Introduction to Adaptive Arrays

2nd Edition

Robert A. Monzingo
Randy L. Haupt
Thomas W. Miller

SciTech Publishing, Inc.
Raleigh, NC
scitechpub.com

Published by SciTech Publishing, Inc.
911 Paverstone Drive, Suite B
Raleigh, NC 27615
(919) 847-2434, fax (919) 847-2568
scitechpublishing.com

Editor: Dudley R. Kay
Editorial Assistant: Katie Janelle
Production Manager: Robert Lawless
Typesetting: MPS Limited, a Macmillan Company
Cover Design: Brent Beckley
Printer: Sheridan Books, Inc.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN: 978-1-891121-57-9

Library of Congress Cataloging-in-Publication Data

Monzingo, Robert A., 1938-
Introduction to adaptive arrays / Robert A. Monzingo, Randy L. Haupt, Thomas W. Miller. – 2nd ed.
p. cm.
Includes bibliographical references.
ISBN 978-1-891121-57-9 (hardcover : alk. paper)
1. Antenna arrays. I. Haupt, Randy L. II. Miller, Thomas W. III. Title.
TK7871.6.M66 2011
621.382'4–dc22

2010047941

Preface

This book is intended to serve as an introduction to the subject of adaptive array sensor systems whose principal purpose is to enhance the detection and reception of certain desired signals. Array sensor systems have well-known advantages for providing flexible, rapidly configurable, beamforming and null-steering patterns. The advantages of array sensor systems are becoming more important, and this technology has found applications in the fields of communications, radar, sonar, radio astronomy, seismology and ultrasonics. The growing importance of adaptive array systems is directly related to the widespread availability of compact, inexpensive digital computers that make it possible to exploit certain well-known theoretical results from signal processing and control theory to provide the critical self-adjusting capability that forms the heart of the adaptive structure.

There are a host of textbooks that treat adaptive array systems, but few of them take the trouble to present an integrated treatment that provides the reader with the perspective to organize the available literature into easily understood parts. With the field of adaptive array sensor systems now a maturing technology, and with the applications of these systems growing more and more numerous, the need to understand the underlying principles of such systems is a paramount concern of this book. It is of course necessary to appreciate the limitations imposed by the hardware adopted to implement a design, but it is more informative to see how a choice of hardware "fits" within the theoretical framework of the overall system. Most of the contents are derived from readily available sources in the literature, although a certain amount of original material has been included.

This book is intended for use both as a textbook at the graduate level and as a reference work for engineers, scientists, and systems analysts. The material presented will be most readily understood by readers having an adequate background in antenna array theory, signal processing (communication theory and estimation theory), optimization techniques, control theory, and probability and statistics. It is not necessary, however, for the reader to have such a complete background since the text presents a step-by-step discussion of the basic theory and important techniques required in the above topics, and appropriate references are given for readers interested in pursuing these topics further. Fundamental concepts are introduced and illustrated with examples before more current developments are introduced. Problems at the end of each chapter have been chosen to illustrate and extend the material presented in the text. These extensions introduce the reader to actual adaptive array engineering problems and provide motivation for further reading of the background reference material. In this manner both students and practicing engineers may easily gain familiarity with the modern contributions that adaptive arrays have to offer practical signal reception systems.

The book is organized into three parts. Part One (Chapters 1 to 3) introduces the advantages that obtain with the use of array sensor systems, define the principal system components, and develop the optimum steady-state performance limits that any array system can theoretically achieve. This edition also includes two new topics that have practical interest: the subject of a performance index to grade the effectiveness of the overall adaptive system, and the important theme of polarization sensitive arrays. Part Two

(Chapters 4 through 9) provides the designer with a survey of adaptive algorithms and a performance summary for each algorithm type. Some important modern developments in matrix inversion computation and random search algorithms are treated. With this information available, the designer may then quickly identify those approaches most likely to lead to a successful design for the signal environment and system constrains that are of concern. Part Three (Chapters 10, 11, and 12) considers the problem of compensation for adaptive array system errors that inevitably occur in any practical system, explores the important topic of direction of arrival (DOA) estimation, and introduces current trends in adaptive array research. It is hoped that this edition succeeds in presenting this exciting field using mathematical tools that make the subject interesting, accessible, and appealing to a wide audience.

The authors would like to thank Northrop Grumman (Dennis Lowes and Dennis Fortner), the National Electronics Museum (Ralph Strong and Michael Simons), Material Systems Inc. (Rick Foster), and Remcom Inc. (Jamie Knapil Infantolino) for providing some excellent pictures.

Contents

PART I

Adaptive Array Fundamental Principles: System Uses, System Elements, Basic Concepts, and Optimum Array Processing

CHAPTER

1 Introduction

Chapter Outline

An array of sensor elements has long been an attractive solution for severe reception problems that commonly involve signal detection and estimation. The basic reason for this attractiveness is that an array offers a means of overcoming the directivity and sensitivity limitations of a single sensor, offering higher gain and narrower beamwidth than that experienced with a single element. In addition, an array has the ability to control its response based on changing conditions of the signal environment, such as direction of arrival, polarization, power level, and frequency. The advent of highly compact, inexpensive digital computers has made it possible to exploit well-known results from signal processing and control theory to provide optimization algorithms that automatically adjust the response of an adaptive array and has given rise to a new domain called "smart arrays." This self-adjusting capability renders the operation of such systems more flexible and reliable and (more importantly) offers improved reception performance that would be difficult to achieve in any other way. This revised edition acquaints the reader with the historical background of the field and presents important new developments that have occurred over the last quarter century that have improved the utility and applicability of this exciting field.

1.1 MOTIVATION FOR USING ADAPTIVE ARRAYS

An array consists of two or more sensors in which the signals are coherently combined in a way that increases the antenna's performance. Arrays have the following advantages over a single sensor:

1. Higher gain. The gain is higher, because the array gain is on the order of the number of elements in the array. Higher resolution or narrower main beam follows from the larger aperture size.
2. Electronic beam scanning. Moving large antennas to steer the main beam is slow. Arrays with phase shifters at each element are able to steer the beam without mechanical motion, because the signals are made to add in phase at the beam steering angle.
3. Low sidelobes. If the desired signal enters the main beam while interfering signals enter the sidelobes, then lowering the sidelobes relative to the main beam improves the signal to interference ratio.
4. Multiple beams. Certain array feeds allow simultaneous multiple main beams.
5. Adaptive nulling. Adaptive arrays automatically move nulls in the directions of signals over the sidelobe region.

On the other hand, these advantages are countered by the significant disadvantages of increased cost and complexity.

Conventional signal reception systems are susceptible to degradation in signal-to-noise ratio (SNR) [or more generally, the signal-to-interference plus noise ratio (SINR)] performance because of the inevitable presence in the signal environment of undesired "noise" signals that enter the sidelobes or main beam of an array. Signals that interfere with the desired signal include deliberate electronic countermeasures (ECMs), nonhostile radiofrequency interference (RFI), clutter, multipath, and natural noise sources. The resulting SNR degradation is further aggravated by array motion, poor siting, sensor failures, and a nonstationary interference environment. As traffic in the spectrum increases, the suppression of interference becomes even more paramount.

Adaptive arrays improve the reception of desired signals in the presence of interference signals in radar, sonar, seismic, and communications systems. They automatically sense the presence of interference and suppress them while simultaneously enhancing desired signal reception without prior knowledge of the signal–interference environment. Adaptive arrays are designed to complement other interference suppression techniques, such as low sidelobes, spread-spectrum techniques, and high directivity.

An adaptive array has a computer algorithm that controls the signal levels at the elements until a measure of the quality of the array performance improves. It adjusts its pattern to form nulls, to modify gain, to lower sidelobes, or to do whatever it takes to improve its performance. An adaptive array offers enhanced reliability compared with that of a conventional array. When a single sensor element in a conventional array fails, the sidelobe structure of the array pattern degrades. With an adaptive array, however, the remaining operational sensors in the array automatically adjust to restore the pattern. Adaptive arrays are more reliable than conventional arrays, because they fail gracefully. The reception pattern of an array in place on an aircraft or ship is often quite different from the array pattern measured in isolation (in an anechoic chamber) as a result of signal scattering that occurs from vehicle structures located in the vicinity of the antenna. An adaptive array often yields successful operation even when antenna patterns are severely

distorted by near-field effects. The adaptive capability overcomes any distortions that occur in the near field (i.e., at distances from the radiating antenna closer than $\lambda/2\pi$ where λ is the wavelength) and merely responds to the signal environment that results from any such distortion. Likewise, in the far field (at distances from the radiating antenna greater than 2λ) the adaptive antenna is oblivious to the absence of any distortion.

