# PART 4
# Nonlinear Adaptive Filtering

The last part of the book is devoted to some aspects of nonlinear adaptive filtering. In particular, in Chapter 18 we study the blind deconvolution problem, the solution of which may require the use of higher-order statistics and therefore some form of nonlinearity built into the design of the adaptive filtering algorithm. The use of cyclostationarity for solving the blind equalization problem is also discussed in Chapter 18.

In the remaining two chapters we consider two important types of feedforward multilayer neural networks, the design of which relies on some form of supervised learning. Chapter 19 discusses the backpropagation learning algorithm for the training of multilayer perceptrons; this algorithm may be viewed as a generalization of the LMS algorithm. Chapter 20 discusses radial basis-function networks that operate in a manner entirely different from multilayer perceptrons.

CHAPTER

# 18

# Blind Deconvolution

*Deconvolution* is a signal processing operation that ideally unravels the effects of convolution performed by a linear time-invariant system operating on an input signal. More specifically, in deconvolution, the output signal and the system are both known, and the requirement is to reconstruct what the input signal must have been. In *blind deconvolution*, or in more precise terms, *unsupervised deconvolution*, only the output signal is known (both the system and the input signal are unknown), and the requirement is to find both the input signal and the system itself. Clearly, blind deconvolution is a more difficult signal-processing task than ordinary deconvolution.

We may identify two broadly defined families of blind deconvolution algorithms, depending on the additional information that is used by the algorithm to make up for the unavailability of the system (channel) input:

1. *Higher-order statistics (HOS)-based algorithms:* This family of blind deconvolution algorithms may itself be subdivided into two groups:
   - *Implicit HOS-based algorithms,* which exploit higher-order statistics of the received signal in an implicit sense; this group of blind deconvolution algorithms includes *Bussgang algorithms,* so called because the deconvolved sequence assumes Bussgang statistics when the algorithm converges in the mean value.

- *Explicit HOS-based algorithms,* which explicitly use *higher-order cumulants* or their discrete Fourier transforms known as *polyspectra;* the property of polyspectra to preserve phase information makes them well suited for blind deconvolution.

2. *Cyclostationary statistics-based algorithms,* which exploit the second-order cyclostationary statistics of the received signal; the property of cyclostationarity is known to arise in a modulated signal that results from varying the amplitude, phase, or frequency of a sinusoidal carrier, which is basic to the electrical communications process.

We begin our study of the blind deconvolution problem by discussing its theoretical implications and practical importance, which we do in the next section.

## 18.1 THEORETICAL AND PRACTICAL CONSIDERATIONS

Consider an *unknown* linear time-invariant system $\mathcal{L}$ with input $x(n)$ as depicted in Fig. 18.1. The input data (information-bearing) sequence $x(n)$ is assumed to consist of *independently and identically distributed (iid) symbols;* the only thing known about the input is its probability distribution. The problem is to restore $x(n)$ or equivalently, *to identify the inverse* $\mathcal{L}^{-1}$ *of the system* $\mathcal{L}$, *given the observed sequence* $u(n)$ *at the system output.*

If the system $\mathcal{L}$ is *minimum-phase* (i.e., the transfer function of the system has all of its poles and zeros confined to the interior of the unit circle in the $z$-plane), then not only is the system $\mathcal{L}$ stable, but so is the inverse system $\mathcal{L}^{-1}$. In this case, we may view the input sequence $x(n)$ as the "innovation" of the system output $u(n)$, and the inverse system $\mathcal{L}^{-1}$ is just a *whitening filter;* with it, the blind deconvolution problem is solved. These observations follow from the study of linear prediction presented in Chapter 6.

In many practical situations, however, the system $\mathcal{L}$ may *not* be minimum phase. A system is said to be *nonminimum phase* if its transfer function has any of its zeros located outside the unit circle in the $z$-plane; exponential stability of the system dictates that the poles must be located inside the unit circle. Practical examples of a nonminimum phase system include a telephone channel and a fading radio channel. In this situation, the restoration of the input sequence $x(n)$, given the channel output, is a more difficult problem.

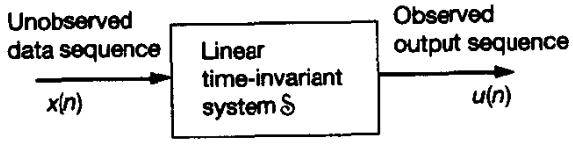



**Figure 18.1** Setting the stage for blind deconvolution.

Typically, adaptive equalizers used in digital communications require an initial *training period*, during which a *known* data sequence is transmitted. A replica of this sequence is made available at the receiver in proper synchronism with the transmitter, thereby making it possible for adjustments to be made to the equalizer coefficients in accordance with the adaptive filtering algorithm employed in the equalizer design. When the training is completed, the equalizer is switched to its *decision-directed mode*, and normal data transmission may then commence. (These two modes of operation of an adaptive equalizer were discussed in Section 7 of the introductory chapter.) However, there are practical situations where it would be highly desirable for a receiver to be able to achieve complete adaptation *without* the cooperation of the transmitter. For example, in a *multipoint data network* involving a *control unit* connected to several *data terminal equipments* (DTEs), we have a "master-slave" situation, in that a DTE is permitted to transmit only when its modem is polled by the modem of the control unit. A problem peculiar to these networks is that of retraining the receiver of a DTE unable to recognize data and polling messages, due to severe variations in channel characteristics or simply because that particular receiver was not powered on during initial synchronization of the network. Clearly, in a large or heavily loaded multipoint network, data throughput is increased and the burden of monitoring the network is eased if some form of *blind equalization* is built into the receiver design (Godard, 1980).

Another class of communication systems that may need blind equalization is *wireless communication systems* using digital technology. In particular, in a *mobile communications channel* it is impractical to employ a training sequence of long duration for two reasons:

- The system *cost* involved in the repeated transmission of a known sequence to train the equalizer at the receiving end of the system is typically too *high*.
- The unavoidable presence of *multipath fading* makes it difficult (if not impossible) to establish data transmission over the channel when outage in the system occurs; the fading phenomenon arises because the transmitted signal tends to propogate along several paths, each of different electrical length.

In reflection seismology, the traditional method of removing the source waveform from a seismogram is to use *linear-predictive deconvolution* (see Section 7 of the introductory chapter). The method of predictive deconvolution is derived from four fundamental assumptions (Gray, 1979):

1. The reflectivity series is *white*. This assumption is, however, often violated by reflection seismograms as the reflectivities result from a differential process applied to acoustic impedances. In many sedimentary basins there are thin beds that cause the reflectivity series to be correlated in sign.
2. The source signal is *minimum phase*, in that its $z$-transform has all of its zeros confined to the interior of the unit circle in the $z$-plane; here, it is presumed that

the source signal is in discrete-time form. This assumption is valid for several explosive sources (e.g., dynamite), but it is only approximate for more complicated sources such as those used in marine exploration.

3. The reflectivity series and noise are statistically independent and stationary in time. The stationarity assumption, however, is violated because of spherical divergence and attenuation of seismic waves. To cope with nonstionarity of the data, we may use adaptive deconvolution, but such a method often destroys primary events of interest.
4. The *minimum mean-square error criterion* is used to solve the linear prediction problem. This criterion is appropriate only when the prediction errors (the reflectivity series and noise) have a Gaussian distribution. Statistical tests performed on reflectivity series, however, show that their kurtosis is much higher than that expected from a Gaussian distribution. The *skewness* and *kurtosis* of a distribution function are defined as follows, respectively:

$$\gamma_1 = \frac{\mu_3}{\sigma^3}$$

and

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3$$

where $\sigma^2$ is the variance of the distribution, and $\mu_3$ and $\mu_4$ are its third- and fourth-order central moments, respectively.

Assumptions 1 and 2 were explicitly mentioned in the presentation of the method of predictive deconvolution in the introductory chapter. Assumptions 3 and 4 are implicit in the application of Wiener filtering that is basic to the solution of the linear prediction problem, as presented in Chapter 6. The main point of the discussion here is that valuable phase information contained in a reflection seismogram is ignored by the method of predictive deconvolution. This limitation is overcome by using *blind deconvolution* (Godfrey and Rocca, 1981).

Blind equalization in digital communications and blind deconvolution in reflection seismology are examples of a special kind of adaptive inverse filtering that operate in an *unsupervised* manner (i.e., without access to a desired response). Only the received signal and some additional information in the form of a *probabilistic source model* are provided. In the case of equalization for digital communications, the model describes the statistics of the transmitted data sequence. In the case of seismic deconvolution, the model describes the statistics of the earth's reflection coefficients.

Having clarified the framework within which the use of blind deconvolution is feasible, we are ready to undertake a detailed study of its operation. Specifically, we begin by considering the Bussgang family of blind deconvolution algorithms in the context of equalization for digital communications.

## 18.2 BUSSGANG ALGORITHM FOR BLIND EQUALIZATION OF REAL BASEBAND CHANNELS

Consider the *baseband model* of a digital communications system, depicted in Fig. 18.2. The model consists of the cascade connection of a *linear communication channel* and a *blind equalizer*.

The channel includes the combined effects of a transmit filter, a transmission medium, and a receive filter. It is characterized by an impulse response $h_n$ that is *unknown*; it may be time varying, albeit slowly. The nature of the impulse response $h_n$ (i.e., whether it is real or complex valued) is determined by the type of modulation employed. To simplify the discussion, for the present, we assume that the impulse response is real, which corresponds to the use of *multilevel pulse-amplitude modulation* (*M-ary PAM*); the case of a complex impulse response is considered in the next section. We may thus describe the sampled input-output relation of the channel by the *convolution sum*

$$u(n) = \sum_{k=-\infty}^{\infty} h_k x(n-k), \qquad n = 0, \pm 1, \pm 2, \ldots \tag{18.1}$$

where $x(n)$ is the *data (message) sequence* applied to the channel input, and $u(n)$ is the resulting *channel output*. For this introductory treatment of blind deconvolution, the effect of receiver noise is ignored in Eq. (18.1). We are justified to do so, because degradation in the performance of data transmission (over a voice-grade telephone channel, say) is usually dominated by *intersymbol interference* due to channel dispersion. We further assume that

$$\sum_k h_k^2 = 1 \tag{18.2}$$

Equation (18.2) implies the use of *automatic gain control* (AGC) that keeps the variance of the channel output $u(n)$ essentially constant. Also, in general, the channel is *noncausal*, which means that

$$h_n \neq 0 \qquad \text{for } n < 0 \tag{18.3}$$

The problem we wish to solve is the following:

Given the received signal $u(n)$, reconstruct the original data sequence $x(n)$ applied to the channel input.

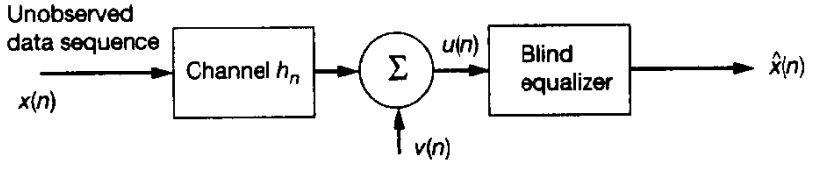



**Figure 18.2** Cascade connection of an unknown channel and blind equalizer.

Equivalently, we may restate the problem as follows:

Design a blind equalizer that is the inverse of the unknown channel, with the channel input being unobservable and with no desired response available.

To solve the blind equalization problem, we need to prescribe a *probabilistic model* for the data sequence $x(n)$. For the problem at hand, we assume the following (Bellini, 1986, 1994):

1. The data sequence $x(n)$ is *white*; that is, the data symbols are *iid random variables*, with zero mean and unit variance, as shown by

$$E[x(n)] = 0 \tag{18.4}$$

and

$$E[x(n)x(k)] = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \tag{18.5}$$

where $E$ is the statistical expectation operator.

2. The *probability density function* of the data symbol $x(n)$ is *symmetric* and *uniform*; that is (see Fig. 18.3),

$$f_X(x) = \begin{cases} 1/2\sqrt{3}, & -\sqrt{3} \leq x < \sqrt{3} \\ 0, & \text{otherwise} \end{cases} \tag{18.6}$$

This distribution has the merit of being independent of the number $M$ of amplitude levels employed in the modulation process.

Note that Eq. (18.4) and the first line of Eq. (18.5) follow from Eq. (18.6).

With the distribution of $x(n)$ assumed to be symmetric, as in Fig. 18.3, we find that the whole data sequence $-x(n)$ has the same law as $x(n)$. Hence we cannot distinguish the desired inverse filter $\mathcal{L}^{-1}$ (corresponding to $x(n)$) from the opposite one $-\mathcal{L}^{-1}$ (corre-

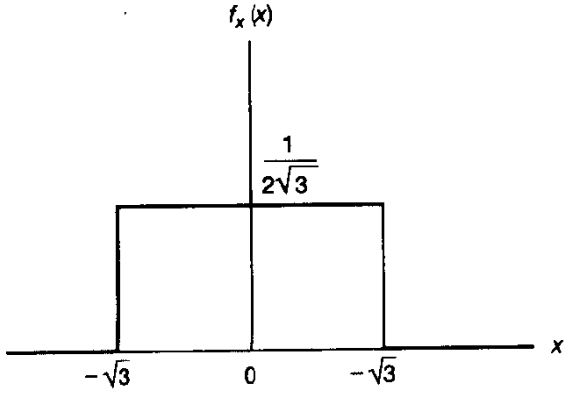



**Figure 18.3** Uniform distribution.

sponding to $-x(n)$). We may overcome this *sign ambiguity problem* by *initializing* the deconvolution algorithm such that there is a single nonzero tap weight with the desired algebraic sign (Benveniste et al., 1980).

### Iterative Deconvolution: The Objective

Let $w_i$ denote the impulse response of the *ideal inverse filter*, which is related to the impulse response $h_i$ of the channel as follows:

$$\sum_i w_i h_{l-i} = \delta_l \tag{18.7}$$

where $\delta_l$ is the *Kronecker delta*:

$$\delta_l = \begin{cases} 1, & l = 0 \\ 0, & l \neq 0 \end{cases} \tag{18.8}$$

An inverse filter defined in this way is "ideal" in the sense that it reconstructs the transmitted data sequence $x(n)$ *correctly*. To demonstrate this, we first write

$$\sum_i w_i u(n-i) = \sum_i \sum_k w_i h_k x(n-i-k) \tag{18.9}$$

Let

$$k = l - i$$

Making this change of indices in Eq. (18.9), and interchanging the order of summation, we get

$$\sum_i w_i u(n-i) = \sum_l x(n-l) \sum_i w_i h_{l-i} \tag{18.10}$$

Hence, using Eq. (18.7) in (18.10) and then applying the definition of Eq. (18.8), we get

$$\begin{aligned} \sum_i w_i u(n-i) &= \sum_l \delta_l x(n-l) \\ &= x(n) \end{aligned} \tag{18.11}$$

which is the desired result.

For the situation described herein, the impulse response $h_n$ is unknown. We cannot therefore use Eq. (18.7) to determine the inverse filter. Instead, we use an *iterative deconvolution procedure* to compute an *approximate inverse filter* characterized by the impulse response $\hat{w}_i(n)$. The index $i$ refers to the *tap-weight nunber* in the *transversal filter* realization of the approximate inverse filter, as indicated in Fig. 18.4. The index $n$ refers to the *iteration number*; each iteration corresponds to the transmission of a data symbol. The computation is performed iteratively in such a way that the convolution of the impulse response $\hat{w}(n)$ with the received signal $u(n)$ results in the complete or partial removal of the

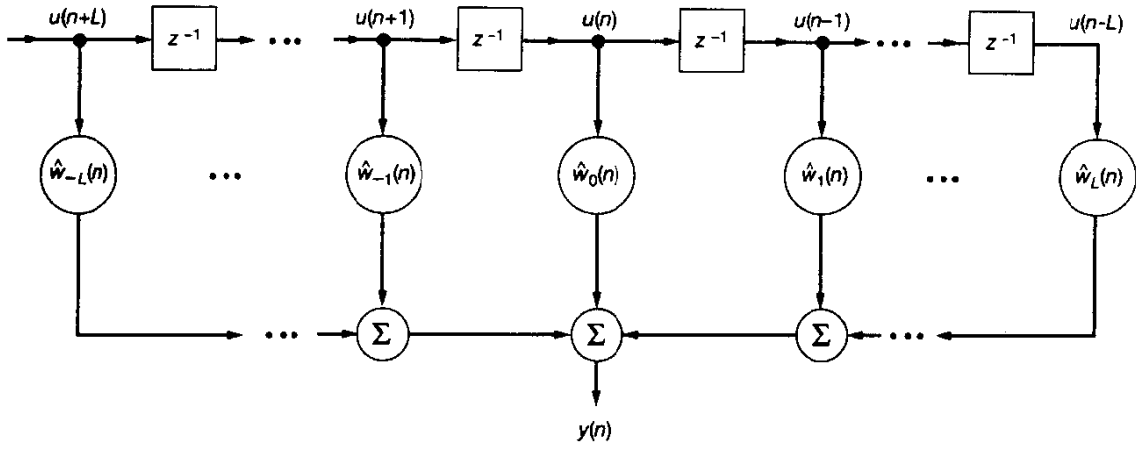



**Figure 18.4** Transversal filter realization of approximate inverse filter; use of real data is assumed.

intersymbol interference (Bellini, 1986). Thus, at the $n$th iteration we have an approximately deconvolved sequence

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i) \tag{18.12}$$

where $2L + 1$ is the truncated *length* of the impulse response $\hat{w}_i(n)$ (see Fig. 18.4). For the sake of simplicity, it is customary to assume that the transversal filter (equalizer) is symmetric about the midpoint $i = 0$ but this assumption is not required yet.

The convolution sum on the left-hand side of Eq. (18.11), pertaining to the ideal inverse filter, is *infinite* in extent, in that the index $i$ ranges from $-\infty$ to $\infty$. In this case, we speak of a *doubly infinite filter (equalizer)*. On the other hand, the convolution sum on the right-hand side of Eq. (18.12) pertaining to the approximate inverse filter is *finite* in extent, in that $i$ extends from $-L$ to $L$. In this latter case, which is how it usually is in practice, we speak of a *finitely parameterized filter (equalizer)*. Clearly, we may rewrite Eq. (18.12) as follows:

$$y(n) = \sum_{i} \hat{w}_i(n)u(n-i), \qquad \hat{w}_i(n) = 0 \text{ for } |i| > L$$

or, equivalently,

$$y(n) = \sum_{i} w_i u(n-i) + \sum_{i} [\hat{w}_i(n) - w_i]u(n-i) \tag{18.13}$$

Let

$$v(n) = \sum_{i} [\hat{w}_i(n) - w_i]u(n-i), \qquad \hat{w}_i = 0 \text{ for } |i| > L \tag{18.14}$$

Then, using the ideal result of Eq. (18.11) and the definition of Eq. (18.14), we may simplify Eq. (18.13) as follows:

$$y(n) = x(n) + v(n) \tag{18.15}$$

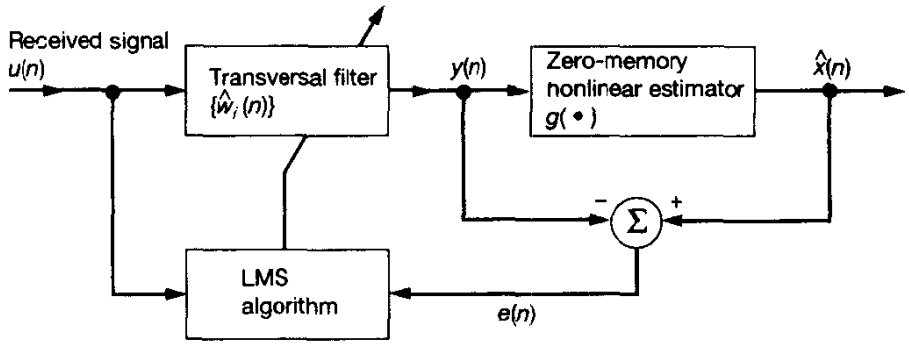



**Figure 18.5** Block diagram of blind equalizer.

The term $v(n)$ is called the *convolutional noise*, representing the *residual* intersymbol interference that results from the use of an approximate inverse filter.

The inverse filter output $y(n)$ is next applied to a *zero-memory nonlinear estimator*, producing the estimate $\hat{x}(n)$ for the data symbol $x(n)$. This operation is depicted in the block diagram of Fig. 18.5. We may thus write

$$\hat{x}(n) = g(y(n)) \tag{18.16}$$

where $g(\cdot)$ is some nonlinear function. The issue of nonlinear estimation is discussed in the next subsection.

Ordinarily, we find that the estimate $\hat{x}(n)$ at iteration $n$ is not reliable enough. Nevertheless, we may use it in an *adaptive* scheme to obtain a "better" estimate at the next iteration, $n + 1$. Indeed, we have a variety of *linear* adaptive filtering algorithms (discussed in previous chapters) at our disposal that we can use to perform this adaptive parameter estimation. In particular, a simple and yet effective scheme is provided by the LMS algorithm. To apply it to the problem at hand, we note the following:

1. The $i$th tap input of the transversal filter at iteration (time) $n$ is $u(n - i)$.
2. Viewing the nonlinear estimate $\hat{x}(n)$ as the "desired" response [since the transmitted data symbol $\hat{x}(n)$ is unavailable to us], and recognizing that the corresponding transversal filter output is $y(n)$, we may express the *estimation error* for the iterative deconvolution procedure as

$$e(n) = \hat{x}(n) - y(n) \tag{18.17}$$

3. The $i$th tap weight $\hat{w}_i(n)$ at iteration $n$ represents the "old" parameter estimate.

Accordingly, the "updated" value of the $i$th tap weight at iteration $n + 1$ is computed as follows:

$$\hat{w}_i(n + 1) = \hat{w}_i(n) + \mu u(n - i)e(n), \qquad i = 0, \pm 1, \ldots, \pm L \tag{18.18}$$

where $\mu$ is the *step-size parameter*. Note that for the situation being considered here, the data are all *real* valued.

Equations (18.12), (18.16), (18.17), and (18.18) constitute the iterative deconvolution algorithm for the blind equalization of a real baseband channel (Bellini, 1986). As remarked earlier, each iteration of the algorithm corresponds to the transmission of a data symbol. It is assumed that the symbol duration is known at the receiver.

A block diagram of the blind equalizer is shown in Fig. 18.5. The idea of generating the estimation error $e(n)$, as detailed in Eqs. (18.16) and (18.17), is similar in philosophy to the decision-directed mode of operating an adaptive equalizer. More will be said on this issue later in the section.

### Nonconvexity of the Cost Function

The ensemble-averaged cost function corresponding to the tap-weight update equation (18.18) is defined by

$$\begin{aligned} J(n) &= E[e^2(n)] \\ &= E[(\hat{x}(n) - y(n))^2] \\ &= E[(g(y(n)) - y(n))^2] \end{aligned} \tag{18.19}$$

where $y(n)$ is defined by Eq. (18.12). In the LMS algorithm, the cost function is a quadratic (convex) function of the tap weights and therefore has a well-defined minimum point. By contrast, the cost function $J(n)$ of Eq. (18.19) is a *nonconvex* function of the tap weights. This means that, in general, the error-performance surface of the iterative deconvolution procedure described here may have *local minima* in addition to *global minima*. More than one global minimum may exist, corresponding to data sequences that are equivalent under the chosen blind deconvolution criterion (e.g., sign ambiguity). The nonconvexity of the cost function $J(n)$ may arise because of the fact that the estimate $\hat{x}(n)$, performing the role of an internally generated "desired response," is produced by passing the linear combiner output $y(n)$ through a zero-memory nonlinearity, and also because $y(n)$ is itself a function of the tap weights.

In any event, the nonconvex form of the cost function $J(n)$ may result in *ill-convergence* of the iterative deconvolution algorithm described by Eqs. (18.12) and (18.16) to (18.18). The important issue of convergence is considered in some greater detail later in this section.

### Statistical Properties of Convolutional Noise

The additive convolutional noise $v(n)$ is defined in Eq. (18.14). To develop a more refined formula for $v(n)$, we note that the tap input $u(n - i)$ involved in the summation on the right-hand side of this equation is given by [see Eq. (18.1)]

$$u(n - i) = \sum_k h_k x(n - i - k) \tag{18.20}$$

We may therefore rewrite Eq. (18.14) as a double summation:

$$v(n) = \sum_i \sum_k h_k[\hat{w}_i(n) - w_i]x(n - i - k) \tag{18.21}$$

Let

$$n - i - k = l$$

Hence, we may also write

$$v(n) = \sum_l x(l)\nabla(n - l) \tag{18.22}$$

where

$$\nabla(n) = \sum_k h_k[\hat{w}_{n-k}(n) - w_{n-k}] \tag{18.23}$$

The sequence $\nabla(n)$ is a sequence of small numbers, representing to the *residual impulse response* of the channel due to imperfect equalization. We imagine the sequence $\nabla(n)$ as a *long and oscillatory wave* that is convolved with the transmitted data sequence $x(n)$ to produce the convolutional noise sequence $v(n)$, as indicated in Eq. (18.22).

The definition of Eq. (18.22) is basic to the statistical characterization of the convolution noise $v(n)$. The mean of $v(n)$ is zero, as shown by

$$\begin{aligned} E[v(n)] &= E\left[\sum_l x(l)\nabla(n - l)\right] \\ &= \sum_l \nabla(n - l)E[x(l)] \\ &= 0 \end{aligned} \tag{18.24}$$

where in the last line we have made use of Eq. (18.4). Next, the autocorrelation function of $v(n)$ for a lag $j$ is given by

$$\begin{aligned} E[v(n)v(n - j)] &= E\left[\sum_l x(l)\nabla(n - l)\sum_m x(m)\nabla(n - m - j)\right] \\ &= \sum_l \sum_m \nabla(n - l)\nabla(n - m - j)E[x(l)x(m)] \\ &= \sum_l \nabla(n - l)\nabla(n - l - j) \end{aligned} \tag{18.25}$$

where in the last line we have made use of Eq. (18.5). Since $\nabla(n)$ is a long and oscillatory waveform, the sum on the right-hand side of Eq. (18.25) is nonzero only for $j = 0$, obtaining

$$E[v(n)v(n - j)] = \begin{cases} \sigma^2, & j = 0 \\ 0, & j \neq 0 \end{cases} \tag{18.26}$$

where

$$\sigma^2(n) = \sum_l \nabla^2(n - l) \tag{18.27}$$

Based on Eqs. (18.24) and (18.26), we may thus describe the convolutional noise process $v(n)$ as a *zero-mean white-noise process of time-varying variance equal to* $\sigma^2(n)$, defined by Eq. (18.27).

According to the model of Eq. (18.22), the convolutional noise $v(n)$ is a weighted sum of iid variables representing different transmissions of data symbols. If, therefore, the residual impulse response $\nabla(n)$ is long enough, the *central limit theorem* makes a *Gaussian* model for $v(n)$ to be plausible.

Having characterized the convolutional noise $v(n)$ by itself, all that remains for us to do is to evaluate the *cross-correlation* between it and the data sample $x(n)$. These two random variables are certainly *correlated* with each other, since $v(n)$ is the result of convolving the residual impulse response $\nabla(n)$ with $x(n)$, as shown in Eq. (18.22). However, the cross-correlation between $v(n)$ and $x(n)$ is negligible compared to the variance of $v(n)$. To demonstrate this, we write

$$\begin{aligned} E[x(n)v(n-j)] &= E[x(n)\sum_l x(l)\nabla(n-l-j)] \\ &= \sum_l \nabla(n-l-j)E[x(n)x(l)] \\ &= \nabla(-j) \end{aligned} \tag{18.28}$$

where, in the last line, we have made use of Eq. (18.5). Here again, using the assumption that $\nabla(n)$ is a long and oscillatory waveform, we deduce that the variance of $v(n)$ is large compared to the magnitude of the cross-correlation $E[x(n)v(n-j)]$.

Since the data sequence $x(n)$ is white by assumption and the convolutional noise sequence $v(n)$ is approximately white by deduction, and since these two sequences are essentially uncorrelated, it follows that their sum $y(n)$ is approximately white too. This suggests that $x(n)$ and $v(n)$ may be taken to be essentially independent. We may thus model the convolutional noise $v(n)$ as an *additive, zero-mean, white Gaussian noise process that is statistically independent of the data sequence* $x(n)$.

Because of the approximations made in deriving the model described herein for the convolutional noise, its use in an iterative deconvolution process yields a *suboptimal* estimator for the data sequence. In particular, given that the iterative deconvolution process is convergent, the intersymbol interference (ISI) during the latter stages of the process may be small enough for the model to be applicable. In the early stages of the iterative deconvolution process, however, the ISI is typically large with the result that the data sequence and the convolutional noise are strongly correlated, and the convolutional noise sequence is more uniform than Gaussian (Godfrey and Rocca, (1981).

### Zero-Memory Nonlinear Estimation of the Data Sequence

We are now ready to consider the next important issue, namely, that of estimating the data sequence $x(n)$, given the deconvolved sequence $y(n)$ at the transversal filter output. Specifically, we may formulate the estimation problem as follows: We are given a (filtered) observation $y(n)$ that consists of the sum of two components (see Fig. 18.6):

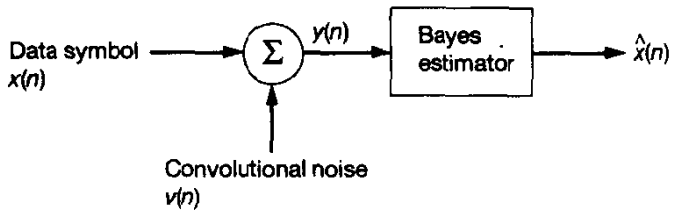



**Figure 18.6** Estimation of the data symbol $x(n)$, given the observation $y(n)$.

1. A uniformly distributed data symbol $x(n)$ with zero mean and unit variance
2. A white Gaussian noise $v(n)$ with zero mean and variance $\sigma^2(n)$, which is statistically independent of $x(n)$

The requirement is to derive a *Bayes estimate* of $x(n)$, optimized in a statistical sense.

Before proceeding with this classical estimation problem, two noteworthy observations are in order. First, the estimate is naturally a *conditional estimate* that depends on the optimization criterion. Second, although the estimate (in theory) is optimum in a mean-square error sense, in the context of our present situation, it is *suboptimum* by virtue of the approximations made in the development of the model described above for the convolutional noise $v(n)$.

An optimization criterion of particular interest is that of minimizing the mean-square value of the error between the actual transmission $x(n)$ and the estimation $\hat{x}(n)$. The choice of this optimization criterion yields a *conditional mean estimator*[1] that is both sensible and robust.

For convenience of presentation, we will supress the dependence of random variables on time $n$. Thus, given the observation $y$, the conditional mean estimate $\hat{x}$ of the random variable $x$ is written as $E[\hat{x}|y]$, where $E$ is the expectation operator. Let $f_X(x|y)$ denote the *conditional probability density function* of $x$, given $y$. We thus have

$$\begin{aligned} \hat{x} &= E[x|y] \\ &= \int_{-\infty}^{\infty} x f_X(x|y)\, dx \end{aligned} \tag{18.29}$$

From *Bayes' rule*, we have

$$f_X(x|y) = \frac{f_Y(y|x)\, f_X(x)}{f_Y(y)} \tag{18.30}$$

where $f_Y(y|x)$ is the conditional probability density function of $y$, given $x$; and $f_X(x)$ and $f_Y(y)$ are the probability density functions of $x$ and $y$, respectively. We may therefore rewrite the formula of Eq. (18.29) as

$$\hat{x} = \frac{1}{f_Y(y)} \int_{-\infty}^{\infty} x f_Y(y|x) f_X(x) dx \tag{18.31}$$

[1] For a derivation of the conditional mean and its relation to mean-squared-error estimation, see Appendix D.

Let the deconvolved sequence $y(n)$ be a scaled version of the original data sequence $x(n)$, except for an additive noise term $v(n)$, as shown by

$$y = c_0 x + v \tag{18.32}$$

The scaling factor $c_0$ is slightly smaller than unity. This factor has been included in Eq. (18.32) so as to keep $E[y^2]$ equal to 1. In accordance with the statistical model for the conventional noise $v$ developed previously, $x$ and $v$ are statistically independent. With $v$ modeled to have zero mean and variance $\sigma^2$, we readily see from Eq. (18.32) that the scaling factor $c_0$ is

$$c_0 = \sqrt{1 - \sigma^2} \tag{18.33}$$

Furthermore, from Eq. (18.32) it follows that

$$f_Y(y|x) = f_V(y - c_0 x) \tag{18.34}$$

Accordingly, the use of Eq. (18.34) in (18.31) yields

$$\hat{x} = \frac{1}{f_Y(y)} \int_{-\infty}^{\infty} x f_V(y - c_0 x) f_X(x)\, dx \tag{18.35}$$

The evaluation of $\hat{x}$ is straightforward but tedious. To proceed with it, we may note the following:

1. The mathematical form of the estimate $\hat{x}(n)$ produced at the output of the Bayes (conditional mean) estimator depends on the probability density function of the original data symbol $x(n)$. For the analysis presented here, we assume that the data symbol $x$ is *uniformly distributed* with zero mean and unit variance; its probability density function is given in Eq. (18.6), which is reproduced here for convenience:

$$f_X(x) = \begin{cases} 1/2\sqrt{3}, & -\sqrt{3} \le x < \sqrt{3} \\ 0, & \text{otherwise} \end{cases} \tag{18.36}$$

2. The convolutional noise $v$ is *Gaussian distributed* with zero mean and variance $\sigma^2$; its probability density function is

$$f_V(v) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{v^2}{2\sigma^2}\right) \tag{18.37}$$

3. The filtered observation $y$ is the sum of $c_0 x$ and $v$; its probability density function is therefore equal to the convolution of the probability density function of $x$ with that of $v$, as shown by

$$f_Y(y) = \int_{-\infty}^{\infty} f_X(x) f_V(y - c_0 x)\, dx \tag{18.38}$$

Using Eqs. (18.36) to (18.38) in (18.35), we get (Bellini, 1988)

$$x = \frac{1}{c_0 y} - \frac{\sigma}{c_0} \frac{Z(y_1) - Z(y_2)}{Q(y_1) - Q(y_2)} \tag{18.39}$$

where the variables $y_1$ and $y_2$ are defined by

$$y_1 = \frac{1}{\sigma}(y + \sqrt{3}\, c_0)$$

and

$$y_2 = \frac{1}{\sigma}(y - \sqrt{3}\, c_0)$$

The function $Z(y)$ is the *standardized* Gausian probability density function

$$Z(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \tag{18.40}$$

The function $Q(y)$ is the corresponding probability distribution function

$$Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-u^2/2}\, du \tag{18.41}$$

A small *gain correction* to the nonlinear estimator of Eq. (18.39) is needed in order to achieve perfect equalization[2] when the iterative deconvolution algorithm [described by Eqs. (18.16) to (18.18)] converges eventually. Perfect equalization requires that $y = x$. Under the minimum mean-square error condition, the estimation error is orthogonal to each of the tap inputs in the transversal filter realization of the approximate inverse filter. Putting all of this together, we find that the following condition must hold (Bellini, 1986, 1988):

$$E[\hat{x}\, g(\hat{x})] = 1 \tag{18.42}$$

where $g(\hat{x})$ is the nonlinear estimator $\hat{x} = g(y)$ with $y = \hat{x}$ for perfect equalization; see Problem 2.

Figure 18.7 shows the nonlinear estimate $\hat{x} = g(z)$ plotted versus $|z|$ for an eight-level PAM system (Bellini, 1986, 1988). The estimator is normalized in accordance with Eq.

[2] In general, for perfect equalization we require that

$$y = (x - D)e^{j\Phi}$$

where $D$ is a constant delay and $\phi$ is a constant phase shift. This condition corresponds to an equalizer whose transfer function has magnitude one and a linear phase response. We note that the input data sequence $x_i$ is stationary and the channel is linear time-invariant. Hence, the observed sequence $y(n)$ at the channel output is also stationary; its probability density function is therefore invariant to the constant delay $D$. The constant phase shift $\phi$ is also of no immediate consequence when the probability density function of the input sequence remains symmetric under rotation, which is indeed the case for the assumed density function given in Eq. (18.36). We may therefore simplify the condition for perfect equalization by requiring that $y = x$.

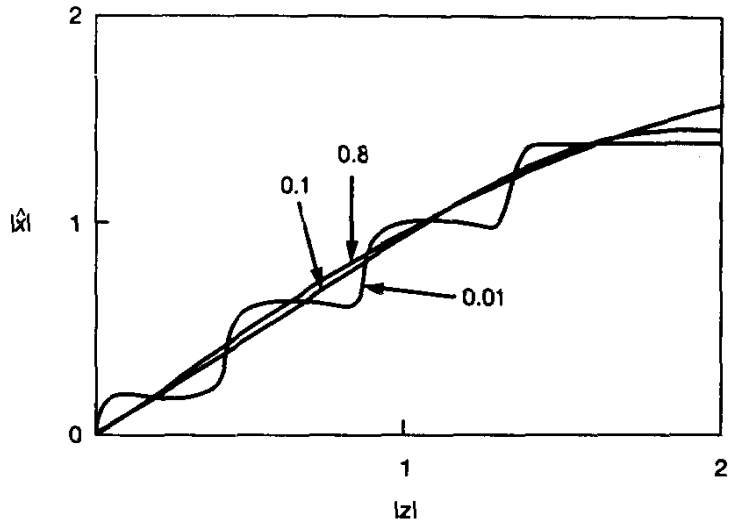



**Figure 18.7** Nonlinear estimators of eight-level data in Gaussian noise with $\hat{x} = g(z)$. The noise-to-(signal + noise) ratios are 0.01, 0.1, and 0.8. [From Bellini (1986), with permission of the IEEE]

(18.42). In this figure, three widely different levels of convolutional noise are considered. Here we note from Eq. (18.32) that the *distortion-to (signal plus distortion) ratio* is given by

$$\begin{aligned} \frac{E[(y - x)^2]}{E[y^2]} &= (1 - c_0)^2 + \sigma^2 \\ &= 2(1 - c_0) \end{aligned} \tag{18.43}$$

where in the last line we have made use of Eq. (18.33). The curves presented in Fig. 18.7 correspond to three values of this ratio, namely, 0.01, 0.1, and 0.8. We observe the following from these curves:

1. When the convolutional noise is low, the blind equalization algorithm approaches a minimum mean-squared-error criterion.
2. When the convolutional noise is high, the nonlinear estimator appears to be independent of the fine structure of the amplitude-modulated data. Indeed, different values of amplitude modulation levels result in only very small gain differences due to the normalization defined by Eq. (18.42). This suggests that the use of a uniform amplitude distribution for multilevel modulation systems is an adequate approximation.
3. The nonlinear estimator is robust with respect to variations in the variance of the convolutional noise.

### Convergence Considerations

For the iterative deconvolutional algorithm described by Eqs. (18.16) to (18.18) to converge in the mean value, we require the expected value of the tap weight $\hat{w}_i(n)$ to approach some constant value as the number of iterations $n$ approaches infinity. Correspondingly, we find that the *condition for convergence in the mean value* is described by

$$E[u(n-i)y(n)] = E[u(n-i)g(y(n))], \qquad \text{large } n, \text{ and } i = 0, \pm 1, \ldots, \pm L \tag{18.44}$$

Multiplying both sides of this equation by $\hat{w}_{i-k}$ and summing over $i$, we get

$$E[y(n)\sum_{i=-L}^{L} \hat{w}_{i-k}(n)u(n-i)] = E[g(y(n))\sum_{i=-L}^{L} \hat{w}_{i-k}(n)u(n-i)], \qquad \text{large } n \tag{18.45}$$

We next note from Eq. (18.12) that

$$\begin{aligned} y(n-k) &= \sum_{i=-L}^{L} \hat{w}_i(n)u(n-k-i) \\ &= \sum_{i=-L-k}^{L-k} \hat{w}_{i-k}(n)u(n-i), \qquad \text{large } n \end{aligned}$$

Provided that $L$ is large enough for the transversal equalizer to achieve perfect equalization, we may approximate the expression for $y(n-k)$ as

$$y(n-k) \simeq \sum_{i=-L}^{L} \hat{w}_{i-k}(n)u(n-i), \qquad \text{large } n \text{ and large } L \tag{18.46}$$

Accordingly, we may use Eq. (18.46) to simplify (18.45) as follows:

$$E[y(n)y(n-k)] \simeq E[g(y(n))y(n-k)], \qquad \text{large } n \text{ and large } L \tag{18.47}$$

We now recognize the following property. A stochastic process $y(n)$ is said to be a *Bussgang process* if it satisfies the condition

$$E[y(n)y(n-k)] = E[y(n)g(y(n-k))] \tag{18.48}$$

where the function $g(\cdot)$ is a zero-memory nonlinearity.[3] In other words, a Bussgang process has the property that its autocorrelation function is equal to the cross-correlation between that process and the output of a zero-memory nonlinearity produced by that process, with both correlations being measured for the same lag. Note that a Bussgang process satisfies Eq. (18.48) up to a multiplicative constant; in the case discussed here, the multiplicative constant is unity by virtue of the assumption made in Eq. (18.42).

[3] A number of stochastic processes belong to the class of Bussgang processes. Bussgang (1952) was the first to recognize that any correlated Gaussian process has the property described in Eq. (18.48). Subsequently, Barrett and Lampard (1955) extended Bussgang's result to all stochastic processes with exponentially decaying autocorrelation functions. This includes an independent process, since its autocorrelation function consists of a delta function that may be viewed as an infinitely fast decaying exponential (Gray, 1979).

Returning to the issue at hand, we may state that the process $y(n)$ acting as the input to the zero-memory nonlinearity in Fig. 18.5 is *approximately* a Bussgang process, provided that $L$ is large; the approximation becomes better as $L$ is made larger. It is for this reason that the blind equalization algorithm described here is referred to as a *Bussgang algorithm* (Bellini, 1986, 1988).

In general, convergence of the Bussgang algorithm is not guaranteed. Indeed, the cost function of the Bussgang algorithm operating with a finite $L$ is *nonconvex*; it may therefore have false minima.

For the idealized case of a doubly *infinite* equalizer, however, a rough proof of convergence of the Bussgang algorithm may be sketched as follows (Bellini, 1988). The proof relies on a theorem derived in Benveniste et al. (1980), which provides sufficient conditions for convergence.[4] Let the function $\psi(y)$ denote the dependence of the estimation error in the LMS algorithm on the transversal filter output $y(n)$. According to our terminology, we have [see Eqs. (18.16) and (18.17)]

$$\psi(y) = g(y) - y \tag{18.49}$$

The *Benveniste–Goursat–Ruget theorem* states that convergence of the Bussgang algorithm is guaranteed if the probability distribution of the data sequence $x(n)$ is *sub-Gaussian* and the second derivative of $\psi(y)$ is negative on the interval $[0, \infty)$. In particular, we may state the following:

1. A random variable $x$, for example, with probability density function

$$f_X(x) = Ke^{-|x/\beta|^\nu}, \qquad K = \text{constant} \tag{18.50}$$

   is sub-Gaussian when $\nu > 2$. For the limiting case of $\nu = \infty$, the probability density function of Eq. (18.50) reduces to that of a uniformly distributed random variable. Also, by choosing $\beta = \sqrt{3}$, we have $E[x^2] = 1$. Thus, the probabilistic model assumed in Eq. (18.6) satisfies the first part of the Benveniste–Goursat–Ruget theorem.

2. The second part of the theorem is also satisfied by the Bussgang algorithm, since we have

$$\frac{\partial^2 \psi}{\partial y^2} < 0 \qquad \text{for } 0 < y < \infty \tag{18.51}$$

   This is readily verified by examining the curves plotted in Fig. 18.7.

The Benveniste–Goursat–Ruget theorem exploited in this proof is based on the assumption of a doubly infinite equalizer. Unfortunately, this assumption breaks down in practice as we have to work with a finitely parameterized equalizer. To date, no zero-memory nonlinear function $g(\cdot)$ is known, which would result in global convergence of the

[4] Note that the function $\psi(y)$ defined in Eq. (18.49) is the negative of that defined in Benveniste et al. (1980).

blind equalizer in Fig. 18.5 to the inverse of the unknown channel (Verdu, 1984; Johnson, 1991). The global convergence of the Bussgang algorithm for an arbitrarily large but finite filter length remains an open problem. Nevertheless, there is practical evidence, supported by convergence analysis presented in Li and Ding (1995), for the conjecture that the Bussgang algorithm will converge to a desired global minimum if the transversal equalizer is long enough and initialized with a nonzero center tap, e.g., $\hat{w}_0(0) = 1$ in Fig. 18.4.

### Decision-Directed Algorithm

When the Bussgang algorithm has converged and the eye pattern appears "open," the equalizer should be switched smoothly to the *decision-directed mode* of operation, and minimum mean-squared-error control of the tap weights of the transversal filter component in the equalizer is exercised, as in a conventional adaptive equalizer.

Figure 18.8 presents a block diagram of the equalizer operating in its decision-directed mode. The only difference between this mode of operation and that of blind equalization lies in the type of zero-memory nonlinearity employed. Specifically, the conditional mean estimation of the blind equalizer in Fig. 18.5 is replaced by a *threshold decision device*. Given the observation $y(n)$, that is, the equalized signal at the transversal filter output, the threshold device makes a *decision in favor of a particular value in the known alphabet of the transmitted data sequence that is closest to* $y(n)$. We may thus write

$$\hat{x}(n) = \text{dec}(y(n)) \tag{18.52}$$

For example, in the simple case of an *equiprobable binary data sequence*, the data levels and decision levels are as follows, respectively:

$$x(n) = \begin{cases} +1 & \text{for symbol 1} \\ -1 & \text{for symbol 0} \end{cases} \tag{18.53}$$

and

$$\text{dec}((y(n)) = \text{sgn}(y(n)) \tag{18.54}$$

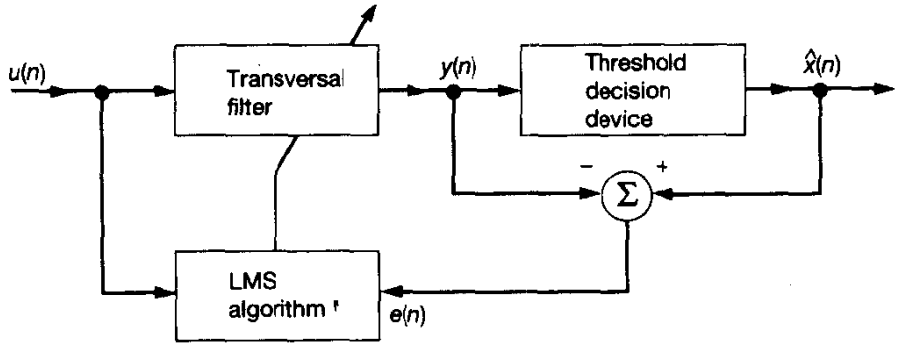



**Figure 18.8** Block diagram of decision-directed mode of operation.

where sgn(•) is the *signum function* equal to +1 if the argument is positive, and −1 if it is negative.

The equations that govern the operation of the decision-directed algorithm are the same as those of the Bussgang algorithm, except for the use of Eq. (18.52) in place of (18.16). Herein lies an important practical advantage of a blind equalizer that is based on the Bussgang algorithm and incorporates the decision-directed algorithm: Its implementation is only slightly more complex than that of a conventional adaptive equalizer, yet it does not require the use of a training sequence.

Suppose that the following conditions are satisfied:

1. The eye pattern is open (which it should be on the completion of blind equalization).
2. The step-size parameter $\mu$ used in the LMS implementation of the decision-directed algorithm is fixed (which is a common practice).
3. The sequence of observations at the channel output, denoted by the vector $\mathbf{u}(n)$, is *ergodic* in the sense that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbf{u}(n)\mathbf{u}^T(n) \to E[\mathbf{u}(n)\mathbf{u}^T(n)] \qquad \text{almost surely} \tag{18.55}$$

Then, under these conditions, *the tap-weight vector in the decision-directed algorithm converges to the optimum (Wiener) solution in the mean-square sense* (Macchi and Eweda, 1984). This is a powerful result, making the decision-directed algorithm an important adjunct of the Bussgang algorithm for blind equalization in digital communications.

## 18.3 EXTENSION OF BUSSGANG ALGORITHMS TO COMPLEX BASEBAND CHANNELS

Thus far we have only discussed the use of Bussgang algorithms for the blind equalization of $M$-ary PAM systems, characterized by a real baseband channel. In this section we extend the use of this family of blind equalization algorithms to *quadrature-amplitude modulation (QAM)* systems that involve a hybrid combination of amplitude and phase modulations.

In the case of a complex baseband channel, the transmitted data sequence $x(n)$, the channel impulse response $h_n$, and the received signal $u(n)$ are all complex valued. We may thus write

$$x(n) = x_I(n) + jx_Q(n) \tag{18.56}$$

$$h_n = h_{I,n} + jh_{Q,n} \tag{18.57}$$

and

$$u(n) = u_I(n) + ju_Q(n) \tag{18.58}$$

**TABLE 18.1 SUMMARY OF BUSSGANG ALGORITHMS FOR BLIND EQUALIZATION OF COMPLEX BASEBAND CHANNELS**

*Initialization:* Set

$$\hat{w}_i(0) = \begin{cases} 1, & i = 0 \\ 0, & i = \pm 1, \ldots, \pm L \end{cases}$$

*Computation:* $n = 1, 2, \ldots$

$$y(n) = y_I(n) + jy_Q(n)$$

$$= \sum_{i=-L}^{L} \hat{w}_i^*(n)u(n-i)$$

$$\hat{x}(n) = \hat{x}_I(n) + j\hat{x}_Q(n)$$

$$= g(y_I(n)) + jg(y_Q(n))$$

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

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e^*(n), \qquad i = 0, \pm 1, \ldots, \pm L$$

where the subscripts $I$ and $Q$ refer to the *in-phase* (*real*) and *quadrature* (*imaginary*) *components*, respectively. Correspondingly, the conditional mean estimate of the complex datum $x(n)$, given the observation $y(n)$ at the transversal filter output, is written as

$$\begin{aligned} \hat{x}(n) &= E[x(n)|y(n)] \\ &= \hat{x}_I(n) + j\hat{x}_Q(n) \\ &= g(y_I(n)) + jg(y_Q(n)) \end{aligned} \tag{18.59}$$

where $g(\cdot)$ describes a zero-memory nonlinearity. Equation (18.59) states that the in-phase and quadrature components of the transmitted data sequence $x(n)$ may be estimated separately from the in-phase and quadrature components of the transversal filter output $y(n)$, respectively. Note, however, that the conditional mean $E[x(n)|y(n)]$ can only be expressed as in Eq. (18.59) if the data transmitted in the in-phase and quadrature channels are statistically independent of each other, which is usually the case.

Clearly, Bussgang algorithms for complex baseband channels include the corresponding algorithms for real baseband channels as a special case. Table 18.1 presents a summary of Bussgang algorithms for a complex baseband channel.

## 18.4 SPECIAL CASES OF THE BUSSGANG ALGORITHM

The Bussgang algorithm discussed in Sections 18.2 and 18.3 is of a general formulation, in that it includes a number of blind equalization algorithms as special cases. Two special cases of the Bussgang algorithm are considered in the sequel.

### Sato Algorithm

The idea of blind equalization in $M$-ary PAM systems dates back to the pioneering work of Sato (1975). The *Sato algorithm* consists of minimizing a *nonconvex* cost function

$$J(n) = E[(\hat{x}(n) - y(n))^2] \tag{18.60}$$

where $y(n)$ is the transversal filter output defined in Eq. (18.12), and $\hat{x}(n)$ is an estimate of the transmitted datum $x(n)$. This estimate is obtained by a zero-memory nonlinearity described as follows:

$$\hat{x}(n) = \gamma \operatorname{sgn}[y(n)] \tag{18.61}$$

The constant $\gamma$ sets the *gain* of the equalizer; it is defined by

$$\gamma = \frac{E[x^2(n)]}{E[|x(n)|]} \tag{18.62}$$

It is apparent that the Sato algorithm is a special (nonoptimal) case of the Bussgang algorithm, with the nonlinear function $g(y)$ defined by

$$g(y) = \gamma \operatorname{sgn}(y) \tag{18.63}$$

where sgn $(\cdot)$ is the signum function. The nonlinearity defined in Eq. (18.63) is similar to that in the decision-directed algorithm for binary PAM, except for the data-dependent gain factor $\gamma$.

The Sato algorithm for blind equalization was introduced originally to deal with one-dimensional multilevel ($M$-ary PAM) signals, with the objective of being more robust than a decision-directed algorithm. Initially, the algorithm treats such a digital signal as a "binary" signal by estimating the most significant bit; the remaining bits of the signal are treated by the algorithm as additive noise insofar as the blind equalization process is concerned. The algorithm then uses the results of this preliminary step to modify the error signal obtained from a conventional decision-directed algorithm.

The Benveniste–Goursat–Ruget theorem for convergence holds for the Sato algorithm even though the nonlinear function $\psi(\cdot)$ is not differentiable for it. According to this theorem, global convergence of the Sato algorithm can be achieved provided that the probability density function of the transmitted data sequence can be approximated by a sub-Gaussian function such as the uniform distribution [Benveniste et al. (1980)]. However, global convergence of the Sato algorithm holds only for the limiting case of a doubly infinite equalizer. Deviations from this ideal behavior have been reported in the literature:

- In Mazo (1980), Verdu (1984), and Macchi and Eweda (1985), it is shown that the Sato algorithm exhibits local minima for discrete QAM input signals.
- In Ding et al. (1989), it is shown that for finitely parameterized equalizers the Sato algorithm may converge to local minima for both discrete and sub-Gaussian inputs.

### Godard Algorithm

Godard (1980) was the first to propose a family of *constant modulus* blind equalization algorithms for use in two-dimentional digital communication systems (e.g., $M$-ary phase-shift keying). Specifically, the *Godard algorithm* minimizes a nonconvex cost function of the form

$$J(n) = E[(|y(n)|^p - R_p)^2] \tag{18.64}$$

where $p$ is a positive integer, and $R_p$ is a positive real constant defined by

$$R_p = \frac{E[|x(n)|^{2p}]}{E[|x(n)|^p]} \tag{18.65}$$

The Godard algorithm is designed to penalize deviations of the blind equalizer output $x(n)$ from a constant modulus. The constant $R_p$ is chosen in such a way that the gradient of the cost function $J(n)$ is zero when perfect equalization [i.e., $\hat{x}(n) = x(n)$] is attained.

The tap-weight vector of the equalizer is adapted in accordance with the stochastic gradient algorithm (Godard, 1980)

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) e^*(n) \tag{18.66}$$

where $\mu$ is the step-size parameter, $\mathbf{u}(n)$ is the tap-input vector, and $e(n)$ is the error signal defined by

$$e(n) = y(n)|y(n)|^{p-2}(R_p - |y(n)|^p) \tag{18.67}$$

From the definition of the cost function $J(n)$ in Eq. (18.64) and from Eq. (18.67), we see that the qualizer adaptation according to the Godard algorithm does not require carrier phase recovery. The algorithm therefore tends to converge slowly. However, it offers the advantage of decoupling the ISI equalization and carrier phase recovery problems from each other.

Two cases of the Godard algorithm are of specific interest:

**Case 1: $p = 1$** The cost function of Eq. (18.64) for this case reduces to

$$J(n) = E[(|y(n)| - R_1)^2] \tag{18.68}$$

where

$$R_1 = \frac{E[|x(n)|^2]}{E[|x(n)|]} \tag{18.69}$$

This case may be viewed as a modification of the Sato algorithm.

**Case 2: $p = 2$** In this case, the cost function of Eq. (18.64) reduces to

$$J(n) = E[(|y(n)|^2 - R_2)^2] \tag{18.70}$$

where

$$R_2 = \frac{E[|x(n)|^4]}{E[|x(n)|^2]} \tag{18.71}$$

This second special case is referred to in the literature as the *constant modulus algorithm* (*CMA*).[5]

The Godard algorithm is considered to be the most successful among the Bussgang family of blind equalization algorithms, as demonstrated by the comparative studies reported in Shynk et al. (1991) and Jablon (1992). In particular, we may say the following (Papadias, 1995):

- The Godard algorithm is more robust than other Bussgang algorithms with respect to carrier phase offset. This important property of the Godard algorithm is due to the fact that the cost function used for its derivation is based solely on the amplitude of the received signal.
- Under steady-state conditions, the Godard algorithm attains a mean-squared error that is lower than other Bussgang algorithms.
- Last but by no means least, the Godard algorithm is often able to equalize a dispersive channel, such that the eye pattern is opened up when it is initially closed for all practical purposes.

### Summary of Special Forms of the Bussgang Algorithm

The decision-directed, Sato, and Godard algorithms may be viewed as special cases of the Bussgang algorithm [Bellini (1986)]. In particular, we may use Eqs. (18.52), (18.61), and (18.67) to set up the entries shown in Table 18.2 for the special forms of the zero-memory nonlinear function $g(\cdot)$ pertaining to these three algorithms [Hatzinakos (1990)]. The entries for the decision-directed and Sato algorithms follow directly from the definition

$$\hat{x}(n) = g(y(n))$$

In the case of the Godard algorithm, we note that

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

or, equivalently,

$$g(y(n)) = y(n) + e(n)$$

[5] The constant modulus algorithm (CMA) was so named by Treichler and Agee (1983), independently of Godard's 1980 paper. It is probably the most widely investigated blind equalization algorithm and the one most widely used in practice (Treichler and Larimore, 1985a, b; Smith and Friedlander, 1985; Johnson et al., 1988).

**TABLE 18.2** SPECIAL CASES OF THE BUSSGANG ALGORITHM

| Algorithm | Zero-memory nonlinear function $g(\cdot)$ | Definitions |
|---|---|---|
| Decision-directed* | $\text{sgn}(\cdot)$ | |
| Sato | $\gamma\ \text{sgn}(\cdot)$ | $\gamma = \dfrac{E[x^2(n)]}{E[\lvert x(n)\rvert]}$ |
| Godard | $\dfrac{y(n)}{\lvert y(n)\rvert}\left(\lvert y(n)\rvert + R_p\lvert y(n)\rvert^{p-1} - \lvert y(n)\rvert^{2p-1}\right)$ | $R_p = \dfrac{E[\lvert x(n)\rvert^{2p}]}{E[\lvert x(n)\rvert^{p}]}$ |

*The zero-memory nonlinear function sgn $(\cdot)$ for the decision-directed algorithm applies if the input data are binary; for the general case of $M$-ary PAM, an $M$-ary slicer is required.

Hence, we may use this relation and Eq. (18.67) to derive the special forms of the Godard algorithm in Table 18.2.

## 18.5 BLIND CHANNEL IDENTIFICATION AND EQUALIZATION USING POLYSPECTRA

The Bussgang algorithm uses the higher-order statistics of the received signal in an implicit sense. We now describe another class of blind deconvolution algorithm, which uses the higher-order statistics of the received signal in an explicit sense. For convenience of presentation, we restrict the discussion to real-valued stochastic processes.

From Chapter 3 we recall that the higher-order statistics of a stationary stochastic process are described in terms of the *cumulants* and their Fourier transforms known as *polyspectra*. Indeed, cumulants and polyspectra may be viewed as generalizations of the autocorrelation function and power spectrum, respectively. Polyspectra provide the basis for the identification (and therefore blind equalization) of a nonminimum-phase channel by virtue of their ability to preserve phase information in the channel output.

Consider then the system model described in Section 18.2 for the baseband transmission of a data sequence $x(n)$ using $M$-ary modulation. The probabilistic model of the sequence $x(n)$ is as described in Eqs. (18.4) to (18.6). We assume that the FIR channel transfer function $H(z)$ admits the following factorization, under the premise that $H(z)$ has no zeros on the unit circle:

$$H(z) = kI(z)O(z^{-1}) \tag{18.72}$$

where $k$ is a scaling factor, $I(z)$ is a *minimum-phase polynomial*, and $O(z^{-1})$ is a *maximum-phase polynomial*. The polynomial $I(z)$ has all its zeros inside the unit circle in the $z$-plane, as shown by

$$I(z) = \prod_{l=1}^{L_1} (1 - a_l z^{-1}), \qquad |a_l| < 1 \tag{18.73}$$

The second polynomial $O(z)$ has all its zeros outside the unit circle, as shown by

$$O(z^{-1}) = \prod_{l=1}^{L_2} (1 - b_l z), \qquad |b_l| < 1 \tag{18.74}$$

According to the representation described in Eqs. (18.72) to (18.74), the channel is characterized by a finite-(length) impulse response and nonminimum-phase transfer function.

For a data sequence $x(n)$ having a symmetric uniform distribution, as described in the probabilistic model of Eq. (18.6), we have

$$E[x(n)] = 0$$

$$E[x^2(n)] = 1$$

$$E[x^3(n)] = 0$$

$$E[x^4(n)] = 9/5$$

Correspondingly, the *skewness* of $x(n)$ is $\gamma_3 = 0$, and its *kurtosis* is

$$\gamma_4 = E[x^4(n)] - 3(E[x^2(n)])^2$$

$$= \frac{9}{5} - 3 = -\frac{6}{5}$$

With $\gamma_3 = 0$, it follows that the third-order cumulant of the channel output $u(n)$ is identically zero. On the other hand, $\gamma_4$ has a nonzero value; we may therefore work in the fourth-order cumulant domain as a basis for blind equalization.

### Tricepstrum

Let $c_4(\tau_1, \tau_2, \tau_3)$ denote the fourth-order cumulant of the channel output $u(n)$. We may express the trispectrum of $u(n)$ as

$$C_4(\omega_1, \omega_2, \omega_3) = F[c_4(\tau_1, \tau_2, \tau_3)] \tag{18.75}$$

where $F[\cdot]$ denotes three-dimensional discrete Fourier transformation. Define

$$\kappa_4(\tau_1, \tau_2, \tau_3) = F^{-1}[\ln C_4(\omega_1, \omega_2, \omega_3)] \tag{18.76}$$

where ln signifies the natural logarithm, and $F^{-1}$ signifies inverse three-dimensional discrete Fourier transformation. The quantity $\kappa_4(\tau_1, \tau_2, \tau_3)$ is called the *complex cepstrum of trispectrum* or *tricepstrum* of the process $u(n)$ (Pan and Nikias, 1988; Hatzinakos and Nikias, 1989, 1991).

When a linear time-invariant system (channel) characterized by impulse response $h_n$ is excited by a process $x(n)$ consisting of *iid* random variables, the fourth-order cumulant of the resulting output $u(n)$ is defined by (see section 3.7)

$$c_4(\tau_1, \tau_2, \tau_3) = \gamma_4 \sum_{i=0}^{\infty} h_i h_{i+\tau_1} h_{i+\tau_2} h_{i+\tau_3} \tag{18.77}$$

Note that the relation of Eq. (18.77) holds even if the linear system (channel) includes additive white Gaussian noise at its output, which is typically the case in a communications environment. In any event, taking the three-dimensional discrete Fourier transforms of both sides of Eq. (18.77), we get

$$C_4(\omega_1, \omega_2, \omega_3) = \gamma_4 H(e^{j\omega_1}) H(e^{j\omega_2}) H(e^{j\omega_3}) H(e^{-j(\omega_1+\omega_2+\omega_3)}) \tag{18.78}$$

Next, taking the natural logarithm of both sides of Eq. (18.78), we get

$$\begin{aligned} \ln C_4(\omega_1, \omega_2, \omega_3) = {} & \ln \gamma_4 + \ln H(e^{j\omega_1}) + \ln H(e^{j\omega_2}) + \ln H(e^{j\omega_3}) \\ & + \ln H(e^{-j(\omega_1+\omega_2+\omega_3)}) \end{aligned} \tag{18.79}$$

The channel transfer function $H(z)$ is defined by Eqs. (18.72) to (18.74); hence, we have

$$\begin{aligned} \ln H(e^{j\omega_i}) &= \ln k + \ln I(e^{j\omega_i}) + \ln O(e^{-j\omega_i}) \\ &= \ln k + \sum_{l=1}^{L_1} \ln(1 - a_l e^{-j\omega_i}) + \sum_{l=1}^{L_2} \ln(1 - b_l e^{+j\omega_i}), \qquad i = 1, 2, 3 \end{aligned} \tag{18.80}$$

and

$$\begin{aligned} \ln H(e^{-j(\omega_1+\omega_2+\omega_3)}) &= \ln k + \ln I(e^{-j(\omega_1+\omega_2+\omega_3)}) + \ln O(e^{j(\omega_1+\omega_2+\omega_3)}) \\ &= \ln k + \sum_{l=1}^{L_1} \ln(1 - a_l e^{j(\omega_1+\omega_2+\omega_3)}) + \sum_{l=1}^{L_2} \ln(1 - b_l e^{-j(\omega_1+\omega_2+\omega_3)}) \end{aligned} \tag{18.81}$$

Thus, returning to Eq. (18.79) and taking the inverse three-dimensional discrete Fourier transform of $\ln C_4(\omega_1, \omega_2, \omega_3)$, we find that the tricepstrum has the following form:[6]

[6] To evaluate $\kappa_4(\tau_1, \tau_2, \tau_3)$, we may use the inversion formula for the three-dimensional $z$-transform:

$$\kappa_4(\tau_1, \tau_2, \tau_3) = \frac{1}{(2\pi j)^3} \oint_{\mathscr{C}_3} \oint_{\mathscr{C}_2} \oint_{\mathscr{C}_1} \ln C_4(z_1, z_2, z_3) z_1^{\tau_1 - 1} z_2^{\tau_2 - 1} z_3^{\tau_3 - 1} \, dz_1 \, dz_2 \, dz_3$$

where $C_4(z_1, z_2, z_3)$ is obtained from $C_4(\omega_1, \omega_2, \omega_3)$ by substituting $z_i$ for $e^{j\omega i}$, where $i = 1, 2, 3$. The closed contours $\mathscr{C}_1$, $\mathscr{C}_2$, and $\mathscr{C}_3$ lie completely within the region of convergence of $\ln C_4(z_1, z_2, z_3)$. Let

$$\hat{a} = \max\{|a_l|\}, \qquad 1 \le l \le L_1$$

$$\hat{b} = \max\{|b_l|\}, \qquad 1 \le l \le L_2$$

$$e = \max\{\hat{a}, \hat{b}\}$$

The region of convergence for $\ln C_4(z_1, z_2, z_3)$ is defined by

$$R_c = \{|z_1| > e, |z_2| > e, |z_3| > e, \text{ and } |z_1 z_2 z_3| < 1/e\}$$

The unit surface defined by $\{|z_1| = 1, |z_2| = 1, \text{ and } |z_3| = 1\}$ lies within the region of convergence $R_c$. Accordingly, it is permissible to use the power series expansion or inversion formula to evaluate $\kappa_4(\tau_1, \tau_2, \tau_3)$.

$$\kappa_4(\tau_1, \tau_2, \tau_3) = \begin{cases} \ln k + 3 \ln \gamma_4, & \tau_1 = \tau_2 = \tau_3 = 0 \\ -\dfrac{1}{\tau_1} A^{(\tau_1)}, & \tau_1 > 0, \tau_2 = \tau_3 = 0 \\ -\dfrac{1}{\tau_2} A^{(\tau_2)}, & \tau_2 > 0, \tau_1 = \tau_3 = 0 \\ -\dfrac{1}{\tau_3} A^{(\tau_3)}, & \tau_3 > 0, \tau_1 = \tau_2 = 0 \\ \dfrac{1}{\tau_1} B^{(-\tau_1)}, & \tau_1 < 0, \tau_2 = \tau_3 = 0 \\ \dfrac{1}{\tau_2} B^{(-\tau_2)}, & \tau_2 < 0, \tau_1 = \tau_3 = 0 \\ \dfrac{1}{\tau_3} B^{(-\tau_3)}, & \tau_3 < 0, \tau_1 = \tau_2 = 0 \\ -\dfrac{1}{\tau_2} B^{(\tau_2)}, & \tau_1 = \tau_2 = \tau_3 > 0 \\ \dfrac{1}{\tau_2} A^{(\tau_2)}, & \tau_1 = \tau_2 = \tau_3 < 0 \\ 0, & \text{otherwise} \end{cases} \tag{18.82}$$

where

$$A^{(m)} = \sum_{l=1}^{L_1} a_l^m \tag{18.83}$$

and

$$B^{(m)} = \sum_{l=1}^{L_2} b_l^m \tag{18.84}$$

The $A^{(m)}$ and $B^{(m)}$ contain *minimum-phase* and *maximum-phase* information about the channel, respectively; that is, they correspond to $I(z)$ and $O(z^{-1})$, respectively.

