\delta_{\tau,0}$ is the Kronecker delta:

$$\delta_{\tau,0} = \begin{cases} 1, & \tau = 0 \\ 0, & \text{otherwise} \end{cases}$$

Time series
representing
a stationary
process

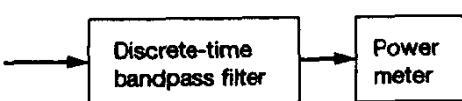


Figure 3.3 Power spectrum analyzer.

3.4 CRAMÉR SPECTRAL REPRESENTATION FOR A STATIONARY PROCESS

Equation (3.12) provides one way of defining the power spectral density of a wide-sense stationary process. Another way of defining the power spectral density is to use the *Cramér spectral representation for a stationary process*. According to this representation, a sample $u(n)$ of a discrete-time stochastic process is written as an inverse Fourier transform (Thomson, 1982, 1988):

$$u(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega n} dZ(\omega) \quad (3.29)$$

If the process represented by the time series $u(n)$ is wide-sense stationary with no periodic components, then the *increment process* $dZ(\omega)$ has the following three properties:

1. The mean of the increment process $dZ(\omega)$ is zero; that is,

$$E[dZ(\omega)] = 0 \quad \text{for all } \omega \quad (3.30)$$

2. The energy of the increment process $dZ(\omega)$ at different frequencies is uncorrelated; that is,

$$E[dZ(\omega)dZ^*(\nu)] = 0 \quad \text{for } \nu \neq \omega \quad (3.31)$$

3. The expected value of $|dZ(\omega)|^2$ defines the spectrum $S(\omega) d\omega$; that is,

$$E[|dZ(\omega)|^2] = S(\omega) d\omega \quad (3.32)$$

In other words, for a wide-sense stationary discrete-time stochastic process represented by the time series $u(n)$, the increment process $dZ(\omega)$ defined by Eq. (3.29) is a *zero-mean orthogonal process*. More precisely, $dZ(\omega)$ may be viewed as a "white process" described in the frequency domain in a manner similar to the time-domain description of ordinary white noise.

Equation (3.32), in conjunction with Eq. (3.31), provides another basic definition for the power spectral density $S(\omega)$. Complex-conjugating both sides of Eq. (3.29) and using ν in place of ω , we get

$$u^*(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\nu n} dZ^*(\nu) \quad (3.33)$$

Hence, multiplying Eq. (3.29) by (3.33), we may express the squared magnitude of $u(n)$ as

$$|u(n)|^2 = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{jn(\omega-\nu)} dZ(\omega) dZ^*(\nu) \quad (3.34)$$

Next, taking the statistical expectation of Eq. (3.34), and interchanging the order of expectation and double integration, we get

$$E[|u(n)|^2] = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{jn(\omega-\nu)} E[dZ(\omega) dZ^*(\nu)] \quad (3.35)$$

If we now use the two basic properties of the increment process $dZ(\omega)$ described by Eqs. (3.31) and (3.32), we may simplify Eq. (3.35) into the form

$$E[|u(n)|^2] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega \quad (3.36)$$

The expectation $E[|u(n)|^2]$ on the left-hand side of Eq. (3.36) is recognized as the mean-square value of the complex sample $u(n)$. The right-hand side of this equation equals the total area under the curve of the power spectral density $S(\omega)$, scaled by the factor $1/2\pi$. Accordingly, Eq. (3.36) is merely a restatement of Property 5 of the power spectral density $S(\omega)$, described by Eq. (3.19).

The Fundamental Equation

Consider the time series $u(0), u(1), \dots, u(N - 1)$, consisting of N observations (samples) of a wide-sense stationary stochastic process. The discrete-time Fourier transform of this time series is given by

$$U_N(\omega) = \sum_{n=0}^{N-1} u(n) e^{-j\omega n} \quad (3.37)$$

According to the Cramér spectral representation of the process, the observation $u(n)$ is given by Eq. (3.29). Hence, using the dummy variable v in place of ω in Eq. (3.29), and then substituting the result in Eq. (3.37), we get

$$U_N(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{n=0}^{N-1} (e^{-j(\omega-v)n}) dZ(v) \quad (3.38)$$

where we have interchanged the order of summation and integration. Define

$$K_N(\omega) = \sum_{n=0}^{N-1} e^{-j\omega n} \quad (3.39)$$

which is known as the *Dirichlet kernel*. The kernel $K_N(\omega)$ represents a geometric series with a first term of unity, a common ratio of $e^{-j\omega}$, and a total number of terms equal to N . Summing this series, we may redefine the kernel $K_N(\omega)$ as follows:

$$\begin{aligned} K_N(\omega) &= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \\ &= \frac{\sin(N\omega/2)}{\sin(\omega/2)} \exp\left[-\frac{1}{2}j\omega(N-1)\right] \end{aligned} \quad (3.40)$$

Note that $K_N(0) = N$. Returning to Eq. (3.38), we may use the definition of the Dirichlet kernel $K_N(\omega)$ given in Eq. (3.39) to rewrite $U_N(\omega)$ as follows:

$$U_N(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} K_N(\omega - v) dZ(v) \quad (3.41)$$

The integral equation (3.41) is a *linear* relation, referred to as the *fundamental equation* of power spectrum analysis.

An integral equation is one that involves an *unknown* function under the integral sign. In the context of power spectrum analysis as described by Eq. (3.41), the increment variable $dZ(\omega)$ is the unknown function, and $U_N(\omega)$ is known. Accordingly, Eq. (3.41) may be viewed as an example of a *Fredholm integral equation of the first kind* (Morse and Feshbach, 1953; Whittaker and Watson, 1965).

Note that $U_N(\omega)$ may be inverse Fourier transformed to recover the original data. It follows therefore that $U_N(\omega)$ is a *sufficient statistic* of the power spectral density. This property makes the use of Eq. (3.41) for spectrum analysis all the more important.

3.5 POWER SPECTRUM ESTIMATION

An issue of practical importance is how to *estimate* the power spectral density of a wide-sense stationary process. Unfortunately, this issue is complicated by the fact that there is a bewildering array of power spectrum estimation procedures, with each procedure purported to have or to show some optimum property. The situation is made worse by the fact that unless care is taken in the selection of the right method, we may end up with misleading conclusions.

Two philosophically different families of power spectrum estimation methods may be identified in the literature: *parametric methods* and *nonparametric methods*. The basic ideas behind these methods are discussed in the sequel.

Parametric Methods

In parametric methods of spectrum estimation we begin by postulating a *stochastic model* for the situation at hand. Depending on the specific form of stochastic model adopted, we may identify three different parametric approaches for spectrum estimation.

1. *Model identification procedures.* In this class of parametric methods, a rational function or a polynomial in $e^{-j\omega}$ is assumed for the transfer function of the model, and a white-noise source is used to drive the model, as depicted in Fig. 3.4. The power spectrum of the resulting model output provides the desired spectrum estimate. Depending on the application of interest, we may adopt one of the following models (Kay and Marple, 1981; Marple, 1987; Kay, 1988):

- (i) *Autoregressive (AR) model* with an all-pole transfer function.
- (ii) *Moving average (MA) model* with an all-zero transfer function.
- (iii) *Autoregressive-moving average (ARMA) model* with pole-zero transfer function.

The resulting power spectra measured at the outputs of these models are referred to as AR, MA, and ARMA spectra, respectively. With reference to the input-output relation of Eq. (3.21), let the power spectrum $S(\omega)$ of the model input be put

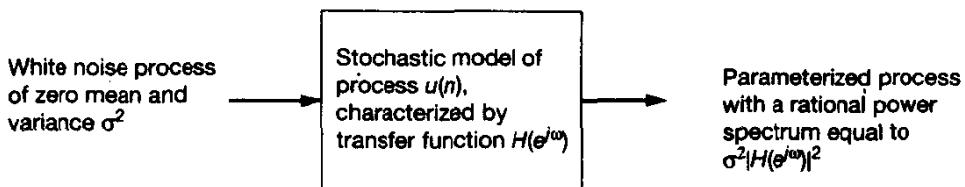


Figure 3.4 Rationale of model identification procedure for power spectrum estimation.

equal to the white noise variance σ^2 . We then find that the power spectrum $S_o(\omega)$ of the model output is equal to the squared amplitude response $|H(e^{j\omega})|^2$ of the model, multiplied by σ^2 . The problem thus becomes one of estimating the model parameters [i.e., parametrizing the transfer function $H(e^{j\omega})$] such that the process produced at the model output provides an acceptable representation (in some statistical sense) of the stochastic process under study. Such an approach to power spectrum estimation may indeed be viewed as a problem in *model (system) identification*.

Among the model-dependent spectra defined herein, the AR spectrum is by far the most popular. The reason for this popularity is twofold: (1) the *linear* form of the system of simultaneous equations involving the unknown AR model parameters, and (2) the availability of efficient algorithms for computing the solution.

2. *Minimum variance distortionless response method.* To describe this second parametric approach for power spectrum estimation, consider the situation depicted in Fig. 3.5. The process $u(n)$ is applied to a transversal filter (i.e., discrete-time filter with an all-zero transfer function). In the *minimum variance distortionless response (MVDR) method*, the filter coefficients are chosen so as to minimize the variance (which is the same as expected power for a zero-mean process) of the filter output, subject to the constraint that the frequency response of the filter is equal to unity at some angular frequency ω_0 . Under this constraint, the process $u(n)$ is passed through the filter with *no distortion* at the angular frequency ω_0 . Moreover, signals at angular frequencies other than ω_0 tend to be attenuated.
3. *Eigendecomposition-based methods.* In this final class of parametric spectrum estimation methods, the eigendecomposition of the ensemble-averaged correla-

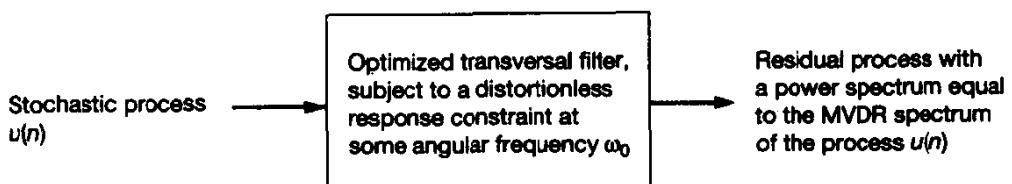


Figure 3.5 Rationale of MVDR procedure for power spectrum estimation.

tion matrix \mathbf{R} of the process $u(n)$ is used to define two disjoint subspaces: *signal subspace* and *noise subspace*. This form of partitioning is then exploited to derive an appropriate algorithm for estimating the power spectrum (Schmidt, 1979, 1981). Eigenanalysis and the notion of subspace decomposition are discussed in the next chapter.

Nonparametric Methods

In nonparametric methods of power spectrum estimation, on the other hand, no assumptions are made with respect to the stochastic process under study. The starting point in the discussion is the fundamental equation (3.41). Depending on the way in which this equation is interpreted, we may distinguish two different nonparametric approaches:

1. *Periodogram-based methods.* Traditionally, the fundamental equation (3.41) is treated as a *convolution* of two frequency functions. One frequency function, $U(\omega)$, represents the discrete-time Fourier transform of an *infinitely long* time series, $u(n)$; this function arises from the definition of the increment variable $dZ(\omega)$ as the product of $U(\omega)$ and the frequency increment $d\omega$. The other frequency function is the kernel $K_N(\omega)$, defined by Eq. (3.40). This approach leads us to consider Eq. (3.12) as the basic definition of the power spectral density $S(\omega)$, and therefore the *periodogram* $|U_N(\omega)|^2/N$ as the starting point for the data analysis. However, the periodogram suffers from a *serious limitation in the sense that it is not a sufficient statistic for the power spectral density*. This implies that the phase information ignored in the use of the periodogram is essential. Consequently, the statistical insufficiency of the periodogram is inherited by any estimate that is based on or equivalent to the periodogram.
2. *Multiple-window method.* A more constructive nonparametric approach is to treat the fundamental equation (3.41) as a *Fredholm integral equation of the first kind* for the increment variable $dZ(\omega)$; the goal here is to obtain an *approximate solution* for the equation with statistical properties that are close to those of $dZ(\omega)$ in some sense (Thomson, 1982). The key to the attainment of this important goal is the use of windows defined by a set of special sequences known as *Slepian sequences*¹ or *discrete prolate spheroidal sequences*, which are fundamental to the study of time- and frequency-limited systems. The remarkable property of this family of windows is that their energy distributions add up in a very special way that collectively defines an ideal (ideal in the sense of the total in-bin versus out-of-bin energy concentration) rectangular frequency bin. This property, in turn, allows us to trade spectral resolution for improved spectral properties (i.e., reduced variance of the spectral estimate).

¹Detailed information on Slepian sequences is given in Slepian (1978). A method for computing them, for large data length, is given in the appendix of the paper by Thomson (1982). For additional information, see the references listed in Thomson's paper. Mullis and Scharf (1991) also present an informative discussion of the role of Slepian sequences in spectrum analysis.

In general, a discrete-time stochastic process $u(n)$ has a *mixed spectrum*, in that its power spectrum contains two components: a deterministic component and a continuous component. The *deterministic component* represents the *first moment* of the increment process $dZ(\omega)$; it is explicitly given by

$$E[dZ(\omega)] = \sum_k a_k \delta(\omega - \omega_k) d\omega \quad (3.42)$$

where $\delta(\omega)$ is the *Dirac delta function* defined in the frequency domain. The ω_k are the angular frequencies of *periodic* or *line components* contained in the process $u(n)$, and the a_k are their amplitudes. The continuous component, on the other hand, represents the *second central moment* of the increment process $dZ(\omega)$, as shown by

$$E[|dZ(\omega) - E[dZ(\omega)]|^2] = S(\omega) d\omega \quad (3.43)$$

It is important that the distinction between the first and second moments is carefully noted.

Spectra computed using the parametric methods tend to have sharper peaks and higher resolution than those obtained from the nonparametric (classical) methods. The application of these parametric methods is therefore well suited for estimating the deterministic component and, in particular, for locating the frequencies of periodic components in additive white noise when the signal-to-noise ratio is high. Another well-proven technique for estimating the deterministic component is the classical method of maximum likelihood, which is discussed in Appendix D. Of course, if the physical laws governing the generation of a process match a stochastic model (e.g., AR model) in an exact manner or approximately in some statistical sense, then the parametric method corresponding to that model may be used to estimate the power spectrum of the process. If, however, the stochastic process of interest has a purely continuous power spectrum, and the underlying physical mechanism responsible for the generation of the process is unknown, then the recommended procedure is the non-parametric method of multiple windows.

In this book, we confine our attention to classes 1 and 2 of parametric methods of spectrum estimation, as their theory fits naturally under the umbrella of adaptive filters. For a comprehensive discussion of the other methods of spectrum analysis, the reader is referred to the books by Gardner (1987), Marple (1987), and Kay (1988), the paper by Thomson (1982), and a chapter contribution by Mullis and Scharf (1991).

3.6 OTHER STATISTICAL CHARACTERISTICS OF A STOCHASTIC PROCESS

In the material presented in the previous chapter and up to this point in the present chapter, we have focused our attention on a partial characterization of a discrete-time stochastic process. According to this particular characterization, we only need to specify the mean as the first moment of the process and its autocorrelation function as the second moment. Since the autocorrelation function and power spectral density form a Fourier-transform pair, we may equally well specify the power spectral density in place of the autocorrelation function. The use of second-order statistics as described herein is adequate for the study of linear adaptive filters. However, when we move on later in the book to consider

difficult applications (e.g., blind deconvolution) that are beyond the reach of linear adaptive filters, we will have to resort to the use of other statistical properties of a stochastic process.

Two particular statistical properties that bring in additional information about a stochastic process, which can prove useful in practice, are as follows:

1. *High-order statistics.* An obvious way of expanding the characterization of a stationary stochastic process is to include *higher-order statistics* (HOS) of the process. This is done by invoking the use of *cumulants* and their Fourier transforms, known as *polyspectra*. Indeed, cumulants and polyspectra of a zero-mean stochastic process may be viewed as generalizations of the autocorrelation function and power spectral density, respectively. It is important to note that higher-order statistics are only meaningful in the context of *non-Gaussian processes*. Furthermore, to exploit them, we need to use some form of nonlinear filtering.
2. *Cyclostationarity.* In an important class of stochastic processes commonly encountered in practice, the mean and autocorrelation function of the process exhibit periodicity, as in

$$\mu(t_1 + T) = \mu(t_1) \quad (3.44)$$

$$r(t_1 + T, t_2 + T) = r(t_1, t_2) \quad (3.45)$$

for all t_1 and t_2 . Both t_1 and t_2 represent values of the continuous-time variable t , and T denotes period. A stochastic process satisfying Eqs. (3.44) and (3.45) is said to be *cyclostationary* in the wide sense (Franks, 1969; Gardner and Franks, 1975; Gardner, 1994). Modeling a stochastic process as cyclostationary adds a new dimension, namely, the period T , to the partial description of the process. Examples of cyclostationary processes include a modulated process obtained by varying the amplitude, phase, or frequency of a sinusoidal carrier. Note that, unlike higher-order statistics, cyclostationarity can be exploited by means of linear filtering.