An adaptive array improves the SNR by preserving the main beam that points at the desired signal at the same time that it places nulls in the pattern to suppress interference signals. Very strong interference suppression is possible by forming pattern nulls over a narrow bandwidth. This exceptional interference suppression capability is a principal advantage of adaptive arrays compared to waveform processing techniques, which generally require a large spectrum-spreading factor to obtain comparable levels of interference suppression. Sensor arrays possessing this key automatic response capability are sometimes referred to as "smart" arrays, since they respond to far more of the signal information available at the sensor outputs than do more conventional array systems.

The capabilities provided by the adaptive array techniques to be discussed in this book offer practical solutions to the previously mentioned realistic interference problems by virtue of their ability to sort out and distinguish the various signals in the spatial domain, in the frequency domain, and in polarization. At the present time, adaptive nulling is considered to be the principal benefit of the adaptive techniques employed by adaptive array systems, and automatic cancellation of sidelobe jamming provides a valuable electronic counter–countermeasure (ECCM) capability for radar systems. Adaptive arrays are designed to incorporate more traditional capabilities such as self-focusing on receive and retrodirective transmit. In addition to automatic interference nulling and beam steering, adaptive imaging arrays may also be designed to obtain microwave images having high angular resolution. It is useful to call self-phasing or retrodirective arrays *adaptive transmitting arrays* to distinguish the principal function of such systems from an *adaptive receiving array*, the latter being the focus of this book.

1.2 HISTORICAL PERSPECTIVE

The term *adaptive antenna* was first used by Van Atta [1] and others [2] to describe a self-phasing antenna system that automatically reradiates a signal in the direction from which it was received. This "retrodirective" system works without prior knowledge of the signal direction. Retrodirective arrays in point-to-point satellite communications systems automatically return a strong pilot signal to overcome the usual beamwidth (and consequent directivity) limitations.

The development of the phase-lock loop was another major step that made possible the self-steering (or self-phasing) type of adaptive array [3]. A self-phased array has each of the array elements independently phased, based on information obtained from the received signals. For example, several large-aperture receiving antennas with slaved steering can be self-phased on received signals from satellites or space vehicles so the effective receiving aperture is the sum of the individual apertures of all participating antennas.

In the early 1960s the key capability of adaptive interference nulling was recognized and developed by Howells [4,5]. Subsequently, Applebaum established the control law associated with the Howells adaptive nulling scheme by analyzing an algorithm that maximizes a generalized SNR [6]. Concurrently, the capability of self-training or self-optimizing control was applied to adaptive arrays by Widrow and others [7-9]. The

self-optimizing control work established the least mean square (LMS) error algorithm that was based on the method of steepest descent. The Applebaum and the Widrow algorithms are very similar, and both converge toward the optimum Wiener solution.

The use of sensor arrays for sonar and radar signal reception had long been common practice by the time the early adaptive algorithm work of Applebaum and Widrow was completed [10,11]. Early work in array processing concentrated on synthesizing a "desirable" pattern. Later, attention shifted to the problem of obtaining an improvement in the SNR [12–14]. Seismic array development commenced about the same period, so papers describing applications of seismic arrays to detect remote seismic events appeared during the late 1960s [15–17].

The major area of current interest in adaptive arrays is their application to problems arising in radar and communications systems, where the designer almost invariably faces the problem of interference suppression [18]. A second example of the use of adaptive arrays is that of direction finding in severe interference environments [19,20]. Another area in which adaptive arrays are proving useful is for systems that require adaptive beamforming and scanning in situations where the array sensor elements must be organized without accurate knowledge of element location [21]. Furthermore, large, unstructured antenna array systems may employ adaptive array techniques for high angular resolution imaging [22,23]. Adaptive antennas are a subset of smart antennas and include topics such as multiple input, multiple output (MIMO) [24], element failure compensation [25], reconfigurable antennas [26], and beam switching [27].

1.3 PRINCIPAL SYSTEM ELEMENTS

Figure 1-1 shows a diagram of an adaptive array. It consists of the sensor array, the beamforming network, and the adaptive processor that adjusts the variable weights in the beamforming network. The array design depends on the propagation medium in which the array operates, the frequency spectrum of interest, and the user's knowledge of the operational signal environment.

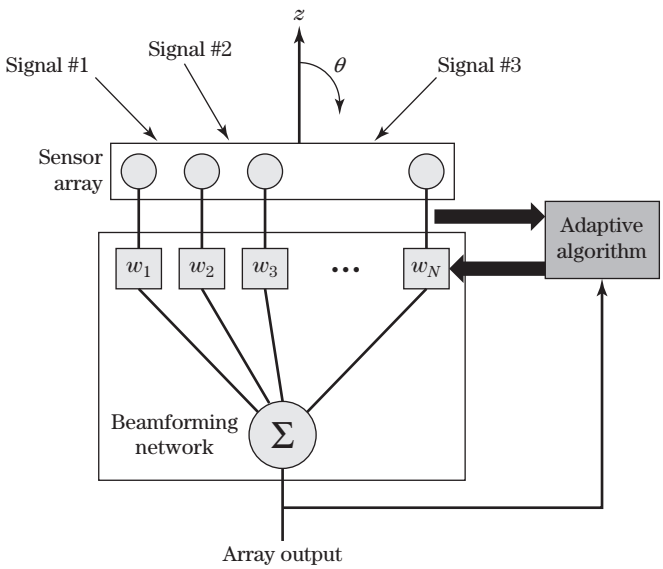


FIGURE 1-1 ■ Functional diagram of an *N*-element adaptive array.

The array consists of N sensors designed to receive (and transmit) signals in the propagation medium. The sensors are arranged to give adequate coverage (pattern gain) over a desired spatial region. The selection of the sensor elements and their physical arrangement place fundamental limitations on the ultimate capability of the adaptive array system. The output of each of the N elements goes to the beamforming network, where the output of each sensor element is first multiplied by a complex weight (having both amplitude and phase) and then summed with all other weighted sensor element outputs to form the overall adaptive array output signal. The weight values within the beamforming network (in conjunction with the sensor elements and their physical arrangement) then determine the overall array pattern. It is the ability to shape this overall array pattern that in turn determines how well the specified system requirements can be met for a given signal environment.

The exact structure of the adaptive algorithm depends on the degree of detailed information about the operational signal environment that is available to the array. As the amount of a priori knowledge (e.g., desired signal location, jammer power levels) concerning the signal environment decreases, the adaptive algorithm selected becomes critical to a successful design. Since the precise nature and direction of all signals present as well as the characteristics of the sensor elements are not known in practice, the adaptive algorithm must automatically respond to whatever signal environment (within broad limits) confronts it. If any signal environment limits are known or can reasonably be construed, such bounds are helpful in determining the adaptive processor algorithm used.

1.4 ADAPTIVE ARRAY PROBLEM STATEMENT

The fundamental problem facing the adaptive array designer is to improve the reception of a desired signal in the presence of undesired interfering signals. The terms *desired signal* and *interfering signals* imply that the characteristics of these two signal classes are different in some respect and that this difference provides the key to improving the desired signal reception. For example, if the direction of arrival of the desired signal is known (or can be deduced), then any signals arriving from different directions are suppressed by forming array pattern nulls in those directions. Likewise, if the interference signals are outside the desired bandwidth, then the interference signals are eliminated by band-pass filtering. Certain characteristics of the desired signal distinguish it from interference signals, so it is reasonable to assume that the nature of the desired signal is known even though certain signal parameters (e.g., direction of arrival, amplitude, phase) must be estimated. If the designer were solely concerned with suppressing interfering signals, then desired signal reception might suffer. Likewise, if desired signal enhancement were the sole focus of attention, then interference signal reception might also be enhanced. Therefore, the twin (and sometimes conflicting) objectives of desired signal enhancement and interference signal suppression are sought so that the overall desired signal reception performance is improved. In many cases, the overall reception performance is best measured by the output SNR. For passive sensor systems, however, the basic problem is that of determining whether a desired signal is present in a background of ambient noise and interfering signals. Determining signal presence or absence requires a decision that is not provided simply by maximizing the output SNR, and statistical decision theory provides solutions to problems of this kind that minimize the risk associated with incorrect decisions. The optimum processors prescribed by statistical decision theory are closely related to those

obtained by maximizing the output SNR, so there is an underlying unity to problems that initially appear to be quite different.

An adaptive array design includes the sensor array configuration, beamforming network implementation, signal processor, and adaptive algorithm that enables the system to meet several different requirements on its resulting performance in as simple and inexpensive a manner as possible. The system performance requirements are conveniently divided into two types: transient response and steady-state response. Transient response refers to the time required for the adaptive array to successfully adjust from the time it is turned on until reaching steady-state conditions or successfully adjusting to a change in the signal environment. Steady-state response refers to the long-term response after the weights are done changing. Steady-state measures include the shape of the overall array pattern and the output signal-to-interference plus noise ratio. Several popular performance measures are considered in detail in Chapter 3. The response speed of an adaptive array depends on the type of algorithm selected and the nature of the operational signal environment. The steady-state array response, however, can easily be formulated in terms of the complex weight settings, the signal environment, and the sensor array structure.