The *differential cepstrum paramaters* $A^{(m)}$ and $B^{(m)}$ exhibit the following properties (Hatzinakos and Nikias, 1994):

1. The differential cepstrum parameters decay exponentially at least as fast as (for positive integer $m$)

$$|A^{(m)}| < c_1 \alpha^m$$

and

$$|B^{(m)}| < c_2 \beta^m$$

where $\max|a_l| < \alpha < 1$ and $\max|b_l| < \beta < 1$ and $c_1$ and $c_2$ are constants.

2. The tricepstrum is invariant under a time shift (i.e., a linear phase shift).
3. Let the *minimum-phase time sequence* $i(n)$ denote the inverse $z$-transform of the polynomial $I(z)$. Then, $i(n)$ is related to the corresponding differential cepstrum parameter $A^{(m)}$ by

$$i(n) = -\frac{1}{n}\sum_{m=1}^{n} A^{(m)} i(n-m), \qquad n = 1, 2, \ldots, L_1 \tag{18.85}$$

For other values of $n$, we have

$$i(n) = \begin{cases} 1, & n = 0 \\ 0, & n < 0 \end{cases} \tag{18.86}$$

Next, let the *maximum-phase time sequence* $o(n)$ denote the inverse $z$-transform of the polynomial $O(z^{-1})$. Then, $o(n)$ is related to the corresponding differential cepstrum parameter $B^{(m)}$ by

$$o(n) = \frac{1}{n}\sum_{m=n}^{-1} B^{(-m)} o(n-m), \qquad n = -1, -2, \ldots, -L_2 \tag{18.87}$$

For other values of $n$, we have

$$o(n) = \begin{cases} 1, & n = 0 \\ 0, & n > 0 \end{cases} \tag{18.88}$$

## Blind Channel Estimation and Equalization

The fourth-order cumulant $c_4(\tau_1, \tau_2, \tau_3)$ and the tricepstrum $\kappa_4(\tau_1, \tau_2, \tau_3)$ are related as follows (Pan and Nikias, 1988):

$$\sum_{r=-\infty}^{\infty}\sum_{s=-\infty}^{\infty}\sum_{t=-\infty}^{\infty} r\kappa_4(r, s, t) c_4(\tau_1 - r, \tau_2 - s, \tau_3 - t) = -\tau_1 c_4(\tau_1, \tau_2, \tau_3) \tag{18.89}$$

The linear convolution formula of Eq. (18.89) is of fundamental importance to the solution of the blind channel estimation/equalization problem. Specifically, substituting Eq. (18.82) into (18.89), we obtain (after some algebra) the following *tricepstral equation*:

$$\begin{aligned}\sum_{m=1}^{p} &\left(A^{(m)}[c_4(\tau_1 - m, \tau_2, \tau_3) - c_4(\tau_1 + m, \tau_2 + m, \tau_3 + m)]\right) \\ &+ \sum_{m=1}^{q} \left(B^{(m)}[c_4(\tau_1 - m, \tau_2 - m, \tau_3 - m) - c_4(\tau_1 + m, \tau_2, \tau_3)]\right) \\ &= -\tau_1 c_4(\tau_1, \tau_2, \tau_3)\end{aligned} \tag{18.90}$$

In theory, the parameters $p$ and $q$ are infinitely large. In practice, however, they can both be approximated by finite (arbitrarily large) values, because $A^{(m)}$ and $B^{(m)}$ decay exponen-

tially as $m$ increases (Hatzinakos and Nikias, 1991, 1994). Assuming that suitable values have been assigned to $p$ and $q$, we may define

$$\alpha_1 = \max(p, q)$$

$$\alpha_2 \leq \frac{\alpha_1}{2}$$

$$\alpha_3 \leq \alpha_2$$

and choose

$$\tau_1 = -\alpha_1, \ldots, -1, 1, \ldots, \alpha_1$$

$$\tau_2 = -\alpha_2, \ldots, 0, \ldots, \alpha_2$$

$$\tau_3 = -\alpha_3, \ldots, 0, \ldots, \alpha_3$$

Let

$$w = 2\alpha_1(2\alpha_2 + 1)(2\alpha_3 + 1) \tag{18.91}$$

Accordingly, we may use Eq. (18.90) to construct the following overdetermined linear system of equations:

$$\mathbf{Ca} = \mathbf{p} \tag{18.92}$$

where the known quantities $\mathbf{C}$ and $\mathbf{p}$ and the unknown $\mathbf{a}$ are defined as follows:

1. The matrix $\mathbf{C}$ is a $w$-by-$(p + q)$ matrix with entries of the form $\{c_4(\tau_1, \tau_2, \tau_3) - c_4(\tau'_1, \tau'_2, \tau'_3)\}$; the dimension $w$ is itself defined in Eq. (18.91).
2. The vector $\mathbf{p}$ is a $w$-by-1 vector with entries of the form $\{-\tau_1 c_4(\tau_1, \tau_3, \tau_3)\}$.
3. The vector $\mathbf{a}$ is a $(p + q)$-by-1 coefficient vector defined in terms of the $A^{(m)}$ and the $B^{(m)}$ by

$$\mathbf{a} = [A^{(1)}, A^{(2)}, \ldots, A^{(p)}, B^{(1)}, B^{(2)}, \ldots, B^{(q)}]^T \tag{18.93}$$

Our main purpose is to formulate a *zero-forcing blind equalization algorithm*. The structural form of the algorithm is depicted in Fig. 18.9, which consists of two major components: a *channel estimator*, and a *channel equalizer*. Accordingly, the algorithm proceeds in two stages as follows (Hatzinakos and Nikias, 1994):

1. *Channel estimation.* Let $\hat{\mathbf{C}}$ and $\hat{\mathbf{p}}$ denote estimates of the matrix $\mathbf{C}$ and the vector $\mathbf{p}$, respectively; these estimates are themselves derived from $\hat{c}_4(\tau_1, \tau_2, \tau_3)$, a time-averaged estimate of the fourth-order cumulant that is obtained from a finite-window length of the channel output $u(n)$. Then, given $\hat{\mathbf{C}}$ and $\hat{\mathbf{p}}$, we may use the pseudoinverse matrix of $\hat{\mathbf{C}}$ to solve Eq. (18.92) for $\hat{\mathbf{a}}$ that denotes an estimate of the vector $\mathbf{a}$. The elements of $\hat{\mathbf{a}}$ define estimates of the differential cepstrum

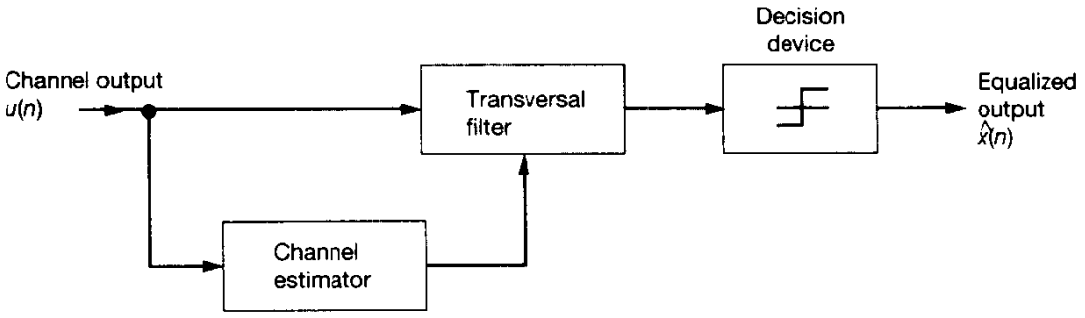



**Figure 18.9** Block diagram of tricepstrum-based blind equalizer.

paramters $A^{(m)}$ and $B^{(m)}$; these estimates are denoted by $\hat{A}^{(m)}$ and $\hat{B}^{(m)}$, respectively.

2. *Channel equalization.* Using the estimates $\hat{A}^{(m)}$, and $\hat{B}^{(m)}$ of the differential cepstrum parameters in Eqs. (18.85) and (18.87), respectively, corresponding estimates of the minimum-phase sequence $i(n)$ and the maximum-phase sequence $o(n)$ are computed. Let these estimates be denoted by $\hat{i}(n)$ and $\hat{o}(n)$, respectively. Then, the impulse response of the transversal equalizer of total length $L_1 + L_2$ is obtained by convolving the inverses of $\hat{i}(n)$ and $\hat{o}(n)$. The resulting equalizer design is an approximation to the zero-forcing condition, under which the transfer function of the equalizer is the inverse of the transfer function of the channel.

The computation described under point 1 is made using block estimation approach. Alternatively, we may use an adaptive estimation approach of the LMS type. In the latter case, the recommended procedure is to permit the adaptive process to converge (i.e., reach a steady state) before proceeding with stage 2 of the estimation procedure.

## 18.6 ADVANTAGES AND DISADVANTAGES OF HOS-BASED DECONVOLUTION ALGORITHMS

The implicit HOS-based blind deconvolution algorithm, exemplified by Bussgang algorithms, are relatively simple to implement and generally capable of delivering good performance, as evidenced by their use in line-of-sight digital radio systems. However, they suffer from some basic limitations: (1) a potential for converging to a local minimum (Ding et al., 1991), and (2) sensitivity to timing jitter (Qureshi, 1985). In contrast, explicit HOS-based blind deconvolution algorithms, such as those that use the tricepstrum, overcome the local minimum problem by avoiding the need for minimizing a cost function, but they are computationally much more complex.

Perhaps the most serious limitation of both implicit and explicit HOS-based blind deconvolution algorithms is their slow rate of convergence (Ding, 1994). To appreciate the reason for this poor behavior, we have to recognize that time-average estimation of higher-

order statistics requires a much larger sample size than is the case for second-order statistics. According to Brillinger (1975), the sample size needed to estimate the $n$th-order statistics of a stochastic process, subject to prescribed values of estimation bias and variance, increases almost exponentially with order $n$. Now, from our discussion of the tricepstrum-based deconvolution method, we know that channel identification/equalization requires at least fourth-order statistics; a similar remark also applies to Bussgang algorithms. It is therefore not surprising to find that HOS-based blind deconvolution algorithms exhibit a slow rate of convergence, compared to conventional adaptive filtering algorithms that rely on a training sequence for their operation. Thus, whereas a conventional adaptive filtering algorithm may require a few hundred iterations to converge, an existing HOS-based blind deconvolution algorithm may require several thousand iterations to converge.

The slow rate of convergence of HOS-based blind deconvolution algorithms is of no serious concern in some applications such as seismic deconvolution. However, in a more difficult environment such as mobile digital communications, the algorithm may simply not have enough time to reach a steady state, and may therefore be unable to track the statistical variations of the environment. Accordingly, this class of blind deconvolution algorithms cannot be used in applications where rapid acquisition is a necessary system requirement.

## 18.7 CHANNEL IDENTIFIABILITY USING CYCLOSTATIONARY STATISTICS

In HOS-based deconvolution algorithms, information about the unknown phase response of a nonminimum-phase channel is extracted by using higher-order statistics of the channel output, which is sampled at the baud rate (i.e., symbol rate). Alternatively, we may extract this phase information by exploiting another inherent characteristic of the channel output, namely, *cyclostationarity*. To explain this latter characteristic, we first rewrite the received signal in a digital communications system in its most general baseboard form as follows:

$$u(t) = \sum_{k=-\infty}^{\infty} x_k\, h(t - kT) + v(t) \tag{18.94}$$

where a symbol $x_k$ is transmitted every $T$ seconds (i.e., $1/T$ is the baud rate), and $t$ denotes continuous time; $h(t)$ is the overall impulse response of the channel (including transmit and ceive filters), and $v(t)$ is the channel noise. (The channel noise $v$ used here should not be confused with the convolutional noise $v$ used in the discussion on the Bussgang algorithm. All the quantities described in Eq. (18.94) are complex valued. Under the assumption that the transmitted sequence $x_k$ and the channel noise $v(t)$ are both wide-sense stationary with zero mean, we may readily show that the received signal $u(t)$ also has zero mean, and its autocorrelation function is *periodic* in the symbol duration $T$ (see Problem 7):

$$\begin{aligned} r_u(t_1, t_2) &= E[u(t_1)u^*(t_2)] \\ &= r_u(t_1 + T, t_2 + T) \end{aligned} \tag{18.95}$$

That is, the received signal $u(t)$ is *cyclostationary in the wide sense*.

What makes the use of cyclostationarity as the basis of an alternative approach to blind deconvolution particularly attractive is the fact that it only uses *second-order statistics*, thereby overcoming the "slow-to-converge" limitation of HOS-based algorithms.

Apparently, Gardner (1991) was the first to recognize that cyclostationary characteristics of modulated signals permit the recovery of a communication channel's amplitude and phase responses using second-order statistics only. However, the idea of blind channel identification and equalization using cyclostationary statistics is attributed to Tong et el. (1991). Indeed, the ability to solve the difficult problem of blind deconvolution on the sole basis of second-order statistics deserves to be viewed as a major technical breakthrough.

The original idea proposed by Tong et al. relies on the use of *temporal diversity* (i.e., oversampling the received signal). Ordinarily, this operation is performed in a digital communications system for the specific purpose of timing and phase recovery. However, in the context of our present discussion, the use of oversampling leads to *fractionally-spaced equalization*, which is so called because the equalization taps are spaced closer than the reciprocal of the incoming symbol rate.

Among the many fractionally-spaced blind channel identification/equalization techniques that have been proposed to date, we have picked the *subspace decomposition method*[7] described in Moulines et al. (1995). This approach bears a close relationship to the *mu*ltiple *si*gnal *c*lassification (MUSIC) algorithm originally proposed by Schmidt (1979) for angle of arrival estimation. Thus, the material presented in the next section points to the fact that much can be gained from the extensive literature on statistical array signal processing for solving the blind deconvolution problem.

## 18.8 SUBSPACE DECOMPOSITION FOR FRACTIONALLY-SPACED BLIND *IDENTIFICATION*

In what follows, we assume that the channel is modeled as an FIR filter, and several measurements are made during each sampling period $T$. The latter requirement can be satisfied in the following ways:

1. The received signal is *oversampled.*
2. *Multiple sensors* are used, with their individual outputs sampled at the symbol rate $1/T$.
3. A combination of techniques 1 and 2 is used.

The material presented in this section focuses on the first technique.

[7] Another interesting approach for blind identification is based on linear prediction theory. Such an approach was first studied by Slock (1994), and has been elaborated on by Slock (1995) and Abed Meriam et al. (1995). The basic premise of time-domain blind identification using linear prediction is presented as Problem 9.

Suppose then the received signal $u(t)$ is oversampled by setting

$$t = \frac{iT}{L} \tag{18.96}$$

where $L$ is a positive integer. Thus Eq. (18.94) takes on the discrete form

$$u\left(\frac{iT}{L}\right) = \sum_{k=-\infty}^{\infty} x_k\, h\left(\frac{iT}{L} - kT\right) + v\left(\frac{iT}{L}\right) \tag{18.97}$$

Let

$$i = nL + l, \qquad l = 0, 1, \ldots, L-1 \tag{18.98}$$

We may then rewrite Eq. (18.97) as

$$u\left(nT + \frac{lT}{L}\right) = \sum_{k=-\infty}^{\infty} x_k\, h\left((n-k)T + \frac{lT}{L}\right) + v\left(nT - \frac{lT}{L}\right) \tag{18.99}$$

For convenience of presentation, let

$$h_n^{(l)} = h\left(nT + \frac{lT}{L}\right)$$

$$u_n^{(l)} = u\left(nT + \frac{lT}{L}\right)$$

$$v_n^{(l)} = v\left(nT + \frac{lT}{L}\right)$$

Correspondingly, we may describe the oversampled channel in the simplified form

$$u_n^{(l)} = \sum_{k=-\infty}^{\infty} x_k\, h_{n-k}^{(l)} + v_n^{(l)}, \qquad l = 0, 1, \ldots, L-1 \tag{18.100}$$

With the channel modeled as an FIR filter, we may write

$$h_k^{(l)} = 0 \qquad \begin{array}{l}\text{for } k < 0 \text{ or } k > M,\\ \text{and all } l.\end{array} \tag{18.101}$$

That is, the channel is causal and has finite time support. Furthermore, we assume that, at time $n$, the processing involves the use of a transmitted signal vector consisting of $(M + N)$ symbols, as shown by

$$\mathbf{x}_n = [x_n, x_{n-1}, \ldots, x_{n-M-N+1}]^T \tag{18.102}$$

At the receiving end, we find that each block consists of $NL$ samples. Depending on how these samples are grouped together, we may distinguish two different matrix representations for the oversampled channel, as described here.

1. *Single input-multiple output (SIMO) model.* This model consists of $L$ *virtual channels* (subchannels) fed from a common input, as depicted in Fig. 18.10 (Moulines et al., 1995; Duhamel, 1995). Each virtual channel has the same time

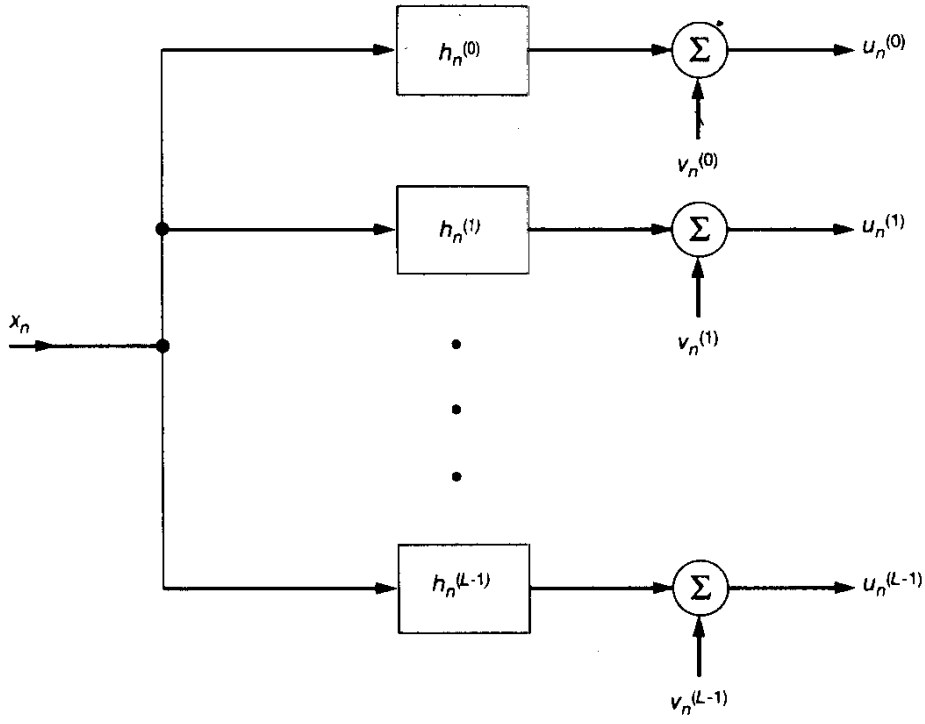



**Figure 18.10** Representation of an oversampled channel as a single input–multiple output model.

support and a noise contribution of its own. Let the $l$th virtual channel be characterized with the following:

- An $(M + 1)$-by-1 tap-weight (coefficient) vector

$$\mathbf{h}^{(l)} = [h_0^{(l)}, h_1^{(l)}, \ldots, h_M^{(l)}]^T$$

- An $N$-by-1 received signal vector

$$\mathbf{u}_n^{(l)} = [u_n^{(l)}, u_{n-1}^{(l)}, \ldots, u_{n-N+1}^{(l)}]^T$$

- An $N$-by-1 noise vector

$$\mathbf{v}_n^{(l)} = [v_n^{(l)}, v_{n-1}^{(l)}, \ldots, v_{n-N+1}^{(l)}]^T$$

We may then represent Eq. (18.100), written for $N$ successive received samples, in the compact form

$$\mathbf{u}_n^{(l)} = \mathbf{H}^{(l)}\mathbf{x}_n + \mathbf{v}_n^{(l)} \qquad l = 0, 1, \ldots, L-1 \tag{18.103}$$

where the transmitted signal vector $\mathbf{x}_n$ is defined in Eq. (18.102). The $N$-by-$(M + N)$ matrix $\mathbf{H}^{(l)}$, termed a *filtering matrix*, has a Toeplitz structure as shown by

$$\mathbf{H}^{(l)} = \begin{bmatrix} h_0^{(l)} & h_1^{(l)} & \cdots & h_M^{(l)} & 0 & \cdots & 0 \\ 0 & h_0^{(l)} & \cdots & h_{M-1}^{(l)} & h_M^{(l)} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & h_0^{(l)} & h_1^{(l)} & \cdots & h_M^{(l)} \end{bmatrix} \tag{18.104}$$

Finally, combining the set of $L$ equations (18.103) into a single relation, we may write

$$\mathcal{u}_n = \mathcal{H}\,\mathbf{x}_n + \mathcal{v}_n \tag{18.105}$$

where $\mathcal{u}_n$ is the $LN$-by-1 multichannel received signal vector

$$\mathcal{u}_n = \begin{bmatrix} \mathbf{u}_n^{(0)} \\ \mathbf{u}_n^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{u}_n^{(L-1)} \end{bmatrix}$$

and $\mathcal{v}_n$ is the $LN$-by-1 multichannel noise vector

$$\mathcal{v}_n = \begin{bmatrix} \mathbf{v}_n^{(0)} \\ \mathbf{v}_n^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{v}_n^{(L-1)} \end{bmatrix}$$

and $\mathcal{H}$ is the $LN$-by-$(M + N)$ *multichannel filtering matrix*:

$$\mathcal{H} = \begin{bmatrix} \mathbf{H}^{(0)} \\ \mathbf{H}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{H}^{(L-1)} \end{bmatrix} \tag{18.106}$$

where the individual matrix entries are themselves defined in Eq. (18.104).

2. *Sylvester matrix representation.* In this second model, the $L$ virtual channel coefficients having the same delay index are all grouped together. Specifically, we write

$$\mathbf{h}'_k = [h_k^{(0)}, h_k^{(1)}, \ldots, h_k^{(L-1)}]^T, \qquad k = 0, 1, \ldots, M$$

Correspondingly, we define an $L$-by-1 received signal vector

$$\mathbf{u}'_n = [u_n^{(0)}, u_n^{(1)}, \ldots, u_n^{(L-1)}]^T$$

and an $L$-by-1 noise vector

$$\mathbf{v}'_n = [v_n^{(0)}, v_n^{(1)}, \ldots, v_n^{(L-1)}]^T$$

Then on this basis, we may use Eq. (18.100) to group the $NL$ received samples as follows:

$$\begin{aligned} \boldsymbol{u}'_n &= \begin{bmatrix} \mathbf{u}'_n \\ \mathbf{u}'_{n-1} \\ \vdots \\ \mathbf{u}'_{n-N+1} \end{bmatrix} \\ &= \mathcal{H}'\mathbf{x}_n + \boldsymbol{v}'_n \end{aligned} \tag{18.107}$$

where the transmitted signal vector $\mathbf{x}_n$ is as previously defined in Eq. (18.102). The $LN$-by-1 noise vector $\boldsymbol{v}'_n$ is defined by

$$\boldsymbol{v}'_n = \begin{bmatrix} \mathbf{v}'_n \\ \mathbf{v}'_{n-1} \\ \vdots \\ \mathbf{v}'_{n-N+1} \end{bmatrix}$$

The $LN$-by-$(M + N)$ *matrix* $\mathcal{H}'$ *is defined by:*

$$\mathcal{H}' = \begin{bmatrix} \mathbf{h}'_0 & \mathbf{h}'_1 & \cdots & \mathbf{h}'_M & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}'_0 & \cdots & \mathbf{h}'_{M-1} & \mathbf{h}'_M & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}'_0 & \mathbf{h}'_1 & \cdots & \mathbf{h}'_M \end{bmatrix} \tag{18.108}$$

The block-Toeplitz matrix $\mathcal{H}'$ is called a *Sylvester resultant matrix* (Rosenbrock, 1970; Tong et al., 1993), hence the terminology used to refer to this second matrix representation of an oversampled channel.

### Filtering-matrix Rank Theorem

The matrices $\mathcal{H}$ and $\mathcal{H}'$, defined in Eqs. (18.106) and (18.108), respectively, differ primarily in the way in which their individual rows are arranged; they contain the same information about the channel but display it differently. Most importantly, the spaces spanned

by the columns of $\mathcal{H}$ and $\mathcal{H}'$ are *canonically equivalent*. From here on, we therefore restrict the discussion to the single input-multiple output model of Fig. 18.10.

The multichannel filtering matrix $\mathcal{H}$ plays a central role in the blind identification problem. In particular, the problem is solvable if and only if the matrix $\mathcal{H}$ is of full column rank. This requirement is covered by a crucial theorem due to Tong et al. (1993), which may be stated as follows:

- The $LN$-by-$(M + N)$ multichannel filtering matrix $\mathcal{H}$ is of full column rank, that is, $\text{rank}(\mathcal{H}) = M + N$, provided that the following three conditions are satisfied:
  1. The polynomials
  $$H^{(l)}(z) = \sum_{m=0}^{M} h_m^{(l)} z^{-m} \quad \text{for} \quad l = 0, 1, \ldots, L-1$$
  have no common zeros.
  2. At least one of the polynomials $H^{(l)}(z)$, $l = 0, 1, \ldots, L-1$, has the maximum possible degree $M$.
  3. The size $N$ of the received signal vector $\mathbf{u}_n^{(l)}$ for each virtual channel is greater than $M$.

Equipped with this theorem, hereafter referred to as the *filtering-matrix rank theorem*, we are ready to describe the subspace decomposition-based procedure for blind identification, which we do next.

### Blind Identification

The basic equation (18.106) provides a matrix description of an oversampled channel. A block diagram representation of this equation is shown in Fig. 18.11, which may be viewed as a condensed version of the single input-multiple output model of Fig. 18.10. To proceed with a statistical characterization of the channel, we make the following assumptions:

- The transmitted signal vector $\mathbf{x}_n$ and multichannel noise vector $\mathbf{v}_n$ originate from wide-sense stationary processes that are statistically independent.
- The $(M + N)$-by-1 transmitted signal vector $\mathbf{x}_n$ has zero mean and correlation matrix
$$\mathbf{R}_x = E[\mathbf{x}_n \mathbf{x}_n^H]$$

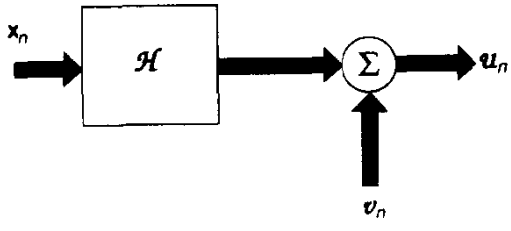



**Figure 18.11** Matrix representation of an oversampled channel.

The $(M + N)$-by-$(M + N)$ matrix $\mathbf{R}_x$ has full column rank; otherwise, it is unknown.

- The $N$-by-1 noise vector $\mathbf{v}_n$ has zero mean and correlation matrix

$$\begin{aligned}\mathbf{R}_v &= E[\mathbf{v}_n\mathbf{v}_n^H] \\ &= \sigma^2\mathbf{I}\end{aligned}$$

The noise variance $\sigma^2$ is known.

Accordingly, the $LN$-by-1 received signal vector $\boldsymbol{u}_n$ has zero mean and a correlation matrix defined by

$$\begin{aligned}\mathbf{R} &= E[\boldsymbol{u}_n\boldsymbol{u}_n^H] \\ &= E[(\boldsymbol{\mathcal{H}}\mathbf{x}_n + \boldsymbol{v}_n)(\boldsymbol{\mathcal{H}}\mathbf{x}_n + \boldsymbol{v}_n)^H] \\ &= E[\boldsymbol{\mathcal{H}}\mathbf{x}_n\mathbf{x}_n^H\boldsymbol{\mathcal{H}}^H] + E[\boldsymbol{v}_n\boldsymbol{v}_n^H] \\ &= \boldsymbol{\mathcal{H}}\mathbf{R}_x\boldsymbol{\mathcal{H}}^H + \mathbf{R}_v\end{aligned} \tag{18.109}$$

To gain some insight into the blind identification problem, we cast it in a geometrical framework first proposed by Schmidt (1979, 1981). First, we invoke the spectral theorem of Chapter 4 to describe the $LN$-by-$LN$ correlation matrix $\mathbf{R}$ in terms of its eigenvalues and associated eigenvectors as follows:

$$\mathbf{R} = \sum_{k=1}^{LN} \lambda_k\mathbf{q}_k\mathbf{q}_k^H \tag{18.110}$$

where the eigenvalues are arranged in decreasing order:

$$\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{LN-1}$$

Next, we invoke the filtering-matrix rank theorem to divide these eigervalues into two groups:

1. $\lambda_k > \sigma^2, \quad k = 0, 1, \ldots, M + N - 1$
2. $\lambda_k = \sigma^2, \quad k = M + N, M + N + 1, \ldots, LN - 1$

Correspondingly, the space spanned by the eigenvectors of matrix $\mathbf{R}$ is divided into two subspaces:

1. *Signal subspace* $\mathcal{S}$, spanned by the eigenvectors associated with the eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{M+N-1}$. These eigenvectors are written as

$$\mathbf{s}_k = \mathbf{q}_k, \quad k = 0, 1, \ldots, M + N - 1$$

2. *Noise subspace* $\mathcal{N}$, spanned by the eigenvectors associated with the remaining eigenvalues $\lambda_{M+N}, \lambda_{M+N+1}, \ldots, \lambda_{LN-1}$. These eigenvectors are written as

$$\mathbf{g}_k = \mathbf{q}_{M+N+k}, \quad k = 0, 1, \ldots, LN - M - N - 1$$

The noise subspace is the orthogonal complement of the signel subspace.

By definition, we have

$$\mathbf{R}\mathbf{g}_k = \sigma^2\mathbf{g}_k, \qquad k = 0, 1, \ldots, LN - M - N - 1 \tag{18.111}$$

Substituting Eq. (18.109), with $\mathbf{R}_v = \sigma^2\mathbf{I}$, in Eq. (18.111) and then simplifying, we get

$$\mathcal{H}\mathbf{R}_x\mathcal{H}^H\mathbf{g}_k = 0, \qquad k = 0, 1, \ldots, LN - M - N - 1$$

Since both matrices $\mathcal{H}$ and $\mathbf{R}_x$ are of full column rank, it follows that we must have

$$\mathcal{H}^H\mathbf{g}_k = \mathbf{0}, \qquad k = 0, 1, \ldots, LN - M - N - 1 \tag{18.112}$$

Equation (18.112) provides the theoretical framework of the *subspace decomposition procedure for blind identification* described in Moulines et al. (1995). Specifically, it builds on two items:

- Knowledge of the eigenvectors associated with the $LN - M - N$ smallest eigenvalues of the correlation matrix $\mathbf{R}$ of the received signal vector $u_n$.
- Orthogonality of the columns of the unknown multichannel filtering matrix $\mathcal{H}$ to the noise subspace $\mathcal{N}$.

In other words, the cyclostationary statistics of the received signal $u_n$, exemplified by the correlation matrix $\mathbf{R}$, are indeed sufficient for blind identification of the channel to within a multiplicative constant.

### Alternative Formulation of the Orthogonality Condition

From a computational point of view, we find it more convenient to work with an alternative formulation of the *orthogonality condition* described in Eq. (18.112). To begin with, we rewrite this condition in the equivalent scalar form

$$\|\mathcal{H}^H\mathbf{g}_k\|^2 = \mathbf{g}_k^H\mathcal{H}\mathcal{H}^H\mathbf{g}_k = 0, \qquad k = 0, 1, \ldots, LN - M - N - 1 \tag{18.113}$$

Recognizing the partitioned structure of the multiparameter filtering matrix $\mathcal{H}$ displayed in Eq. (18.106), in a corresponding way we may partition the $LN$-by-1 eigenvector $\mathbf{g}_k$ as follows:

$$\mathbf{g}_k = \begin{bmatrix} \mathbf{g}_k^{(0)} \\ \mathbf{g}_k^{(1)} \\ \vdots \\ \mathbf{g}_k^{(L-1)} \end{bmatrix} \tag{18.114}$$

where $\mathbf{g}_k^{(l)}$, $l = 0, 1, \ldots, L - 1$, is an $N$-by-1 vector. Next, guided by the composition of matrix $\mathbf{H}^{(l)}$ given in E (18.104), we formulate the $(M + 1)$-by-$(M + N)$ matrix:

$$\mathbf{G}_k^{(l)} = \begin{bmatrix} g_{k,0}^{(l)} & g_{k,1}^{(l)} & \cdots & g_{k,N-1}^{(l)} & 0 & \cdots & 0 \\ 0 & g_{k,0}^{(l)} & \cdots & g_{k,N-2}^{(l)} & g_{k,N-1}^{(l)} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & g_{k,0}^{(l)} & g_{k,1}^{(l)} & \cdots & g_{k,N-1}^{(l)} \end{bmatrix} \tag{18.115}$$

Finally, in light of Eq. (18.106) describing the multichannel filtering matrix $\mathcal{H}$, we use the matrices defined in Eq. (18.115) for $l = 0, 1, \ldots, L-1$ to set up the $L(M+1)$-by-$(M+N)$ matrix

$$\mathcal{Q}_k = \begin{bmatrix} \mathbf{G}_k^{(0)} \\ \mathbf{G}_k^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{G}_k^{(L-1)} \end{bmatrix}, \qquad k = 0, 1, \ldots, LN - M - N - 1 \tag{18.116}$$

Given the $\mathcal{Q}_k$ as defined here, it may be shown that (Moulines et al., 1995)

$$\mathbf{g}_k^H \mathcal{H}\, \mathcal{H}^H \mathbf{g}_k = \mathbf{h}^H \mathcal{Q}_k \mathcal{Q}_k^H \mathbf{h} \tag{18.117}$$

where $\mathbf{h}$ is an $L(M+1)$-by-1 vector defined in terms of the multichannel coefficients by

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}^{(0)} \\ \mathbf{h}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{h}^{(L-1)} \end{bmatrix}$$

Accordingly, we may reformulate the orthogonality condition of Eq. (18.113) in the equivalent form

$$\mathbf{h}^H \mathcal{Q}_k \mathcal{Q}_k^H \mathbf{h} = 0, \qquad k = 0, 1, \ldots, LN - M - N - 1 \tag{18.118}$$

which is the desired relation. In Eq. (18.118) the unknown multichannel coefficients feature in the simple form of vector $\mathbf{h}$, whereas in Eq. (18.113) they feature in the highly elaborate structure of matrix $\mathcal{H}$.

### Estimation of the Channel Coefficients

In practice, we have to work with estimates of the eigenvectors $\mathbf{g}_k$. Let these estimates be denoted by $\hat{\mathbf{g}}_k$, $k = 0, 1, \ldots, LN - M - N - 1$. To derive a corresponding estimate of the

multichannel coefficient vector $\mathbf{h}$, we use the orthogonality condition of Eq. (18.118) to define the cost function

$$\mathscr{C}(\mathbf{h}) = \mathbf{h}^H \mathcal{Z} \mathbf{h} \tag{18.119}$$

where $\mathcal{Z}$ is an $L(M+1)$-by-$L(M+1)$ matrix defined by

$$\mathcal{Z} = \sum_{k=0}^{LN-M-N-1} \hat{\mathcal{Q}}_k \hat{\mathcal{Q}}_k^H \tag{18.120}$$

The estimated matrix $\hat{\mathcal{Q}}_k$ is itself defined by Eqs. (18.115) and (18.116) with $\hat{\mathbf{g}}_k$ used in place of $\mathbf{g}_k$. In the ideal case of a true correlation matrix $\mathbf{R}$, the true multichannel coefficient vector $\mathbf{h}$ is uniquely defined (except for a multiplicative constant) by the condition $\mathscr{C}(\mathbf{h}) = 0$. Working with the matrix $\mathcal{Z}$ based on the estimates $\hat{\mathbf{g}}_k$, a least-squares estimate of the vector $\mathbf{h}$ is computed by minimizing the cost function $\mathscr{C}(\mathbf{h})$. However, this minimization would have to be performed subject to a properly chosen constraint, so as to avoid the trivial solution $\mathbf{h} = \mathbf{0}$. Moulines et al. (1995) suggest two possible optimization criteria:

1. *Linear constraint.* Minimize the cost function $\mathscr{C}(\mathbf{h})$ subject to $\mathbf{c}^H\mathbf{h} = 1$, where $\mathbf{c}$ is some $L(M+1)$-by-1 vector.
2. *Quadratic constraint.* Minimize the cost function $\mathscr{C}(\mathbf{h})$ subject to $\|\mathbf{h}\| = 1$.

The first criterion requires the prescription of an arbitrary vector $\mathbf{c}$, whereas the second criterion appears to be more natural but computationally more demanding.

A successful use of the subspace-decomposition method for blind identification rests on the premise that the transfer functions of the virtual channels have no common zeros. A test would therefore have to be performed to satisfy this requirement. Such a test would, in turn, require exact knowledge of the channel model order $M$. The important point to note here is that, given that these requirements are satisfied, it is feasible to perform the blind equlization of a communication channel using cyclostationary second-order statistics.

## 18.9 SUMMARY AND DISCUSSION

Blind deconvolution is an example of *unsupervised learning* in the sense that it identifies the inverse of an unknown linear time-invariant (possibly nonminimum-phase) system *without* having access to a training sequence (i.e., desired response). This operation requires the identification of both the magnitude and phase of the system's transfer function. To identify the magnitude component, we only need second-order statistics of the received signal (i.e., system output). However, to identify the phase component is a more difficult task.

One class of procedures for blind deconvolution relies on higher-order statistics of the received signal in an implicit or explicit sense. This, in turn, requires the use of some form of nonlinearity. Most, importantly, for higher-order statistics-based approaches to blind deconvolution to succeed, the received signal must be non-Gaussian. In this chapter, we described two such procedures, one called the Bussgang algorithm and the other the tricepstrum-based identification algorithm.

The Bussgang algorithm, using higher-order statistics in an implicit sense, performs blind equalization by subjecting the received signal to an iterative deconvolution process. When the algorithm has converged in the mean value, the deconvolved sequence assumes Bussgang statistics, hence the name of the algorithm. The distinguishing features of the Bussgang algorithm are as follows:

- The minimization of a nonconvex cost function, and therefore the potential likelihood of being trapped in a local minimum
- A low computational complexity, which is slightly greater than that of a conventional adaptive equalizer having access to a training sequence.

The tricepstrum-based blind identification algorithm explicitly exploits the inherent ability of the fourth-order cumulant of the received signal to extract phase information about the channel. This second algorithm has the following characteristics:

- Channel estimation by identifying the minimum-phase and maximum-phase parts of the channel transfer function; this is done without involving the use of a cost function and thereby avoiding the local minimum problem
- A high computational complexity

A limitation common to both of these approaches to blind channel identification and equalization, based on higher-order statistics, is a slow rate of convergence, which may inhibit their use in a difficult environment that requires rapid acquisition. This limitation may be overcome by using cyclostationary second-order statistics rather than higher-order statistics of the channel output. In this chapter, we have shown that it is indeed feasible to identify an unknown channel solely on the basis of cyclostationary statistics of the received signal, as exemplified by the subspace decomposition-based blind identification procedure. However, the use of cyclostationarity for blind identification and equalization is in its early stages of development, and its commercial use is yet to be demonstrated.

## PROBLEMS

1. Equation (18.39) defines the conditional mean estimate of the datum $x$, assuming that the convolutional noise $v$ is additive, white, Gaussian, and statistically independent of $x$. Derive this formula.

2. For perfect equalization, we require that the equalizer output $y(n)$ be exactly equal to the transmitted datum $x(n)$. Show that when the Bussgang algorithm has converged in the mean value and perfect equalization has been attained, the nonlinear estimator must satisfy the condition

$$E[\hat{x}g(\hat{x})] = 1$$

where $\hat{x}$ is the conditional mean estimate of $x$.

3. Equation (18.18) provides an adaptive method for finding the tap weights of the transversal filter in the Bussgang algorithm for performing the iterative deconvolution. Develop an alternative method for doing this computation, assuming the availability of an overdetermined system of equations and the use of the method of least squares.

4. Derive the linear convolution formula given in Eq. (18.89), which defines the relation between the fourth-order cumulant $c_4(\tau_1, \tau_2, \tau_3)$ and the trispectrum $\kappa_4(\tau_1, \tau_2, \tau_3)$.

5. Derive the tricepstral equation (18.90) that relates the fourth-order cumulant $c_4(\tau_1, \tau_2, \tau_3)$, to the $A^{(m)}$ and the $B^{(m)}$ that contain minimum-phase and maximum-phase information about the channel.

6. Formulate an adaptive estimation approach of the LMS type for solving the overdetermined system of equations (18.92). Your formulation should also include an upper bound on the step-size parameter $\mu(n)$ used in the LMS algorithm.

7. In this problem we explore the possibility of extracting the phase response of an unknown channel using cyclostationary statistics (Ding, 1993).

   (a) Using Eq. (18.94) for the received signal $u(t)$ in a digital communications system, and invoking the assumptions made in Section 18.8 on the transmitted signal $x_k$ and channel noise $v(t)$, show that the autocorrelation function of $u(t)$ evaluated at the two time instants $t_1$ and $t_2$ is given by

$$r_u(t_1, t_2) = E[u(t_1)u^*(t_2)]$$

$$= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} r_x(kT - lT)h(t_1 - kT)h^*(t_2 - lT) + \sigma_v^2 \delta(t_1 - t_2)$$

   where $r_x(kT)$ is the autocorrelation function of the transmitted signal for lag $kT$ and $\sigma_v^2$ is the noise variance. Hence, demonstrate that $u(t)$ is cyclostationary in the wide sense.

   (b) The cyclic autocorrelation function and spectral density of a cyclostationary process $u(t)$ are defined by, respectively (see Chapter 3)

$$r_u^{\alpha}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} r_u\left(t + \frac{\tau}{2}, t - \frac{\tau}{2}\right) \exp(j2\pi\alpha t) dt$$

$$S_u^{\alpha}(\omega) = \int_{-\infty}^{\infty} r_u^{\alpha}(\tau) \exp(-j2\pi f\tau) d\tau, \qquad \omega = 2\pi f$$

   where

$$\alpha = \frac{k}{T}, \qquad k = 0, \pm 1, \pm 2, \ldots$$

   Let $\Psi_k(\omega)$ denote the phase response of $S_u^{k/T}(\omega)$ and $\Phi(\omega)$ denote the phase response of the channel. Show that

$$\Psi_k(\omega) = \Phi\left(\omega + \frac{k\pi}{T}\right) - \Phi\left(\omega - \frac{k\pi}{T}\right), \qquad k = 0, \pm 1, \pm 2, \ldots$$

(c) Let $\psi_k(\tau)$ and $\phi(\tau)$ denote the inverse Fourier transforms of $\Psi_k(\omega)$ and $\Phi(\omega)$, respectively. Using the result of part (b), show that

$$\psi_k(\tau) = -2j\phi(\tau)\sin\left(\frac{\pi k\tau}{T}\right), \qquad k = 0, \pm 1, \pm 2, \ldots$$

What conclusions can you draw from this relation with regard to the possibility of extracting the phase response $\Phi(\omega)$ from $\psi_k(\tau)$?

8. Suppose that the multichannel filtering matrix $\mathcal{M}$ of the SIMO model depicted in Fig. 18.10 has been estimated using the subspace-decomposition procedure described in Section 18.8.

Show that, in the noise-free case, perfect equalization is achieved by using a multichannel structure whose own filtering matrix is defined by the pseudoinverse of $\mathcal{M}$.

9. The use of linear prediction provides the basis of other procedures for blind identification (Slock, 1994, 1995; Abed Meriam et al., 1995). The basic idea behind these procedures resides in the *generalized Bezout identity* (Kailath, 1980). Define the $L$-by-1 polynomial vector

$$\mathbf{H}(z) = [H^{(0)}(z), H^{(1)}(z), \ldots, H^{(L-1)}(z)]^T$$

where $H^{(l)}(z)$ is the transfer function of the $l$th virtual channel. Under the condition that $\mathbf{H}(z)$ is irreducible, the generalized Bezout identity states that there exists a 1-by-$L$ polynomial vector

$$\mathbf{G}(z) = [G^{(0)}(z), G^{(1)}(z), \ldots, G^{(L-1)}(z)]$$

such that

$$\mathbf{G}(z)\mathbf{H}(z) = 1,$$

that is,

$$\sum_{l=0}^{L-1} G^{(l)}(z)H^{(l)}(z) = 1$$

The implication of this identity is that a set of moving average processes described in terms of a white noise process $v(n)$ by the operation $\mathbf{y}(n) = \mathbf{H}(z)[v(n)]$ may also be represented by an autoregressive process of finite order.

Consider the ideal case of a noiseless channel, for which the received signal of the $l$th virtual channel is defined by

$$u_n^{(l)} = \sum_{m=0}^{M} h_m^{(l)} x_{n-m}, \qquad l = 0, 1, \ldots, L-1$$

where $x_n$ is the transmitted symbol, and $h_n^{(l)}$ is the impulse response of the $l$th virtual channel. Using the generalized Bezout identity, show that

$$\sum_{l=0}^{L-1} G^{(l)}(z)[u_n^{(l)}] = x_n$$

and $x_n$ is thus reproduced exactly; in this relation, $G^{(l)}(z)$ acts as an operator. How would you interpret this result in light of linear prediction?

CHAPTER

# 19

# Back-Propagation Learning

In this chapter and the next, we consider a class of *neural networks* that are quite different from the adaptive filtering structures considered in previous chapters of the book. A neural network is made up of the interconnection of a large number of nonlinear processing units referred to as *neurons*. The internal structure of the neural network may involve feedforward paths only, or feedforward as well as feedback paths. In this book we will confine our attention to the class of *feedforward* neural networks.

From a signal-processing perspective, interest in neural networks is motivated by the following important properties (Haykin, 1994):

- *Nonlinearity*. This property, attributed to the nonlinear nature of neurons in the network, is particularly useful if the underlying physical mechanism responsible for the generation of an input signal (e.g., speech signal) is inherently nonlinear.
- *Weak statistical assumptions*. A neural network relies on the availability of training data for its design; it is therefore able to capture the statistical characteristics of the environment in which it operates, provided the training data are large enough to be "representative" of the environment. In other words, a neural network permits "the dataset to speak for itself."
- *Learning*. A neural network has a built-in capability to learn from its environment by undergoing a training session for the purpose of adjusting its free parameters.

We begin our discussion of neural networks by describing the different models of a neuron that constitutes the basic processing unit of a neural network.

## 19.1 MODELS OF A NEURON

Figure 19.1 shows the *model* of an artificial neuron referred to hereafter simply as a neuron; we have labeled it as neuron $i$ for the purpose of reference. The model consists of a *linear combiner* followed by a *nonlinear unit*. The linear combiner itself consists of a set of *synaptic weights* (adjustable parameters) connected to respective input terminals, and whose weighted outputs are combined in a *summing* junction. An external bias plus the linear combiner output constitute the net input of the nonlinear unit, which is denoted by $\text{net}_i$ in Fig. 19.1.

We may distinguish four basic types of neuron models, depending on the exact description of the nonlinear unit:

1. **Linear model.** In this model the nonlinear unit is replaced by a *direct connection*, with the result that the output of the neuron is a weighted sum of its inputs. This special form of a neuron is basic to the operation of linear adaptive filters on which the material presented in Part III of this book is based.
2. **McCulloch–Pitts model.** In this second model of a neuron the nonlinear unit is characterized by a *threshold function* as depicted in Fig. 19.2(a); it is so named in recognition of the pioneering work done by McCulloch and Pitts on neural networks that dates back to 1943.
3. **Piecewise linear model.** The input–output characteristic of the nonlinear unit for this model of a neuron is described in Fig. 19.2(b). The piecewise linear model includes the linear model and the McCulloch–Pitts model as special cases. If the linear region of the input-output characteristic in Fig. 19.2(b) is made infinitely wide, we get the linear model. If, on the other hand, it is made infinitely narrow (i.e., the slope of the linear region is made infinitely large), we get the McCulloch–Pitts model.

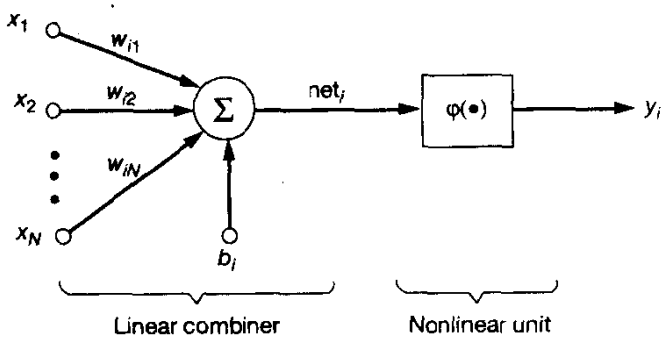



**Figure 19.1** Simplified model of a neuron.

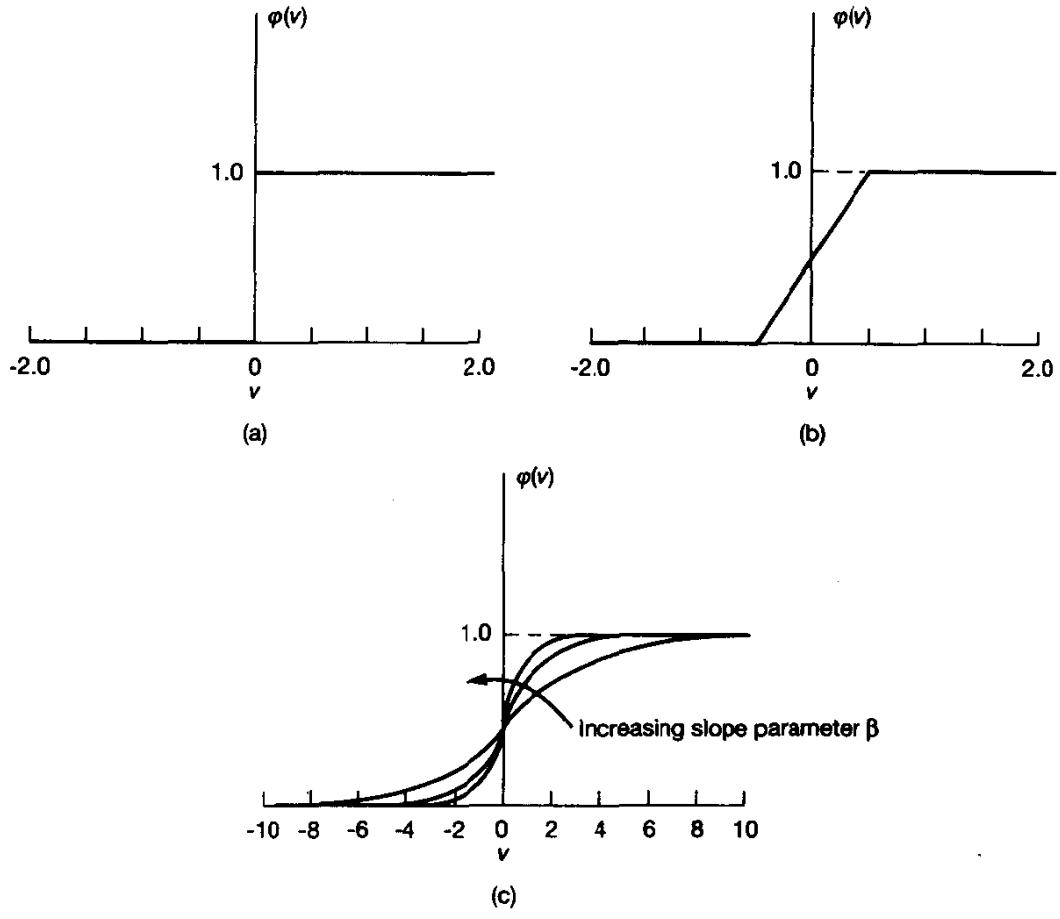



**Figure 19.2** Unipolar activation functions: (a) Threshold function; (b) piecewise linear function; (c) sigmoid function.

4. **Sigmoidal model.** This last model of a neuron is so called because the *activation function* that defines the input–output characteristic of the nonlinear unit is *S-shaped*. Let the activation function be denoted by $\varphi(\cdot)$. We may then write

$$\varphi(\text{net}) = \begin{cases} 1, & \text{net} = \infty \\ \dfrac{1}{2}, & \text{net} = 0 \\ 0, & \text{net} = -\infty \end{cases} \tag{19.1}$$

where net is the sum of the linear combiner output plus the bias. A highly popular form of sigmoidal nonlinearity is the *logistic function*, defined by

$$\varphi(\text{net}) = \frac{1}{1 + e^{-\beta\,\text{net}}} \tag{19.2}$$

where $\beta$ is the *slope parameter*. Figure 19.2(c) shows a depiction of the sigmoidal nonlinearity for varying $\beta$. The derivative of this nonlinearity with respect to its input is given by

$$\begin{aligned}\varphi'(\text{net}) &= \frac{d\varphi}{d\text{net}} \\ &= \beta\varphi(\text{net})\,(1 - \varphi(\text{net}))\end{aligned} \tag{19.3}$$

where, in the first line, the prime denotes differentiation; this practice is followed in the material that follows. The maximum slope of the logistic function of Eq. (19.2) equals $\beta/4$. When the slope parameter $\beta$ is made infinitely large, the sigmoidal model of a neuron reduces to the McCulloch–Pitts neuron.

In practical terms, the sigmoidal model of a neuron is by far the most widely used of all models for two reasons. First, it introduces a well-defined form of nonlinearity into the operation of a neuron. Second, it is *differentiable*. Indeed, the sigmoidal model of a neuron is basic to the construction of an important neural network structure called a multilayer perceptron using the back-propagation algorithm for training; more will be said on this important neural network in the next two sections.

The activation functions described in Fig. 19.2 are all of a *unipolar kind*, in that in each case the model's output is always nonnegative, regardless of the polarity of the input. Alternatively, the model's output is permitted to assume both positive and negative values, in which case the activation function is said to be of a *bipolar kind*. Figure 19.3 shows three examples of a bipolar activation function. Of particular interest is the sigmoid function shown in Fig. 19.3(c), an example of which is the *hyperbolic tangent function* defined by

$$\begin{aligned}\varphi(\text{net}_i) &= \tanh\left(\frac{1}{2}\,\text{net}_i\right) \\ &= \frac{1 - \exp(-\text{net}_i)}{1 + \exp(-\text{net}_i)}\end{aligned} \tag{19.4}$$

Another way of distinguishing between the activation functions of Fig. 19.2 and those of Fig. 19.3 is to note that the former are asymmetric, whereas the latter are antisymmetric.

**Example 1. Conditional mean estimator.**

In Section 18.2 we derived a zero-memory nonlinear estimator as an integral part of a blind equalizer of the Bussgang type. The input–output characteristic of this estimator is plotted in Fig. 18.7. A point of particular interest is that for high levels of convolutional noise, the input–output characteristic of this nonlinear estimator is closely approximated by the bipolar sigmoidal nonlinearity:

$$x = a_1 \tanh\left(\frac{a_2 y}{2}\right)$$

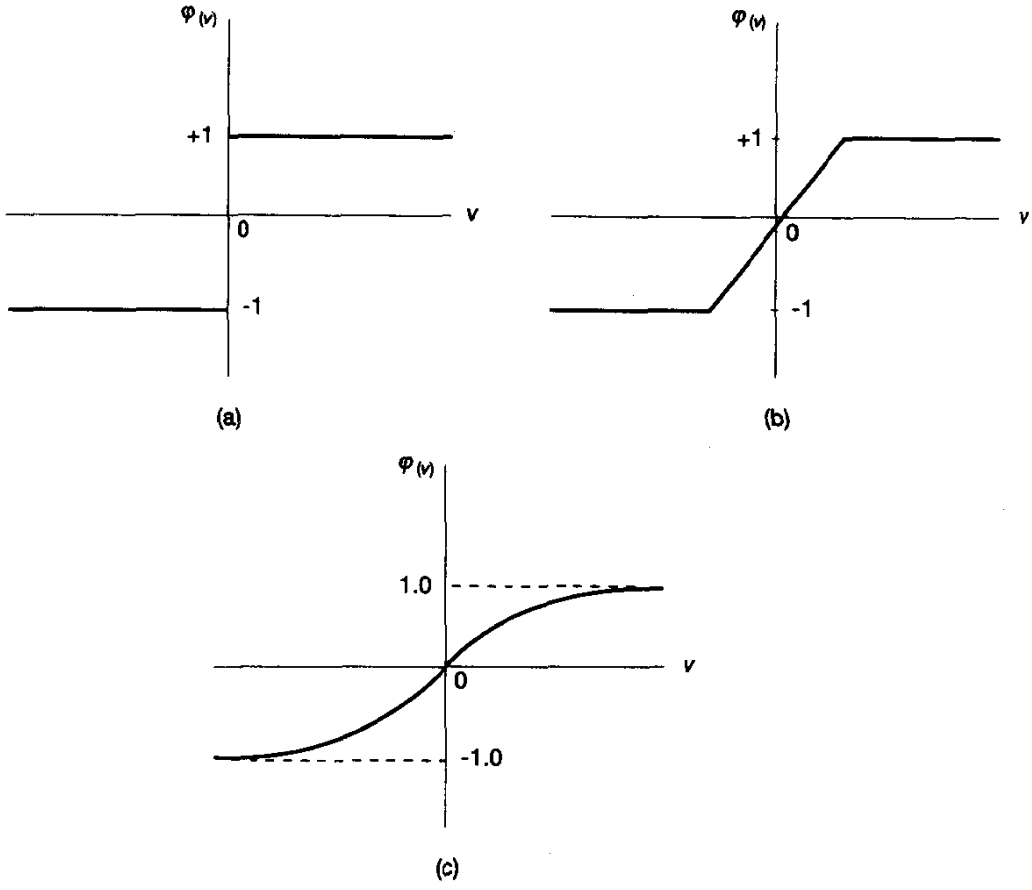



**Figure 19.3** Bipolar activation functions: (a) Threshold function; (b) piecewise linear function; (c) sigmoid function.

For the situation described in Fig. 18.7, the following values for the constants $a$ and $b$ provide a good fit:

$$a_1 = 1.945$$
$$a_2 = 1.25$$

The resulting input–output characteristic shown in Fig. 19.4.

Accordingly, we may view the blind equalizer of the Bussgang type depicted in Fig. 18.5 as being essentially a single neuron with its linear combiner and sigmoidal nonlinearity represented by the transversal filter and zero-memory nonlinear estimator, respectively. The error signal for adjusting the synaptic weights of the neuron is obtained by comparing the input and output signals of the nonlinear unit in the neuron.

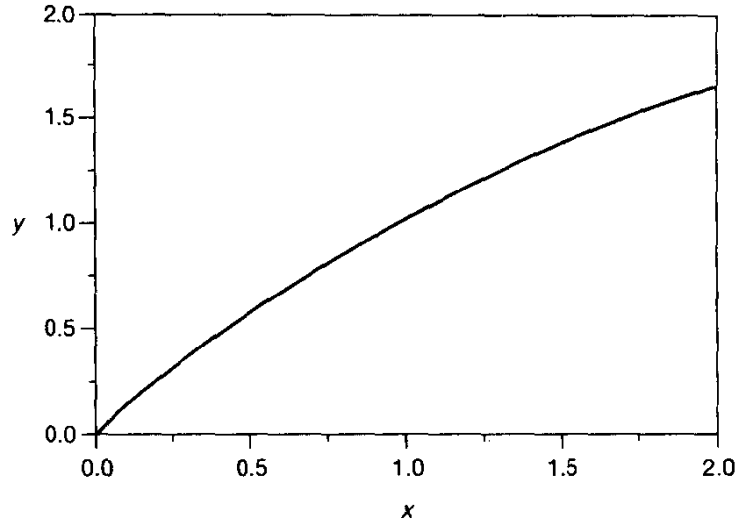



**Figure 19.4** Input–output characteristic of sigmoid nonlinearity for a blind equalizer of the Bussgang type.

## 19.2 MULTILAYER PERCEPTRON

The *multilayer perceptron (MLP)* is a neural network that consists of an *input layer* of source nodes, one or more *hidden layers* of computational nodes (neurons), and an *output layer* also made up of computational nodes (neurons). The source nodes provide physical access points for the application of input signals. The neurons in the hidden layers act as "feature detectors"; these neurons are referred to as "hidden" neurons because they are physically inaccessible from the input end or output end of the network. Finally, the neurons in the output layer present to a user the conclusions reached by the network in response to the input signals.

Figure 19.5 depicts a multilayer perceptron with a pair of input nodes, a single layer of five hidden neurons, and a single output neuron. Two features of such a structure are immediately apparent from this figure:

1. A multilayer perceptron is a *feedforward network*, in the sense that the input signals produce a response at the output(s) of the network by propagating in the forward direction only. Simply put, there is *no* feedback in the network.
2. The network may be *fully connected*, as shown in Fig. 19.5, in that each node in a layer of the network is connected to every other node in the layer adjacent to it. Alternatively, the network may be *partially connected* in that some of the synaptic links may be missing. Locally connected networks represent an important type of partially connected networks; the term "local" refers to the connectivity of a neuron in a layer of the network only to a subset of possible inputs.

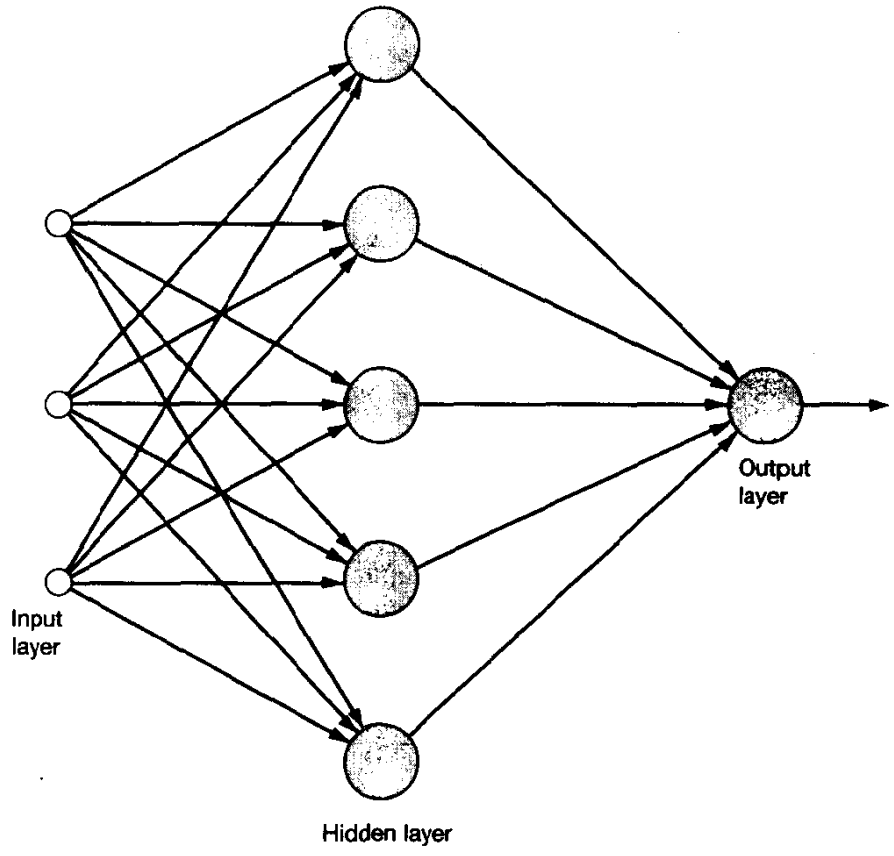



**Figure 19.5** Multilayer perceptron with a single hidden layer.

The number of source nodes in the input layer is determined by the dimensionality of the observation space that is responsible for the generation of the input signals. The number of computational nodes in the output layer is determined by the required dimensionality of the desired response. Thus, the design of a multilayer perceptron requires that we address three issues:

1. The determination of the number of hidden layers
2. The determination of the number of neurons in each of the hidden layers
3. The specification of the synaptic weights that interconnect the neurons in the different layers of the network

Issues 1 and 2 relate to *neural (model) complexity*. Unfortunately, these two issues represent the weakest link in our present knowledge of how to design a multilayer perceptron. More will be said on network complexity later in the chapter. To resolve issue 3, we may use the back-error propagation algorithm, also referred to in the literature as the *back-propagation algorithm*, or simply the backprop.

In the next section, we present a derivation of the complex back-propagation algorithm, which is designed to handle complex signals. The derivation of the back-propagation algorithm for real signals follows immediately as a special case of the complex back-propagation algorithm; the latter case is treated in Section 19.4.

## 19.3 COMPLEX BACK-PROPAGATION ALGORITHM

The *complex back-propagation (BP) algorithm* is a generalization of the complex LMS algorithm, with an appropriately chosen nonlinear activation function. There are two passes of signals in the implementation of the BP algorithm.

1. **Forward pass.** In the forward pass, also termed the *function level adaptation*, the synaptic weights are *fixed*, and the response of the network is computed by subjecting it to a prescribed set of input signals. The forward pass in the BP algorithm is analogous to the filtering process in the LMS algorithm.
2. **Backward pass.** In the backward pass, also termed the *parameter level adaptation*, the adjustments to the synaptic weights are computed for the purpose of minimizing a cost function defined as the sum of error squares. In particular, we start by computing the error signals in the output layer, and then work *backwards* through the network, layer by layer, until the complete network is covered. The BP algorithm derives its name from the backward nature of the error computations involved in its implementation. Note also that the backward pass in the BP algorithm is analogous to the adaptive process in the LMS algorithm.

The derivation of the BP algorithm is usually presented for real-valued data (Rumelhart et al., 1986; Werbos, 1993; Haykin, 1994). However, we will pursue a different course here by first deriving the complex form of the algorithm. In this context, the key question is: How do we handle the use of complex-valued data? This need often arises in processing coherent data, for example, those found in radar, sonar, and communications fields. We may accommodate the use of complex data in either one of two ways:

1. The real and imaginary parts of each member of the input set of data are treated as two separate entities; similarly, the real and imaginary parts of each member of the network output are treated as two separate entities. The synaptic weights of the network are then computed in accordance with the *real* (conventional) form of the BP algorithm.
2. The synaptic weights are assigned complex values and their computations are performed using the *complex* form of the BP algorithm.