In the sequel, we will discuss these two specific aspects of stochastic processes under the section headings "polyspectra" and "spectral-correlation density." As already mentioned, polyspectra provide a frequency-domain description of the higher-order statistics of a stationary stochastic process. By the same token, spectral-correlation density provides a frequency-domain description of a cyclostationary stochastic process.

3.7 POLYSPECTRA

Consider a stationary stochastic process $u(n)$ with zero mean; that is,

$$E[u(n)] = 0 \quad \text{for all } n$$

Let $u(n); u(n + \tau_1), \dots, u(n + \tau_{k-1})$ denote the random variables obtained by observing this stochastic process at times $n, n + \tau_1, \dots, n + \tau_{k-1}$, respectively. These random variables form a k -by-1 vector:

$$\mathbf{u} = [u(n), u(n + \tau_1), \dots, u(n + \tau_{k-1})]^T \quad (3.46)$$

Correspondingly, define a k -by-1 vector:

$$\mathbf{z} = [z_1, z_2, \dots, z_k]^T \quad (3.47)$$

We may then define the k th-order cumulant of the stochastic process $u(n)$, denoted by $c_k(\tau_1, \tau_2, \dots, \tau_{k-1})$, as the coefficient of the vector \mathbf{z} in the Taylor expansion of the cumulant-generating function (Priestley, 1981; Swami and Mendel, 1990; Gardner, 1994):

$$K(\mathbf{z}) = \ln E[\exp(\mathbf{z}^T \mathbf{u})] \quad (3.48)$$

The k th-order cumulant of the process $u(n)$ is thus defined in terms of its joint moments of orders up to k ; to simplify the presentation in this section, we assume that $u(n)$ is real valued. Specifically, the second-, third-, and fourth-order cumulants are given, respectively, by

$$c_2(\tau) = E[u(n)u(n + \tau)] \quad (3.49)$$

$$c_3(\tau_1, \tau_2) = E[u(n)u(n + \tau_1)u(n + \tau_2)] \quad (3.50)$$

and

$$\begin{aligned} c_4(\tau_1, \tau_2, \tau_3) &= E[u(n)u(n + \tau_1)u(n + \tau_2)u(n + \tau_3)] \\ &\quad - E[u(n)u(n + \tau_1)]E[u(n + \tau_2)u(n + \tau_3)] \\ &\quad - E[u(n)u(n + \tau_2)]E[u(n + \tau_3)u(n + \tau_1)] \\ &\quad - E[u(n)u(n + \tau_3)]E[u(n + \tau_1)u(n + \tau_2)] \end{aligned} \quad (3.51)$$

From the definitions given in Eqs. (3.49) to (3.51), we note the following:

1. The second-order cumulant $c_2(\tau)$ is the same as the autocorrelation function $r(\tau)$.
2. The third-order cumulant $c_3(\tau_1, \tau_2)$ is the same as the third-order moment $E[u(n)u(n + \tau_1)u(n + \tau_2)]$.
3. The fourth-order cumulant $c_4(\tau_1, \tau_2, \tau_3)$ is *different* from the fourth-order moment $E[u(n)u(n + \tau_1)u(n + \tau_2)u(n + \tau_3)]$. In order to generate the fourth-order cumulant, we need to know the fourth-order moment and six different values of the autocorrelation function.

Note that the k th-order cumulant $c(\tau_1, \tau_2, \dots, \tau_{k-1})$ does not depend on time n . For this to be valid, however, the process $u(n)$ has to be stationary up to order k . A process $u(n)$ is said to be *stationary up to order k* if, for any admissible $\{n_1, n_2, \dots, n_p\}$ all the joint moments up to order k of $\{u(n_1), u(n_2), \dots, u(n_p)\}$ exist and equal the corresponding

joint moments up to order k of $\{u(n_1 + \tau), u(n_2 + \tau), \dots, u(n_p + \tau)\}$ where $\{n_1 + \tau, n_2 + \tau, \dots, n_p + \tau\}$ is an admissible set too (Priestley, 1981).

Consider next a linear time-invariant system, characterized by the impulse response h_n . Let the system be excited by a process $x(n)$ consisting of independent and identically distributed (iid) random variables. Let $u(n)$ denote the resulting system output. The k th-order cumulant of $u(n)$ is given by

$$c_k(\tau_1, \tau_2, \dots, \tau_{k-1}) = \gamma_k \sum_{i=-\infty}^{\infty} h_i h_{i+\tau_1} \cdots h_{i+\tau_{k-1}} \quad (3.52)$$

where γ_k is the k th-order cumulant of the input process $x(n)$. Note that the summation term on the right-hand side of Eq. (3.52) has a form similar to that of a k th-order moment, except that the expectation operator has been replaced by a summation.

The k th-order polyspectrum (or k th-order cumulant spectrum) is defined by (Priestley, 1981; Nikias and Raghubeer, 1987):

$$\begin{aligned} C_k(\omega_1, \omega_2, \dots, \omega_{k-1}) &= \sum_{\tau_1=-\infty}^{\infty} \cdots \sum_{\tau_{k-1}=-\infty}^{\infty} c_k(\tau_1, \tau_2, \dots, \tau_{k-1}) \\ &\cdot \exp[-j(\omega_1 \tau_1 + \omega_2 \tau_2 + \dots + \omega_{k-1} \tau_{k-1})] \end{aligned} \quad (3.53)$$

A sufficient condition for the existence of the polyspectrum $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$ is that the associated k th-order cumulant $c_k(\tau_1, \tau_2, \dots, \tau_{k-1})$ be absolutely summable, as shown by

$$\sum_{\tau_1=-\infty}^{\infty} \cdots \sum_{\tau_{k-1}=-\infty}^{\infty} |c_k(\tau_1, \tau_2, \dots, \tau_{k-1})| < \infty \quad (3.54)$$

The *power spectrum*, *bispectrum*, and *trispectrum* are special cases of the k th-order polyspectrum defined in Eq. (3.53). Specifically, we may state the following:

1. For $k = 2$, we have the ordinary power spectrum:

$$C_2(\omega_1) = \sum_{\tau_1=-\infty}^{\infty} c_2(\tau_1) \exp(-j\omega_1 \tau_1) \quad (3.55)$$

which is a restatement of the Einstein–Wiener–Khintchine relation, namely, Eq. (3.15).

2. For $k = 3$, we have the *bispectrum*, defined by

$$C_3(\omega_1, \omega_2) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} c_3(\tau_1, \tau_2) \exp[-j(\omega_1 \tau_1 + \omega_2 \tau_2)] \quad (3.56)$$

3. For $k = 4$, we have the *trispectrum*, defined by

$$C_4(\omega_1, \omega_2, \omega_3) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} \sum_{\tau_3=-\infty}^{\infty} c_4(\tau_1, \tau_2, \tau_3) \exp[-j(\omega_1 \tau_1 + \omega_2 \tau_2 + \omega_3 \tau_3)] \quad (3.57)$$

An outstanding property of polyspectrum is that all polyspectra of order higher than the second vanish when the process $u(n)$ is Gaussian. This property is a direct consequence of the fact that all joint cumulants of order higher than the second are zero for multivariate Gaussian distributions. Accordingly, the bispectrum, trispectrum, and all higher-order polyspectra are identically zero if the process $u(n)$ is Gaussian. Thus, higher-order spectra provide measures of the *departure of a stochastic process from Gaussianity* (Priestley, 1981; Nikias and Raghuvir, 1987).

The k th-order cumulant $c_k(\tau_1, \tau_2, \dots, \tau_{k-1})$ and the k th-order polyspectrum $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$ form a pair of multidimensional Fourier transforms. Specifically, the polyspectrum $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$, is the *multidimensional discrete-time Fourier transform* of $c_k(\tau_1, \tau_2, \dots, \tau_{k-1})$, and $c_k(\tau_1, \tau_2, \dots, \tau_{k-1})$ is the *inverse multidimensional discrete-time Fourier transform* of $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$.

For example, given the bispectrum $C_3(\omega_1, \omega_2)$, we may determine the third-order cumulant $c_3(\tau_1, \tau_2)$ by using the inverse two-dimensional discrete-time Fourier transform:

$$c_3(\tau_1, \tau_2) = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} C_3(\omega_1, \omega_2) \exp[j(\omega_1\tau_1 + \omega_2\tau_2)] d\omega_1 d\omega_2 \quad (3.58)$$

We may use this relation to develop an alternative definition of the bispectrum as follows. According to the *Cramér spectral representation*, we have

$$u(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega n} dZ(\omega) \quad \text{for all } n \quad (3.59)$$

Hence, using Eq. (3.59) in (3.50), we get

$$\begin{aligned} c_3(\tau_1, \tau_2) &= \left(\frac{1}{2\pi}\right)^3 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \exp[jn(\omega_1 + \omega_2 + \omega_3)] \\ &\quad \cdot \exp[j(\omega_1\tau_1 + \omega_2\tau_2)] E[dZ(\omega_1) dZ(\omega_2) dZ(\omega_3)] \end{aligned} \quad (3.60)$$

Comparing the right-hand sides of Eqs. (3.58) and (3.60), we deduce the following result:

$$E[dZ(\omega_1) dZ(\omega_2) dZ(\omega_3)] = \begin{cases} C_3(\omega_1, \omega_2) d\omega_1 d\omega_2, & \omega_1 + \omega_2 + \omega_3 = 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.61)$$

It is apparent from Eq. (3.61) that the bispectrum $C_3(\omega_1, \omega_2)$ represents the contribution to the mean product of three Fourier components whose *individual frequencies add up to zero*. This is an extension of the interpretation developed for the ordinary power spectrum in Section 3.3. In a similar manner we may develop an interpretation of the trispectrum.

In general, the polyspectrum $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$ is *complex for order k higher than two*, as shown by

$$C_k(\omega_1, \omega_2, \dots, \omega_{k-1}) = |C_k(\omega_1, \omega_2, \dots, \omega_{k-1})| \exp[j\phi_k(\omega_1, \omega_2, \dots, \omega_{k-1})] \quad (3.62)$$

where we note that $|C_k(\omega_1, \omega_2, \dots, \omega_{k-1})|$ is the *magnitude* of the polyspectrum, and $\phi_k(\omega_1, \omega_2, \dots, \omega_{k-1})$ is the *phase*. Moreover, the polyspectrum is a *periodic function* with

period 2π ; that is,

$$C_k(\omega_1, \omega_2, \dots, \omega_{k-1}) = C_k(\omega_1 + 2\pi, \omega_2 + 2\pi, \dots, \omega_{k-1} + 2\pi) \quad (3.63)$$

Whereas the power spectral density of a stationary stochastic process is *phase blind*, the polyspectra of the process are *phase sensitive*. More specifically, the power spectral density is real-valued; referring to the input-output relation of Eq. (3.21), we clearly see that in passing a stationary stochastic process through a linear system, information about the phase response of the system is completely destroyed in the power spectrum of the output. In contrast, the polyspectrum is complex-valued, with the result that in a similar situation the polyspectrum of the output signal preserves information about the phase response of the system. It is for this reason that polyspectra provide a useful tool for the "blind" identification of an unknown system, where we only have access to the output signal and some additional information in the form of a probabilistic model of the input signal. We will have more to say on this issue in Chapter 18.

3.8 SPECTRAL-CORRELATION DENSITY

Polyspectra preserve phase information about a stochastic process by invoking higher-order statistics of the process, which is feasible only if the process is non-Gaussian. The preservation of phase information is also possible if the process is *cyclostationary* in the wide sense, as defined in Eqs. (3.44) and (3.45). This latter approach has two important advantages over the higher-order statistics approach:

- The phase information is contained in second-order cyclostationary statistics of the process; hence, the phase information can be exploited in a computationally efficient manner that avoids the use of higher-order statistics.
- Preservation of the phase information holds, irrespective of Gaussianity.

Consider then a discrete-time stochastic process $u(n)$ that is cyclostationary in the wide sense. Without loss of generality, the process is assumed to have zero mean. The ensemble-average autocorrelation function of the process $u(n)$ is defined in the usual way by Eq. (2.6), reproduced here for convenience of presentation:

$$r(n, n - k) = E[u(n)u^*(n - k)] \quad (3.64)$$

Under the condition of cyclostationarity, the autocorrelation function $r(n, n - k)$ is periodic in n for every k . Keeping in mind the discrete-time nature of the process $u(n)$, we may expand the autocorrelation function $r(n, n - k)$ into a complex Fourier series as follows (Gardner, 1994):

$$r(n, n - k) = \sum_{\{\alpha\}} r^\alpha(k) e^{j2\pi\alpha n - j\pi\alpha k} \quad (3.65)$$

where both n and k take on only integer values, and the set $\{\alpha\}$ includes all values of α for which the corresponding Fourier coefficient $r^\alpha(k)$ is not zero. The Fourier coefficient $r^\alpha(k)$

is itself defined by

$$r^\alpha(k) = \frac{1}{N} \sum_{n=0}^{N-1} r(n, n-k) e^{-j2\pi\alpha n + j\pi\alpha k} \quad (3.66)$$

where the number of samples N denotes the period. Equivalently, in light of Eq. (3.64), we may define $r^\alpha(k)$ as

$$r^\alpha(k) = \frac{1}{N} \left\{ \sum_{n=0}^{N-1} E[u(n)u^*(n-k)e^{-j2\pi\alpha n}] \right\} e^{j\pi\alpha k} \quad (3.67)$$

The quantity $r^\alpha(k)$ is called the *cyclic autocorrelation function*, which has the following properties:

1. The cyclic autocorrelation function $r^\alpha(k)$ is periodic in α with period two.

2. For any α , we have from Eq. (3.67):

$$r^{\alpha+1}(k) = (-1)^k r^\alpha(k) \quad (3.68)$$

3. For the special case of $\alpha = 0$, Eq. (3.67) reduces to

$$r^0(k) = r(k) \quad (3.69)$$

where $r(k)$ is the ordinary autocorrelation function of a stationary process.

According to the Einstein–Wiener–Khintchine relations of Eqs. (3.15) and (3.16), the ordinary versions of the autocorrelation function and power spectral density of a wide-sense stationary stochastic process form a Fourier-transform pair. In a corresponding way, we may define the discrete-time Fourier transform of the cyclic autocorrelation function $r^\alpha(k)$ as follows (Gardner, 1994):

$$S^\alpha(\omega) = \sum_{k=-\infty}^{\infty} r^\alpha(k) e^{-j\omega k}, \quad -\pi < \omega \leq \pi \quad (3.70)$$

The new quantity $S^\alpha(\omega)$ is called the *spectral-correlation density*, which is complex valued for $\alpha \neq 0$. Note that for the special case of $\alpha = 0$, Eq. (3.70) reduces to

$$S^0(\omega) = S(\omega) \quad (3.71)$$

where $S(\omega)$ is the ordinary power spectral density.