A fundamental trade-off exists between the rapidity of change in a nonstationary noise field and the steady-state performance of an adaptive system: generally speaking, the slower the variations in the noise environment, the better the steady-state performance of the adaptive array. Any adaptive array design needs to optimize the trade-off between the speed of adaptation and the accuracy of adaptation.

System requirements place limits on the transient response speed. In an aircraft communication system, for example, the signal modulation rate limits the fastest response speed (since if the response is too fast, the adaptive weights interact with the desired signal modulation). Responding fast enough to compensate for aircraft motion limits the slowest speed.

The weights in an adaptive array may be controlled by any one of a variety of different algorithms. The "best" algorithm for a given application is chosen on the basis of a host of factors including the signal structures, the a priori information available to the adaptive processor, the performance characteristics to be optimized, the required speed of response of the processor, the allowable circuit complexity, any device or other technological limitations, and cost-effectiveness.

Referring to Figure 1-1, the received signal impinges on the sensor array and arrives at each sensor at different times as determined by the direction of arrival of the signal and the spacing of the sensor elements. The actual received signal for many applications consists of a modulated carrier whose information-carrying component consists only of the complex envelope. If $s(t)$ denotes the modulated carrier signal, then $\tilde{s}(t)$ is commonly used to denote the complex envelope of $s(t)$ (as explained in Appendix B) and is the only quantity that conveys information. Rather than adopt complex envelope notation, however, it is simpler to assume that all signals are represented by their complex envelopes so the common carrier reference never appears explicitly. It is therefore seen that each of the N channel signals $x_k(t)$ represents the complex envelope of the output of the element of a sensor array that is composed of a signal component and a noise component, that is,

$$x_k(t) = s_k(t) + n_k(t), \quad k = 1, 2, \ldots, N \tag{1.1}$$

In a linear sensor array having equally spaced elements and assuming ideal propagation conditions, the $s_k(t)$ are determined by the direction of the desired signal. For example, if

the desired signal direction is located at an angle θ from mechanical boresight, then (for a narrowband signal)

$$s_k(t) = s(t) \exp\left\{ j \frac{2\pi k d}{\lambda} \sin\theta \right\} \tag{1.2}$$

where d is the element spacing, λ is the wavelength of the incident planar wavefront, and it is presumed that each of the sensor elements is identical.

For the beamforming network of Figure 1-1, the adaptive array output signal is written as

$$y(t) = \sum_{k=1}^{N} w_k x_k(t) \tag{1.3}$$

Equation (1.3) can be conveniently expressed in matrix notation as

$$y(t) = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w} \tag{1.4}$$

where the superscript T denotes transpose, and the vectors $\mathbf{w}$ and $\mathbf{x}$ are given by

$$\mathbf{w}^T = [w_1 w_2 \ldots w_N] \tag{1.5}$$

$$\mathbf{x}^T = [x_1 x_2 \ldots x_N] \tag{1.6}$$

Throughout this book the boldface lowercase symbol (e.g., $\mathbf{a}$) denotes a vector, and a boldface uppercase symbol (e.g., $\mathbf{A}$) denotes a matrix.

The adaptive processor must select the complex weights, w_k, to optimize a stipulated performance criterion. The performance criterion that governs the operation of the adaptive processor is chosen to reflect the steady-state performance characteristics that are of concern. The most popular performance measures that have been employed include the mean square error [9,28–31]; SNR ratio [6,14,32–34]; output noise power [35]; maximum array gain [36,37]; minimum signal distortion [38,39]; and variations of these criteria that introduce various constraints into the performance index [16,40–43]. In Chapter 3, selected performance measures are formulated in terms of the signal characterizations of (1.1)–(1.4). Solutions are found that determine the optimum choice for the complex weight vector and the corresponding optimum value of the performance measure. The operational signal environment plays a crucial role in determining the effectiveness of the adaptive array to operate. Since the array configuration has pronounced effects on the resulting system performance, it is useful to consider sensor spacing effects before proceeding with an analysis using an implicit description of such effects. The consideration of array configuration is undertaken in Chapter 2.

1.5 EXISTING TECHNOLOGY

In any echo-ranging system, the maximum detection range $R_{\max}$ determines the minimum period between consecutive pulses $T_{\min}$ [and hence the pulse repetition frequency (PRF)], according to

$$T_{\min} = \frac{2R_{\max}}{\mathfrak{v}} \tag{1.7}$$

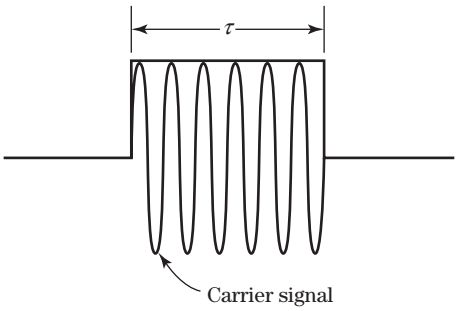


FIGURE 1-2 ■ Pulse modulated carrier signal.

where $\mathfrak{v}$ is the velocity of propagation of the transmitted signal. For underwater applications the velocity of sound in water varies widely with temperature, although a nominal value of 1,500 m/sec can be used for rough calculations. The velocity of electromagnetic wave propagation in the atmosphere can be taken approximately to be the speed of light or 3×10^8 m/sec.

If the range discrimination capability between targets is to be r_d, then the maximum pulse length $t_{\max}$ (in the absence of pulse compression) is given by

$$t_{\max} = \frac{2r_d}{\mathfrak{v}} \tag{1.8}$$

It will be noted that r_d also corresponds to the "blind range"— that is, the range within which target detection is not possible. Since the signal bandwidth $\cong 1/\text{pulse length}$, the range discrimination capability determines the necessary bandwidth of the transducers and their associated electrical channels.

The transmitted pulses form a pulse train in which each pulse modulates a carrier frequency as shown in Figure 1-2. The carrier frequency f_0 in turn determines the wavelength of the propagated wavefront since

$$\lambda_0 = \frac{\mathfrak{v}}{f_0} \tag{1.9}$$

where λ_0 is the wavelength. For sonar systems, frequencies in the range 100–100,000 Hz are commonly employed [44], whereas for radar systems the range can extend from a few megahertz up into the optical and ultraviolet regions, although most equipment is designed for microwave bands between 1 and 40 GHz. The wavelength of the propagated wavefront is important because the array element spacing (in units of λ) is an important parameter in determining the array pattern.

1.5.1 Radar Technology

There has been a steady increase in the demand for increased radar system performance and additional capability for both military and civilian purposes, and the vast number of applications of modern radar technology precludes anything more than the briefest mention of the major areas in which radar systems are found [45]. Military applications may very well involve a number of requirements that in the past would have involved a separate radar system for each different requirement. For example, a fire control system radar may be required to search large volumes of space, to detect and track both high- and low-speed targets ranging from very low to extremely high altitudes, to provide fire

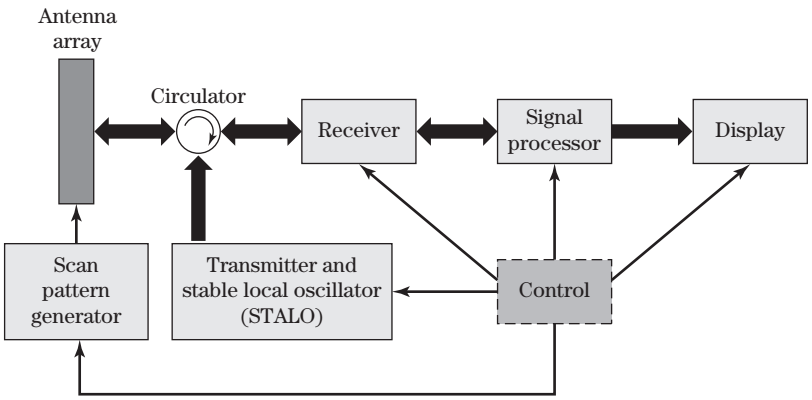


FIGURE 1-3 ■ Simplified block diagram of a typical radar system.

control for both missiles and guns against both airborne and ground (or sea) targets, and additionally to provide navigational aid and perform reconnaissance. Current civil aeronautical needs include air traffic control, collision avoidance, instrument approach systems, weather sensing, and navigational aids. Additional applications in the fields of law enforcement, transportation, and Earth resources are just beginning to grow to sizable proportions [48].