We can show that the two approaches are equivalent (Haykin and Ukrainec, 1993). Let the desired mapping be written as

$$\mathbf{z} = \mathbf{z}_I + j\mathbf{z}_Q \rightarrow \varphi(\mathbf{z}) = u(\mathbf{z}_I, \mathbf{z}_Q) + jv(\mathbf{z}_I, \mathbf{z}_Q) \tag{19.5}$$

where

$$(\mathbf{z}_I, \mathbf{z}_Q) \rightarrow u(\mathbf{z}_I, \mathbf{z}_Q) \text{ and } (\mathbf{z}_I, \mathbf{z}_Q) \rightarrow v(\mathbf{z}_I, \mathbf{z}_Q) \tag{19.6}$$

and where $u$ and $v$ are real functions of the complex input vector $\mathbf{z}$; the subscripts $I$ and $Q$ refer to the in-phase and quadrature components (i.e., real and imaginary parts), respectively. Two real-valued feedforward networks (or one real-valued feedforward network with two outputs) can thus be used to compute the resultant mapping, one giving the real part of the mapping, the other the imaginary part of the mapping. There is, however, an advantage in using a network with complex weights. Referring to the linear combiner section of Fig. 19.1, its scalar output can be written in vector notation as follows (dropping subscript $i$ for convenience of notation):

$$\begin{aligned} \mathbf{x}^H\mathbf{w} &= (\mathbf{x}_I + j\mathbf{x}_Q)^H(\mathbf{w}_I + j\mathbf{w}_Q) \\ &= (\mathbf{x}_I^T\mathbf{w}_I + \mathbf{x}_Q^T\mathbf{w}_Q) + j(-\mathbf{x}_Q^T\mathbf{w}_I + \mathbf{x}_I^T\mathbf{w}_Q) \end{aligned} \tag{19.7}$$

where $\mathbf{x} = \mathbf{x}_I + j\mathbf{x}_Q$ is a complex input vector; likewise $\mathbf{w} = \mathbf{w}_I + j\mathbf{w}_Q$ is the corresponding complex weight vector. The equivalent real-valued combiner can also be constructed using only real-valued vectors, so that

$$[\mathbf{x}_I^T \quad \mathbf{x}_Q^T]\begin{bmatrix} \mathbf{u}_I & \mathbf{v}_I \\ \mathbf{u}_Q & \mathbf{v}_Q \end{bmatrix} = [\mathbf{x}_I^T\mathbf{u}_I + \mathbf{x}_Q^T\mathbf{u}_Q \quad \mathbf{x}_I^T\mathbf{v}_I + \mathbf{x}_Q^T\mathbf{v}_Q] \tag{19.8}$$

where the weight matrix consists of real vectors $\mathbf{u}_I$, $\mathbf{u}_Q$, $\mathbf{v}_I$, and $\mathbf{v}_Q$. The resultant real vector in Eq. (19.8) contains the real and imaginary components of the complex output in Eq. (19.7). Comparing Eqs. (19.7) and (19.8), we may readily see that

$$\begin{aligned} \mathbf{u}_I &= \mathbf{w}_I, \qquad \mathbf{v}_I = \mathbf{w}_Q \\ \mathbf{u}_Q &= \mathbf{w}_Q, \qquad \mathbf{v}_Q = -\mathbf{w}_I \end{aligned} \tag{19.9}$$

and therefore

$$\mathbf{u}_I = -\mathbf{v}_Q \qquad \mathbf{u}_Q = \mathbf{v}_I \tag{19.10}$$

It is apparent that a network with real-valued weights has more degrees of freedom than absolutely necessary to solve the complex mapping problem. In general, the real-valued learning algorithm treats all the weights as independent parameters, adjusting them to decrease the cost function. In the case of a complex-valued mapping, symmetries exist that are not taken advantage of by the learning algorithm.[1] In other words, the network

[1] It is possible to constrain the weights so that the above mentioned symmetry exists between the real-valued weights. However, the usual gradient descent algorithm does not make use of this information, which ends up being lost.

with complex-valued weights gives a parsimonious solution as compared to the network with real-valued weights. Other considerations include those of convergence. It has been shown (Horowitz and Senne, 1981) that for the LMS algorithm, superior performance is achieved for the complex LMS algorithm over that of the real version of the LMS algorithm. The algorithm is more stable, and the rate of mean-squared convergence is almost twice that of the real LMS algorithm. Since the BP algorithm is a generalization of the LMS algorithm, we may conjecture that this behavior carries over to feedforward neural networks as well.

### Derivation of the complex back-propagation algorithm

We now present a detailed derivation of the complex form of the back-propagation algorithm. The complex algorithm was developed independently by several researchers (Clarke, 1990; Kim and Guest,1990; Hensler and Braspenning, 1990; Leung and Haykin, 1991; Georgiou and Koustougeras, 1992; Birx and Pipenberg, 1992; Benvenuto and Piazza, 1992). All of the approaches are fundamentally a generalization of the complex least-mean square (LMS) algorithm to a network with multiple layers of multiple linear combiners with nonlinearities. The introduction of nonlinearity into the network raises the basic question: What form does the complex activation function take? The answer to this question requires a consideration of the nature of differentiable functions of complex variables.

A multilayer perceptron, shown in Fig. 19.6, consists of many adaptive linear combiners with a nonlinearity at the output; such a combiner is shown in Fig. 19.1. The input–output relationship of such a unit in layer $l$ of the network is characterized by the nonlinear difference equation

$$x_i^{(l+1)} = \varphi\left(\sum_{p=1}^{N} w_{ip}^{*(l)} x_p^{(l)} + b_i^{*(l)}\right) \tag{19.11}$$

with the output being the $i$th node in the $(l + 1)$th layer. The parameter $b$ is a bias term, equivalent to a weight with a constant + 1 input. Equation (19.11) is generalized to all units in the multilayer perceptron as shown in Fig. 19.6.

The error signal is defined to be the difference between some desired response and the actual output of the network. Specifically, for the $i$th output neuron we may write

$$e_i(n) = d_i - y_i(n), \qquad i = 1, 2, \ldots, N_M \tag{19.12}$$

where $d_i$ is the desired response at the $i$th node of the output layer, $y_i(n)$ is the output at the $i$th node of the output layer, and $N_M$ is the number of neurons in the output layer of the neural network, referred to hereafter as the $M$th layer; $n$ refers to the number of iterations of the algorithm. The sum of error squares produced by the network defines the cost function

$$\mathscr{E}(n) = \frac{1}{2}\sum_{i=1}^{N_M} e_i(n)e_i^*(n) = \frac{1}{2}\sum_{i=1}^{N_M} |e_i(n)|^2 \tag{19.13}$$

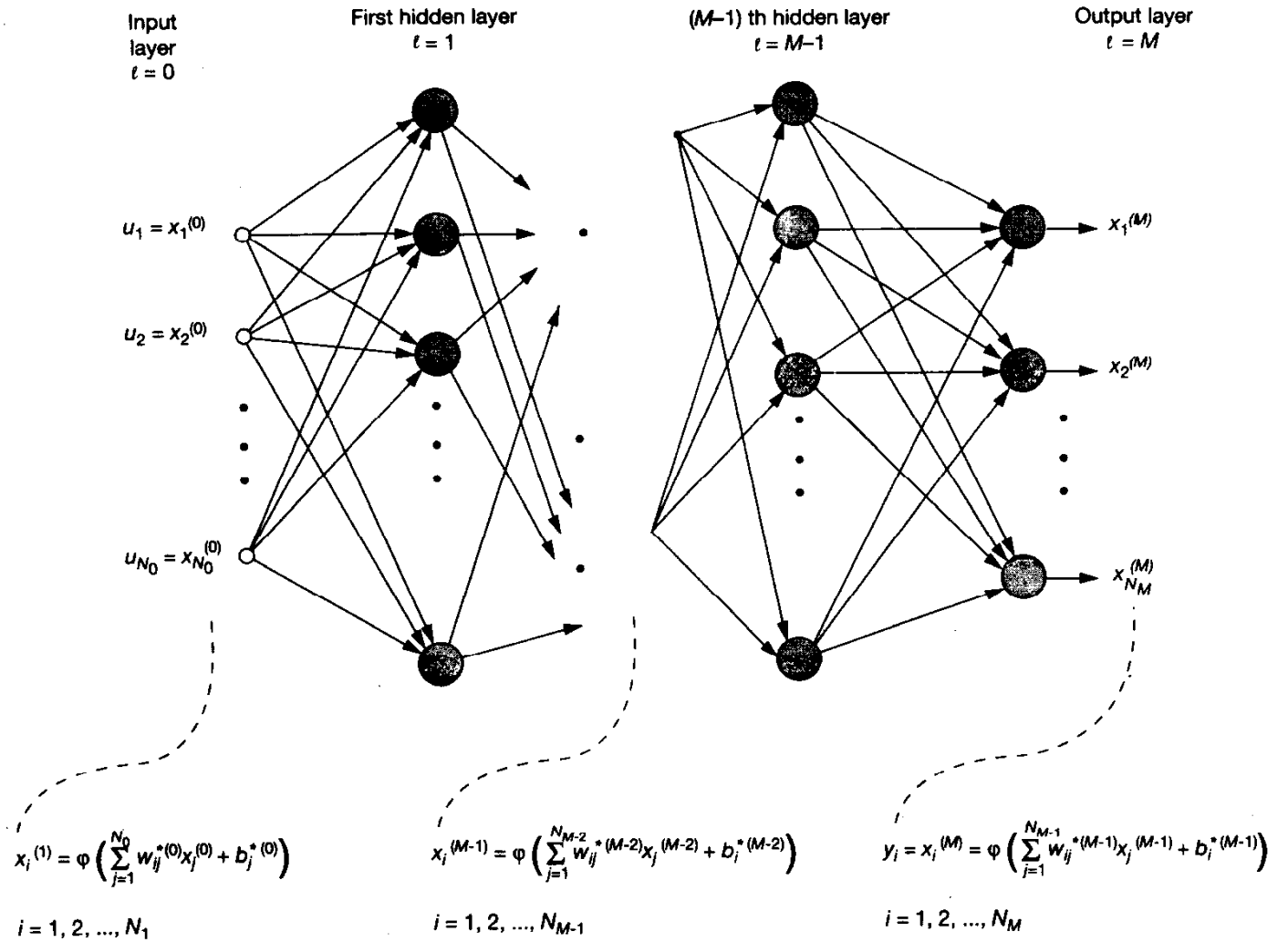



**Figure 19.6** Definitions of signals in various layers of multilayer perceptron.

The BP algorithm minimizes the cost function $\mathscr{E}(n)$ by recursively adjusting the complex weights of the multilayer perceptron, using an approximation to the gradient descent technique. The weight update equation is

$$w_{ip}^{(l)}(n+1) = w_{ip}^{(l)}(n) + \Delta w_{ip}^{(l)}(n) \tag{19.14}$$

The weights are changed in proportion to the negative of the gradient. The update term is defined to be

$$\Delta w_{ip}^{(l)}(n) = -\mu \nabla_{w_{ip}}^{(l)} \mathscr{E}(n) \tag{19.15}$$

where $\mu$ is a learning-rate parameter and $\nabla_{w_{ip}}^{(l)} \mathscr{E}(n)$ is the gradient of the cost function $\mathscr{E}(n)$ with respect to the weight $w_{ip}^{(l)}$. We must first find the partial derivative of $\mathscr{E}(n)$ with respect to the complex weights of the $(M-1)$th layer, and then extend it to the coefficients of all the hidden layers. The gradient of the cost function $\mathscr{E}(n)$ with respect to the complex weights in the $(M-1)$th layer is defined as

$$\nabla_{w_{ip}}^{(M-1)} \mathscr{E}(n) = \frac{\partial \mathscr{E}(n)}{\partial w_{I,ip}^{(M-1)}(n)} + j \frac{\partial \mathscr{E}(n)}{\partial w_{Q,ip}^{(M-1)}(n)} \tag{19.16}$$

where the complex weight connecting the $p$th node to the $i$th node for layer $(M-1)$ at iteration $n$ is given by

$$w_{ip}^{(M-1)}(n) = w_{I,ip}^{(M-1)}(n) + j w_{Q,ip}^{(M-1)}(n) \tag{19.17}$$

where the subscripts $I$ and $Q$ signify the real and imaginary components of the complex weight in question, respectively. The output $y_i(n)$ is therefore

$$y_i = x_i^{(M)} = \varphi(\text{net}_i^{(M-1)}) \tag{19.18}$$

where

$$\begin{aligned} \text{net}_i^{(M-1)} &= \text{net}_{I,i}^{(M-1)} + j\text{net}_{Q,i}^{(M-1)} \\ &= \sum_{p=1}^{N_{M-1}} w_{ip}^{*(M-1)} x_p^{(M-1)} + b_i^{*(M-1)} \end{aligned} \tag{19.19}$$

Assume that $\varphi(\text{net}_i^{(M-1)})$ is a suitable complex activation function; hence, let

$$\begin{aligned} \varphi(\text{net}_i) &= \varphi(\text{net}_{I,i} + j\text{net}_{Q,i}) \\ &= u(\text{net}_{I,i}, \text{net}_{Q,i}) + jv(\text{net}_{I,i}, \text{net}_{Q,i}) \end{aligned} \tag{19.20}$$

where $u$ and $v$ are real functions. The derivative of the activation function, if it exists, is defined by

$$\varphi'(\text{net}_i) = \frac{d\varphi(\text{net}_i)}{d\text{net}_i} \tag{19.21}$$

The partial derivatives of $u$ and $v$ with respect to the real and imaginary parts of the internal signal $\text{net}_i$ are defined by

$$\begin{aligned} u'_{I,i} &= \frac{\partial u_i}{\partial \text{net}_{I,i}} \\ u'_{Q,i} &= \frac{\partial u_i}{\partial \text{net}_{Q,i}} \\ v'_{I,i} &= \frac{\partial v_i}{\partial \text{net}_{I,i}} \\ v'_{Q,i} &= \frac{\partial v_i}{\partial \text{net}_{Q,i}} \end{aligned} \tag{19.22}$$

Next, we need to find expressions for $\partial \mathscr{E}(n)/\partial w_{I,ij}^{(M-1)}$ and $\partial \mathscr{E}(n)/\partial w_{Q,ij}^{(M-1)}$. Using the chain rule of calculus, we may write

$$\begin{aligned} \frac{\partial \mathscr{E}}{\partial w_{I,ip}^{(M-1)}} &= \frac{\partial \mathscr{E}}{\partial u_i}\left(\frac{\partial u_i}{\partial \text{net}_{I,i}}\frac{\partial \text{net}_{I,i}}{\partial w_{I,ip}} + \frac{\partial u_i}{\partial \text{net}_{Q,i}}\frac{\partial \text{net}_{Q,i}}{\partial w_{I,ip}}\right) \\ &\quad + \frac{\partial \mathscr{E}}{\partial v_i}\left(\frac{\partial v_i}{\partial \text{net}_{I,i}}\frac{\partial \text{net}_{I,i}}{\partial w_{I,ip}} + \frac{\partial v_i}{\partial \text{net}_{Q,i}}\frac{\partial \text{net}_{Q,i}}{\partial w_{I,ip}}\right) \end{aligned} \tag{19.23}$$

$$\begin{aligned} \frac{\partial \mathscr{E}}{\partial w_{Q,ip}^{(M-1)}} &= \frac{\partial \mathscr{E}}{\partial u_i}\left(\frac{\partial u_i}{\partial \text{net}_{I,i}}\frac{\partial \text{net}_{I,i}}{\partial w_{Q,ip}} + \frac{\partial u_i}{\partial \text{net}_{Q,i}}\frac{\partial \text{net}_{Q,i}}{\partial w_{Q,ip}}\right) \\ &\quad + \frac{\partial \mathscr{E}}{\partial v_i}\left(\frac{\partial v_i}{\partial \text{net}_{I,i}}\frac{\partial \text{net}_{I,i}}{\partial w_{Q,ip}} + \frac{\partial v_i}{\partial \text{net}_{Q,i}}\frac{\partial \text{net}_{Q,i}}{\partial w_{Q,ip}}\right) \end{aligned}$$

Evaluating the partial derivative $\partial \text{net}_{I,i}/\partial w_{I,ip}$, we may write

$$\frac{\partial \text{net}_{I,i}}{\partial w_{I,ip}} = \frac{\partial}{\partial w_{I,ip}}[w_{I,ip}x_{I,p} + w_{Q,ip}x_{Q,p} + b_{I,i}] = x_{I,p} \tag{19.24}$$

The other partial derivatives of Eqs. (19.23) can be found in a similar manner. Summarizing our findings up to this point in the discussion:

$$\frac{\partial \text{net}_{I,i}}{\partial w_{I,ip}} = x_{I,p} \tag{19.25}$$

$$\frac{\partial \text{net}_{I,i}}{\partial w_{Q,ip}} = x_{Q,p} \tag{19.26}$$

$$\frac{\partial \text{net}_{Q,i}}{\partial w_{I,ip}} = x_{Q,p} \tag{19.27}$$

$$\frac{\partial \text{net}_{Q,i}}{\partial w_{Q,ip}} = -x_{I,p} \tag{19.28}$$

Substituting Eqs. (19.22) and (19.25) to (19.28) into Eq. (19.23), the partial derivatives in the two lines of the latter equation can be expressed respectively as

$$\frac{\partial \mathscr{E}(n)}{\partial w_{I,ip}^{(M-1)}} = \frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-1)}} (u'_{I,i} x_{I,p}^{(M-1)} + u'_{Q,i} x_{Q,p}^{(M-1)}) + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-1)}} (v'_{I,i} x_{I,p}^{(M-1)} + v'_{Q,i} x_{Q,p}^{(M-1)}) \tag{19.29}$$

$$\frac{\partial \mathscr{E}(n)}{\partial w_{Q,ip}^{(M-1)}} = \frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-1)}} (u'_{I,i} x_{Q,p}^{(M-1)} - u'_{Q,i} x_{I,p}^{(M-1)}) + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-1)}} (v'_{I,i} x_{Q,p}^{(M-1)} - v'_{Q,i} x_{I,p}^{(M-1)}) \tag{19.30}$$

where, as mentioned previously, primes indicate differentiation. For the weights belonging to layer $(M-1)$ in the network, the partial derivatives of the cost function can be readily found as follows:

$$\frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-1)}} = -[d_{I,i} - y_{I,i}(n)] = -e_{I,i}(n) \tag{19.31}$$

$$\frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-1)}} = -[d_{Q,i} - y_{Q,i}(n)] = -e_{Q,i}(n) \tag{19.32}$$

Substituting Eqs. (19.29) and (19.30) into (19.16) and simplifying, we get

$$\nabla_{w_{ip}}^{(M-1)} \mathscr{E}(n) = x_p^{(M-1)}(n) \left( \frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-1)}} (u_{I,i}'^{(M-1)} - j u_{Q,i}'^{(M-1)}) + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-1)}} (v_{I,i}'^{(M-1)} - j v_{Q,i}'^{(M-1)}) \right) \tag{19.33}$$

Then, using Eqs. (19.31) and (19.32):

$$\nabla_{w_{ip}}^{(M-1)} \mathscr{E}(n) = -x_p^{(M-1)}(n)[e_{I,i}(n)(u'_{I,i} - j u'_{Q,i}) + e_{Q,i}(n)(v'_{I,i} - j v'_{Q,i})] \tag{19.34}$$

Hence, the weight update rule of Eq. (19.14) becomes

$$w_{ip}^{(M-1)}(n+1) = w_{ip}^{(M-1)}(n) + \mu x_p^{(M-1)}(n)[e_{I,i}(n)(u'_{I,i} - j u'_{Q,i}) + e_{Q,i}(n)(v'_{I,i} - j v'_{Q,i})] \tag{19.35}$$

The update rule for the bias term $b$ can be derived in a similar manner. We will just state it here to be

$$b_i^{(M-1)}(n+1) = b_i^{(M-1)}(n) + \mu[e_{I,i}(n)(u'_{I,i} - j u'_{Q,i}) + e_{Q,i}(n)(v'_{I,i} - j v'_{Q,i})] \tag{19.36}$$

Pattern classification tasks often require a mapping from a multidimensional feature space to a class label. Feature data belonging to a class $\mathscr{C}_k$ are trained to map to a constant value at an output node $k$ in the neural network. For this application, a bounded, nonlinear activation function at the output is desirable. However, if a continuous mapping is

required, as in a nonlinear prediction problem that is an example of nonlinear regression, it is necessary to remove the nonlinearity from the output unit and thereby have the final layer operate as a linear combiner; this modification allows the output to vary in an unbounded fashion. In this latter case, we let

$$\varphi(\text{net}_i) = \text{net}_i \tag{19.37}$$

The partial derivatives of this function reduce to

$$u'_{I,i} = 1$$

$$u'_{Q,i} = 0$$

$$v'_{I,i} = 0$$

$$v'_{Q,i} = 1$$

Substituting these values into the weight update rule of Eq. (19.35), we get

$$w_{ip}^{(M-1)}(n+1) = w_{ip}^{(M-1)}(n) + \mu x_p^{(M-1)}(n) e_i^*(n) \tag{19.38}$$

which corresponds to the familiar complex LMS algorithm.

We have now shown the process for updating the $(M-1)$th layer (i.e., output layer) of weights. The next step is to derive the relations necessary to update the weights in the hidden layers of the multilayer perceptron. The main idea is to find expressions that relate the error in the $l$th layer to the $(l-1)$th layer. In this way we may *back-propagate* the error, stepping from the output layer back towards the input layer in a layer-by-layer manner.

Restating Eqs. (19.29) and (19.30) in terms of a hidden layer of the multilayer perceptron, the expressions for the partial derivatives of the cost function $\mathscr{E}(n)$ with respect to the weights in layer $(M-2)$ are as follows:

$$\begin{aligned}\frac{\partial \mathscr{E}(n)}{\partial w_{I,ip}^{(M-2)}} &= \frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-2)}}\left(u'_{I,i}x_{I,p}^{(M-2)} + u'_{Q,i}x_{Q,p}^{(M-2)}\right) \\ &\quad + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-2)}}\left(v'_{I,i}x_{I,p}^{(M-2)} + v'_{Q,i}x_{Q,p}^{(M-2)}\right)\end{aligned} \tag{19.39}$$

$$\begin{aligned}\frac{\partial \mathscr{E}(n)}{\partial w_{Q,ip}^{(M-2)}} &= \frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-2)}}\left(u'_{I,i}x_{Q,p}^{(M-2)} - u'_{Q,i}x_{I,p}^{(M-2)}\right) \\ &\quad + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-2)}}\left(v'_{I,i}x_{Q,p}^{(M-2)} - v'_{Q,i}x_{I,p}^{(M-2)}\right)\end{aligned} \tag{19.40}$$

Using the chain rule, we may now write

$$\begin{aligned}\frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-2)}} &= \sum_k \frac{\partial \mathscr{E}(n)}{\partial u_k^{(M-1)}}\left(\frac{\partial u_k^{(M-1)}}{\partial \text{net}_{I,k}}\frac{\partial \text{net}_{I,k}}{\partial u_i^{(M-2)}} + \frac{\partial u_k^{(M-1)}}{\partial \text{net}_{Q,k}}\frac{\partial \text{net}_{Q,k}}{\partial u_i^{(M-2)}}\right) \\ &\quad + \sum_k \frac{\partial \mathscr{E}(n)}{\partial v_k^{(M-1)}}\left(\frac{\partial v_k^{(M-1)}}{\partial \text{net}_{I,k}}\frac{\partial \text{net}_{I,k}}{\partial u_i^{(M-2)}} + \frac{\partial v_k^{(M-1)}}{\partial \text{net}_{Q,k}}\frac{\partial \text{net}_{Q,k}}{\partial u_i^{(M-2)}}\right)\end{aligned} \tag{19.41}$$

$$\frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-2)}} = \sum_k \frac{\partial \mathscr{E}(n)}{\partial u_k^{(M-1)}} \left( \frac{\partial u_k^{(M-1)}}{\partial \text{net}_{I,k}} \frac{\partial \text{net}_{I,k}}{\partial v_i^{(M-2)}} + \frac{\partial u_k^{(M-1)}}{\partial \text{net}_{Q,k}} \frac{\partial \text{net}_{Q,k}}{\partial v_i^{(M-2)}} \right) + \sum_k \frac{\partial \mathscr{E}(n)}{\partial v_k^{(M-1)}} \left( \frac{\partial v_k^{(M-1)}}{\partial \text{net}_{I,k}} \frac{\partial \text{net}_{I,k}}{\partial v_i^{(M-2)}} + \frac{\partial v_k^{(M-1)}}{\partial \text{net}_{Q,k}} \frac{\partial \text{net}_{Q,k}}{\partial v_i^{(M-2)}} \right) \tag{19.42}$$

The partial derivatives of the cost function $\mathscr{E}(n)$ in Eqs. (19.41) and (19.42) are expressed in terms of the partial derivatives pertaining to the previous layer $(M-1)$. We now only need to determine the partial derivatives of

$$\begin{aligned} \text{net}_k^{(M-1)} &= \sum_i w_{ki}^{*(M-1)} \varphi(\text{net}_i^{(M-2)}) + b_k^{*(M-1)} \\ &= \sum_i (u_i^{(M-2)} w_{I,ki}^{(M-1)} + v_i^{(M-2)} w_{Q,ki}^{(M-1)} + b_{I,k}^{(M-1)}) \\ &\quad + j(v_i^{(M-2)} w_{I,ki}^{(M-1)} - u_i^{(M-2)} w_{Q,ki}^{(M-1)} - b_{Q,k}^{(M-1)}) \end{aligned} \tag{19.43}$$

with respect to the $u$ and the $v$ of the previous layer $(M-2)$. Summarizing these partial derivatives, we have

$$\frac{\partial \text{net}_{I,k}^{(M-1)}}{\partial u_i^{(M-2)}} = w_{I,ki}^{(M-1)} \tag{19.44}$$

$$\frac{\partial \text{net}_{I,k}^{(M-1)}}{\partial v_i^{(M-2)}} = w_{Q,ki}^{(M-1)} \tag{19.45}$$

$$\frac{\partial \text{net}_{Q,k}^{(M-1)}}{\partial u_i^{(M-2)}} = -w_{Q,ki}^{(M-1)} \tag{19.46}$$

$$\frac{\partial \text{net}_{Q,k}^{(M-1)}}{\partial v_i^{(M-2)}} = w_{I,ki}^{(M-1)} \tag{19.47}$$

Substituting Eqs. (19.44) to (19.47) into (19.41) and (19.42), we may express the partial derivatives of interest as

$$\frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-2)}} = \sum_k \frac{\partial \mathscr{E}(n)}{\partial u_k^{(M-1)}} (u_{I,k}'^{(M-1)} w_{I,ki}^{(M-1)} - u_{Q,k}'^{(M-1)} w_{Q,ki}^{(M-1)}) + \sum_k \frac{\partial \mathscr{E}(n)}{\partial v_k^{(M-1)}} (v_{I,k}'^{(M-1)} w_{I,ki}^{(M-1)} - v_{Q,k}'^{(M-1)} w_{Q,ki}^{(M-1)}) \tag{19.48}$$

$$\frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-2)}} = \sum_k \frac{\partial \mathscr{E}(n)}{\partial u_k^{(M-1)}} (u_{I,k}'^{(M-1)} w_{Q,ki}^{(M-1)} + u_{Q,k}'^{(M-1)} w_{I,ki}^{(M-1)}) + \sum_k \frac{\partial \mathscr{E}(n)}{\partial v_k^{(M-1)}} (v_{I,k}'^{(M-1)} w_{Q,ki}^{(M-1)} + v_{Q,k}'^{(M-1)} w_{I,ki}^{(M-1)}) \tag{19.49}$$

Combining these results as the real and imaginary parts of a complex partial derivative, we may simplify matters by writing

$$\frac{\partial \mathscr{E}(n)}{\partial u_i^{(M-2)}} + j\frac{\partial \mathscr{E}(n)}{\partial v_i^{(M-2)}} = \sum_k w_{ki}^{(M-1)}(n)\left(\frac{\partial \mathscr{E}(n)}{\partial u_k^{(M-1)}}(u'_{I,k} + ju'_{Q,k}) + \frac{\partial \mathscr{E}(n)}{\partial v_k^{(M-1)}}(v'_{I,k} + jv'_{Q,k})\right) \tag{19.50}$$

where the primed variables refer to layer $M - 1$.

Using induction, we can extend this relationship to the other hidden layers of the multilayer perceptron. Equation (19.50) gives us the means to back-propagate the error from the output layer, ($M$), to the input layer (0). After the values for $\partial \mathscr{E}(n)/\partial u$ and $\partial \mathscr{E}(n)/\partial v$ for the particular layer have been determined, Eq. (19.33) gives the gradient, and hence the weight update values.

### Complex-valued activation function

One of the difficulties encountered in extending the real BP algorithm to the complex domain involves the appropriate choice of activation function. The straightforward extension of the sigmoidal from the real domain to the complex domain is inadequate, due to the fact that it has singularities, such that

$$\frac{1}{1 + e^{-z}} \to \infty \qquad \text{for } z = \pm j(2k + 1)\pi, \qquad k \text{ any integer} \tag{19.51}$$

For a practical implementation of the complex multilayer perceptron, it is necessary that the activation function be bounded. Without such a guarantee, there is a risk of arithmetic overflow in software implementation of multilayer perceptron; hardware implementation would suffer in an analogous manner, with unbounded outputs resulting in possible clipping at node outputs. Singularities in an activation function must therefore be avoided.

Georgiou and Koutsougeras (1992) have developed a set of properties, which a complex activation function must satisfy in order to be useful in a multilayer perceptron trained with the back-propagation algorithm. These properties are summarized here:

1. The activation function $\varphi(z)$ should be nonlinear in both $z_I$ and $z_Q$, which denote the real and imaginary parts of the argument $z$; otherwise, there is no advantage in having a multilayer perceptron. A multilayer perceptron that is linear may always be collapsed to an equivalent single-layer network. The motivation here is to have a nonlinear network that can compute a more general set of functions than is possible with a linear network.
2. The function $\varphi(z)$ should be bounded. The computation of the forward pass of the multilayer perceptron is required to be bounded; otherwise, clipping or numerical overflow can occur.
3. The partial derivatives of $\varphi(z)$ should exist and be bounded. The learning phase updates the complex weights of the multilayer perceptron by amounts proportional to the partial derivatives, so they also need to be bounded.

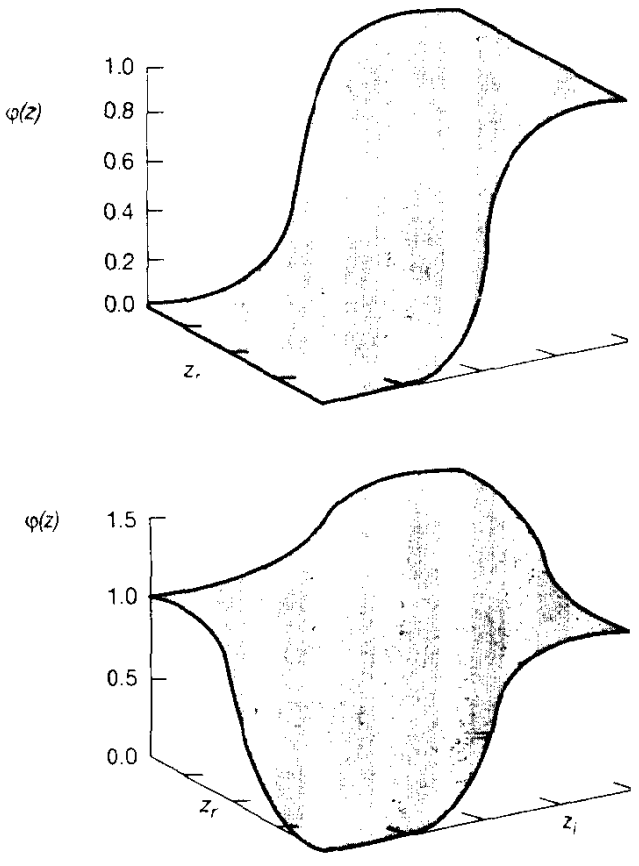



**Figure 19.7** Real part (top) and magnitude (bottom) of the activation function $\varphi(z) = c/(1 + e^{-kz_r}) + jc/(1 + e^{-kz_i})$

4. The function $\varphi(z)$ should not be an entire function. *Entire functions* are defined as complex functions that are analytic everywhere in the complex domain. A function is defined to be *analytic* at some point $z_0$ if it is differentiable in some neighborhood of $z_0$. By *Liouville's theorem*,[2] we know that a bounded, entire function is constant. Clearly, a function that is entire is not a suitable choice for an activation function for the reasons stated in Property 1.
5. The partial derivatives of the cost function $\mathscr{E}(n)$ should satisfy the condition:

$$\frac{\partial \mathscr{E}}{\partial u} + \frac{j\partial \mathscr{E}}{\partial v} \neq 0$$

[2] A review of complex variable theory, including Liouville's theorem, is presented in Appendix A.

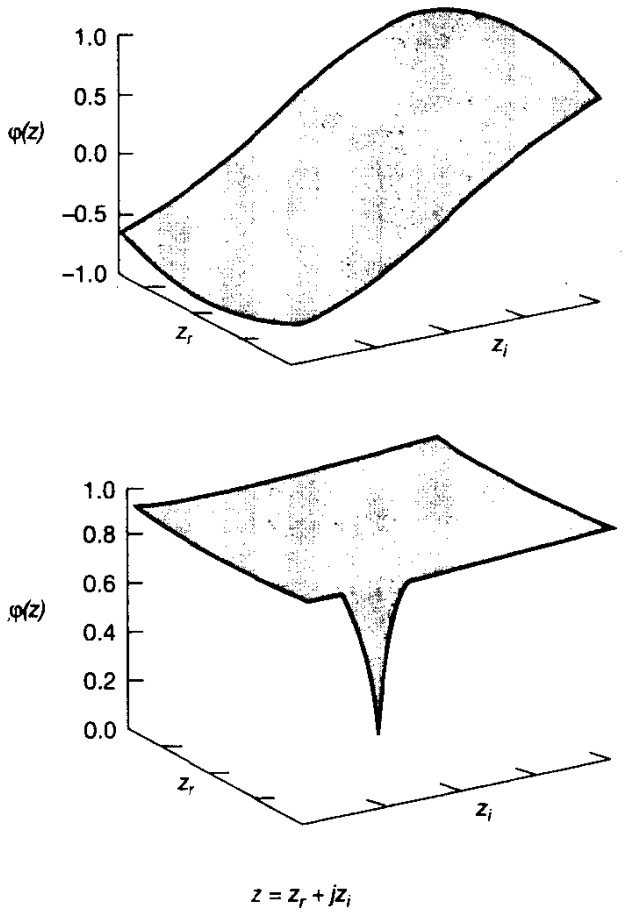



**Figure 19.8** Real part (top) and magnitude (bottom) of the activation function $\varphi(z) = z/[c + (1/r)\,|z|]$.

For this condition to hold, the partial derivatives of $\varphi(z)$ should not satisfy the relation $u_I v_Q = u_Q v_I$. This relationship can be satisfied by simultaneously setting the real and imaginary parts of Eq. (19.33) equal to zero, for $x_p(n) \neq 0$. Should the partial derivatives of the activation function satisfy the above relation, this would imply that in the presence of both nonzero input and error, it would be possible that $\nabla_w \mathscr{E} = 0$, and therefore a stationary point would be reached. No further learning could then take place, since the weight update is proportional to the gradient.

Figures 19.7 and 19.8 show two possible choices for the complex activation function. Figure 19.7 shows the complex activation function suggested by Benvenuto and Piazza (1992). The function is a superposition of real and imaginary sigmoids, as shown by

$$\varphi(z) = \frac{c}{1 + \exp(-kz_r)} + j\,\frac{c}{1 + \exp(-kz_i)} \tag{19.52}$$

where $z_r$ and $z_i$ are the real and imaginary parts of $z$, respectively.

**TABLE 19.1** SUMMARY OF THE COMPLEX BACK-PROPAGATION ALGORITHM

1. *Initialization*
   Set all weights and biases to small complex random values
2. *Present input and desired outputs*
   Present input vector $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)$ and desired response $\mathbf{d}(1), \mathbf{d}(2), \ldots, \mathbf{d}(N)$, where $N$ is the total number of training patterns
3. *Calculate actual outputs*
   Use the formulas in Fig. 19.6 to calculate the output signals $y_1, y_2, \ldots, y_{N_M}$
4. *Adapt weights and biases*

$$\Delta w_{ip}^{(l-1)}(n) = -\mu x_p^{(l-1)}(n)\left(\frac{\partial \mathscr{E}(n)}{u_i^{(l-1)}}(u'_{I,i} - ju'_{Q,i}) + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(l-1)}}(v'_{I,i} - jv'_{Q,i})\right)$$

$$\Delta b_i^{(l-1)}(n) = -\mu\left(\frac{\partial \mathscr{E}(n)}{\partial u_i^{(l-1)}}(u'_{I,i} - ju'_{Q,i}) + \frac{\partial \mathscr{E}(n)}{\partial v_i^{(l-1)}}(v'_{I,i} - jv'_{Q,i})\right)$$

where

$$\frac{\partial \mathscr{E}(n)}{\partial u_i^{(l-1)}} + j\frac{\partial \mathscr{E}(n)}{\partial v_i^{(l-1)}} = \begin{cases} -[d_i - y_i(n)] & \text{for } l = M \\ \displaystyle\sum_k w_{ki}^{(l)}\left(\frac{\partial \mathscr{E}(n)}{\partial u_k^{(l)}}(u'_{I,k} + ju'_{Q,k}) + \frac{\partial \mathscr{E}(n)}{\partial v_k^{(l)}}(v'_{I,k} + jv'_{Q,k})\right) & \text{for } 1 \le l < M \end{cases}$$

where $x_p(n)$ = output of node $p$ or input to node $i$ at iteration $n$

Another possible activation function is suggested by Georgiou and Koutsougeras (1992); it is also a sigmoidlike function, as shown in Fig. 19.8, with

$$\varphi(z) = \frac{z}{c + (1/r)|z|} \tag{19.53}$$

This function maps the $z$-domain to an open disk $|z| < r$; hence, the activation function effectively squashes the range of $|\varphi(z)|$ to the interval $[0,r]$.

Now that we have identified suitable choices for the complex activation functions, we may finish our discussion of the complex back-propagation algorithm by summarizing the important steps involved in its application, as outlined in Table 19.1.

## Incorporation of a Momentum Term

The back-propagation learning process may be accelerated by incorporating a *momentum* term (Rumelhart et al., 1986). Specifically, the correction $\Delta w_{ip}^{(l)}(n)$ applied to the synaptic $w_{ip}^{(l)}(n)$ in layer $l$ of the network, defined in Eq. (19.15), is modified as follows:

$$\Delta w_{ip}^{(l)}(n) = \alpha \Delta w_{ip}^{(l)}(n-1) - \frac{1}{2}\mu \nabla w_{ip}^{(l)} \mathscr{E}(n) \tag{19.54}$$

where $\alpha$ is called the *momentum constant*, and $\Delta w_{ip}^{(l)}(n-1)$ is the previous value of the correction. As before, $\mu$ is the learning-rate parameter.

The use of momentum introduces a feedback loop around $\Delta w_{ip}^{(l)}(n)$. As such, it can have a highly beneficial effect on learning behavior of the back-propagation algorithm. In particular, it may have the benefit of preventing the learning process from being stuck at a local minimum on the error-performance surface of the multilayer perceptron (Rumelhart et al., 1986; Haykin, 1994).

## 19.4 BACK-PROPAGATION ALGORITHM FOR REAL PARAMETERS

The common development of the back-propagation algorithm is for real-valued data and parameters. We now show that this is merely a special case of the more general, complex back-propagation algorithm developed in the previous section.

We proceed by considering all the parameters to be real-valued, including the input and desired output data. In terms of the complex-valued neural network, the quadrature components are all set equal to zero. Applying this principle to Eq. (19.23), we readily observe that only the first term survives, so that

$$\frac{\partial \mathscr{E}}{\partial w_{ip}} = \frac{\partial \mathscr{E}}{du_i} \frac{\partial u_i}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial w_{ip}} \tag{19.55}$$

Note that the in-phase designation, $I$, is dropped from the variables in this equation; since there is no longer a quadrature signal component to consider, it is a redundant notation. We also replace all occurrences of $u$ with

$$\varphi(\text{net}_i) = u(\text{net}_i, 0) \tag{19.56}$$

and

$$\varphi'(\text{net}_i) = u'_{I,i} = \frac{\partial u_i}{\partial \text{net}_{I,i}} \tag{19.57}$$

Equation (19.55) can now be rewritten as

$$\frac{\partial \mathscr{E}}{\partial w_{ip}} = \frac{\partial \mathscr{E}}{\partial \varphi(\text{net}_i)} \frac{\partial \varphi(\text{net}_i)}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial w_{ip}} \tag{19.58}$$

The activation function, $\varphi(\text{net})$, can be any bounded, differentiable, monotonically increasing function. The sigmoid function is often the function of choice.

We now define a new variable

$$\delta_i^{(l-1)}(n) = -\frac{\partial \mathscr{E}(n)}{\partial \text{net}_i^{(l-1)}} \tag{19.59}$$

For the case of the output layer of the multilayer perceptron, that is, $l = M$, we may write

$$\delta_i^{(M-1)}(n) = -\frac{\partial \mathscr{E}}{\partial \varphi(\text{net}_i)} \frac{\partial \varphi(\text{net}_i)}{\partial \text{net}_i} = \varphi'(\text{net}_i^{(M-1)}) e_i(n) \tag{19.60}$$

**TABLE 19.2 SUMMARY OF THE REAL BACK-PROPAGATION ALGORITHM**

1. *Initialization*
   Set all weights and biases to small real random values
2. *Present input and desired outputs*
   Present input vector $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)$ and desired response $\mathbf{d}(1), \mathbf{d}(2), \ldots, \mathbf{d}(N)$, where $N$ is the number of training patterns
3. *Calculate actual outputs*
   Use the formulas in Fig. 19.6 to calculate the output signals $y_1, y_2, \ldots, y_{N_M}$
4. *Adapt weights and biases*

$$\Delta w_{ij}^{(l-1)}(n) = \mu x_j(n)\delta_i^{(l-1)}(n)$$
$$\Delta b_i^{(l-1)}(n) = \mu\, \delta_i^{(l-1)}(n)$$

where

$$\delta_i^{(l-1)}(n) = \begin{cases} \varphi'(\text{net}_i^{(l-1)})\,[d_i - y_i(n)], & l = M \\ \varphi'(\text{net}_i^{(l-1)}) \displaystyle\sum_k w_{ki}\,\delta_k^{(l)}(n), & 1 \le l < M \end{cases}$$

where $x_j(n)$ = output of node $j$ or input to node $i$ at iteration $n$

where the crime on the right-hand side of the equation signifies differentiation. For a hidden layer of the network, that is, $1 \le l < M$, we have

$$\delta_i^{(l-1)}(n) = \varphi'(\text{net}_i^{(l-1)}) \sum_k w_{ki}(n)\delta_k^l(n) \tag{19.61}$$

The variable $\delta$ is interpreted as a back-propagated error term, which can be recursively computed for each layer of the multilayer perceptron, starting from the output layer.

A summary of the real-back-propagation algorithm is presented in Table 19.2. Note that a momentum term can also be added here in a manner similar to that described for the complex back-propagation algorithm.

## 19.5 UNIVERSAL APPROXIMATION THEOREM

A multilayer perceptron trained with the back-propagation algorithm provides a powerful device for approximating a *nonlinear input–output mapping* of a general nature. In this context, a key question needs to be considered: What is the number of hidden layers that would be needed in the design of the multilayer perceptron to do the approximation in a uniform manner? The answer to this fundamental question lies in the *universal approximation theorem*, which was developed independently by Cybenko (1989), Funahashi

(1989), and Hornik et al. (1989). The universal approximation theorem may be stated as follows:

Let $f(\cdot)$ be a nonconstant, bounded, and monotone-increasing continuous function. Let $I_{N_0}$ denote the $N_0$-dimensional unit hypercube. The space of continuous functions on $I_{N_0}$ is denoted by $C(I_{N_0})$. Then, given any function $f \in C(I_{N_0})$ and $\varepsilon > 0$, there exist an integer $N_1$ and sets of real constants $\alpha_i$, $b_i$, and $w_{ip}$, where $i = 1, 2, \ldots, N_1$ and $p = 1, 2, \ldots, N_0$ such that we may define

$$F(u_1, u_2, \ldots, u_{N_0}) = \sum_{i=1}^{N_1} \alpha_i \varphi\left(\sum_{p=1}^{N_0} w_{ip} x_p + b_i\right) \tag{19.62}$$

as an approximate realization of the function $f(\cdot)$; that is, the absolute value of the approximation satisfies the condition

$$|F(u_1, u_2, \ldots, u_{N_0}) - f(u_1, u_2, \ldots, u_{N_0})| < \varepsilon$$

for all $\{u_1, u_2, \ldots, u_{N_0}\} \in I_{N_0}$.

The universal approximation theorem is directly applicable to a multilayer perceptron having the following description:

- An input layer of $N_0$ nodes, whose individual inputs are denoted by $x_1, x_2, \ldots, x_{N_0}$
- A single hidden layer of $N_1$ sigmoidal neurons, with the synaptic weights of the $i$th hidden neuron being denoted by $w_{i1}, w_{i2}, \ldots, w_{iN_0}$
- An output layer consisting of a single linear neuron

It should be emphasized that the universal approximation theorem is an existence theorem, in the sense that it provides the mathematical justification for the approximation of an arbitrary continuous function as opposed to exact representation. Equation (19.62), which is the backbone of the theorem, merely generalizes approximations by finite Fourier series. In effect, the theorem states that a single hidden layer is sufficient for a multilayer perceptron to compute a uniform $\varepsilon$ approximation into a given training set represented by the sets of inputs $x_1, x_2, \ldots, x_{N_0}$ and a desired (target) output denoted by $f(x_1, x_2, \ldots, x_{N_0})$. From a theoretical viewpoint, the universal approximation theorem is therefore important. Without such a theorem we could be conceivably searching for a solution that cannot exist. However, the theorem does not say that a single hidden layer is optimum in the sense of learning time or ease of implementation.

From a practical perspective, the problem with multilayer perceptrons using a single hidden layer is that the hidden neurons tend to interact with each other. In complex situations, this interaction makes it difficult to improve the approximation at one point without worsening it at some other point. On the other hand, with two hidden or more layers the approximation (curve-fitting) process may become more manageable (Chester, 1990). It is

for this reason that we find in solving large-scale problems, the recommended procedure is to use a multilayer perceptron with two (or possibly more) hidden layers.

## 19.6 NETWORK COMPLEXITY

To solve real-world problems with multilayer perceptrons, we usually require the use of highly structured networks of a rather large size. A practical issue that arises in this context is that of minimizing the size of the network and yet maintaining good performance. As a general rule, a neural network with minimum size is less likely to learn the idiosyncrasies or noise in the training data, and may therefore generalize better to new data. *Generalization*, a term borrowed from psychology, refers to the ability of a neural network, having learned the essential information content of training data, to achieve "reasonable" performance for test data drawn from the same input space but not seen before.

We may achieve the design objective of minimum network size by proceeding in either one of the following two ways:

- *Network growing*, in which we start with a multilayer perceptron that is small for accomplishing the task at hand, and then add a new neuron or a new layer of hidden neurons only when we are unable to meet the design specification.
- *Network pruning*, in which we start with a large multilayer perceptron with an adequate performance for the task at hand, and then prune it by eliminating "unreliable" synaptic weights in a selective and orderly manner.

Although both of these approaches are used in practice, it is safe to say that in current practice, network pruning is the more popular one of the two.

In this section we describe the so-called *weight-eliminating procedure* (Weigend et al., 1991), the objective of which is to find a weight vector $\mathbf{w}$ that minimizes the total *risk*

$$\mathscr{R}(\mathbf{w}) = \mathscr{E}_s(\mathbf{w}) + \lambda \mathscr{E}_c(\mathbf{w}) \tag{19.63}$$

The first term, $\mathscr{E}_s(\mathbf{w})$, is the *standard* performance measure that depends on both the network (model) and the input data. In back-propagation learning, $\mathscr{E}_s(\mathbf{w})$ is typically defined as a mean-squared error whose evaluation extends over the output neurons of the network, and which is carried out for all the training data. The second term, $\mathscr{E}_c(\mathbf{w})$, is the *complexity penalty*, which depends on the network (model) alone. The evaluation of $\mathscr{E}_c(\mathbf{w})$ is confined to the synaptic connections of the network. In the weight-elimination procedure, the complexity penalty is defined by

$$\mathscr{E}_c(\mathbf{w}) = \sum_{i \in l} \frac{(w_i/w_o)^2}{1 + (w_i/w_o)^2} \tag{19.64}$$

where $w_o$ is a prescribed free parameter of the procedure, and $w_i$ refers to the weight of synapse $i$ in the network. The set $l$ refers to all the synaptic connections in the network. An

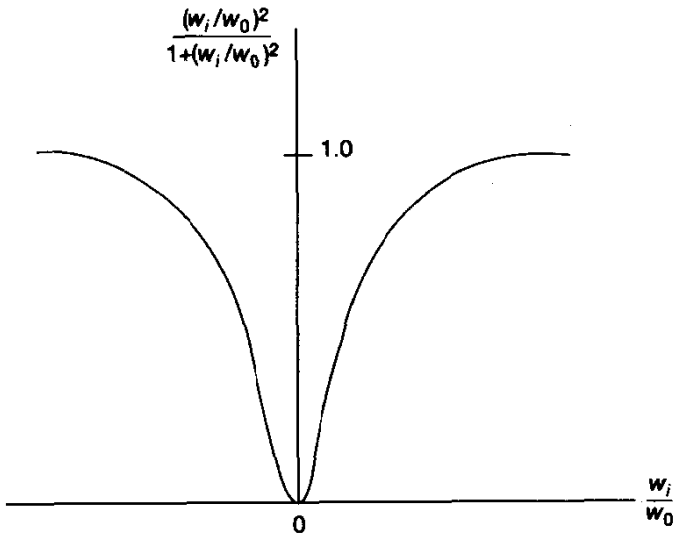



**Figure 19.9** Complex penalty function.

individual penalty term varies with $w_i/w_o$ in a symmetric fashion, as shown in Fig. 19.9. We may identify two limiting conditions:

- When $|w_i| << w_o$, the complexity penalty (cost) for that weight approaches zero. The implication of this condition is that insofar as learning from training data is concerned, the $i$th synaptic weight is unreliable and should therefore be eliminated from the network.
- When $|w_i| >> w_o$, the complexity penalty (cost) for that weight approaches the maximum value of unity, which means that $w_i$ is important to the back-propagation learning process.

The parameter $\lambda$ in Eq. (19.63) plays the role of a *regularization parameter*. When $\lambda$ is zero, the back-propagation learning process is unconstrained, in which case the network is completely determined by the training data in the manner described in Section 19.3 for complex data and Section 19.4 for real data. When $\lambda$ is made infinitely large, on the other hand, the implication is that the constraint imposed by the complexity penalty is by itself sufficient to specify the network. This is another way of saying that the training data are unreliable and should therefore be ignored. In practical applications of the weight-elimination procedure, the regularization parameter $\lambda$ is assigned a value somewhere between these two limiting cases.

Thus, starting with a multilayer perceptron designed by means of the back-propagation algorithm, and having chosen a value for the regularization parameter $\lambda$ that is appropriate for the particular situation under study, the network is pruned by minimizing the total risk $\mathcal{R}(\mathbf{w})$ defined in Eq. (19.63). Clearly, the computational effort involved in this

minimization is highly dependent on the size of the network. The ultimate aim is, of course, to *make the network complexity a close match to the complexity of the data used to train the network.*

## 19.7 FILTERING APPLICATIONS

Before proceeding to describe some filtering applications of multilayer perceptrons, it is instructive to distinguish between learning and adaptation, which go on, for example, in the back-propagation (BP) and the least-mean-square (LMS) algorithms, respectively. In the LMS algorithm, adjustments to the tap weights of a transversal filter are made while at the same time the input signal is being processed. This kind of a process is an example of *continuous learning*, which never stops. In contrast, in the BP algorithm, the synaptic weights of the multilayer perceptron are adjusted during the training phase; and once steady-state conditions are reached, all the synaptic weights in the network are fixed thereafter. In other words, in a multilayer perceptron, learning precedes signal processing. Clearly then, signal-processing (filtering) applications of multilayer perceptrons have to take full account of the way in which this class of neural networks learns from its environment.

For our present discussion, we have chosen three applications of multilayer perceptrons: the first relating to system identification, the second involving the time-delay neural network for temporal signal processing, and the third dealing with target detection. In the sequel, these three applications are described in that order. In all three cases the emphasis is on nonlinear signal processing in one form or another.

### System Identification

In light of what we have just said about back-propagation learning, the multilayer perceptron is basically a *static* network. We may extend its use for the identification of a nonlinear dynamic system by the incorporation of unit-delay elements at its input, output, or both, as described next.

For the identification of a nonlinear dynamic system, we may formulate four different models, depending on how the output of the system is defined in terms of past values of the output and past values of the input. Specifically, we may describe the models in terms of nonlinear difference equations as described in Narendra and Pasthasarathy (1990).

**Model I.** The output $y(n + 1)$ at time $n + 1$ depends linearly on $N$ past values of the output, $y(n), \ldots, y(n - N + 1)$, and nonlinearly on $M$ past values of the input, $u(n), \ldots, u(n - M + 1)$, as shown by

$$y(n + 1) = \sum_{i=0}^{N-1} \alpha_i y(n-i) + g(u(n), u(n - 1), \ldots, u(n - M + 1)) \quad (19.65)$$

where $g(\cdot, \cdot, \cdots, \cdot)$ is a nonlinear function that is differentiable with respect to its arguments. It is assumed that $M \leq N$ for all four models.

**Model II.** The output $y(n+1)$ at time $n+1$ depends nonlinearly on $N$ past values of the output and linearly on $M$ past values of the input, as shown by

$$y(n+1) = f(y(n), y(n-1), \ldots, y(n-N+1)) + \sum_{i=0}^{M-1} \beta_i u(n-i) \tag{19.66}$$

where $f(\cdot, \cdot, \cdots, \cdot)$ is another nonlinear function that is also differentiable with respect to its arguments.

**Model III.** The output $y(n+1)$ at time $n+1$ depends nonlinearly on past values of both the output and the input in a separable manner, as shown by

$$\begin{aligned} y(n+1) = f(y(n), y(n-1), \ldots, y(n-N+1)) \\ + g(u(n), u(n-1), \ldots, u(n-M+1)) \end{aligned} \tag{19.67}$$

**Model IV.** The output at time $n+1$ depends nonlinearly on past values of both the output and the input in a nonseparable manner, as shown by

$$y(n+1) = f(y(n), y(n-1), \ldots, y(n-N+1); u(n), u(n-1), \ldots, u(n-M+1)) \tag{19.68}$$

Clearly, Model IV is the most general one, in that it includes the other three models as special cases. However, in spite of its generality, model IV is the least tractable in analytic terms, which makes the other three models more attractive for practical applications (Narendra and Pasthasarathy, 1990).

Figure 19.10 presents block diagram descriptions of the four models. The elements labeled $z^{-1}$ at the input and output ends of each model represent unit-delay elements.

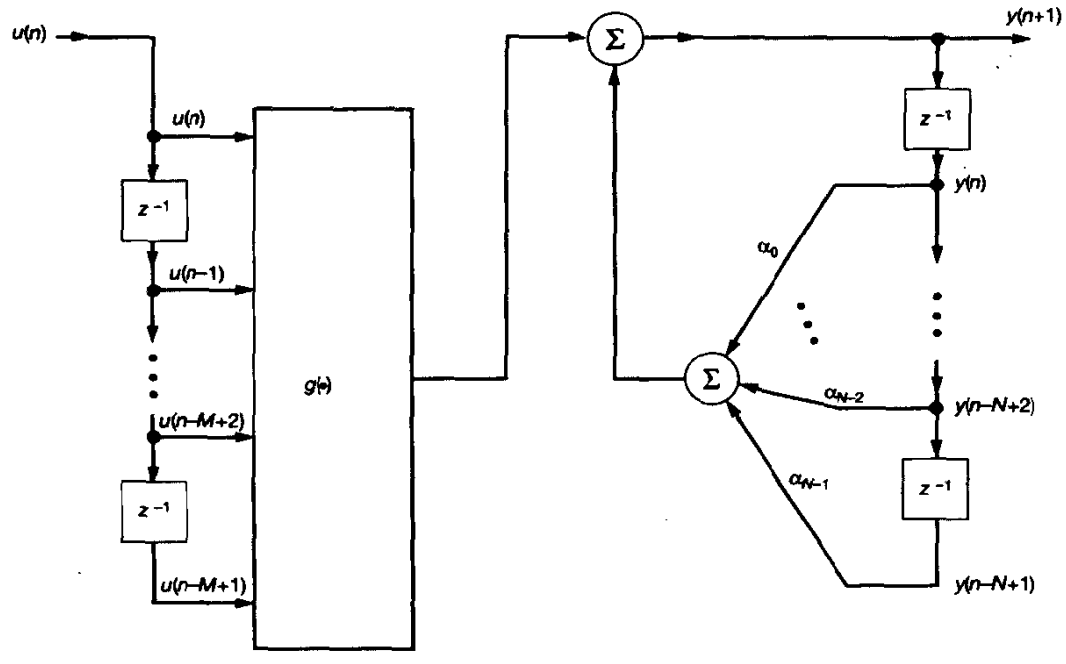
In light of the universal approximation theorem described in Section 19.5, we may say that under fairly weak conditions on the nonlinear function $f$ and/or $g$ in Eqs. (19.65) to (19.68), multilayer perceptrons can indeed be designed using the back-propagation algorithm to approximate the input-output mapping described by models I to IV over compact sets (Narendra and Pasthasarathy, 1990). Thus, although the multilayer perceptron is a static network by itself, it assumes a dynamic behavior by embedding it in models I through IV, described in Fig. 19.10. The choice of a particular model is dictated by the application of interest.

It is of interest to note that if the coefficients, $\alpha_i$, $i = 0, 1, \ldots, N-1$, were all to be reduced to zero, then Eq. (19.65) takes the form

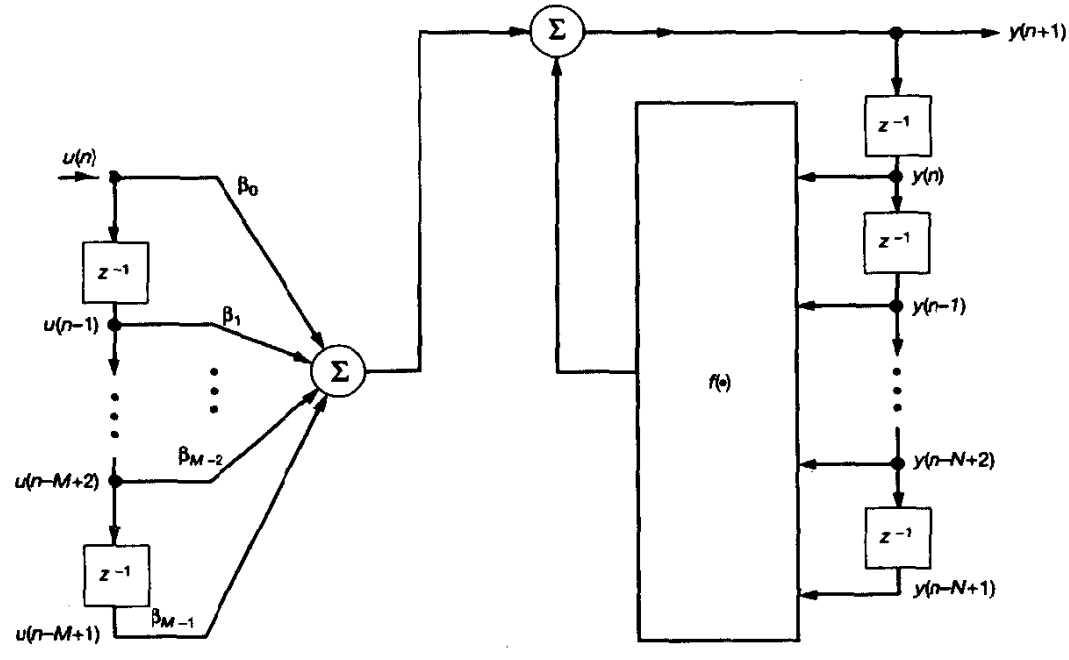
$$y(n+1) = g(u(n), u(n-1), \ldots, u(n-M+1)) \tag{19.69}$$

The model output $y(n+1)$ is now recognized as the one-step prediction

$$y(n+1) = \hat{u}(n+1 \mid \mathcal{U}_n) \tag{19.70}$$

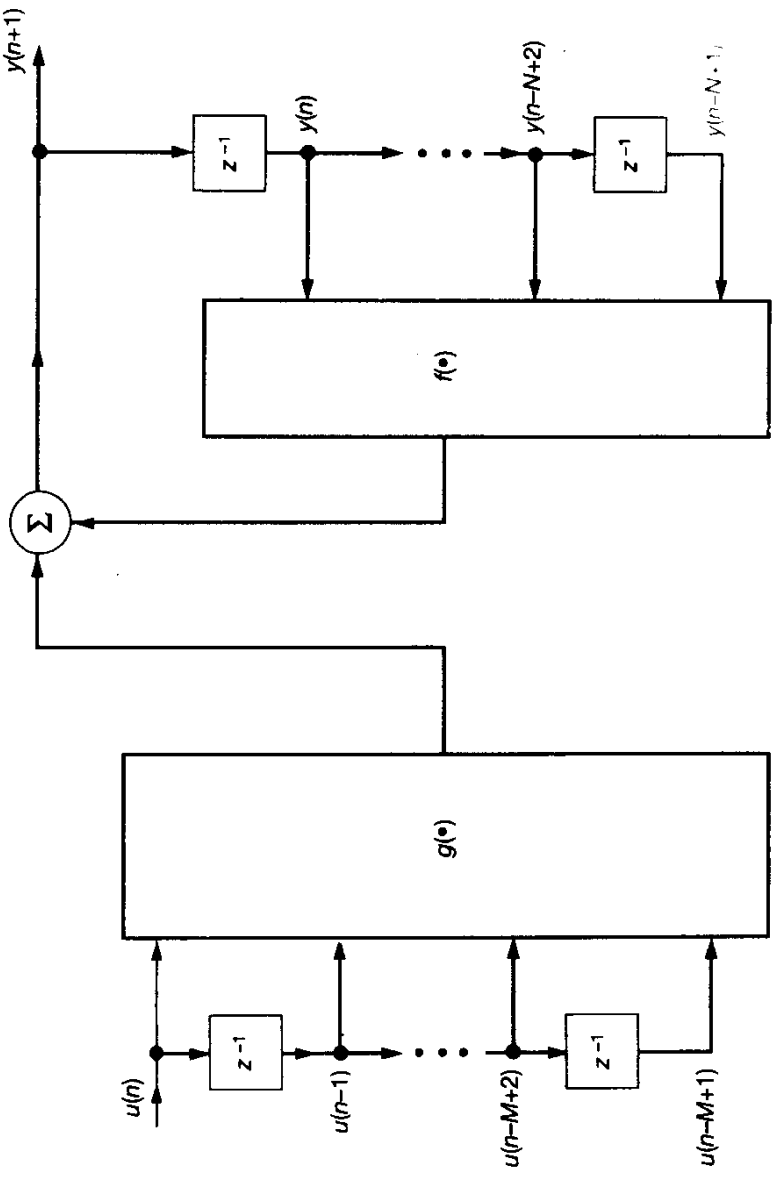



(a)

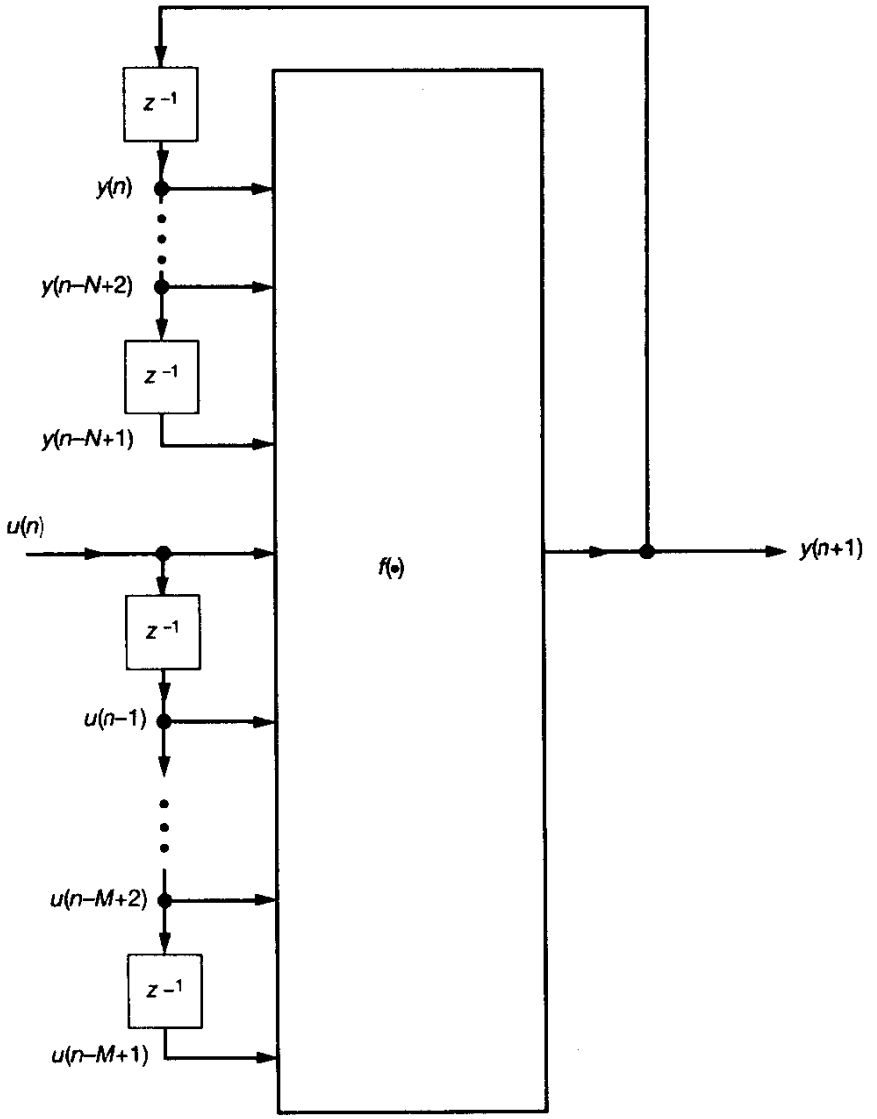



(b)

**Figure 19.10** Four different models for system identification. Parts (c) and (d) of the figure are presented on the next two pages.




(c)




(d)

**Figure 19.10** (concluded)

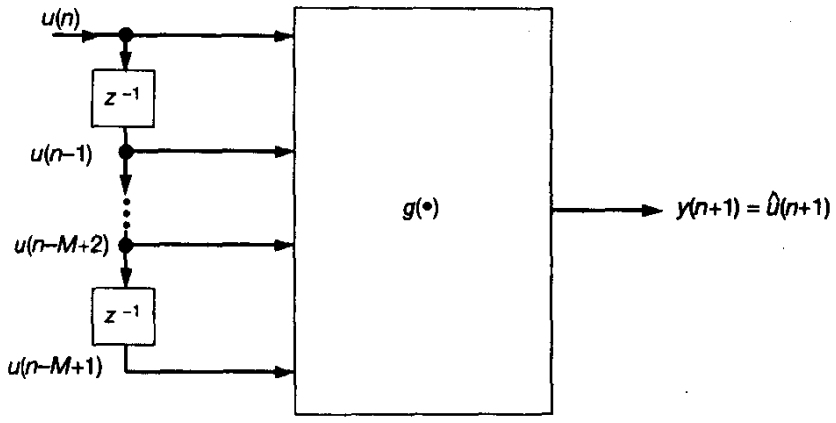



**Figure 19.11** Nonlinear one-step predictor.

where $\mathcal{U}_n$ denotes the $M$-dimensional space defined by past values of the input, $u(n)$, $u(n-1), \ldots, u(n-M+1)$. Accordingly, model II includes the nonlinear predictor as a special case, as depicted in the block diagram of Fig. 19.11.

### Time-Delay Neural Network

The models described in Fig. 19.10 describe a straightforward method for extending the use of a multilayer perceptron to account for *time*, which is accomplished by the inclusion of *memory* in the form of unit-delay elements outside of the network. In another structure known as the *time-delay neural network (TDNN)*, time delays are incorporated inside the network, as depicted in Fig. 19.12. The TDNN consists of a multilayer feedforward network, whose hidden neurons and output neurons are all *replicated across time*. It was originally devised by Lang and Hinton (1988) to capture explicitly the notion of time symmetry as encountered in the recognition of an isolated word (phoneme) using a *spectrogram*. The *spectrogram* is a two-dimensional image in which the vertical dimension corresponds to frequency and the horizontal dimension corresponds to time; the intensity (darkness) of the image corresponds to signal energy (Rabiner and Schafer, 1978). In effect, the spectrogram provides a method for making speech "visible."