In light of the defining equations (3.67) and (3.70), we may set up the block diagram of Fig. 3.6 for measuring the spectral-correlation density $S^\alpha(\omega)$. For this measurement, it is assumed that the process $u(n)$ is *cycloergodic* (Gardner, 1994), which means that time averages may be substituted for ensemble averages "with samples taken once per period." According to the instrumentation described here, $S^\alpha(\omega)$ is the bandwidth-normalized version of the cross-correlation narrow-band spectral components contained in the time series $u(n)$ at the angular frequencies $\omega + \alpha\pi$ and $\omega - \alpha\pi$, in the limit as the bandwidth of these spectral components is permitted to approach zero (Gardner, 1994). Note that the two narrow-band filters in Fig. 3.6 are identical, both having a mid-band (angular) frequency ω and a bandwidth $\Delta\omega$ that is small compared to ω , but large compared to the reciprocal of

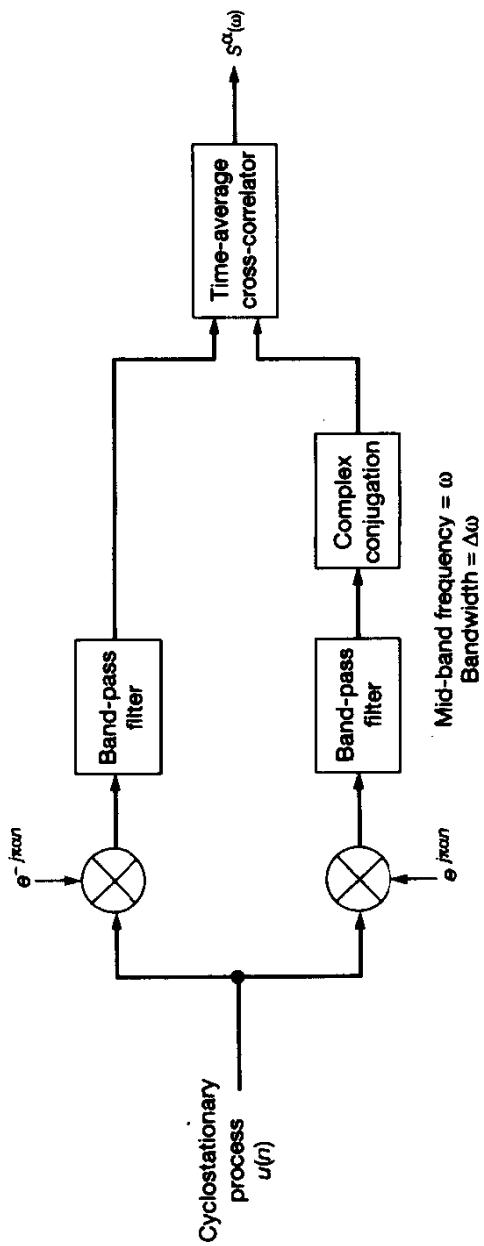


Figure 3.6 Scheme for measuring the spectral-correlation density of a cyclostationary process.

the averaging time used in the cross-correlator at the output end in Fig. 3.6. In one channel of this scheme the input $u(n)$ is multiplied by $\exp(-j\pi\alpha n)$, and in the other channel it is multiplied by $\exp(j\pi\alpha n)$; the resulting filtered signals are then applied to the cross-correlator. It is these two multiplications (prior to correlation) that provide the spectral-correlation density $S^\alpha(\omega)$ with a phase-preserving property for nonzero values of α .

3.9 SUMMARY AND DISCUSSION

In this chapter we discussed various aspects of spectrum analysis pertaining to a discrete-time stochastic process. In particular, we identified three distinct spectral parameters, depending on the statistical characterization of the process, as summarized here:

1. *Power spectral density*, $S(\omega)$, defined as the discrete-time Fourier transform of the ordinary autocorrelation function of a wide-sense stationary process. For such a process, the autocorrelation function is Hermitian, which makes the power spectral density $S(\omega)$ a real-valued quantity. Accordingly, $S(\omega)$ destroys phase information about the process. Despite this limitation, the power spectral density is commonly accepted as a useful parameter for displaying the correlation properties of a wide-sense stationary process.
2. *Polyspectra*, $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$, defined as the multidimensional Fourier transform of the cumulants of a stationary process. For second-order statistics, $k = 2$, and $C_2(\omega_1)$ reduces to the ordinary power spectral density $S(\omega)$. For higher-order statistics, $k > 2$, and the polyspectra $C_k(\omega_1, \omega_2, \dots, \omega_{k-1})$ take on complex forms. It is this property of polyspectra that makes them a useful tool for dealing with situations where knowledge of phase is a necessary requirement. However, for polyspectra to be meaningful, the process has to be non-Gaussian, and the exploitation of phase information contained in polyspectra requires the use of nonlinear filtering.
3. *Spectral-correlation density*, $S^\alpha(\omega)$, defined as the discrete-time Fourier transform of the cyclic autocorrelation function of a process that is cyclostationary in the wide sense. For $\alpha \neq 0$, $S^\alpha(\omega)$ is complex valued; for $\alpha = 0$, it reduces to $S(\omega)$. The useful feature of $S^\alpha(\omega)$ is that it preserves phase information, which can be exploited by means of linear filtering.

The different properties of the ordinary power spectral density, polyspectra, and spectral-correlation density give these statistical parameters their own individual areas of application.

One last comment is in order. The theories of second-order cyclostationary processes and conventional polyspectra have been brought together under the umbrella of *cyclic polyspectra*. Simply stated, cyclic polyspectra are spectral cumulants, in which the individual frequencies involved can add up to any cycle frequency α , whereas they must

add up to zero for conventional polyspectra. For a detailed treatment of cyclic polyspectra, the interested reader is referred to (Gardner and Spooner, 1994; Spooner and Gardner, 1994).

PROBLEMS

1. Consider the definition given in Eq. (3.12) for the power spectral density. Is it permissible to interchange the operation of taking the limit and that of the expectation in this equation? Justify your answer.
2. In deriving Eq. (3.25), we invoked the notion that if a wide-sense stationary process is applied to a linear, time-invariant, and stable filter, the stochastic process produced at the filter output is wide-sense stationary too. Show that, in general,

$$r_y(n, m) = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(i) h^*(k) r_u(n-i, m-k)$$

which includes the result of Eq. (3.25) as a special case.

3. The mean-square value of the filter output in Eq. (3.28) assumes that the bandwidth of the filter is small compared to its midband frequency. Is this assumption necessary for the corresponding result obtained in Example 1 for a white-noise process? Justify your answer.
4. A white-noise process with a variance of 0.1 V squared is applied to a low-pass discrete-time filter whose bandwidth is 1 Hz. The process is real.
 - (a) Calculate the variance of the filter output.
 - (b) Assuming that the input is Gaussian, determine the probability density function of the filter output.
5. Justify the fact that the expectation of $|dZ(\omega)|^2$ has the physical significance of power.
6. Show that the third- and higher-order cumulants of a Gaussian process are all identically zero.
7. Develop a physical interpretation of the trispectrum $C_4(\omega_1, \omega_2, \omega_3)$ of a stationary stochastic process $u(n)$; assume that $u(n)$ is real valued.
8. Consider a linear time-invariant system whose transfer function is $H(z)$. The system is excited by a real-valued sequence $x(n)$ of independently and identically distributed (iid) random variables with zero mean and unit variance. The probability distribution of $x(n)$ is nonsymmetric.
 - (a) Evaluate the third-order cumulant and bispectrum of the system output $u(n)$.
 - (b) Show that the phase component of the bispectrum of $u(n)$ is related to the phase response of the system transfer function $H(z)$ as follows:

$$\arg[C_3(\omega_1, \omega_2)] = \arg[H(e^{j\omega_1})] + \arg[H(e^{j\omega_2})] - \arg[H(e^{j(\omega_1+\omega_2)})]$$

9. Equation (3.52) gives the k th-order cumulant of the output of a linear time-invariant system of impulse response h_n that is driven by a sequence $x(n)$ of independent and identically distributed random variables. Prove the validity of this equation.

10. Show that for a stochastic process $u(n)$ that is cyclostationary in the wide sense, the cyclic autocorrelation function $r^\alpha(k)$ satisfies the property

$$r^\alpha(-k) = r^{\alpha*}(k)$$

where the asterisk denotes complex conjugation.

11. Figure 3.6 describes a method for measuring the spectral-correlation density of a time series $u(n)$ that is representative of a cyclostationary process in the wide sense. For $\alpha = 0$, show that Fig. 3.6 reduces to the simpler form shown in Fig. 3.3.

CHAPTER

4

Eigenanalysis

In this chapter we continue the statistical characterization of a discrete-time stochastic process that is stationary in the wide sense. From Chapter 2 we recall that the ensemble-averaged correlation matrix of such a process is Hermitian. An important aspect of a Hermitian matrix is that it permits a useful decomposition of the matrix in terms of its eigenvalues and associated eigenvectors. This form of representation is commonly referred to as *eigenanalysis*, which is basic to the study of digital signal processing.

We begin the discussion of eigenanalysis by outlining the eigenvalue problem in the context of the correlation matrix. We then study the properties of eigenvalues and eigenvectors of the correlation matrix and a related optimum filtering problem. We finish the discussion by briefly describing canned routines for eigenvalue computations and some related issues.

4.1 THE EIGENVALUE PROBLEM

Let the Hermitian matrix \mathbf{R} denote the M -by- M correlation matrix of a wide-sense stationary discrete-time stochastic process represented by the M -by-1 observation vector $\mathbf{u}(n)$. In general, this matrix may contain complex elements. We wish to find an M -by-1 vector \mathbf{q} that satisfies the condition

$$\mathbf{R}\mathbf{q} = \lambda\mathbf{q} \quad (4.1)$$

for some constant λ . This condition states that the vector \mathbf{q} is linearly transformed to the vector $\lambda\mathbf{q}$ by the Hermitian matrix \mathbf{R} . Since λ is a constant, the vector \mathbf{q} therefore has special significance in that it is left *invariant in direction* (in the M -dimensional space) by a linear transformation. For a typical M -by- M matrix \mathbf{R} , there will be M such vectors. To show this, we first rewrite Eq. (4.1) in the form

$$(\mathbf{R} - \lambda\mathbf{I})\mathbf{q} = \mathbf{0} \quad (4.2)$$

where \mathbf{I} is the M -by- M identity matrix, and $\mathbf{0}$ is the M -by-1 null vector. The matrix $(\mathbf{R} - \lambda\mathbf{I})$ has to be singular. Hence, Eq. (4.2) has a nonzero solution in the vector \mathbf{q} if and only if the determinant of the matrix $(\mathbf{R} - \lambda\mathbf{I})$ equals zero; that is,

$$\det(\mathbf{R} - \lambda\mathbf{I}) = 0 \quad (4.3)$$

This determinant, when expanded, is clearly a polynomial in λ of degree M . We thus find that, in general, Eq. (4.3) has M distinct roots. Correspondingly, Eq. (4.3) has M solutions in the vector \mathbf{q} .

Equation (4.3) is called the *characteristic equation* of the matrix \mathbf{R} . Let $\lambda_1, \lambda_2, \dots, \lambda_M$ denote the M roots of this equation. These roots are called the *eigenvalues* of the matrix \mathbf{R} . Although the M -by- M matrix \mathbf{R} has M eigenvalues, they need not be distinct. When the characteristic equation (4.3) has multiple roots, the matrix \mathbf{R} is said to have *degenerate eigenvalues*. Note that, in general, the use of root finding in the characteristic equation (4.3) is a poor method for computing the eigenvalues of the matrix \mathbf{R} ; the issue of eigenvalue computations is considered later in Section 4.5.

Let λ_i denote the i th eigenvalue of the matrix \mathbf{R} . Also, let \mathbf{q}_i be a nonzero vector such that

$$\mathbf{R}\mathbf{q}_i = \lambda_i\mathbf{q}_i \quad (4.4)$$

The vector \mathbf{q}_i is called the *eigenvector* associated with λ_i . An eigenvector can correspond to only one eigenvalue. However, an eigenvalue may have many eigenvectors. For example, if \mathbf{q}_i is an eigenvector associated with eigenvalue λ_i , then so is $a\mathbf{q}_i$ for any scalar $a \neq 0$.

Example 1: White Noise

Consider the M -by- M correlation matrix of a white-noise process that is described by the diagonal matrix

$$\mathbf{R} = \text{diag}(\sigma^2, \sigma^2, \dots, \sigma^2)$$

where σ^2 is the variance of a sample of the process. This correlation matrix \mathbf{R} has a single degenerate eigenvalue equal to the variance σ^2 with multiplicity M . Any M -by-1 random vector qualifies as the associated eigenvector, which shows that (for white noise) one eigenvalue σ^2 has M eigenvectors.

Example 2: Complex Sinusoid

Consider next the M -by- M correlation matrix of a time series whose elements are samples of a complex sinusoid with random phase and unit power. This correlation matrix may be written as

$$\mathbf{R} = \begin{bmatrix} 1 & e^{j\omega} & \cdots & e^{j(M-1)\omega} \\ e^{-j\omega} & 1 & \cdots & e^{j(M-2)\omega} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ e^{-j(M-1)\omega} & e^{-j(M-2)\omega} & \cdots & 1 \end{bmatrix}$$

where ω is the angular frequency of the complex sinusoid. The M -by-1 vector

$$\mathbf{q} = [1, e^{-j\omega}, \dots, e^{-j(M-1)\omega}]^T$$

is an eigenvector of the correlation matrix \mathbf{R} , and the corresponding eigenvalue is M (i.e., the dimension of the matrix \mathbf{R}). In other words, a complex sinusoidal vector represents an eigenvector of its own correlation matrix, except for the trivial operation of complex conjugation.

Note that the correlation matrix \mathbf{R} has *rank 1*, which means that any column of \mathbf{R} may be expressed as a linear combination of the remaining columns (i.e., the matrix \mathbf{R} has only one independent column). It also means that the other eigenvalue is zero with multiplicity $M - 1$, and this eigenvalue has $M - 1$ eigenvectors.

4.2 PROPERTIES OF EIGENVALUES AND EIGENVECTORS

In this section we discuss the various properties of the eigenvalues and eigenvectors of the correlation matrix \mathbf{R} of a stationary discrete-time stochastic process. Some of the properties derived here are direct consequences of the Hermitian property and the nonnegative definiteness of the correlation matrix \mathbf{R} , which were established in Section 2.3.

Property 1. *If $\lambda_1, \lambda_2, \dots, \lambda_M$ denote the eigenvalues of the correlation matrix \mathbf{R} , then the eigenvalues of the matrix \mathbf{R}^k equal $\lambda_1^k, \lambda_2^k, \dots, \lambda_M^k$ for any integer $k > 0$.*

Repeated premultiplication of both sides of Eq. (4.1) by the matrix \mathbf{R} yields

$$\mathbf{R}^k \mathbf{q} = \lambda^k \mathbf{q} \quad (4.5)$$

This shows that (1) if λ is an eigenvalue of \mathbf{R} , then λ^k is an eigenvalue of \mathbf{R}^k , which is the desired result, and (2) every eigenvector of \mathbf{R} is also an eigenvector of \mathbf{R}^k .

Property 2. *Let $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ be the eigenvectors corresponding to the distinct eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ of the M -by- M correlation matrix \mathbf{R} , respectively. Then the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ are linearly independent.*

We say that the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ are *linearly dependent* if there are scalars v_1, v_2, \dots, v_M , not all zero, such that

$$\sum_{i=1}^M v_i \mathbf{q}_i = \mathbf{0} \quad (4.6)$$

If no such scalars exist, we say that the eigenvectors are *linearly independent*.

We will prove the validity of Property 2 by contradiction. Suppose that Eq. (4.6) holds for certain not all zero scalars v_i . Repeated multiplication of Eq. (4.6) by matrix \mathbf{R} and the use of Eq. (4.5) yield the following set of M equations:

$$\sum_{i=1}^M v_i \lambda_i^{k-1} \mathbf{q}_i = \mathbf{0}, \quad k = 1, 2, \dots, M \quad (4.7)$$

This set of equations may be written in the form of a single matrix equation:

$$[v_1 \mathbf{q}_1, v_2 \mathbf{q}_2, \dots, v_M \mathbf{q}_M] \mathbf{S} = \mathbf{0} \quad (4.8)$$

where

$$\mathbf{S} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{M-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_M & \lambda_M^2 & \cdots & \lambda_M^{M-1} \end{bmatrix} \quad (4.9)$$

The matrix \mathbf{S} is called a *Vandermonde matrix* (Strang, 1980). When the λ_i are distinct, the Vandermonde matrix \mathbf{S} is nonsingular. Therefore, we may postmultiply Eq. (4.8) by the inverse matrix \mathbf{S}^{-1} , obtaining

$$[v_1 \mathbf{q}_1, v_2 \mathbf{q}_2, \dots, v_M \mathbf{q}_M] = \mathbf{0}$$

Hence, each column $v_i \mathbf{q}_i = \mathbf{0}$. Since the eigenvectors \mathbf{q}_i are not zero, this condition can be satisfied if and only if the v_i are all zero. This contradicts the assumption that the scalars v_i are not all zero. In other words, the eigenvectors are linearly independent.