Figure 1-3 is a block diagram of a typical radar system. These major blocks and their corresponding functions are described in Table 1-1 [46]. The antenna, receiver, and signal processing blocks are of primary interest for our purposes, and these are now each briefly discussed in turn.

1.5.1.1 Radiating Elements and Antenna Arrays

The vast frequency range and power over which modern radar systems operate have led to an astonishing variety of radiator elements ranging from parabolic reflectors to horns, dipoles, bow ties [47], multiturn loops [48], spirals [49], log periodics [50], microstrip patches [51], and Vivaldis [52]. Large antenna apertures result in narrow beamwidths that are required for long-range detection and high resolution if targets close to one another are to be distinguished. Microwave frequencies are by far the most popular for radar applications since antenna apertures of relatively small physical size (but large in terms of wavelengths) are reasonable to build.

The antenna type selected for a radar application usually differs from one selected for a communications system. Shaped beam patterns that can be scanned are most popular for radar uses, whereas most communication applications require beams designed for omnidirectional coverage or for fixed point-to-point transmission. The earliest radars (developed

TABLE 1-1 ■ Functions of the Radar Blocks in Figure 1-3

Block	Function
Transmitter	Generates high power RF waveform
Antenna	Determines direction and shape of transmit-and-receive beam
Receiver	Provides frequency conversion and low-noise amplification
Signal processing	Provides target detections, target and clutter tracking, and target trajectory estimates
Display	Converts processed signals into meaningful tactical information

FIGURE 1-4 ■ FuG 65 Wurzburg Riese radar antenna (Courtesy of the National Electronics Museum).

during World War II) operated in the very high frequency (VHF) and ultra high frequency (UHF) bands. Sometimes, a parabolic dish was used, such as the FuG 65 Wurzburg Riese radar antenna in Figure 1-4. Its 3 m parabolic dish operated at 560 MHz and was used to guide German intercept fighters during WWII [53]. Arrays, such as the SCR-270 in Figure 1-5, were used by the United States in WWII for air defense [54]. It has four rows of eight dipoles that operate at 110 MHz. After WWII, radar and communications systems began operating at higher frequencies. Different types of antennas were tried for various systems. A microwave lens was used in the Nike AJAX MPA-4 radar shown in Figure 1-6 [55]. In time, phased array antennas became small enough to place in the nose of fighter airplanes. In the 1980s, the AN/APG-68 (Figure 1-7) was used in the F-16 fighter [56]. The array is a planar waveguide with slots for elements. Active electronically scanned arrays (AESA) provide fast wide angle scanning in azimuth and elevation and include advanced transmit/receive modules [57]. An example is the AN/APG-77 array for the F-22 fighter shown in Figure 1-8.

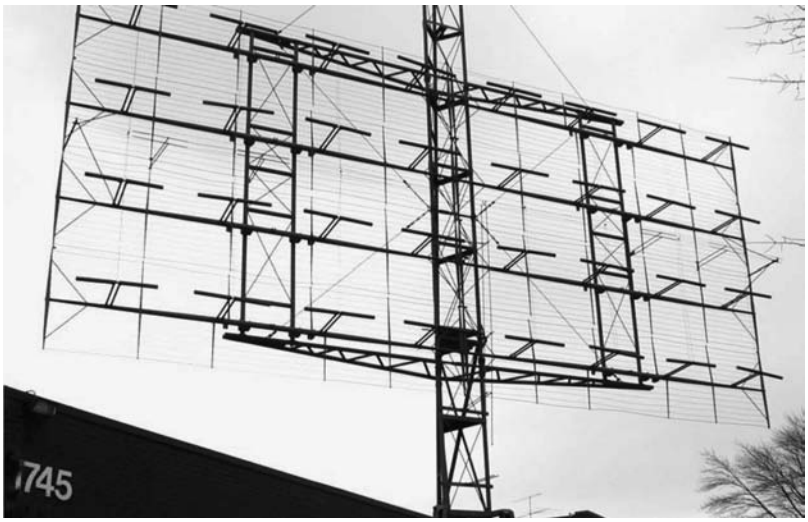


FIGURE 1-5 ■ SCR-270 antenna array (Courtesy of the National Electronics Museum).

FIGURE 1-6 ■ Waveguide lens antenna for the Nike AJAX MPA-4 radar (Courtesy of the National Electronics Museum).

The two most common forms of antenna arrays for radar applications are the linear array and the planar array. A linear array consists of antenna elements arranged in a straight line. A planar array, on the other hand, is a two-dimensional configuration in which the antenna elements are arranged to lie in a plane. Conformal arrays lie on a nonplanar surface. The linear array generates a fan beam that has a broad beamwidth in one plane and a narrow beamwidth in the orthogonal plane. The planar array is most frequently used in radar applications where a pencil beam is needed. A fan-shaped beam is easily produced by a rectangular-shaped aperture. A pencil beam may easily be generated by a square- or circular-shaped aperture. With proper weighting, an array can be made to simultaneously generate multiple search or tracking beams with the same aperture.

Array beam scanning requires a linear phase shift across the elements in the array. The phase shift is accomplished by either software in a digital beamformer or by hardware phase shifters. A phase shifter is often incorporated in a transmit/receive module. Common phase-shifter technology includes ferroelectrics, monolithic microwave integrated

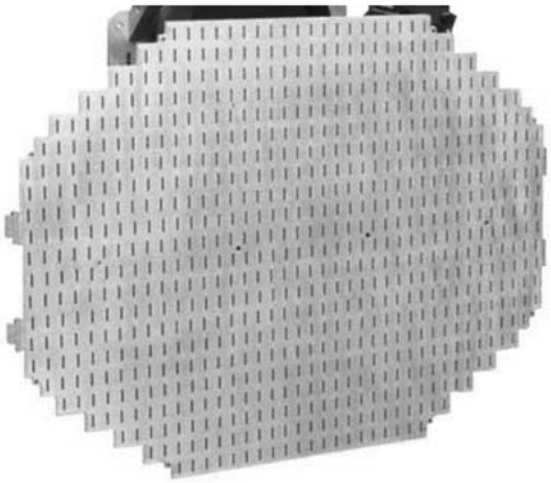


FIGURE 1-7 ■ AN/APG-68 array (Courtesy of Northrop Grumman and available at the National Electronics Museum).

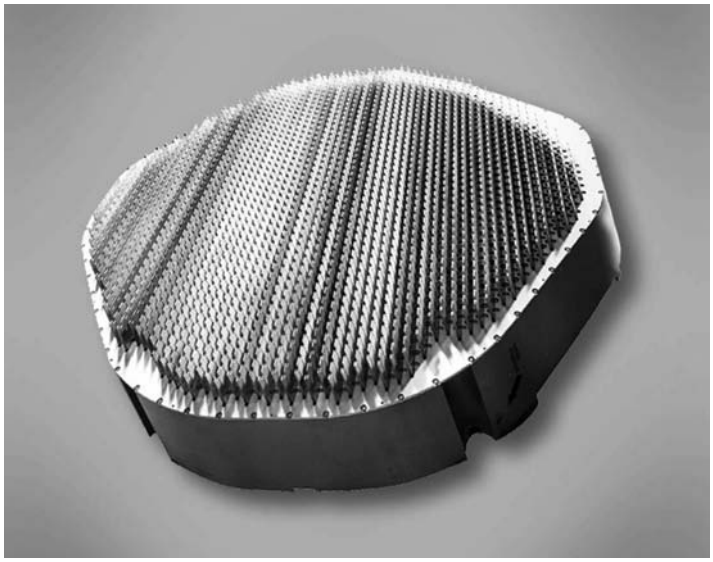


FIGURE 1-8 ■ AN/APG-77 array (Courtesy of Northrop Grumman and available at the National Electronics Museum).

circuit (MMIC), ferrites, and micro-electro-mechanical systems (MEMS). Some of their characteristics are shown in Table 1.2.

1.5.1.2 Receivers

A receiver design based on a matched filter or a cross-correlator maximizes the SNR in the linear portion of the receiver. Different types of receivers that have been employed in radar applications include the superheterodyne, superregenerative, crystal video, and tuned radio frequency (TRF) [58]. The most popular and widely applied receiver type is the superheterodyne, which is useful in applications where simplicity and compactness are especially important. A received signal enters the system through the antenna, then passes through the circulator and is amplified by a low-noise RF amplifier. Following RF amplification, a mixer stage is entered to translate the RF to a lower intermediate frequency (IF) where the necessary gain is easier to obtain and filtering is easier to synthesize. The gain and filtering are then accomplished in an IF amplifier section.