Figure 19.12(a) illustrates a single-layer hidden version of the TDNN. For the example considered here (Lang and Hinton, 1988), the input layer consists of $16 \times 12 = 192$ sensory nodes encoding the spectrogram. The hidden layer consists of 10 copies of 8 hidden neurons. The output layer consists of 6 copies of 4 output neurons. The various replicas of a hidden neuron apply the same set of synaptic weights to narrow (3 time-step) windows of the spectrogram. Similarly, the various replicas of an output neuron apply the same set of synaptic weights to narrow (5 time-step) windows of the pseudospectrogram computed by the hidden layer. Figure 19.12(b) presents a time-delay interpretation of the replicated neural network of Fig. 19.12(a), hence the name "time-delay neural network." For the example of a single hidden layer considered here, the TDNN has a total of 544 synaptic weights. In a more elaborate structure described by Waibel et al. (1989), the

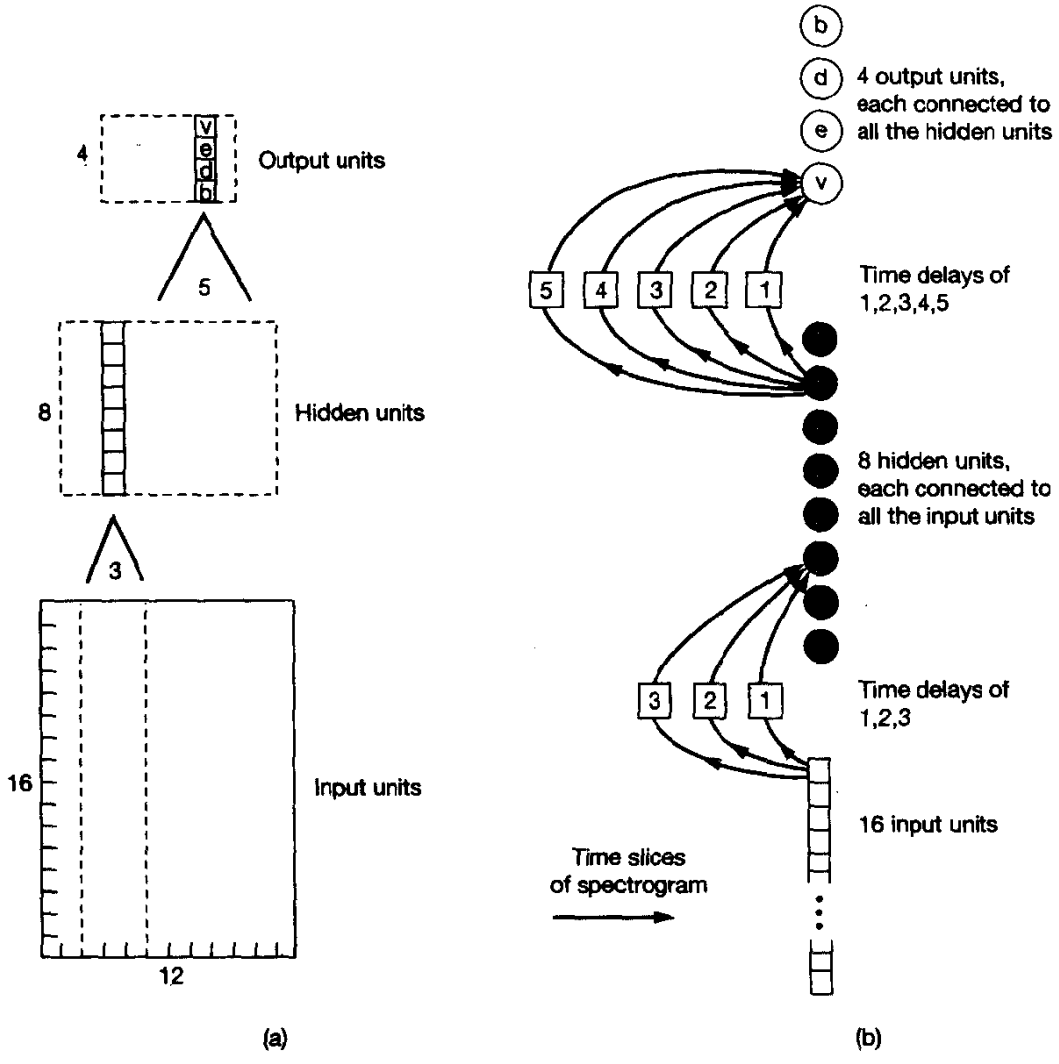



**Figure 19.12** (a) A three-layer network whose hidden units and output units are replicated across time. (b) Time-delay neural network (TDNN) representation. (From K.J. Lang and G.E. Hinton, 1988 with permission.)

TDNN is expanded to include two hidden layers. In any case, the standard back-propagation algorithm may be used to train the TDNN.

The TDNN has been used by several investigators for speech recognition (Lang and Hinton, 1988; Waibel et al., 1989). In this context, it appears that the temporal processing power of the TDNN lies in its ability to develop shift-invariant internal representations of speech and to use them for making "optimal" classifications. Another useful application of the TDNN for acoustic echo cancelation is described in Birkett and Goubran (1995).

Specifically, the TDNN is used to model the system nonlinearities and acoustic path in a "hands-free telephone" environment. Simulations are presented therein, demonstrating that such a nonlinear model can provide a significant improvement in system performance over a linear acoustic echo canceler using the normalized LMS algorithm.

The TDNN topology is in fact embodied in a multilayer perceptron in which each synapse is represented by a finite-duration impulse response (FIR) filter. In such a generalization, known as the *FIR multilayer perceptron*, time takes on a "distributed" representation at the synaptic level, thereby enhancing the temporal signal-processing power of the multilayer perceptron in a significant way. To train the FIR multilayer perceptron, we may *unfold it in time* and thereby develop an equivalent structure in the form of a static multilayer perceptron of much larger size, to which the standard back-propagation algorithm may be applied in the usual way. However, such a procedure is highly inefficient. A more practical approach is to use a *temporal back-propagation algorithm* devised by Wan (1990), which works directly with the FIR multilayer perceptron.[3]

### Target Detection in Clutter

For our third application we have chosen the detection of a radar target signal buried in a background of clutter. In radar terminology, *clutter* refers to reflections (echoes) of the transmitted signal produced by unwanted objects. In such a situation, the clutter is typically dominant, not only overpowering the receiver noise but also the wanted target signal. We thus have a binary hypothesis testing problem that may be described essentially as follows:

- *Hypothesis that a target is present*, in which the received signal $u(n)$ at time $n$ consists of a target signal $s(n)$ plus clutter $c(n)$, as shown by

$$u(n) = s(n) + c(n)$$

- *Null hypothesis*, in which the received signal $u(n)$ consists of clutter alone, as shown by

$$u(n) = c(n)$$

In the traditional approach to the detection problem described here, a parametric model is formulated for the clutter process and a detection strategy (e.g., Neyman–Pearson criterion) is used to solve the problem. However, with such an approach it is difficult to account fully for an inherent characteristic of radar clutter, that it is in reality the product of a *nonlinear dynamical process*. Indeed, a detailed experimental study reported in Haykin and Li (1995), using real-life radar data, has shown that sea clutter (i.e., radar backscatter from an ocean surface) is largely chaotic. A *chaotic process* is the result of a

[3] For a detailed discussion of temporal processing using the multilayer perceptron and other neural networks, see Haykin (1994).

deterministic mechanism, but it exhibits many of the characteristics ordinarily associated with a stochastic process. The important point to note here is that radar clutter is deterministically predictable in a short-term sense.

Recognizing that learning is a natural attribute of neural networks, we may propose a new strategy for the detection of a radar target signal in clutter as follows (Li and Haykin, 1993; Haykin and Li, 1995):

- Starting with actual clutter data that are representative of the environment of interest, a neural network such as a multilayer perceptron is trained (using the back-propagation algorithm) as a *one-step predictor*. Provided that the network is of the right size and the training data set is large enough, the prediction error produced at the output of the network, under the null hypothesis should closely approximate the sample function of a white Gaussian noise process. In effect, the network is trained to perform the function of a clutter model.
- When the network is fed with a received signal that consists of a target signal plus clutter, the presence of the target signal in the input causes a corresponding perturbation at the output of the network. That is, the network tends to preserve the essential characteristics of the target signal at its output. Thus, under the hypothesis that a target is present, the output signal consists essentially of a component identifiable with the target of interest, superimposed on a white Gaussian noise background.

The novelty of the detection strategy described here lies in the fact that a difficult signal detection problem is transformed into the detection of an unknown signal in additive white Gaussian noise, which may be viewed as the communication theorist's dream. The complete receiver thus consists of a one-step nonlinear predictor followed by a conventional constant false-alarm rate (CFAR) processor, as depicted in Fig. 19.13. The important advantages of this receiver include the following:

- Weak statistical assumptions about the environment in which the radar operates
- Inherent ability to account for nonlinear characteristics of the received radar signal

Most importantly, in a clutter dominated environment, the receiver of Fig. 19.13 has the potential to outperform a conventional radar receiver.

Figure 19.14, taken from Haykin et al. (1995b), shows the results of applying the neural-network approach described herein to an operational marine environment using a noncoherent radar. Specifically, Fig. 19.14(a) shows a sample azimuthal time series taken along a range ring containing a 10 square-meter target. The corresponding output of the neural network is shown in Fig. 19.14(b). As can be readily observed, the neural network has captured the dynamics of the clutter, such that the learned clutter component has been effectively removed, yet the target gives a significant response. Fig. 19.14 thus clearly

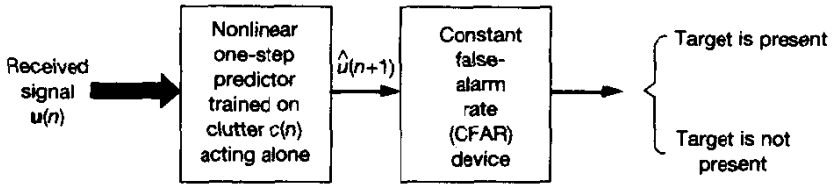



**Figure 19.13** Neural network-based radar receiver.

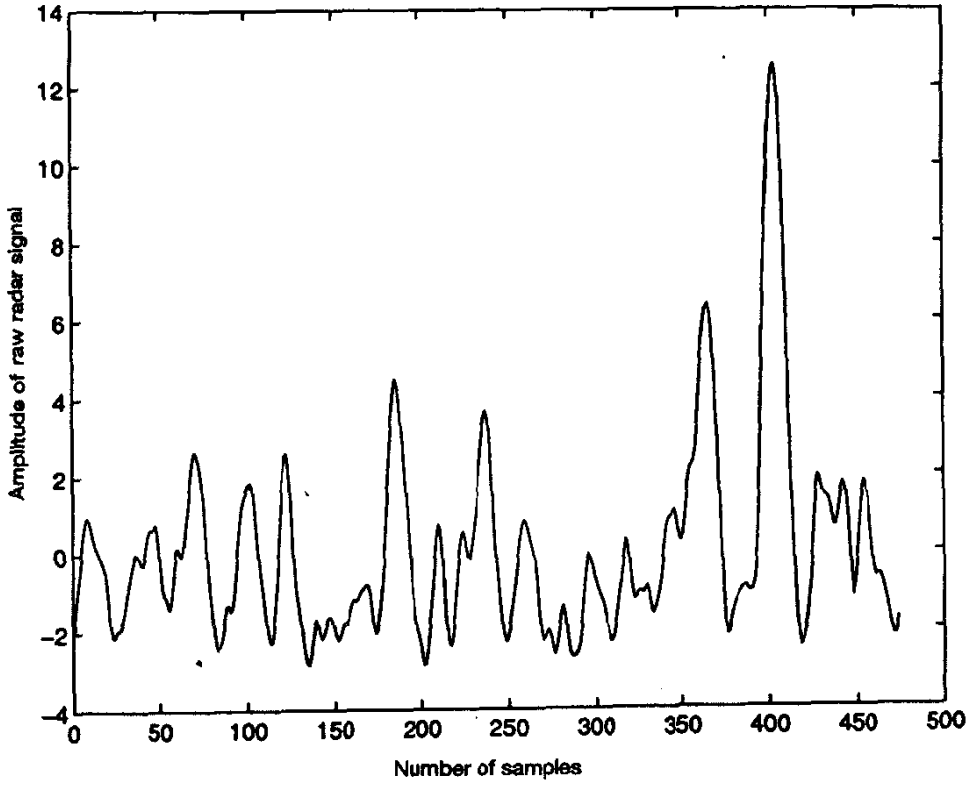



**Figure 19.14** (a) Azimuthal time series consisting of target signal plus clutter.

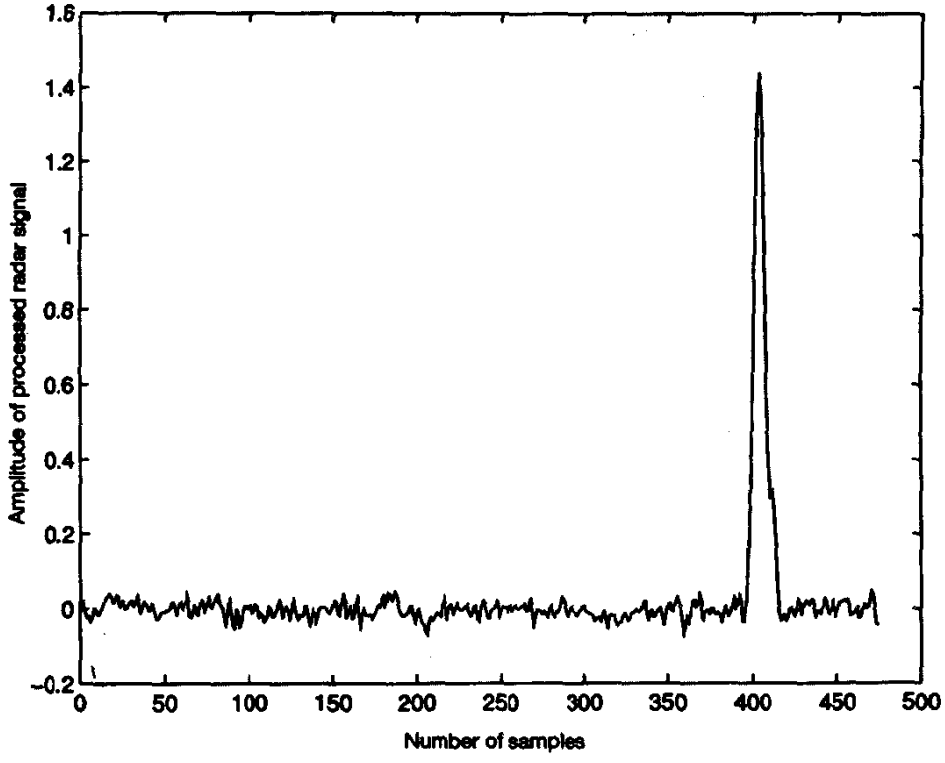



**Figure 19.14** (b) Output of neural network trained as a predictive model of clutter.

illustrates the action of the neural network based predictive model as a radar clutter canceler.

## 19.8 SUMMARY AND DISCUSSION

Just as the LMS algorithm has established itself as the workhorse of linear adaptive filters, so it is with the back-propagation algorithm in the context of neural networks. The back-propagation algorithm is relatively simple to implement, which has made it the most popular algorithm in use today for the design of neural networks. In particular, *it provides a powerful device for storing the information content of the training data in the synaptic weights of the network.* As long as the dataset used to train the neural network is large enough to be representative of the environment in which the network is embedded, the net-

work develops the capability to *generalize*. Specifically, the network delivers a "satisfactory" performance when it is fed with test data drawn from the same input space as the training data but not previously seen by the network.

The multilayer perceptron, consisting of a feedforward network with one or more hidden layers, is the neural network structure commonly used in conjunction with the back-propagation algorithm. With terminology in mind, it is wrong to speak of a "back-propagation network." Rather, we have a multilayer perceptron as the neural network, which is trained with the back-propagation algorithm.

Multilayer perceptrons have been applied successfully in a variety of diverse areas. In terms of functional tasks, the applications may be categorized as follows:

- Pattern classification (recognition)
- Control
- Signal processing

The ability of the multilayer perceptron to *learn* from its environment befits its use for these tasks, each one in its own specific way.

Back-propagation learning is an example of *supervised learning*, so called by virtue of the fact that the desired response (target signal) in the training data plays the role of a "teacher." The important issue to note here is that the learning process is *statistical* in nature. The reason for stochasticity is rooted in the environment in which the neural network is embedded. The net result is that the network is merely one form in which "empirical" knowledge about the environment is represented (White, 1989). The difficulties encountered in a study of the learning process are twofold (Haykin, 1994):

1. A neural network is *nonlinear*, which makes a detailed statistical analysis of the learning process to be a challenging undertaking.
2. In a neural network, *knowledge* about the environment is represented by the values taken on by the free parameters (synaptic weights and biases) of the network; the distributed nature of the knowledge stored in this manner in the network makes for a difficult interpretation.

Finally, from a computational point of view we should stress that the back-propagation algorithm is characterized by a slow rate of convergence. The problem becomes particularly serious when the requirement is to solve a large-scale task. In this context, we are once again reminded of analogy with the LMS algorithm, which is known for its own slow rate of convergence. Various procedures have been devised to accelerate the application of the back-propagation algorithm through the use of learning-rate adaptation (Jacobs, 1988). Alternatively, we may resort to the use of other supervised learning algorithms rooted in nonlinear system identification (Palmieri et al., 1991; Feldkamp, 1994) or function optimization theory (Battiti, 1992; Johansson et al., 1992). For a discussion of the issues raised herein, the reader is referred to the book by Haykin (1994).

## PROBLEMS

1. A neuron $j$ receives inputs from four other neurons whose activity levels are 10, −20, 4, and −2. The respective synaptic weights of neuron $j$ are 0.8, 0.2, −1.0, and −0.9. Calculate the output of neuron $j$ for the following two situations:
   (a) The neuron is linear.
   (b) The neuron is represented by the McCulloch–Pitts model.
2. Repeat Problem 1 for a neuron $j$ whose model is based on the logistic function

$$\varphi(\text{net}) = \frac{1}{1 + \exp(-\text{net})}$$

3. Consider a multilayer feedforward network, all the neurons of which operate in their linear regions. Justify the statement that such a network is equivalent to a feedforward network with a single layer of computation nodes.
4. Consider the following nonlinear functions:

   (a) $\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt$

   (b) $\varphi(x) = \frac{2}{\pi} \tan^{-1}(x)$

   Explain why both of these functions satisfy the properties of an activation function fitting the requirements of the universal approximation theorem. How do these two activation functions differ from each other?
5. The momentum constant $\alpha$ is normally assigned a positive value in the range $0 \leq \alpha < 1$. Justify the fact that $\alpha$ may also be assigned a negative value in the range $-1 < \alpha \leq 0$.
6. A time series is created using a discrete *Volterra model* of the form

$$u(n) = \sum_i a_i v(n-i) + \sum_i \sum_j a_{ij} v(n-i)\, v(n-j) + \cdots$$

   where $a_i$, $a_{ij}$, . . . are the Volterra coefficients, the $v(n)$ are samples of a white Gaussian noise sequence, and $u(n)$ is the model output. Using a neural network, construct an implemenation of this Volterra model made up as follows:
   (a) The linear term has coefficients corresponding to $i = 1, 2, 3$.
   (b) The quadratic term has coefficients corresponding to $i,j = 1, 2$.
   (c) The cubic and all higher-order terms are zero.
7. The risk $\mathcal{R}$ defined in Eq. (19.63) has a form similar to that for the minimum-description length (MDL) criterion for stochastic model complexity. Discuss how these criteria are related.
8. Construct an FIR multilayer perceptron equivalent of the TDNN described in Fig. 19.12, in which each synapse consists of a simple FIR filter with a single coefficient and a single delay element.
9. In neural network terminology, a *recurrent network* is a network whose output is a function of both its input samples and past samples of the output. With this definition in mind, which of the networks described in Fig. 19.10 would qualify as a recurrent network?

   Can we refer to the TDNN as a recurrent network? Why?

CHAPTER

# 20

# Radial Basis Function Networks

The training process of a neural network may be viewed as one of *curve fitting*. In particular, we are given a set of data points in the observation space defined by specified values of the input signal and a desired response (target signal), and the requirement is to find an input–output mapping that passes through these points. In a corresponding way, the generalization process may be viewed as one of *interpolation*, in that the network is called upon to express its response to test data never seen before. This viewpoint is exploited in the design of another important type of neural network known as a *radial basis function (RBF) network* (Broomhead and Lowe, 1988). The RBF network is a multilayer feedforward network with a single layer of hidden units which operate as "kernel" nodes. As such, it represents an alternative to the multilayer perceptron. Advantages of RBF networks over multilayer perceptrons trained with the back-propagation algorithm include a more straightforward training process, and a simpler network structure.

Ordinarily, the development of RBF networks is pursued assuming real data and real free parameters. In the study of RBF networks presented in this chapter, we will consider the more general case of *complex RBF networks*, which maintains the precise formulation and elegant structure of complex signals as encountered in radar, sonar, and communication systems (Chen et al., 1994). Naturally, the treatment presented herein includes real RBF networks as a special case.

We begin the discussion by considering the structure of RBF networks, emphasizing the features that distinguish them from multilayer perceptrons.

## 20.1 STRUCTURE OF RBF NETWORKS

A radial basis-function (RBF) network consists of an input layer of source nodes, a single hidden layer of nonlinear processing units, and an output layer of linear weights, as depicted in Fig. 20.1. Using the outputs computed by the hidden layer in response to an input vector in combination with a desired response presented to the output layer, the weights are trained in a supervised fashion using an appropriate linear filtering method, thereby providing a bridge between linear adaptive filters and neural networks.

RBF networks differ from multilayer perceptrons in the following structural/operational respects:

- RBF networks have a single hidden layer, whereas multilayer perceptrons may have one or more hidden layers.
- In RBF networks, the transfer functions connecting the input layer to the hidden layer are nonlinear and those connecting the hidden layer to the output layer are linear. In multilayer perceptrons, the transfer functions of each hidden layer con-

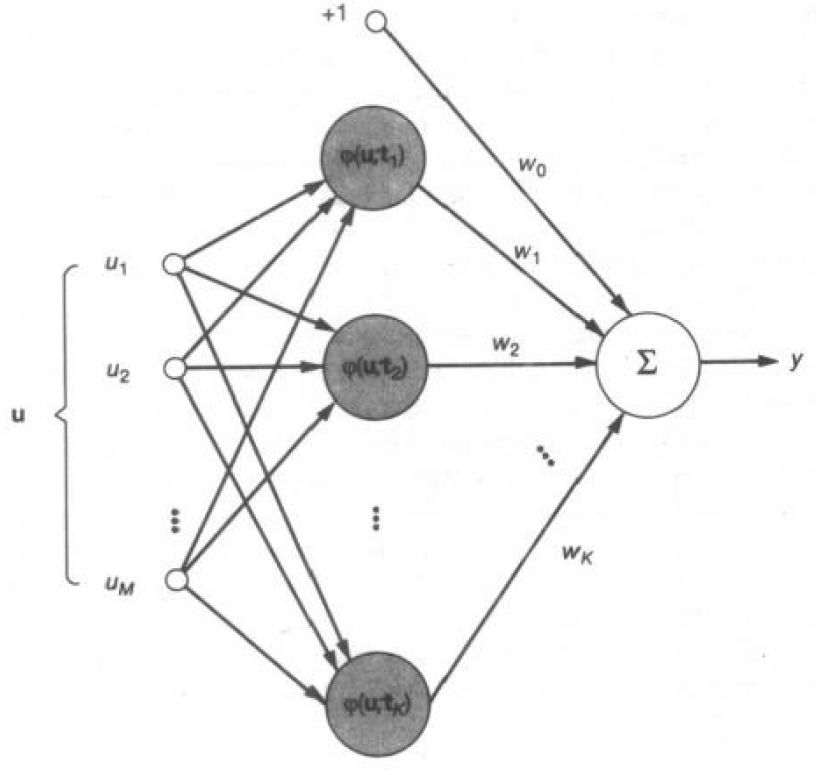



**Figure 20.1** RBF network.

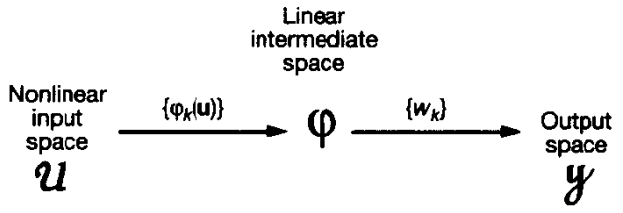



**Figure 20.2** Illustrating the transformations involved in an RBF network.

necting it to the previous layer are all nonlinear, and the transfer functions of the output layer may be nonlinear or linear, depending on the application of interest.

- Each hidden unit of an RBF network computes a distance function between the input vector and the center of a *radial basis function* characterizing that particular unit. On the other hand, each neuron of a multilayer perceptron computes the inner product (dot product) of the input vector applied to that neuron and the vector of associated synaptic weights.

RBF networks and multilayer perceptrons do, however, share a common property: they are both universal approximators of the feedforward type. Naturally, they perform their input–output mapping in different ways, as explained later.

Without loss of generality, the RBF network of Fig. 20.1 is shown to have a single output node. Using the terminology of this figure, we may describe the input–output mapping performed by the RBF network as follows:

$$y = \sum_{k=1}^{K} w_k \varphi(\mathbf{u}; \mathbf{t}_k) + w_0 \tag{20.1}$$

The term $\varphi(\mathbf{u}; \mathbf{t}_k)$ is the $k$th radial-basis function (kernel) that computes the "distance" between an input vector $\mathbf{u}$ and its own center $\mathbf{t}_k$; the output signal produced by the $k$th hidden unit (also referred to as the kernel node) is a *nonlinear function* of that distance. The scaling factor $w_k$ in Eq. (20.1) represents a *complex weight* that connects the $k$th hidden node to the output node of the network. Finally, the constant term $w_0$ in Eq. (20.1) represents a *bias* that may be complex.

The input–output mapping performed by the RBF network is accomplished in two stages, as depicted in Fig. 20.2:

- A *nonlinear transformation*, which maps the complex-valued input space $\mathcal{U}$ onto a real-valued intermediate space $\varphi$
- A *linear transformation*, which maps the intermediate space $\varphi$ onto the complex-valued output space $\mathcal{Y}$

The nonlinear transformation is defined by the set of radial-basis functions $\phi_k$, and the linear transformation is defined by the set of weights $w_k$, $k = 1, 2, \ldots, K$.

For the second transformation to be effective, the vector of random variables produced by the hidden layer should desirably represent a "linear process." How to test for the linearity of this process is an open problem.[1]

For the present it suffices to say that "linearization" of the input space is highly likely if the dimensionality of the intermediate space $\varphi$ (i.e., the number of radial-basis functions) is large enough compared to the dimensionality of the input space. This observation is made in view of an earlier result by Cover (1965) on nonlinear separability of patterns in the context of pattern classification.

## 20.2 RADIAL-BASIS FUNCTIONS

At the heart of an RBF network is the hidden layer that is defined by a set of *radial-basis functions*, from which the network derives its name. Typical examples of real-valued radial-basis functions are the following (Broomhead and Lowe, 1988; Poggio and Girosi, 1990):

1. *Thin-plate-spline function*:

$$\varphi(r) = \left(\frac{r}{\sigma}\right)^2 \ln\left(\frac{r}{\sigma}\right) \qquad \text{for some } \sigma > 0, \text{ and } r \geq 0 \tag{20.2}$$

2. *Gaussian function*:

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad \text{for some } \sigma > 0, \text{ and } r \geq 0 \tag{20.3}$$

Of these two examples, the Gaussian function is the one most commonly used in practice. In the remainder of this chapter, we will confine our discussion to the use of Gaussian functions.

The selection of a radial-basis function for a complex-valued RBF network is in fact the same as that for a real-valued RBF network, with some minor modifications. Specifically, given an input vector $\mathbf{u}$, the $k$th Gaussian radial-basis function of the RBF network is defined by

$$\varphi(\mathbf{u}; \mathbf{t}_k) = \exp[-(\mathbf{u} - \mathbf{t}_k)^H \mathbf{C}_k(\mathbf{u} - \mathbf{t}_k)], \qquad k = 1, 2, \ldots, K \tag{20.4}$$

[1]Tugnait (1994), building on earlier work by Subba Rao and Gabr (1980, 1984), presents an approach based solely on the bispectrum of the input data to test for linearity of a stationary time series. The stochastic model used for this approach assumes the possible presence of an additive noise component. In the case of an RBF network acting on "noisy" data, the noise appearing at the output of the hidden layer may be of a multiplicative kind due to the highly nonlinear nature of the processing units in that layer. The applicability of Tugnait's method to the hidden layer of an RBF network for noisy input data may therefore be somewhat uncertain.

where the vector $\mathbf{t}_k$ defines the center of the $k$th radial basis function and the matrix $\mathbf{C}_k$ defines its *width* or *smoothing factor*;[2] the superscript $H$ denotes Hermitian transposition. Using the concept of the *Mahalanobis metric (distance)*, we may rewrite Eq. (20.4) in the compact form

$$\varphi(\mathbf{u}; \mathbf{t}_k) = \exp(-\|\mathbf{u} - \mathbf{t}_k\|^2_{Mc}), \qquad k = 1, 2, \ldots, K \tag{20.5}$$

A simple choice for the matrix $\mathbf{C}_k$ is the diagonal matrix

$$\mathbf{C}_k = \frac{1}{\sigma_k^2}\mathbf{I} \tag{20.6}$$

where $\mathbf{I}$ is the identity matrix. On this basis, we may redefine the $k$th radial-basis function as follows:

$$\varphi(\mathbf{u}; \mathbf{t}_k) = \exp\left(-\frac{1}{\sigma_k^2}\|\mathbf{u} - \mathbf{t}_k\|^2\right), \qquad k = 1, 2, \ldots, K \tag{20.7}$$

where $\mathbf{t}_k$ is the center, $\sigma_k$ is the width, and $\|\mathbf{u} - \mathbf{t}_k\|$ denotes the Euclidean distance between $\mathbf{u}$ and $\mathbf{t}_k$. Note that $\varphi(\mathbf{u};\mathbf{t}_k)$ is *radially symmetric* in the sense that

$$\varphi(\mathbf{u}_i; \mathbf{t}_k) = \varphi(\mathbf{t}_k; \mathbf{u}_i) \qquad \text{for all } i \text{ and } k$$

Thus, substituting Eq. (20.7) in (20.1), we may formulate the input–output mapping realized by a Gaussian RBF network as follows:

$$y = \sum_{k=1}^{K} w_k \exp(-\frac{1}{\sigma_k^2}\|\mathbf{u} - \mathbf{t}_k\|^2) \tag{20.8}$$

From a design point of view, the requirement is to select suitable values for the parameters of each of the $K$ Gaussian radial-basis functions, namely $\sigma_k$ and $\mathbf{t}_k$, $k = 1, 2, \ldots, K$, and solve for the weights of the output layer. In the sequel, we describe three different procedures for the design of a Gaussian RBF network, each with its own merit.

## 20.3 FIXED CENTERS SELECTED AT RANDOM

The simplest approach for the design of an RBF network involves selecting a set of *fixed* radial basis functions for the hidden units of the network. In particular, the locations of the centers may be chosen *randomly* from the training data set. Such an approach, first described by Broomhead and Lowe (1988), is considered to be "sensible," since random sampling would distribute the centers according to the probability density function of the

[2]From analogy with a multivariate complex Gaussian distribution (Miller, 1974), the vector $\mathbf{t}_k$ and the matrix $\mathbf{C}_k$ in Eq. (20.4) play the roles of a mean vector and the inverse of a covariance matrix. A discussion of complex Gaussian functions is presented in Chapter 2.

training data. This assumes that the training data are distributed in a representative manner for the problem at hand. We may thus write

$$\varphi(\mathbf{u}; \mathbf{t}_k) = \exp\left(-\frac{K}{d_{\max}^2}\|\mathbf{u} - \mathbf{t}_k\|^2\right), \qquad k = 1, 2, \ldots, K \tag{20.9}$$

where $K$ is the number of centers, and $d_{\max}$ is the maximum distance between the chosen centers. In effect, the width $\sigma_k$ for each Gaussian radial-basis function is fixed at the common value

$$\sigma = \frac{d_{\max}}{\sqrt{K}} \tag{20.10}$$

This formula ensures that the individual functions are not too peaked or too flat; clearly both of these two extremes are to be avoided. Alternatively, we may use individually scaled centers with broader widths in areas of lower data density.

Having fixed the radial-basis functions, we then move on to compute the weights in the output layer of the RBF network. For this computation, we may use the *method of least squares* (described in Chapter 11), which is of a *block (batch) processing* kind. Let the training set be denoted by $\{\mathbf{u}_i, d_i\}$, where $\mathbf{u}_i$ denotes the input vector and $d_i$ denotes the desired response belonging to the $i$th example, with $i = 1, 2, \ldots, N$. We may then define the following matrix and vector quantities:

$$\Phi = \begin{bmatrix} 1 & \varphi(\mathbf{u}_1; \mathbf{t}_1) & \varphi(\mathbf{u}_1; \mathbf{t}_2) & \cdots & \varphi(\mathbf{u}_1; \mathbf{t}_K) \\ 1 & \varphi(\mathbf{u}_2; \mathbf{t}_1) & \varphi(\mathbf{u}_2; \mathbf{t}_2) & \cdots & \varphi(\mathbf{u}_2; \mathbf{t}_K) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \varphi(\mathbf{u}_N; \mathbf{t}_1) & \varphi(\mathbf{u}_N; \mathbf{t}_2) & \cdots & \varphi(\mathbf{u}_N; \mathbf{t}_K) \end{bmatrix} \tag{20.11}$$

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T \tag{20.12}$$

The real-valued matrix $\Phi$, called an *interpolation matrix*, is of size $N$-by-$(K + 1)$, where $N$ is the number of training examples and $K$ is the number of radial-basis functions; the first column of unity terms is included in this matrix to account for the use of a bias. The *desired response vector* $\mathbf{d}$ *is of size* $N$-by-1.

Evaluating Eq. (20.1) for each of the $N$ examples in the training set, we may write

$$y_i = \sum_{k=1}^{K} w_k \varphi(\mathbf{u}_i; \mathbf{t}_k) + w_0, \qquad i = 1, 2, \ldots, N \tag{20.13}$$

Using the matrix notation of Eq. (20.11), we may rewrite the set of $N$ equations (20.13) in the compact form:

$$\mathbf{y} = \Phi \mathbf{w} \tag{20.14}$$

where $\mathbf{y}$ is the $N$-by-1 *output vector*:

$$\mathbf{y} = [y_1, y_2, \ldots, y_N]^T \tag{20.15}$$

and $\mathbf{w}$ is the $(K + 1)$-by-1 weight vector:

$$\mathbf{w} = [w_0, w_1, w_2, \ldots, w_K]^T \tag{20.16}$$

According to the definitions of Eqs. (20.11) and (20.16), the bias term may be viewed as a weight $w_0$ connected to an input $\varphi_0$ fixed at $+1$, as indicated in Fig. 20.1.

Suppose that during training, the RBF network output vector is constrained to equal the desired response vector:

$$\mathbf{y} = \mathbf{d} \tag{20.17}$$

We may then rewrite Eq. (20.14) as[3]

$$\mathbf{d} = \boldsymbol{\Phi}\,\mathbf{w} \tag{20.18}$$

With $N > K$, Eq. (20.18) represents an *overdetermined* system of equations in that we have more equations than unknowns. To solve Eq. (20.18) for the weight vector $\mathbf{w}$, we may use the method of least squares. In particular, a robust solution for $\mathbf{w}$ is provided by the *minimum norm solution*, written as follows (see Eq. (11.13)):

$$\mathbf{w} = \boldsymbol{\Phi}^{+} \tag{20.19}$$

where $\boldsymbol{\Phi}^{+}$ is the pseudoinverse of the interpolation matrix $\boldsymbol{\Phi}$. The recommended procedure for computing the pseudoinverse matrix $\boldsymbol{\Phi}+$ is to use the method of singular value decomposition (SVD) described in Chapters 11 and 12.

Summarizing, the *method of fixed centers* based on batch (block) processing proceeds as follows:

1. For a specified number of radial-basis factors, $K$, select the centers (and therefore their widths) randomly from the training data. Hence, using a Gaussian model, define the radial-basis functions in accordance with Eq. (20.9).
2. Use Eq. (20.11) to determine the interpolation matrix $\boldsymbol{\Phi}$ for the given set of $N$ training examples.
3. Compute the weight vector of the output layer using Eq. (20.19).

As an alternative to the block processing method used to compute the weight vector $\mathbf{w}$, we may use an iterative procedure such as the LMS algorithm described in Chapter 9 or the RLS algorithm described in Chapter 13.

[3]For *strict interpolation*, we should have $K + 1 = N$. In this case, $\boldsymbol{\Phi}$ assumes the form of a square matrix. Equation (20.18) may then be solved for $\mathbf{w}$, as shown by

$$\mathbf{w} = \boldsymbol{\Phi}^{-1}\,\mathbf{d}$$

where $\boldsymbol{\Phi}^{-1}$ is the inverse of matrix $\boldsymbol{\Phi}$. Although, in theory, we are always assured of a solution to the strict interpolation problem, in practice we cannot always solve for $\mathbf{w}$ particularly when the matrix $\boldsymbol{\Phi}$ is arbitrarily close to singular. Moreover, for large $N$, the size of the RBF network becomes prohibitively expensive to implement. Both of these problems are overcome by choosing the number of centers $K$ small compared to the size $N$ of the training data, in which case $\boldsymbol{\Phi}$ assumes the form of a rectangular matrix. In a strict sense, when $K < N$ the matrix $\boldsymbol{\Phi}$ is no longer an interpolation matrix.

## 20.4 RECURSIVE HYBRID LEARNING PROCEDURE

The main problem with the use of fixed centers just described for the design of an RBF network is the fact it may require a large dataset for a prescribed level of performance. One way of overcoming this limitation is to use a *hybrid learning procedure*, which combines the following:

- *Self-organized learning algorithm* for the selection of the centers of the radial-basis functions in the hidden layer
- *Supervised learning algorithm* for the computation of the weights in the output layer

Although block (batch) processing can be used to implement these two operations, it is particularly advantageous to take an adaptive (iterative) approach. For example, we may use the $k$-means clustering algorithm[4] (among others) for the self-organized learning part of the hybrid procedure. As for the supervised learning part, we may use the RLS or LMS algorithm, depending on complexity requirements.

The *k-means clustering algorithm* computes $k$ centers and thereby partitions the input data into $k$ clusters (Duda and Hart, 1973). Specifically, it places the centers of the radial-basis functions in only those regions of the input space $\mathcal{U}$ where significant data are present. Let $K$ denote the number of radial-basis functions; the determination of a suitable value for $K$ may require experimentation. Let $\mathbf{t}_k(n)$, $k = 1, 2, \ldots, K$, denote the centers of the radial-basis functions at iteration $n$. Then, the $k$-means clustering algorithm may proceed as follows:[5]

1. *Initialization.* Choose random values for the initial centers $\mathbf{t}_k(0)$; the only restriction here is that the $\mathbf{t}_k(0)$ be different for $k = 1, 2, \ldots, K$. It may also be desirable to keep the Euclidean norm of the centers small.
2. *Sampling.* Draw a sample vector $\mathbf{u}$ from the input space $\mathcal{U}$ with a certain probability. The vector $\mathbf{u}$ represents the input applied to the RBF network.
3. *Similarity matching.* Find the best-matching (winning) center $\tilde{k}(\mathbf{u})$ at iteration $n$, using the minimum-distance Euclidean criterion:

$$\tilde{k}(\mathbf{u}) = \arg\min_k \|\mathbf{u}(n) - \mathbf{t}_k(n)\|, \qquad k = 1, 2, \ldots, K \tag{20.20}$$

4. *Updating.* Adjust the locations of the centers, using the update rule

$$\mathbf{t}_k(n+1) = \begin{cases} \mathbf{t}_k(n) + \eta[\mathbf{u}(n) - \mathbf{t}_k(n)], & k = \tilde{k}(\mathbf{u}) \\ \mathbf{t}_k(n), & \text{otherwise} \end{cases} \tag{20.21}$$

where $\eta$ is the *learning-rate parameter* that lies in the range $0 < \eta < 1$.

[4]The use of the $k$-means clustering algorithm for the design of RBF network was first proposed by Moody and Darken (1989). Its use for complex RBF networks is discussed in Chen et al. (1994).

[5]The procedure described herein is a special case of a more general self-organized learning algorithm known as the *self-organizing feature map (SOFM)*, originally developed by Kohonen (1982, 1990).

A limitation of the conventional $k$-means algorithm described above is that it can only achieve a local optimum solution that depends on the initial choice of cluster centers. Consequently, computing resources may be wasted in that some initial centers get stuck in regions of the input space $\mathcal{U}$ with a scarcity of data points and therefore never move to new locations where they are needed. The net result is an unnecessarily large network. To overcome this limitation, Chen (1995) proposes the use of an *enhanced k-means clustering algorithm* due to Chirungrueng and Sequin (1994), which is based on a cluster variation-weighted measure that enables the algorithm to converge to an optimum or near-optimum configuration, independent of the initial center locations.

In any event, having identified the individual centers of the Gaussian radial-basis functions and their common width using the $k$-means algorithm or its enhanced version, we may move onto the output layer. If computational complexity is of no particular concern here, we may use the RLS algorithm or one of its variants to compute the weight vector $\mathbf{w}$ in the output layer. If, on the other hand, the requirement is to minimize computational complexity, the recommended procedure is to use the LMS algorithm. For a complex RBF network, the complex form of the RLS or LMS algorithm would naturally be used, with one important modification. Specifically, the vector of output signals produced by the hidden layer, which constitutes the input vector for the RLS or LMS algorithm, is *real-valued* in the present scenario. However, the weight vector $\mathbf{w}$ is complex-valued, since the RBF network is required to produce a complex-valued overall output to approximate the complex-valued desired response. Note also that the $k$-means clustering algorithm for the hidden layer and the RLS or LMS algorithm for the output layer can proceed with their own individual computations concurrently.

The hybrid approach described in this section and the method of fixed centers described in the previous section share a common feature: In both cases, the selection of centers in the hidden layer is decoupled from the design of linear weights in the output layer, which makes a theoretical understanding of what goes on inside the network somewhat difficult. This observation leads us to consider a fully supervised learning procedure, described next.

## 20.5 STOCHASTIC GRADIENT APPROACH

In the stochastic gradient approach for the design of an RBF network, the centers of the radial-basis functions and all other free parameters of th network undergo a *supervised learning process* (Lowe, 1989). In other words, the RBF network design takes on its more generalized form. A natural candidate for such a process is *error-correction learning*, which is most conveniently implemented using a stochastic gradient descent of the error criterion (Poggio and Girosi, 1990; Kassam and Cha, 1993), and whose basic concept is similar to the LMS algorithm.

The first step in the development of this supervised learning procedure is to define the instantaneous value of the cost function

$$\mathcal{E}(n) = \frac{1}{2}|e(n)|^2, \quad n = 1, 2, \ldots, N \tag{20.22}$$

**TABLE 20.1** SUMMARY OF THE STOCHASTIC GRADIENT ALGORITHM FOR THE DESIGN OF RBF NETWORKS USING COMPLEX-VALUED DATA

$$y(n) = \sum_{k=1}^{K} w_k(n)\phi(\mathbf{u}(n); \mathbf{t}_k(n))$$

$$e(n) = d(n) - y(n)$$

$$w_k(n+1) = w_k(n) \ \mu_w e^*(n)\, \varphi(\mathbf{u}(n); \mathbf{t}_k(n))$$

$$\mathbf{t}_k(n+1) = \mathbf{t}_k(n) + 2\mu_t e^*(n)\, w_k(n)\varphi(\mathbf{u}(n); \mathbf{t}_k(n)) \frac{\mathbf{u}(n) - \mathbf{t}_k(n)}{\sigma_k^2(n)}$$

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma\, e^*(n)\, w_k(n)\varphi(\mathbf{u}(n); \mathbf{t}_k(n)) \frac{\|\mathbf{u}(n) - \mathbf{t}_k(n)\|^2}{\sigma_k^2(n)}$$

where

$$\varphi(\mathbf{u}(n); \mathbf{t}_k(n)) = \exp\left(-\frac{1}{\sigma_k^2(n)}\|\mathbf{u}(n) - \mathbf{t}_k(n)\|^2\right)$$

where $e(n)$ is the error signal produced in response to the $n$th example, and $N$ is the total number of examples in the training set. The error signal is defined by

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= d(n) - \sum_{k=1}^{K} w_k(n) \exp\left(-\frac{1}{\sigma_k^2(n)}\|\mathbf{u}(n) - \mathbf{t}_k(n)\|^2\right) \end{aligned} \tag{20.23}$$

where $\mathbf{t}_k(n)$ is the center and $\sigma_k^2(n)$ is the squared width of the $k$th radial-basis function for example $n$, and $w_k(n)$ is the corresponding value of the $k$th weight in the output layer. The objective is to find the values of these free parameters that *minimize* $\mathscr{E}(n)$. The results of a stochastic gradient procedure aimed at this minimization are summarized in Table 20.1; the derivations of these results are presented as an exercise to the reader as Problem 3.

The following points are noteworthy in Table 20.1:

1. The cost function $\mathscr{E}(N)$, averaged over the entire set of $N$ training examples, is convex with respect to the weights $w_k(n)$ of the output layer, but nonconvex with respect to the centers $\mathbf{t}_k(n)$ and squared widths $\sigma_k^2(n)$ of the radial-basis functions in the hidden layer; in the latter case, the search for optimality may get stuck at a local minimum of the error-performance surface.
2. The update rules for $\mathbf{t}_k(n)$, $\sigma_k^2(n)$, and $w_k(n)$ are (in general) assigned different learning-rate parameters $\mu_t$, $\mu_\sigma$, and $\mu_w$, respectively.
3. Unlike the back-propagation algorithm, the stochastic gradient descent for the RBF network described herein does *not* involve the back-propagation of errors.
4. The gradient vector $\partial\mathscr{E}/\partial\mathbf{t}_k$ has an effect similar to a clustering effect that is task-dependent (Poggio and Girosi, 1990).

For the *initialization* of the stochastic gradient algorithm, the free parameters of the RBF network may be assigned a subset of values drawn at random from the training set. In so doing, we are building on an idea described in Section 20.3. In particular, the search in parameter space begins from a *structured* initial condition, in which case the likelihood of converging to an undesirable local minimum on the error-performance surface is reduced.

## 20.6 UNIVERSAL APPROXIMATION THEOREM (REVISITED)

In the previous chapter we presented a form of the universal approximation theorem that applies directly to multilayer perceptrons. RBF networks, however, differ from multilayer perceptrons, in that the activation functions of their hidden units (i.e., the radial-basis functions) have an argument that is *nonlinearly* dependent on the input vector $\mathbf{u}$. Hence, the universal approximation theorem as stated in Chapter 19 is not applicable to RBF networks.

In this section, we consider another form of the *universal approximation theorem* that is directly applicable to RBF networks. This issue, in the context of Gaussian hidden units, was apparently first considered by Brown.[6] Then it was reconsidered independently by Hartman et al. (1990), and in a broader setting by Park and Sandberg (1991).

Let $\mathcal{U}$ be any convex compact subset of $\mathbb{R}^M$. Let $\mathbf{u} \in \mathcal{U}$, and $\mathbf{t}_k \in \mathcal{U}$ for $k = 1, 2, \ldots, K$. Consider then a two-parameter family $\mathcal{F}$ of restricted Gaussian functions for real-valued data:

$$\varphi(\mathbf{u}; \mathbf{t}_k) = \exp\left(\frac{-\|\mathbf{u} - \mathbf{t}_k\|^2}{2\sigma_k^2}\right) \qquad \sigma_k > 0 \tag{20.24}$$

Let $\mathcal{L}$ be the set of all finite linear combinations of elements (with real coefficients) drawn from $\mathcal{F}$. Then, we may state the following theorem (Hartman et al., 1990; Park and Sandberg, 1991):

Any function in the algebra $C(\mathcal{U})$ of all continuous functions on $\mathcal{U}$ with the supremum norm can be uniformly approximated to an arbitrary accuracy by elements of $\mathcal{L}$.

In other words, RBF networks with Gaussian hidden units are universal approximators. The universal approximation theorem as stated herein is formulated in the context of real RBF networks. Its extension to complex RBF networks is intuitively obvious.

In light of the version of the universal approximation theorem stated here and the version of it stated in the previous chapter, we may now justifiably say that RBF networks and multilayer perceptrons are both universal approximators. Accordingly, it is not sur-

[6] In an appendix to a chapter contribution by Powell (1992), based on a lecture presented in 1990, credit is given to a result due to A. L. Brown. The result, apparently obtained in 1981, states that an RBF network can map an arbitrary function from a *closed* domain in $\mathbf{R}^M$ to $\mathbf{R}$.

prising to find that there always exists an RBF network capable of accurately mimicking a specified multilayer perceptron, or vice versa. However, these two neural networks perform their individual approximation tasks in entirely different ways. Multilayer perceptrons construct *global* approximations to nonlinear input–output mapping. Consequently, they are capable of extrapolation in regions of the input space where there is a scarcity of training data. In contrast, RBF networks construct *local* approximations to nonlinear input–output mapping, with the result that these networks are capable of fast learning and reduced sensitivity to the order of presentation of training data. In many cases, however, we find that in order to represent a mapping to some desired degree of smoothness, the number of radial-basis functions required to span the input space adequately may have to be very large. This problem, largely due to the number of available data points, becomes particularly acute in trying to solve large-scale problems such as image processing and speech recognition.

## 20.7 FILTERING APPLICATIONS

In light of the universal approximation theorem just discussed, it is apparent that many, if not all, of the signal processing applications that befit the use of multilayer perceptrons would befit RBF networks just as well, and vice versa. System identification and target detection, considered as possible applications of multilayer perceptrons in Section 19.7, may be equally served by means of RBF networks. By the same token, the first application of RBF networks selected for discussion in this section, namely, adaptive equalization, qualifies equally well for the use of multilayer perceptrons.[7]

### Adaptive Equalization

In the conventional form of an adaptive equalizer, based on the linear adaptive filter theory presented in Part III of the book, the equalizer operates as an inverse model. In particular, in the absence of noise and in the case of a minimum-phase channel, the cascade combination of the channel and the equalizer provides distortionless transmission. When, however, noise is present and/or the channel is nonminimum phase, the use of an inverse-model is no longer optimum.

An alternative viewpoint to that of inverse modeling is to approach the equalization process as a *pattern classification* problem (Theodoridis et al., 1992). For the simple case of bipolar data transmission, the received samples, corrupted by intersymbol interference (ISI) and noise, would have to be classified as $-1$ and $+1$. The equalizer now has the function of assigning each received sample to the correct decision region. According to this

[7]For the application of multilayer perceptrons (trained with the back-propagation algorithm) to simultaneous equalization and decoding of severe intersymbol interference due to data transmission over a communication channel, see Al-Mashouq and Reed (1994); the experimental results presented in that paper show a substantial improvement over conventional methods for equalization and decoding.

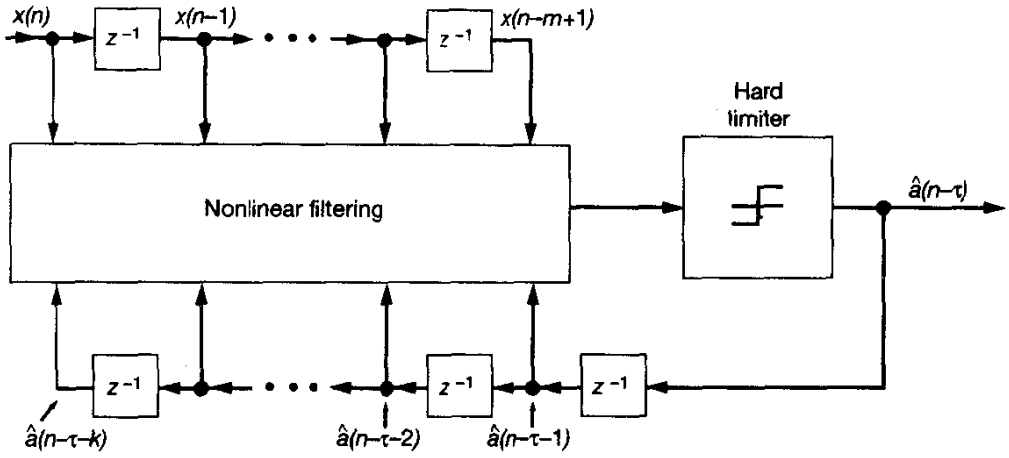



**Figure 20.3** Block diagram of decision-feedback equalizer using a nonlinear filter.

viewpoint, a linear equalizer is equivalent to a linear pattern classifier. However, in realistic situations where noise is present in the received signal and/or the channel is nonminimum phase, the optimum pattern classifier is in fact nonlinear (Gibson and Cowan, 1990), which may therefore benefit from the use of a neural network.

In order to see how the design of a nonlinear adaptive equalizer can be based on neural networks, consider first the block diagram of Fig. 20.3, which depicts a *nonlinear decision-feedback equalizer*. The equalizer consists of a nonlinear filter with feedforward inputs denoted by $u(n)$, $u(n-1)$, . . . , $u(n-M)$ and feedback inputs denoted by $a(n-\tau-1)$, $a(n-\tau-2)$, . . . , $a(n-\tau-k)$, where $u(n)$ is the channel output (received signal) at time $n$, and $a(n-\tau)$ is the equalizer output representing an estimate of the transmitted symbol $a(n)$ delayed by $\tau$ seconds. The equalizer output is produced by passing the output of a nonlinear filter inside the equalizer through a hard limiter, whereby decisions are made on a symbol-by-symbol basis. Using Bayesian considerations, Chen et al. (1992a, b, 1994), have shown that the optimum form of the nonlinear filter in the decision-feedback equalizer of Fig. 20.3 has an identical structure to that of the RBF network.

Chen et al. have also used computer simulation to investigate the performance of an RBF decision-feedback equalizer and compared it with that of (1) a standard decision-feedback equalizer using a transversal filter, and (2) a maximum-likelihood sequential estimator known as the Viterbi algorithm. The investigations were carried out for both stationary and nonstationary channels; the highly nonstationary channel considered in the study was chosen to be representative of a mobile radio environment. The results of the investigations reported by Chen et al. may be summarized as follows:

- The maximum-likelihood sequential estimator provides the best attainable performance for the case of a stationary channel; the corresponding performance of the RBF decision-feedback equalizer is worse by about 2 dB, but better than that of the standard decision-feedback equalizer by roughly an equal amount.

- In the case of a highly nonstationary channel, the RBF decision-feedback equalizer outperforms the maximum-likelihood sequential estimator; in the latter case, the degradation in performance is attributed to the accumulation of errors.

The results of the study of Chen et al. appear to show that the RBF decision-feedback equalizer is robust, and that it provides a viable solution for the equalization of highly nonstationary communication channels.

### Dynamic RBF Network for Time Series Prediction

For the second application, we consider a simplified implementation of the *dynamic Gaussian RBF network* used as a nonlinear predictor, in which learning takes place on a *continuous* basis, hence the description of the network as "dynamic." There are three key parameters of the network to be determined: The *centers*, the common *width*, and the linear *weighting coefficients*. Collectively, these parameters completely define the prediction of an input sample $u(n+1)$, which is computed at time step $n$ as

$$\begin{aligned}\hat{u}(n+1) &= F(\mathbf{u}(n)) \\ &= \sum_{k=1}^{K} w_k(n) \exp\left(-\frac{1}{2\sigma^2(n)} \|\mathbf{u}(n) - \mathbf{t}_{n-k}\|^2\right)\end{aligned} \tag{20.25}$$

where it is assumed that the input data are real valued. Specifically, the input vector $\mathbf{u}(n) = [u(n), u(n-1), \ldots, u(n-M+1)]^T$ is available for processing at time step $n$, where $M$ is the prediction order. The network parameters are described as follows:

1. The centers $\mathbf{t}_{n-k}$, $k = 1, 2, \ldots, K$, constitute the set of *process-state vectors*.
2. The width $\sigma(n)$ is typically computed as a function of the empirical covariance of the time series data, and is common to all centers; this forces the interpolation matrix $\mathbf{\Phi}(n)$ to be symmetric, thereby improving numerical stability.
3. The coefficient vector $\mathbf{w}(n) = [w_1(n), w_2(n), \ldots, w_K(n)]^T$ satisfies the *strict interpolation (SI) condition:*

$$[\mathbf{\Phi}(n) + \lambda(n)\,\mathbf{I}]\,\mathbf{w}(n) = \mathbf{d}(n) \tag{20.26}$$

where the interpolation matrix $\mathbf{\Phi}(n)$ is defined by

$$\mathbf{\Phi}(n) = \left\{\exp\left(-\frac{1}{2\sigma^2(n)} \|\mathbf{u}_{n-i} - \mathbf{u}_{n-k}\|^2\right)\right\}, \qquad (i,k) = 1, 2, \ldots, K \tag{20.27}$$

and $\lambda(n)$ is the *regularization parameter* at time step $n$, typically estimated from a window of the available time series data or fixed *a priori*. The desired response vector $\mathbf{d}(n)$ is defined by

$$\mathbf{d}(n) = [u(n), u(n-1), \ldots, u(n-K+1)]^T \tag{20.28}$$

From this formulation, we see that as new time series data become available, the dynamic Gaussian RBF predictor naturally evolves from time step $k$ to $k + 1$ via a shift in the predictor centres and the subsequent recomputation of $\mathbf{\Phi}(n)$, $\sigma(n)$, $\lambda(n)$, and $\mathbf{w}(n)$. The recomputation of $\mathbf{w}(n)$ is $O(K^3)$ in general but it can be shown that if $\lambda(n)$ is fixed and $\mathbf{\Phi}(n)$ is only partially updated by a low-rank matrix, then the complexity can be reduced to $O(K^2)$. For further discussion on this reduced complexity algorithm and its experimental effects, the reader is referred to Yee and Haykin (1995).

We shall illustrate the essential characteristics of the network by way of a nonlinear time series prediction experiment. This experiment involves the prediction of a reasonably noise-free male speech signal represented by a total of 10,000 samples at 8kHz and 8 bits per sample. For the purposes of the experiment, the speech signal is shifted to zero mean and scaled to unit total amplitude for both training and testing.

Intuitively, we would expect that in the case of a (strictly) stationary time series, there would exist an optimum set of dynamic Gaussian RBF predictor parameters that (at least, in principle) could be learned by some appropriate means as in the case of the LMS algorithm for linear processes. Where the interest lies, however, is in the prediction of a nonstationary, nonlinear process. Here the current state of the art revolves around the use of local *linear* approximations to the process state-space mapping, leading to algorithms such as the *extended Kalman Filter (EKF)* and its generalized counterparts. The *dynamic* component of the Gaussian RBF predictor extends this idea to a series of local *nonlinear* approximations to the process state-space mapping via continuously updated Gaussian RBF curve fits over the most currently available windows of time series data. Again, we may expect this dynamic updating to yield improved tracking for a significantly nonstationary process. Indeed, in Fig. 20.4, we compare the performance of a static RBF predictor to a dynamic one over the speech signal. By "static Gaussian RBF predictor", we mean a Gaussian RBF predictor with $K = 250$ centers trained once with a given initial window of time series data and then frozen thereafter. In contrast, the dynamic predictor with $K = 100$ centers has its parameters updated once per time step according to the scheme previously outlined. Both predictors use a state-space order of $M = 50$ and a fixed regularization parameter $\lambda = 0.01$. The plot in Fig. 20.4(a) clearly demonstrates how the static predictor tracks well over those segments of the speech signal that are similar to the initial segment (unvoiced speech) upon which it was trained, but is unable to track the middle segment of more quickly varying speech (voiced speech). On the other hand, the plot in Fig. 20.4(b) shows that the dynamic predictor, despite having fewer than half the number of centers of the static predictor, is able to adapt to and maintain tracking over all of the speech signal.

The use of regularization in the solution of interpolation and approximation problems is well-established (Morozov, 1993). Roughly speaking, regularization stabilizes the solution of *ill-posed* problems (of which nonparametric curve fitting is one), in the sense that it can make the solution less sensitive to noise and errors in the given data. As a simple example, the interpolation matrix $\mathbf{\Phi}(n)$ for the dynamic Gaussian RBF predictor should be, in principle, nonsingular for any distinct choice of centers drawn from the time series data; in practice, however, we observe that the likelihood of ill-conditioning in $\mathbf{\Phi}(n)$

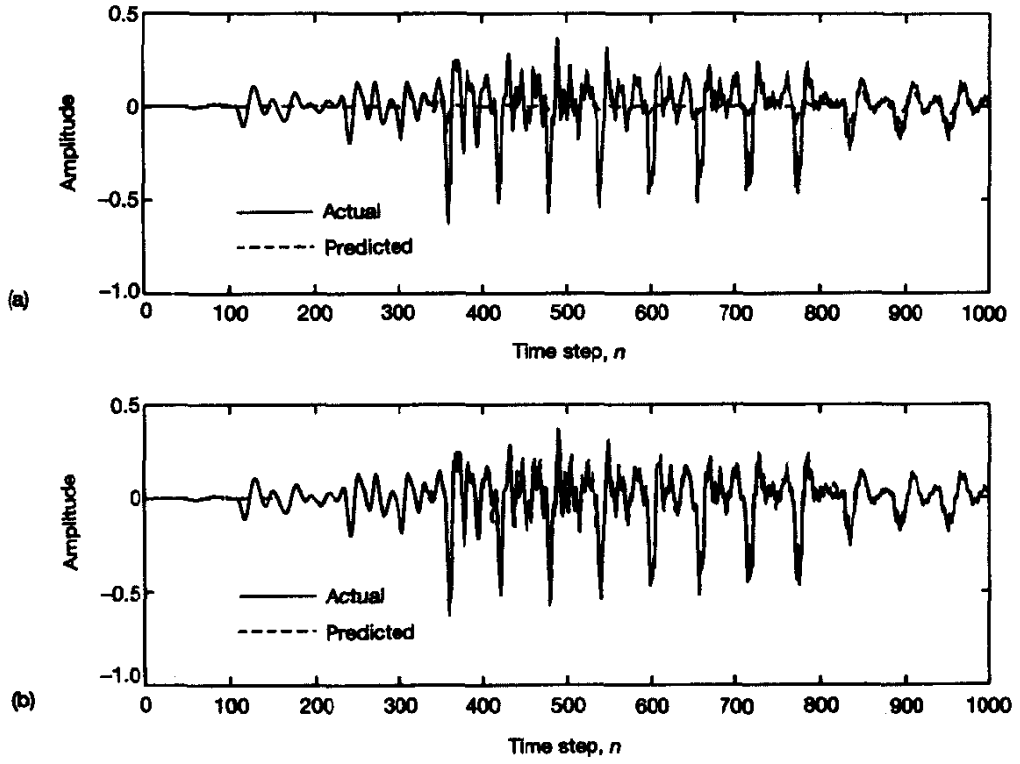



**Figure 20.4** Tracking ability of (a) static nonlinear predictor versus (b) dynamic nonlinear predictor.

increases as the size $K$ increases. The stabilizing effect of regularization on the dynamic predictor in this context can be seen in Fig. 20.5, where a 100-point segment of the speech signal shown along with two regularized dynamic Gaussian RBF predictors with $K = 100$ centers of state-space order $M = 2$. The essentially "nonregularized" predictor, which uses only a minimal regularization parameter $\lambda = 0.01$ to avoid singularity in the interpolation matrix, exhibits numerical instability near time step $n = 50$; on the other hand, the regularized predictor with $\lambda = 0.1$ has no such difficulty. Note also that even where the nonregularized predictor achieves some degree of tracking, the regularized one tracks the speech signal more closely.

As a final note, we should mention that when compared with the pseudo-linear adaptive predictor specified in CCITT Recommendation G.726 operating at 32kbits/s, a dynamic Gaussian RBF predictor with $K = 100$ centers shows a nontrivial improvement of approximately 4 dB in prediction SNR, as measured both segmentally and completely over the entire speech signal (Yee and Haykin, 1995). This result suggests that nonlinear

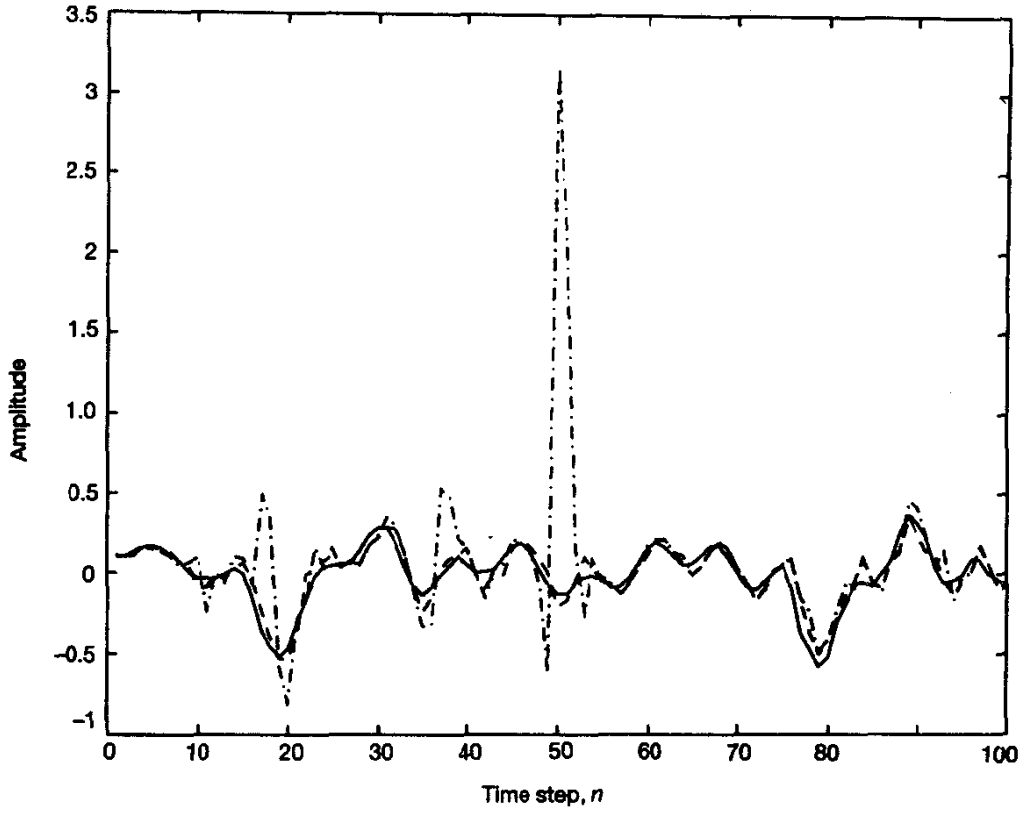



**Figure 20.5** Regularized versus nonregularized predictor, $K = 100$, $M = 2$: ('—' is actual, '- - -' is nonlinear prediction for regularization parameter $\lambda = 0.1$, and '-.-.-' is nonlinear prediction for $\lambda = 0.01$)

methods, as opposed to the current linear ones, of predicting speech can yield yet better performance than previously thought possible.

## 20.8 SUMMARY AND DISCUSSION

The structure of an RBF network is unusual in that the constitution of its hidden units is entirely different from that of its output units. The theory of RBF networks is linked intimately with that of radial-basis functions, which is nowadays one of the main fields of study in numerical analysis (Singh, 1992). Another interesting point to note is that with linear weights of the output layer providing a set of adjustable parameters, much can be gained by studying the well-developed theory of linear adaptive filters presented in Part III of the book.