We may put this property to an important use by having the linearly independent eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ serve as a *basis* for the representation of an arbitrary vector \mathbf{w} with the same dimension as the eigenvectors themselves. In particular, we may express the arbitrary vector \mathbf{w} as a linear combination of the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ as follows:

$$\mathbf{w} = \sum_{i=1}^M v_i \mathbf{q}_i \quad (4.10)$$

where v_1, v_2, \dots, v_M are constants. Suppose now we apply a linear transformation to the vector \mathbf{w} by premultiplying it by the matrix \mathbf{R} , obtaining

$$\mathbf{R}\mathbf{w} = \sum_{i=1}^M v_i \mathbf{R}\mathbf{q}_i \quad (4.11)$$

By definition, we have $\mathbf{R}\mathbf{q}_i = \lambda_i \mathbf{q}_i$. Therefore, we may express the result of this linear transformation in the equivalent form

$$\mathbf{R}\mathbf{w} = \sum_{i=1}^M v_i \lambda_i \mathbf{q}_i \quad (4.12)$$

We thus see that when a linear transformation is applied to an arbitrary vector \mathbf{w} defined in Eq. (4.10), the eigenvectors remain independent of each other, and the effect of the transformation is simply to multiply each eigenvector by its respective eigenvalue.

Property 3. *Let $\lambda_1, \lambda_2, \dots, \lambda_M$ be the eigenvalues of the M -by- M correlation matrix \mathbf{R} . Then all these eigenvalues are real and nonnegative.*

To prove this property, we first use Eq. (4.1) to express the condition on the i th eigenvalue λ_i as

$$\mathbf{R}\mathbf{q}_i = \lambda_i\mathbf{q}_i, \quad i = 1, 2, \dots, M \quad (4.13)$$

Premultiplying both sides of this equation by \mathbf{q}_i^H , the Hermitian transpose of eigenvector \mathbf{q}_i , we get

$$\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i = \lambda_i \mathbf{q}_i^H \mathbf{q}_i, \quad i = 1, 2, \dots, M \quad (4.14)$$

The inner product $\mathbf{q}_i^H \mathbf{q}_i$ is a positive scalar, representing the squared Euclidean length of the eigenvector \mathbf{q}_i ; that is, $\mathbf{q}_i^H \mathbf{q}_i > 0$. We may therefore divide both sides of Eq. (4.14) by $\mathbf{q}_i^H \mathbf{q}_i$ and so express the i th eigenvalue λ_i as the ratio

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i}, \quad i = 1, 2, \dots, M \quad (4.15)$$

Since the correlation matrix \mathbf{R} is always nonnegative definite, the Hermitian form $\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i$ in the numerator of this ratio is always real and nonnegative; that is $\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i \geq 0$. Therefore, it follows from Eq. (4.15) that $\lambda_i \geq 0$ for all i . That is, all the eigenvalues of the correlation matrix \mathbf{R} are always real and nonnegative.

The correlation matrix \mathbf{R} is positive definite, except in noise-free sinusoidal and noise-free array signal-processing problems; and so we have $\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i > 0$ and, correspondingly, $\lambda_i > 0$ for all i . That is, the eigenvalues of the correlation matrix \mathbf{R} are almost always real and positive.

The ratio of the Hermitian form $\mathbf{q}_i^H \mathbf{R}\mathbf{q}_i$ to the inner product $\mathbf{q}_i^H \mathbf{q}_i$ on the right-hand side of Eq. (4.15) is called the *Rayleigh quotient* of the vector \mathbf{q}_i . We may thus state that an eigenvalue of the correlation matrix equals the Rayleigh quotient of the corresponding eigenvector.

Property 4. *Let $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ be the eigenvectors corresponding to the distinct eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ of the M -by- M correlation matrix \mathbf{R} , respectively. Then the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ are orthogonal to each other.*

Let \mathbf{q}_i and \mathbf{q}_j denote any two eigenvectors of the correlation matrix \mathbf{R} . We say that these two eigenvectors are *orthogonal* to each other if

$$\mathbf{q}_i^H \mathbf{q}_j = 0, \quad i \neq j \quad (4.16)$$

Using Eq. (4.1) we may express the conditions on the eigenvectors \mathbf{q}_i and \mathbf{q}_j as follows, respectively,

$$\mathbf{R}\mathbf{q}_i = \lambda_i\mathbf{q}_i \quad (4.17)$$

and

$$\mathbf{R}\mathbf{q}_j = \lambda_j \mathbf{q}_j \quad (4.18)$$

Premultiplying both sides of Eq. (4.17) by the Hermitian-transposed vector \mathbf{q}_j^H , we get

$$\mathbf{q}_j^H \mathbf{R} \mathbf{q}_i = \lambda_i \mathbf{q}_j^H \mathbf{q}_i \quad (4.19)$$

Since the correlation matrix \mathbf{R} is Hermitian, we have $\mathbf{R}^H = \mathbf{R}$. Also, from Property 3 we know that the eigenvalue λ_j is real for all j . Hence, taking the Hermitian transpose of both sides of Eq. (4.18) and using these two properties, we get

$$\mathbf{q}_j^H \mathbf{R} = \lambda_j \mathbf{q}_j^H \quad (4.20)$$

Postmultiplying both sides of Eq. (4.20) by the vector \mathbf{q}_i ,

$$\mathbf{q}_j^H \mathbf{R} \mathbf{q}_i = \lambda_j \mathbf{q}_j^H \mathbf{q}_i \quad (4.21)$$

Subtracting Eq.(4.21) from (4.19), we get

$$(\lambda_i - \lambda_j) \mathbf{q}_j^H \mathbf{q}_i = 0 \quad (4.22)$$

Since the eigenvalues of the correlation matrix \mathbf{R} are assumed to be distinct, we have $\lambda_i \neq \lambda_j$. Accordingly, the condition of Eq. (4.22) holds if and only if

$$\mathbf{q}_j^H \mathbf{q}_i = 0, \quad i \neq j \quad (4.23)$$

which is the desired result. That is, the eigenvectors \mathbf{q}_i and \mathbf{q}_j are *orthogonal* to each other for $i \neq j$.

Property 5: Unitary Similarity Transformation. Let $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ be the eigenvectors corresponding to the distinct eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ of the M -by- M correlation matrix \mathbf{R} , respectively. Define the M -by- M matrix

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M]$$

where

$$\mathbf{q}_i^H \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Define the M -by- M diagonal matrix

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$$

Then the original matrix \mathbf{R} may be diagonalized as follows:

$$\mathbf{Q}^H \mathbf{R} \mathbf{Q} = \Lambda$$

The condition that $\mathbf{q}_i^H \mathbf{q}_i = 1$ for $i = 1, 2, \dots, M$ requires that each eigenvector be *normalized* to have a *length* of 1. The *squared length* or *squared norm* of a vector \mathbf{q}_i is defined as the inner product $\mathbf{q}_i^H \mathbf{q}_i$. The orthogonality condition that $\mathbf{q}_i^H \mathbf{q}_j = 0$, for $i \neq j$, follows from Property 4. When both of these conditions are simultaneously satisfied; that is,

$$\mathbf{q}_i^H \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (4.24)$$

we say the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ form an *orthonormal* set. By definition, the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ satisfy the equations [see Eq. (4.1)]

$$\mathbf{R}\mathbf{q}_i = \lambda_i \mathbf{q}_i, \quad i = 1, 2, \dots, M \quad (4.25)$$

The M -by- M matrix \mathbf{Q} has as its columns the orthonormal set of eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$; that is,

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M] \quad (4.26)$$

The M -by- M diagonal matrix Λ has the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ for the elements of its main diagonal:

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M) \quad (4.27)$$

Accordingly, we may rewrite the set of M equations (4.25) as a single matrix equation:

$$\mathbf{R}\mathbf{Q} = \mathbf{Q}\Lambda \quad (4.28)$$

Owing to the orthonormal nature of the eigenvectors, as defined in Eq. (4.24), we find that

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

Equivalently, we may write

$$\mathbf{Q}^{-1} = \mathbf{Q}^H \quad (4.29)$$

That is, the matrix \mathbf{Q} is nonsingular with an inverse \mathbf{Q}^{-1} equal to the Hermitian transpose of \mathbf{Q} . A matrix that has this property is called a *unitary matrix*.

Thus, premultiplying both sides of Eq. (4.28) by the Hermitian-transposed matrix \mathbf{Q}^H and using the property of Eq. (4.29), we get the desired result:

$$\mathbf{Q}^H \mathbf{R} \mathbf{Q} = \Lambda \quad (4.30)$$

This transformation is called the *unitary similarity transformation*.

We have thus proved an important result. The correlation matrix \mathbf{R} may be *diagonalized* by a unitary similarity transformation. Furthermore, the matrix \mathbf{Q} that is used to diagonalize \mathbf{R} has as its columns an orthonormal set of eigenvectors for \mathbf{R} . The resulting diagonal matrix Λ has as its diagonal elements the eigenvalues of \mathbf{R} .

By postmultiplying both sides of Eq. (4.28) by the inverse matrix \mathbf{Q}^{-1} and then using the property of Eq. (4.29), we may also write

$$\begin{aligned} \mathbf{R} &= \mathbf{Q}\Lambda\mathbf{Q}^H \\ &= \sum_{i=1}^M \lambda_i \mathbf{q}_i \mathbf{q}_i^H \end{aligned} \quad (4.31)$$

where M is the dimension of matrix \mathbf{R} . Let the projection \mathbf{P}_i denote the outer product $\mathbf{q}_i\mathbf{q}_i^H$. Then, it is a straightforward matter to show that

$$\mathbf{P}_i = \mathbf{P}_i^2 = \mathbf{P}_i^H$$

which, in effect, means that $\mathbf{P}_i = \mathbf{q}_i\mathbf{q}_i^H$ is a *rank-one projection*. Thus, Eq. (4.31) states that the correlation matrix of a wide-sense stationary process equals the linear combination of all such rank-one projections, with each projection being weighted by the respective eigenvalue. This result is known as *Mercer's theorem*. It is also referred to as the *spectral theorem*.

Property 6. *Let $\lambda_1, \lambda_2, \dots, \lambda_M$ be the eigenvalues of the M -by- M correlation matrix \mathbf{R} . Then the sum of these eigenvalues equals the trace of matrix \mathbf{R} .*

The *trace* of a square matrix is defined as the sum of the diagonal elements of the matrix. Taking the trace of both sides of Eq. (4.30), we may write

$$\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\Lambda] \quad (4.32)$$

The diagonal matrix Λ has as its diagonal elements the eigenvalues of \mathbf{R} . Hence, we have

$$\text{tr}[\Lambda] = \sum_{i=1}^M \lambda_i \quad (4.33)$$

Using a rule in matrix algebra, we may write¹

$$\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\mathbf{R} \mathbf{Q} \mathbf{Q}^H]$$

However, $\mathbf{Q}\mathbf{Q}^H$ equals the identity matrix \mathbf{I} . Hence we have

$$\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\mathbf{R}]$$

Accordingly, we may rewrite Eq. (4.32) as

$$\text{tr}[\mathbf{R}] = \sum_{i=1}^M \lambda_i \quad (4.34)$$

We have thus shown that the trace of the correlation matrix \mathbf{R} equals the sum of its eigenvalues. Although in proving this result we used a property that requires the matrix \mathbf{R} to be Hermitian with distinct eigenvalues, nevertheless, the result applies to any square matrix.

Property 7. *The correlation matrix \mathbf{R} is ill conditioned if the ratio of the largest eigenvalue to the smallest eigenvalue of \mathbf{R} is large.*

To appreciate the impact of Property 7, it is important that we recognize the fact that the development of an algorithm for the effective solution of a signal processing problem

¹ This result follows from the following rule in matrix algebra. Let \mathbf{A} be an M -by- N matrix and \mathbf{B} be an N -by- M matrix. The trace of the matrix product \mathbf{AB} equals the trace of \mathbf{BA} .

and the understanding of associated *perturbation theory* go hand-in-hand (Van Loan, 1989). We may illustrate the synergism between these two fields by considering the following linear system of equations:

$$\mathbf{Aw} = \mathbf{d}$$

where the matrix \mathbf{A} and the vector \mathbf{d} are data-dependent quantities, and \mathbf{w} is a coefficient vector characterizing a linear FIR filter of interest. An elementary formulation of perturbation theory tells us that if the matrix \mathbf{A} and vector \mathbf{d} are perturbed by small amounts $\delta\mathbf{A}$ and $\delta\mathbf{d}$, respectively, and if $\|\delta\mathbf{A}\|/\|\mathbf{A}\|$ and $\|\delta\mathbf{d}\|/\|\mathbf{d}\|$ are both on the order of some ϵ with $\epsilon \ll 1$, we then have (Golub and Van Loan, 1989)

$$\frac{\|\delta\mathbf{w}\|}{\|\mathbf{w}\|} \leq \epsilon \chi(\mathbf{A})$$

where $\delta\mathbf{w}$ is the change produced in \mathbf{w} , and $\chi(\mathbf{A})$ is the *condition number* of matrix \mathbf{A} with respect to inversion. The condition number is so called because it describes the ill condition or bad behavior of matrix \mathbf{A} quantitatively. In particular, it is defined as follows (Wilkinson, 1963; Strang, 1980; Golub and Van Loan, 1989):

$$\chi(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \quad (4.35)$$

where $\|\mathbf{A}\|$ is a *norm* of matrix \mathbf{A} , and $\|\mathbf{A}^{-1}\|$ is the corresponding norm of the inverse matrix \mathbf{A}^{-1} . The norm of a matrix is a number assigned to the matrix that is in some sense a measure of the magnitude of the matrix. We find it natural to require that the norm of a matrix satisfy the following conditions:

1. $\|\mathbf{A}\| \geq 0$, $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = \mathbf{0}$
2. $\|c\mathbf{A}\| = |c| \|\mathbf{A}\|$, where c is any real number and $|c|$ is its magnitude
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$

Condition 3 is the *triangle inequality*, and condition 4 is the *mutual consistency*. There are several ways of defining the norm $\|\mathbf{A}\|$, which satisfy the preceding conditions (Ralston, 1965). For our present discussion, however, we find it convenient to use the *spectral norm*² defined as the square root of the largest eigenvalue of the matrix product $\mathbf{A}^H \mathbf{A}$, where \mathbf{A}^H is the Hermitian transpose of \mathbf{A} ; that is,

$$\|\mathbf{A}\|_s = (\text{largest eigenvalue of } \mathbf{A}^H \mathbf{A})^{1/2} \quad (4.36)$$

² Another matrix norm of interest is the *Frobenius norm*, $\|\mathbf{A}\|_F$, defined by (Stewart, 1973):

$$\|\mathbf{A}\|_F = \sqrt{\sum_{j=1}^M \sum_{i=1}^N |a_{ij}|^2}$$

where M and N are the dimensions of matrix \mathbf{A} , and a_{ij} is its ij th element.

Since for any matrix \mathbf{A} the product $\mathbf{A}^H \mathbf{A}$ is always Hermitian and nonnegative definite, it follows that the eigenvalues of $\mathbf{A}^H \mathbf{A}$ are all real and nonnegative, as required. Moreover, from Eq. (4.15) we note that an eigenvalue of $\mathbf{A}^H \mathbf{A}$ equals the Rayleigh coefficient of the corresponding eigenvector. Squaring both sides of Eq. (4.36) and using this property, we may therefore write³

$$\begin{aligned}\|\mathbf{A}\|_s^2 &= \max \frac{\mathbf{x}^H \mathbf{A}^H \mathbf{A} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \\ &= \max \frac{\|\mathbf{Ax}\|^2}{\|\mathbf{x}\|^2}\end{aligned}$$

where $\|\mathbf{x}\|^2$ is the inner product of vector \mathbf{x} with itself, and likewise for $\|\mathbf{Ax}\|^2$. We refer to $\|\mathbf{x}\|$ as the *Euclidean norm* or *length* of vector \mathbf{x} . We may thus express the spectral norm of matrix \mathbf{A} in the equivalent form

$$\|\mathbf{A}\|_s = \max \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} \quad (4.37)$$

According to this relation, the spectral norm of \mathbf{A} measures the largest amount by which any vector (eigenvector or not) is amplified by matrix multiplication, and the vector that is amplified the most is the eigenvector that corresponds to the largest eigenvalue of $\mathbf{A}^H \mathbf{A}$ (Strang, 1980).