1.5.1.3 Signal Processing

Having maximized the SNR in the receiver section, the next step is to perform two basic operations by signal processing as follows: (1) detection of the presence of any targets, and

TABLE 1-2 ■ Phase-Shifter Characteristics [57]

Type	Ferroelectric	MMIC	Ferrite	MEMS
Cost	Low	High	Very high	Low
Reliability	Good	Very good	Excellent	Good
Power handling	> 1 W	> 10 W	kW	< 50 mW
Switch speed	ns	ns	10 to 100 μs	10 to 100 μs
Direct current power consumption	Low	low	High	Negligible
Size	Small	Small	Large	Small

(2) extraction of information from the received waveform to obtain target trajectory data such as position and velocity. Detecting a signal imbedded in a noise field is treated by means of statistical decision theory. Similarly, the problem of the extraction of information from radar signals can be regarded as a problem concerning the statistical estimation of parameters.

1.5.2 Sonar Technology

Operational active sonar systems may be classified as (1) search light sonar, (2) scanning sonar, or (3) rotational directional transmission (RDT) sonar [59]. Searchlight sonar has very narrow transmit and receive beams and provides an azimuth search capability by mechanically rotating the directional hydrophone array. Since the array is mechanically scanned and aimed, the data rate is correspondingly low, and the system does not provide a multiple-target detection and tracking capability. The requirement for mechanical directional training also limits the array size, so that operational frequencies are usually greater than 15 kHz, thereby increasing attenuation loss.

Scanning sonar systems overcome the data rate limitation of searchlight sonars by transmitting an omnidirectional, short-duration pulse and using electronic means to rapidly rotate a narrow receive beam continuously over a 360° azimuthal sector. The receiving beam output is presented to a panoramic display called a plan position indicator (PPI) that is used extensively in both radar and sonar systems. Scanning-type sonar systems thereby provide a multiple-target detection and tracking capability, and lower operating frequencies can be used, thereby decreasing attenuation losses. The scan speed of the receive beam is a compromise between the desired target resolution and the maximum receiver bandwidth (or minimum input SNR) that is permissible.

An RDT sonar system is characterized by RDT and a scanned preformed beam (PFB) receiver. Consequently, high transmitting directivity and a high data rate are accompanied by a low operational frequency. An RDT system combines the best features of searchlight and scanning sonars. A PFB receiver can have a smaller bandwidth than a scanning receiver, thereby improving the SNR. Furthermore, a PFB receiver can be corrected for Doppler due to own ship's motion employing a method called own Doppler nullifying (ODN), whereas a scanning receiver cannot.

The principal elements of a sonar receiver (Figure 1-9) are the hydrophone array, the beamformer, the signal processor, and the information processor. Each of these principal elements (except for the information processor, which involves display formatting and other command and control functions) is briefly discussed in turn.

1.5.2.1 Sonar Transducers and Hydrophones

A *hydrophone* produces an output voltage proportional to the acoustic signals incident on it; whereas a transducer generates and receives sound. For underwater applications, a very wide frequency range is involved—from about 10 Hz to more than 1 MHz [60].

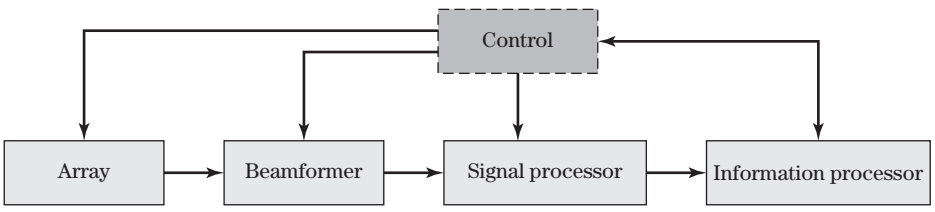


FIGURE 1-9 ■ Sonar receiver block diagram.

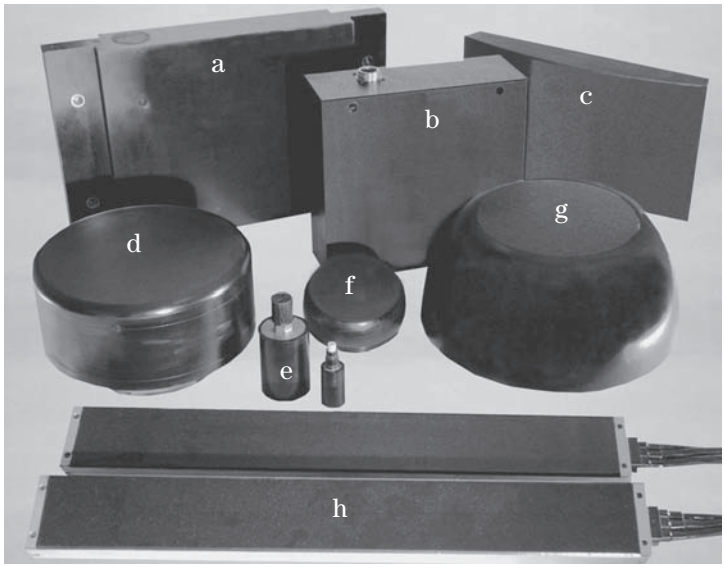


FIGURE 1-10 ■ Various acoustic transducers. a: Seabed mapping—20 kHz multibeam receive array module; 8 shaded elements per module; 10 modules per array. b: Subbottom profiling (parametric)—200 kHz primary frequency; 25 kHz secondary frequency. c: Port and harbor Security—curved 100 kHz transmit/receive array. d. Obstacle avoidance—10 × 10 planar receive array with curved transmitter. e. ACOMMS—Broadband piezocomposite transducers for wideband communication signals. f. AUV FLS—high-frequency, forward-looking sonar array. g. Mine hunting—10 × 10 transmit/receive broadband array; available with center frequencies between 20 kHz to 1MHz h. Side scan—multibeam transmit/receive array (Courtesy of Materials Systems Inc.).

A transmitting power ranging from a few acoustic watts up to several thousand acoustic watts at ocean depths up to 20,000 ft can be achieved [61]. Figure 1-10 shows a sampling of different acoustic transducers manufactured by Materials Systems Inc. for various applications.

The basic physical mechanisms most widely used in transducer technology include the following [62]:

1. Moving coil. This is long familiar from use as a loudspeaker in music reproduction systems and used extensively in water for applications requiring very low frequencies.
2. Magnetorestrictive. Magnetic materials vibrate in response to a changing magnetic field. Magnetorestrictive materials are rugged and easily handled, and magnetorestrictive transducers were highly developed and widely used during World War II.
3. Piezoelectric. The crystalline structure of certain materials results in mechanical vibration when subjected to an alternating current or an oscillating electric field. The relationship between mechanical strain and electric field is linear. Certain ceramic materials also exhibit a similar effect and have outstanding electromechanical properties. Consequently, over the last decade the great majority of underwater sound transducers have been piezoceramic devices that can operate over a wide frequency band and have both high sensitivity and high efficiency.
4. Electrostrictive. Similar to piezoelectric but has a nonlinear relationship between mechanical strain and electric field.

5. Electrostatic. These capacitive transducers use the change in force between two charged parallel plates due to mechanical movement. These have found use with MEMS but not in underwater acoustics.
6. Variable reluctance and hydroacoustic transducers have also been used for certain experimental and sonar development work, but these devices have not challenged the dominance of piezoceramic transducers for underwater sound applications [59].

1.5.2.2 Sonar Arrays

Sonar transducer arrays have low sidelobes and beams that scan over wide angular sectors. Acoustic array configurations include linear, planar, cylindrical, spherical, conformal, volumetric, reflector, and acoustic lenses [63]. These different array types lend themselves to towing, conformal mounting on hulls (where the array surface conforms to the shape of the underwater hull, so that no appendage is required), beam steering, and side-looking sonar and synthetic-aperture applications. Figure 1-11 is an example of a 10×10 acoustic planar array. A circular acoustic array is shown in Figure 1-12. Sometimes, the array must be conformal to the surface on which it is mounted. Figure 1-13 shows a towfish with an acoustic communications (ACOMMS) receive array that operates from 10 to 40 kHz and a sidescan transducer that transmits and receives at 900 kHz. A towfish is a sidescan sonar that is towed underwater by a boat.

A simple 2 ft $\times$ 4 ft planar array having more than 500 sensor elements for deep submergence applications is shown in the diagram of Figure 1-14. In the quest for larger power and lower frequency (with attendant lower attenuation losses), some arrays are very large. In one case a 35 ft $\times$ 50 ft low-frequency planar array weighs 150 tons and transmits close to 10^6 watts [63]. Large arrays with many closely spaced elements develop "hot spots" from mutual coupling. Consequently, the concept of "velocity control" was developed [64,65] to protect individual transducer elements against extreme impedance variations.