RBF networks have a rich theory of their own, as summarized here:

- With appropriate extensions, RBF networks are an important case in *regularization theory* (Poggio and Girosi, 1990). In the context of approximation problems, the basic idea of regularization is to *stabilize* the solution by means of some auxiliary nonnegative functional that embeds prior knowledge (e.g., smoothness constraints on the input–output mapping), and thereby makes an ill-posed problem into a well-posed one.
- For function approximation, an RBF network has been shown (under certain conditions) to provide the minimum variance approximation to a function when the input data are corrupted by additive noise (Webb, 1994). In this case, the nonlinear activation functions are determined by the probability density function of the additive noise in the input signal. The necessary conditions are that the standard deviation of the noise be large compared to the sample spacing of the data points. This form of an RBF network solution for a finite number of training examples is not imposed *a priori*; rather it follows naturally as a direct consequence of a least-squares approach to the problem of function approximation and generalization.
- The input–output mapping of a Gaussian RBF network, described by Eq. (20.8), is closely related to *mixture models*, that is, mixtures of Gaussian distributions. Mixtures of probability distributions, in particular, Gaussian distributions, have been used extensively as models in a wide variety of applications where the data of interest arise from two or more populations mixed together in some varying properties (Tetterington et al., 1985; McLachlan and Basford, 1988).
- RBF networks are closely related to *kernel-based methods*, on which there is a large amount of literature (Duda and Hart, 1973; Davijver and Kittler, 1982; Fukuraga, 1990). A *kernel* is a function $K(\mathbf{x}, \mathbf{x}_j)$ that attains its maximum value at the point $\mathbf{x} = \mathbf{x}_j$ and decreases monotonically as the distance between the vectors $\mathbf{x}$ and $\mathbf{x}_j$ increases, subject to the condition that

$$\int K(\mathbf{x}, \mathbf{x}_j)\, d\mathbf{x} = 1 \qquad \text{for all } \mathbf{x}_j$$

The kernel function $K(\mathbf{x}, \mathbf{x}_j)$ provides information about the unknown conditional probability density function $f(\mathbf{x}|\omega_i)$ of a $p$-dimensional vector (query point) $\mathbf{x}$, given that a particular class $\omega_i$ is true; the information is built up by using a set of training examples $\mathbf{x}_j$, $j = 1, 2, \ldots, N$, that belong to class $\omega_i$. In the well-known *Parzen density estimator* (Parzen, 1962), a Gaussian kernel of fixed width $\sigma$ is commonly used, and an estimate of the unknown probability density function is given by

$$\hat{f}(\mathbf{x}|\omega_i) = \frac{1}{N} \sum_{j=1}^{N} K(\mathbf{x}, \mathbf{x}_j)$$

where (assuming real-valued data)

$$K(\mathbf{x}, \mathbf{x}_j) = \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left( -\frac{1}{2\sigma^2} (\mathbf{x} - \mathbf{x}_j)^T (\mathbf{x} - \mathbf{x}_j) \right)$$

The estimate $\hat{f}(\mathbf{x}|\omega_i)$ is simply the sum of a number of multivariate Gaussian distributions that are centered on the particular set of training examples. The sum so defined can approximate any smooth density function. An important advantage of the Parzen density estimator is the fact that it is probably *consistent*, which means that the estimate $\hat{f}(\mathbf{x}|\omega_i)$ approaches the optimal Bayes estimate for class $\omega_i$ as the size $N$ of the training set for that class approaches infinity. However, it has the disadvantage of requiring a very large amount of storage. The *probabilistic neural network* described in Specht (1990) is an implementation of the Parzen window estimator.

Perhaps the most interesting aspect of RBF networks is that the basic expansion described in Eq. (20.1) with the output $y$ viewed as a probability density function over a space of real numbers, the $\phi(\mathbf{u}; \mathbf{t}_k)$ viewed as a set of expansion functions, and the $w_k$ viewed as the corresponding amplitudes, may be given a *neurobiological interpretation* (Anderson and Van Essen, 1994). According to such an interpretation, the amplitudes are represented physically as neuronal firing rates, while the functions forming the mathematical basis of the representation provide a convenient set of rules by which the amplitudes are manipulated.

In conclusion, multilayer perceptrons (trained with the back-propagation algorithm) and RBF networks constitute the backbone of supervised neural networks in their own individual ways. They are both examples of multilayer feedforward networks that are universal approximators. The basic difference between them is that Gaussian RBF networks provide local approximation, whereas multilayer perceptrons provide global approximation. This, in turn, means that for the approximation of a nonlinear input–output mapping, the multilayer perceptron may be more *parimoneous* (i.e., require a smaller number of scalar coefficients) than the RBF network for a prescribed degree of accuracy. On the other hand, when continuous learning is required as in the tracking of a time-varying environment, the use of nested nonlinearities in a multilayer perceptron makes it difficult to evolve the network in a dynamic fashion. That is, if we want to include a new example in the training set or enlarge the multilayer perceptron by adding new synaptic weights, then the whole network must be retrained all over again. In contrast, the structure of an RBF network permits it to operate dynamically, such that the centers of the radial-basis functions in the hidden layer and the linear weights of the output layer may be updated without having to recompute them from scratch (Yee and Haykin, 1995).

## PROBLEMS

1. It may be argued, by virtue of the central limit theorem, that for a Gaussian RBF network the output produced by the network in response to a random input vector may be approximated by a Gaussian distribution, and that the approximation gets better as the number of centers in the network is increased. Rationalize the validity of this statement.
2. In describing the recursive hybrid learning procedure for the design of a complex RBF network presented in Section 20.5, we mentioned the RLS algorithm and the LMS algorithm as possible candidates for computing the weight vector of the output layer. Formulate the algorithm for performing this computation, using:

(a) The RLS algorithm
(b) The LMS algorithm

3. Table 20.1 presents a summary of the stochastic gradient algorithm for computing the centers and widths of the Gaussian hidden units and the linear weights in the output layer of a complex RBF network. Present detailed derivations of the results summarized in Table 20.1.
4. A requirement exists for the design of an RBF network to perform interference cancelation, in which the reference signal and interference are nonlinearly correlated. Describe how this requirement can be achieved.
5. Investigate the possible use of a Gaussian RBF network to perform the blind equalization of a nonminimum-phase communication channel.
6. A *normalized* Gaussian basis function is defined in Moody and Danken (1989) as follows:

$$\varphi_i = \frac{\exp\left(-\frac{1}{\sigma_i^2}\|\mathbf{u} - \mathbf{t}_i\|^2\right)}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\sigma_k^2}\|\mathbf{u} - \mathbf{t}_k\|^2\right)}, \qquad i = 1, 2, \ldots, K$$

(a) On this basis, $\varphi_i$ may be viewed as the probability that the hidden neuron with center $\mathbf{t}_i$ is the "winning" neuron (i.e., the neuron closest to the input vector $\mathbf{u}$ in Euclidean norm). Explain the rationale for this statement.
(b) In what way is the $\varphi_i$ for $i = 1$ or $i = K$ different from other values of $\varphi_i$?

# APPENDIX A

# Complex Variables

This Appendix presents a brief review of the functional theory of complex variables. In the context of the material considered in this book, a complex variable of interest is the variable $z$ associated with the $z$-transform. We begin the review by defining analytic functions of a complex variable, and then derive the important theorems that make up the important subject of complex variables[1]. .

## A.1 CAUCHY–REIMANN EQUATIONS

Consider a complex variable $z$ defined by

$$z = x + jy$$

where $x = \mathrm{Re}[z]$, and $y = \mathrm{Im}[z]$. We speak of the plane in which the complex variable $z$ is represented as the *z-plane*. Let $f(z)$ denote a *function of the complex variable* $z$, written as

$$w = f(z) = u + jv$$

The function $w = f(z)$ is *single-valued* if there is only one value of $w$ for each $z$ in a given region of the $z$-plane. If, on the other hand, more than one value of $w$ corresponds to $z$, the function $w = f(z)$ is said to be *multiple-valued.*

[1]For a detailed treatment of the functional theory of complex variables, see Guillemin (1949), Levinson and Redheffer (1970), and Wylie and Barrett (1982).

We say that a point $z = x + jy$ in the $z$-plane approaches a fixed point $z_0 = x_0 + jy_0$ if $x \to x_0$ and $y \to y_0$. Let $f(z)$ denote a single-valued function of $z$ that is defined in some neighborhood of the point $z = z_0$. The *neighborhood* of $z_0$ refers to the set of all points in a sufficiently small circular region centered at $z_0$. Let

$$\lim_{z \to z_0} f(z) = w_0$$

In particular, if $f(z_0) = w_0$, then the function $f(z)$ is said to be *continuous* at $z = z_0$.

Let $f(z)$ be written in terms of its real and imaginary parts as

$$f(z) = u(x, y) + jv(x, y)$$

Then, if $f(z)$ is continuous at $z_0 = x_0 + jy_0$, its real and imaginary parts $u(x, y)$ and $v(x, y)$ are continuous functions at $(x_0, y_0)$, and vice versa.

Let $w = f(z)$ be continuous at each point of some region of interest in the $z$-plane. The complex quantities $w$ and $z$ may then be represented on separate planes of their own, referred to as the $w$- and $z$-planes, respectively. In particular, a point $(x, y)$ in the $z$-plane corresponds to a point $(u, v)$ in the $w$-plane by virtue of the relationship $w = f(z)$.

Consider an incremental change $\Delta z$ such that the point $z_0 + \Delta z$ may lie anywhere in the neighborhood of $z_0$, and throughout which the function $f(z)$ is defined. We may then define the *derivative* of $f(z)$ with respect to $z$ at $z = z_0$ as

$$f'(z_0) = \lim_{\Delta z \to 0} \frac{f(z_0 + \Delta z) - f(z_0)}{\Delta z} \tag{A.1}$$

Clearly, for the derivative $f'(z_0)$ to have a unique value, the limit in Eq. (A.1) must be independent of the way in which $\Delta z$ approaches zero.

For a function $f(z)$ to have a unique derivative at some point $z = x + jy$, it is necessary that its real and imaginary parts satisfy certain conditions, as shown next. Let

$$w = f(z) = u(x, y) + jv(x, y)$$

With $\Delta w = \Delta u + j\Delta v$ and $\Delta z = \Delta x + j\Delta y$, we may write

$$\begin{aligned} f'(z) &= \lim_{\Delta z \to 0} \frac{\Delta w}{\Delta z} \\ &= \lim_{\substack{\Delta x \to 0 \\ \Delta y \to 0}} \frac{\Delta u + j\Delta v}{\Delta x + j\Delta y} \end{aligned} \tag{A.2}$$

Suppose that we let $\Delta z \to 0$ by first letting $\Delta y \to 0$ and then $\Delta x \to 0$, in which case $\Delta z$ is purely real. We then deduce from Eq. (A.2) that

$$\begin{aligned} f'(z) &= \lim_{\Delta x \to 0} \frac{\Delta u}{\Delta x} + j \frac{\Delta v}{\Delta x} \\ &= \frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} \end{aligned} \tag{A.3}$$

Suppose next that we let $\Delta z \to 0$ by first letting $\Delta x \to 0$ and then $\Delta y \to 0$, in which case $\Delta z$ is purely imaginary. This time we deduce from Eq. (A.2) that

$$\begin{aligned} f'(z) &= \lim_{\Delta y \to 0} \frac{\Delta v}{\Delta y} - j \frac{\Delta u}{\Delta y} \\ &= \frac{\partial v}{\partial y} - j \frac{\partial u}{\partial y} \end{aligned} \tag{A.4}$$

If the derivative $f'(z)$ is to exist, it is necessary that the two expressions in Eqs. (A.3) and (A.4) be one and the same. Hence, we require

$$\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} - \frac{\partial u}{\partial y}$$

Accordingly, equating real and imaginary parts, we get the following pair of relations, respectively:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \tag{A.5}$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} \tag{A.6}$$

Equations (A.5) and (A.6), known as the *Cauchy–Riemann equations*, were derived from a consideration of merely two of the infinitely many ways in which $\Delta z$ can approach zero. For $\Delta w/\Delta z$ evaluated along these other paths to also approach $f'(z)$, we need only make the additional requirement that the partial derivatives in Eqs. (A.5) and (A.6) are continuous at the point $(x, y)$. In other words, provided that the real part $u(x, y)$ and the imaginary part $v(x, y)$ together with their first partial derivatives are continuous at the point $(x, y)$, the Cauchy–Riemann equations are not only necessary but also sufficient for the existence of a derivative of the complex function $w = u(x, y) + jv(x, y)$ at the point $(x, y)$.

A function $f(z)$ is said to be *analytic*, or *homomorphic*, at some point $z = z_0$ in the $z$-plane if it has a derivative at $z = z_0$ and at every point in the neighborhood of $z_0$; the point $z_0$ is called a *regular point* of the function $f(z)$. If the function $f(z)$ is *not* analytic at a point $z_0$, but if every neighborhood of $z_0$ contains points at which $f(z)$ is analytic, the point $z_0$ is referred to as a *singular point* of $f(z)$.

## A.2 CAUCHY'S INTEGRAL FORMULA

Let $f(z)$ be any continuous function of the complex variable $z$, analytic or otherwise. Let $\mathscr{C}$ be a sectionally smooth path joining the points $A = z_0$ and $B = z_n$ in the $z$ plane. Suppose that the path $\mathscr{C}$ is divided into $n$ segments $\Delta s_k$ by the points $z_k$, $k = 1, 2, \ldots, n-1$, as illustrated in Fig. A.1. This figure also shows an arbitrary point $\zeta_k$ on segment $\Delta s_k$, depicted as an elementary arc of length $\Delta z_k$. Consider then the summation $\sum_{k=1}^{n} f(\zeta_k)\, \Delta z_k$.

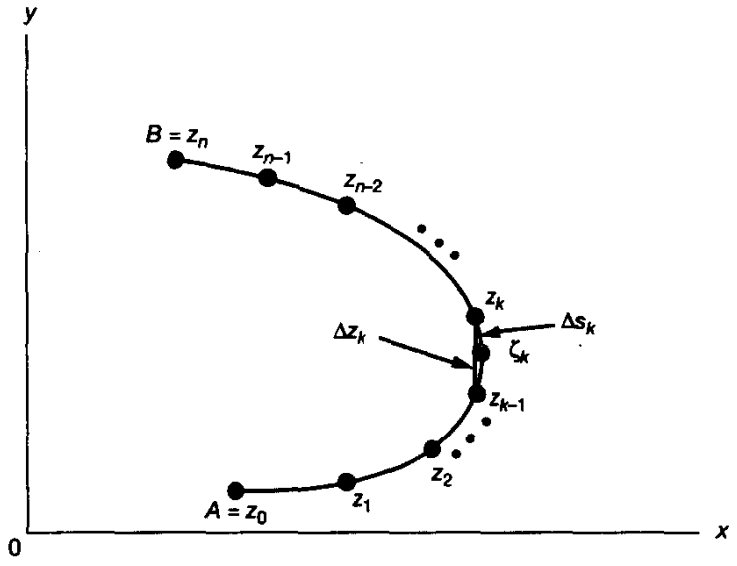



**Figure A.1** Sectionally smooth path.

The *line integral* of $f(z)$ along the path $\mathscr{C}$ is defined by the limiting value of this summation as the number $n$ of segments is allowed to increase indefinitely in such a way that $\Delta z_k$ approaches zero. That is

$$\oint_{\mathscr{C}} f(z)\,dz = \lim_{n\to\infty} \sum_{k=1}^{n} f(\zeta_k)\,\Delta z_k \tag{A.7}$$

In the special case when the points $A$ and $B$ coincide and $\mathscr{C}$ is a closed curve, the integral in Eq. (A.7) is referred to as a *contour integral* that is written as $\oint_{\mathscr{C}} f(z)dz$. Note that, according to the notation described herein, the contour $\mathscr{C}$ is transversed in a *counterclockwise direction*.

Let $f(z)$ be an analytic function in a given region $R$, and let the derivative $f'(z)$ be continuous there. The line integral $\oint_{\mathscr{C}} f(z)dz$ is then independent of the path $\mathscr{C}$ that joins any pair of points in the region $R$. If the path $\mathscr{C}$ is a closed curve, the value of this integral is zero. We thus have *Cauchy's integral theorem*, stated as follows:

If a function $f(z)$ is analytic throughout a region $R$, then the contour integral of $f(z)$ along any closed path $\mathscr{C}$ lying inside the region $R$ is zero, as shown by

$$\oint_{\mathscr{C}} f(z)\,dz = 0 \tag{A.8}$$

This theorem is of cardinal importance in the study of analytic functions.

An important consequence of Cauchy's theorem is known as *Cauchy's integral formula*. Let $f(z)$ be analytic within and on the boundary $\mathscr{C}$ of a simple connected region. Let $z_0$ be any point in the interior of $\mathscr{C}$. Then Cauchy's integral formula states that

$$f(z_0) = \frac{1}{2\pi j} \oint_{\mathscr{C}} \frac{f(z)}{z - z_0}\, dz \tag{A.9}$$

where the contour integration around $\mathscr{C}$ is taken in the counterclockwise direction.

Cauchy's integral formula expresses the value of the analytic function $f(z)$ at an interior point $z_0$ of $\mathscr{C}$ in terms of its values on the boundary of $\mathscr{C}$. Using this formula, it is a straightforward matter to express the derivative of $f(z)$ of all orders as follows:

$$f^{(n)}(z_0) = \frac{n!}{2\pi j} \oint_{\mathscr{C}} \frac{f(z)}{(z - z_0)^{n+1}}\, dz \tag{A.10}$$

where $f^{(n)}(z_0)$ is the $n$th derivative of $f(z)$ evaluated at $z = z_0$. Equation (A.10) is obtained by repeated differentiation of Eq. (A.9) with respect to $z_0$.

### Cauchy's Inequality

Let the contour $\mathscr{C}$ consist of a circle of radius $r$ and center $z_0$. Then, using Eq. (A.10) to evaluate the magnitude of $f^{(n)}(z_0)$, we may write

$$\begin{aligned} |f^{(n)}(z_0)| &= \frac{n!}{2\pi} \left| \oint_{\mathscr{C}} \frac{f(z)}{(z - z_0)^{n+1}}\, dz \right| \\ &\leq \frac{n!}{2\pi} \oint_{\mathscr{C}} \frac{|f(z)|}{|z - z_0|^{n+1}}\, |dz| \\ &\leq \frac{n!}{2\pi} \frac{M}{r^{n+1}} \oint_{\mathscr{C}} |dz| \\ &= \frac{n!}{2\pi} \frac{M}{r^{n+1}}\, 2\pi r \\ &= n! \frac{M}{r^n} \end{aligned} \tag{A.11}$$

where $M$ is the maximum value of $f(z)$ on $\mathscr{C}$. The inequality of (A.11) is known as *Cauchy's inequality*.

## A.3 LAURENT'S SERIES

Let the function $f(z)$ be analytic in the annular region of Fig. A.2, including the boundary of the region. The annular region consists of two concentric circles $\mathscr{C}_1$ and $\mathscr{C}_2$, whose common center is $z_0$. Let the point $z = z_0 + h$ be located inside the annular region as depicted in Fig. A.2. According to *Lauren's series*, we have

$$f(z_0 + h) = \sum_{k=-\infty}^{\infty} a_k h^k \tag{A.12}$$

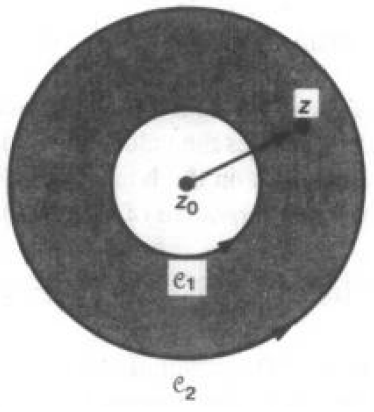



**Figure A.2** Annular region.

where the coefficients $a_k$ for varying $k$ are given by

$$a_k = \begin{cases} \dfrac{1}{2\pi j} \oint_{c_2} \dfrac{f(z)\,dz}{(z - z_0)^{k+1}}, & k = 0, 1, 2, \ldots \\ \dfrac{1}{2\pi j} \oint_{c_1} \dfrac{f(z)\,dz}{(z - z_0)^{k+1}}, & k = -1, -2, \ldots \end{cases} \tag{A.13}$$

Note that we may also express the Laurent expansion of $f(z)$ around the point $z$ as

$$f(z) = \sum_{k=-\infty}^{\infty} a_k (z - z_0)^k \tag{A.14}$$

When all the coefficients of negative index have the value zero, then Eq. (A.14) reduces to *Taylor's series*:

$$f(z) = \sum_{k=0}^{\infty} a_k (z - z_0)^k \tag{A.15}$$

In light of Eq. (A.10) and the first line of Eq. (A.13), we may define the coefficient $a_k$ as

$$a_k = \frac{f^{(k)}(z_0)}{k!}, \qquad k = 0, 1, 2, \ldots \tag{A.16}$$

Taylor's series provides the basis of Liouville's theorem, considered next.

### Liouville's Theorem

Let a function $f(z)$ of the complex variable $z$ be bounded and analytic for all values of $z$. Then, according to *Liouville's theorem*, $f(z)$ is simply a constant.

To prove this theorem, we first note that since $f(z)$ is analytic everywhere inside the $z$-plane, we may use Taylor's series to expand $f(z)$ about the origin:

$$f(z) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} z^k \tag{A.17}$$

The power series of Eq. (A.17) is convergent, and therefore provides a valid representation of $f(z)$. Let contour $\mathscr{C}$ consist of a circle of radius $r$ and origin as center. Then, invoking Cauchy's inequality of (A.11), we may write

$$|f^{(k)}(0)| \leq \frac{k!\, M_c}{r^k} \tag{A.18}$$

where $M_c$ is the maximum value of $f(z)$ on $\mathscr{C}$. Correspondingly, the value of the $k$th coefficient in the power series expansion of Eq. (A.17) is bounded as

$$|a_k| = \frac{|f^{(k)}(0)|}{k!} \leq \frac{M_c}{r^k} \leq \frac{M}{r^k} \tag{A.19}$$

where $M$ is the bound on $|f(z)|$ for all values of $z$. Since, by hypothesis, $M$ does exist, it follows from (A.19) that for an arbitrarily large $r$:

$$a_k = \begin{cases} f(0), & k=0 \\ 0, & k = 1, 2, \ldots \end{cases} \tag{A.20}$$

Accordingly, Eq. (A.17) reduces to

$$f(z) = f(0) = \text{constant}$$

which proves Liouville's theorem.

A function $f(z)$ that is analytic for all values of $z$ is said to be an *entire function*. Thus, Liouville's theorem may be restated as follows: *An entire function that is bounded for all values of* $z$ *is a constant* (Wylie and Barrett, 1982).

## A.4 SINGULARITIES AND RESIDUES

Let $z = z_0$ be a singular point of an analytic function $f(z)$. If the neighborhood of $z = z_0$ contains *no* other singular points of $f(z)$, the singularity at $z = z_0$ is said to be *isolated*. In the neighborhood of such a singularity, the function $f(z)$ may be represented by the Laurent series

$$\begin{aligned} f(z) &= \sum_{k=-\infty}^{\infty} a_k (z - z_0)^k \\ &= \sum_{k=0}^{\infty} a_k (z - z_0)^k + \sum_{k=-\infty}^{-1} a_k (z - z_0)^k \\ &= \sum_{k=0}^{\infty} a_k (z - z_0)^k + \sum_{k=1}^{\infty} \frac{a_{-k}}{(z - z_0)^k} \end{aligned} \tag{A.21}$$

The particular coefficient $a_{-1}$ in the Laurent expansion of $f(z)$ in the neighborhood of the isolated singularity at the point $z = z_0$ is called the *residue* of $f(z)$ at $z = a$. The residue plays an important role in the evaluation of integrals of analytic functions. In particular,

putting $k = -1$ in Eq. (A.13) we get the following connection between the residue $a_{-1}$ and the integral of the function $f(z)$:

$$a_{-1} = \frac{1}{2\pi j} \oint_{\mathscr{C}} f(z)\, dz \tag{A.22}$$

There are two nontrivial cases to be considered:

1. The Laurent expansion of $f(z)$ contains *infinitely* many terms with negative powers of $z - z_0$, as in Eq. (A.21). The point $z - z_0$ is then called an *essential singular point* of $f(z)$.
2. The Laurent expansion of $f(z)$ contains at most *a finite* number of terms, $m$, with negative powers of $z - z_0$, as shown by

$$f(z) = \sum_{k=0}^{\infty} a_k (z - z_0)^k + \frac{a_{-1}}{z - z_0} + \frac{a_{-2}}{(z - z_0)^2} + \cdots + \frac{a_{-m}}{(z - z_0)^m} \tag{A.23}$$

According to this latter representation, $f(z)$ is said to have a *pole of order m* at $z = z_0$. The *finite sum* of all the terms containing *negative powers* on the right-hand side of Eq. (A.22) is called the *principal part* of $f(z)$ at $z = z_0$.

Note that when the singularity at $z = z_0$ is a pole of order $m$, the residue of the pole may be determined by using the formula

$$a_{-1} = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^m f(z)]_{z=z_0} \tag{A.24}$$

In effect, by using this formula we avoid the need for the deduction of the Laurent series. For the special case when the order $m = 1$, the pole is said to be *simple*. Correspondingly, the formula of Eq. (A.24) for the residue $a_{-1}$ of a simple pole reduces to

$$a_{-1} = \lim_{z \to z_0} (z - z_0) f(z) \tag{A.25}$$

## A.5 CAUCHY'S RESIDUE THEOREM

Consider a closed contour $\mathscr{C}$ in the $z$-plane containing within it a number of isolated singularities of some function $f(z)$. Let $z_1, z_2, \ldots, z_n$ define the locations of these isolated singularities. Around each singular point of the function $f(z)$, we draw a circle small enough to ensure that it does not enclose the other singular points of $f(z)$, as depicted in Fig. A.3. The original contour $\mathscr{C}$ together with these small circles constitute the boundary of a *multiply connected region* in which $f(z)$ is analytic everywhere and to which Cauchy's integral theorem may therefore be applied. Specifically, for the situation described in Fig. A.3 we may write

$$\frac{1}{2\pi j} \oint_{\mathscr{C}} f(z)\, dz + \frac{1}{2\pi j} \oint_{\mathscr{C}_1} f(z)\, dz + \cdots + \frac{1}{2\pi j} \oint_{\mathscr{C}_n} f(z)\, dz = 0 \tag{A.26}$$

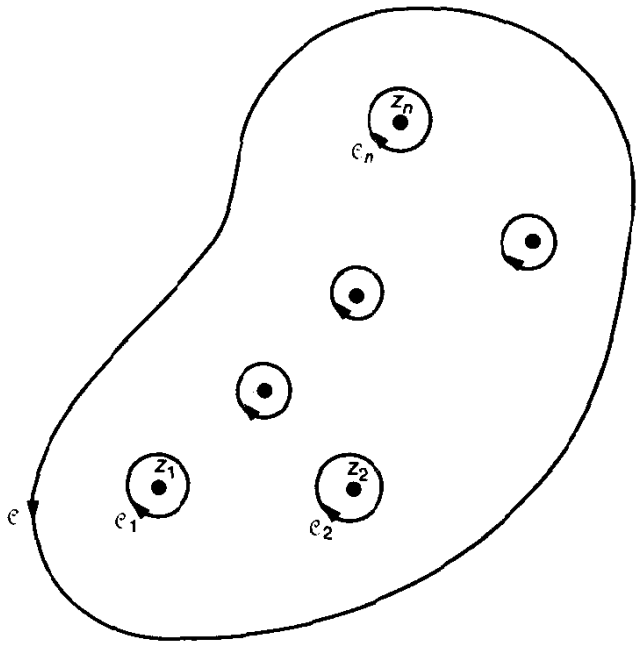



**Figure A.3** Multiply connected region.

Note that in Fig. A.3 the contour $\mathscr{C}$ is traversed in the *positive* sense (i.e., counterclockwise direction), whereas the small circles are traversed in the *negative* sense (i.e., clockwise direction).

Suppose now we *reverse* the direction along which the integral around each small circle in Fig. A.3 is taken. This operation has the equivalent effect of applying a minus sign to each of the integrals in Eq. (A.26) that involve the small circles $\mathscr{C}_1, \ldots, \mathscr{C}_n$. Accordingly, for the case when *all* the integrals around the original contour $\mathscr{C}$ and the small circles $\mathscr{C}_1, \ldots, \mathscr{C}_n$ are taken in the counterclockwise direction, we may rewrite Eq. (A.26) as

$$\frac{1}{2\pi j}\oint_{\mathscr{C}} f(z)\,dz = \frac{1}{2\pi j}\oint_{\mathscr{C}_1} f(z)\,dz + \cdots + \frac{1}{2\pi j}\oint_{\mathscr{C}_n} f(z)\,dz \tag{A.27}$$

By definition, the integrals on the right-hand side of Eq. (A.27) are the residues of the function $f(z)$ evaluated at the various isolated singularities of $f(z)$ within the contour $\mathscr{C}$. We may thus express the integral of $f(z)$ around the contour $\mathscr{C}$ simply as

$$\oint_{\mathscr{C}} f(z)\,dz = 2\pi j \sum_{k=1}^{n} \text{Res}(f(z), z_k) \tag{A.28}$$

where $\text{Res}(f(z), z_k)$ stands for the residue of the function $f(z)$ evaluated at the isolated singular point $z = z_k$. Equation (A.28) is called *Cauchy's residue theorem*. This theorem is extremely important in the theory of functions in general and in evaluating definite integrals in particular.

## A.6 PRINCIPLE OF THE ARGUMENT

Consider a complex function $f(z)$, characterized as follows:

1. The function $f(z)$ is analytic in the interior of a closed contour $\mathscr{C}$ in the $z$-plane, except at a finite number of poles.
2. The function $f(z)$ has neither poles nor zeros on the contour $\mathscr{C}$. By a "zero" we mean a point in the $z$-plane at which $f(z) = 0$. In contrast, at a "pole" as defined previously, we have $f(z) = \infty$. Let $N$ be the *number of zeros* and $P$ be the *number of poles* of the function $f(z)$ in the interior of contour $\mathscr{C}$, where each zero or pole is counted according to its *multiplicity*.

We may then state the following theorem (Levinson and Redheffer, 1970; Wylie and Barrett, 1982):

$$\frac{1}{2\pi j}\oint_{\mathscr{C}} \frac{f'(z)}{f(z)}\,dz = N - P \tag{A.29}$$

where $f'(z)$ is the derivative of $f(z)$. We note that

$$\frac{d}{dz}\ln f(z) = \frac{f'(z)}{f(z)}\,dz$$

where ln denotes the natural logarithm. Hence,

$$\begin{aligned}\oint_{\mathscr{C}} \frac{f'(z)}{f(z)}\,dz &= \ln f(z)|_{\mathscr{C}} \\ &= \ln |f(z)|_{\mathscr{C}} + j \arg f(z)|_{\mathscr{C}}\end{aligned} \tag{A.30}$$

where $|f(z)|$ denotes the magnitude of $f(z)$, and $\arg f(z)$ denotes its argument. The first term on the right-hand side of Eq. (A.30) is zero, since the logarithmic function $\ln f(z)$ is single-valued and the contour $\mathscr{C}$ is closed. Hence,

$$\oint_{\mathscr{C}} \frac{f'(z)}{f(z)}\,dz = j \arg f(z)|_{\mathscr{C}} \tag{A.31}$$

Thus, substituting Eq. (A.31) in (A.29), we get

$$N - P = \frac{1}{2\pi}\arg f(z)|_{\mathscr{C}} \tag{A.32}$$

This result, which is a reformulation of the theorem described in Eq. (A.29), is called the *principle of the argument.*

For a geometrical interpretation of this principle, let $\mathscr{C}$ be a closed contour in the $z$-plane as in Fig. A.4(a). As $z$ traverses the contour $\mathscr{C}$ in a counterclockwise direction, we find that $w = f(z)$ traces out a contour $\mathscr{C}'$ of its own in the $w$-plane; for the purpose of illustration, $\mathscr{C}'$ is shown in Fig. A.4(b). Suppose now a line is drawn in the $w$-plane from the

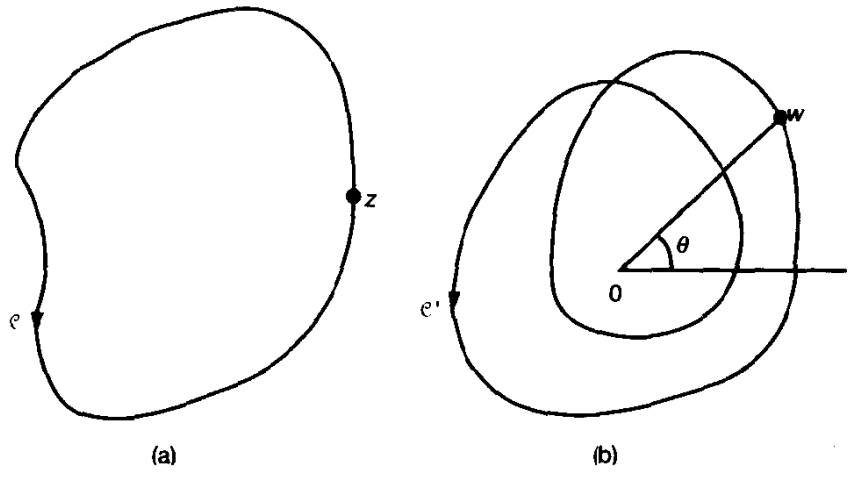



**Figure A.4** (a) Contour $\mathscr{C}$ in the $z$-plane; (b) Contour $\mathscr{C}'$ in the $w$-plane, where $w = f(z)$.

origin to the point $w = f(z)$, as depicted in Fig. A.4(b). Then the angle $\theta$ which this line makes with a fixed direction (shown as the horizontal direction in Fig. A.4(b)) is $\arg f(z)$. The principle of the argument thus provides a description of the number of times the point $w = f(z)$ winds around the origin of the $w$-plane (i.e., the point $w = 0$) as the complex variable $z$ traverses the contour $\mathscr{C}$ in a counterclockwise direction.

### Rouché's Theorem

Let the function $f(z)$ be analytic on a closed contour $\mathscr{C}$ and in the interior of $\mathscr{C}$. Let $g(z)$ be a second function which, in addition to satisfying the same condition for analyticity as $f(z)$, also fulfills the following condition on the contour $\mathscr{C}$:

$$|f(z)| > |g(z)|$$

In other words, on the contour $\mathscr{C}$ we have

$$\left|\frac{g(z)}{f(z)}\right| < 1 \tag{A.33}$$

Define the function

$$F(z) = 1 + \frac{g(z)}{f(z)} \tag{A.34}$$

which has no poles or zeros on $\mathscr{C}$. By the principle of the argument applied to $F(z)$, we have

$$N - P = \frac{1}{2\pi} \arg F(z)|_{\mathscr{C}} \tag{A.35}$$

However, the implication of the condition (A.33) is that when $z$ is on the contour $\mathscr{C}$, then

$$|F(z) - 1| < 1 \tag{A.36}$$

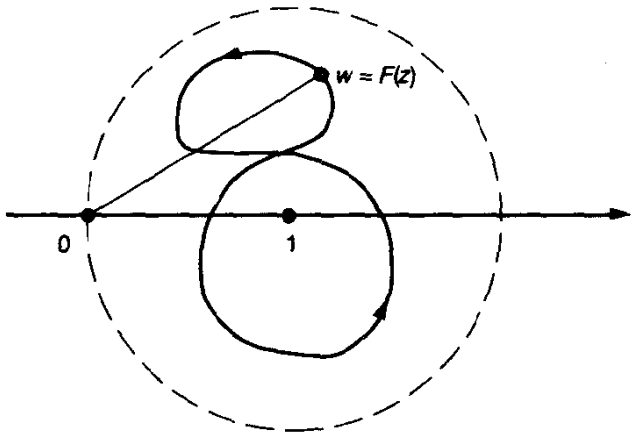



**Figure A.5** Point $w = F(z)$ on a closed contour inside the unit circle.

In other words, the point $w = F(z)$ lies inside a circle with center at $w = 1$ and unit radius, as illustrated in Fig. A.5. It follows therefore that

$$|\arg F(z)| < \frac{\pi}{2} \qquad \text{for } z \text{ on } \mathscr{C} \tag{A.37}$$

Equivalently, we may write

$$\arg F(z)|_{\mathscr{C}} = 0 \tag{A.38}$$

Hence, from Eq. (A.38) we deduce that $N = P$, where both $N$ and $P$ refer to $f(z)$. From the definition of the function $F(z)$ given in Eq. (A.34) we note that the poles of $F(z)$ are the zeros of $f(z)$, and the zeros of $F(z)$ are the zeros of the sum $f(z) + g(z)$. Accordingly, the fact that $N = P$ means that $f(z) + g(z)$ and $f(z)$ have the same numbers of zeros. The result that we have just established is known as *Rouché's theorem*, which may be formally stated as follows:

Let $f(z)$ and $g(z)$ be analytic on a closed contour $\mathscr{C}$ and in the interior of $\mathscr{C}$. Let $|f(z)| > |g(z)|$ on $\mathscr{C}$. Then $f(z)$ and $f(z) + g(z)$ have the same number of zeros inside contour $\mathscr{C}$.

**Example**

Consider the contour depicted in Fig. A.6(a) that constitutes the boundary of a multiply connected region in the $z$-plane. Let $F(z)$ and $G(z)$ be two polynomials in $z^{-1}$, both of which are analytic on this contour and in the interior of it. Moreover, Let $|F(z)| > |G(z)|$. Then, according to Rouché's theorem, both $F(z)$ and $F(z) + G(z)$ have the same number of zeros inside the contour described in Fig. A.6(a).

Suppose now that we let the radius $R$ of the outside circle $\mathscr{C}$ in Fig. A.6(a) approach infinity. Also, let the separation $l$ between the two straight-line portions of the contour approach zero. Then, in the limit, the region enclosed by the contour described in Fig. A.6(a)

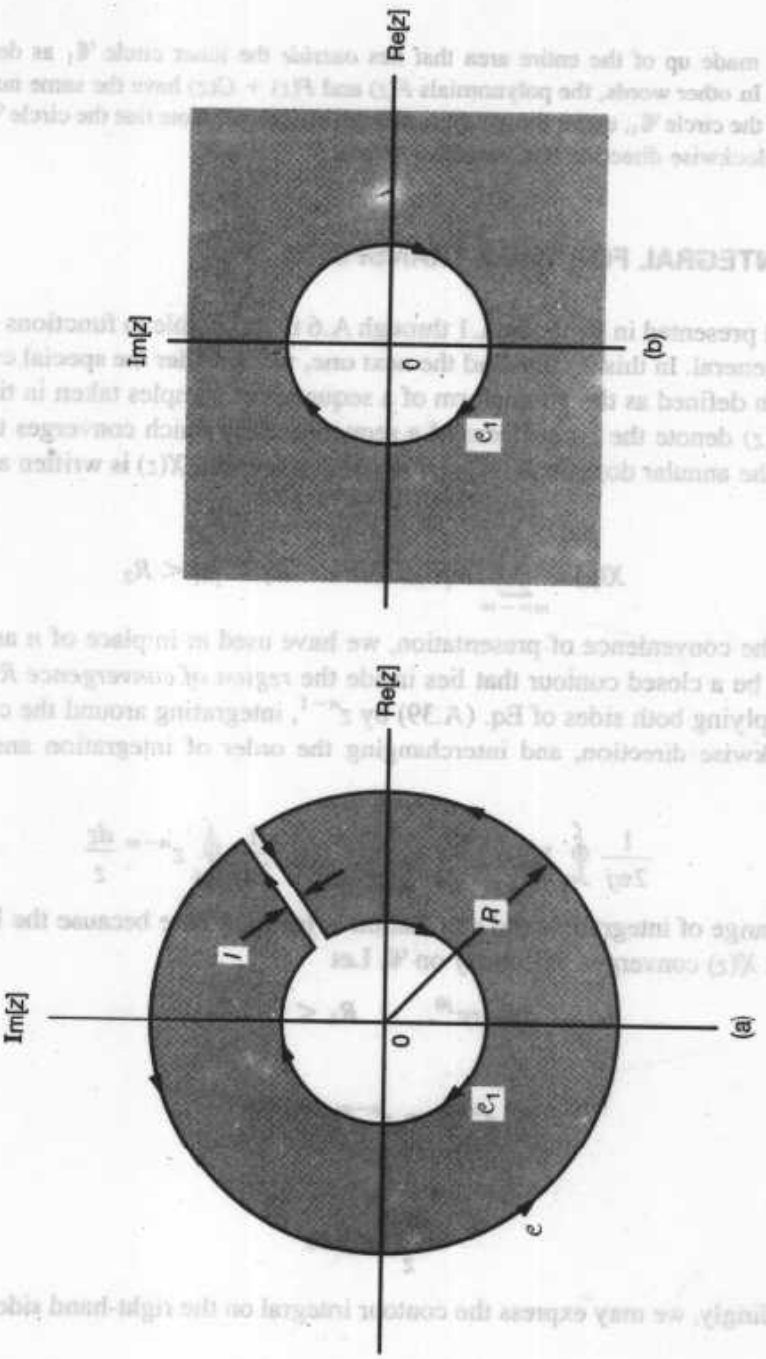



**Figure A.6** (a) Multiply connected region; (b) limiting form of the region in (a) as $R \to \infty$ and $l \to 0$.

will be made up of the entire area that lies *outside* the inner circle $\mathscr{C}_i$ as depicted in Fig. A.6(b). In other words, the polynomials $F(z)$ and $F(z) + G(z)$ have the same number of zeros outside the circle $\mathscr{C}_1$, under the conditions described above. Note that the circle $\mathscr{C}_1$ is traversed in the clockwise direction (i.e., negative sense).

## A.7 INVERSION INTEGRAL FOR THE *z*-TRANSFORM

The material presented in Sections A.1 through A.6 is applicable to functions of a complex variable in general. In this section and the next one, we consider the special case of a complex function defined as the $z$-transform of a sequence of samples taken in time.

Let $X(z)$ denote the $z$-transform of a sequence $x(n)$, which converges to an analytic function in the annular domain $R_1 < |z| < R_2$. By definition, $X(z)$ is written as the Laurent series

$$X(z) = \sum_{m=-\infty}^{\infty} x(m)z^{-m}, \qquad R_1 < |z| < R_2 \tag{A.39}$$

where, for the convenience of presentation, we have used $m$ in place of $n$ as the index of time. Let $\mathscr{C}$ be a closed contour that lies inside the *region of convergence* $R_1 < |z| < R_2$. Then, multiplying both sides of Eq. (A.39) by $z^{n-1}$, integrating around the contour $\mathscr{C}$ in a counterclockwise direction, and interchanging the order of integration and summation, we get

$$\frac{1}{2\pi j}\oint_{\mathscr{C}} X(z)z^n \frac{dz}{z} = \sum_{m=-\infty}^{\infty} x(n)\frac{1}{2\pi j}\oint_{\mathscr{C}} z^{n-m}\frac{dz}{z} \tag{A.40}$$

The interchange of integration and summation is justified here because the Laurent series that defines $X(z)$ converges uniformly on $\mathscr{C}$. Let

$$z = re^{j\theta}, \qquad R_1 < r < R_2 \tag{A.41}$$

Hence,

$$z^{n-m} = r^{n-m}e^{j(n-m)\theta}$$

and

$$\frac{dz}{z} = j\,d\theta$$

Correspondingly, we may express the contour integral on the right-hand side of Eq. (A.40) as

$$\frac{1}{2\pi j}\oint_{\mathscr{C}} z^{n-m}\frac{dz}{z} = \frac{1}{2\pi}\int_0^{2\pi} r^{n-m}e^{j(n-m)\theta}\,d\theta \tag{A.42}$$

$$= \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases}$$

Inserting Eq. (A.42) in (A.40), we get

$$x(n) = \frac{1}{2\pi j} \oint_{\mathscr{C}} X(z) z^n \frac{dz}{z} \tag{A.43}$$

Equation (A.43) is called the *inversion integral formula* for the $z$-transform.

## A.8 PARSEVAL'S THEOREM

Let $X(z)$ denote the $z$-transform of the sequence $x(n)$ with the region of convergence $R_{1x} < |z| < R_{2x}$. Let $Y(z)$ denote the $z$-transform of a second sequence $y(n)$ with the region of convergence $R_{1y} < |z| < R_{2y}$. Then *Parseval's theorem* states that

$$\sum_{n=-\infty}^{\infty} x(n)y^*(n) = \frac{1}{2\pi j} \oint_{\mathscr{C}} X(z) Y^*\left(\frac{1}{z^*}\right) \frac{dz}{z} \tag{A.44}$$

where $\mathscr{C}$ is a closed contour defined in the overlap of the regions of convergence of $X(z)$ and $Y(z)$, both of which are analytic. The function $Y^*(1/z^*)$ is obtained from the $z$-transform $Y(z)$ by using $1/z^*$ in place of $z$, and then complex-conjugating the resulting function. Note that the function $Y^*(1/z^*)$ obtained in this way is analytic too.

To prove Parseval's theorem, we use the inversion integral of Eq. (A.43) to write

$$\sum_{n=-\infty}^{\infty} x(n)y^*(n) = \frac{1}{2\pi j} \sum_{n=-\infty}^{\infty} y^*(n) \oint_{\mathscr{C}} X(z) z^n \frac{dz}{z} \tag{A.45}$$

$$= \frac{1}{2\pi j} \oint_{\mathscr{C}} X(z) \sum_{n=-\infty}^{\infty} y^*(n) z^n \frac{dz}{z}$$

From the definition of the $z$-transform of $y(n)$, namely,

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n}$$

we note that

$$Y^*\left(\frac{1}{z^*}\right) = \sum_{n=-\infty}^{\infty} y^*(n) z^n \tag{A.46}$$

Hence, using Eq. (A.46) in (A.45), we get the result given in Eq. (A.44), and the proof of Parseval's theorem is completed.

# APPENDIX B

# Differentiation with Respect to a Vector

An issue commonly encountered in the study of optimization theory is that of differentiating a cost function with respect to a parameter vector of interest. In the text we used an ordinary gradient operation. The purpose of Appendix B is to address the more difficult issue of differentiating a cost function with respect to a complex-valued parameter vector. We begin by introducing some basic definitions.

## B.1 BASIC DEFINITIONS

Consider a complex function $f(\mathbf{w})$ that is dependent on a parameter vector $\mathbf{w}$. When $\mathbf{w}$ is complex valued, there are two different mathematical concepts that require individual attention: (1) the vector nature of $\mathbf{w}$, and (2) the fact that each element of $\mathbf{w}$ is a complex number.

Dealing with the issue of complex numbers first, let $x_k$ and $y_k$ denote the real and imaginary parts of the $k$th element $w_k$ of the vector $\mathbf{w}$; that is,

$$w_k = x_k + jy_k \tag{B.1}$$

We thus have a function of the real quantities $x_k$ and $y_k$. Hence, we may use Eq. (B.1) to express the real part $x_k$ in terms of the pair of *complex conjugate coordinates* $w_k$ and $w_k^*$ as

$$x_k = \tfrac{1}{2}(w_k + w_k^*) \tag{B.2}$$

and express the imaginary part $y_k$ as

$$y_k = \frac{1}{2j}(w_k - w_k^*) \tag{B.3}$$

where the asterisk denotes complex conjugation. The real quantities $x_k$ and $y_k$ are functions of both $w_k$ and $w_k^*$. It is only when we deal with analytic functions $f$ that we are permitted to abandon the complex-conjugated term $w_k^*$ by virtue of the Cauchy–Riemann equations. However, most functions encountered in physical sciences and engineering are *not* analytic.

The notion of a derivative must tie in with the concept of a differential. In particular, the chain rule of changes of variables must be obeyed. With these important points in mind, we may define certain complex derivatives in terms of real derivatives, as shown by (Schwartz, 1967)

$$\frac{\partial}{\partial w_k} = \frac{1}{2}\left(\frac{\partial}{\partial x_k} - j\frac{\partial}{\partial y_k}\right) \tag{B.4}$$

and

$$\frac{\partial}{\partial w_k^*} = \frac{1}{2}\left(\frac{\partial}{\partial x_k} + j\frac{\partial}{\partial y_k}\right) \tag{B.5}$$

The derivatives defined here satisfy the following two basic requirements:

$$\frac{\partial w_k}{\partial w_k} = 1$$

$$\frac{\partial w_k}{\partial w_k^*} = \frac{\partial w_k^*}{\partial w_k} = 0$$

(An analytic function $f$ satisfies $\partial f/\partial z^* = 0$ everywhere.)

The next issue to be considered is that of differentiation with respect to a vector. Let $w_0, w_1, \ldots, w_{M-1}$ denote the elements of an $M$-by-1 complex vector $\mathbf{w}$. We may extend the use of Eqs. (B.4) and (B.5) to deal with this new situation by writing (Miller, 1974)

$$\frac{\partial}{\partial \mathbf{w}} = \frac{1}{2}\begin{bmatrix} \dfrac{\partial}{\partial x_0} - j\dfrac{\partial}{\partial y_0} \\ \dfrac{\partial}{\partial x_1} - j\dfrac{\partial}{\partial y_1} \\ \vdots \\ \dfrac{\partial}{\partial x_{M-1}} - j\dfrac{\partial}{\partial y_{M-1}} \end{bmatrix} \tag{B.6}$$

and

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \dfrac{\partial}{\partial x_0} + j\dfrac{\partial}{\partial y_0} \\ \dfrac{\partial}{\partial x_1} + j\dfrac{\partial}{\partial y_1} \\ \vdots \\ \dfrac{\partial}{\partial x_{M-1}} + j\dfrac{\partial}{\partial y_{M-1}} \end{bmatrix} \tag{B.7}$$

where we have $w_k = x_k + jy_k$ for $k = 0, 1, \ldots, M-1$. We refer to $\partial/\partial \mathbf{w}$ as a *derivative* with respect to the vector $\mathbf{w}$, and to $\partial/\partial \mathbf{w}^*$ as a *conjugate derivative* also with respect to the vector $\mathbf{w}$. These two derivatives must be considered together. They obey the following relations:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{w}} = \mathbf{I}$$

and

$$\frac{\partial \mathbf{w}}{\partial \mathbf{w}^*} = \frac{\partial \mathbf{w}^*}{\partial \mathbf{w}} = \mathbf{O}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{O}$ is the null matrix.

For subsequent use, we will adopt the definition of (B.7) as the derivative with respect to a complex-valued vector.

## B.2 EXAMPLES

In this section, we illustrate some applications of the derivative defined in Eq. (B.7). The examples are taken from Chapter 5 dealing with optimum linear filtering, and Chapter 11 dealing with the method of least squares.

**Example 1**

Let $\mathbf{p}$ and $\mathbf{w}$ denote two complex-valued $M$-by-1 vectors. There are two inner products, $\mathbf{p}^H\mathbf{w}$ and $\mathbf{w}^H\mathbf{p}$, to be considered

Let $c_1 = \mathbf{p}^H\mathbf{w}$. The conjugate derivative of $c_1$ with respect to the vector $\mathbf{w}$ is

$$\frac{\partial c_1}{\partial \mathbf{w}^*} = \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{p}^H\mathbf{w}) = \mathbf{0} \tag{B.8}$$

where $\mathbf{0}$ is the null vector. Here we note that $\mathbf{p}^H\mathbf{w}$ is an analytic function; see Problem 1 of Chapter 5. We therefore find that the derivative of $\mathbf{p}^H\mathbf{w}$ with respect to $\mathbf{w}$ is zero, in agreement with Eq. (B.8).

Consider next $c_2 = \mathbf{w}^H\mathbf{p}$. The conjugate derivative of $c_2$ with respect to $\mathbf{w}$ is

$$\frac{\partial c_2}{\partial \mathbf{w}^*} = \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H\mathbf{p}) = \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{p}^T\mathbf{w}^*) = \mathbf{p} \tag{B.9}$$

Here we note that $\mathbf{w}^H\mathbf{p}$ is not an analytic function; see Problem 1 of Chapter 5. Hence, the derivative of $\mathbf{w}^H\mathbf{p}$ with respect to $\mathbf{w}^*$ is nonzero, as in Eq. (B.9).

**Example 2**

Consider next the quadratic form

$$c = \mathbf{w}^H\mathbf{R}\mathbf{w}$$

where $\mathbf{R}$ is a Hermitian matrix. The conjugate derivative of $c$ (which is real) with respect to $\mathbf{w}$ is

$$\begin{aligned}\frac{\partial c}{\partial \mathbf{w}^*} &= \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H\mathbf{R}\mathbf{w}) \\ &= \mathbf{R}\mathbf{w}\end{aligned} \tag{B.10}$$

**Example 3**

Consider the real-valued cost function (see Chapter 5)

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

Using the results of Examples 1 and 2, we find that the conjugate derivative of $J$ with respect to the tap-weight vector $\mathbf{w}$ is

$$\frac{\partial J}{\partial \mathbf{w}^*} = -\mathbf{p} + \mathbf{R}\mathbf{w} \tag{B.11}$$

Let $\mathbf{w}_o$ be the optimum value of the tap-weight vector $\mathbf{w}$ for which the cost function $J$ is minimum or, equivalently, the derivative $(\partial J/\partial \mathbf{w}^*) = \mathbf{0}$. Hence, from Eq. (B.11) we deduce that

$$\mathbf{R}\mathbf{w}_o = \mathbf{p} \tag{B.12}$$

This is the matrix form of the Wiener–Hopf equations for a transversal filter operating in a stationary environment.

**Example 4**

Consider the real log-likelihood function (see Chapter 11)

$$l(\tilde{\mathbf{w}}) = F - \frac{1}{\sigma^2}\boldsymbol{\epsilon}^H\boldsymbol{\epsilon} \tag{B.13}$$

where $F$ is a constant and

$$\boldsymbol{\epsilon} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{w}} \tag{B.14}$$

Substituting Eq. (B.14) in (B.13), we get

$$l(\tilde{\mathbf{w}}) = F - \frac{1}{\sigma^2}\mathbf{b}^H\mathbf{b} + \frac{1}{\sigma^2}\mathbf{b}^H\mathbf{A}\tilde{\mathbf{w}} + \frac{1}{\sigma^2}\tilde{\mathbf{w}}^H\mathbf{A}^H\mathbf{b} - \frac{1}{\sigma^2}\tilde{\mathbf{w}}^H\mathbf{A}^H\mathbf{A}\tilde{\mathbf{w}} \tag{B.15}$$

Evaluating the conjugate derivative of $l$ with respect to $\tilde{\mathbf{w}}$, and adapting the results of Examples 1 and 2 to fit our present situation, we get

$$\frac{\partial l}{\partial \tilde{\mathbf{w}}^*} = \frac{1}{\sigma^2}\mathbf{A}^H\mathbf{b} - \frac{1}{\sigma^2}\mathbf{A}^H\mathbf{A}\tilde{\mathbf{w}}$$

Setting $(\partial l/\partial \tilde{\mathbf{w}}^*) = \mathbf{0}$, and then simplifying, we thus get

$$\mathbf{A}^H\mathbf{b} - \mathbf{A}^H\mathbf{A}\mathbf{w}_o = \mathbf{0}$$

where $\mathbf{w}_o$ is the special value of $\tilde{\mathbf{w}}$ for which the log-likelihood function is maximum. Hence,

$$\mathbf{A}^H\mathbf{A}\mathbf{w}_o = \mathbf{A}^H\mathbf{b} \tag{B.16}$$

This is the matrix form of the normal equations for the method of least squares.

## B.3 RELATION BETWEEN THE DERIVATIVE WITH RESPECT TO A VECTOR AND THE GRADIENT VECTOR

Consider the real cost function $J(\mathbf{w})$ that defines the error-performance surface of a linear transversal filter whose tap-weight vector is $\mathbf{w}$. In Chapter 5, we defined the *gradient vector* of the error-performance surface as

$$\nabla J = \begin{bmatrix} \dfrac{\partial J}{\partial x_0} + j\dfrac{\partial J}{\partial y_0} \\ \dfrac{\partial J}{\partial x_1} + j\dfrac{\partial J}{\partial y_1} \\ \vdots \\ \dfrac{\partial J}{\partial x_{M-1}} + j\dfrac{\partial J}{\partial y_{M-1}} \end{bmatrix} \tag{B.17}$$

where $x_k + jy_k$ is the $k$th element of the tap-weight vector $\mathbf{w}$, and $k = 0, 1, \ldots, M-1$. The gradient vector is *normal* to the error-performance surface. Comparing Eqs. (B.7) and (B.17), we see that the conjugate derivative $\partial J/\partial \mathbf{w}^*$ and the gradient vector $\nabla J$ are related by

$$\nabla J = 2\frac{\partial J}{\partial \mathbf{w}^*} \tag{B.18}$$

Thus, except for a scaling factor, the definition of the gradient vector introduced in Chapter 5 is the same as the conjugate derivative defined in Eq. (B.7).

APPENDIX

# C

# Method of Lagrange Multipliers

*Optimization* consists of determining the values of some specified variables that minimize or maximize an *index of performance* or *cost function*, which combines important properties of a system into a single real-valued number. The optimization may be *constrained* or *unconstrained*, depending on whether the variables are also required to satisfy side equations or not. Needless to say, the additional requirement to satisfy one or more side equations complicates the issue of constrained optimization. In this appendix, we derive the classical *method of Lagrange multipliers* for solving the *complex* version of a constrained optimization problem. The notation used in the derivation is influenced by the nature of applications that are of interest to us. We consider first the case when the problem involves a single side equation, followed by the more general case of multiple side equations.

## C.1 OPTIMIZATION INVOLVING A SINGLE EQUALITY CONSTRAINT

Consider the minimization of a real-valued function $f(\mathbf{w})$ that is a quadratic function of a vector $\mathbf{w}$, subject to the *constraint*

$$\mathbf{w}^H\mathbf{s} = g \tag{C.1}$$

where $\mathbf{s}$ is a prescribed vector and $g$ is a complex constant. We may redefine the constraint by introducing a new function $c(\mathbf{w})$ that is linear in $\mathbf{w}$, as shown by

$$\begin{aligned} c(\mathbf{w}) &= \mathbf{w}^H\mathbf{s} - g \\ &= 0 + j0 \end{aligned} \tag{C.2}$$

In general, the vectors $\mathbf{w}$ and $\mathbf{s}$ and the function $c(\mathbf{w})$ are all *complex*. For example, in a beamforming application the vector $\mathbf{w}$ represents a set of complex weights applied to the individual sensor outputs, and $\mathbf{s}$ represents a steering vector whose elements are defined by a prescribed "look" direction; the function $f(\mathbf{w})$ to be minimized represents the mean-square value of the overall beamformer output. In a harmonic retrieval application, $\mathbf{w}$ represents the tap-weight vector of a transversal filter, and $\mathbf{s}$ represents a sinusoidal vector whose elements are determined by the angular frequency of a complex sinusoid contained in the filter input; the function $f(\mathbf{w})$ represents the mean-square value of the filter output. In any event, assuming that the issue is one of minimization, we may state the constrained optimization problem as follows:

$$\begin{aligned} &\text{Minimize a real-valued function } f(\mathbf{w}), \\ &\text{subject to the constraint } c(\mathbf{w}) = 0 + j0 \end{aligned} \tag{C.3}$$

The *method of Lagrange multipliers* converts the problem of constrained minimization described above into one of unconstrained minimization by the introduction of *Lagrange multipliers*. First we use the real function $f(\mathbf{w})$ and the complex constraint function $c(\mathbf{w})$ to define a new real-valued function

$$h(\mathbf{w}) = f(\mathbf{w}) + \lambda_1 \operatorname{Re}[c(\mathbf{w})] + \lambda_2 \operatorname{Im}[c(\mathbf{w})] \tag{C.4}$$

where $\lambda_1$ and $\lambda_2$ are *real Lagrange* multipliers, and

$$c(\mathbf{w}) = \operatorname{Re}[c(\mathbf{w})] + j \operatorname{Im}[c(\mathbf{w})] \tag{C.5}$$

Define a *complex Lagrange multiplier*:

$$\lambda = \lambda_1 + j\lambda_2 \tag{C.6}$$

We may then rewrite Eq. (C.4) in the form

$$h(\mathbf{w}) = f(\mathbf{w}) + \operatorname{Re}[\lambda^* c(\mathbf{w})] \tag{C.7}$$

where the asterisk denotes complex conjugation.

Next, we minimize the function $h(\mathbf{w})$ with respect to the vector $\mathbf{w}$. To do this, we set the conjugate derivative $\partial h/\partial \mathbf{w}^*$ equal to the null vector, as shown by

$$\frac{\partial f}{\partial \mathbf{w}^*} + \frac{\partial}{\partial \mathbf{w}^*}(\operatorname{Re}[\lambda^* c(\mathbf{w})]) = \mathbf{0} \tag{C.8}$$

The system of simultaneous equations, consisting of Eq. (C.8) and the original constraint given in Eq. (C.2), define the optimum solutions for the vector $\mathbf{w}$ and the Lagrange multiplier $\lambda$. We call Eq. (C.8) the *adjoint equation* and Eq. (C.2) the *primal equation* (Dorny, 1975).

## C.2 OPTIMIZATION INVOLVING MULTIPLE EQUALITY CONSTRAINTS

Consider next the minimization of a real function $f(\mathbf{w})$ that is a quadratic function of the vector $\mathbf{w}$, subject to a set of *multiple linear constraints*

$$\mathbf{w}^H\mathbf{s}_k = g_k, \qquad k = 1, 2, \ldots, K \tag{C.9}$$

where the number of constraints, $K$, is less than the dimension of the vector $\mathbf{w}$, and the $g_k$ are complex constants. We may state the multiple-constrained optimization problem as follows:

> Minimize a real function $f(\mathbf{w})$, subject to the constraints $c_k(\mathbf{w}) = 0 + j0$ for $k = 1, 2, \ldots, K$ (C.10)

The solution to this optimization problem is readily obtained by generalizing the previous results of Section C.1. Specifically, we formulate a system of simultaneous equations, consisting of the adjoint equation

$$\frac{\partial f}{\partial \mathbf{w}^*} + \sum_{k=1}^{K} \frac{\partial}{\partial \mathbf{w}^*}(\mathrm{Re}[\lambda_k^* c_k(\mathbf{w})]) = \mathbf{0} \tag{C.11}$$

and the primal equation

$$c_k(\mathbf{w}) = 0^* + \mathrm{j}0, \qquad k = 1, 2, \ldots, K \tag{C.12}$$

This system of equations defines the optimum solutions for the vector $\mathbf{w}$ and the set of complex Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_K$.

### C.3 Example

By way of an example, consider the problem of finding the vector $\mathbf{w}$ that minimizes the function

$$f(\mathbf{w}) = \mathbf{w}^H\mathbf{w} \tag{C.13}$$

and which satisfies the constraint

$$c(\mathbf{w}) = \mathbf{w}^H\mathbf{s} - g = 0 + j0 \tag{C.14}$$

The adjoint equation for this problem is

$$\frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H\mathbf{w}) + \frac{\partial}{\partial \mathbf{w}^*}(\mathrm{Re}[\lambda^*(\mathbf{w}^H\mathbf{s} - g)]) = \mathbf{0} \tag{C.15}$$

Using the rules for differentiation developed in Appendix B, we have

$$\frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H\mathbf{w}) = \mathbf{w}$$

and

$$\frac{\partial}{\partial \mathbf{w}^*}(\mathrm{Re}[\lambda^*(\mathbf{w}^H\mathbf{s} - g)]) = \lambda^*\mathbf{s}$$

Substituting these results in Eq. (C.15), we get

$$\mathbf{w} + \lambda^* \mathbf{s} = \mathbf{0} \tag{C.16}$$

or, equivalently,

$$\mathbf{w}^H + \lambda \mathbf{s}^H = \mathbf{0}^T \tag{C.17}$$

Next, postmultiplying both sides of Eq. (C.17) by s and then solving for the unknown $\lambda$, we obtain

$$\begin{aligned} \lambda &= -\frac{\mathbf{w}^H \mathbf{s}}{\mathbf{s}^H \mathbf{s}} \\ &= -\frac{g}{\mathbf{s}^H \mathbf{s}} \end{aligned} \tag{C.18}$$

Finally, substituting Eq. (C.18) in (C.16) and solving for the optimum value $\mathbf{w}_o$ of the weight vector w, we get

$$\mathbf{w}_o = \left( \frac{g^*}{\mathbf{s}^H \mathbf{s}} \right) \mathbf{s} \tag{C.19}$$

This solution is optimum in the sense that $\mathbf{w}_o$ satisfies the constraint of Eq. (C.14) and has minimum length.

# APPENDIX D

# Estimation Theory

*Estimation theory* is a branch of probability and statistics that deals with the problem of deriving information about properties of random variables and stochastic processes, given a set of observed samples. This problem arises frequently in the study of communications and control systems. *Maximum likelihood* is by far the most general and powerful method of estimation. It was first used by the famous statistician R. A. Fisher in 1906. In principle, the method of maximum likelihood may be applied to any estimation problem with the proviso that we formulate the joint probability density function of the available set of observed data. As such, the method yields almost all the well-known estimates as special cases.

## D.1 LIKELIHOOD FUNCTION

The method of maximum likelihood is based on a relatively simple idea: Different populations generate different data samples and any given data sample is more *likely* to have come from some population than from others (Kmenta, 1971).

Let $f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})$ denote the *conditional joint probability density function* of the *random vector* $\mathbf{U}$ represented by the observed *sample* vector $\mathbf{u}$, where the sample vector $\mathbf{u}$ has $u_1$, $u_2$, . . ., $u_M$ for its elements, and $\boldsymbol{\theta}$ is a *parameter vector* with $\theta_1$, $\theta_2$, . . ., $\theta_K$ as elements.

The method of maximum likelihood is based on the principle that we should estimate the parameter vector $\boldsymbol{\theta}$ by its most *plausible values*, given the observed sample vector $\mathbf{u}$. In other words, the maximum-likelihood estimators of $\theta_1, \theta_2, \ldots, \theta_K$ are those values of the parameter vector for which the conditional joint probability density function $f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})$ is at maximum.

The name *likelihood function*, denoted by $l(\boldsymbol{\theta})$, is given to the conditional joint probability density function $f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})$, viewed as a function of the parameter vector $\boldsymbol{\theta}$. We thus write

$$l(\boldsymbol{\theta}) = f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta}) \tag{D.1}$$

Although the conditional joint probability density function and the likelihood function have exactly the same formula, nevertheless, it is vital that we appreciate the physical distinction between them. In the case of the conditional joint probability density function, the parameter vector $\boldsymbol{\theta}$ is fixed and the observation vector $\mathbf{u}$ is variable. On the other hand, in the case of the likelihood function, the parameter vector $\boldsymbol{\theta}$ is variable and the observation vector $\mathbf{u}$ is fixed.

In many cases, it turns out to be more convenient to work with the natural logarithm of the likelihood function rather than with the likelihood itself. Thus, using $L(\boldsymbol{\theta})$ to denote the *log-likelihood function*, we write

$$\begin{aligned} L(\boldsymbol{\theta}) &= \ln[l(\boldsymbol{\theta})] \\ &= \ln[f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})] \end{aligned} \tag{D.2}$$

The logarithm of $l(\boldsymbol{\theta})$ is a *monotonic transformation* of $l(\boldsymbol{\theta})$. This means that whenever $l(\boldsymbol{\theta})$ decreases, its logarithm $L(\boldsymbol{\theta})$ also decreases. Since $l(\boldsymbol{\theta})$, being a formula for conditional joint probability density function, can never become negative, it follows that there is no problem in evaluating its logarithm $L(\boldsymbol{\theta})$. We conclude therefore that the parameter vector for which the likelihood function $l(\boldsymbol{\theta})$ is at maximum is exactly the same as the parameter vector for which the log-likelihood function $L(\boldsymbol{\theta})$ is at its maximum.

To obtain the $i$th element of the maximum-likelihood estimate of the parameter vector $\boldsymbol{\theta}$, we differentiate the log-likelihood function with respect to $\theta_i$ and set the result equal to zero. We thus get a set of first-order conditions:

$$\frac{\partial L}{\partial \theta_i} = 0, \qquad i = 1, 2, \ldots, K \tag{D.3}$$

The first derivative of the log-likelihood function with respect to parameter $\theta_i$ is called the *score* for that parameter. The vector of such parameters is known as the *scores vector* (i.e., the gradient vector). The scores vector is identically zero at the maximum-likelihood estimates of the parameters, that is, at the values of $\boldsymbol{\theta}$ that result from the solutions of Eq. (D.3).

To find how effective the method of maximum likelihood is, we can compute the *bias* and *variance* for the estimate of each parameter. However, this is frequently difficult to do. Rather than approach the computation directly, we may derive a *lower bound* on the

variance of any *unbiased* estimate. We say an estimate is unbiased if the average value of the estimate equals the parameter we are trying to estimate. Later we show how the variance of the maximum-likelihood estimate compares with this lower bound.