Consider now the application of the definition in Eq. (4.36) to the correlation matrix \mathbf{R} . Since \mathbf{R} is Hermitian, we have $\mathbf{R}^H = \mathbf{R}$. Hence, from Property 1 we deduce that if λ_{\max} is the largest eigenvalue of \mathbf{R} , the largest eigenvalue of $\mathbf{R}^H \mathbf{R}$ equals λ_{\max}^2 . Accordingly, the spectral norm of the correlation matrix \mathbf{R} is

$$\|\mathbf{R}\|_s = \lambda_{\max} \quad (4.38)$$

Similarly, we may show that the spectral norm of \mathbf{R}^{-1} , the inverse of the correlation matrix, is

$$\|\mathbf{R}^{-1}\|_s = \frac{1}{\lambda_{\min}} \quad (4.39)$$

where λ_{\min} is the smallest eigenvalue of \mathbf{R} . Thus, by adopting the spectral norm as the basis of the condition number, we have shown that the condition number of the correlation matrix \mathbf{R} equals

$$\chi(\mathbf{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (4.40)$$

This ratio is commonly referred to as the *eigenvalue spread* or the *eigenvalue ratio* of the correlation matrix. Note that we always have $\chi(\mathbf{R}) \geq 1$.

³ Note that the vector \mathbf{x} is one of the eigenvectors. Hence, at this stage, we can only say that $\|\mathbf{A}\|_s^2$ is the maximum Rayleigh quotient of the eigenvectors. However, this may be extended to any vector after the minimax theorem is proved; see Property 9.

Suppose that the correlation matrix \mathbf{R} is *normalized* so that the magnitude of the largest element, $r(0)$, equals 1. Then, if the condition number or eigenvalue spread of the correlation matrix \mathbf{R} is large, we find that the inverse matrix \mathbf{R}^{-1} contains some very large elements. This behavior may cause trouble in solving a system of equations involving \mathbf{R}^{-1} . In such a case, we say that the correlation matrix \mathbf{R} is *ill conditioned*, hence the justification of Property 7.

Property 8. *The eigenvalues of the correlation matrix of a discrete-time stochastic process are bounded by the minimum and maximum values of the power spectral density of the process.*

Let λ_i and \mathbf{q}_i , $i = 1, 2, \dots, M$, denote the eigenvalues of the M -by- M correlation matrix \mathbf{R} of a discrete-time stochastic process $u(n)$ and their associated eigenvectors, respectively. From Eq. (4.15), we have

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i}, \quad i = 1, 2, \dots, M \quad (4.41)$$

The Hermitian form in the numerator may be expressed in its expanded form as follows

$$\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i = \sum_{k=1}^M \sum_{l=1}^M q_{ik}^* r(l-k) q_{il} \quad (4.42)$$

where q_{ik}^* is the k th element of the row vector \mathbf{q}_i^H , $r(l-k)$ is the kl th element of the matrix \mathbf{R} , and q_{il} is the l th element of the column vector \mathbf{q}_i . Using the Einstein–Wiener–Khintchine relation of Eq. (3.16), we may write

$$r(l-k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(l-k)} d\omega \quad (4.43)$$

where $S(\omega)$ is the power spectral density of the process $u(n)$. Hence, we may rewrite Eq. (4.42) as

$$\begin{aligned} \mathbf{q}_i^H \mathbf{R} \mathbf{q}_i &= \frac{1}{2\pi} \sum_{k=1}^M \sum_{l=1}^M q_{ik}^* q_{il} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(l-k)} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega S(\omega) \sum_{k=1}^M q_{ik}^* e^{-j\omega k} \sum_{l=1}^M q_{il} e^{j\omega l} \end{aligned} \quad (4.44)$$

Let the discrete-time Fourier transform of the sequence $q_{i1}^*, q_{i2}^*, \dots, q_{iM}^*$ be denoted by

$$Q_i(e^{j\omega}) = \sum_{k=1}^M q_{ik}^* e^{-j\omega k} \quad (4.45)$$

Therefore, using Eq. (4.45) in (4.44), we get

$$\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 S(\omega) d\omega \quad (4.46)$$

Similarly, we may show that

$$\mathbf{q}_i^H \mathbf{q}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 d\omega \quad (4.47)$$

Accordingly, we may use Eq. (4.15) to redefine the eigenvalue λ_i of the correlation matrix \mathbf{R} in terms of the associated power spectral density as

$$\lambda_i = \frac{\int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 S(\omega) d\omega}{\int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 d\omega} \quad (4.48)$$

Let S_{\min} and S_{\max} denote the absolute minimum and maximum values of the power spectral density $S(\omega)$, respectively. Then it follows that

$$\int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 S(\omega) d\omega \geq S_{\min} \int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 d\omega \quad (4.49)$$

and

$$\int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 S(\omega) d\omega \leq S_{\max} \int_{-\pi}^{\pi} |Q_i(e^{j\omega})|^2 d\omega \quad (4.50)$$

Hence, we deduce that the eigenvalues λ_i are bounded by the maximum and minimum values of the associated power spectral density as follows:

$$S_{\min} \leq \lambda_i \leq S_{\max}, \quad i = 1, 2, \dots, M \quad (4.51)$$

Correspondingly, the eigenvalue spread $\chi(\mathbf{R})$ is bounded as

$$\chi(\mathbf{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{S_{\max}}{S_{\min}} \quad (4.52)$$

It is of interest to note that as the dimension M of the correlation matrix approaches infinity, the maximum eigenvalue λ_{\max} approaches S_{\max} , and the minimum eigenvalue λ_{\min} approaches S_{\min} . Accordingly, the eigenvalue spread $\chi(\mathbf{R})$ of the correlation matrix \mathbf{R} approaches the ratio S_{\max}/S_{\min} as the dimension M of the matrix \mathbf{R} approaches infinity.

Property 9. Minimax Theorem. *Let the M -by- M correlation matrix \mathbf{R} have eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ that are arranged in decreasing order as follows:*

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_M$$

The minimax theorem states that

$$\lambda_k = \min_{\dim(\mathcal{S})=k} \max_{\substack{\mathbf{x} \in \mathcal{S} \\ \mathbf{x} \neq \mathbf{0}}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}}, \quad k = 1, 2, \dots, M \quad (4.53)$$

where \mathcal{S} is a subspace of the vector space of all M -by-1 complex vectors, $\dim(\mathcal{S})$ denotes the dimension of subspace \mathcal{S} , and $\mathbf{x} \in \mathcal{S}$ signifies that the vector \mathbf{x} (assumed to be nonzero) varies over the subspace \mathcal{S} .

Let \mathbb{C}^M denote a complex vector space of dimension M . For the purpose of our present discussion, we define the *complex (linear) vector space* \mathbb{C}^M as the set of all complex vectors that can be expressed as a linear combination of M basis vectors. Specifically, we may write

$$\mathbb{C}^M = \{\mathbf{y}\} \quad (4.54)$$

where \mathbf{y} is any complex vector defined by

$$\mathbf{y} = \sum_{i=1}^M a_i \mathbf{q}_i \quad (4.55)$$

The \mathbf{q}_i are the basis vectors, and the a_i are scalars. For the basis vectors we may use any *orthonormal set* of vectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ that satisfy two requirements:

$$\mathbf{q}_i^H \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (4.56)$$

In other words, each basis vector is *normalized* to have a *Euclidean length* or *norm* of unity, and it is *orthogonal* to every other basis vector in the set. The *dimension* M of the complex vector space \mathbb{C}^M is the minimum number of basis vectors required to span the entire space.

The basis functions define the “coordinates” of a complex vector space. Any complex vector of compatible dimension may then be represented simply as a “point” in that space. Indeed, the idea of a complex vector space is a natural generalization of Euclidean geometry. Central to this idea is that of a *subspace*. We say that \mathcal{S} is a subspace of the complex vector space \mathbb{C}^M if it involves a *subset* of the M basis vectors that define \mathbb{C}^M . In other words, a subspace of dimension k is defined as the set of complex vectors that can be written as a linear combination of the basis vectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k$, as shown by

$$\mathbf{x} = \sum_{i=1}^k a_i \mathbf{q}_i \quad (4.57)$$

Obviously, we have $k \leq M$. Note, however, that the dimension of the vector \mathbf{x} is M .

These ideas are illustrated in the three-dimensional (real) vector space depicted in Fig. 4.1. The $\mathbf{q}_1, \mathbf{q}_2$ -plane represents a subspace \mathcal{S} of dimension 2. The representation of vector \mathbf{y} and that of vector \mathbf{x} (i.e., the part of \mathbf{y} lying in subspace \mathcal{S}) are indicated in Fig. 4.1.

Returning to the issue at hand, namely, a proof of the *minimax theorem* described in Eq. (4.53), we may proceed as follows. We first use the spectral theorem of Eq. (4.31) to decompose the M -by- M correlation matrix \mathbf{R} as

$$\mathbf{R} = \sum_{i=1}^M \lambda_i \mathbf{q}_i \mathbf{q}_i^H$$

where the λ_i are the eigenvalues of the correlation matrix \mathbf{R} and the \mathbf{q}_i are the associated eigenvectors. In view of the orthonormality conditions of Eq. (4.24) satisfied by the eigen-

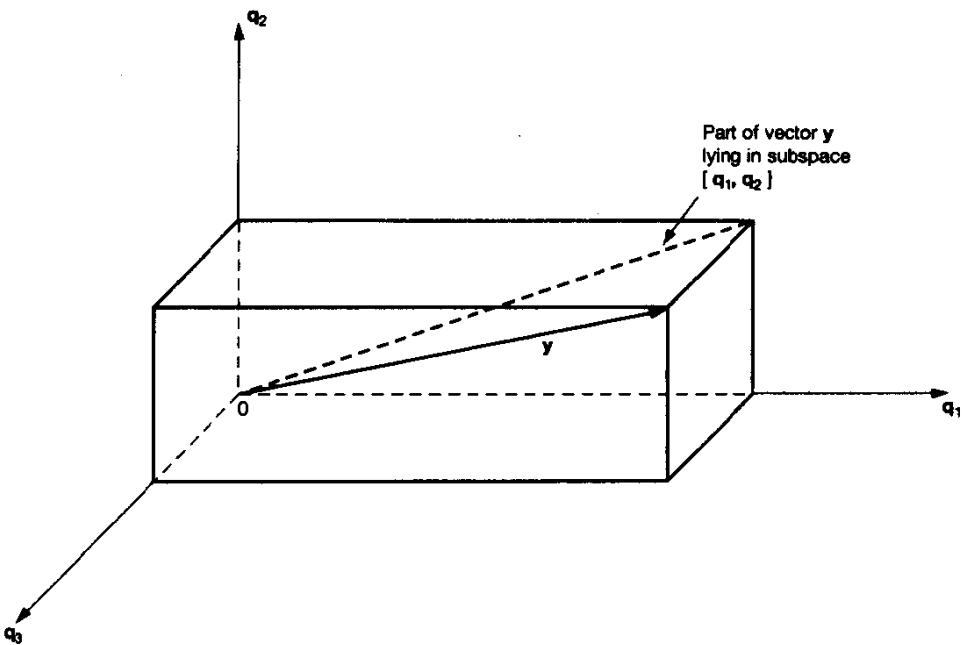


Figure 4.1 Illustrating the projection of a vector onto a subspace for a three-dimensional (real) vector space.

vectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$, we may adopt them as the M basis vectors of the complex vector space \mathbb{C}^M . Let an M -by-1 vector \mathbf{x} be constrained to lie in a subspace \mathcal{S} of dimension k , as defined in Eq. (4.57). Then, using Eq. (4.31), we may express the Rayleigh quotient of the vector \mathbf{x} as

$$\frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} = \frac{\sum_{i=1}^k a_i^2 \lambda_i}{\sum_{i=1}^k a_i^2} \quad (4.58)$$

Equation (4.58) states that the Rayleigh quotient of a vector \mathbf{x} lying in the subspace \mathcal{S} of dimension k (i.e., the subspace spanned by the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k$) is a *weighted mean* of the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$. Since, by assumption, we have $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_k$, it follows that for any subspace \mathcal{S} of dimension k ,

$$\max_{\substack{\mathbf{x} \in \mathcal{S} \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \leq \lambda_k$$

This result implies that

$$\min_{\dim(\mathcal{S})=k} \max_{\substack{\mathbf{x} \in \mathcal{S} \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \leq \lambda_k \quad (4.59)$$

We next prove that for any subspace \mathcal{S} of dimension k , spanned by the eigenvectors $\mathbf{q}_{i_1}, \mathbf{q}_{i_2}, \dots, \mathbf{q}_{i_k}$ where $\{i_1, i_2, \dots, i_k\}$ is a subset of $\{1, 2, \dots, M\}$, there exists at least

one nonzero vector \mathbf{x} common to \mathcal{S} and the subspace \mathcal{S}' spanned by the eigenvectors $\mathbf{q}_k, \mathbf{q}_{k+1}, \dots, \mathbf{q}_M$. To do so, we consider the system of M homogeneous equations:

$$\sum_{j=1}^k a_j \mathbf{q}_{i_j} = \sum_{i=k}^M b_i \mathbf{q}_i \quad . \quad (4.60)$$

where the $(M + 1)$ unknowns are made up as follows:

1. A total of k scalars, namely a_1, a_2, \dots, a_k , on the left-hand side.
2. A total of $M - k + 1$ scalars, namely b_k, b_{k+1}, \dots, b_M on the right-hand side.

Hence the system of equations (4.60) will always have a nontrivial solution. Moreover, we know from Property 2 that the eigenvectors $\mathbf{q}_{i_1}, \mathbf{q}_{i_2}, \dots, \mathbf{q}_{i_k}$ are *linearly independent*, as are the eigenvectors $\mathbf{q}_k, \mathbf{q}_{k+1}, \dots, \mathbf{q}_M$. It follows therefore that there is at least one *nonzero vector* $\mathbf{x} = \sum_{j=1}^k a_j \mathbf{q}_{i_j}$ that is *common* to the space of $\mathbf{q}_{i_1}, \mathbf{q}_{i_2}, \dots, \mathbf{q}_{i_k}$ and the space of $\mathbf{q}_k, \mathbf{q}_{k+1}, \dots, \mathbf{q}_M$. Thus, using Eqs. (4.60), (4.57), and (4.31), we may also express the Rayleigh quotient of the vector \mathbf{x} as a weighted mean of the eigenvalues $\lambda_k, \lambda_{k+1}, \dots, \lambda_M$, as shown by

$$\frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} = \frac{\sum_{i=k}^M b_i^2 \lambda_i}{\sum_{i=k}^M b_i^2} \quad (4.61)$$

Since, by assumption, we have $\lambda_k \leq \lambda_{k+1} \leq \dots \leq \lambda_M$, and since \mathbf{x} is also a vector in the subspace \mathcal{S} , we may write

$$\max_{\substack{\mathbf{x} \in \mathcal{S} \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \geq \lambda_k$$

Therefore,

$$\min_{\dim(\mathcal{S})=k} \max_{\substack{\mathbf{x} \in \mathcal{S} \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \geq \lambda_k \quad (4.62)$$

because \mathcal{S} is an arbitrary subspace of dimension k .

All that remains for us to do is to combine the results of Eqs. (4.59) and (4.62), and the minimax theorem of Eq. (4.53) describing Property 9 follows immediately.

From Property 9, we may make two important observations:

1. The minimax theorem as stated in Eq. (4.53) does not require any special knowledge of the eigenstructure (i.e., eigenvalues and eigenvectors) of the correlation matrix \mathbf{R} . Indeed, it may be adopted as the basis for defining the eigenvalues λ_k for $k = 1, 2, \dots, M$.
2. The minimax theorem points to a unique two-fold feature of the eigenstructure of the correlation matrix: (a) the eigenvectors represent the particular basis for an M -dimensional space that is most efficient in the energy sense, and (b) the eigen-

values are certain energies of the M -by-1 input (observation) vector $\mathbf{u}(n)$. This issue is pursued in greater depth under Property 10.

Another noteworthy point is that Eq. (4.53) may also be formulated in the following alternative but equivalent form:

$$\lambda_k = \max_{\dim(\mathcal{S}')=M-k+1} \min_{\substack{\mathbf{x} \in \mathcal{S}' \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \quad (4.63)$$

Equation (4.63) is referred to as the *maximin theorem*.