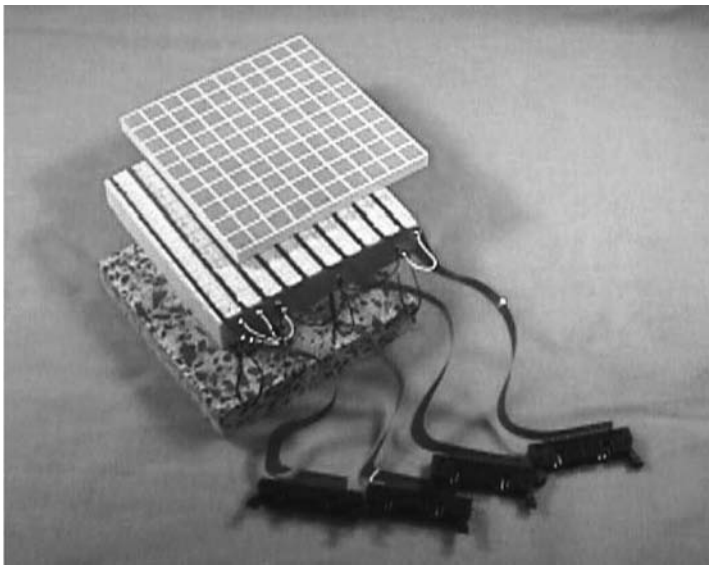


FIGURE 1-11 ■ Picture of a 100-element receive acoustic array manufactured in four layers: matching layer, piezocomposite, flex circuit, and absorbing back (Courtesy of Materials Systems Inc.).

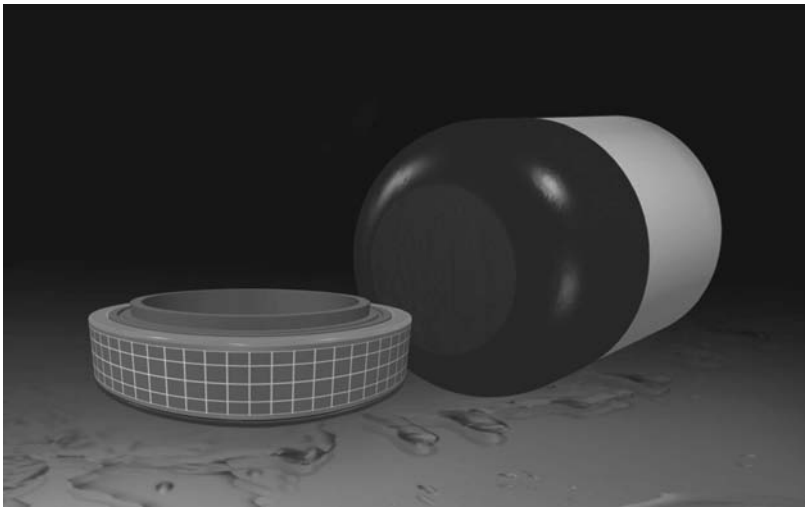


FIGURE 1-12 ■ Left—Port and harbor surveillance piezocomposite array; 100 kHz transmit/receive array. Right—Forward-looking piezocomposite sonar for AUV; piezocomposite facilitates broad bandwidth, high element count arrays, and curved geometries (Courtesy of Materials Systems Inc.).

Surface ships often use a bubble-shaped bow dome in which a cylindrical array is placed like that shown in Figure 1-15. This array uses longitudinal vibrator-type elements composed of a radiating front end of light weight and a heavy back mass, with a spring having active ceramic rings or disks in the middle [63]. The axial symmetry of a cylindrical array renders beam steering fairly simple with the azimuth direction in which the beam is formed, because the symmetry allows identical electronic equipment for the phasing and time delays required to form the beam. Planar arrays do not have this advantage,

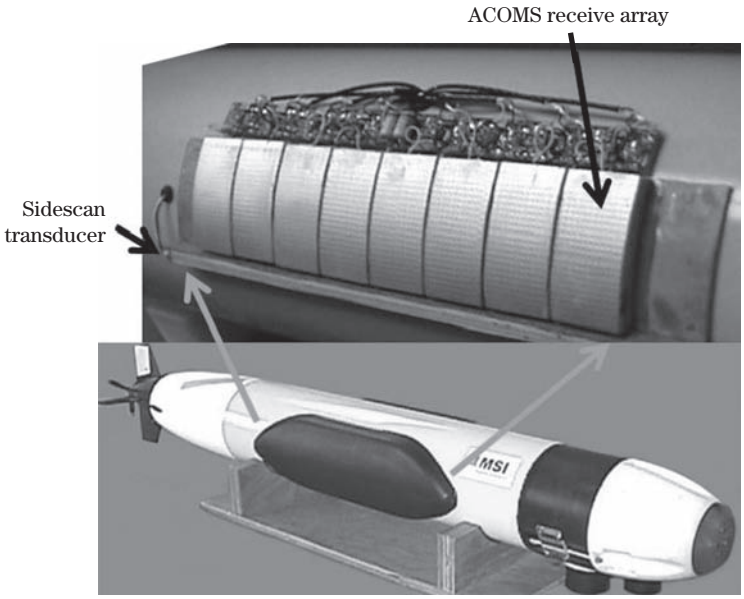


FIGURE 1-13 ■ The acoustic communications (ACOMMS) receive array operates from 10 to 40 kHz, and the sidescan transducer transmits and receives at 900 kHz (Courtesy of Materials Systems Inc.).

FIGURE 1-14 ■ Rectangular planar array for searchlight type sonar system having more than 500 elements.

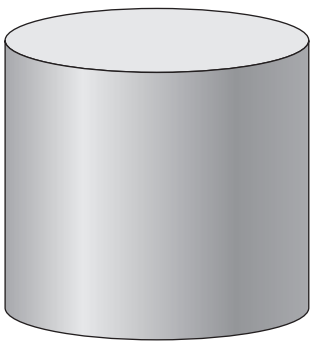


FIGURE 1-15 ■ Cylindrical sonar array used in bow-mounted dome.

since each new direction in space (whether in azimuth or in elevation) requires a new combination of electronic equipment to achieve the desired pointing.

A spherical array is the ideal shape for the broadest array coverage in all directions. Spherical arrays like that shown in the diagram of Figure 1-16 have been built with a diameter of 15 ft and more than 1,000 transducer elements. This spherical arrangement can be integrated into the bow of a submarine by means of an acoustically transparent dome that provides minimum beam distortion. For instance, the bow dome of a Virginia class submarine is a 25 ton hydrodynamically shaped composite structure that houses a sonar transducer sphere. The bow dome is 21 feet tall and has a maximum diameter of 26 feet. A two-inch thick, single-piece rubber boot is bonded to the dome to enhance acoustic performance. Minimal sound energy absorption and reflection properties inherent in the rubber material minimally reflect and absorb acoustic signals. Figure 1-17 shows a spherical microphone array that was constructed by placing a rigid spherical array at the center of a larger open spherical array [66]. Both arrays have 32 omnidirectional microphones and a relatively constant directivity from about 900 Hz to 16 kHz.

FIGURE 1-16 ■ Spherical array having 15 ft diameter and more than 1,000 transducer elements.

FIGURE 1-17 ■ A dual, concentric SMA. (A. Parthy, C. Jin, and A. van Schaik, "Acoustic holography with a concentric rigid and open spherical microphone array," IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 2173–2176.)

1.5.2.3 Beamformer

Beamforming ordinarily involves forming multiple beams from multielement arrays through the use of appropriate delay and weighting matrices. Such beams may be directionally fixed or steerable. After that, sonar systems of the 1950s and 1960s consisted largely of independent sonar sets for each transducer array. More recently, the sophisticated use of multiple sensors and advances in computer technology have led to integrated sonar systems that allow the interaction of data from different sensor arrays [67,68]. Such integrated sonar systems have software delays and weighting matrices, thereby generalizing the structure of digital time domain beamformers. Consequently several units of a single (programmable) beamformer design may be used for all the arrays in an integrated system. Furthermore, programmable beamformer matrices make it possible to adapt the receive pattern to the changing structure of the masking noise background.

1.5.2.4 Signal Processor

Signal processing involves filtering, spectral analysis, correlation, and the related operations of dynamic range compression and normalization (DRCN) that "match" the received array signals to the display/decision functions contained in the information processing block of Figure 1-9 [59,69]. The DRCN removes some of the spatial and temporal characteristics of the acoustic channel that are impressed on the propagating signal. Whether a sufficient degree of DRCN is achieved usually determines whether a system design will succeed or fail since it specifies the integration of a theoretically designed optimum signal processor to its system interfaces.