## D.2 CRAMER–RAO INEQUALITY

Let U be a random vector with conditional joint probability density function $f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})$, where $\mathbf{u}$ is the observed sample vector with elements $u_1, u_2, \ldots, u_M$ and $\boldsymbol{\theta}$ is the parameter vector with elements $\theta_1, \theta_2, \ldots, \theta_K$. Using the definition of Eq. (D.2) for the log-likelihood function $L(\hat{\mathbf{u}})$ in terms of the conditional joint probability density function $f_{\mathbf{U}}(\mathbf{u}|\boldsymbol{\theta})$, we form the $K$-by-$K$ matrix:

$$\mathbf{J} = -\begin{bmatrix} E\left[\frac{\partial^2 L}{\partial\theta_1^2}\right] & E\left[\frac{\partial^2 L}{\partial\theta_1\partial\theta_2}\right] & \cdots & E\left[\frac{\partial^2 L}{\partial\theta_1\partial\theta_K}\right] \\ E\left[\frac{\partial^2 L}{\partial\theta_2\partial\theta_1}\right] & E\left[\frac{\partial^2 L}{\partial\theta_2^2}\right] & \cdots & E\left[\frac{\partial^2 L}{\partial\theta_2\partial\theta_K}\right] \\ \vdots & \vdots & & \vdots \\ E\left[\frac{\partial^2 L}{\partial\theta_K\partial\theta_1}\right] & E\left[\frac{\partial^2 L}{\partial\theta_K\partial\theta_2}\right] & \cdots & E\left[\frac{\partial^2 L}{\partial\theta_K^2}\right] \end{bmatrix} \tag{D.4}$$

The matrix $\mathbf{J}$ is called *Fisher's information matrix.*

Let $\mathbf{I}$ denote the inverse of Fisher's information matrix $\mathbf{J}$. Let $I_{ii}$ denote the $i$th diagonal element (i.e., the element in the $i$th row and $i$th column) of the inverse matrix $\mathbf{I}$. Let $\hat{\theta}_i$ be *any* unbiased estimate of the parameter $\theta_i$, based on the observed sample vector $\mathbf{u}$. We may then write (Van Trees, 1968; Nahi, 1969)

$$\text{var}[\hat{\theta}_i] \geq I_{ii}, \qquad i = 1, 2, \ldots, K \tag{D.5}$$

Equation (D.5) is called the *Cramér–Rao inequality*. This theorem enables us to construct a lower limit (greater than zero) for the variance of any unbiased estimator, provided, of course, that we know the functional form of the log-likelihood function. The lower limit specified in the theorem is called the *Cramér–Rao lower bound* (CRLB).

If we can find an unbiased estimator whose variance equals the Cramér–Rao lower bound, then according to the theorem of Eq. (D.5) there is no other unbiased estimator with a smaller variance. Such an estimator is said to be *efficient.*

## D.3 PROPERTIES OF MAXIMUM-LIKELIHOOD ESTIMATORS

Not only is the method of maximum likelihood based on an intuitively appealing idea (that of choosing those parameters from which the actually observed sample vector is most likely to have come), but also the resulting estimates have some desirable properties.

Indeed, under quite general conditions, the following *asymptotic* properties may be proved (Kmenta, 1971):

1. Maximum-likelihood estimators are *consistent*. That is, the value of $\theta_i$ for which the score $\partial L/\partial\theta_i$ is identically zero *converges in probability* to the true value of the parameter $\theta_i$, $i = 1, 2, \ldots, K$, as the *sample size* $M$ approaches infinity.
2. Maximum-likelihood estimators are *asymptotically efficient*; that is,

$$\lim_{M\to\infty}\left\{\frac{\text{var}[\theta_{i,\text{ml}} - \theta_i]}{I_{ii}}\right\} = 1, \qquad i = 1, 2, \ldots, K$$

   where $\theta_{i,\text{ml}}$ is the maximum-likelihood estimate of parameter $\theta_i$, and $I_{ii}$ is the $i$th diagonal element of the inverse of Fisher's information matrix.
3. Maximum-likelihood estimators are *asymptotically Gaussian*.

In practice, we find that the large-sample (asymptotic) properties of maximum-likelihood estimators hold rather well for sample size $M \geq 50$.

## D.4 CONDITIONAL MEAN ESTIMATOR

Another classic problem in estimation theory is that of the *Bayes estimation of a random parameter*. There are different answers to this problem, depending on how the *cost function* in the Bayes estimation is formulated (Van Trees, 1968). A particular type of the Bayes estimator of interest to us in this book is the so-called *conditional mean estimator*. We now wish to do two things: (1) derive the formula for the conditional mean estimator from first principles, and (2) show that such an estimator is the same as a minimum mean-squared-error estimator.

Consider a *random parameter* $x$. We are given an *observation* $y$ that depends on $x$, and the requirement is to estimate $x$. Let $\hat{x}(y)$ denote an *estimate* of the parameter $x$; the symbol $\hat{x}(y)$ emphasizes the fact that the estimate is a function of the observation $y$. Let $C(x, \hat{x}(y))$ denote a *cost function*. Then, according to Bayes estimation theory, we may write an expression for the *risk* as follows (Van Trees, 1968):

$$\begin{aligned}\mathcal{R} &= E[C(x, \hat{x}(y))] \\ &= \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} C(x, \hat{x}(y)) f_{X,Y}(x, y)\, dy\end{aligned} \tag{D.6}$$

where $f_{X,Y}(x, y)$ is the joint probability density function of $x$ and $y$. For a specified cost function $C(x, \hat{x}(y))$, the *Bayes estimate* is defined as the estimate $\hat{x}(y)$ that *minimizes* the risk $\mathcal{R}$.

A cost function of particular interest (and which is very much in the spirit of the material covered in this book) is the *mean-squared error*. In this case, the cost function is

specified as the square of the *estimation error*. The estimation error is itself defined as the difference between the actual parameter value $x$ and the estimate $\hat{x}(y)$, as shown by

$$\epsilon = x - \hat{x}(y) \tag{D.7}$$

Correspondingly, the cost function is defined by

$$C(x, \hat{x}(y)) = C(x - \hat{x}(y)) \tag{D.8}$$

or, more simply,

$$C(\epsilon) = \epsilon^2 \tag{D.9}$$

Thus, the cost function varies with the estimation error $\epsilon$ in the manner indicated in Fig. D.1. It is assumed here that $x$ and $y$ are both real. Accordingly, for the situation at hand, we may rewrite Eq. (D.6) as follows:

$$\mathcal{R}_{\text{ms}} = \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} [x - \hat{x}(y)]^2 f_{X,Y}(x, y)\, dy \tag{D.10}$$

where the subscripts in the risk $\mathcal{R}_{\text{ms}}$ indicate the use of mean-squared error as its basis.

Using *Bayes' rule*, we have

$$f_{X,Y}(x, y) = f_X(x|y)\, f_Y(y) \tag{D.11}$$

where $f_X(x|y)$ is the conditional probability density function of $x$, given $y$, and $f_Y(y)$ is the (marginal) probability density function of $y$. Hence, using Eq. (D.11) in (D.10), we have

$$\mathcal{R}_{\text{ms}} = \int_{-\infty}^{\infty} dy f_Y(y) \int_{-\infty}^{\infty} [x - \hat{x}(y)]^2 f_X(x|y)\, dx \tag{D.12}$$

We now recognize that the inner integral and $f_Y(y)$ in Eq. (D.12) are both nonnegative. We may therefore minimize the risk $\mathcal{R}_{\text{ms}}$ by simply minimizing the inner integral. Let the estimate so obtained be denoted by $\hat{x}_{\text{ms}}(y)$. We find $\hat{x}_{\text{ms}}(y)$ by differentiating the inner integral with respect to $\hat{x}(y)$ and then setting the result equal to zero.

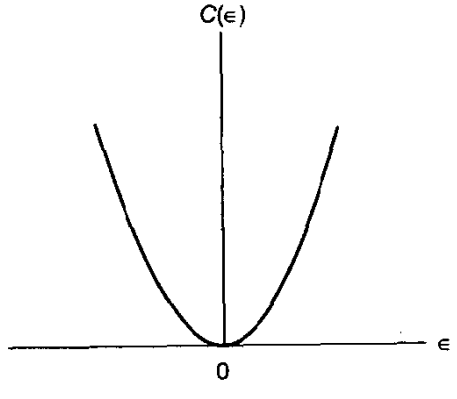



**Figure D.1** Mean-squared error as the cost function.

To simplify the presentation, let $I$ denote the inner integral in Eq. (D.12). Then differentiating $I$ with respect to $\hat{x}(y)$ yields

$$\frac{dI}{d\hat{x}} = -2\int_{-\infty}^{\infty} x f_X(x|y)\,dx + 2\hat{x}(y)\int_{-\infty}^{\infty} f_X(x|y)\,dx \tag{D.13}$$

The second integral on the right-hand side of Eq. (D.13) represents the total area under a probability density function and therefore equals 1. Hence, setting the derivative $dI/d\hat{x}$ equal to zero, we obtain

$$\hat{x}_{\text{ms}}(y) = \int_{-\infty}^{\infty} x f_X(x|y)\,dx \tag{D.14}$$

The solution defined by Eq. (D.14) is a unique minimum.

The estimator $\hat{x}_{ms}(y)$ defined in Eq. (D.14) is naturally a *minimum mean-squared-error estimator*, hence the use of the subscripts "ms." For another interpretation of this estimator, we recognize that the integral on the right-hand side of Eq. (D.14) is just the *conditional mean* of the parameter $x$, given the observation $y$.

We therefore conclude that the minimum mean-squared error estimator and the conditional mean estimator are indeed one and the same. In other words, we have

$$\hat{x}_{\text{ms}}(y) = E[x|y] \tag{D.15}$$

Substituting Eq. (D.15) for the estimate $\hat{x}(y)$ in Eq. (D.12), we find that the inner integral is just the *conditional variance* of the parameter $x$, given $y$. Accordingly, the minimum value of the risk $\mathscr{R}_{\text{ms}}$ is just the average of this conditional variance over all observations $y$.

# APPENDIX E

# Maximum-Entropy Method

The *maximum-entropy method (MEM)* was originally devised by Burg (1967, 1975) to overcome fundamental limitations of Fourier-based methods for estimating the power spectrum of a stationary stochastic process. The basic idea of MEM is to choose the particular spectrum that corresponds to the most *random* or the most *unpredictable* time series whose autocorrelation function agrees with a set of known values. This condition is equivalent to an extrapolation of the autocorrelation function of the available time series by *maximizing* the *entropy* of the process, hence the name of the method. Entropy is a measure of the average information content of the process (Shannon, 1948). Thus, MEM bypasses the problems that arise from the use of window functions, a feature that is common to all Fourier-based methods of spectrum analysis. In particular, MEM avoids the use of a periodic extension of the data (as in the method based on smoothing the periodogram and its computation using the fast Fourier transform algorithm) or of the assumption that data outside the available record length are zero (as in the Blackman–Tukey method based on the sample autocorrelation function). An important feature of the MEM spectrum is that it is *nonnegative at all frequencies*, which is precisely the way it should be.

## E.1 MAXIMUM-ENTROPY SPECTRUM

Suppose that we are given $2M + 1$ values of the autocorrelation function of a stationary stochastic process $u(n)$ of zero mean. We wish to obtain the special value of the power

spectrum of the process that corresponds to the most random time series whose autocorrelation function is consistent with the set of $2M + 1$ known values. In terms of information theory, this statement corresponds to the *principle of maximum entropy* (Jaynes, 1982).

In the case of a set of Gaussian-distributed random variables of zero mean, the entropy is given by (Middleton, 1960)

$$H = \tfrac{1}{2}\ln[\det(\mathbf{R})] \tag{E.1}$$

where $\mathbf{R}$ is the correlation matrix of the process. When the process is of infinite duration, however, we find that the entropy $H$ diverges, and so we cannot use it as a measure of information content. To overcome this divergence problem, we may use the *entropy rate* defined by

$$\begin{aligned} h &= \lim_{M\to\infty} \frac{H}{M+1} \\ &= \lim_{M\to\infty} \tfrac{1}{2}\ln[\det(\mathbf{R})]^{1/(M+1)} \end{aligned} \tag{E.2}$$

Let $S(\omega)$ denote the power spectrum of the process $u(n)$. The limiting form of the determinant of the correlation matrix $\mathbf{R}$ is related to the power spectrum $S(\omega)$ as follows (see Problem 14 of Chapter 4):

$$\lim_{M\to\infty} [\det(\mathbf{R})]^{1/(M+1)} = \exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S(\omega)\, d\omega\right\} \tag{E.3}$$

Hence, substituting Eq. (E.3) in (E.2), we get

$$h = \frac{1}{4\pi}\int_{-\pi}^{\pi} \ln[S(\omega)]d\omega \tag{E.4}$$

Although this relation was derived on the assumption that the process $u(n)$ is Gaussian, nevertheless, the form of the relation is valid for any stationary process.

We may now restate the MEM problem in terms of the entropy rate. We wish to find a real positive-valued power spectrum characterized by entropy rate $h$, satisfying two simultaneous requirements:

1. The entropy rate $h$ is *stationary* with respect to the *unknown* values of the autocorrelation function of the process.
2. The power spectrum is *consistent* with respect to the *known* values of the autocorrelation function of the process.

We will address these two requirements in turn.

Since the autocorrelation sequence $r(m)$ and power spectrum $S(\omega)$ of a stationary process $u(n)$ form a discrete-time Fourier-transform pair, we write

$$S(\omega) = \sum_{m=-\infty}^{\infty} r(m)\exp(-jm\omega) \tag{E.5}$$

Equation (E.5) assumes that the sampling period of the process $u(n)$ is normalized to unity. Substituting Eq. (E.5) in (E.4), we get

$$h = \frac{1}{4\pi}\int_{-\pi}^{\pi} \ln\left[\sum_{m=-\infty}^{\infty} r(m)\exp(-jm\omega)\right] d\omega \tag{E.6}$$

We extrapolate the autocorrelation sequence $r(m)$ outside the range of known values, $-M \le m \le M$, by choosing the unknown values of the autocorrelation function in such a way that no information or entropy is added to the process. That is, we impose the condition

$$\frac{\partial h}{\partial r(m)} = 0, \qquad |m| \ge M + 1 \tag{E.7}$$

Hence, differentiating Eq. (E.6) with respect to $r(m)$ and setting the result equal to zero, we find that the conditions for *maximum entropy* are as follows:

$$\int_{-\pi}^{\pi} \frac{\exp(-jm\omega)}{S_{\text{MEM}}(\omega)}\, d\omega = 0, \qquad |m| \ge M + 1 \tag{E.8}$$

where $S_{\text{MEM}}(\omega)$ is the special value of the power spectrum resulting from the imposition of the condition in Eq. (E.7). Equation (E.8) implies that the power spectrum $S_{\text{MEM}}(\omega)$ is expressible in the form of a truncated Fourier series:

$$\frac{1}{S_{\text{MEM}}(\omega)} = \sum_{k=-M}^{M} c_k \exp(-jk\omega) \tag{E.9}$$

The complex Fourier coefficient $c_k$ of the expansion satisfies the Hermitian condition

$$c_k^* = c_{-k} \tag{E.10}$$

so as to ensure that $S_{\text{MEM}}(\omega)$ is real for all $\omega$.

The next requirement is to make the power spectrum $S_{\text{MEM}}(\omega)$ consistent with the set of known values of the autocorrelation function $r(m)$ for the interval $-M \le m \le M$. Since $r(m)$ is a Hermitian function, we need only concern ourselves with $0 \le m \le M$. Accordingly, $r(m)$ must equal the inverse discrete-time Fourier transform of $S_{\text{MEM}}(\omega)$ for $0 \le m \le M$, as shown by

$$r(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{\text{MEM}}(\omega)\exp(jm\omega)\, d\omega, \qquad 0 \le m \le M \tag{E.11}$$

Therefore, substituting Eq. (E.9) in (E.11), we get

$$r(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\exp(jm\omega)}{\sum_{k=-M}^{M} c_k \exp(-jk\omega)}\, d\omega, \qquad 0 \le m \le M \tag{E.12}$$

Clearly, in the set of complex Fourier coefficients $\{c_k\}$, we have the available degrees of freedom needed to satisfy the conditions of Eq. (E.12).

To proceed with the analysis, however, we find it convenient to use $z$-transform notation by changing from the variable $\omega$ to $z$. Define

$$z = \exp(j\omega) \tag{E.13}$$

Hence,

$$d\omega = \frac{1}{j}\frac{dz}{z}$$

and so we rewrite Eq. (E.12) in terms of the variable $z$ as the contour integral

$$r(m) = \frac{1}{j2\pi}\oint \frac{z^{m-1}}{\sum_{k=-M}^{M} c_k z^{-k}}\, dz, \qquad 0 \le m \le M \tag{E.14}$$

The contour integration in Eq. (E.14) is performed on the unit circle in the $z$-plane in a counterclockwise direction. Since the complex Fourier coefficient $c_k$ satisfies the Hermitian condition of Eq. (E.10), we may express the summation in the denominator of the integral in Eq. (E.14) as the product of two polynomials, as follows:

$$\sum_{k=-M}^{M} c_k z^{-k} = G(z)G^*\left(\frac{1}{z^*}\right) \tag{E.15}$$

where

$$G(z) = \sum_{k=0}^{M} g_k z^{-k} \tag{E.16}$$

and

$$G^*\left(\frac{1}{z^*}\right) = \sum_{k=0}^{M} g_k^* z^k \tag{E.17}$$

We choose the first polynomial $G(z)$ to be minimum phase, in that its zeros are all located inside the unit circle in the $z$-plane. Correspondingly, we choose the second polynomial $G^*(1/z^*)$ to be maximum phase, in that its zeros are all located outside the unit circle in the $z$-plane. Moreover, the zeros of these two polynomials are the inverse of each other with respect to the unit circle. Thus, substituting Eq. (E.15) in (E.14), we get

$$r(m) = \frac{1}{j2\pi}\oint \frac{z^{m-1}}{G(z)G^*(1/z^*)}\, dz, \qquad 0 \le m \le M \tag{E.18}$$

We next form the summation

$$\sum_{k=0}^{M} g_k r(m-k) = \frac{1}{j2\pi} \oint \frac{z^{m-1} \sum_{k=0}^{M} g_k z^{-k}}{G(z)G^*(1/z^*)}\, dz$$
$$= \frac{1}{j2\pi} \oint \frac{z^{m-1}}{G^*(1/z^*)}\, dz, \qquad 0 \le m \le M \tag{E.19}$$

where in the first line we have used Eq. (E.18), and in the second line we have used Eq. (E.16).

To evaluate the contour integral of Eq. (E.19), we use *Cauchy's residue theorem* of complex variable theory (see Appendix A). According to this theorem, the contour integral equals $2\pi j$ times the sum of *residues* of the poles of the integral $z^{m-1}/G^*(1/z^*)$ that lie inside the unit circle used as the contour of integration. Since the polynomial $G^*(1/z^*)$ is chosen to have no zeros inside the unit circle, it follows that the integral in Eq. (E.19) is analytic on and inside the unit circle for $m \ge 1$. For $m = 0$ the integral has a simple pole at $z = 0$ with a *residue* equal to $1/g_0^*$. Hence, application of Cauchy's residue theorem yields

$$\oint \frac{z^{m-1}}{G^*(1/z^*)}\, dz = \begin{cases} \dfrac{2\pi j}{g_0^*}, & m = 0 \\ 0, & m = 1, 2, \ldots, M \end{cases} \tag{E.20}$$

Thus, substituting Eq. (E.20) in (E.19), we get

$$\sum_{k=0}^{M} g_k r(m-k) = \begin{cases} \dfrac{1}{g_0^*}, & m = 0 \\ 0, & m = 1, 2, \ldots, M \end{cases} \tag{E.21}$$

We recognize that the set of $(M + 1)$ equations in (E.21) has a mathematical form similar to that of the augmented Wiener–Hopf equations for forward prediction of order $M$ (see Chapter 6). In particular, by comparing Eqs. (E.21) and (6.16), we deduce that

$$g_k^* = \frac{1}{g_0 P_M} a_{M,k}, \qquad 0 \le k \le M \tag{E.22}$$

where the $a_{M,k}$ are coefficients of a prediction-error filter of order $M$, and $P_M$ is the average output power of the filter. Since $a_{M,0} = 1$ for all $M$, by definition, we find from Eq. (E.22) that for $k = 0$:

$$|g_0|^2 = \frac{1}{P_M} \tag{E.23}$$

Finally, substituting Eqs. (E.15), (E.22), and (E.23) in (E.9) with $z = \exp(j\omega)$, we get

$$S_{\text{MEM}}(\omega) = \frac{P_M}{\left|1 + \sum_{k=1}^{M} a_{M,k} e^{-jk\omega}\right|^2} \tag{E.24}$$

We refer to the formula of Eq. (E.24) as the *MEM spectrum*.

## E.2 COMPUTATION OF THE MEM SPECTRUM

The formula for the MEM spectrum given in Eq. (E.24) may be recast in the alternative form

$$S_{\text{MEM}}(\omega) = \frac{1}{\sum_{k=-M}^{M} \psi(k) e^{-j\omega k}} \tag{E.25}$$

where $\psi(k)$ is defined in terms of the prediction-error filter coefficients as follows:

$$\psi(k) = \begin{cases} \dfrac{1}{P_M} \displaystyle\sum_{i=0}^{M-k} a_{M,i} a^*_{M,i+k} & \text{for } k = 0, 1, \ldots, M \\ \psi^*(-k) & \text{for } k = -M, \ldots, -1 \end{cases} \tag{E.26}$$

The parameter $\psi(k)$ may be viewed as some form of a *correlation coefficient for prediction-error filter coefficients*.

Examination of the denominator polynomial in Eq. (E.25) reveals that it represents the *discrete Fourier transform* of the sequence $\psi(k)$. Accordingly, we may use the *fast Fourier transform (FFT)* algorithm (Oppenheim and Schafer, 1989) for the efficient computation of the denominator polynomial and therefore the MEM spectrum. Given the autocorrelation sequence $r(0), r(1), \ldots, r(M)$, pertaining to a wide-sense stationary stochastic process $u(n)$, we may now summarize an efficient procedure for computing the MEM spectrum:

**Step 1: Levinson–Durbin Recursion.**

Initialize the algorithm by setting

$$a_{0,0} = 1$$
$$P_0 = r(0)$$

For $m = 1, 2, \ldots, M$, compute

$$\kappa_m = -\frac{1}{P_{m-1}} \sum_{i=0}^{M-1} r(i-m) a_{m-1,i}$$

$$a_{m,i} = \begin{cases} 1 & \text{for } i = 0 \\ a_{m-1,i} + \kappa_m a^*_{m-1,m-i} & \text{for } i = 1, 2, \ldots, m-1 \\ \kappa_m & \text{for } i = m \end{cases}$$

$$P_m = P_{m-1}(1 - |\kappa_m|^2)$$

**Step 2: Correlation for Prediction-Error Filter Coefficients.**

Compute the correlation coefficient

$$\psi(k) = \begin{cases} \dfrac{1}{P_M} \displaystyle\sum_{i=0}^{M-k} a_{M,i} a^*_{M,i+k} & \text{for } k = 0, 1, \ldots, M \\ \psi^*(-k) & \text{for } k = -M, \ldots, -1 \end{cases} \tag{E.26}$$

**Step 3: MEM Spectrum.**

Use the fast Fourier transform algorithm to compute the MEM spectrum for varying angular frequency:

$$S_{\text{MEM}}(\omega) = \frac{1}{\displaystyle\sum_{k=-M}^{M} \psi(k) e^{-j\omega k}}$$

# APPENDIX F

# Minimum-Variance Distortionless Response Spectrum

In Section 5.8, we derived the formula for the *minimum-variance distortionless response (MVDR)* spectrum for a wide-sense stationary stochastic process. In this appendix we do two things. First, we develop a fast algorithm for computing the MVDR spectrum, given the ensemble-averaged correlation matrix of the process (Musicus, 1985); the algorithm exploits the Toeplitz property of the correlation matrix. Second, in deriving the algorithm, we develop an insightful relationship between the MVDR and MEM spectra.

## F.1 FAST MVDR SPECTRUM COMPUTATION

Consider a zero-mean wide-sense stationary stochastic process $u(n)$ characterized by an $(M + 1)$-by-$(M + 1)$ ensemble-averaged correlation matrix $\mathbf{R}$. The *minimum-variance distortionless response (MVDR) spectrum* for such a process is defined in terms of the inverse matrix $\mathbf{R}^{-1}$ by

$$S_{\text{MVDR}}(\omega) = \frac{1}{\mathbf{s}^H(\omega)\mathbf{R}^{-1}\mathbf{s}(\omega)} \tag{F.1}$$

where

$$\mathbf{s}(\omega) = [1, e^{-j\omega}, e^{-j2\omega}, \ldots, e^{-jM\omega}]^T$$

Let $R^{-1}_{l,k}$ denote the $(l, k)$th element of $\mathbf{R}^{-1}$. Then, we may rewrite Eq. (F.1) in the form

$$S_{\text{MVDR}}(\omega) = \frac{1}{\sum_{k=-M}^{M} \mu(k) e^{-j\omega k}} \tag{F.2}$$

where

$$\mu(k) = \sum_{l=\max(0,k)}^{\min(M-k,M)} R^{-1}_{l,l+k} \tag{F.3}$$

We recognize that the correlation matrix $\mathbf{R}$ is Toeplitz. We may therefore use the *Gohberg–Semencul formula* (Kailath et al., 1979) to express the $(l,k)$th element of the inverse matrix $\mathbf{R}^{-1}$ as follows:

$$R^{-1}_{l,k} = \frac{1}{P_M} \sum_{i=0}^{l} (a_{M,i} a^*_{M,i+k-l} - a^*_{M,M+1-i} a_{M,M+1-i-k+l}), \qquad k \geq l \tag{F.4}$$

where $1, a_{M,1}, \ldots, a_{M,M}$ are the coefficients of a prediction-error filter of order $M$, and $P_M$ is the average prediction-error power. Substituting Eq. (F.4) in (F.3) and confining attention to $k \geq 0$, we get

$$\mu(k) = \frac{1}{P_M} \sum_{l=0}^{M-k} \sum_{i=0}^{l} a_{M,i} a^*_{M,i+k} - \frac{1}{P_M} \sum_{l=0}^{M-k} \sum_{i=0}^{l} a^*_{M,M+1-i} a_{M,M+1-i-k} \tag{F.5}$$

Interchanging the order of summations and setting $j = M + 1 - i - k$, we may rewrite $\mu(k)$ as

$$\mu(k) = \frac{1}{P_M} \sum_{i=0}^{M-k} \sum_{l=i}^{M-k} a_{M,i} a^*_{M,i+k} - \frac{1}{P_M} \sum_{j=1}^{M+1-k} \sum_{l=M+1-j-k}^{M-k} a^*_{M,j+k} a_{M,j} \tag{F.6}$$

The terms that do not involve the index $l$ permit us to collapse the summation over $l$ into a multiplicative integer constant. We may thus combine the two summations in Eq. (F.6). Moreover, we may use the Levinson–Durbin recursion for computing the prediction-error filter coefficients. Given the autocorrelation sequence $r(0), r(1), \ldots, r(M)$, we may now formulate a fast algorithm for computing the MVDR spectrum as follows (Musicus, 1985):

**Step 1: Levinson–Durbin Recursion.**

Initialize the algorithm by setting

$$a_{0,0} = 1$$

$$P_0 = r(0)$$

Hence, compute for $m = 1, 2, \ldots, M$:

$$\kappa_m = -\frac{1}{P_{m-1}} \sum_{i=0}^{m-1} r(i-m) a_{m-1,i}$$

$$a_{m,i} = \begin{cases} 1 & \text{for } i = 0 \\ a_{m-1,i} + \kappa_m a^*_{m-1,m-i} & \text{for } i = 1, 2, \ldots, m-1 \\ \kappa_m & \text{for } i = m \end{cases}$$

$$P_m = P_{m-1}(1 - |\kappa_m|^2)$$

**Step 2: Correlation of the Predictor Coefficients.**

Compute the parameter $\mu(k)$ for varying $k$:

$$\mu(k) = \begin{cases} \dfrac{1}{P_M} \displaystyle\sum_{i=0}^{M-k} (M + 1 - k - 2i) a_{M,i} a^*_{M,i+k} & \text{for } k = 0, \ldots, M \\ \mu^*(-k) & \text{for } k = -M, \ldots, -1 \end{cases} \tag{F.7}$$

**Step 3: MVDR Spectrum Computation.**

Use the fast Fourier transform algorithm to compute the MVDR spectrum for varying angular frequency:

$$S_{\text{MVDR}}(\omega) = \frac{1}{\displaystyle\sum_{k=-M}^{M} \mu(k) e^{-j\omega k}} \tag{F.8}$$

## F.2 COMPARISON OF MVDR AND MEM SPECTRA

Comparing the formula for computing the MVDR spectrum with that for computing the MEM spectrum, we see that the only difference between the MVDR formula in Eq. (F.8) and the MEM formula in Eq. (E.25) lies in the definitions of their respective correlations of predictor coefficients. In particular, a *linear taper* is used in the definition of $\mu(k)$ given in Eq. (F.7) for the MVDR formula. On the other hand, the definition of the corresponding parameter $\psi(k)$ given in Eq. (E.26) for the MEM formula does *not* involve a taper. This means that for a large-enough model order $M$, such that $a_{M,i} = 0$ for $i > M/2$, the linear taper involved in the computation of $\mu(k)$ acts like a triangular window on the product terms $a_{M,i} a^*_{M,i+k}$. This has the effect of deemphasizing higher-order terms with large $i$ for large values of lag $k$ (Musicus, 1985). Accordingly, for a given process, an MVDR spectrum is *smoother* in appearance than the corresponding MEM spectrum.

# APPENDIX G

# Gradient Adaptive Lattice Algorithm

The adaptive lattice filtering algorithms considered in Chapter 15 are all *exact* manifestations of recursive least-squares estimation, exact in the sense that no approximations are made in their derivations. In this appendix we derive another adaptive lattice filtering algorithm known as the *gradient adaptive lattice (GAL) algorithm* (Griffiths, 1977, 1978), which is a natural extension of the least-mean-square (LMS) algorithm.

Consider a single-stage lattice structure the input–output relation of which is characterized by a single parameter, namely, the *reflection coefficient* $\kappa_m$. We assume that the input data are wide-sense stationary and that $\kappa_m$ is complex valued. Define a cost function for this stage as

$$J_m = E[|f_m(n)|^2 + |b_m(n)|^2] \tag{G.1}$$

where $f_m(n)$ is the forward prediction error and $b_m(n)$ is the backward prediction error, both measured at the output of the stage; $E$ is the statistical expectation operator. The input–output relations of the lattice stage under consideration are described by

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

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

The gradient of the cost function $J_m$ with respect to the real and imaginary parts of the reflection coefficient $\kappa_m$ is given by

$$\nabla J_m = 2E[f_m^*(n)b_{m-1}(n-1) + b_m(n)f_{m-1}^*(n)] \tag{G.2}$$

where $f_{m-1}(n)$ is the forward prediction error and $b_{m-1}(n-1)$ is the delayed backward prediction error, both measured at the input of the lattice stage; the other two prediction errors in Eq. (G.2) refer to the output of the stage. Following the development of the LMS algorithm as presented in Chapter 9, we may use instantaneous estimates of the expectations in Eq. (G.2) and thus write

$$E[f_m^*(n)b_{m-1}(n-1)] \simeq f_m^*(n)b_{m-1}(n-1)$$
$$E[b_m(n)f_{m-1}^*(n)] \simeq b_m(n)f_{m-1}^*(n)$$

Correspondingly, we may express the *instantaneous estimate* of the gradient $\nabla_m J$ as

$$\hat{\nabla}_m J(n) = 2[f_m^*(n)b_{m-1}(n-1) + b_m(n)f_{m-1}^*(n)] \tag{G.3}$$

Let $\hat{\kappa}_m(n-1)$ denote the *old estimate* of the reflection coefficient $\kappa_m$ of the $m$th lattice stage. Let $\hat{\kappa}_m(n)$ denote the *updated estimate* of this reflection coefficient. We may compute this updated estimate by adding to the old estimate $\kappa_m(n-1)$ a *correction* term proportional to the gradient estimate $\hat{\nabla}_m J(n)$, as shown by

$$\hat{\kappa}_m(n) = \hat{\kappa}_m(n-1) - \frac{1}{2}\mu_m(n)\hat{\nabla}_m J(n) \tag{G.4}$$

where $\mu_m$ denotes a *time-varying step-size parameter* associated with the $m$th lattice stage. Substituting Eq. (G.3) in (G.4), we thus get

$$\hat{\kappa}_m(n) = \hat{\kappa}_m(n-1) - \mu_m(n)[f_m^*(n)b_{m-1}(n-1) + b_m(n)f_{m-1}^*(n)] \tag{G.5}$$

The adaptation parameter $\mu_m(n)$ is chosen as

$$\mu_m(n) = \frac{\tilde{\mu}}{\mathscr{E}_{m-1}(n)} \tag{G.6}$$

where

$$\begin{aligned}\mathscr{E}_{m-1}(n) &= \sum_{i=1}^{n} [|f_{m-1}(i)|^2 + |b_{m-1}(i-1)|^2] \\ &= \mathscr{E}_{m-1}(n-1) + |f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2]\end{aligned} \tag{G.7}$$

For a well-behaved convergence of the algorithm, we usually set $\tilde{\mu} < 0.1$. The parameter $\mathscr{E}_{m-1}(n)$ represents the total energy of both the forward and backward prediction errors at the input of the $m$th stage, measured up to and including time $n$.

In practice, a minor modification is made to the *energy estimator* of Eq. (G.7) by writing it in the form of a *single-pole average* of squared data, as shown by (Griffiths, 1977, 1978)

$$\mathscr{E}_{m-1}(n) = \beta\mathscr{E}_{m-1}(n-1) + (1-\beta)[|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2] \tag{G.8}$$

where $0 < \beta < 1$. The introduction of the parameter $\beta$ in Eq. (G.8) provides the GAL algorithm with a finite *memory*, which helps it deal better with statistical variations when operating in a nonstationary environment.

**TABLE G.1** SUMMARY OF THE GAL ALGORITHM

*Parameters*: $M$ = final prediction order
$\beta$ = constant, lying in the range $0 < \beta < 1$
$\tilde{\mu} < 0.1$

*Initialization*: For prediction order $m = 1, 2, \ldots, M$, put

$$f_m(0) = b_m(0) = 0$$

$$\mathscr{E}_{m-1}(0) = \delta, \qquad \delta = \text{small constant}$$

$$\hat{\kappa}_m(0) = 0$$

For time $n = 1, 2, \ldots$, put

$$f_0(n) = b_0(n) = u(n), \qquad u(n) = \text{lattice predictor input}$$

*Prediction*: For prediction order $m = 1, 2, \ldots, M$ and time $n = 1, 2, \ldots$, compute

$$f_m(n) = f_{m-1}(n) + \hat{\kappa}_m^*(n) b_{m-1}(n-1)$$

$$b_m(n) = b_{m-1}(n-1) + \hat{\kappa}_m(n) f_{m-1}(n)$$

$$\mathscr{E}_{m-1}(n) = \beta \mathscr{E}_{m-1}(n-1) + (1-\beta)(|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2)$$

$$\hat{\kappa}_m(n) = \hat{\kappa}_m(n-1) - \frac{\tilde{\mu}}{\mathscr{E}_{m-1}(n)} [f_{m-1}^*(n) b_m(n) + b_{m-1}(n-1) f_m^*(n)]$$

A summary of the GAL algorithm is presented in Table G.1.

## Properties of the GAL Algorithm

The use of time-varying step-size parameter $\mu_m(n) = \tilde{\mu}/\mathscr{E}_{m-1}(n)$ in the update equation for the reflection coefficient $\hat{\kappa}_m(n)$ introduces a form of *normalization* similar to that in the normalized LMS algorithm. From Eq. (G.8) we see that for small magnitudes of the prediction errors $f_{m-1}(n)$ and $b_{m-1}(n)$ the value of the parameter $\mathscr{E}_{m-1}(n)$ is correspondingly small or, equivalently, the step-size parameter $\mu_m(n)$ has a correspondingly large value. Such a behavior is desirable from a practical point of view. Basically, a small value for the prediction errors means that the adaptive lattice predictor is providing an accurate model of the external environment in which it is operating. Hence, if there is any increase in the prediction errors, it should be due to variations in the external environment, in which case it is highly desirable for the adaptive lattice predictor to respond rapidly to such variations. This objective is indeed realized by having the step-size parameter $\mu_m(n)$ assume a large value, which makes it possible for the GAL algorithm to provide an initially rapid convergence to the new environmental conditions. If, on the other hand, the input data applied to the adaptive lattice predictor are too noisy (i.e., they contain a strong white-noise component in addition to the signal of interest), we find that the prediction errors produced by the

adaptive lattice predictor are correspondingly large. In such a situation, the parameter $\mathscr{E}_{m-1}(n)$ has a large value or, equivalently, the step-size parameter $\mu_m(n)$ has a small value. Accordingly, the GAL algorithm does *not* respond rapidly to variations in the external environment, which is precisely the way we would like the algorithm to behave (Alexander, 1986a).

Another point of interest is that the convergence behavior of the GAL algorithm is somewhat more rapid than that of the LMS algorithm, but inferior to that of exact recursive LSL algorithms.

# APPENDIX H

# Solution of the Difference Equation (9.75)

In this appendix we fill in the mathematical details concerning the mean-squared error analysis of the LMS algorithm. We begin by reproducing Eq. (9.75):

$$\mathbf{x}(n+1) = \mathbf{B}\mathbf{x}(n) + \mu^2 J_{\min}\boldsymbol{\lambda} \tag{H.1}$$

where $\mathbf{B}$ is a real, positive, and symmetric matrix; $\boldsymbol{\lambda}$ is a vector of eigenvalues pertaining to an ensemble-averaged correlation matrix $\mathbf{R}$ of size $M$-by-$M$.

Equation (H.1) is a difference equation of order 1 in the vector $\mathbf{x}(n)$. Therefore, assuming an initial value $\mathbf{x}(0)$, the solution to this equation is[1]

$$\mathbf{x}(n) = \mathbf{B}^n\mathbf{x}(0) + \mu^2 J_{\min}\sum_{i=0}^{n-1}\mathbf{B}^i\boldsymbol{\lambda} \tag{H.2}$$

By analogy with the formula for the sum of a geometric series, we may express the finite sum $\sum_{i=0}^{n-1}\mathbf{B}^i$ as follows:

$$\sum_{i=0}^{n-1}\mathbf{B}^i = (\mathbf{I} - \mathbf{B}^n)(\mathbf{I} - \mathbf{B})^{-1} \tag{H.3}$$

where $\mathbf{I}$ is the identity matrix. Substituting Eq. (H.3) in (H.2), we thus get

$$\mathbf{x}(n) = \mathbf{B}^n[\mathbf{x}(0) - \mu^2 J_{\min}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda}] + \mu^2 J_{\min}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda} \tag{H.4}$$

[1] The approach we follow here is adapted from Mazo (1979). However, we differ from Mazo in that our analysis is for complex data, whereas that of Mazo is for real data.

The first term on the right-hand side of Eq. (H.4) is the *transient* component of the vector $\mathbf{x}(n)$, and the second term is the *steady-state* component. Since the matrix $\mathbf{B}$ is symmetric, we may apply to it an orthogonal similarity transformation. We may thus write

$$\mathbf{G}^T\mathbf{B}\mathbf{G} = \mathbf{C} \tag{H.5}$$

The matrix $\mathbf{C}$ is a diagonal matrix with elements $c_i = 1, 2, \ldots, M$, which are the eigenvalues of $\mathbf{B}$. The matrix $\mathbf{G}$ is an *orthonormal matrix* whose $i$th column is the eigenvector $\mathbf{g}_i$ of $\mathbf{B}$, associated with eigenvalue $c_i$. Because of the property

$$\mathbf{G}\mathbf{G}^T = \mathbf{I} \tag{H.6}$$

we find that

$$\mathbf{B}^n = \mathbf{G}\mathbf{C}^n\mathbf{G}^T \tag{H.7}$$

Hence, we may rewrite Eq. (H.4) in the form

$$\mathbf{x}(n) = \mathbf{G}\mathbf{C}^n\mathbf{G}^T[\mathbf{x}(0) - \mu^2 J_{\min}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda}] + \mu^2 J_{\min}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda} \tag{H.8}$$

Since $\mathbf{C}$ is a diagonal matrix, we have

$$\mathbf{C}^n = \text{diag}[c_1^n, c_2^n, \ldots, c_M^n] \tag{H.9}$$

It follows therefore that the solution defined by Eq. (H.8) is stable if and only if the eigenvalues of matrix $\mathbf{B}$ all have a magnitude less than 1. The eigenvalues of matrix $\mathbf{B}$ are all positive, since the matrix $\mathbf{B}$ is positive definite. For stability, we therefore require the condition

$$0 < c_i < 1 \qquad \text{for all } i \tag{H.10}$$

When this condition is satisfied, the transient component in Eq. (H.8) decays to zero as the number of iterations, $n$, approaches infinity. This would then leave the steady-state component as the only component. We may thus write

$$\mathbf{x}(\infty) = \mu^2 J_{\min}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda} \tag{H.11}$$

Substituting Eq. (H.11) in (H.8), we may rewrite the solution as

$$\mathbf{x}(n) = \mathbf{G}\mathbf{C}^n\mathbf{G}^T[\mathbf{x}(0) - \mathbf{x}(\infty)] + \mathbf{x}(\infty) \tag{H.12}$$

In view of the diagonal nature of matrix $\mathbf{C}^n$, and since the orthonormal matrix $\mathbf{G}$ consists of the eigenvectors of $\mathbf{B}$ as its columns, we may express the matrix product $\mathbf{G}\mathbf{C}^n\mathbf{G}^T$ as follows:

$$\mathbf{G}\mathbf{C}^n\mathbf{G}^T = \sum_{i=1}^{M} c_i^n \mathbf{g}_i\mathbf{g}_i^T \tag{H.13}$$

Accordingly, we may rewrite Eq. (H.12) one more time in the equivalent form

$$\mathbf{x}(n) = \sum_{i=1}^{M} c_i^n \mathbf{g}_i\mathbf{g}_i^T[\mathbf{x}(0) - \mathbf{x}(\infty)] + \mathbf{x}(\infty) \tag{H.14}$$

This is the desired solution to the difference equation (H.1).

# APPENDIX I

# Steady-State Analysis of the LMS Algorithm Without Invoking the Independence Assumption

In this Appendix, we revisit the steady-state analysis of the LMS algorithm by taking an iterative approach that avoids the independence assumption (Butterweck, 1995a). The theory applies to small values of the step-size parameter. It proceeds in two stages. First, a power series solution is derived for the weight-error vector in terms of the step-size parameter. The result so obtained is next used to derive a corresponding expansion for the weight-error correlation matrix.

## I.1 ITERATIVE SOLUTION FOR THE WEIGHT-ERROR VECTOR

The weight-error vector $\boldsymbol{\epsilon}(n)$ computed by the LMS algorithm is defined by the stochastic difference equation (9.55), reproduced here for convenience of presentation:

$$\boldsymbol{\epsilon}(n+1) = [\mathbf{I}-\mu\mathbf{u}(n)\mathbf{u}^H(n)]\boldsymbol{\epsilon}(n) + \mu\mathbf{u}(n)e_o^*(n) \tag{I.1}$$

where $\mathbf{u}(n)$ is the tap-input vector, $\mu$ is the step-size parameter, and $e_o(n)$ is the estimation error produced by the Wiener solution. Under the condition that $\mu$ is small, the direct-averaging method leads us to say that the solution of this equation is approximately the same as that of Eq. (9.56), reproduced here in the form

$$\boldsymbol{\epsilon}_0(n+1) = (\mathbf{I}-\mu\mathbf{R})\boldsymbol{\epsilon}_0(n) + \mu\mathbf{u}(n)e_o^*(n) \tag{I.2}$$

where $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$. For reasons that will become apparent presently, we have used a different symbol for the weight-error vector in Eq. (I.2). Note that the solutions of Eqs. (I.1) and (I.2) become equal for the limiting case of a vanishing step-size parameter $\mu$.

In the iterative procedure described by Butterweck (1995a), the solution of Eq. (I.2) is used as a starting point for generating a whole set of solutions of the original stochastic difference equation (I.1). The accuracy of the solution so obtained improves with increasing iteration order. Thus, starting with the solution $\boldsymbol{\epsilon}_0(n)$, the solution of Eq. (I.1) is expressed as a sum of partial functions, as shown by

$$\boldsymbol{\epsilon}(n) = \boldsymbol{\epsilon}_0(n) + \boldsymbol{\epsilon}_1(n) + \boldsymbol{\epsilon}_2(n) + \cdots \tag{I.3}$$

Define the zero-mean difference matrix:

$$\mathbf{P}(n) = \mathbf{u}(n)\mathbf{u}^H(n) - \mathbf{R} \tag{I.4}$$

Then, substituting Eq. (I.4) in (I.1) yields

$$\begin{aligned}\boldsymbol{\epsilon}_0(n+1) + \boldsymbol{\epsilon}_1(n+1) + \boldsymbol{\epsilon}_2(n+1) + \cdots \\ = (\mathbf{I} - \mu\mathbf{R})[\boldsymbol{\epsilon}_0(n) + \boldsymbol{\epsilon}_1(n) + \boldsymbol{\epsilon}_2(n) + \cdots \\ - \mu\mathbf{P}(n)[\boldsymbol{\epsilon}_1(n) + \boldsymbol{\epsilon}_2(n) + \cdots] + \mu\mathbf{u}(n)e_o^*(n)\end{aligned}$$

from which we readily deduce that

$$\boldsymbol{\epsilon}_i(n+1) = (\mathbf{I} - \mu\mathbf{R})\boldsymbol{\epsilon}_i(n) + \mathbf{f}_i(n), \qquad i = 0, 1, 2, \cdots \tag{I.5}$$

where the subscript $i$ refers to the iteration order. The "driving force" $\mathbf{f}_i(n)$ for the difference equation (I.5) is defined by

$$\mathbf{f}_i(n) = \begin{cases} \mu\mathbf{u}(n)e_o^*(n), & i = 0 \\ -\mu\mathbf{P}(n)\boldsymbol{\epsilon}_{i-1}(n), & i = 1, 2, \cdots \end{cases} \tag{I.6}$$

Thus, a time-varying system characterized by the stochastic difference equation (I.1) is transformed into a set of equations having the same basic format as described in (I.5), such that the solution to the $i$th equation in the set (i.e., step $i$ in the iterative procedure) follows from the $(i-1)$th equation. In particular, the problem is reduced to a study of the transmission of a stationary stochastic process through a low-pass filter with an extremely low cutoff frequency.

## I.2 SERIES EXPANSION OF THE WEIGHT-ERROR CORRELATION MATRIX

On the basis of Eq. (I.3), we may express the weight-error correlation matrix in the form of a corresponding series as follows:

$$\begin{aligned}\mathbf{K}(n) &= E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] \\ &= \sum_i \sum_k E[\boldsymbol{\epsilon}_i(n)\boldsymbol{\epsilon}_k^H(n)], \qquad (i,k) = 0, 1, 2, \cdots\end{aligned} \tag{I.7}$$

Expanding this series in light of the definitions given in Eqs. (I.5) and (I.6), and then grouping equal-order terms in the step-size parameter $\mu$, we get the following series expansion:

$$\mathbf{K}(n) = \mathbf{K}_0(n) + \mu\mathbf{K}_1(n) + \mu^2\mathbf{K}_2(n) + \cdots \tag{I.8}$$

where the various matrix coefficients are themselves defined as follows:

$$\mathbf{K}_j(n) = \begin{cases} E[\boldsymbol{\epsilon}_0(n)\boldsymbol{\epsilon}_0^H(n)] & \text{for } j = 0 \\ \displaystyle\sum_i \sum_k E[\boldsymbol{\epsilon}_i(n)\boldsymbol{\epsilon}_k^H(n)] & \text{for all } (i,k) \geq 0 \\ & \text{such that } i + k = 2j-1, 2j \end{cases} \tag{I.9}$$

These matrix coefficients are defined, albeit in a rather complex fashion, by the spectral and probability distribution of the environment in which the LMS algorithm operates. In a general setting with arbitrarily colored signals, the calculation of $\mathbf{K}_j(n)$ for $j \geq 1$ can be rather tedious, except in some special cases (Butterweck, 1995a).

The zero-order term $\mathbf{K}_0(n)$ in Eq. (I.8) is of special interest for two reasons. First, for a small $\mu$ it may be used as an approximation to the actual $\mathbf{K}(n)$, as discussed in Section 9.4. Second, it lends itself to examination without any statistical assumptions concerning the environment in which the LMS algorithm operates. In particular, we find that under steady-state conditions (i.e., large $n$), $\mathbf{K}_0(n)$ is determined as the solution to the equation (Butterweck, 1995b):

$$\mathbf{R}\mathbf{K}_0(n) + \mathbf{K}_0(n)\mathbf{R} = \mu \sum_l J_{\min}^{(l)} \mathbf{R}^{(l)}, \qquad \text{large } n \tag{I.10}$$

where

$$J_{\min}^{(l)} = E[e_o(n)\, e_o^*(n-l)], \qquad l = 0, 1, 2, \ldots \tag{I.11}$$

$$\mathbf{R}^{(l)} = E[\mathbf{u}(n)\mathbf{u}^H(n-l)], \qquad l = 0, 1, 2, \ldots \tag{I.12}$$

Note that for $l = 0$, we have $J_{\min}^{(0)} = J_{\min}$ and $\mathbf{R}^{(0)} = \mathbf{R}$.

The steady-state value of the misadjustment $\mathcal{M}$ derived in Chapter 9 under the independence assumption corresponds to setting $l = 0$ in Eq. (I.10) and ignoring all higher-order terms. This special case corresponds to the assumption that the estimation error $e(n)$ produced by the LMS algorithm is drawn from a white noise process. Thus, Eq. (I.10) is approximated by

$$\mathbf{R}\mathbf{K}_0(n) + \mathbf{K}_0(n)\mathbf{R} \simeq \mu J_{\min}\mathbf{R}, \qquad \text{large } n$$

from which we readily find that the misadjustment is

$$\begin{aligned} \mathcal{M} &= \frac{\text{tr}[\mathbf{R}\mathbf{K}_0(n)]}{J_{\min}} \\ &\simeq \frac{\mu}{2}\,\text{tr}[\mathbf{R}] \\ &= \frac{\mu}{2}\sum_{i=1}^{M} \lambda_i \end{aligned}$$

This is indeed the result derived in Eq. (9.95).

# APPENDIX J

# The Complex Wishart Distribution

The Wishart distribution plays an important role in statistical signal processing. In this appendix we present a summary of some important properties of the Wishart distribution for complex-valued data. In particular, we derive a result that is pivotal to a rigorous analysis of the convergence behavior of the standard RLS algorithm, presented in Chapter 13. We begin the discussion with a definition of the complex Wishart distribution.

## J.1 DEFINITION

Consider an $M$-by-$M$ time-averaged (sample) correlation matrix $\mathbf{\Phi}(n)$, defined by

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \mathbf{u}(i)\mathbf{u}^H(i) \tag{J.1}$$

where

$$\mathbf{u}(i) = [u_1(i), u_2(i), \ldots, u_M(i)]^T$$

In what follows, we assume that $\mathbf{u}(1), \mathbf{u}(2), \ldots, \mathbf{u}(n)$ $(n > M)$ are *independently and identically distributed.* We may then formally define the *complex Wishart distribution* as follows (Muirhead, 1982):

If $\{u_1(i), u_2(i), \ldots, u_M(i) \mid i = 1, 2, \ldots, n\}$, $n \geq M$, is a sample from the $M$-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{R})$, and if $\mathbf{\Phi}(n)$ is the time-averaged correlation matrix defined in Eq. (J.1), then the elements of $\mathbf{\Phi}(n)$ have the complex Wishart distribution $\mathcal{W}_M(n,\mathbf{R})$, which is characterized by the parameters $M$, $n$, and $\mathbf{R}$.

In specific terms, we may say that if matrix $\mathbf{\Phi}$ is $\mathcal{W}_M(n,\mathbf{R})$, then the probability density function of $\mathbf{\Phi}$ is

$$f(\mathbf{\Phi}) = \frac{1}{2^{Mn/2}\Gamma_M\left(\frac{1}{2}n\right)(\det(\mathbf{R}))^{n/2}} \operatorname{etr}\left(-\frac{1}{2}\mathbf{R}^{-1}\mathbf{\Phi}\right)(\det(\mathbf{\Phi}))^{(n-M-1)/2} \tag{J.2}$$

where $\det(\bullet)$ denotes the determinant of the enclosed matrix, $\operatorname{etr}(\bullet)$ denotes the exponential raised to the trace of the enclosed matrix, and $\Gamma_M(a)$ is the *multivariate gamma function* defined by

$$\Gamma_M(a) = \int_{\mathbf{A}} \operatorname{etr}(-\mathbf{A})(\det(\mathbf{A}))^{a-(M+1)/2} d\mathbf{A} \tag{J.3}$$

where $\mathbf{A}$ is a positive definite matrix.

## J.2 THE CHI-SQUARE DISTRIBUTION AS A SPECIAL CASE

For the special case of a univariate distribution, that is, $M = 1$, Eq. (J.1) reduces to the scalar form:

$$\varphi(n) = \sum_{i=1}^{n} |u(i)|^2 \tag{J.4}$$

Correspondingly, the correlation matrix $\mathbf{R}$ reduces to the variance $\sigma^2$. Let

$$\chi^2(n) = \frac{\varphi(n)}{\sigma^2} \tag{J.5}$$

Then, using Eq. (J.2) we may define the normalized probability density function of the normalized random variable $\chi^2(n)$ as

$$f(\chi^2) = \frac{\left(\frac{\chi^2}{2}\right)^{n/2-1} e^{-\chi^2/2}}{2^{n/2}\Gamma\left(\frac{1}{2}n\right)} \tag{J.6}$$

where $\Gamma(1/2n)$ is the (scalar) *gamma function*.[1] The variable $\chi^2(n)$, defined above, is said to have a *chi-square distribution with n degrees of freedom*. We may thus view the complex Wishart distribution as a generalization of the univariate chi-square distribution.

A useful property of a chi-square distribution with $n$ degrees of freedom is the fact that it is *reproductive with respect to* $1/2n$ (Wilks, 1962). That is, the $r$th moment of $\chi^2(n)$ is

$$E[\chi^{2r}(n)] = \frac{2^r \Gamma\left(\frac{n}{2} + r\right)}{\Gamma\left(\frac{n}{2}\right)} \tag{J.7}$$

Thus, the mean, mean-square, and variance of $\chi^2(n)$ are as follows, respectively:

$$E[\chi^2(n)] = n \tag{J.8}$$

$$E[\chi^4(n)] = n(n+2) \tag{J.9}$$

$$\text{var}[\chi^2(n)] = n(n+2) - n^2 = 2n \tag{J.10}$$

Moreover, from Eq. (J.7) we find that the mean of the reciprocal of $\chi^2(n)$ is

$$\begin{aligned} E\left[\frac{1}{\chi^2(n)}\right] &= \frac{1}{2} \frac{\Gamma\left(\frac{n}{2} - 1\right)}{\Gamma\left(\frac{n}{2}\right)} \\ &= \frac{1}{2} \frac{\Gamma\left(\frac{n}{2} - 1\right)}{\left(\frac{n}{2} - 1\right)\Gamma\left(\frac{n}{2} - 1\right)} = \frac{1}{n-2} \end{aligned} \tag{J.11}$$

[1]For the general case of a complex number $g$ whose real part is positive, the *gamma function* $\Gamma(g)$ is defined by the definite integral (Wilks, 1962)

$$\Gamma(g) = \int_0^\infty x^{g-1} e^{-x}\, dx$$

Integrating it by parts, we readily find that

$$\Gamma(g) = (g-1)\Gamma(g-1)$$

For the case when $g$ is a positive integer, we may thus express the gamma function $\Gamma(g)$ as the factorial

$$\Gamma(g) = (g-1)!$$

When $g > 0$, but not an integer, we have

$$\Gamma(g) = (g-1)\,\Gamma(\delta)$$

where $0 < \delta < 1$. For the particular case of $\delta = 1/2$, we have $\Gamma(\delta) = \sqrt{\pi}$.

## J.3 PROPERTIES OF THE COMPLEX WISHART DISTRIBUTION

Returning to the main theme of this appendix, the complex Wishart distribution has some important properties of its own, which are summarized as follows (Muirhead, 1982; Anderson, 1984):

1. If $\Phi$ is $\mathcal{W}_M(n, \mathbf{R})$ and $\mathbf{a}$ is any $M$-by-1 random vector distributed independently of $\Phi$ with $P(\mathbf{a} = \mathbf{0}) = 0$ (i.e., the probability that $\mathbf{a} = \mathbf{0}$ is zero), then $\mathbf{a}^H\Phi\mathbf{a}/\mathbf{a}^H\mathbf{R}\mathbf{a}$ is chi-square distributed with $n$ degrees of freedom, and is independent of $\mathbf{a}$.
2. If $\Phi$ is $\mathcal{W}_M(n,\mathbf{R})$ and $\mathbf{Q}$ is a matrix of dimensions $M$-by-$k$ and rank $k$, then $\mathbf{Q}^H\Phi\mathbf{Q}$ is $\mathcal{W}_k(n,\mathbf{Q}^H\mathbf{R}\mathbf{Q})$.
3. If $\Phi$ is $\mathcal{W}_M(n,\mathbf{R})$ and $\mathbf{Q}$ is a matrix of dimensions $M$-by-$k$ and rank $k$, then $(\mathbf{Q}^H\Phi^{-1}\mathbf{Q})^{-1}$ is $\mathcal{W}_k(n - M + k,(\mathbf{Q}^H\mathbf{R}^{-1}\mathbf{Q})^{-1})$.
4. If $\Phi$ is $\mathcal{W}_M(n,\mathbf{R})$ and $\mathbf{a}$ is any $M$-by-1 random vector distributed independently of $\Phi$ with $P(\mathbf{a} = \mathbf{0}) = 0$, then $\mathbf{a}^H\mathbf{R}^{-1}\mathbf{a}/\mathbf{a}^H\Phi^{-1}\mathbf{a}$ is chi-square distributed with $n - M + 1$ degrees of freedom.
5. Let $\Phi$ and $\mathbf{R}$ be partitioned into $p$ and $M - p$ rows and columns, as shown by

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

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

If $\Phi$ is distributed according to $\mathcal{W}_M(n,\mathbf{R})$, then $\Phi_{11}$ is distributed according to $\mathcal{W}_p(n,\mathbf{R}_{11})$.

## J.4 EXPECTATION OF THE INVERSE CORRELATION MATRIX $\Phi^{-1}(n)$

Property 4 of the complex Wishart distribution may be used to find the expectation of the inverse correlation matrix $\Phi^{-1}(n)$, which is associated with the convergence of the RLS algorithm in the mean square. Specifically, for any fixed and nonzero $\alpha$ in $\mathbb{R}^M$, we know from Property 4 described above that $\alpha^H\mathbf{R}^{-1}\alpha/\alpha^H\Phi^{-1}\alpha$ is chi-square distributed with $n - M + 1$ degrees of freedom. Let $\chi^2(n - M + 1)$ denote this ratio. Then, using the result described in Eq. (J.11), we may write

$$E[\alpha^H\Phi^{-1}(n)\alpha] = \alpha^H\mathbf{R}^{-1}\alpha E\left[\frac{1}{\chi^2(n - M + 1)}\right]$$

$$= \frac{1}{n - M - 1}\alpha^H\mathbf{R}^{-1}\alpha, \qquad n > M + 1$$

which, in turn, implies that

$$E[\Phi^{-1}(n)] = \frac{1}{n - M - 1}\mathbf{R}^{-1}, \qquad n > M + 1 \tag{J.12}$$

# Glossary

## TEXT CONVENTIONS

1. Boldfaced lowercase letters are used to denote column vectors. Boldfaced uppercase letters are used to denote matrices.
2. The estimate of a scalar, vector, or matrix is designated by the use of a hat (^) placed over the pertinent symbol.
3. The symbol $|\ |$ denotes the magnitude or absolute value of a complex scalar enclosed within. The symbol ang[ ] or arg[ ] denotes the phase angle of the scalar enclosed within.
4. The symbol $\|\ \|$ denotes the Euclidean norm of the vector or matrix enclosed within.
5. The symbol det( ) denotes the determinant of the square matrix enclosed within.
6. The open interval $(a, b)$ of the variable $x$ signifies that $a < x < b$. The closed interval $[a, b]$ signifies that $a \leq x \leq b$, and $(a, b]$ signifies that $a < x \leq b$.
7. The inverse of nonsingular (square) matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{-1}$.
8. The pseudoinverse of matrix A (not necessarily square) is denoted by $\mathbf{A}^{+}$.
9. Complex conjugation of a scalar, vector, or matrix is denoted by the use of an asterisk as superscript. Transposition of a vector or matrix is denoted by superscript $T$. Hermitian transposition (i.e., complex conjugation and transposition

combined) of a vector or matrix is denoted by superscript $H$. Backward rearrangement of the elements of a vector is denoted by superscript $B$.