From Eqs. (4.53) and (4.63) we may readily deduce the following two special cases:

1. For $k = M$, the subspace \mathcal{S} occupies the complex vector space \mathbb{C}^M entirely. Under this condition, Eq. (4.53) reduces to

$$\lambda_M = \max_{\substack{\mathbf{x} \in \mathbb{C}^M \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \quad (4.64)$$

where λ_M is the *largest eigenvalue* of the correlation matrix \mathbf{R} .

2. For $k = 1$, the subspace \mathcal{S}' occupies the complex vector space \mathbb{C}^M entirely. Under this condition, Eq. (4.63) reduces to

$$\lambda_1 = \min_{\substack{\mathbf{x} \in \mathbb{C}^M \\ \mathbf{x} \neq 0}} \frac{\mathbf{x}^H \mathbf{R} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \quad (4.65)$$

where λ_1 is the *smallest eigenvalue* of the correlation matrix \mathbf{R} .

Property 10. Karhunen–Loève expansion. Let the M -by-1 vector $\mathbf{u}(n)$ denote a data sequence drawn from a wide-sense stationary process of zero mean and correlation matrix \mathbf{R} . Let $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ be eigenvectors associated with the M eigenvalues of the matrix \mathbf{R} . The vector $\mathbf{u}(n)$ may be expanded as a linear combination of these eigenvectors as follows:

$$\mathbf{u}(n) = \sum_{i=1}^M c_i(n) \mathbf{q}_i \quad (4.66)$$

The coefficients of the expansion are zero-mean, uncorrelated random variables defined by the inner product

$$c_i(n) = \mathbf{q}_i^H \mathbf{u}(n), \quad i = 1, 2, \dots, M \quad (4.67)$$

The representation of the random vector $\mathbf{u}(n)$ described by Eqs. (4.66) and (4.67) is the discrete-time version of the *Karhunen–Loève expansion*. In particular, Eq. (4.67) is the “analysis” part of the expansion in that it defines the $c_i(n)$ in terms of the input vector $\mathbf{u}(n)$. On the other hand, Eq. (4.66) is the “synthesis” part of the expansion in that it reconstructs

the original input vector $\mathbf{u}(n)$ from the $c_i(n)$. Given the expansion of Eq. (4.66), the definition of $c_i(n)$ in Eq. (4.67) follows directly from the fact that the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ form an orthonormal set, assuming they are all normalized to have unit length. Conversely, this same property may be used to derive Eq. (4.66), given (4.67).

The coefficients of the expansion are random variables characterized as follows:

$$E[c_i(n)] = 0, \quad i = 1, 2, \dots, M \quad (4.68)$$

and

$$E[c_i(n)c_j^*(n)] = \begin{cases} \lambda_i & i = j \\ 0, & i \neq j \end{cases} \quad (4.69)$$

Equation (4.68) states that all the coefficients of the expansion have zero mean; this follows directly from (Eq. 4.67) and the fact the random vector $\mathbf{u}(n)$ is itself assumed to have zero mean. Equation (4.69) states that the coefficients of the expansion are uncorrelated, and that each one of them has a mean-square value equal to the respective eigenvalue. This second equation is readily obtained by using the expansion of Eq. (4.66) in the definition of the correlation matrix \mathbf{R} as the expectation of the outer product $\mathbf{u}(n)\mathbf{u}^H(n)$, and then invoking the unitary similarity transformation (i.e., Property 5).

For a physical interpretation of the Karhunen–Loëve expansion, we may view the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ as the coordinates of an M -dimensional space, and thus represent the random vector $\mathbf{u}(n)$ by the set of its projections $c_1(n), c_2(n), \dots, c_M(n)$ onto these axes, respectively. Moreover, we deduce from Eq. (4.66) that

$$\sum_{i=1}^M |c_i(n)|^2 = \|\mathbf{u}(n)\|^2 \quad (4.70)$$

where $\|\mathbf{u}(n)\|$ is the Euclidean norm of $\mathbf{u}(n)$. That is to say, the coefficient $c_i(n)$ has an energy equal to that of the observation vector $\mathbf{u}(n)$ measured along the i th coordinate. Naturally, this energy is a random variable whose mean value equals the i th eigenvalue, as shown by

$$E[|c_i(n)|^2] = \lambda_i, \quad i = 1, 2, \dots, M \quad (4.71)$$

This result follows directly from Eqs. (4.67) and (4.69).

4.3 LOW-RANK MODELING

A key problem in statistical signal processing is that of *feature selection*, which refers to a process whereby a *data space* is transformed into a *feature space* that, in theory, has exactly the same dimension as the original data space. However, it would be desirable to design the transformation in such a way that the data vector can be represented by a reduced number of “effective” features and yet retain most of the intrinsic information content of the input data. In other words, the data vector undergoes a *dimensionality reduction*.

To be specific, suppose we have an M -dimensional data vector $\mathbf{u}(n)$ representing a particular realization of a wide-sense stationary process. We would like to transmit this vector over a noisy channel using a new set of p distinct numbers, where $p < M$. Basically, this is a feature-selection problem, which may be solved using the Karhunen–Loëve expansion, as described next.

According to Eq. (4.66), the data vector $\mathbf{u}(n)$ may be expanded as a linear combination of the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ associated with the respective eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ of the correlation matrix \mathbf{R} of $\mathbf{u}(n)$. It is assumed that the eigenvalues are all distinct and arranged in decreasing order, as shown by

$$\lambda_1 > \lambda_2 > \dots > \lambda_i > \dots > \lambda_M \quad (4.72)$$

The data representation described in Eq. (4.66) using all the eigenvalues of matrix \mathbf{R} is *exact* in the sense that it involves *no* loss of information. Suppose, however, that we have *prior knowledge* that the $(M - p)$ eigenvalues $\lambda_{p+1}, \dots, \lambda_M$ at the tail end of Eq. (4.72) are all very small. We may take advantage of this prior knowledge by retaining the p largest eigenvalues of matrix \mathbf{R} and thereby truncating the Karhunen–Loëve expansion of Eq. (4.66) at the term $i = p$. Accordingly, we may define an *approximate reconstruction* of the data vector $\mathbf{u}(n)$ as follows:

$$\hat{\mathbf{u}}(n) = \sum_{i=1}^p c_i(n) \mathbf{q}_i, \quad p < M \quad (4.73)$$

The vector $\hat{\mathbf{u}}(n)$ has rank p , which is lower than the rank M of the original data vector $\mathbf{u}(n)$. For this reason, the data model defined by Eq. (4.73) is referred to as a *low-rank model*. The important point to note here is that we may reconstruct the approximation $\hat{\mathbf{u}}(n)$ by using the set of p numbers: $\{c_i(n); i = 1, 2, \dots, p\}$. The $c_i(n)$ are themselves defined in terms of the data vector $\mathbf{u}(n)$ by Eq. (4.67). In other words, the new vector $\mathbf{c}(n)$, having $c_1(n), c_2(n), \dots, c_p(n)$ as elements, may be viewed as the reduced-rank *representation* for the original data vector $\mathbf{u}(n)$.

Figure 4.2 depicts the essence of the feature selection procedure described above. We start with an M -dimensional *data space*, in which a particular point defines the location of the data vector $\mathbf{u}(n)$. This point is transformed, via Eq. (4.67), into a new point in a *feature space* of dimension p that is lower than M . The transformation described here is sometimes referred to as a *subspace decomposition*.

Clearly, in using Eq. (4.73) to reconstruct the data vector $\mathbf{u}(n)$, an error is incurred due to the fact that $\hat{\mathbf{u}}(n)$ is of lower rank than $\mathbf{u}(n)$. The *reconstruction error vector* is defined by

$$\mathbf{e}(n) = \mathbf{u}(n) - \hat{\mathbf{u}}(n) \quad (4.74)$$

Hence, using Eqs. (4.66) and (4.73) in Eq. (4.74) yields

$$\mathbf{e}(n) = \sum_{i=p+1}^M c_i(n) \mathbf{q}_i \quad (4.75)$$

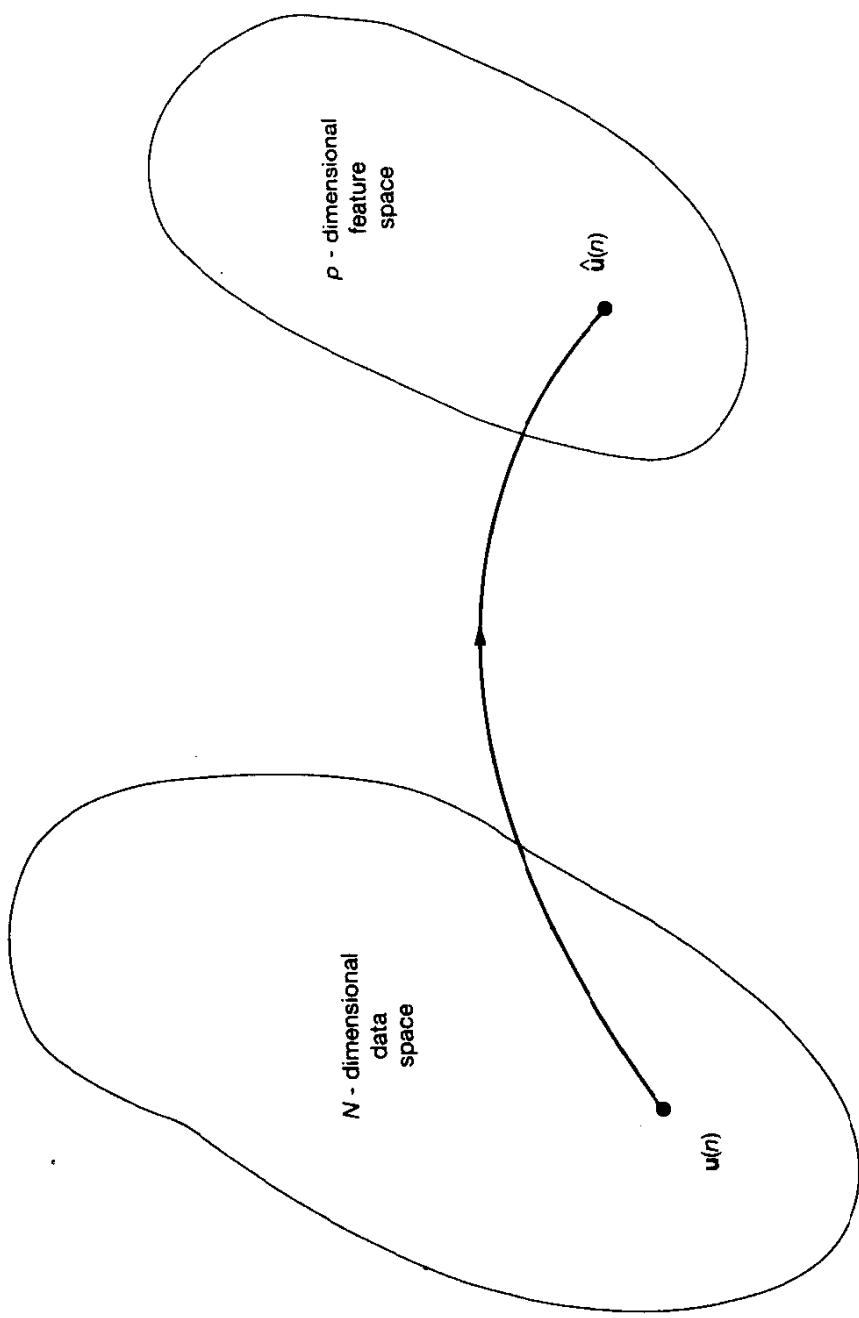


Figure 4.2 Illustrating the transformation involved in subspace decomposition.

The mean-square error is therefore

$$\begin{aligned}
 \epsilon &= E[\|\mathbf{e}(n)\|^2] \\
 &= E[\mathbf{e}^H(n)\mathbf{e}(n)] \\
 &= E\left[\sum_{i=p+1}^M \sum_{j=p+1}^M c_i^*(n)c_j(n)\mathbf{q}_i^H\mathbf{q}_j\right] \\
 &= \sum_{i=p+1}^M \sum_{j=p+1}^M E[c_i^*(n)c_j(n)]\mathbf{q}_i^H\mathbf{q}_j \\
 &= \sum_{i=p+1}^M \lambda_i
 \end{aligned} \tag{4.76}$$

which confirms that the data reconstruction defined by Eq. (4.73) is a good one, provided that the eigenvalues $\lambda_{p+1}, \dots, \lambda_M$ are all very small.

An Application of Low-rank Modeling

To appreciate the practical value of the low-rank model based on Eq. (4.73), consider the transmission of data vector $\mathbf{u}(n)$ over a *noisy communication channel*. In particular, the received signal is corrupted by *channel noise* $\mathbf{v}(n)$, which is modeled as additive white noise of zero mean.

Specifically, we have

$$E[\mathbf{u}(n)\mathbf{v}^H(n)] = \mathbf{0} \tag{4.77}$$

and

$$E[\mathbf{v}(n)\mathbf{v}^H(n)] = \sigma^2 \mathbf{I} \tag{4.78}$$

Equation (4.77) says that the noise vector $\mathbf{v}(n)$ is uncorrelated with the data vector $\mathbf{u}(n)$. Equation (4.78), with \mathbf{I} denoting the identity matrix, says that the elements of the noise vector are uncorrelated with each other and that each element has a variance of σ^2 .

In Fig. 4.3 we describe two methods for accomplishing the data transmission over the channel. One method is *direct*, and the other is *indirect*, as described next.

In the direct method depicted in Fig. 4.3(a), the received signal vector is given by

$$\mathbf{y}_{\text{direct}}(n) = \mathbf{u}(n) + \mathbf{v}(n) \tag{4.79}$$

The mean-square value of the *transmission error* is therefore

$$\begin{aligned}
 \epsilon_{\text{direct}} &= E[\|\mathbf{y}_{\text{direct}}(n) - \mathbf{u}(n)\|^2] \\
 &= E[\|\mathbf{v}(n)\|^2] \\
 &= E[\mathbf{v}^H(n)\mathbf{v}(n)]
 \end{aligned}$$

From Eq. (4.78) we see that each element $v_i(n)$, say, of the noise vector $\mathbf{v}(n)$ has variance σ^2 . We may therefore express ϵ_{direct} simply as

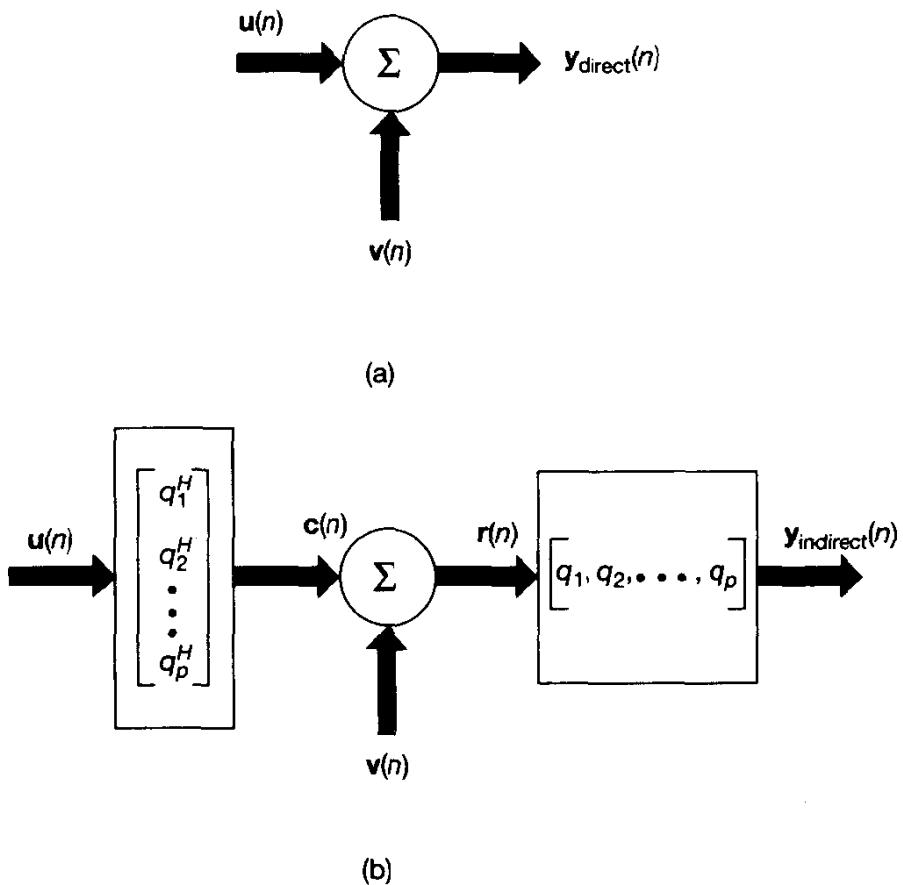


Figure 4.3 Data transmission using (a) direct method, and (b) indirect method inspired by low-rank modeling.

$$\begin{aligned}\epsilon_{\text{direct}} &= \sum_{i=1}^M E[|v_i(n)|^2] \\ &= M \sigma^2\end{aligned}\quad (4.80)$$

where M is the size of the noise vector $\mathbf{v}(n)$.