1.6 ORGANIZATION OF THE BOOK

1.6.1 Part 1

Chapter 2 introduces the adaptive array concept by first considering the nature of the signal environment in which an adaptive array is expected to improve the overall reception of a desired signal. The question of array architecture is considered, and the impact of the architecture selection on the ultimate system performance that can be realized is discussed. The potential of an array to enhance the output SNR performance by adjusting the beamforming network is demonstrated.

An adaptive array manipulates the received signals to meet a defined performance measure. This performance measure is an objective function to be maximized (or minimized). Several widely used performance measures for both narrowband and broadband applications are presented in Chapter 3 that are formulated in terms of the elements found within the adaptive array functional model.

Any adaptive array system must have performance limits:

1. Imposed by the array physical structure.
2. Resulting from the nature of the signal environment

The consideration of performance limits leads to a discussion of the array performance that results after the automatic adaptation process has been permitted to operate long enough to reach a steady-state solution. A steady-state analytic solution to the adaptive array control problem can be found for each performance measure, which enables the designer to determine ultimate system performance limits. The mathematical foundation required to develop the steady-state analytic solution for various performance measures is laid, and the relationships among the solutions obtained to the optimal "Wiener solution" are presented.

1.6.2 Part 2

The heart of the adaptive capability within an adaptive array system is the adaptive algorithm that adjusts the array pattern in response to the signal information found at the sensor element outputs. Part 2, including Chapters 4 through 8, introduces different classes of adaptation algorithms. In some cases adaptation algorithms are selected according to the kind of signal information available to the receiver:

1. The desired signal is known.
2. The desired signal is unknown, but its direction of arrival is known.

3. The desired signal is unknown and its direction of arrival is known.
4. No signal information is available at the outset, but as array operation progresses such information must be "learned" by the adaptive processor. These algorithms are called blind adaptive algorithms.

The selection of an adaptive algorithm has important consequences for the system transient performance.

The convergence properties of the various adaptation algorithms are analyzed and performance comparisons are made. Furthermore, the shortcomings of the algorithms under certain conditions are discussed. These results are summarized for convenience and provide the designer with a means for assessing which candidate algorithms are most appropriate for the signal conditions and system requirements.

1.6.3 Part 3

The adaptive array operating conditions considered so far were nonideal only in that interference signals were present with which the array had to contend. In actual practice, however, the effects of several other nonideal operating conditions often result in unacceptable degradation of array performance unless compensation of such effects is undertaken. Such nonideal operating conditions include processing of broadband signals, multipath effects, channel mismatching, and array propagation delay effects. Compensation for these factors by means of tapped delay-line processing is considered, and the question of how to design a tapped delay line to achieve a desired degree of compensation is addressed. Finally, current trends in adaptive array research that provide an indication of the direction that future developments are likely to take are discussed.

1.7 SUMMARY AND CONCLUSIONS

The motivation for and actual use of adaptive array systems are presented. The principal elements of an adaptive array system are defined, and the fundamental problems facing an adaptive array designer are given. Adaptive array design is a compromise among such factors as [70]:

1. Hardware complexity and cost
2. Data rate
3. Maximum range of detection (for radar and sonar)
4. Resolution in angle (and range and Doppler for radar and sonar)
5. Precision in the measurement of range, bearing, and Doppler (for radar and sonar)
6. Ability of the adaptive array to meet both transient and steady-state system performance requirements

A suboptimal acoustical array processor known as the DICANNE processor operated in sea tests against ship-generated interferences and consistently formed cancellation nulls 10–15 dB deep [59]. Use of an optimal wideband processor based on the minimum signal distortion performance measure in a computer-simulated sonar experiment resulted in effectively suppressing a strong coherent interfering signal by forming cancellation nulls 50 dB deep [59]. Such deep cancellation nulls were found, however, to be quite sensitive to

(1) small changes in interference signal bearing, (2) small errors in the adaptive weight values, and (3) statistical fluctuations of measured correlations due to finite integration time.

A lightweight four-element adaptive array using hybrid microwave integrated circuitry and weighing only 1 pound, intended for communication applications, was built and tested [71]. This unit employed a null-steering algorithm appropriate for a coherent sidelobe canceller and succeeded in forming broadband nulls over a 60–100 MHz bandwidth having a cancellation depth of 25–30 dB under weak desired signal and strong interference signal conditions. To attain this degree of interference signal cancellation, it was essential that the element channel circuitry be very well matched over a 20% bandwidth.

Another experimental four-element adaptive array system for eliminating interference in a communication system was also tested [48]. Pattern nulls of 10–20 db for suppressing interference signals over a 200–400 MHz band were easily achieved so long as the desired signal and interference signal had sufficient spatial separation (greater than the resolution capability of the antenna array), assuming the array has no way to distinguish between signals on the basis of polarization. Exploiting polarization differences between desired and interference signals by allowing full polarization flexibility in the array, an interference signal located at the same angle as the desired signal can be suppressed without degrading the reception of the desired signal. Yet another system employing digital control was developed for UHF communications channels and found capable of suppressing jammers by 20–32 dB [72].

In summary, interference suppression levels of 10–20 dB are consistently achieved in practice. It is more difficult but nevertheless practicable to achieve suppression levels of 20–35 dB and usually very difficult to form cancellation nulls greater than 35 dB in a practical operating system.

The rapid development of digital technology is presently having the greatest impact on signal reception systems. The full adaptation of digital techniques into the processing and interpretation of received signals is making possible the realization of practical signal reception systems whose performance approaches that predicted by theoretical limits. Digital processors and their associated memories have made possible the rapid digestion, correlation, and classification of data from larger search volumes, and new concepts in the spatial manipulation of signals have been developed. Adaptive array techniques started out with limited numbers of elements in the arrays, and the gradual increase in the numbers of elements and in the sophistication of the signal processing will likely result in an encounter with techniques employed in optical and acoustical holography [69,73]. Holography techniques are approaching such an encounter from the other direction, since they start out with a nearly continuous set of spatial samples (as in optical holography) and move down to a finite number of samples (in the case of acoustic holography).

1.8 PROBLEMS

1. **Radar Pulse Waveform Design** Suppose it is desired to design a radar pulse waveform that would permit two Ping-Pong balls to be distinguished when placed only 6.3 cm apart in range up to a maximum range from the radar antenna of 10 m.

   (a) What is the maximum PRF of the resulting pulse train?

   (b) What bandwidth is required for the radar receiver channel?

   (c) If it is desired to maintain an array element spacing of $d = 2$ cm where $d = \lambda_0/2$, what pulse carrier frequency should the system be designed for?

2. **Sonar Pulse Carrier Frequency Selection** In the design of an actual sonar system many factors must be considered—all the sonar parameters (e.g., source level, target strength) and the environment parameters. The effect of environmental parameters depends largely on frequency. Suppose in a highly oversimplified example that only the factors of transmission loss (due to attenuation) and ambient noise are of concern. Let the attenuation coefficient α be given by

$$\log_{10}(\alpha) = \frac{1}{4}[-21 + 5\log_{10}(f)]$$

Furthermore, let the ambient noise spectrum level N_0 be given by

$$10\log_{10}(N_0) = \frac{1}{3}[20 - 50\log_{10}(f)]$$

If the cost to system performance is given by $J = C_1\alpha + C_2N_0$ where C_1 and C_2 denote the relative costs of attenuation and noise to the system, what value of pulse carrier frequency f should be selected to optimize the system performance?

1.9 REFERENCES

[1] L. C. Van Atta, "Electromagnetic Reflection," U.S. Patent 2908002, October 6, 1959.

[2] *IEEE Trans. Antennas Propag.* (Special Issue on Active and Adaptive Antennas), Vol. AP-12, March 1964.

[3] D. L. Margerum, "Self-Phased Arrays," in *Microwave Scanning Antennas*, Vol. 3, *Array Systems*, edited by R. C. Hansen, Academic Press, New York, 1966, Ch. 5.

[4] P. W. Howells, "Intermediate Frequency Sidelobe Canceller," U.S. Patent 3202990, August 24, 1965.

[5] P. W. Howells, "Explorations in Fixed and Adaptive Resolution at GE and SURC," *IEEE Trans. Antennas Propag.*, Special Issue on Adaptive Antennas, Vol. AP-24, No. 5, pp. 575–584, September 1976.

[6] S. P. Applebaum, "Adaptive Arrays," Syracuse University Research Corporation, Rep. SPL TR66-1, August 1966.

[7] B. Widrow, "Adaptive Filters I: Fundamentals," Stanford University Electronics Laboratories, System Theory Laboratory, Center for Systems Research, Rep. SU-SEL-66-12, Tech. Rep. 6764-6, December 1966.

[8] B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, edited by R. E. Kalman and N. DeClaris, Holt, Rinehard and Winston, New York, 1971, Ch. 5.

[9] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, No. 12, p. 2143–2159, December 1967.