10. The symbol $\mathbf{A}^{-H}$ denotes the Hermitian transpose of the inverse of a nonsingular (square) matrix $\mathbf{A}$.
11. The square root of a square matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{1/2}$.
12. The symbol $\text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]$ denotes a diagonal matrix whose elements on the main diagonal equal $\lambda_1, \lambda_2, \ldots, \lambda_M$.
13. The order of linear predictor or the order of autoregressive model is signified by a subscript added to the pertinent scalar or vector parameter.
14. The statistical expectation operator is denoted by $E[\cdot]$, where the quantity enclosed is the random variable or random vector of interest. The variance of a random variable is denoted by $\text{var}[\cdot]$, where the quantity enclosed is the random variable.
15. The conditional probability density function of random variable $U$, given that hypothesis $H_i$ is true, is denoted by $f_U(u|H_i)$, where $u$ is the sample value of random variable $U$.
16. The inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as $\mathbf{x}^H\mathbf{y} = \mathbf{y}^T\mathbf{x}^*$. Another possible inner product is $\mathbf{y}^H\mathbf{x} = \mathbf{x}^T\mathbf{y}^*$. These two inner products are the complex conjugate of each other. The outer product of the vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as $\mathbf{x}\mathbf{y}^H$. The inner product is a scalar, whereas the outer product is a matrix.
17. The trace of a square matrix $\mathbf{R}$ is denoted by $\text{tr}[\mathbf{R}]$; it is defined as the sum of the diagonal elements of $\mathbf{R}$. The exponential raised to the trace of matrix $\mathbf{R}$ is denoted by $\text{etr}[\mathbf{R}]$.
18. The autocorrelation function of stationary discrete-time stochastic process $u(n)$ is defined by

$$r(k) = E[u(n)u^*(n-k)]$$

The cross-correlation function between two jointly stationary discrete-time stochastic process $u(n)$ and $d(n)$ is defined by

$$p(-k) = E[u(n-k)d^*(n)]$$

19. The ensemble-averaged correlation matrix of a random vector $\mathbf{u}(n)$ is defined by

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

20. The ensemble-averaged cross-correlation vector between a random vector $\mathbf{u}(n)$ and a random variable $d(n)$ is defined by

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)]$$

21. The time-averaged (sample) correlation matrix of a vector $\mathbf{u}(i)$ over the observation interval $1 \le i \le n$ is defined by

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \mathbf{u}(i)\mathbf{u}^H(i)$$

The exponentially weighted version of $\mathbf{\Phi}(n)$ is

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \lambda^{n-i}\, \mathbf{u}(i)\mathbf{u}^H(i)$$

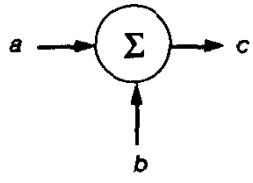
22. The time-averaged cross-correlation vector between a vector $\mathbf{u}(i)$ and a scalar $d(i)$ over the observation interval $1 \leq i \leq n$ is defined by

$$\mathbf{z}(n) = \sum_{i=1}^{n} \mathbf{u}(i)d^*(i)$$

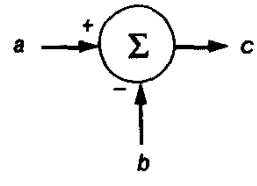
Its exponentially weighted version is

$$\mathbf{z}(n) = \sum_{i=1}^{n} \lambda^{n-i}\, \mathbf{u}(i)d^*(i)$$

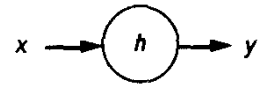
23. The discrete-time Fourier transform of a time function $u(n)$ is denoted by $F[u(n)]$. The inverse discrete-time Fourier transform of a frequency function $U(\omega)$ is denoted by $F^{-1}[U(\omega)]$.
24. In constructing block diagrams (signal-flow graphs) involving scalar quantities, the following symbols are used. The symbol



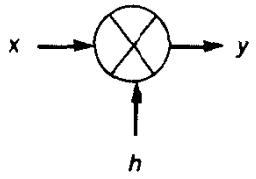

denotes an adder with $c = a + b$. The same symbol with algebraic signs added as in the following



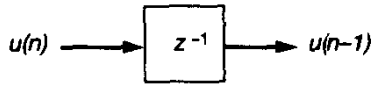

denotes a substractor with $c = a - b$. The symbol



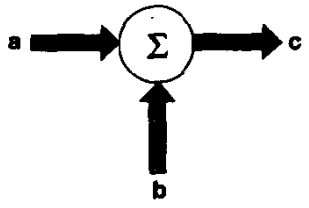
denotes a multiplier with $y = hx$. This multiplication is also represented as



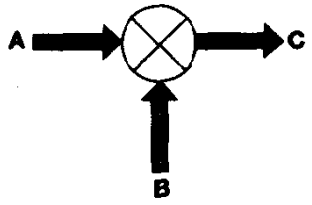

The unit-sample (delay) operator is denoted by



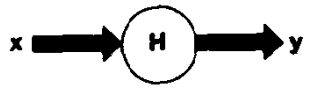

25. In constructing block diagrams (signal-flow graphs) involving matrix quantities, the following symbols are used. The symbol



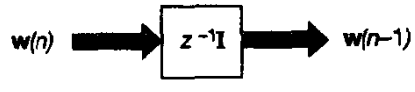

denotes summation with $\mathbf{c} = \mathbf{a} + \mathbf{b}$. The symbol




denotes multiplication with $\mathbf{C} = \mathbf{AB}$. The symbol




denotes a branch having transmittance $\mathbf{H}$, with $\mathbf{y} = \mathbf{Hx}$. The unit-sample operator is denoted by the symbol



## ABBREVIATIONS

| | |
|---|---|
| ADPCM | Adaptive differential pulse-code modulation |
| AGC | Automatic gain control |
| AIC | An information-theoretic criterion |
| ALE | Adaptive line enhancer |
| AR | Autoregressive |
| ARMA | Autoregressive-moving average |
| as | Almost surely |
| BIBO | Bounded input-bounded output |
| b/s | Bits per second |
| BLP | Backward linear prediction |
| BP | Back-propagation |
| BPSK | Binary phase-shift keying |
| CMA | Constant modulus adaptive |
| DCT | Discrete cosine transform |
| DFT | Discrete Fourier transform |
| DPCM | Differential pulse-code modulation |
| DTE | Data terminal equipment |
| EKF | Extended Kalman filter |
| FBLP | Forward-backward linear prediction |
| FDAF | Frequency-domain adaptive filter |
| FFT | Fast Fourier transform |
| FIR | Finite-duration impulse response |
| FLP | Forward linear prediction |
| FTF | Fast transversal filters ((algorithm) |
| GAL | Gradient adaptive lattice |
| GSLC | Generalized sidelobe canceler |
| HF | High frequency |
| HOS | Higher-order statistics |
| Hz | Hertz |
| IFFT | Inverse fast Fourier transform |
| iid | Independent and identically distributed |
| IIR | Infinite-duration impulse response |
| INR | Interference-to-noise ratio |
| kb/s | Kilobits per second |
| kHz | Kilohertz |
| LCMV | Linearly constrained minimum variance |

| | |
|---|---|
| LMS | Least-mean-square |
| LBS | Least significant bit |
| LSL | Least-squares lattice |
| LPC | Linear predictive coding |
| MA | Moving average |
| MDL | Minimum description length (criterion) |
| MEM | Maximum entropy method |
| MN | Minimum norm |
| MLP | Multilayer perceptron |
| MSE | Mean-squared error |
| MVDR | Minimum-variance distortionless response |
| PAM | Pulse amplitude modulation |
| PARCOR | Partial correlation |
| PCM | Pulse-code modulation |
| PN | Pseudonoise |
| QAM | Quadrature amplitude modulation |
| QPSK | Quadrature phase-shift keying |
| QR-RLS | QR-decomposition–based recursive least squares |
| QRD-LSL | QR-decomposition–based least-squares lattice |
| RBF | Radial basis function |
| RLS | Recursive least squares |
| rms | root-mean-square |
| ROC | Region of convergence (z-transform) |
| ROC | Rate of convergence (adaptive filter) |
| s | Second |
| SIMO | Single input–multiple output |
| SNR | Signal-to-noise ratio |
| SOBAF | Self-orthogonalizing block adaptive filter |
| SRF | Square-root filtering |
| SVD | Singular value decomposition |
| TDNN | Time-delay neural network |
| wp1 | With probability one |

## PRINCIPAL SYMBOLS

| | |
|---|---|
| $a_{M,k}(n)$ | $k$th tap weight of forward prediction-error filter of order $M$ (at iteration $n$), with $k = 0, 1, \ldots, M$; note that $a_{m,0}(n) = 1$ |

| | |
|---|---|
| $\mathbf{a}_M(n)$ | Tap-weight vector of forward prediction-error filter of order $M$ (at iteration $n$) |
| $\mathbf{A}$ | Data matrix in the covariance method |
| $\mathbf{A}(n)$ | Data matrix in the pre-windowing method |
| $b_M(n)$ | Backward (*a posteriori*) prediction error produced at iteration $n$ by prediction-error filter of order $M$ |
| $\mathbf{b}(n)$ | Backward (*a posteriori*) prediction-error vector representing sequence of errors produced by backward prediction-error filters of orders $0, 1, \ldots, M$ |
| $\mathcal{B}_M(n)$ | Sum of weighted backward prediction error squares produced by backward prediction-error filter of order $M$ |
| $c_{M,k}(n)$ | $k$th tap weight of backward prediction-error filter of order $M$ (at iteration $n$), with $k = 0, 1, \ldots, M$; note that $c_{M,M}(n) = 1$ |
| $\mathbf{c}_M(n)$ | Tap-weight vector of backward prediction-error filter of order $M$ (at iteration $n$) |
| $\mathbf{c}(n)$ | Weight-error vector in steepest-descent algorithm |
| $c_k(\tau_1, \tau_2, \ldots, \tau_k)$ | $k$th-order cumulant |
| $C_k(\omega_1, \omega_2, \ldots, \omega_k)$ | $k$th-order polyspectrum |
| $\mathscr{C}$ | Contour |
| $\mathscr{C}^M$ | Complex $M$-dimensional parameter space |
| $\mathscr{C}(n)$ | Convergence ratio |
| det() | Determinant of the enclosed matrix |
| diag() | Diagonal matrix |
| $d(n)$ | Desired response |
| $\mathbf{d}$ | Desired response vector in the covariance method |
| $\mathbf{d}(n)$ | Desired response vector in the pre-windowing method |
| $D$ | Unit-delay operator (same as $z^{-1}$) |
| $\mathbf{D}_{m+1}(n)$ | Correlation matrix of backward prediction errors |
| $\mathscr{D}$ | Mean-square deviation |
| dec() | Function describing the decision performed by a threshold device |
| $e(n)$ | *A posteriori* estimation error |
| $e_m(n)$ | *A posteriori* estimation error at the output of stage $m$ in the joint-process estimator using the recursive LSL algorithm or QRD–LSL algorithm |
| $e$ | Base of natural logarithm |
| etr() | Exponential raised to the trace of the enclosed matrix |
| exp | Exponential |

| | |
|---|---|
| $E$ | Expectation operator |
| $\mathscr{E}(\mathbf{w}, n)$ | Cost function defined as the sum of weighted error squares expressed as a function of iteration $n$ |
| $\mathscr{E}(\mathbf{w})$ | Cost function defined as the sum of error squares, expressed as a function of the tap-weight vector $\mathbf{w}$ |
| $\mathscr{E}_{\min}$ | Minimum value of $\mathscr{E}(\mathbf{w})$ |
| $\mathscr{E}(n)$ | Cost function defined as the sum of weighted error squares, expressed as a function of iteration $n$ |
| $f_M(n)$ | Forward (*a posteriori*) prediction error produced at iteration $n$ by forward prediction-error filter of order $M$ |
| $\mathbf{f}(n)$ | Forward (*a posteriori*) prediction error vector representing sequence of errors produced by forward prediction-error filters of orders $0, 1, \ldots, M$ |
| $f_U(n)$ | Probability density function of random variable $U$, whose sample value equals $u$ |
| $f_{\mathbf{U}}(\mathbf{u})$ | Joint probability density function of the elements of random vector $\mathbf{U}$, whose sample value equals $\mathbf{u}$ |
| $F_M(z)$ | $z$-transform of sequence of forward prediction errors produced by forward prediction-error filter of order $M$ |
| $\mathbf{F}(n+1,n)$ | Transition matrix |
| $\mathscr{F}_M(n)$ | Weighted sum of forward prediction-error squares produced by forward prediction-error filter of order $M$ |
| $F[\,]$ | Fourier transform operator |
| $F^{-1}[\,]$ | Inverse Fourier transform operator |
| $g(\cdot)$ | Nonlinear function used in blind equalization |
| $\mathbf{G}(n)$ | Kalman gain |
| $h_k$ | $k$th regression coefficient of joint-process estimation based on lattice predictor for stationary impulse |
| $h_n$ | Minimum-phase polynomial used in blind equalization |
| $H_i$ | $i$th hypothesis |
| $H(z)$ | Transfer function of discrete-time linear filter |
| $I$ | Subscript for signifying the in-phase (real) component of a complex baseband signal |
| $\mathbf{I}$ | Identity matrix |
| $\mathbf{I}$ | Inverse of Fisher's information matrix $\mathbf{J}$ |
| $j$ | Square root of $-1$ |
| $J(\mathbf{w})$ | Cost function used to formulate the Wiener filtering problem, expressed as a function of the tap-weight vector $\mathbf{w}$ |

| | |
|---|---|
| $\mathbf{J}$ | Fisher's information matrix |
| $\mathbf{k}(n)$ | Gain vector in the RLS algorithm |
| $K$ | Final order of moving average model |
| $\mathbf{K}(n)$ | Correlation matrix of weight-error vector $\boldsymbol{\epsilon}(n)$ |
| ln | Natural logarithm |
| $\mathbf{L}(n)$ | Transformation matrix in the form of lower triangular matrix |
| $m$ | Variable order of linear predictor or autoregressive model |
| $M$ | Final order of linear predictor or autoregressive model |
| $M, K$ | Final order of autoregressive-moving average model |
| $\mathcal{M}$ | Misadjustment |
| $n$ | Discrete-time or number of iterations applied to recursive algorithm |
| $N$ | Data length |
| $\mathcal{N}$ | Symbol signifying the Gaussian (normal) distribution |
| $O(z)$ | Maximum phase polynomial |
| $O(M^k)$ | Order of $M^k$ |
| $p(-k)$ | Element of cross-correlation vector $\mathbf{p}$ for lag $k$ |
| $\mathbf{p}$ | Cross-correlation vector between tap-input vector $\mathbf{u}(n)$ and desired response $d(n)$ |
| $P_M$ | Average value of (forward or backward) prediction-error power for prediction order $M$ for stationary inputs |
| $\mathbf{P}(n)$ | Matrix equal to the inverse of the time-averaged correlation matrix $\boldsymbol{\Phi}(n)$ used in formulating the RLS algorithm |
| $q_{ki}$ | $i$th element of $k$th eigenvector |
| $\mathbf{q}_k$ | $k$th eigenvector |
| $Q$ | Subscript for signifying the quadrature (imaginary) component of a complex baseband signal |
| $\mathbf{Q}$ | Unitary matrix that consists of normalized eigenvectors in the set $\{\mathbf{q}_k\}$ used as columns |
| $Q(y)$ | Probability distribution function of standardized Gaussian random variable |
| $r(k)$ | Element of (ensemble-averaged) correlation matrix $\mathbf{R}$ for lag $k$ |
| $\mathbf{R}$ | Ensembled-average correlation matrix of stationary discrete-time process $u(n)$ |
| $\mathbb{R}^M$ | Real $M$-dimensional parameter space |

| | |
|---|---|
| $\mathbf{s}$ | Signal vector; steering vector |
| sgn() | Signum function |
| $S(\omega)$ | Power spectral density |
| $S_{AR}(\omega)$ | Autoregressive spectrum |
| $S_{MEM}(\omega)$ | MEM (maximum entropy method) spectrum |
| $S_{MVDR}(\omega)$ | Minimum variance distortionless response spectrum |
| $\mathcal{S}$ | System |
| $\mathcal{S}_d$ | Decreasingly excited subspace |
| $\mathcal{S}_o$ | Otherwise excited subspace |
| $\mathcal{S}_p$ | Persistently excited subspace |
| $\mathcal{S}_u$ | Unexcited subspace |
| $t$ | Time |
| $\mathbf{t}$ | Vector arising in joint-process estimation for nonstationary inputs |
| $\mathbf{t}_k$ | Vector defining the center of the $k$th kernel in RBF network |
| $u(n)$ | Sample value of tap input in transversal filter at time $n$ |
| $\mathbf{u}(n)$ | Tap-input vector consisting of $u(n)$, $u(n-1)$, . . . , as elements |
| $u_I(n)$ | In-phase component of $u(n)$ |
| $u_Q(n)$ | Quadrature component of $u(n)$ |
| $\mathbf{u}_k$ | $k$th left-singular vector of data matrix $\mathbf{A}$ |
| $\mathbf{U}$ | Matrix of left-singular vectors of data matrix $\mathbf{A}$ |
| $\mathcal{U}_n$ | Space spanned by tap inputs $u(n)$, $u(n-1)$, . . . |
| $\mathcal{U}(n)$ | Sum of weighted squared values of tap inputs $u(i)$, $i = 1, 2, \ldots, n$ |
| $v(n)$ | Sample value of white-noise process of zero mean |
| $\mathbf{v}_1(n)$ | Process noise vector |
| $\mathbf{v}_2(n)$ | Measurement noise vector |
| $\mathbf{v}(n)$ | Process noise vector in random-walk state model |
| $\mathbf{v}_k(n)$ | $k$th right-singular vector of data matrix $\mathbf{A}$ |
| $\mathbf{V}$ | Matrix of right singular vectors of data matrix $\mathbf{A}$ |
| $w_k(n)$ | $k$th tap weight of transversal filter at time $n$ |
| $w_{b,m,k}(n)$ | $k$th tap weight of backward predictor of order $m$ at iteration $n$ |
| $w_{f,m,k}(n)$ | $k$th tap weight of forward predictor of order $m$ at iteration $n$ |
| $\mathbf{w}(n)$ | Tap-weight vector of transversal filter at time $n$ |

| | |
|---|---|
| $\mathbf{w}_{b,m}(n)$ | Tap-weight vector of backward predictor of order $m$ at iteration $n$ |
| $\mathbf{w}_{f,m}(n)$ | Tap-weight vector of forward predictor of order $m$ at iteration $n$ |
| $\mathcal{W}$ | Symbol signifying the Wishart distribution |
| $\mathbf{x}(n)$ | State vector |
| $\mathbf{y}(n)$ | Observation vector used in formulating Kalman filter theory |
| $\mathcal{Y}_n$ | Vector space spanned by $y(n), y(n-1), \ldots$ |
| $z^{-1}$ | Unit-sample (delay) operator used in defining the $z$-transform of a sequence |
| $\mathbf{z}$ | Time-averaged cross-correlation vector between tap-input vector $\mathbf{u}(i)$ and desired response $d(i)$ |
| $Z(y)$ | Standardized Gaussian probability density function |
| $\alpha(n)$ | Innovation at time $n$ |
| $\boldsymbol{\alpha}(n)$ | Innovations vector |
| $\beta$ | Constant used in the DCT–LMS algorithm |
| $\beta$ | Constant used in the GAL algorithm |
| $\beta_m(n)$ | Backward prediction error of order $m$ |
| $\gamma(n)$ | Conversion factor used in the FTF algorithm, recursive LSL algorithm, and recursive QRD–LSL algorithm |
| $\Gamma(g)$ | Gamma function of $g$ |
| $\gamma_3$ | Skewness of a random variable |
| $\gamma_4$ | Kurtosis of a random variable |
| $\delta$ | Constant used in the initialization of the RLS family of algorithms |
| $\boldsymbol{\delta}$ | First coordinator vector |
| $\delta_l$ | Kronecker delta, equal to 1 for $l = 0$ and zero for $l \neq 0$ |
| $\Delta_m$ | Cross-correlation between forward prediction error $f_m(n)$ and delayed backward prediction error $b_m(n-1)$ |
| $\Delta_m(n)$ | Parameter in recursive LSL algorithm |
| $\epsilon_m(n)$ | Angle-normalized joint-process estimation error for prediction order $m$ |
| $\epsilon_{b,m}(n)$ | Angle-normalized backward prediction error for prediction order $m$ |
| $\epsilon_{f,m}(n)$ | Angle-normalized forward prediction error for prediction order $m$ |
| $\boldsymbol{\epsilon}(n)$ | Weight-error vector |
| $\boldsymbol{\epsilon}$ | Estimation error vector in the covariance method |

| | |
|---|---|
| $\epsilon(n)$ | Estimation error vector in the prewindowed method |
| $\eta(n)$ | Forward (*a priori*) prediction error |
| $\boldsymbol{\theta}$ | Parameter vector |
| $\boldsymbol{\Theta}$ | Unitary rotation |
| $\kappa_m$ | $m$th reflection of a lattice predictor for stationary environment |
| $\kappa_{b,m}(n)$ | $m$th backward reflection coefficient of a least-squares lattice for a non-stationary environment |
| $\kappa_{f,m}(n)$ | $m$th forward reflection coefficient of a least squares lattice for a non-stationary environment |
| $\kappa_m(n)$ | $m$th joint-process regression coefficient in the recursive LSL and QLD-LSL algorithms at iteration $n$ |
| $\kappa_4(\tau_1, \tau_2, \tau_3)$ | Tricepstrum |
| $\lambda$ | Exponential weighting vector in RLS, FTF, LSL, QR -RLS, and QRD–LSL algorithms |
| $\lambda_k$ | $k$th eigenvalue of correlation matrix $\mathbf{R}$ |
| $\lambda_{\max}$ | Maximum eigenvalue of correlation matrix $\mathbf{R}$ |
| $\lambda_{\min}$ | Minimum eigenvalue of correlation matrix $\mathbf{R}$ |
| $\Lambda$ | Likelihood ratio |
| $\ln \Lambda$ | Log-likelihood ratio |
| $\boldsymbol{\Lambda}(n)$ | Diagonal matrix of exponential weighting factors |
| $\mu$ | Mean value |
| $\mu$ | Step-size parameter in steepest-descent algorithm or LMS algorithm |
| $\mu$ | Constant used in the FTF algorithm with soft constraint |
| $\nu(n)$ | Normalized weight-error vector in steepest-descent algorithm |
| $\boldsymbol{\pi}$ | Vector in RLS algorithm |
| $\pi_m(n)$ | $m$th parameter in QRD–LSL algorithm |
| $\xi(n)$ | *Apriori* estimation error |
| $\phi(t, k)$ | $t, k$th element of time-averaged correlation matrix $\boldsymbol{\Phi}$ |
| $\varphi(n, n_0)$ | Transition matrix arising in finite-precision analysis of RLS algorithms |
| $\boldsymbol{\Phi}$ | Interpolation matrix in RBF network |
| $\boldsymbol{\Phi}$ | Time-averaged correlation matrix |
| $\boldsymbol{\Phi}(n)$ | Time-averaged correlation matrix expressed as a function of the observation interval $n$ |
| $\chi^2(n)$ | Chi-square distributed random variable with $n$ degrees of freedom |

| | |
|---|---|
| $\varphi(v)$ | Activation function of a neuron, expressed as a function of input $v$ |
| $\chi(\mathbf{R})$ | Eigenvalue spread (i.e., ratio of maximum eigenvalue to minimum eigenvalue) of correlation matrix $\mathbf{R}$ |
| $\omega$ | Normalized angular frequency; $0 < \omega \leq 2\pi$ |
| $\boldsymbol{\omega}(n)$ | Process noise vector in Markov model |
| $\rho_m$ | Correlation coefficient or normalized value of autocorrelation function for lag $m$ |
| $\sigma^2$ | Variance |
| $\tau_k$ | Time constant of $k$th natural mode of steepest-descent algorithm |
| $\tau_{\text{mse,av}}$ | Time constant of a single decaying exponential that approximates the learning curve of LMS algorithm |
| $\nabla(n)$ | Convolutional noise in blind equalization |
| $\nabla$ | Gradient vector |

# Bibliography


ABRAHAM, J. A., ET AL. (1987). "Fault tolerance techniques for systolic arrays," *Computer*, vol. 20, pp. 65–75.

AKAIKE, H. (1973). "Maximum likelihood identification of Gaussian autoregressive moving average models," *Biometrika*, vol. 60, pp. 255–265.

AKAIKE, H. (1974). "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, pp. 716–723.

AKAIKE, H. (1977). "An entropy maximisation principle," in *Proceedings of Symposium on Applied Statistics*, ed. P. Krishnaiah, North-Holland, Amsterdam.

ALBERT, A. E., and L. S. GARDNER, JR. (1967). *Stochastic Approximation and Nonlinear Regression*, MIT Press, Cambridge, Mass.

AL-MASHOUQ, K.A., and I. S. REED (1994). "The use of neural nets to combine equalization with decoding for severe intersymbol interference channels," *IEEE Trans. Neural Networks*, vol. 5, pp. 982–988.

ALEXANDER, S. T. (1986a). *Adaptive Signal Processing: Theory and Applications*, Springer-Verlag, New York.

ALEXANDER, S. T. (1986b). "Fast adaptive filters: a geometrical approach," *IEEE ASSP Mag.*, pp. 18–28.

ALEXANDER, S. T. (1987). "Transient weight misadjustment properties for the finite precision LMS algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 1250–1258.

ALEXANDER, S. T., and A. L. GHIRNIKAR (1993). "A method for recursive least-squares filtering based upon an inverse QR decomposition," *IEEE Trans. Signal Process.*, vol. 41, pp. 20–30.

ANDERSON, T. W. (1963). "Asymptotic theory for principal component analysis," *Ann. Math. Stat.*, vol. 34, pp. 122–148.

ANDERSON, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*, 2nd ed., Wiley, New York.

ANDERSON, B. D. O., and J. B. MOORE (1979). *Linear Optimal Control*, Prentice-Hall, Englewood Cliffs, N.J.

ANDERSON, C. H., and D. C. VAN ESSEN (1994). "Neurobiological computational systems," presented at the *IEEE World Congress on Computational Intelligence*, Orlando, Fla., June 26–July 1, 11 pages.

ANDREWS, H. C., and C. L. PATTERSON (1975). "Singular value decomposition and digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-24, pp. 26–53.

APPLEBAUM, S. P. (1966). "Adaptive arrays," Syracuse University Research Corporation, Rep. SPL TR 66-1.

APPLEBAUM, S. P., and D. J. CHAPMAN (1976). "Adaptive arrays with main beam constraints," *IEEE Trans. Antennas Propag.*, vol. AP-24, pp. 650–662.

ARDALAN, S. H. (1986). "Floating-point error analysis of recursive least-squares and least-mean-squares adaptive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 1192–1208.

ARDALAN, S. H., and S. T. ALEXANDER (1987). "Fixed-point roundoff error analysis of the exponentially windowed RLS algorithm for time varying systems," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 770–783.

ÅSTRÖM, K. J., and P. EYKHOFF (1971). "System identification—a survey," *Automatica*, vol. 7, pp. 123–162.

ATAL, B. S., and S. L. HANAUER (1971). "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Am.*, vol. 50, pp. 637–655.

ATAL, B. S., and M. R. SCHROEDER (1970). "Adaptive predictive coding of speech signals," *Bell Syst. Tech. J.*, vol. 49, pp. 1973–1986.

AUSTIN, M. E. (1967). "*Decision-feedback equalization for digital communication over dispersive channels*," Tech. Rep. 437, MIT Lincoln Laboratory, Lexington, Mass.

AUTONNE, L. (1902). "Sur les groupes linéaires, réels et orthogonaux," *Bull. Soc. Math., France*, vol. 30, pp. 121–133.

BARRETT, J. F. and D. G. LAMPARD (1955). "An expansion for some second-order probability distributions and its application to noise problems," *IRE Trans. Information Theory*, vol. IT-1, pp. 10–15.

BARRON, A. R. (1993). "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Information Theory*, vol. 39, pp. 930–945.

BATTITI, R. (1992). "First- and second-order methods for learning: between steepest descent and Newton's method," *Neural Computation*, vol. 4, pp. 141–166.

BEAUFAYS, F. (1995a). "Transform-domain adaptive filters: an analytical approach," *IEEE Trans. Signal Process.*, vol. 43, pp. 422–431.

BEAUFAYS, F. (1995b). "Two-layer linear structures for fast adaptive filtering," Ph.D. dissertation, Stanford University, Stanford, Calif.

BEAUFAYS, F., and B. WIDROW (1994) "Two-layer linear structures for fast adaptive filtering," *World Congress on Neural Networks*, vol. III, San Diego, Calif., pp. 87–93.

BELFIORE, C. A., and J. H. PARK, JR. (1979). "Decision feedback equalization," *Proc. IEEE*, vol. 67, pp. 1143–1156.

BELLANGER, M. G. (1988a). *Adaptive Filters and Signal Analysis*, Dekker, New York.

BELLANGER, M. G. (1988b). "The FLS-QR algorithm for adaptive filtering," *Signal Process.*, vol. 17, pp. 291–304.

BELLINI, S. (1986). "Bussgang techniques for blind equalization," in GLOBECOM, Houston, Tex., pp. 1634–1640.

BELLINI, S. (1988). "Blind equalization," *Alta Freq.*, vol. 57, pp. 445–450.

BELLINI, S. (1994). "Bussgang techniques for blind deconvolution and equalization," in *Blind Deconvolution*, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J.

BELLINI, S., and F. ROCCA (1986). "Blind deconvolution: polyspectra or Bussgang techniques?" in *Digital Communications*, ed. E. Biglieri and G. Prati, North-Holland, Amsterdam, pp. 251–263.

BELLMAN, R. (1960). *Introduction to Matrix Analysis*, McGraw-Hill, New York.

BENVENISTE, A. (1987). "Design of adaptive algorithms for the tracking of time-varying systems," *Int. J. Adaptive Control Signal Proc.*, vol. 1, pp. 3–29.

BENVENISTE, A., and M. GOURSAT (1984). "Blind equalizers," *IEEE Trans. Commun.*, vol. COM-32, pp. 871–883.

BENVENISTE, A., M. GOURSAT, and G. RUGET (1980). "Robust identification of a nonminimum phase system: blind adjustment of a linear equalizer in data communications," *IEEE Trans. Autom. Control*, vol. AC-25, pp. 385–399.

BENVENISTE, A., M. MÉTIVIER, and P. PRIOURET (1987). *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, New York.

BENVENUTO, N., ET AL. (1986). "The 32 kb/s ADPCM coding standard," *AT&T J.*, vol. 65, pp. 12–22.

BENVENUTO, N., and F. PIAZZA (1992). "On the complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 40, pp. 967–969.

BERGMANS, J. W. M. (1990). "Tracking capabilities of the LMS adaptive filter in the presence of gain variations," *IEEE Trans. Acoust. Speech Signal Process.* vol. 38, pp. 712–714.

BERKHOUT, A. J., and P. R. ZAANEN (1976). "A comparison between Wiener filtering, Kalman filtering, and deterministic least squares estimation," *Geophysical Prospect.*, vol. 24, pp. 141–197.

BERSHAD, N. J. (1986). "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 793–806.

BERSHAD, N. J., and P. L. FEINTUCH (1986). "A normalized frequency domain LMS adaptive algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 452–461.

BERSHAD, N. J., and L. Z. QU (1989). "On the probability density function of the LMS adaptive filter weights," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-37, pp. 43–56.

BERSHAD, N. J., and O. MACCHI (1991). "Adaptive recovery of a chirped sinusoid in noise, Part 2: Performance of the LMS algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 39, pp. 595–602.

BIERMAN, G. J. (1977). *Factorization Methods for Discrete Sequential Estimation*, Academic Press New York.

BIERMAN, G. J., and C. L. THORNTON (1977). "Numerical comparison of Kalman filter algorithms orbit determination case study," *Automatica*, vol. 13, pp. 23–35.

BIRKETT, A. N., and R. A. GOUBRAN, 1995. "Acoustic echo cancellation using NI MS-neural network structures," in *Proc. ICASSP*, Detroit, Michigan. vol. 5, pp. 3035–3038.

BIRX, D. L., and S. J. PIPENBERG (1992). "Chaotic oscillators and complex mapping feed forward networks (CMFFNS) for signal detection in noisy environments," in *International Joint Conference on Neural Networks*, Baltimore, MD, vol. 2, pp. 881–888.

BITMEAD, R. R. and B. D. O. ANDERSON (1980a). "Lyapunov techniques for the exponential stability of linear difference equations with random coefficients," *IEEF Trans. Autom. Control*, vol. AC-25, pp. 782–787.

BITMEAD, R. R., and B. D. O. ANDERSON (1980b). "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Trans. Autom. Control*, vol. AC-25, pp. 788–794.

BITMEAD, R. P., and B. D. O. ANDERSON (1981). "Adaptive frequency sampling filters, *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 524–535.

BJÖRCK, A. (1967). "Solving linear least squares problems by Gram-Schmidt orthogonalization," *BIT*, vol. 7, pp. 1–21.

BODE, H. W., and C. E. SHANNON (1950). "A simplified derivation of linear least square smoothing and prediction theory," *Proc. IRE*, vol. 38, pp. 417–425.

BOJANCZYK, A. W., and F. T. LUK (1990). "A unified systolic array for adaptive beamforming," *J. Parallel Distrib. Comput.*, vol. 8, pp. 388–392.

BORAY, G. K., and M. D. SRINATH (1992). "Conjugate gradient techniques for adaptive filtering," *IEEE Trans. Circuits Syst. Fundam. Theory Appli.*, vol. 39, pp. 1–10.

BOTTO, J. L., and G. V. MOUSTAKIDES (1989). "Stabilizing the fast Kalman algorithms," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-37, pp. 1342–1348.

BOTTOMLEY, G. E., and S. T. ALEXANDER (1989). "A theoretical basis for the divergence of conventional recursive least squares filters," in *Proc. ICASSP*, Glasgow, Scotland, pp. 908–911.

BOX, G. E. P., and G. M. JENKINS (1976). *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.

BRACEWELL, R. N. (1978). *The Fourier Transform and Its Applications*, 2nd ed., McGraw-Hill, New York.

BRADY, D. M. (1970). "An adaptive coherent diversity receiver for data transmission through dispersive media," in *Conf. Rec. ICC 70*, pp. 21-35–21-40.

BRENT, R. P., F. T. LUK, and C. VAN LOAN (1983). "Decomposition of the singular value decomposition using mesh-connected processors," *J. VLSI Comput. Syst.*, vol. 1, pp. 242–270.

BRILLINGER, D. R. (1974). *Time Series: Data Analysis and Theory*, Holt, Rinehart, and Winston, New York.

BROGAN, W. L. (1985). *Modern Control Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, N.J.

BROOKS, L. W., and I. S. REED (1972). "Equivalence of the likelihood ratio processor, the maximum signal-to-noise ratio filter, and the Wiener filter," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-8, pp. 690–692.

BROOMHEAD, D. S., and D. LOWE (1988). "Multi-variable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 269–303.

BROSSIER, J. M. (1992). "Egalisation adaptive et estimation de phase: Application aux communications sous-marines," These de Docteur, del'Institut National Polytechnique de Grenoble, France.

BRUCKSTEIN, A., and T. KAILATH (1987). "An inverse scattering framework for several problems in signal processing," *IEEE ASSP Mag.*, vol. 4, pp. 6–20.

BUCKLEW, J. A., T. KURTZ, and W. A. SETHARES (1993). "Weak convergence and local stability properties of fixed stepsize recursive algorithms," *IEEE Trans. Information Theory*, vol. 39, pp. 966–978.

BUCKLEY, K. M., and L. J. GRIFFITHS (1986). "An adaptive generalized sidelobe canceller with derivative constraints," *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 311–319.

BUCY, R. S. (1994). *Lectures on Discrete Time Filtering*, Springer-Verlag, New York.

BURG, J. P. (1967). "Maximum entropy spectral analysis," in *37th Ann. Int. Meet., Soc. Explor. Geophys.*, Oklahoma City, Okla.

BURG, J. P. (1968). "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing, Enschede, The Netherlands.

BURG, J. P. (1972). "The relationship between maximum entropy spectra and maximum likelihood spectra," *Geophysics*, vol. 37, pp. 375–376.

BURG, J. P. (1975). "Maximum Entropy Spectral Analysis," Ph.D. dissertation, Stanford University, Stanford, Calif.

BUSSGANG, J. J. (1952). Cross Correlation Functions of Amplitude-Distorted Gaussian Signals, Tech. Rep. 216, MIT Research Laboratory of Electronics, Cambridge, Mass.

BUTTERWECK, H. J. (1995a). "A steady-state analysis of the LMS adaptive algorithm without use of the independence assumption," in *Proc. ICASSP*, Detroit, Michigan, pp. 1404–1407.

BUTTERWECK, H. J. (1995b). "Iterative analysis of the steady-state weight fluctuations in LMS-type adaptive filters," private communication.

CAPMAN, F., J. BOUDY, and P. LOCKWOOD (1995). "Acoustic echo cancellation using a fast QR-RLS algorithm and multirate schemes," in *Proc. ICASSP*, Detroit, Michigan, pp. 969–971.

CAPON, J. (1969). "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, pp. 1408–1418.

CARAISCOS, C., and B. LIU (1984). "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-32, pp. 34–41.

CARAYANNIS, G., D. G. MANOLAKIS, and N. KALOUPTSIDIS (1983). "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 1394–1402.

CHANG, R. W. (1971). "A new equalizer structure for fast start-up digital communications," *Bell Syst. Tech. J.*, vol. 50, pp. 1969-2014.

CHAO, J., H. PEREZ, and S. TSUJII (1990). "A fast adaptive filter algorithm using eigenvalue reciprocals as step sizes," *IEEE Trans. Acoust. Speech Signal Process.* vol. ASSP-38, pp. 1343–1352.

CHAZAN, D., Y. MEDAN, and U. SHVADRON (1988). "Noise cancellation for hearing aids," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-36, pp. 1697–1705.

CHEN, S. (1995). "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electronics Letters*, vol. 31, no. 2, pp. 117–118.

CHEN, S., S. MCLAUGHLIN, and B. MULGREW, (1994a). "Complex-valued radial basis function network, Part I: Network architecture and learning algorithms," *Signal Proc.* vol. 35, pp. 19–31.

CHEN, S., S. MCLAUGHLIN, and B. MULGREW, (1994b). "Complex-valued radial basis function network, Part II: Application to digital communications channel equalisation," *Signal Proc.*, vol. 36, pp. 175–188.

CHESTER, D. L. (1990). "Why two hidden layers are better than one," *International Joint Conference on Neural Networks*, Washington, D.C., vol. 1, pp. 265–268.

CHILDERS, D. G., ed. (1978). *Modern Spectrum Analysis*, IEEE Press, New York.

CHINRUNGRUENG, C., and C. H. SÉQUIN (1995). "Optimal adaptive $k$-means algorithm with dynamic adjustment of learning rate," *IEEE Trans. Neural Networks*, vol. 6, pp. 157–169.

CHUI, C. K., and G. CHEN (1987). *Kalman Filtering with Real-time Application*, Springer-Verlag, New York.

CIOFFI, J. M. (1987). "Limited-precision effects in adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 821–833.

CIOFFI, J. M. (1988). "High speed systolic implementation of fast QR adaptive filters," in *Proc* ICASSP, New York, pp. 1584–1588.

CIOFFI, J. M. (1990). "The fast adaptive rotor's RLS algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-38, pp. 631–653.

CIOFFI, J. M., and T. KAILATH (1984). "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-32, pp. 304–337.

CLASSEN, T. A. C. M., and W. F. G. MECKLANBRÄUKER (1985). "Adaptive techniques for signal processing in communications," *IEEE Commun.*, vol. 23, pp. 8–19.

CLARK, G. A., S. K. MITRA, and S. R. PARKER (1981). "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 584–592.

CLARK, G. A., S. R. PARKER, and S. K. MITRA (1983). "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 1073–1083.

CLARKE, T. L. (1990). "Generalization of neural networks to the complex plane," *International Joint Conference on Neural Networks*, San Diego, Calif., vol. II, pp. 435–440.

CLARKSON, P. M. (1993). *Optimal and Adaptive Signal Processing*, CRC Press, Boca Raton, Fla.

COHN, A. (1922). "Über die Anzahl der Wurzeln einer algebraischen Gleichung in einem Kreise," *Math. Z.*, vol. 14, pp. 110–148.

COMPTON, R. T. (1988). *Adaptive Antennas: Concepts and Performance*, Prentice-Hall, Englewood Cliffs, N.J.

COWAN, C. F. N. (1987). "Performance comparisons of finite linear adaptive filters," *IEE Proc. (London)*, part F, vol. 134, pp. 211–216.

COWAN, C. F. N., and P. M. GRANT (1985). *Adaptive Filters*, Prentice-Hall, Englewood Cliffs, N. J.

COWAN, J. D. (1990). "Neural networks: the early days." In *Advances in Neural Information Processing Systems* 2, ed. D. S. Touretzky, pp. 828–842, Morgan Kaufman, San Mateo, Calif.

COX, H., R. M. ZESKIND, and M. M. OWEN (1987). "Robust adaptive beamforming," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 1365–1376.

CROCHIERE, R. E., and L. R. RABINER (1983). *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

CUTLER, C. C. (1952). *Differential Quantization for Communication Signals*, U. S. Patent 2,605,361.

CYBENKO, G. (1989). *Approximation by superpositions of a sigmoidal function,"* Mathematics of Control, Signals, and Systems, vol. 2, pp. 303–314.

DE COURVILLE, M., and P. DUHAMEL (1995). "Adaptive filtering in subbands using a weighted criterion," in *Proc. ICASSP*, Detroit, Michigan, vol. 2, pp. 985–988.

DEIFT, P. J., DEMMEL, C. TOMAL, and L.-C. LI, (1989). The Bidiagonal Singular Value Decomposition and Hamiltonian Mechanics, Rep. 458, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York.

DEMMEL, J. (1994). "Designing High Performance Linear Algebra Software for Parallel Computers," CS Division and Math Dept., UC Berkeley, December 9.

DEMMEL, J., and W. KAHAN (1990). "Accurate singular values of bidiagonal matrices," *SIAM J. Sci. Stat. Comp.*, vol. 11, pp. 873–912.

DEMMEL, J., and K. VESELIĆ (1989). *Jacobi's Method Is More Accurate than QR*, Tech. Rep. 468, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York.

DEMOOR, B. L. R., and G. H. GOLUB (1989). "Generalized Singular Value Decompositions: A Proposal for a Standardized Nomenclature," Manuscript NA-89-05, Numerical Analysis Project, Computer Science Department, Standard University, Stanford, Calif.

DENTINO, M., J. MCCOOL, and B. WIDROW (1978). "Adaptive filtering in the frequency domain," *Proc. IEEE*, vol. 66, no. 12, pp. 1658–1659.

DEPRETTERE, E. F., ed. (1988). *SVD and Signal Processing: Algorithms, Applications, and Architectures*, North-Holland, Amsterdam.

DEVIJVER, P. A., and J. KITTLER (1982). *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, London.

DEWILDE, P. (1969). "Cascade Scattering Matrix Synthesis," Ph.D. dissertation, Stanford University, Stanford, Calif.

DEWILDE, P., A. C. VIEIRA, and T. KAILATH (1978). "On a generalized Szegö–Levinson realization algorithm for optimal linear predictors based on a network synthesis approach," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 663–675.

DHRYMUS, P. J. (1970). *Econometrics: Statistical Foundations and Applications*, Harper & Row, New York.

DING, Z. (1994). "Blind channel identification and equalization using spectral correlation measurements, Part I: Frequency-domain approach," in *Cyclostationarity in Communications and Signal Processing*," ed. W. A. Gardner, IEEE Press, New York, pp. 417–436.

DING, Z., and Z. MAO (1995). "Knowledge based identification of fractionally sampled channels," in *Proc. ICASSP*, Detroit, Michigan, vol. 3, pp. 1996–1999.

DING, Z., C. R. JOHNSON, JR., and R. A. KENNEDY (1994). "Global convergence issues with linear blind adaptive equalizers," in *Blind Deconvolution*, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J.

DINIZ, P. S. R., and L. W. P. BISCAINHO (1992). "Optimal variable step size for the LMS/Newton algorithm with application to subband adaptive filtering," *IEEE Trans. Signal Process.*, vol. 40, pp. 2825–2829.

DITORO, M. J. (1965). "A new method for high speed adaptive signal communication through any time variable and dispersive transmission medium," in *1st IEEE Annu. Commun. Conf.*, pp. 763–767.

DONGARRA, J. J., ET AL. (1979). *LINPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia.

DONOHO, D. L. (1981). "On minimum entropy deconvolution," in *Applied Time Series Analysis II*, ed. D. F. Findlay, Academic Press, New York.

DOOB, L. J., (1953). *Stochastic Processes*, Wiley, New York.

DORNY, C. N. (1975). *A Vector Space Approach to Models and Optimization*, Wiley-Interscience, New York.

DOUGLAS, S. C. (1994). "A family of normalized LMS algorithms," *IEEE Signal Processing Letters*, vol. 1, pp. 49–51.

DOUGLAS, S. C., and T. H.-Y. MENG (1994). "Normalized data nonlinearities for LMS adaptation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 42, pp. 1352–1365.

DOYLE, J. C., K. GLOVER, P. KHARGONEKAR, and B. FRANCIS (1989). "State-space solutions to standard $H_2$ and $H_\infty$ control problems," *IEEE Trans. Autom. Control*, vol. AC-34, pp. 831–847.

DUGARD, L., M. M'SAAD, and I. D. LANDAU (1993). *Adaptive Systems in Control and Signal Processing*, Pergamon Press, Oxford, United Kingdom.

DUDA, R. O., and P. E. HART (1973). *Pattern Classification and Scene Analysis*, Wiley, New York.

DUHAMEL, P. (1995). "Tutorial: Blind equalization," *The 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, Michigan.

DURBIN, J. (1960). "The fitting of time series models," *Rev. Int. Stat. Inst.*, vol. 28, pp. 233–244.

DUTTWEILER, D. L., and Y. S. CHEN (1980). "A single-chip VLSI echo canceler," *Bell Syst. Tech. J.*, vol. 59, pp. 149–160.

ECKART, G., and G. YOUNG (1936). "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218.

ECKART, G., and G. YOUNG (1939). "A principal axis transformation for non-Hermitian matrices," *Bull. Am. Math. Soc.*, vol. 45, pp. 118–121.

EDWARDS, A. W. F. (1972). *Likelihood*, Cambridge University Press, New York.

ELEFTHERIOU, E., and D. D. FALCONER (1986). "Tracking properties and steady state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 1097–1110.

EWEDA, E. (1994). "Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels," *IEEE Trans. Signal Process.*, vol. 42, pp. 2937–2944.

EWEDA, E., and O. MACCHI (1985). "Tracking error bounds of adaptive nonstationary filtering," *Automatica*, vol. 21, pp. 293–302.

EWEDA, E., and O. MACCHI (1987). "Convergence of the RLS and LMS adaptive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 799–803.

FALCONER, D. D., and L. LJUNG (1978). "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446.

FARDEN, D. C. (1981a). "Stochastic approximation with correlated data," *IEEE Trans. Information Theory*, vol. IT-27, pp. 105–113.

FARDEN, D. C. (1981b). "Tracking properties of adaptive signal processing algorithms," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 439–446.

FELDKAMP, L. A., G. V. PUSKORIUS, L. I. DAVIS, JR., and F. YUAN (1994). "Enabling concepts for application of neurocontrol," in *Proc. Eighth Yale Workshop on Adaptive and Learning Systems*, Yale University, New Haven, Conn., pp. 168–173.

FERNANDO, K. V., and B. N. PARLETT (1994). "Accurate singular values and differential qd algorithms," *Numerische Mathematik*, vol. 67, pp. 191–229.

FERRARA, E. R., JR., (1980). "Fast implementation of LMS adaptive filters," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-28, pp. 474–475.

FERRARA, E. R., JR., (1985). "Frequency-domain adaptive filtering," in *Adaptive Filters*, ed. C. F. N. Cowan and P. M. Grant, pp. 145–179, Prentice-Hall, Englewood Cliffs, N.J.

FIALKOW, I., J. R. TREICHLER, and C. R. JOHNSON JR. (1995). "Fractionally spaced blind equalization: loss of channel disparity," in *Proc. ICASSP*, Detroit, Michigan, pp. 1988–1991.

FISHER, B., and N. J. BERSHAD (1983). "The complex LMS adaptive algorithm—transient weight mean and covariance with applications to the ALE," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 34–44.

FLANAGAN, J. L. (1972). *Speech Analysis, Synthesis and Perception*, 2nd ed., Springer-Verlag, New York.

FLANAGAN, J. L., ET AL. (1979). "Speech coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 710–737.

FOLEY, J. B., and F. M. BOLAND (1987). "Comparison between steepest descent and LMS algorithms in adaptive filters," *IEE Proc. (London), Part F*, vol. 134, pp. 283–289.

FOLEY, J. B., and F. M. BOLAND (1988). "A note on the convergence analysis of LMS adaptive filters with Gaussian data," *IEEE Trans Acoust Speech Signal Process.*, vol. 36, pp. 1087–1089.

FORNEY, G. D. (1972). "Maximum-likelihood sequence estimation of digital sequence in the presence of intersymbol interference," *IEEE Trans. Information Theory*, vol. IT-18, pp. 363–378.

FORSYTHE, G. E., and P. HENRICI (1960). "The cyclic Jacobi method for computing the principal values of a complex matrix," *Trans. Am. Math. Soc.*, vol. 94, pp. 1–23.

FOSCHINI, G. J. (1985). "Equalizing without altering or detecting data," *AT&T Tech. J.*, vol. 64, pp. 1885–1911.

FRANKS, L. E. (1969). *Signal Theory*. Prentice-Hall, Englewood Cliffs, N.J.

FRANKS, L. E., ed. (1974). *Data Communication: Fundamentals of Baseband Transmission*, Benchmark Papers in Electrical Engineering and Computer Science, Dowden, Hutchinson & Ross, Stroudsburg, Pa.

FRASER, D. C. (1967). "A new technique for the optimal smoothing of data," Sc.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass.

FRIEDLANDER, B. (1982). "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, pp. 829–867.

FRIEDLANDER, B. (1988). "A signal subspace method for adaptive interference cancellation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-36, pp. 1835–1845.

FRIEDLANDER, B., and B. PORAT (1989). "Adaptive IIR algorithm based on high-order statistics," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-37, pp. 485–495.

FRIEDRICHS, B. (1992). "Analysis of finite-precision adaptive filters. I. Computation of the residual signal variance," *Frequenz*, vol. 46, pp. 218–223.

FROST III, O. L. (1972). "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, pp. 926–935.

FUKUNAGA, K. (1990). *Statistical Pattern Recognition*, 2nd ed., Academic Press, New York.

FUNAHASHI, K. (1989). "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192.

GABOR, D., W. P. L. WILBY, and R. WOODCOCK (1961). "A universal non-linear filter, predictor and simulator which optimizes itself by a learning process," *IEE Proc. (London)*, vol. 108, pt. B, pp. 422–438.

GABRIEL, W. F. (1976). "Adaptive arrays: an introduction," *Proc. IEEE*, vol. 64, pp. 239–272.

GALLIVAN, K. A., and C. E. LEISERSON (1984). "High-performance architectures for adaptive filtering based on the Gram–Schmidt algorithm," in *Proc. SPIE*, vol. 495, Real Time Signal Processing VII, pp. 30–38.

GARBOW, B. S., ET AL. (1977). *Matrix Eigensystem Routines—EISPACK Guide Extension*, Lecture Notes in Computer Science, vol. 51, Springer-Verlag, New York.

GARDNER, W. A. (1984). "Learning characteristics of stochastic-gradient-descent algorithms: a general study, analysis and critique," *Signal Process.*, vol. 6, pp. 113–133.

GARDNER, W. A. (1987). "Nonstationary learning characteristics of the LMS algorithm," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1199–1207.

GARDNER, W. A. (1990). *Introduction to Random Processes with Applications to Signals and Systems*, McGraw-Hill, New York.

GARDNER, W. A. (1991). "A new method of channel identification," *IEEE Trans. Commun.*, vol. COM-39, pp. 813–817.

GARDNER, W. A. (1993). "Cyclic Wiener filtering: Theory and method", *IEEE Trans. Signal Process.*, vol. 41, pp. 151–163.

GARDNER, W. A. ed. (1994a). *Cyclostationarity in Communications and Signal Processing*, IEEE Press, New York.

GARDNER, W. A. (1994b). "An introduction to cyclostationary signals," in *Cyclostationarity in Communications and Signal Processing*, ed., W. A. Gardner, IEEE Press, New York, pp. 1–90.

GARDNER, W. A., and L. E. FRANKS (1975). "Characterization of cyclostationary random signal processes," *IEEE Trans. Information Theory*, vol. IT-21, pp. 4–14.

GARDNER, W. A., and C. M. SPOONER (1994). "The cumulant theory of cyclostationary time-series, Part I: foundation," *IEEE Trans. Signal Process.*, vol. 42, pp. 3387–3408.

GAUSS, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicus solem ambientum*, Hamburg (translation: Dover, New York, 1963).

GELB, A., ed. (1974). *Applied Optimal Estimation*, MIT Press, Cambridge, Mass.

GENTLEMAN, W. M. (1973). "Least squares computations by Givens transformations without square-roots," *J. Inst. Math. Its Appl.*, vol. 12, pp. 329–336.

GENTLEMAN, W. M:, and H. T. KUNG (1981). "Matrix triangularization by systolic arrays," in *Proc. SPIE*, vol. 298, Real Time Signal Processing IV, pp. 298–303.

GEORGIOU, G. N., and C. KOUTSOUGERAS (1992). "Complex domain backpropagation," *IEEE Trans. Circuits Syst. Part II: Analog and Digital Signal Processing*, vol. 39, pp. 330–334.

GERSHO, A. (1968). "Adaptation in a quantized parameter space," in *Proc. Allerton Conf. on Circuit and System Theory*, Urbana, Il., pp. 646–653.

GERSHO, A. (1969). "Adaptive equalization of highly dispersive channels for data transmission," *Bell Syst. Tech. J.*, vol. 48, pp. 55–70.

GERSHO, A., B. GOPINATH, and A.M. OLDYZKO (1979). "Coefficient inaccuracy in transversal filtering," *Bell Syst. Tech. J.*, vol. 58, pp. 2301–2316.

GHOLKAR, V. A. (1990). "Mean square convergence analysis of LMS algorithm (adaptive filters)," *Electron Letters*, vol. 26, pp. 1705–1706.

GIANNAKIS, G. B., and S. D. HALFORD (1995). "Blind fractionally-spaced equalization of noisy FIR channels: adaptive and optimal solutions," in *Proc. ICASSP*, Detroit, Michigan, pp. 1972–1975.

GIBSON, J. D. (1980). "Adaptive prediction in speech differential encoding systems," *Proc. IEEE*, vol. 68, pp. 488–525.

GIBSON, G. J. and C. F. N. COWAN (1990). "On the decision regions of multilayer perceptrons," *Proc. IEEE*, vol. 78, pp. 1590–1599.

GILLOIRE, A., and M. VETTERLI, 1992. "Adaptive filtering in subbands with critical sampling: analysis, experiments, and applications to acoustic echo cancellation," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 1862–1875.

GILL, P. E., G. H. GOLUB, W. MURRAY, and M. A. SAUNDERS (1974). "Methods of modifying matrix factorizations," *Math. Comput.*, vol. 28, pp. 505–535.

GITLIN, R. D., and F. R. MAGEE, Jr., (1977). "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Commun.*, vol. COM-25, pp. 666–672.

GITLIN, R. D., and S. B. WEINSTEIN (1979). "On the required tap-weight precision for digitally implemented mean-squared equalizers," *Bell Syst. Tech. J.*, vol. 58, pp. 301–321.

GITLIN, R. D., and S. B. WEINSTEIN (1981). "Fractionally spaced equalization: an improved digital transversal equalizer," *Bell Syst. Tech. J.*, vol. 60, pp. 275–296.

GITLIN, R. D., J. E. MAZO, and M. G. TAYLOR (1973). "On the design of gradient algorithms for digitally implemented adaptive filters," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 125–136.

GIVENS, W. (1958). "Computation of plane unitary rotations transforming a general matrix to triangular form," *J. Soc. Ind. Appl. Math*, vol. 6, pp. 26–50.

GLASER, E. M. (1961). "Signal detection by adaptive filters," *IRE Trans. Information Theory*, vol. IT-7, pp. 87–98.

GLOVER, J. R., JR. (1977). "Adaptive noise cancelling applied to sinusoidal interferences," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-25, pp. 484–491.

GODARA, L. C., and A. CANTONI (1986). "Analysis of constrained LMS algorithm with application to adaptive beamforming using perturbation sequences," *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 368–379.

GODARD, D. N. (1974). "Channel equalization using a Kalman filter for fast data transmission," *IBM J. Res. Dev.*, vol. 18, pp. 267–273.

GODARD, D. N. (1980). "Self-recovering equalization and carrier tracking in a two-dimensional data communication system," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867–1875.

GODFREY R., and F. ROCCA (1981). "Zero memory non-linear deconvolution," *Geophys. Prospect.*, vol. 29, pp. 189–228.

GOLD, B. (1977). "Digital speech networks," *Proc. IEEE*, vol. 65, pp. 1636–1658.

GOLOMB, S. W., ed. (1964). *Digital Communications with Space Applications*, Prentice-Hall, Englewood Cliffs, N.J.

GOLUB, G. H., (1965). "Numerical methods for solving linear least squares problems," *Numer. Math.*, vol. 7, pp. 206–216.

GOLUB, G. H., and W. KAHAN (1965). "Calculating the singular values and pseudo-inverse of a matrix," *J. SIAM Numer. Anal. B.*, vol. 2, pp. 205–224.

GOLUB, G. H., and C. REINSCH (1970). "Singular value decomposition and least squares problems," *Numer. Math.*, vol. 14, pp. 403–420.

GOLUB, G. H., and C. F. VAN LOAN (1989). *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, Md.

GOLUB, G. H., F. T. LUK, and M. L. OVERTON (1981). "A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix," *ACM Trans. Math. Software*, vol. 7, pp. 149–169.

GOODWIN, G. C., and R. L. PAYNE (1977). *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York.

GOODWIN, G. C., and K. S. SIN (1984). *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, N.J.

GRAY, R. M. (1972). "On the asymptotic eigenvalue distribution of Toeplitz matrices," *IEEE Trans. Information Theory*, vol. IT-18, pp. 725–730.

GRAY, R. M. (1977). Toeplitz and Circulant Matrices: II, Tech. Rep. 6504-1, Information Systems Laboratory, Stanford University, Stanford, Calif.

GRAY, R. M. (1990). "Quantization noise spectra," *IEEE Trans. Inform Theory*, vol. 36, pp. 1220–1244.

GRAY, R. M., and L. D. DAVISSON (1986). *Random Processes: A Mathematical Approach for Engineers*, Prentice-Hall, Englewood Cliffs, N.J.

GRAY, W. (1979). "Variable Norm Deconvolution," Ph.D. dissertation, Department of Geophysics, Stanford University, Stanford, Calif.

GREEN, M., and D. J. N. LIMEBEER (1995). *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, N.J.

GRENANDER, U., and G. SZEGÖ (1958). *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, Calif.

GRIFFITHS, L. J. (1975). "Rapid measurement of digital instantaneous frequency," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-23, pp. 207–222.

GRIFFITHS, L. J. (1977). "A continuously adaptive filter implemented as a lattice structure," in *Proc. ICASSP*, Hartford, Conn., pp. 683–686.

GRIFFITHS, L. J. (1978). "An adaptive lattice structure for noise-cancelling applications," in *Proc. ICASSP*, Tulsa, Okla., pp. 87–90.

GRIFFITHS, L. J., and C. W. JIM (1982). "An alternative approach to linearly constrained optimum beamforming," *IEEE Trans. Antennas Propag.*, vol. AP-30, pp. 27–34.

GRIFFITHS, L. J., and R. PRIETO-DIAZ (1977). "Spectral analysis of natural seismic events using autoregressive techniques," *IEEE Trans. Geosci. Electron.*, vol. GE-15, pp. 13–25.

GRIFFITHS, L. J., F. R. SMOLKA, and L. D. TREMBLY (1977). "Adaptive deconvolution: a new technique for processing time-varying seismic data," *Geophysics*, vol. 42, pp. 742–759.

GU, M., and S. C. EISENSTAT (1994). "A Divide-and-Conquer Algorithm for the Bidiagonal SVD," Research Report YALEU/DCS/RR-933, UC Berkeley, Calif., April 4.

GU, M., J. DEMMEL, and I. DHILLON (1994). "Efficient Computation of the Singular Value Decomposition with Applications to Least Squares Problems," Department of Mathematics, UC Berkeley, Calif., September 29.

GUNNARSSON, S., and L. LJUNG (1989). "Frequency domain tracking characteristics of adaptive algorithms," *IEEE Trans. Signal Process.*, vol. 37, pp. 1072–1089.

GUILLEMIN, E. A. (1949). *The Mathematics of Circuit Analysis*, Wiley, New York.

GUPTA, I. J., and A. A. KSIENSKI (1986). "Adaptive antenna arrays for weak interfering signals," *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 420–426.

GUTOWSKI, P. R., E. A. ROBINSON, and S. TREITEL (1978). "Spectral estimation: fact or fiction," *IEEE Trans. Geosci. Electron.*, vol. GE-16, pp. 80–84.

HADHOUD, M. M., and D. W. THOMAS (1988). "The two-dimensional adaptive LMS (TDLMS) algorithm," *IEEE Trans. Circuits Syst.*, vol. CAS-35, pp. 485–494.

HAMPEL, F. R., ET AL. (1986). *Robust Statistics: The Approach Based on Influence Functions*, Wiley, New York.

HANSON, R. J., and C. L. LAWSON (1969). "Extensions and applications of the Householder algorithm for solving linear least squares problems," *Math. Comput.*, vol. 23, pp. 787–812.

HARIHARAN, S., and A. P. CLARK (1990). "HF channel estimation using a fast transversal filter algorithm," *IEEE Trans Acoust. Speech Signal Process.* vol. 38, pp. 1353–1362.

HARTMAN, E. J., J. D. KEELER, and J. M. KOWALSKI (1990). "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation*, vol. 2, pp. 210–215.

HASTINGS-JAMES, R., and M. W. SAGE (1969). "Recursive generalized-least-squares procedure for online identification of process parameters," *IEE Proc. (London)*, vol. 116, pp. 2057–2062.

HATZINAKOS, D. (1990). "Blind equalization based on polyspectra," Ph.D. thesis, Northeastern University, Boston, Mass.

HATZINAKOS, D., and C. L. NIKIAS (1989). "Estimation of multipath channel response in frequency selective channels," *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 12–19.

HATZINAKOS, D., and C. L. NIKIAS (1991). "Blind equalization using a tricepstrum based algorithm," *IEEE Trans. Commun.*, vol. COM-39, pp. 669–682.

HATZINAKOS, D., and C. L. NIKIAS (1994). "Blind equalization based on higher-order statistics (HOS)," in *Blind Deconvolution*, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J.

HAYKIN, S., ed. (1983). *Nonlinear Methods of Spectral Analysis*, 2nd ed., Springer-Verlag, New York.

HAYKIN, S., ed. (1984). *Array Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

HAYKIN, S. (1989a). *Modern Filters*, Macmillan, New York.

HAYKIN, S. (1989b). "Adaptive filters: past, present, and future," *Proc. IMA Conf. Math. Signal Process.*, Warwick, England.

HAYKIN, S. (1991). *Adaptive Filter Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

HAYKIN, S. (1994a). *Communication Systems*, 3rd ed., Wiley, New York.

HAYKIN, S. (1994b). *Neural Networks: A Comprehensive Foundation*, Macmillan, New York.

HAYKIN, S., and X. B. LI (1995). "Detection of signals in chaos," *Proc. IEEE*, vol. 83, pp. 95–122.

HAYKIN, S., and A. UKRAINEC (1993). "Neural networks for adaptive signal processing," in *Adaptive System Identification and Signal Processing Algorithms*, ed. N. Kalouptsidis and S. Theodoridis, pp. 512–553, Prentice-Hall, Englewood Cliffs, N.J.

HAYKIN, S., A. H. SAYED, J. R. ZEIDLER, P. YEE, and P. WEI (1995a). "Tracking of linear time-variant systems," MILCOM 95, San Diego, CA.

HAYKIN, S., A.UKRAINEC, B. CURRIE, B. LI, AND M. AUDETTE (1995b). A neural network-based non-coherent radar processor for a chaotic ocean environment," ANNIE, St. Louis, Missouri.

HENSELER, J., and P. J. BRASPENNING (1990). *Training Complex Multi-Layer Neural Networks*, Tech. Rep. CS90-02, University of Limburg, Maastricht, Department of Computer Science, The Netherlands.

HERZBERG, H., and R. HAIMI-COHEN (1992). "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation," *IEEE Trans. Signal Process.* vol. 40, pp. 2799–2803.

HO, Y. C. (1963). "On the stochastic approximation method and optimal filter theory," *J. Math. Anal. Appl.*, vol. 6, pp. 152–154.

HODGKISS, W. S., JR., and D. ALEXANDROU (1983). "Applications of adaptive least-squares lattice structures to problems in underwater acoustics," in *Proc. SPIE*, vol. 431, Real Time Signal Processing VI, pp. 48–54.

HONIG, M. L., and D. G. MESSERSCHMITT (1981). "Convergence properties of an adaptive digital lattice filter," *IEEE Trans. Acous. Speech Signal Process.*, vol. ASSP-29, pp. 642–653.

HONIG, M. L., and D. G. MESSERSCHMITT (1984). *Adaptive Filters: Structures, Algorithms and Applications*, Kluwer, Boston, Mass.

HORNIK, K., M. STINCHCOMBE, and H. WHITE (1989). "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366.

HOROWITZ, L. L., and K. D. SENNE (1981). "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 722–736.

HOUSEHOLDER, A. S. (1958a). "Unitary triangularization of a nonsymmetric matrix," *J. Assoc. Comput. Mach.*, vol. 5, pp. 339–342.

HOUSEHOLDER, A. S. (1958b). "The approximate solution of matrix problems," *J. Assoc. Comput. Mach.*, vol. 5, pp. 204–243.

HOUSEHOLDER, A. S. (1964). *The Theory of Matrices in Numerical Analysis*, Blaisdell, Waltham, Mass.

HOWELLS, P. W. (1965). *Intermediate Frequency Sidelobe Canceller*, U.S. Patent 3,202,990, August 24.

HOWELLS, P. W. (1976). "Explorations in fixed and adaptive resolution at GE and SURC," *IEEE Trans. Antennas Propag.*, vol. AP-24, Special Issue on Adaptive Antennas, pp. 575–584.

HSIA, T. C. (1983). "Convergence analysis of LMS and NLMS adaptive algorithms," in *Proc. ICASSP*, Boston, Mass., pp. 667–670.

HSU, F. M. (1982). "Square root Kalman filtering for high-speed data received over fading dispersive HF channels," *IEEE Trans. Information Theory*, vol. IT-28, pp. 753–763.

HU, Y. H. (1992). "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Process Magazine*, vol. 9, pp. 16–35.

HUBER, P. J. (1981). *Robust Statistics*, Wiley, New York.

HUBING, N. E., and S. T. ALEXANDER (1990). "Statistical analysis of the soft constrained initialization of recursive least squares algorithms," in *Proc. ICASSP*, Albuquerque, N. Mex.

HUDSON, J. E. (1981). *Adaptive Array Principles*, Peregrinus, London.

HUDSON, J. E., and T. J. SHEPHERD (1989). "Parallel weight extraction by a systolic least squares algorithm," in *Proc. SPIE, Advanced Algorithms and Architectures for Signal Processing IV*, vol. 1152, pp. 68–77.

HUHTA, J. C., and J. G. WEBSTER (1973). "60-Hz interference in electrocardiography," *IEEE Trans. Biomed. Eng.*, vol. BME-20, pp. 91–101.

IOANNOU, P. A., (1990). "Robust adaptive control," in *Proc. Sixth Yale Workshop on Adaptive and Learning Systems*, Yale University, New Haven, Conn., pp. 32–39.

IOANNOU, P. A., and P. V. KOKOTOVIC (1983). *Adaptive Systems with Reduced Models*, Springer-Verlag, New York.

ITAKURA, F., and S. SAITO (1971). "Digital filtering techniques for speech analysis and synthesis," in *Proc. 7th Int. Conf. Acoust.*, Budapest, vol. 25-C-1, pp. 261–264.

ITAKURA, F., and S. SAITO (1972). "On the optimum quantization of feature parameters in the PARCOR speech synthesizer," in *IEEE 1972 Conf. Speech Commun. Process.*, New York, pp. 434–437.

JABLON, N. K. (1986). "Steady state analysis of the generalized sidelobe canceller by adaptive noise canceling techniques," *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 330–337.

JABLON, N. K. (1991). "On the complexity of frequency-domain adaptive filtering", *IEEE Trans. Signal Process.*, vol. 39, pp. 2331–2334.

JABLON, N. K. (1992). "Joint blind equalization, carrier recovery, and timing recovery for high-order QAM constellations," *IEEE Trans. Signal Process.*, vol. 40, pp. 1383–1398.

JACOBI, C. G. J. (1846). "Über ein leichtes verfahren, die in der theorie der säkularstörungen vorkommenden gleichungen numerisch aufzulösen," *J. Reine Angew. Math.* vol. 30, pp. 51–95.

JACOBS, R. A. (1988). "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, pp. 295–307.

JAYANT, N. S., and P. NOLL (1984). *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, N.J.

JAYANT, N. S. (1986). "Coding speech," *IEEE Spectrum*, vol. 23, pp. 58–63.

JAYNES, E. T. (1982). "On the rationale of maximum-entropy methods," *Proc. IEEE*, vol. 70, pp. 939–952.

JAZWINSKI, A. H. (1969). "Adaptive filtering," *Automatica*, vol. 5, pp. 475–485.

JAZWINSKI, A. H. (1970). *Stochastic Processes and Filtering Theory*, Academic Press, New York.

JOHANSSON, E. M., F. U. DOWLA, and D. M. GOODMAN (1990) "Back-propagation learning for multilayer feed-forward networks using the conjugate gradient method," Report UCRL-JC-104850, Lawrence Livermore National Laboratory, Livermore, Calif.

JOHNSON, C. R., JR. (1984). "Adaptive IIR filtering: current results and open issues," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 237–250.

JOHNSON, C. R., JR. (1988). *Lectures on Adaptive Parameter Estimation*, Prentice-Hall, Englewood Cliffs, N.J.

JOHNSON, C. R., JR. (1991). "Admissibility in blind adaptive channel equalization: a tutorial survey of an open problem," *IEEE Control Systems Magazine*, vol. 11, pp. 3–15.

JOHNSON, C. R., JR., S. DASGUPTA, and W. A. SETHARES (1988). "Averaging analysis of local stability of a real constant modulus algorithm adaptive filter," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-36, pp. 900–910.

JOHNSON, C. R., JR., B. EGARDT, and G. KUBIN. (1995). "Frequency-domain interpretation of LMS performance," *IEEE Trans. Signal Process.*, (submitted).

JOHNSON, D. H., and P. S. RAO (1990). "On the existence of Gaussian noise," in The 1990 Digital Signal Processing Workshop, New Paltz, NY, Sponsored by IEEE Signal Processing Society, pp. 8.14.1–8.14.2.

JONES, S. K., R. K. CAVIN III, and W. M. REED (1982). "Analysis of error-gradient adaptive linear equalizers for a class of stationary-dependent processes," *IEEE Trans. Information Theory*, vol. IT-28, pp. 318–329.

JOU, J.-Y., and A. ABRAHAM (1986). "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures," *Proc. IEEE*, Special Issue on Fault Tolerance in VLSI, vol. 74, pp. 732–741.

JUSTICE, J. H. (1985). "Array processing in exploration seismology," in *Array Signal Processing*, ed. S. Haykin, pp. 6–114, Prentice-Hall, Englewood Cliffs, N.J.

KAILATH, T. (1960). Estimating Filters for Linear Time-Invariant Channels, Quarterly Progress Rep. 58, MIT Research Laboratory for Electronics, Cambridge, Mass., pp. 185–197.

KAILATH, T. (1968). "An innovations approach to least-squares estimation: Part 1. Linear filtering in additive white noise," *IEEE Trans. Autom. Control*, vol. AC-13, pp. 646–655.

KAILATH, T. (1969). "A generalized likelihood ratio formula for random signals in Gaussian noise," *IEEE Trans. Information Theory*, vol. IT-15, pp. 350–361.

KAILATH, T. (1970). "The innovations approach to detection and estimation theory," *Proc. IEEE*, vol. 58, pp. 680–695.

KAILATH, T. (1974). "A view of three decades of linear filtering theory," *IEEE Trans. Information Theory*, vol. IT-20, pp. 146–181.

KAILATH, T., ed. (1977). *Linear Least-Squares Estimation*, Benchmark Papers in Electrical Engineering and Computer Science, Dowden, Hutchinson & Ross, Stroudsburg, Pa.

KAILATH, T. (1980). *Linear Systems*, Prentice-Hall, Englewood Cliffs, N.J.

KAILATH, T. (1981). *Lectures on Linear Least-Squares Estimation*, Springer-Verlag, New York.

KAILATH, T. (1982). "Time-variant and time-invariant lattice filters for nonstationary processes," in *Outils et Modèles Mathématique pour l'Automatique, l'Analyse de Systèms et le Traitement du Signal*, vol. 2, ed. I. Laudau, CNRS, Paris, pp. 417–464.

KAILATH, T., and P. A. FROST (1968). "An innovations approach to least-squares estimation: Part 2. Linear smoothing in additive white noise," *IEEE Trans. Autom. Control*, vol. AC-13, pp. 655–660.

KAILATH, T., and R. A. GEESEY (1973). "An innovations approach to least-squares estimation: Part 5. Innovation representations and recursive estimation in colored noise," *IEEE Trans. Autom. Control*, vol. AC-18, pp. 435–453.

KAILATH, T., A. VIEIRA, and M. MORF (1978). "Inverses of Toeplitz operators, innovations, and orthogonal polynomials," *SIAM Rev.*, vol. 20, pp. 106–119.

KALLMANN, H. J. (1940). "Transversal filters," *Proc. IRE*, vol. 28, pp. 302–310.

KALMAN, R. E. (1960). "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, pp. 35–45.

KALMAN, R. E., and R. S. BUCY (1961). "New results in linear filtering and prediction theory," *Trans. ASME, J. Basic Eng.*, vol. 83, pp. 95–108.

KALOUPTSIDIS, N., and S. THEODORIDIS (1987). "Parallel implementation of efficient LS algorithms for filtering and prediction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 1565–1569.