Consider next the indirect method depicted in Fig. 4.3(b), where the input vector $\mathbf{u}(n)$ is first applied to a *transmit filter bank*, whose individual tap-weight vectors are set equal to the Hermitian transpose of the eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p$ associated with the p largest eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_p$ of the correlation matrix \mathbf{R} of the input vector $\mathbf{u}(n)$. The resulting p -by-1 vector $\mathbf{c}(n)$, whose elements are made up of the inner products of $\mathbf{u}(n)$ with $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p$ in accordance with Eq. (4.67), constitutes the transmitted signal vector $\mathbf{c}(n)$ as shown by

$$\mathbf{c}(n) = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p]^H \mathbf{u}(n) \quad (4.81)$$

Correspondingly, the received signal vector is defined by

$$\mathbf{r}(n) = \mathbf{c}(n) + \mathbf{v}(n) \quad (4.82)$$

where the channel noise vector $v(n)$ is now of size p to be compatible with that of $c(n)$. To reconstruct the original data vector, the received signal vector $r(n)$ is applied to a *receive filter bank*, whose individual tap-weight vectors are defined by the eigenvectors q_1, q_2, \dots, q_p . The resulting output vector of the receiver in Fig. 4.3(b) is given by

$$\begin{aligned} y_{\text{indirect}}(n) &= [q_1, q_2, \dots, q_p]r(n) \\ &= [q_1, q_2, \dots, q_p]c(n) + [q_1, q_2, \dots, q_p]v(n) \end{aligned} \quad (4.83)$$

Hence, evaluating the mean-square value of the overall reconstruction error for the indirect method, we get (see Problem 20)

$$\begin{aligned} \epsilon_{\text{indirect}} &= E[\|y_{\text{indirect}}(n) - u(n)\|^2] \\ &= \sum_{i=p+1}^M \lambda_i + p\sigma^2 \end{aligned} \quad (4.84)$$

The first term of Eq. (4.84) is due to the low-rank modeling of the data vector $u(n)$ prior to transmission over the channel. The second term is due to the effect of channel noise.

Comparing Eq. (4.84) for the indirect method with Eq. (4.81) for the direct method, we readily see that the use of low-rank modeling offers an advantage, provided that we have

$$\sum_{i=p+1}^M \lambda_i < (M-p)\sigma^2 \quad (4.85)$$

This is an interesting result (Scharf and Tufts, 1987). It states that if the tail-end eigenvalues $\lambda_{p+1}, \dots, \lambda_M$ of the correlation matrix of the data vector $u(n)$ are all very small, the mean-square error produced by transmitting a low-rank approximation to the original data vector [as in Fig. 4.3(b)] is less than the mean-square error produced by transmitting the original data vector without any approximation [as in Fig. 4.3(a)].

The result described in Eq. (4.84) is particularly important in that it highlights the essence of what is commonly referred to as the “bias–variance tradeoff.” Specifically, a low-rank model is used for representing the data vector $u(n)$, thereby incurring a *bias*. Interestingly enough, this is done knowingly, in return for a reduction in *variance*, namely, the part of the mean-square error due to the additive noise vector $v(n)$. Indeed, the example described herein clearly illustrates the motivation for using a parsimonious (i.e., simpler) model that may not exactly match the underlying physics responsible for generating the data vector $u(n)$, hence the bias; but the model is less susceptible to noise, hence a reduction in variance.

4.4 EIGENFILTERS

A fundamental issue in communication theory is that of determining an optimum finite (duration) impulse response (FIR) filter, with the optimization criterion being that of maximizing the output signal-to-noise ratio. In this section we show that this filter optimization is linked to an eigenvalue problem.

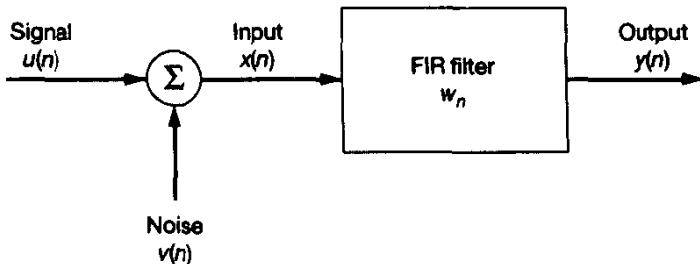


Figure 4.4 Linear filtering.

Consider a linear FIR filter whose impulse response is denoted by the sequence w_n . The sequence $x(n)$ applied to the filter input consists of a useful *signal* component $u(n)$ plus an additive *noise* component $v(n)$. The signal $u(n)$ is drawn from a wide-sense stationary stochastic process of zero mean and correlation matrix \mathbf{R} . The zero-mean noise $v(n)$ is white with a constant power spectral density determined by the variance σ^2 . It is assumed that the signal $u(n)$ and the noise $v(n)$ are uncorrelated; that is,

$$E[u(n)v^*(m)] = 0 \quad \text{for all } (n, m)$$

The filter output is denoted by $y(n)$. The situation described herein is depicted in Fig. 4.4.

Since the filter is linear, the principle of superposition applies. We may therefore consider the effects of signal and noise separately. Let P_o denote the average power of the signal component of the filter output $y(n)$. We may therefore show that (see Problem 9 of Chapter 2):

$$P_o = \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (4.86)$$

where the elements of the vector \mathbf{w} are the filter coefficients, and \mathbf{R} is the correlation matrix of the signal component $u(n)$ in the filter input.

Consider next the effect of noise acting alone. Let N_o denote the average power of the noise component in the filter output $y(n)$. This is a special case of Eq. (4.86), as shown by

$$N_o = \sigma^2 \mathbf{w}^H \mathbf{w} \quad (4.87)$$

where σ^2 is the variance of the white noise in the filter input.

Let $(\text{SNR})_o$ denote the *output signal-to-noise ratio*. Dividing Eq. (4.86) by (4.87), we may thus write

$$\begin{aligned}
 (\text{SNR})_o &= \frac{P_o}{N_o} \\
 &= \frac{\mathbf{w}^H \mathbf{R} \mathbf{w}}{\sigma^2 \mathbf{w}^H \mathbf{w}}
 \end{aligned} \quad (4.88)$$

The optimum problem may now be stated as follows:

Determine the coefficient vector \mathbf{w} of an FIR filter so as to maximize the output signal-to-noise ratio $(\text{SNR})_o$ subject to the constraint

$$\mathbf{w}^H \mathbf{w} = 1$$

Equation (4.88) shows that except for the scaling factor $1/\sigma^2$, the output signal-to-noise ratio $(\text{SNR})_o$ is equal to the Rayleigh quotient of the coefficient vector \mathbf{w} of the FIR filter. We see therefore that the optimum filtering problem, as stated herein, may be viewed as an eigenvalue problem. Indeed, the solution to the problem follows directly from the minimax theorem. Specifically, using the special form of the minimax theorem given in Eq. (4.64), we may state the following:

- The maximum value of the output signal-to-noise ratio is given by

$$(\text{SNR})_{o,\max} = \frac{\lambda_{\max}}{\sigma^2} \quad (4.89)$$

where λ_{\max} is the largest eigenvalue of the correlation matrix \mathbf{R} . Note that λ_{\max} and σ^2 have the same units but different physical interpretations.

- The coefficient vector of the optimum FIR filter that yields the maximum output signal-to-noise ratio of Eq. (4.89) is defined by

$$\mathbf{w}_o = \mathbf{q}_{\max} \quad (4.90)$$

where \mathbf{q}_{\max} is the eigenvector associated with the largest eigenvalue λ_{\max} of the correlation matrix \mathbf{R} . The correlation matrix \mathbf{R} belongs to the signal component $u(n)$ in the filter input.

An FIR filter whose impulse response has coefficients equal to the elements of an eigenvector is called an *eigenfilter* (Makhoul, 1981). Accordingly, we may state that *the maximum eigenfilter (i.e., the eigenfilter associated with the largest eigenvalue of the correlation matrix of the signal component in the filter input) is the optimum filter*. It is important to note that the optimum filter described in this way is uniquely characterized by an eigendecomposition of the correlation matrix of the signal component in the filter input. The power spectrum of the white noise at the filter input merely affects the maximum value of the output signal-to-noise ratio. In particular, we may proceed as follows:

1. An eigendecomposition of the correlation matrix \mathbf{R} is performed.
2. Only the largest eigenvalue λ_{\max} and the associated eigenvector \mathbf{q}_{\max} are retained.
3. The eigenvector \mathbf{q}_{\max} defines the impulse response of the optimum filter. The eigenvalue λ_{\max} , divided by the noise variance σ^2 , defines the maximum value of the output signal-to-noise ratio.

The optimum filter so characterized may be viewed as the "stochastic" counterpart of a matched filter. The optimum filter described herein maximizes the output signal-to-noise ratio for a *random signal* (i.e., a sample function of a discrete-time wide-sense stationary stochastic process) in additive white noise. A *matched filter*, on the other hand, maximizes the output signal-to-noise ratio for a *known signal* in additive white noise (North, 1963; Haykin, 1994).

4.5 EIGENVALUE COMPUTATIONS

The computation of the eigenvalues of a square matrix can, in general, be a complicated issue. Special and cooperative efforts by a group of experts between 1958 and 1970 resulted in the development of several *canned routines* that are widely available for matrix eigenvalue computations (Parlett, 1985). Special mention should be made of the following program libraries:

- MATLAB: a matrix-based numerical system for interactive computation, visualization, modeling, and algorithm development (Riddle, 1994)
- MATHEMATICA: an integrated mathematical system for numerical, symbolic, and graphical computation and visualization (Riddle, 1994)
- LINPACK: standard subroutine packages for computational linear algebra (Dongarra et al., 1979)
- LAPACK: a linear algebra library for single-address space machines
- ScaLAPACK: a linear algebra library for multiple-address space machines (Demmel, 1994)

The canned *eigen-routines* in these libraries are well documented and well tested.

The origin of almost all canned eigen-routines may be traced back to routines published in Volume II, *Linear Algebra*, of the *Handbook for Automatic Computation* co-edited by Wilkinson and Reinsch (1971). This reference is the bible of eigenvalue computations.

Another useful source of routines, written in the *C programming language*, is the book by Press et al. (1988); a companion book by the same authors, with routines written in FORTRAN and Pascal, is also available. The eigen-routines written in C can only handle real matrices. It is, however, a straightforward matter to extend the use of these eigen-routines to deal with Hermitian matrices, as shown next.

Let \mathbf{A} denote an M -by- M Hermitian matrix, written in terms of its real and imaginary parts as follows:

$$\mathbf{A} = \mathbf{A}_r + j\mathbf{A}_i \quad (4.91)$$

Correspondingly, let an associated M -by-1 eigenvector \mathbf{q} be written as

$$\mathbf{q} = \mathbf{q}_r + j\mathbf{q}_i \quad (4.92)$$

The M -by- M complex eigenvalue problem

$$(\mathbf{A}_r + j\mathbf{A}_i)(\mathbf{q}_r + j\mathbf{q}_i) = \lambda(\mathbf{q}_r + j\mathbf{q}_i) \quad (4.93)$$

may then be reformulated as the $2M$ -by- $2M$ real eigenvalue problem:

$$\begin{bmatrix} \mathbf{A}_r & -\mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_i \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_i \end{bmatrix} \quad (4.94)$$

where the eigenvalue λ is a real number. The Hermitian property

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

is equivalent to $\mathbf{A}_r^T = \mathbf{A}_r$ and $\mathbf{A}_i^T = -\mathbf{A}_i$. Accordingly, the $2M$ -by- $2M$ matrix in Eq. (4.94) is not only real but also symmetric. Note, however, that for a given eigenvalue λ , the vector

$$\begin{bmatrix} -\mathbf{q}_i \\ \mathbf{q}_r \end{bmatrix}$$

is also an eigenvector. This means that if $\lambda_1, \lambda_2, \dots, \lambda_M$ are the eigenvalues of the M -by- M Hermitian matrix \mathbf{A} , then the eigenvalues of the $2M$ -by- $2M$ symmetric matrix of Eq. (4.94) are $\lambda_1, \lambda_1, \lambda_2, \lambda_2, \dots, \lambda_M, \lambda_M$. We may therefore make two observations.

1. Each eigenvalue of the matrix in Eq. (4.94) has a multiplicity of 2.
2. The associated eigenvectors consist of pairs, each of the form $\mathbf{q}_r + j\mathbf{q}_i$ and $j(\mathbf{q}_r + j\mathbf{q}_i)$, differing merely by a rotation through 90° .

Thus, to solve the M -by- M complex eigenvalue problem of Eq. (4.93) with the aid of real eigen-routines, we choose one eigenvalue and eigenvector from each pair associated with the augmented $2M$ -by- $2M$ real eigenvalue problem of Eq. (4.94).

Strategies for Matrix Eigenvalue Computations

There are two different “strategies” behind practically all modern eigen-routines: *diagonalization* and *triangularization*. Since not all matrices can be diagonalized through a sequence of unitary similarity transformations, the diagonalization strategy applies only to Hermitian matrices such as a correlation matrix. On the other hand, the triangularization strategy is general in that it applies to any square matrix. These two strategies are described in the following sections.

Diagonalization. The idea behind this strategy is to nudge a Hermitian matrix \mathbf{A} toward a diagonal form by the repeated application of unitary similarity transformations, as described here:

$$\begin{aligned} \mathbf{A} &\rightarrow \mathbf{Q}_1^H \mathbf{A} \mathbf{Q}_1 \\ &\rightarrow \mathbf{Q}_2^H \mathbf{Q}_1^H \mathbf{A} \mathbf{Q}_1 \mathbf{Q}_2 \\ &\rightarrow \mathbf{Q}_3^H \mathbf{Q}_2^H \mathbf{Q}_1^H \mathbf{A} \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \end{aligned} \quad (4.95)$$

and so on. This sequence of unitary similarity transformation is, in theory, infinitely long. In practice, however, it is continued until we are close to a diagonal matrix. The elements of the diagonal matrix so obtained define the eigenvalues of the original Hermitian matrix \mathbf{A} . The associated eigenvectors are the column vectors of the accumulated sequence of transformations, as shown by

$$\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \dots \quad (4.96)$$

One method for implementing the diagonalization strategy of Eq. (4.95) is to use *Givens rotations*. This method is discussed in Chapter 12.

Triangularization. The idea behind this second strategy is to reduce a Hermitian matrix \mathbf{A} to a triangular form by a sequence of unitary similarity transformations. The resulting iterative procedure is called the *QL algorithm*.⁴ Suppose that we are given an M -by- M Hermitian matrix \mathbf{A}_n , where the subscript n refers to a particular step in the iterative procedure. Let the matrix \mathbf{A}_n be *factored* in the form

$$\mathbf{A}_n = \mathbf{Q}_n \mathbf{L}_n \quad (4.97)$$

where \mathbf{Q}_n is a *unitary matrix* and \mathbf{L}_n is a *lower triangular matrix* (i.e., the elements of the matrix \mathbf{L}_n located above the main diagonal are all zero). At step $n + 1$ in the iterative procedure, we use the known matrices \mathbf{Q}_n and \mathbf{L}_n to compute a new M -by- M matrix

$$\mathbf{A}_{n+1} = \mathbf{L}_n \mathbf{Q}_n \quad (4.98)$$

Note that the factorization in Eq. (4.98) is written in the opposite order to that in Eq. (4.97). Since \mathbf{Q}_n is a unitary matrix, we have $\mathbf{Q}_n^{-1} = \mathbf{Q}_n^H$, so we may rewrite Eq. (4.97) as

$$\begin{aligned} \mathbf{L}_n &= \mathbf{Q}_n^{-1} \mathbf{A}_n \\ &= \mathbf{Q}_n^H \mathbf{A}_n \end{aligned} \quad (4.99)$$

Therefore, substituting Eq. (4.99) into (4.98), we get

$$\mathbf{A}_{n+1} = \mathbf{Q}_n^H \mathbf{A}_n \mathbf{Q}_n \quad (4.100)$$

Equation (4.100) shows that the Hermitian matrix \mathbf{A}_{n+1} at iteration $n + 1$ is indeed unitarily related to the Hermitian matrix \mathbf{A}_n at iteration n .