KALOUPTSIDIS, N., and S. THEODORIDIS, eds. (1993). *Adaptive System Identification and Signal Processing Algorithms*, Prentice-Hall, Englewood Cliffs, N.J.

KAMINSKI, P. G., A. E. BRYSON, and S. F. SCHMIDT (1971). "Discrete square root filtering: A survey of current techniques," *IEE Trans. Autom. Control*, vol. AC-16, pp.727–735.

KANG, G. S., and L. J. Fransen (1987). "Experimentation with an adaptive noise-cancellation filter," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 753–758.

KASSAM, S. A., and I. CHA (1993). "Radial basis functions networks in nonlinear signal processing applications," in Conf. Rec. Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, Calif., pp. 1021–1025.

KAY, S. M. (1988). *Modern Spectral Estimation: Theory and Application*, Prentice-Hall, Englewood Cliffs, N.J.

KAY, S. M., and L. S. MARPLE, JR. (1981). "Spectrum analysis—a modern perspective, *Proc. IEEE*, vol. 69, pp. 1380–1419.

KELLY, J. L, JR., and R. F. LOGAN (1970). *Self-Adaptive Echo Canceller*, U.S. Patent 3,500,000, March 10.

KELLY, E. J., I. S. REED, and W. L. ROOT (1960). "The detection of radar echoes in noise: I," *J. SIAM*, vol. 8, pp. 309–341.

KHARGONEKAR, P. P., and K. M. NAGPAL (1991). "Filtering and smoothing in an $H^{\infty}$-setting," *IEEE Trans. Autom. Control*, vol. AC-36, pp. 151–166.

KIM, M. S., and C. C. GUEST (1990). "Modification of backpropagation networks for complex-valued signal processing in frequency domain," *International Joint Conference on Neural Networks*, San Diego, Calif., vol. III, pp. 27–31.

KIMURA, H. (1984). "Robust realizability of a class of transfer functions," *IEEE Trans. Autom. Control*, vol. AC-29, pp. 788–793.

KLEMA, V. C., and A.J. LAUB (1980). "The singular value decomposition: Its computation and some applications," *IEEE Trans Autom. Control*, vol. AC-25, pp. 164–176.

KMENTA, J. (1971). *Elements of Econometrics*, Macmillan, New York.

KNIGHT, W. C., R. G. PRIDHAM, and S. M. KAY (1981). "Digital signal processing for sonar," *Proc. IEEE*, vol. 69, pp. 1451–1506.

KOH, T., and E. J. POWERS (1985). "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 1445–1455.

KOHONEN, T. (1982). "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69.

KOHONEN, T. (1990). "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480.

KOLMOGOROV, A. N. (1939). "Sur l'interpolation et extrapolation des suites stationaries," *C.R. Acad. Sci.*, Paris, vol. 208, pp. 2043–2045. [English translation reprinted in Kailath, 1977.]

KOLMOGOROV, A. N. (1968). "Three approaches to the quantitative definition of information," *Probl. Inf. Transm. USSR*, vol. 1, pp. 1–7.

KREIN, M. G. (1945). "On a problem of extrapolation of A. N. KOLMOGOROV," *C. R. (Dokl.) Akad. Nauk SSSR*, vol. 46, pp. 306–309. [Reproduced in Kailath, 1977.]

KULLBACK, S., and R. A. LEIBLER (1951). "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79–86.

KUMAR, R. (1983). "Convergence of a decision-directed adaptive equalizer," in *Proc. Conf. Decision Control*, vol. 3, pp. 1319–1324.

KUNG, H. T. (1982). "Why systolic architectures?" *Computer*, vol. 15, pp. 37–46.

KUNG, H. T., and C. E. LEISERSON (1978). "Systolic arrays (for VLSI)," *Sparse Matrix Proc. 1978, Soc. Ind. Appl. Math.*, 1978, pp. 256–282. [A version of this paper is reproduced in Mead and Conway, 1980.]

KUNG, S. Y. (1988). *VLSI Array Proqessors*, Prentice-Hall, Englewood Cliffs, N.J.

KUNG, S. Y., H. J. WHITEHOUSE, and T. KAILATH, eds. (1985). *VLSI and Modern Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

KUNG, S. Y., ET AL. (1987). "Wavefront array processors—concept to implementation," *Computer*, vol. 20, pp. 18–33.

KUSHNER, H. J. (1984). *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic System Theory*, MIT Press, Cambridge, Mass.

KUSHNER, H. J., and D. S. CLARK (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York.

KUSHNER, H. J., and J. YANG (1995). "Analysis of adaptive step size SA algorithms for parameter tracking," *IEEE Trans. Autom. Control*, Vol. 40, pp. 1403–1410.

LANDAU, I. D. (1984). "A feedback system approach to adaptive filtering," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 251–262.

LANG, K. J., and G. E. HINTON (1988). "The development of the time-delay neural network architecture for speech recognition," Technical Report, CMU-CS-88-152, Carnegie-Mellon University, Pittsburgh, Pa.

LANG, S. W., and J. H. MCCLELLAN (1979). "A simple proof of stability for all-pole linear prediction models," *Proc. IEEE*, vol. 67, pp. 860–861.

LAWRENCE, R. E., and H. KAUFMAN (1971). "The Kalman filter for the equalization of a digital communication channel," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 1137–1141.

LAWSON, C. L., and R. J. HANSON (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N.J.

LEE, D. T. L. (1980). "Canonical ladder form realizations and fast estimation algorithms," Ph.D. dissertation, Stanford University, Stanford, Calif.

LEE, D. T. L., M. MORF, and B. FRIEDLANDER (1981). "Recursive least-squares ladder estimation algorithms," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 467–481.

LEE, J. C., and C. K. UN (1986). "Performance of transform-domain LMS adaptive algorithms," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 499–510.

LEGENDRE, A. M. (1810). "Méthode des moindres quarrés, pour trouver le milieu le plus probable entre les résultats de différentes observations," *Mem. Inst. France*, pp. 149–154.

LEHMER, D. H. (1961). "A machine method for solving polynomial equations," *J. Assoc. Comput. Mach.*, vol. 8, pp. 151–162.

LEUNG, H., and S. HAYKIN (1989). "Stability of recursive QRD-LS algorithms using finite-precision systolic array implementation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-37, pp. 760–763.

LEUNG, H., and S. HAYKIN (1991). "The complex backpropagation algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-39, pp. 2101–2104.

LEV-ARI, H., T. KAILATH, and J. CIOFFI (1984). "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Information Theory*, vol. IT-30, pp. 222–236.

LEVIN, M. D., and C. F. N. COWAN (1994). "The performance of eight recursive least squares adaptive filtering algorithms in a limited precision environment," in *Proc. European Signal Process. Conf.*, Edinburgh, Scotland, pp. 1261–1264.

LEVINSON, N. (1947). "The Wiener RMS (root-mean-square) error criterion in filter design and prediction," *J. Math Phys.*, vol. 25, pp. 261–278.

LEVINSON, N., and R. M. REDHEFFER (1970). *Complex Variables*, Holden-Day, San Francisco.

LEWIS, A. (1992). "Adaptive filtering-applications in telephony," *BT Technol. J.*, vol. 10, pp. 49–63.

LI, Y., and Z. DING (1995). "Convergence analysis of finite length blind adaptive equalizers," *IEEE Trans. Signal Process.*, vol. 43, pp. 2120–2129.

LIAPUNOV, A. M. (1966). *Stability of Motion*, trans. F. Abramovici and M. Shimshoni, Academic Press, New York.

LII, K. S. and M. ROSENBLATT (1982). "Deconvolution and estimation of transfer function phase and coefficients for non-Gaussian linear processes," *Ann. Stat.*, vol. 10, pp. 1195–1208.

LILES, W. C., J. W. DEMMEL, and L. E. BRENNAN (1980). Gram–Schmidt Adaptive Algorithms, Tech. Rep. RADC-TR-79-319, RADC, Griffiss Air Force Base, N. Y.

LIN, D. W. (1984). "On digital implementation of the fast Kalman algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-32, pp. 998–1005.

LING, F. (1989). "Efficient least-squares lattice algorithms based on Givens rotation with systolic array implementations," in *Proc. ICASSP*, Glasgow, Scotland, pp. 1290–1293.

LING, F., 1991. "Givens rotation based least-squares lattice and related algorithms," *IEEE Trans. Signal Process.*, vol. 39, pp. 1541–1551.

LING, F., and J. G. PROAKIS (1984a). "Numerical accuracy and stability: Two problems of adaptive estimation algorithms caused by round-off error," in *Proc. ICASSP*, San Diego, Calif., pp. 30.3.1–30.3.4.

LING, F., and J. G. PROAKIS (1984b). "Nonstationary learning characteristics of least squares adaptive estimation algorithms," in *Proc. ICASSP*, San Diego, Calif., pp. 3.7.1–3.7.4.

LING, F., and J. G. PROAKIS (1986). "A recursive modified Gram–Schmidt algorithm with applications to least squares estimation and adaptive filtering," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 829–836.

LING, F., D. MANOLAKIS, and J. G. PROAKIS (1985). "New forms of LS lattice algorithms and an analysis of their round-off error characteristics," in *Proc. ICASSP*, Tampa, Fla., pp. 1739–1742.

LING, F., D. MANOLAKIS, and J. G. PROAKIS, (1986). "Numerically robust least-squares lattice-ladder algorithm with direct updating of the reflection coefficients," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 837–845.

LIPPMANN, R. P. (1987). "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, pp. 4–22.

LITTLE, G. R., S. C. GUSTAFSON, and R. A. SENN (1990). "Generalization of the backpropagation neural network learning algorithm to permit complex weights," *Appl. Opt.*, vol. 29, pp. 1591–1592.

LIU, K. J. R., S.-F. HSIEH, and K. YAO (1992). "Systolic block Householder transformation for RLS algorithm with two-level pipelined implementation," *IEEE Trans. Signal Process.*, vol. 40, pp. 946–958.

LIU, Z.-S. (1995). "QR methods of O(N) complexity in adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 43, pp. 720–729.

LJUNG, L. (1977). "Analysis of recursive stochastic algorithms," *IEEE Trans. Autom. Control.*, vol. AC-22, pp. 551–575.

LJUNG, L. (1984). "Analysis of stochastic gradient algorithms for linear regression problems," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 151–160.

LJUNG, L. (1987). *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, N.J.

LJUNG, L., and T. SÖDERSTRÖM (1983). *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Mass.

LJUNG, L., M. MORF, and D. FALCONER (1978). "Fast calculation of gain matrices for recursive estimation schemes," *Int. J. Control*, vol. 27, pp. 1–19.

LJUNG, L, and S. GUNNARSSON (1990). "Adaptation and tracking in system identification—A survey," *Automatica*, vol. 26, pp. 7–21.

LJUNG, S., and L. LJUNG (1985). "Error propagation properties of recursive least-squares adaptation algorithms," *Automatica*, vol. 21, pp. 157–167.

LORD, RAYLEIGH. (1879). "Investigations in optics with special reference to the spectral scope," *Philos. Mag.*, vol. 8, pp. 261–274.

LORENZ, H., G. M. RICHTER, M. CAPACCIOLI, and G. LONGO (1993). "Adaptive filtering in astronomical image processing. I. Basic considerations and examples," *Astron. Astrophys.*, vol. 277, pp. 321–330.

LOWE, D. (1989). "Adaptive radial basis function nonlinearities and the problem of generalization," in *First IEE Int. Conf. Artif. Neural Networks*, London, pp. 171–175.

LUCKY, R. W. (1965). "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, pp. 547–588.

LUCKY, R. W. (1966). "Techniques for adaptive equalization of digital communication systems," *Bell Syst. Tech. J.*, vol. 45, pp. 255–286.

LUCKY, R. W. (1973). "A survey of the communication literature: 1968–1973," *IEEE Trans. Information Theory*, vol. IT-19, pp. 725–739.

LUCKY, R. W., J. SALZ, and E. J. WELDON, JR. (1968). *Principles of Data Communication*, McGraw-Hill, New York.

LUENBERGER, D. G. (1969). *Optimization by Vector Space Methods*, Wiley, New York.

LUK, F. T. (1986). "A triangular processor array for computing singular values," *Linear Algebra Applications*, vol. 77, pp. 259–273.

LUK, F. T., and H. PARK (1989). "A proof of convergence for two parallel Jacobi SVD algorithms," *IEEE Trans. Comput.*, vol. 38, pp. 806–811.

LUK, F. T., and S. QIAO (1989). "Analysis of a recursive least-squares signal-processing algorithm," *SIAM J. Sci. Stat. Comput.*, vol. 10, pp. 407–418.

LYNCH, M. R., and P. J. RAYNER (1989), "The properties and implementation of the non-linear vector space connectionist model," in *Proc. First IEE Int. Conf. Artif. Neural Networks*, London, pp. 186–190.

MACCHI, O. (1986a). "Advances in Adaptive Filtering," in *Digital Communications*, ed. E. Biglieri and G. Prati, North-Holland, Amsterdam, pp. 41–57.

MACCHI, O. (1986b). "Optimization of adaptive identification for time-varying filters," *IEEE Trans. Autom. Control*, vol. AC-31, pp. 283–287.

MACCHI, O. (1995). *Adaptive Processing: The LMS Approach with Applications in Transmission*, Wiley, New York.

MACCHI, O., and N. J. BERSHAD (1991). "Adaptive recovery of a chirped sinusoid in noise, Part I: Performance of the RLS algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 39, pp. 583–594.

MACCHI, O., and M. TURKI (1992). "The nonstationarity degree: can an adaptive filter be worse than no processing?" in *Proc. IFAC International Symposium on Adaptive Systems in Control and Signal Processing*, Grenoble, France, pp. 743–747.

MACCHI, O., N. J. BERSHAD, and M. M-BOUP (1991). "Steady-state superiority of LMS over LS for time-varying line enhancer in noisy environment," *IEE Proc. (London), part F*, vol. 138, pp. 354–360.

MACCHI, O., and E. EWEDA (1984). "Convergence analysis of self-adaptive equalizers," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 161–176.

MACCHI, O., and M. JAIDANE-SAIDNE (1989). "Adaptive IIR filtering and chaotic dynamics: application to audio-frequency coding," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 591–599.

MAKHOUL, J. (1975). "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580.

MAKHOUL, J. (1977). "Stable and efficient lattice methods for linear prediction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-25, pp. 423–428.

MAKHOUL, J. (1978). "A class of all-zero lattice digital filters: properties and applications," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-26, pp. 304–314.

MAKHOUL, J. (1981). "On the eigenvectors of symmetric Toeplitz matrices," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 868–872.

MAKHOUL, J., and L. K. COSSELL (1981), "Adaptive lattice analysis of speech," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 494–499.

MANOLAKIS, D., F. LING, and J. G. PROAKIS (1987). "Efficient time-recursive least-squares algorithms for finite-memory adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 400–408.

MANSOUR, D., and A. H. GRAY, JR. (1982). "Unconstrained frequency-domain adaptive filters," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-30, pp. 726–734.

MARCOS, S., and O. MACCHI (1987). "Tracking capability of the least mean square algorithm: Application to an asynchronous echo canceller," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 1570–1578.

MARDEN, M. (1949). "The geometry of the zeros of a polynomial in a complex variable," *Am. Math. Soc. Surveys*, no. 3, chap. 10, American Mathematical Society, New York.

MARKEL, J. D., and A. H. GRAY JR. (1976). *Linear Prediction of Speech*, Springer-Verlag, New York.

MARPLE, S. L., JR. (1980). "A new autoregressive spectrum analysis algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-28, pp. 441–454.

MARPLE, S. L., JR. (1981). "Efficient least squares FIR system identification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 62–73.

MARPLE, S. L., JR. (1987). *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, N.J.

MARSHALL, D. F., W. K. JENKINS, and J. J. MURPHY (1989). "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 474–484.

MASON, S. J. (1956). "Feedback theory; further properties of signal flow graphs," *Proc. IRE*, vol. 44, pp. 920–926.

MATHEWS, V. J., and Z. XIE (1993). "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Process.*, vol. 41, pp. 2075–2087.

MATHIAS, R. (1995). "Accurate eigen system computation by Jacobi methods," *SIMAX*, vol. 16, pp. 977–1003.

MAYBECK, P. S. (1979). *Stochastic Models, Estimation, and Control*, vol. 1, Academic Press, New York.

MAYBECK, P. S. (1982). *Stochastic Models, Estimation, and Control*, vol. 2, Academic Press, New York.

MAZO, J. E. (1979). "On the independence theory of equalizer convergence," *Bell Syst. Tech. J.*, vol. 58, pp. 963–993.

MAZO, J. E. (1980). "Analysis of decision-directed equalizer convergence," *Bell Syst. Tech. J.*, vol. 59, pp. 1857–1876.

McCanny, J. V., and J. G. McWhirter (1987). "Some systolic array developments in the United Kingdom," *Computer*, vol. 2, pp. 51–63.

McCulloch, W. S., and W. Pitts (1943). "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133.

McCool, J. M., et al. (1980). *Adaptive Line Enhancer*, U.S. Patent 4,238,746, December 9.

McCool, J. M., et al. (1981). *An Adaptive Detector*, U.S. Patent 4,243,935, January 6.

McDonald, R. A. (1966). "Signal-to-noise performance and idle channel performance of differential pulse code modulation systems with particular applications to voice signals," *Bell Syst. Tech. J.*, vol. 45, pp. 1123–1151.

McGee, W. F. (1971). "Complex Gaussian noise moments," *IEEE Trans. Information Theory*, vol. IT-17, pp. 149–157.

McLachlan, G. J., and K. E. Basford (1988). *Mixture Models: Inference and Applications to Clustering*, Dekker, New York.

McWhirter, J. G. (1983). "Recursive least-squares minimization using a systolic array," *Proc. SPIE, Real-Time Signal Processing VI*, vol. 431, San Diego, Calif., pp. 105–112.

McWhirter, J. G. (1989). "Algorithmic engineering-an emerging technology," *Proc. SPIE, Real-Time Signal Processing VI*, vol. 1152, San Diego, Calif.

McWhirter, J. G., and I. K. Proudler (1993). "The QR family," in *Adaptive System Identification and Signal Processing Algorithms*, ed. N. Kalouptsidis and S. Theodoridis, pp. 260–321, Prentice-Hall, Englewood Cliffs, N.J.

McWhirter, J. G., and T. J. Shepherd (1989). "Systolic array processor for MVDR beamforming," *IEE Proc. (London)*, part F, vol. 136, pp. 75–80.

Mead, C., and L. Conway (1980). *Introduction to VLSI Systems*, Addison-Wesley, Reading, Mass.

Medaugh, R. S., and L. J. Griffiths (1981). "A comparison of two linear predictors," in *Proc. ICASSP*, Atlanta, Ga., pp. 293–296.

Mehra, R. K. (1972). "Approaches to adaptive filtering," *IEEE Trans. Autom. Control*, vol. AC-17, pp. 693–698.

Mendel, J. M. (1973). *Discrete Techniques of Parameter Estimation: The Equation Error Formulation*, Dekker, New York.

Mendel, J. M. (1974). "Gradient estimation algorithms for equation error formulations," *IEEE Trans. Autom. Control*, vol. AC-19, pp. 820–824.

Mendel, J. M. (1986). "Some modeling problems in reflection seismology," *IEEE ASSP Mag.*, vol. 3, pp. 4–17.

Mendel, J. M. (1990a). *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*, Springer-Verlag, New York.

Mendel, J. M. (1990b). "Introduction," *IEEE Trans. Autom. Control*, vol. AC-35, Special Issue on Higher Order Statistics in System Theory and Signal Processing, p. 3.

Mendel, J. M. (1995). *Lessons in Digital Estimation Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, N.J.

Meriam, K. A. et al. (1995). "Prediction error methods for time-domain blind identification of multichannel FIR filters," in *Proc. ICASSP*, Detroit, Michigan, vol. 3, pp. 1968–1971.

Mermoz, H. F. (1981). "Spatial processing beyond adaptive beamforming," *J. Acoust. Soc. Am.*, vol. 70, pp. 74–79.

MESSERCHMITT, D. G. (1984). "Echo cancellation in speech and data transmission," *IEEE J. Sel. Areas Commun.*, vol. SAC-2, pp. 283–297.

METFORD, P. A. S., and S. Haykin (1985). "Experimental analysis of an innovations-based detection algorithm for surveillance radar," *IEE Proc. (London)*, vol. 132, part F, pp. 18–26.

MIDDLETON, D. (1960). *An Introduction to Statistical Communication Theory*, McGraw-Hill, New York.

MILLER, K. S. (1974). *Complex Stochastic Processes: An Introduction to Theory and Application*, Addison-Wesley, Reading, Mass.

MINSKY, M. L., and S. A. PAPPERT (1969). *Perceptrons*, MIT Press, Cambridge, Mass.

MONSEN, P. (1971). "Feedback equalization for fading dispersive channels," *IEEE Trans. Information Theory*, vol. IT-17, pp. 56–64.

MONZINGO, R. A., and T. W. MILLER (1980). *Introduction to Adaptive Arrays*, Wiley-Interscience, New York.

MOODY, J. E., and C. J. DARKEN (1989). "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294.

MOONEN, M., and J. VANDEWALLE (1990). "Recursive least squares with stabilized inverse factorization," *Signal Process.*, vol. 21, pp. 1–15.

MORF, M. (1974). "Fast algorithms for multivariable systems," Ph.D. dissertation, Stanford University, Stanford, Calif.

MORF, M., and D. T. LEE (1978). "Recursive least squares ladder forms for fast parameter tracking," in *Proc. 1978 Conf. Decision Control*, San Diego, Calif., pp. 1362–1367.

MORF, M., and T. KAILATH (1975). "Square-root algorithms for least-squares estimation," *IEEE Trans. Autom. Control*, vol. AC-20, pp. 487–497.

MORF, M., T. KAILATH, and L. LJUNG (1976). "Fast algorithms for recursive identification," in *Proc. 1976 Conf. Decision Control*, Clearwater Beach, Fla., pp. 916–921.

MORF, M., A. VIEIRA, and D. T. LEE (1977). "Ladder forms for identification and speech processing," in *Proc. 1977 IEEE Conf. Decision Control*, New Orleans, pp. 1074–1078.

MORONEY, P. (1983). *Issues in the Implementation of Digital Feedback Compensators*, MIT Press, Cambridge, Mass.

MOROZOV, V. A. (1993). *Regularization Methods for Ill-posed Problems*, CRC Press, Boca Raton, Fla.

MORSE, P. M., and H. FESHBACK (1953). *Methods of Theoretical Physics*, Pt. I, McGraw-Hill, New York.

MOSCHNER, J. L. (1970). Adaptive Filter with Clipped Input Data, Tech. Rep. 6796-1, Stanford University Center for Systems Research, Stanford, Calif.

MOULINES, E., P. DUHAMEL, J.-F. CARDOSO, and S. MAYRARGUE (1995). "Subspace methods for blind identification of multichannel FIR filters," *IEEE Trans. Signal Process.*, vol. 43, pp. 516–525.

MUELLER, M. S. (1981a). Least-squares algorithms for adaptive equalizers," *Bell Syst. Tech. J.*, vol. 60, pp. 1905–1925.

MUELLER, M. S. (1981b). "On the rapid initial convergence of least-squares equalizer adjustment algorithms," *Bell Syst. Tech. J.*, vol. 60, pp. 2345–2358.

MUIRHEAD, R. J. (1982). *Aspects of Multivariate Statistical Theory*, Wiley, New York.

MULGREW, B. (1987). "Kalman filter techniques in adaptive filtering," *IEE Proc. (London)*, part F, vol. 134, pp. 239–243.

MULGREW, B., and C. F. N. COWAN (1987). "An adaptive Kalman equalizer: structure and performance," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 1727–1735.

MULGREW, B., and C. F. N. COWAN (1988). *Adaptive Filters and Equalizers*, Kluwer, Boston, Mass.

MURANO, K., ET AL. (1990). "Echo cancellation and applications," *IEEE Commun.*, vol. 28, pp. 49–55.

MUSICUS, B. R. (1985). "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 1333–1335.

NAGUMO, J. I., and A. NODA (1967). "A learning method for system identification," *IEEE Trans. Autom. Control*, vol. AC-12, pp. 282–287.

NAHI, N. E. (1969). *Estimation Theory and Applications*, Wiley, New York.

NARAYAN, S. S., A. M. PETERSON, and M. J. NARASHIMA (1983). "Transform domain LMS algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 609–615.

NARENDRA, K. S., and A. M. ANNASWAMY (1989). *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, N.J.

NARENDRA, K. S., and K. PARTHASARATHY (1990). "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27.

NAU, R. F., and R. M. OLIVER (1979). "Adaptive filtering revisited," *J. Oper. Res. Soc.*, vol. 30, pp. 825–831.

NIELSEN, P. A., and J. B. THOMAS (1988). "Effect of correlation on signal detection in arctic under-ice noise," in Conf. Rec. Twenty-Second Asilomar Conference on Signals, Systems and Computers," Pacific Grove, Calif., pp. 445–450.

NIKIAS, C. L. (1991). "Higher-order spectral analysis," in *Advances in Spectrum Analysis and Array Processing*, vol. 1, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J.

NIKIAS, C. L., and M. R. RAGHUVEER (1987). "Bispectrum estimation: A digital signal processing framework," *Proc. IEEE*, vol. 75, pp. 869–891.

NISHITANI, T., ET AL. (1987). "A CCITT standard 32 kbits/s ADPCM LSI codec," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 219–225.

NORMILE, J. O. (1983). "Adaptive filtering with finite wordlength constraints," *IEE Proc. (London)*, part E vol. 130, pp. 42–46.

NORTH, D. O. (1963). "An analysis of the factors which determine signal/noise discrimination in pulsed carrier systems," *Proc. IEEE*, vol. 51, pp. 1016–1027.

NORTH, R. C., J. R. ZEIDLER, W. H. KU, and T. R. ALBERT, 1993. "A floating-point arithmetic error analysis of direct and indirect coefficient updating techniques for adaptive lattice filters," *IEEE Trans. Signal Process.*, vol. 41, pp. 1809–1823.

NUTTAL, A. H. (1976). Spectral Analysis of a Univariate Process with Bad Data Points via Maximum Entropy and Linear Predictive Techniques, Naval Underwater Systems Center (NUSC) Scientific and Engineering Studies, New London, Conn.

OPPENHEIM, A. V., and J. S. LIM (1981). "The importance of phase in signals," *Proc. IEEE*, vol. 69, pp. 529–541.

OPPENHEIM, A. V., and R. W. SCHAFER (1989). *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

OWSLEY, N. L. (1973). "A recent trend in adaptive spatial processing for sensor arrays: constrained adaptation," in *Signal Processing*, ed. J. W. R. Griffiths et al., Academic Press, New York, pp. 591–604.

OWSLEY, N. L. (1985). "Sonar array processing," in *Array Signal Processing*, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J., pp. 115–193.

PALMIERI, F., and S. A. SHAH (1990). "Fast training of multi-layer perceptrons using multilinear parameterization," in *International Joint Conference on Neural Networks*, Washington, D.C., vol. 1, pp. 696–699.

PAN, C. T., and R. J. PLEMMONS (1989). "Least squares modifications with inverse factorizations: Parallel implications," *J. Comput. Appl. Math.*, vol. 27, pp. 109–127.

PAN, R., and C. L. NIKIAS (1988). "The complex cepstrum of higher order cumulants and nonminimum phase identification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-36, pp. 186–205.

PANDA, G., B. MULGREW, C. F. N. COWAN, and P. M. GRANT (1986). "A self-orthogonalizing efficient block adaptive filter," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 1573–1582.

PAPADIAS, C. (1995). "Methods for blind equalization and identification of linear channels," Ph. D. thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France.

PAPOULIS, A. (1984). *Probability, Random Variables, and Stochastic Processes*, 2nd ed., McGraw-Hill, New York.

PARK, J., and I. W. SANDBERG (1991). "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246–257.

PARLETT, B. N. (1980). *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J.

PARZEN, E. (1962). "On the estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076.

PATRA, J. C., and G. PANDA (1992). "Performance evaluation of finite precision LMS adaptive filters using probability density approach," *J. Inst. Electron. Telecommun. Eng.*, vol. 38, pp. 192–195.

PEACOCK, K. L., and S. TREITEL (1969). "Predictive deconvolution: theory and practice," *Geophysics*, vol. 34, pp. 155–169.

PERRIER, A., B. DELYON, and E. MOULINES (1994). "On the validity of the independence assumption for stochastic gradient identification algorithm," submitted for publication.

PETRAGLIA, M. R., and S. K. MITRA, 1993. "Performance analysis of adaptive filter structures based on subband decomposition," in *Proceedings of International Symposium on Circuits and Systems*, pp. I.60–I.63, Chicago, Illinois.

PICCHI, G., and G. PRATI (1984). "Self-orthogonalizing adaptive equalization in the discrete frequency domain," *IEEE Trans. Commun.*, vol. COM-32, pp. 371–379.

PICCHI, G., and G. PRATI (1987). "Blind equalization and carrier recovery using a 'stop-and-go' decision-directed algorithm," *IEEE Trans. Commun.*, vol. COM-35, pp. 877–887.

PLACKETT, R. L. (1950). "Some theorems in least squares," *Biometrika*, vol. 37, p. 149.

POGGIO, T., and F. GIROSI (1990). "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497.

PORAT, B., and T. KAILATH (1983). "Normalized lattice algorithms for least-squares FIR system identification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 122–128.

PORAT, B., B. FRIEDLANDER, and M. MORF (1982). "Square root covariance ladder algorithms," *IEEE Trans. Autom. Control*, vol. AC-27, pp. 813–829.

POTTER, J. E. (1963). "New Statistical Formulas," Instrumentation Laboratory, MIT, Cambridge, Mass., Space Guidance Analysis Memo No. 40.

POWELL, M. J. D., 1992. "The theory of radial basis function approximation in 1990," in *Advances in Numerical Analysis*, Vol. II: *Wavelets, Subdivision Algorithms, and Radial Basis Functions*. ed. W. Light, pp. 105–210, Oxford Science Publications, Oxford, United Kingdom.

PRESS, W. H., ET AL. (1988). *Numerical Recipes in C*, Cambridge University Press, Cambridge, United Kingdom.

PRIESTLEY, M. B. (1981). *Spectral Analysis and Time Series*, vols. 1 and 2, Academic Press, New York.

PROAKIS, J. G. (1975). "Advances in equalization for intersymbol interference," in *Advances in Communication Systems*, ed. A. V. Balakrishnan, vol. 4, Academic Press, New York, pp. 123–198.

PROAKIS, J. G. (1989). *Digital Communications*, 2nd ed., McGraw-Hill, New York.

PROAKIS, J. G. (1991). "Adaptive equalization for TDMA digital mobile radio," *IEEE Trans. Vehicular Technol.*, vol. 40, pp. 333–341.

PROAKIS, J. G., and J. H. MILLER (1969). "An adaptive receiver for digital signaling through channels with intersymbol interference," *IEEE Trans. Information Theory*, vol. IT-15, pp. 484–497.

PROUDLER, I. K., J. G. MCWHIRTER, and T. J. SHEPHERD (1988). "Fast QRD-based algorithms for least squares linear prediction," in *Proc. IMA Conf. Math. Signal Process.*, Warwick, England.

PROUDLER, I. K., J. G. MCWHIRTER, and T. J. SHEPHERD (1991). "Computationally efficient, QR decomposition approach to least squares adaptive filtering," *IEE Proc. (London)*, part F, vol. 138, pp. 341–353.

QURESHI, S. (1982). "Adaptive equalization," *IEEE Commun. Soc. Mag.*, vol. 20, pp. 9–16.

QURESHI, S. U. H. (1985). "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387.

RABINER, L. R., and B. GOLD (1975). *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

RABINER, L. R., and R. W. SCHAFER (1978). *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J.

RADER, C. M. (1990). "Linear systolic array for adaptive beamforming," in The 1990 Digital Signal Processing Workshop, New Paltz, N.Y., Sponsored by IEEE Signal Processing Society, pp. 5.2.1–5.2.2.

RADER, C. M., and A. O. STEINHARDT (1986). "Hyperbolic householder transformations", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. ASSP-34, pp. 1589–1602.

RALSTON, A. (1965). *A First Course in Numerical Analysis*, McGraw-Hill, New York.

RAO, C. R., (1973). *Linear Statistical Inference and its Applications*, 2nd ed., Wiley, New York.

RAO, S. K., and T. KAILATH (1986). "What is a systolic algorithm?" in *Proc. SPIE, Highly Parallel Signal Processing Architectures*, San Diego, Calif., vol. 614, pp. 34–48.

RAO, K. R., and P. YIP (1990). *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, San Diego, Calif.

RAYNER, P. J. W., and M. F. LYNCH (1989), "A new connectionist model based on a non-linear adaptive filter," in *Proc. ICASSP*, Glasgow, Scotland, pp. 1191–1194.

REDDI, S. S. (1979). "Multiple source location—A digital approach," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-15, pp. 95–105.

REDDI, S. S. (1984). "Eigenvector properties of Toeplitz matrices and their application to spectral analysis of time series," *Signal Process.*, pp. 45–56.

REDDY, V. U., B. EGARDT, and T. KAILATH (1981). "Optimized lattice-form adaptive line enhancer for a sinusoidal signal in broad-band noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 702–710.

REDDY, V. U., and A. NEHORAI (1981). "Response of adaptive line enhancer to a sinusoid in lowpass noise," *IEE Proc. (London)*, part F, vol. 128, no. 3, pp. 6–66.

REED, I. S. (1962). "On a moment theorem for complex Gaussian processes," *IRE Trans. Information Theory*, vol. IT-8, pp. 194–195.

REED, I. S., J. D. MALLET, and L. E. BRENNAN (1974). "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-10, pp. 853–863.

REEVES, A. H. (1975). "The past, present, and future of PCM," *IEEE Spectrum*, vol. 12, pp. 58–63.

REGALIA, P. A. (1992). "Numerical stability issues in fast least-squares adaptation algorithms," *Optical Engineering*, vol. 31, pp. 1144–1152.

REGALIA, P. A. (1993). "Numerical stability properties of a QR-based fast least squares algorithm," *IEEE Trans. Signal Process.*, vol. 41, pp. 2096–2109.

REGALIA, P. A. (1994). *Adaptive IIR Filtering in Signal Processing and Control*, Dekker, New York.

REGALIA, P. A., and G. BELLANGER (1991). "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Process.*, vol. 39, pp. 879–891.

RICKARD, J. T., ET AL. (1981). "A performance analysis of adaptive line enhancer-augmented spectral detectors," *IEEE Trans. Circuits Sys.*, vol. CAS-28, pp. 534–541.

RICKARD, J. T., and J. R. ZEIDLER (1979). "Second-order output statistics of the adaptive line enhancer," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-27, pp. 31–39.

RICKARD, J. T., J. R. ZEIDLER, M. J. DENTINO, and M. SHENSA (1981). "A performance analysis of adaptive line enhancer-augmented spectral detectors," *IEEE Trans. Circuits Syst.*, vol. CAS-28, no. 6, pp. 534–541.

RIDDLE, A. (1994). "Engineering software: Mathematical power tools," *IEEE Spectrum*, vol. 31, pp. 35–47, 95, November.

RISSANEN, J. (1978). "Modelling by shortest data description," *Automatica*, vol. 14, pp. 465–471.

RISSANEN, J. (1986). "Stochastic complexity and modeling," *Ann. Stat.*, vol. 14, pp. 1080–1100.

RISSANEN, J. (1989). *Stochastic complexity in statistical enquiry*, Series in Computer Science, vol. 15, World Scientific, Singapore.

ROBBINS, H., and S. MONRO (1951). "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, pp. 400–407.

ROBINSON, E. A. (1954). "Predictive decomposition of time series with applications for seismic exploration," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass.

ROBINSON, E. A. (1982). "A historical perspective of spectrum estimation," *Proc. IEEE*, vol. 70, Special Issue on Spectral Estimation, pp. 885–907.

ROBINSON, E. A. (1984). "Statistical pulse compression," *Proc. IEEE*, vol. 72, pp. 1276–1289.

ROBINSON, E. A., and T. DURRANI (1986). *Geophysical Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

ROBINSON, E. A., and S. TREITEL (1980). *Geophysical Signal Analysis*, Prentice-Hall, Englewood Cliffs, N.J.

ROSENBLATT, M. (1985). *Stationary Sequences and Random Fields*, Birkhäuser, Stuttgart.

ROSENBROCK, H. H., 1970. *State-space and Multivariable Theory*, Wiley, New York.

Ross, F. J. (1989). "Blind equalization for digital microwave radio," Masters thesis, McMaster University, Hamilton, Ontario, Canada.

Rumelhart, D. E., and J. L. McClelland, eds. (1986). *Parallel Distributed Processing*, vol. 1. Foundations, MIT Press, Cambridge, Mass.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning representations by back-propogating errors," *Nature*, vol. 323, pp. 533–536.

Saito, S., and F. Itakura (1966). *The Theoretical Consideration of Statistically Optimum Methods for Speech Spectral Density*, Rep. 3107, Electrical Communication Laboratory, N. T. T., Tokyo (in Japanese).

Sakrison, D. (1966). "Stochastic approximation: a recursive method for solving regression problems," in *Advances in Communication Systems*, vol. 2, ed. A. V. Balakrishnan, pp. 51–106, Academic Press, New York.

Sambur, M. R. (1978). "Adaptive noise cancelling for speech signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-26, pp. 419–423.

Samson, C. (1982). "A unified treatment of fast Kalman algorithms for identification," *Int. J. Control*, vol. 35, pp. 909–934.

Sanders, J. A., and F. Verhulst (1985). *Averaging Methods in Nonlinear Dynamical Systems*, Springer-Verlag, New York.

Sari, H. (1992). "Adaptive equalization of digital line-of-sight radio systems," in *Adaptive Systems in Control and Signal Processing 1992*, ed. L. Dugard, M. M'Saad, and I. D. Landau, Pergamon Press, Oxford, United Kingdom, pp. 505–510.

Sato, Y. (1975). "Two extensional applications of the zero-forcing equalization method," *IEEE Trans. Commun.*, vol. COM-23, pp. 684–687.

Sato, Y., 1994. "Blind equalization and blind sequence estimation," *IEICE Trans Commun.*, vol. E77-B, pp. 545–556.

Satorius, E. H., and S. T. Alexander (1979). "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899–905.

Satorius, E.H., and J. D. Pack (1981). "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-29, pp. 136–142.

Satorius, E. H., et al. (1983), "Fixed-point implementation of adaptive digital filters," in *Proc. ICAASP*, Boston, Mass., pp. 33–36.

Sayed, A. H., and T. Kailath (1994). "A state-space approach to adaptive RLS filtering," *IEEE Signal Process. Mag.*, vol. 11, pp. 18–60.

Sayed, A. H., and M. Rupp, 1994. "Local and global optimality criteria for gradient-type algorithms," *IEEE Trans. Signal Process.* (submitted).

Scharf, L. L., and D. W. Tufts (1987). "Rank reduction for modeling stationary signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 350–355.

Scharf, L. L., and L. T. McWhorter (1994). "Quadratic estimators of the correlation matrix," in IEEE ASSP Workshop on Statistical Signal and Array Processing, Quebec City, Quebec, June 27–29.

Scharf, L. L., and J. K. Thomas (1995). "Data adaptive low rank modelling," in *National Radio Science Meeting*, Boulder, Colorado, p. 200.

Schell, S. V., and W. A. Gardner (1993). "Spatio-temporal filtering and equalization for cyclostationary signals," in *Control and Dynamic Systems*, ed. C. T. Leondes, vol. 66, Academic Press, New York, pp. 1–85.

SCHETZEN, M. (1981). "Nonlinear system modeling based on the Wiener theory," *Proc. IEEE*, vol. 69, pp. 1557–1572.

SCHMIDT, R. O. (1979). "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectral Estimation Workshop*, pp. 243–258, Griffith AFB, Rome, N.Y.

SCHMIDT, R. O. (1981). "A signal subspace approach to multiple emitter location and spectral estimation," Ph.D. dissertation, Stanford University, Stanford, Calif.

SCHREIBER, R. J. (1986). "Implementation of adaptive array algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 1038–1045.

SCHROEDER, M. R. (1966). "Vocoders: analysis and synthesis of speech," *Proc. IEEE*, vol. 54, pp. 720–734.

SCHROEDER, M. R. (1985). "Linear predictive coding of speech: review and current directions," *IEEE Commun. Mag.*, vol. 23, pp. 54–61.

SCHUR, I. (1917). "Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind," *J. Reine Angew. Math.*, vol. 147, pp. 205–232; vol. 148, pp. 122–145.

SCHUSTER, A. (1898). "On the investigation of hidden periodicities with applications to a supposed 26-day period of meterological phenomena," *Terr. Magn. Atmos. Electr.*, vol. 3, pp. 13–41.

SCHWARTZ, G. (1978). "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461–464.

SCHWARTZ, L. (1967). *Cours d'Analyse*, vol. II. Hermann, Paris, pp. 271–278.

SENNE, K. D. (1968). *Adaptive Linear Discrete-Time Estimation*, Tech. Rep., 6778-5, Stanford University Center for Systems Research, Stanford, Calif.

SETHARES, W. A. (1993). "The least mean square family," in *Adaptive System Identification and Signal Processing Algorithms*, ed. N. Kalouptsidis and S. Theodoridis, pp. 84–122, Prentice-Hall, Englewood Cliffs, N.J.

SETHARES, W. A., D. A. LAWRENCE, C. R. JOHNSON, JR., and R. R. BITMEAD (1986). "Parameter drift in LMS adaptive filters," *IEEE Trans. Acous. Speech Signal Process.*, vol. ASSP-34, pp. 868–879.

SHALVI, O., and E. WEINSTEIN (1990). "New criteria for blind equalization of non-minimum phase systems (channels)," *IEEE Trans. Inf. Theory*, vol. 36, pp. 312–321.

SHAN, T.-J., and T. KAILATH (1985). "Adaptive beamforming for coherent signals and interference," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 527–536.

SHANBHAG, N. R., and K. K. PARHI, 1994. *Pipelined Adaptive Digital Filters*, Kluwer, Boston, Mass.

SHANNON, C. E. (1948). "The mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656.

SHARPE, S. M., and L. W. NOLTE (1981). "Adaptive MSE estimation," in *Proc. ICASSP*, Atlanta, Ga., pp. 518–521.

SHENSA, M. J. (1980). "Non-Wiener solutions of the adaptive noise canceller with a noisy reference," *IEEE Trans. Acoust. Speech Signal Process.* vol. ASSP-28, pp. 468–473.

SHEPHERD, T. J., and J. G. MCWHIRTER (1993). "Systolic adaptive beamforming," in *Radar Array Processing*, ed. S. Haykin, J. Litva, and T. J. Shepherd, pp. 153–243, Springer-Verlag, New York.

SHERWOOD, D. T., and N. J. BERSHAD (1987). "Quantization effects in the complex LMS adaptive algorithm: linearization using dither-theory," *IEEE Trans. Circuits Systems*, vol. CAS-34, pp. 848–854.

SHI, K. H., and F. KOZIN (1986). "On almost sure convergence of adaptive algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, pp. 471–474.

SHICHOR, E. (1982). "Fast recursive estimation using the lattice structure," *Bell Syst. Tech. J.*, vol. 61, pp. 97–115.

SHYNK, J. J. (1989). "Adaptive IIR filtering," *IEEE ASSP Mag.*, vol. 6, pp. 4–21.

SHYNK, J. J. (1992). "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, vol. 9, no. 1, pp. 14–37.

SHYNK, J. J., R. P. GOOCH, G. KRISHNAMURTHY, and C. K. CHAN (1991). "A comparative performance study of several blind equalization algorithms," in *Proc. SPIE, Adaptive Signal Processing*, vol. 1565, pp. 102–117, San Diego, Calif.

SIBUL, L. H. (1984). "Application of singular value decomposition to adaptive beamforming," in *Proc. ICASSP*, San Diego, Calif., vol. 2, pp. 32.11/1–4.

SICURANZA, G. L. (1985). "Nonlinear digital filter realization by distributed arithmetic," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp 939–945.

SICURANZA, G. L., and G. RAMPONI (1986). "Adaptive nonlinear digital filters using distributed arithmetic," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 518–526.

SINGH, S. P., ed. (1992). *Approximation Theory, Spline Functions and Applications*, Kluwer, The Netherlands.

SKOLNIK, M. I. (1982). *Introduction to Radar Systems*, 2nd ed., McGraw-Hill, New York.

SLEPIAN, D. (1978). "Prolate spheroidal wave functions, Fourier analysis, and uncertainty-V: The discrete case," *Bell Syst. Tech. J.*, vol. 57, pp. 1371–1430.

SLOCK, D. T. M. (1989). "Fast algorithms for fixed-order recursive least-squares parameter estimation," Ph.D. dissertation, Stanford University, Stanford, Calif.

SLOCK, D. T. M. (1992). "The backward consistency concept and roundoff error propagation dynamics in RLS algorithms," *Optical Engineering*, vol. 31, pp. 1153–1169.

SLOCK, D. T. M. (1994). "Blind fractionally-spaced equalization, perfect-reconstruction filter banks and multichannel linear prediction," in *Proc. ICASSP*, Adelaide, Australia, vol. 4, pp. 585–588.

SLOCK, D. T. M., and T. KAILATH (1991). "Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Trans. Signal Process.* vol. 39, pp. 92–114.

SLOCK, D. T. M., and T. KAILATH (1993). "Fast transversal RLS algorithms," in *Adaptive System Identification and Signal Processing Algorithms*, eds. N. Kalouptsidis and S. Theodoridis, pp. 123–190, Prentice-Hall, Englewood Cliffs, N.J.

SLOCK, D. T. M., and C. B. PAPADIAS (1995). "Further results on blind identification and equalization of multiple FIR channels," in *Proc. ICASSP*, Detroit, Michigan, vol. 3, pp. 1964–1967.

SOLO, V. (1989). "The limiting behavior of LMS," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, pp. 1909–1922.

SOLO, V. (1992). "The error variance of LMS with time-varying weights," *IEEE Trans. Signal Process.*, vol. 40, pp. 803–813.

SOLO, V., and X. KONG (1995). *Adaptive Signal Processing Algorithms*, Prentice-Hall, Englewood Cliffs, N.J.

SOMMEN, P. C. W., and J. A. K. S. JAYASINGHE (1988). "On frequency-domain adaptive filters using the overlap-add method," in *Proc. IEEE Int. Symp. Circuits Systems*, Espoo, Finland, pp. 27–30.

SOMMEN, P. C. W., P. J. VAN GERWEN, H. J. KOTMANS, and A. E. J. M. JANSEN (1987). "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 788–798.

SONDHI, M. M., (1967). "An adaptive echo canceller," *Bell Syst. Tech. J.*, vol. 46, pp. 497–511.

SONDHI, M. M. (1970). *Closed Loop Adaptive Echo Canceller Using Generalized Filter Networks*. U.S. Patent, 3,499,999, March 10.

SONDHI, M., and D. A. BERKLEY (1980). "Silencing echoes in the telephone network," *Proc. IEEE*, vol. 68, pp. 948–963.

SONDHI, M. M., and A. J. PRESTI (1966). "A self-adaptive echo canceller," *Bell Syst. Tech. J.*, vol. 45, pp. 1851–1854.

SONI, T., J. R. ZEIDLER, and W. H. KU; 1995. "Behavior of the partial correlation coefficients of a least squares lattice filter in the presence of a nonstationary chirp input," *IEEE Trans. Signal Process.*, vol. 43, pp. 852–863.

SOO, J.-S., and K. K. PAGN (1991). "A multistep size (MSS) frequency domain adaptive filter," *IEEE Trans. Signal Process.*, vol. 39, pp. 115–121.

SUBBA RAO, T., and M. M. GABR (1980). "A test for linearity of stationary time series," *J. Time Series Analysis*, vol. 1, pp. 145–158.

SORENSON, H. W. (1967). "On the error behavior in linear minimum variance estimation problems," *IEEE Trans. Autom. Control*, vol. AC-12, pp. 557–562.

SORENSON, H. W. (1970). "Least-squares estimation: from Gauss to Kalman," *IEEE Spectrum*, vol. 7, pp. 63–68.

SORENSON, H. W., ed. (1985). Kalman Filtering: Theory and Application, IEEE Press, New York.

Special Issue on Adaptive Antennas (1976). *IEEE Trans. Antennas Propaga.*, vol. AP-24, September.

Special Issue on Adaptive Arrays (1983). *IEE Proc. Commun. Radar Signal Process.*, London, vol. 130, pp. 1–151.

Special Issue on Adaptive Filters (1987). *IEE Proc. Commun. Radar Signal Process.*, London, vol. 134, pt. F.

Special Issue on Adaptive Processing Antenna Systems (1986). *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 273–462.

Special Issue on Adaptive Signal Processing (1981). *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 465–602.

Special Issue on Adaptive Systems (1976). *Proc. IEEE*, vol. 64, pp. 1123–1240.

Special Issue on Adaptive Systems and Applications (1987). *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 705–854.

Special Issue on Higher Order Statistics in System Theory and Signal Processing (1990). *IEEE Trans. Autom. Control*, vol. AC-35, pp. 1–56.

Special Issue on Linear Adaptive Filtering (1984). *IEEE Trans. Information Theory*, vol. IT-30, pp. 131–295.

Special Issue on Linear-Quadratic-Gaussian Problem (1971). *IEEE Trans. Autom. Control*, vol. AC-16, December.

Special Issue on Neural Networks (1990). *Proc. IEEE*, vol. 78: Neural Nets I, September; Neural Nets II, October.

Special Issue on Spectral Estimation (1982). *Proc. IEEE*, vol. 70, pp. 883–1125.

Special Issue on System Identification and Time-series Analysis (1974). *IEEE Trans. Autom. Control*, vol. AC-19, pp. 638–951.

Special Issue on Systolic Arrays (1987). *Computer*, vol. 20, No. 7.

SPECHT, D. F. (1990). "Probabilistic neural networks and the polynomial Adaline as complementary techniques for classification," *IEEE Trans. Neural Networks*, vol. 1, pp. 111–121.

STEINHARDT, A. O. (1988). "Householder transforms in signal processing," *IEEE ASSP Mag.*, vol. 5, pp. 4–12.

STEWART, G. W. (1973). *Introduction to Matrix Computations*, Academic Press, New York.

STEWART, R. W., and R. CHAPMAN (1990). "Fast stable Kalman filter algorithms utilizing the square root," in *Proc. ICASSP*, Albuquerque, N. Mexico, pp. 1815–1818.

STOER, J., and BULLIRSCH (1980). *Introduction to Numerical Analysis*, Springer-Verlag, New York.

STRANG, G. (1980). *Linear Algebra and Its Applications*, 2nd ed., Academic Press, New York.

STROBACH, P. (1990). *Linear Prediction Theory*, Springer-Verlag, New York.

SUZUKI, H. (1994). "Adaptive signal processing for optimal transmission in mobile radio communications," *IEICE Trans. Communi.*, vol. E77-B, pp. 535–544.

SWAMI, A., and J. M. MENDEL (1990). "Time and lag recursive computation of cumulants from a state-space model," *IEEE Trans. Autom. Control*, vol AC-35, pp. 4–17.

SWERLING, P. (1958). A Proposed Stagewise Differential Correction Procedure for Satellite Tracking and Prediction, Rep. P-1292, Rand Corporation.

SWERLING, P. (1963). "Comment on 'A statistical optimizing navigation procedure for space flight,'" *AIAA J.*, vol. 1, p. 1968.

SZEGÖ, G. (1939). *Orthogonal polynomials*, Colloquium Publications, no. 23, American Mathematical Society, Providence, R.I.

TARRAB, M., and A. FEUER (1988). "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Information Theory*, vol. IT-34, pp. 680–691.

TETTERINGTON, D. M., A. F. M. SMITH, and U. E. MAKOV (1985). *Statistical Analysis of Finite Mixture Distributions*, Wiley, New York.

THAKOR, N. V., and Y.-S. ZHU (1991). "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection," *IEEE Trans. Biomed. Eng.*, vol. 38, pp. 785–794.

THEODORIDIS, S., C. M. S. SEE, and C. F. N. COWAN, 1992. "Nonlinear channel equalization using clustering techniques," in *ICC*, Chicago, Il., vol. 3, pp. 1277–1279.

THOMSON, D. J. (1982). "Spectral estimation and harmonic analysis," *Proc. IEEE*, vol. 70, pp. 1055–1096.

THOMSON, W. T. (1950). "Transmission of elastic waves through a stratified solid medium," *J. Appl. Phys.*, vol. 21, pp. 89–93.

TONG, L., G. XU, and T. KAILATH, 1993. "Fast blind equalization via antenna arrays," in *Proc. ICASSP*, Minneapolis, Minnesota, vol. 4. 272–275.

TONG, L., G. XU, and T. KAILATH (1994a). "Blind identification and equalization based on second-order statistics: a time-domain approach," *IEEE Trans. Information Theory*, vol. 40, pp. 340–349.

TONG, L., G. XU, and T. KAILATH (1994b). "Blind channel identification and equalization using spectral correlation measurements, Part II: A time-domain approach," in *Cyclostationarity in Communications and Signal Processing*, ed. W. A. Gardner, IEEE Press, New York, pp. 437–454.

TREICHLER, J. R. (1979). "Transient and convergent behavior of the adaptive line enhancer," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-27, pp. 53–62.

TREICHLER, J. R. and B. G. AGEE (1983). "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 459–471.

TREICHLER, J. R., and M. G. LARIMORE (1985a). "New processing techniques based on the constant modulus adaptive algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 420–431.

TREICHLER, J. R., and M. G. LARIMORE (1985b). "The tone capture properties of CMA-based interference suppressions," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 946–958.

TREICHLER, J. R., C. R. JOHNSON, JR., and M. G. LARIMORE (1987). *Theory and Design of Adaptive Filters*, Wiley-Interscience, New York.

TRETTER, S. A. (1976). *Introduction to Discrete-Time Signal Processing*, Wiley, New York.

TUGNAIT, J. K. (1994). "Testing for linearity of noisy stationary signals," *IEEE Trans. Signal Process.*, vol. 42, pp. 2742–2748.

TUGNAIT, J. K. (1995). "On fractionally-spaced blind adaptive equalization under symbol timing offsets using Godard and related equalizers," in *Proc. ICASSP*, Detroit, Michigan, vol. 3, pp. 1976–1979.

UNGERBOECK, G. (1972). "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Dev.*, vol. 16, pp. 546–555.

UNGERBOECK, G. (1976). "Fractional tap-spacing equalizer and consequences for clock recovery in data modems," *IEEE Trans. Commun.*, vol. COM-24, pp. 856–864.

ULRYCH, T. J., and R. W. CLAYTON, (1976). "Time series modelling and maximum entropy," *Phys. Earth Planet. Inter.*, vol. 12, pp. 188–200.

ULRYCH, T. J. and M. ODE (1983). "Autoregressive and mixed autoregressive-moving average models and spectra," in *Nonlinear Methods of Spectral Analysis*, ed. S. Haykin, pp. 73–125, Springer-Verlag, New York.

VAIDYANATHAN, P. P. (1987). "Quadrature mirror filter bands, M-band extensions and perfect reconstruction techniques," *IEEE ASSP Magazine*, vol. 4, pp. 4–20.

VAIDYANATHAN, P. P. (1993). *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, N.J.

VALENZUELA, R. A. (1989). "Performance of adaptive equalization for indoor radio communications," *IEEE Trans. Commun.*, vol. 37, pp. 291–293.

VAN DE KERKHOF, L. M., and W. J. W. KITZEN (1992). "Tracking of a time-varying acoustic impulse response by an adaptive filter," *IEEE Trans. Signal Process.*, vol. 40, pp. 1285–1294.

VAN DEN BOS, A. (1971). "Alternative interpretation of maximum entropy spectral analysis," *IEEE Trans. Information Theory*, vol. IT-17, pp. 493–494.

VAN HUFFEL, S., J. VANDEWALLE, and A. HAEGEMANS (1987). "An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values," *J. Comput. Appl. Math.*, vol. 19, pp. 313–330.

VAN HUFFEL, S., and J. VANDEWALLE (1988). "The partial total least squares algorithm," *J. Comput. Appl. Math.*, vol. 21, pp. 333–341.

VAN LOAN, C. (1989). "Matrix computations in signal processing," in *Selected Topics in Signal Processing*, ed. S. Haykin, Prentice-Hall, Englewood Cliffs, N.J.

VAN TREES, H. L. (1968). *Detection, Estimation and Modulation Theory*, part I, Wiley, New York.

VAN VEEN, B. (1992). "Minimum variance beamforming," in *Adaptive Radar Detection and Estimation*, ed. S. Haykin and A. Steinhardt, Wiley-Interscience, New York.

VAN VEEN, B. D., and K. M. BUCKLEY (1988). "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, pp. 4–24.

VARVITSIOTIS, A. P., S. THEODORIDIS, and G. MOUSTAKIDES (1989). "A new novel structure for adaptive LS FIR filtering based on QR decomposition," in *Proc. ICASSP*, Glasgow, Scotland, pp. 904–907.

VEMBU, S., S. VERDÚ, R. A. KENNEDY, and W. SETHARES (1994). "Convex cost functions in blind equalization," *IEEE Trans. Signal Process.*, vol. 42, pp. 1952–1960.

VERDÚ, S. (1984). "On the selection of memoryless adaptive laws for blind equalization in binary communications," in *Proc. 6th Intern. Conference on Analysis and Optimization of Systems*, Nice, France, pp. 239–249.

VERHAEGEN, M. H. (1989). "Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes," *Automatica*, vol. 25, pp. 437–444.

VERHAEGEN, M. H., and P. VAN DOOREN (1986). "Numerical aspects of different Kalman filter implementations," *IEEE Trans. Autom. Control*, vol. AC-31, pp. 907–917.

VETTERLI, M., and J. KOVAČCEVIĆ (1995). *Wavelets and Subband Coding*, Prentice-Hall, Englewood Cliffs, N.J.

VOLDER, J. E. (1959). "The CORDIC trigonometric computing technique, *IEEE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334.

WAKITA, H. (1973). "Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 417–427.

WALACH, E., and B. WIDROW (1984). "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 275–283.

WALKER, G. (1931). "On periodicity in series of related terms," *Proc. Royal Soc.*, vol. A131, pp. 518–532.

WALZMAN, T., and M. SCHWARTZ (1973). "Automatic equalization using the discrete frequency domain," *IEEE Trans. Information Theory*, vol. IT-19, pp. 59–68.

WAN, E. (1990). "Temporal backpropagation for FIR neural networks," *IEEE International Joint Conference on Neural Networks*, San Diego, Calif., vol. 1, pp. 575–580.

WARD, C. R., ET AL. (1984). "Application of a systolic array to adaptive beamforming," *IEE Proc. (London)*, pt. F, vol. 131, pp. 638–645.

WARD, C. R., P. H. HARGRAVE, and J. G. MCWHIRTER (1986). "A novel algorithm and architecture for adaptice digital beamforming," *IEEE Trans. Antennas Propag.*, vol. AP-34, pp. 338–346.

WAX, M., 1995. "Model based processing in sensor arrays," in *Advances in Spectrum Analysis and Array Processing*, vol. 3, ed. S. Haykin, pp. 1–47, Prentice-Hall, Englewood Cliffs, N. J.

WAX, M., and T. KAILATH (1985). "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, pp. 387–392.

WAX, M., and I. ZISKIND (1989). "Detection of the number of coherent signals by the MDL principle," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-37, pp. 1190–1196.

WEBB, A. R. (1994). "Functional approximation by feed-forward networks: a least-squares approach to generalisation," *IEEE Trans. Neural Networks*, vol. 6, pp. 363–371.

WEI, P., J. R. ZEIDLER, and W. H. KU (1994). "Adaptive recovery of a Doppler-shifted mobile communications signal using the RLS algorithm," in *Conf. Rec. Asilomar Conference on Signals, Systems, and Computers*, Pacific Groves, Calif., vol. 2, pp. 1180–1184.

WEIGEND, A. S., D. E. RUMELHART, and B. A. HUBERMAN (1991). "Generalization by weight elimination with application to forecasting," in *Advances in Neural Information Processing Systems 3*, pp. 875–882, Morgan Kaufman, San Mateo, Calif.

WEISBERG, S. (1980). *Applied Linear Regression*, Wiley, New York.

WEISS, A., and D. MITRA (1979). "Digital adaptive filters: conditions for convergence, rates of convergence, effects of noise and errors arising from the implementation," *IEEE Trans. Information Theory*, vol. IT-25, pp. 637–652.

WELLSTEAD, P. E., G. R. WAGNER, and J. R. CALDAS-PINTO (1987). "Two-dimensional adaptive prediction, smoothing and filtering," *IEE Proc. (London)*, part F, vol. 134, pp. 253–268.

WERBOS, P. J. (1974). "Beyond regression: new tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard University, Cambridge, Mass.

WERBOS, P. J. (1993). *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*, Wiley-Interscience, New York.

WERNER, J. J. (1983). *Control of Drift for Fractionally Spaced Equalizers*, U.S. Patent 438 4355.

WHEELWRIGHT, S. C. and S. MAKRIDAKIS. (1973). "An examination of the use of adaptive filtering in forecasting," *Oper. Res Q.*, vol. 24, pp. 55–64.

WHITTAKER, E. T., and G. N. WATSON (1965). *A Course of Modern Analysis*, Cambridge University Press, Cambridge, United Kingdom.

WHITTLE, P. (1963). "On the fitting of multivariate autoregressions and the approximate canonical factorization of a spectral density matrix," *Biometrika*, vol. 50, pp. 129–134.

WIDROW, B. (1966). *Adaptive Filters I: Fundamentals*, Rep. SEL-66-126 (TR 6764-6), Stanford Electronics Laboratories, Stanford, Calif.

WIDROW, B. (1970). "Adaptive filters," in *Aspects of Network and System Theory*, ed. R. E. Kalman and N. DeClaris, Holt, Rinehart and Winston, New York.

WIDROW, B., and M. E. HOFF, JR. (1960). "Adaptive switching circuits," *IRE WESCON Conv. Rec.*, pt. 4, pp. 96–104.

WIDROW, B. and M. LEHR (1990). "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc IEEE*, Special Issue on Neural Networks I, vol. 78, September.

WIDROW, B. and S. D. STEARNS (1985). *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

WIDROW, B., and E. WALACH (1984). "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Trans. Information Theory*, vol. IT-30, Special Issue on Linear adaptive Filtering, pp. 211–221.

WIDROW, B., et al. (1967). "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159.

WIDROW, B., J. MCCOOL, and M. BALL (1975a). "The complex LMS algorithm," *Proc. IEEE*, vol. 63, pp. 719–720.

WIDROW, B., et al. (1975b). "Adaptive noise cancelling: principles and applications," *Proc. IEEE*, vol. 63, pp. 1692–1716.

WIDROW, B., et al. (1976). "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151–1162.

WIDROW, B., K. M. DUVALL, R. P. GOOCH, and W. C. Newman (1982). "Signal cancellation phenomena in adaptive antennas: Causes and cures," *IEEE Trans. Antennas Propag.*, vol. AP-30, pp. 469–478.

WIDROW, B., et al. 1987. "Fundamental relations between the LMS algorithm and the DFT," *IEEE Trans. Circuits Syst.*, vol. CAS. 34, pp. 814–819.

WIENER, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications*, MIT Press, Cambridge, Mass. (originally issued as a classified National Defense Research Report in February 1942).

WIENER, N. (1958). *Nonlinear Problems in Random Theory*, Wiley, New York.

WIENER, N., and E. HOPF (1931). "On a class of singular integral equations," *Proc. Prussian Acad. Math-Phys. Ser.*, p. 696.

WILKINSON, J. H. (1963). *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J.

WILKINSON, J. H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, United Kingdom.

WILKINSON, J. H., and C. REINSCH, eds. (1971). *Handbook for Automatic Computation*, vol. 2, *Linear Algebra*, Springer-Verlag, New York.

WILKS, S. S. (1962). *Mathematical Statistics*, Wiley, New York.

WILLIAMS, J. R., and G. G. RICKER (1972). "Signal detectability performance of optimum Fourier receivers," *IEEE Trans. Audio and Electroacoustics*, vol. AU-20, pp. 254–270.

WILSKY, A. S. (1979). *Digital Signal Processing and Control and Estimation Theory: Points of Tangency, Areas of Intersection, and Parallel Directions*, MIT Press, Cambridge, Mass.

WOLD, H. (1938). *A Study in the Analysis of Stationary Time Series*, Almqvist and Wiksell, Uppsala, Sweden.

WOODBURY, M. (1950). Inverting Modified Matrices, Mem. Rep. 42, Statistical Research Group, Princeton University, Princeton, N.J.

WOZENCRAFT, J. M., and I. M. JACOBS (1965). *Principles of Communications Engineering*, Wiley, New York.

WYLIE, C. R., and L. C. BARRETT (1982). *Advanced Engineering Mathematics*, 5th ed., McGraw-Hill, New York.

YANG, B. (1994). "A note on the error propagation analysis of recursive least squares algorithms," *IEEE Trans. Signal Process.*, vol. 42, pp. 3523–3525.

YANG, V., and J. F. BÖHME (1992). "Rotation-based RLS algorithms: unified derivations, numerical properties and parallel implementations," *IEEE Trans. Signal Process.*, vol. 40, pp. 1151–1167.

YASSA, F. F. (1987). "Optimality in the choice of the convergence factor for gradient-based adaptive algorithms," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 48–59.

YEE, P., and S. HAYKIN (1995). "A dynamic regularized Gaussian radial basis function network for nonlinear, nonstationary time series prediction," in *Proc. ICASSP*, Detroit, Michigan, vol. 5, pp. 3419–3422.

YOGANANDAM, Y., V. U. REDDY, and T. KAILATH (1988). "Performance analysis of the adaptive line enhancer for sinusoidal signals in broad-band noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-36, pp. 1749–1757.

YOUNG, P. C. (1984). *Recursive Estimation and Time-Series Analysis*, Springer-Verlag, New York.

YUAN, J.-T., and J. A. STULLER (1995). "Least-squares order-recursive lattice smoothers," *IEEE Trans. Signal Process.*, vol. 43, pp. 1058–1067.

YULE, G. U. (1927). "On a method of investigating periodicities in disturbed series, with special reference to Wölfer's sunspot numbers," *Philos. Trans. Royal Soc. London*, vol. A226, pp. 267–298.

ZAMES, G. (1981). "Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Trans. Autom. Control*, vol. AC-26, pp. 301–320.

ZAMES, G., and B. A. FRANCIS (1983). "Feedback, minimax sensitivity, and optimal robustness," *IEEE Trans. Autom. Control*, vol. AC-28, pp. 585–601.

ZEIDLER, J. R. (1990). "Performance analysis of LMS adaptive prediction filters," *Proc. IEEE*, vol. 78, pp. 1781–1806.

ZEIDLER, J. R., E. H. SATORIUS, D. M. CHABRIES, and H. T. WEXLER (1978). "Adaptive enhancement of multiple sinusoids in uncorrelated noise," *IEEE Trans. Acous. Speech Signal Process.*, vol. ASSP-26, pp. 240–254.

ZHANG, QI-TU, and S. HAYKIN (1983). "Tracking characteristics of the Kalman filter in a nonstationary environment for adaptive filter applications," in *Proc. ICASSP*, Boston, pp. 671–674.

ZHANG, Q-T., S. HAYKIN, and P. YIP (1989). "Performance limits of the innovations-based detection algorithm," *IEEE Trans. Information Theory*, vol. IT-35, pp. 1213–1222.

ZIEGLER, R. A., and J. M. CIOFFI (1989). "A comparison of least squares and gradient adaptive equalization for multipath fading in wideband digital mobile radio," in *GLOBECOM*, vol. 1, New York, pp. 102–106.

ZIEGLER, R. A., and J. M. CIOFFI (1992). "Adaptive equalization for digital wireless data transmission," in *Virginia Tech Second Symposium on Wireless Personal Communications Proceedings*, pp. 5/1–5/12.

# Index