The QL algorithm thus consists of a sequence of unitary similarity transformations, summarized by writing

$$\mathbf{A}_n = \mathbf{Q}_n \mathbf{L}_n$$

$$\mathbf{A}_{n+1} = \mathbf{L}_n \mathbf{Q}_n$$

⁴ The QL algorithm uses a lower triangular matrix. There is a companion algorithm, called the QR algorithm, which uses an upper triangular matrix. The QR algorithm is not to be confused with the QR-decomposition; the latter is discussed in Chapter 14.

where $n = 0, 1, 2, \dots$. The algorithm is *initialized* by setting

$$\mathbf{A}_0 = \mathbf{A}$$

where \mathbf{A} is the given M -by- M Hermitian matrix.

For general matrix \mathbf{A} , the following theorem is the basis of the QL algorithm⁵:

If matrix \mathbf{A} has eigenvalues of different absolute values, then the matrix \mathbf{A}_n approaches a lower triangular form as the number of iterations n approaches infinity.

The eigenvalues of the original matrix \mathbf{A} appear on the main diagonal of the lower triangular matrix resulting from the QL algorithm in increasing order of absolute value.

To implement the factorization in Eq. (4.98), we may use Givens rotations. Here again, however, we defer a discussion of the Givens rotation to Chapter 12. In that chapter we discuss computations for the singular value decomposition of a general matrix, which includes eigendecomposition as a special case.

4.6 SUMMARY AND DISCUSSION

In this chapter we studied the decomposition of the ensemble-averaged correlation matrix of a discrete-time wide-sense stationary stochastic process in terms of its eigenvalues and associated eigenvectors. Eigendecomposition provides an invaluable bridge between matrix algebra and stochastic processes, thereby placing it at the forefront of discrete-time linear filter theory.

Building on the properties of a discrete-time wide-sense stationary stochastic process described in Chapters 2 and 3, we established the following properties:

- The eigenvalues of the correlation matrix of the process are always nonnegative and bounded by the maximum and minimum values of the power spectral density of the process.
- The associated eigenvectors form an orthonormal set.

Another important result that we established is the Karhunen–Loëve expansion, according to which a data vector (drawn from a wide-sense stationary stochastic process) may be expanded as a linear combination of the eigenvectors pertaining to the correlation matrix of the process. This important result provides the theoretical basis for the design of a low-rank model of the data vector, which means that the dimensionality of the data vector may be reduced without sacrificing the intrinsic information content of the data.

⁵For a proof of this theorem, see Stoer and Bulirsch (1980). See also Stewart (1973), Golub and Van Loan (1989), and Press et al. (1992) for an improved version of the QL algorithm.

The final result established in the chapter is the notion of a maximum eigenfilter, defined by the eigenvector associated with the largest eigenvalue of the correlation matrix of a wide-sense stationary stochastic process. This filter optimizes the detection of a random signal, representing a particular realization of the process, embedded in a white-noise background.

PROBLEMS

1. The correlation matrix \mathbf{R} of a wide-sense stationary process $u(n)$ has the following values for its two eigenvalues:

$$\lambda_1 = 0.5$$

$$\lambda_2 = 1.5$$

- (a) Find the trace of matrix \mathbf{R} .
 - (b) Write an expression for the decomposition of matrix \mathbf{R} in terms of its two eigenvalues and associated eigenvectors. Comment on the uniqueness of this decomposition.
2. Show that the eigenvalues of a triangular matrix equal the diagonal elements of the matrix.
3. Consider the $2M$ -by- $2M$ real eigenvalue problem described in Eq. (4.94). Show that if $\mathbf{q}_r + j\mathbf{q}_i$ is an eigenvector of the matrix described herein, so is $\mathbf{q}_r - j\mathbf{q}_i$, with both eigenvectors being associated with the same eigenvalue.
4. Let $\lambda_1, \lambda_2, \dots, \lambda_M$ denote the eigenvalues of the correlation matrix of an observation vector $\mathbf{u}(n)$ taken from a stationary process of zero mean and variance σ_u^2 . Show that

$$\sum_{i=1}^M \lambda_i = M\sigma_u^2$$

5. An M -by- M correlation matrix \mathbf{R} is represented in terms of its eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ and their associated eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ as follows:

$$\mathbf{R} = \sum_{i=1}^M \lambda_i \mathbf{q}_i \mathbf{q}_i^H$$

- (a) Show that the corresponding representation for the *square root* of matrix \mathbf{R} is
- $$\mathbf{R}^{1/2} = \sum_{i=1}^M \lambda_i^{1/2} \mathbf{q}_i \mathbf{q}_i^H$$
- (b) By definition, we have $\mathbf{R} = \mathbf{R}^{1/2} \mathbf{R}^{1/2}$. Using this result, describe a procedure for computing the square root of a square matrix.
6. Consider a stationary process $u(n)$ whose M -by- M correlation matrix equals \mathbf{R} . Show that the determinant of the correlation matrix \mathbf{R} equals
- $$\det(\mathbf{R}) = \prod_{i=1}^M \lambda_i$$
7. (a) Show that the product of two unitary matrices is also a unitary matrix.
 (b) Show that the inverse of a unitary matrix is also a unitary matrix.

8. Let A be an M -by- M matrix. The *Schur decomposition theorem* states there exists a unitary matrix Z such that

$$Z^H A Z = T$$

where T is an upper triangular matrix. The theorem also states that:

- (i) The diagonal of matrix T is made up of the eigenvalues of the matrix A .
- (ii) If $Z = [z_1, z_2, \dots, z_M]$, then $\text{span}(z_1, z_2, \dots, z_k)$ is an invariant subspace associated with the eigenvalues $t_{11}, t_{22}, \dots, t_{kk}$ where $k \leq M$.
- (a) Apply the Schur decomposition to the correlation matrix R of a wide-stationary stochastic process. Hence, show that in this case the matrix T is a diagonal matrix.
- (b) What is the implication of the statement under (ii) in the context of the correlation matrix R ?

9. Consider the factorization

$$A_n - k_n I = Q_n L_n$$

where A_n is an M -by- M Hermitian matrix, I is the M -by- M identity matrix, Q_n is an M -by- M unitary matrix, L_n is an M -by- M lower triangular matrix, and k_n is a scalar. Define the matrix

$$A_{n+1} = L_n Q_n + k_n I$$

Hence, show that

$$A_{n+1} = Q_n^H A_n Q_n$$

10. In this problem we consider a *Fourier analyzer* for a single channel. The *Fourier basis* is described by

$$\mathbf{v}_i = \frac{1}{\sqrt{M}} [1, e^{j2\pi i/M}, e^{j4\pi i/M}, \dots, e^{j(M-1)2\pi i/M}]^T$$

where $i = 0, 1, \dots, M-1$. Let an arbitrary M -by-1 vector $\mathbf{u}(n)$ be expanded in terms of this orthonormal set as follows:

$$\mathbf{u}(n) = \sum_{i=0}^{M-1} c_i(n) \mathbf{v}_i$$

- (a) Evaluate the *Fourier coefficients* $c_0(n), c_1(n), \dots, c_{M-1}(n)$ in terms of the vector $\mathbf{u}(n)$.
- (b) Are the Fourier coefficients correlated? Justify your answer.
- (c) What does the expectation of $|c_i(n)|^2$ approximate?

- 11. Show that the condition number of matrix A is unchanged when this matrix is multiplied by a unitary matrix of compatible dimensions.
- 12. Consider an L -by- M matrix A . Show that the M -by- M matrix $A^H A$ and the L -by- L matrix $A A^H$ have the same nonzero eigenvalues.
- 13. A stochastic process $v(n)$ with a wide-band power spectrum is applied to a discrete-time linear filter whose amplitude response $|H(e^{j\omega})|$ is nonuniform. The maximum and minimum values of this response are denoted by H_{\max} and H_{\min} , respectively. Let $\chi(R)$ denote the eigenvalue spread of the correlation matrix R of the stochastic process $u(n)$ produced at the output of the filter. Show that

$$\chi(R) = \left(\frac{H_{\max}}{H_{\min}} \right)^2$$

14. Szegö's theorem states that if $g(\cdot)$ is a continuous function, then

$$\lim_{M \rightarrow \infty} \frac{g(\lambda_1) + g(\lambda_2) + \dots + g(\lambda_M)}{M} = \frac{1}{2\pi} \int_{-\pi}^{\pi} g[S(\omega)]d\omega$$

where $S(\omega)$ is the power spectral density of a stationary discrete-time stochastic process $u(n)$, and $\lambda_1, \lambda_2, \dots, \lambda_M$ are the eigenvalues of the associated correlation matrix \mathbf{R} . It is assumed that the process $u(n)$ is limited to the interval $-\pi < \omega \leq \pi$. Using this theorem, show that

$$\lim_{M \rightarrow \infty} [\det(\mathbf{R})]^{1/M} = \exp \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln [S(\omega)]d\omega \right)$$

15. Consider a linear system of equations described by

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}$$

where \mathbf{R} is an M -by- M matrix, and \mathbf{w}_o and \mathbf{p} are M -by-1 vectors. The vector \mathbf{w}_o represents the set of unknown parameters. Due to a combination of factors (e.g., measurement inaccuracies, computational errors), the matrix \mathbf{R} is perturbed by a small amount $\delta\mathbf{R}$, producing a corresponding change $\delta\mathbf{w}$ in the vector of unknowns.

- (a) Show that

$$\frac{\|\delta\mathbf{w}\|}{\|\mathbf{w}_o\|} \leq \chi(\mathbf{R}) \frac{\|\delta\mathbf{R}\|}{\|\mathbf{R}\|}$$

where $\chi(\mathbf{R})$ is the condition number of \mathbf{R} , and $\|\cdot\|$ denotes the norm of the quantity enclosed within.

- (b) Develop the corresponding formula for a small change in the vector \mathbf{p} . Hint: Use the inequality

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

16. Consider the three-dimensional vector space of Fig. 4.1. Let the subspace \mathcal{S} denote the $\mathbf{q}_{i_1}, \mathbf{q}_{i_2}$ -plane where i_1, i_2 is a subset of $\{1, 2, 3\}$. Let the subspace \mathcal{S}' denote the $\mathbf{q}_2, \mathbf{q}_3$ -plane.

- (a) Specify a vector \mathbf{x} of unit length that is common to the subspaces \mathcal{S} and \mathcal{S}' .
(b) What is the Rayleigh coefficient of the vector \mathbf{x} specified in part (a)? Justify your answer in light of the minimax theorem.

17. Consider an M -by- M doubly symmetric matrix \mathbf{R} that is symmetric about both the main diagonal and the secondary diagonal. Let \mathbf{J} denote an M -by- M matrix that consists of 1's along the secondary diagonal and zeros everywhere else. The matrix \mathbf{J} is called a *reverse operator* or *exchange matrix* because \mathbf{JR} reverses the rows of matrix \mathbf{R} , \mathbf{RJ} reverses the columns of \mathbf{R} , and \mathbf{JRJ} reverses both the rows and columns of \mathbf{R} .

- (a) Show that for the matrix \mathbf{R} to be doubly symmetric, a necessary and sufficient condition is

$$\mathbf{JRJ} = \mathbf{R}$$

Noting that $\mathbf{J}^{-1} = \mathbf{J}$, show that the inverse of matrix \mathbf{R} is also doubly symmetric, as shown by

$$\mathbf{JR}^{-1}\mathbf{J} = \mathbf{R}^{-1}$$

- (b) Assume that the doubly symmetric matrix \mathbf{R} has distinct eigenvalues. Hence show that the matrix \mathbf{R} has $\lfloor (M+1)/2 \rfloor$ symmetric eigenvectors and $\lfloor M/2 \rfloor$ skew-symmetric eigenvectors, where $\lfloor X \rfloor$ denotes the largest integer less than or equal to X . An eigenvector \mathbf{q} is said to be *symmetric* if

$$\mathbf{J}\mathbf{q} = \mathbf{q}$$

and *skew symmetric* if

$$\mathbf{J}\mathbf{q} = -\mathbf{q}$$

where \mathbf{J} is the reverse operator.

- (c) Let $A(z)$ denote the transfer function of an eigenfilter of the doubly symmetric matrix \mathbf{R} . Show that if $A(z)$ is associated with a symmetric eigenvector or skew-symmetric eigenvector of \mathbf{R} and if z_i is a zero of $A(z)$, then so is $1/z_i$.

[The properties described in this problem and the next three are taken from Makhoul (1981) and Reddi (1984).]

18. Let \mathbf{R} denote an M -by- M nonsingular symmetric Toeplitz matrix. Naturally, the properties described in Problem 17 apply to the matrix \mathbf{R} . Note, however, that the inverse matrix \mathbf{R}^{-1} is not Toeplitz, in general. But owing to the special structure of a Toeplitz matrix, the matrix \mathbf{R} has two additional properties:

- (a) Let λ_{\max} denote the largest eigenvalue of the matrix \mathbf{R} , which is assumed to be distinct. Show that the discrete transfer function of the eigenfilter associated with λ_{\max} has all of its zeros located on the unit circle in the z -plane.
 (b) Let λ_{\min} denote the smallest eigenvalue of the matrix \mathbf{R} , which is assumed to be distinct. Show that the discrete transfer function of the eigenfilter associated with λ_{\min} has all of its zeros located on the unit circle in the z -plane.

19. Consider the *normalized* 3-by-3 correlation matrix

$$\mathbf{R} = \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}$$

where

$$\rho_i = \frac{r(i)}{r(0)}, \quad i = 1, 2$$

- (a) Using properties (b) and (c) of Problem 17, demonstrate the following results:

- (1) The matrix \mathbf{R} has a single skew-symmetric eigenvector of the form

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} [1, 0, -1]^T$$

that is associated with the eigenvalue

$$\lambda_1 = 1 - \rho_2$$

- (2) The matrix \mathbf{R} has two symmetric eigenvectors of the form

$$\mathbf{q}_i = \frac{1}{\sqrt{1 + c_i^2}} [1, c_i, 1]^T, \quad i = 2, 3$$

where c_i is related to the corresponding eigenvalue λ_i by

$$c_i = \frac{2\rho_1}{\lambda_i - 1} = \frac{\lambda_i - 1 - \rho_2}{\rho_1}, \quad i = 2, 3$$

Hence, complete the specification of the eigenvalues and the eigenvectors of the matrix \mathbf{R} .

- (b) Given that the eigenvalues of matrix \mathbf{R} are distinct and ordered as $\lambda_1 > \lambda_2 > \lambda_3$, and given that the eigenfilters associated with λ_1 and λ_3 have their zeros on the unit circle in accordance with properties (a) and (b) of Problem 18, respectively, find the condition that the coefficient c_2 must satisfy for the following two situations to occur:
- (1) The eigenfilter associated with eigenvalue λ_2 will *also* have its zeros on the unit circle.
 - (2) The eigenfilter associated with eigenvalue λ_2 will *not* have its zeros on the unit circle.
- Illustrate both of these situations with selected values for the correlation coefficients ρ_2 and ρ_3 .
20. In this problem, we wish to establish the result given in Eq. (4.84) for the mean-square error produced by the transmission system of Fig. 4.3(b), using a low-rank model of the input data $\mathbf{u}(n)$.

The transmitted signal vector $\mathbf{c}(n)$ and the reconstructed signal vector $\mathbf{y}_{\text{direct}}(n)$ at the receiver output are defined by Eqs. (4.81) and (4.83), respectively. Derive Eq. (4.84) using the following properties:

- (a) The eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p$ associated with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_p$ of the correlation matrix \mathbf{R} of the input vector $\mathbf{u}(n)$ form an orthonormal set.
 - (b) The data vector $\mathbf{u}(n)$ and the noise vector $\mathbf{v}(n)$ are uncorrelated.
 - (c) The elements of the noise vector $\mathbf{v}(n)$ are drawn from a white noise process of zero mean and variance σ^2 .
21. To solve the optimum filtering problem described in Section 4.5, we selected an eigenfilter associated with the largest eigenvalue of the correlation matrix of the signal component at the filter input. What would be the result of selecting an eigenfilter associated with the smallest eigenvalue of this correlation matrix? Justify your answer.