[10] L. Spitzer, Jr., "Basic Problems of Underwater Acoustic Research," National Research Council, Committee on Undersea Warfare, Report of Panel on Underwater Acoustics, September 1, 1948, NRC CUW 0027.

[11] R. H. Bolt and T. F. Burke, "A Survey Report on Basic Problems of Acoustics Research," Panel on Undersea Warfare, National Research Council, 1950.

[12] H. Mermoz, "Filtrage adapté et utilisation optimale d'une antenne," Proceedings NATO Advanced Study Institute, September 14–26, 1964. "Signal Processing with Emphasis on Underwater Acoustics" (Centre d'Etude des Phénomenès Aléatoires de Grenoble, Grenoble, France), 1964, pp. 163–294.

[13] F. Bryn, "Optimum Signal Processing of Three-Dimensional Arrays Operating on Gaussian Signals and Noise," *J. Acoust. Soc. Am.*, Vol. 34, March 1962, pp. 289–297.

[14] S. W. W. Shor, "Adaptive Technique to Discriminate Against Coherent Noise in a Narrowband System," *J. Acoust. Soc. Am.*, Vol. 39, January 1966, pp. 74–78.

[15] P. E. Green, Jr., R. A. Frosch, and C. F. Romney, "Principles of an Experimental Large Aperture Seismic Array (LASA)," *Proc. IEEE*, Vol. 53, December 1965, pp. 1821–1833.

[16] R. T. Lacoss, "Adaptive Combining of Wideband Array Data for Optimal Reception," *IEEE Trans. Geosci. Electron.*, Vol. GE-6, May 1968, pp. 78–86.

[17] B. D. Steinberg, "On Teleseismic Beam Formation by Very Large Arrays," *Bull. Seismol. Soc. Am.*, August 1971.

[18] *IEEE Trans. Antennas Propag.* (Special Issue on Adaptive Antennas), Vol. AP-24, September 1965.

[19] A. J. Berni, "Angle of Arrival Estimation Using an Adaptive Antenna Array," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-11, No. 2, March 1975, pp. 278–284; see errata *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-11, May and July 1975.

[20] R. C. Davis, L. E. Brennan, and I. S. Reed, "Angle Estimation with Adaptive Arrays in External Noise Fields," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-12, No. 2, March 1976, pp. 179–186.

[21] B. D. Steinberg, *Principles of Aperture and Array System Design*, Wiley, New York, 1976, Ch. 11.

[22] B. D. Steinberg, "Design Approach for a High-Resolution Microwave Imaging Radio Camera," *J. Franklin Inst.*, Vol. 296, No. 6, December 1973.

[23] E. N. Powers and B. D. Steinberg, "Valley Forge Research Center Adaptive Array," USNC/URSI Annual Meeting, Boulder, CO, October 1975.

[24] E. Biglieri, *MIMO Wireless Communications*, Cambridge, Cambridge University Press, 2007.

[25] M. H. Er, and S. K. Hui, "Beamforming in Presence of Element Failure," *Electronics Letters,* vol. 27, no. 3, pp. 273–275, 1991.

[26] J. T. Aberle, O. Sung-Hoon, D. T. Auckland, et al., "Reconfigurable Antennas for Wireless Devices," *Antennas and Propagation Magazine, IEEE,* vol. 45, no. 6, pp. 148–154, 2003.

[27] C.C. Chang, R.H. Lee, and T.Y. Shih, "Design of a beam switching/steering Butler matrix for phased array system," *IEEE Transactions on Antennas and Propagation,* vol. 58, no. 2, pp. 367–374, Feb. 2010.

[28] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. IEEE*, Vol. 57, No. 10, pp. 1696–1704, October 1969.

[29] J. Chang and F. Tuteur, "Optimum Adaptive Array Processor," Proceedings of the Symposium on Computer Processing in Communications, April 8–10, 1969, Polytechnic Institute of Brooklyn, pp. 695–710.

[30] O. L. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proc. IEEE*, Vol. 60, No. 8, pp. 926–935, August 1972.

[31] C. A. Baird, "Recursive Processing for Adaptive Arrays," *Proceedings of Adaptive Antenna Systems Workshop,* Vol. 1, Naval Research Laboratory, Washington, DC, September 27, 1974, pp. 163–180.

[32] R. T. Adams, "An Adaptive Antenna System for Maximizing Signal-to-Noise Ratio," Wescon Conference Proceedings, Session 24, 1966, pp. 1–4.

[33] L. J. Griffiths, "Signal Extraction Using Real-Time Adaptation of a Linear Multichannel Filter," SEL-68-017, Tech. Rep. No. 6788-1, System Theory Laboratory, Stanford University, February 1968.

[34] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, March 1971, pp. 254–262.

[35] C. A. Baird, Jr. and J. T. Rickard, "Recursive Estimation in Array Processing," Proceedings of the Fifth Asilomar Conference on Circuits and Systems, 1971, Pacific Grove, CA, pp. 509–513.

[36] J. J. Faran and R. Hills, Jr., "Wide-Band Directivity of Receiving Arrays," Harvard University Acoustics Research Laboratory, Tech. Memo 31, May 1, 1953.

[37] H. H. Az-Khatib and R. T. Compton, Jr., "A Gain Optimizing Algorithm for Adaptive Arrays," *IEEE Trans. Antennas Propag.*, Vol. AP-26, No. 2, March 1978, pp. 228–235.

[38] N. Wiener, *Extrapolation, Interpretation, and Smoothing of Stationary Time Series*, New York, MIT Press, 1949.

[39] H. L. Van Trees, "Optimum Processing for Passive Sonar Arrays," *IEEE 1966 Ocean Electronics Symposium,* August 29–31, Honolulu, HI, pp. 41–65.

[40] K. Takao, M. Fujita, and T. Nishi, "An Adaptive Antenna Array Under Directional Constraint," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 662–669.

[41] J. Capon, R. J. Greenfield, and R. J. Kolker, "Multi-Dimensional Maximum-Likelihood Processing of a Large Aperture Seismic Array," *Proc. IEEE*, Vol. 55, February 1967, pp. 192–217.

[42] J. F. Clearbout, "A Summary, by Illustration, of Least-Square Filter With Constraints," *IEEE Trans. Inf. Theory*, Vol. IT-14, March 1968, pp. 269–272.

[43] A. Booker and C. Y. Ong, "Multiple-Constraint Adaptive Filtering," *Geophysics*, Vol. 36, June 1971, pp. 498–509.

[44] R. L. Deavenport, "The Influence of the Ocean Environment . . . ," *EASCON 1975 Record, IEEE Electronics and Aerospace Systems Convention,* September 29–October 1, Washington, DC, pp. 66. A - 66E.

[45] E. Brookner, "Now: Phased-array Radars: Past, Astounding Breakthroughs and Future Trends, January 2008," *Microwave Journal,* Vol. 51, no. 1, pp. 31–48, 2008.

[46] W. G. Bruner, W. N. Case, and R. C. Leedom, "Digital Control in Modern Radar," *EASCON 1971 Record, IEEE Electronics and Aerospace Systems Convention,* October 6-8, 1971, Washington, DC, pp. 200–207.

[47] R. T. Compton, Jr., "An Experimental Four-Element Adaptive Array," *IEEE Trans. Antennas & Prop.,* Vol AP-25, No. 5, September 1965, pp. 697–706.

[48] W. L. Curtis, "Spiral Antennas," *IRE Trans. Antennas & Prop*, Vol AP-8, May, 1960, pp. 298–306.

[49] R. H. DuHamel and D. G. Berry, "Logarithmically Periodic Antenna Arrays," *IRE WESCON Convention Record,* 1958, Vol. 2, Pt. 1, pp. 161–174.

[50] R. Garg, P. Bhartia, I. Bahl, and A. Ittipiboon, *Microstrip Antenna Design Handbook,* Norwood, MA, Artech House, 2001.

[51] P. J. Gibson, "the Vivaldi Aerial," *9th European Microwave Conf. Proc.,* 19779, pp. 103–105.

[52] National Electronics Museum, Wurzburg Riese Radar Antenna, 2009, Display Information.

[53] National Electronics Museum, SCR-270 Antenna, 2009, Display Information.

[54] National Electronics Museum, Nike AJAX MPA-4 Antenna, 2009, Display Information.

[55] National Electronics Museum, AN/APG-68 Antenna, 2009, Display Information.

[56] M. I. Skolnik, *Introduction to Radar Systems,* McGraw-Hill, New York, 1962, Ch. 2.

[57] C. H. Sherman, "Underwater Sound - A Review: I. Underwater Sound Transducers," *IEEE Trans. Sonics and Ultrason.,* Vol. SU-22, No. 5, September 1975, pp. 281–290.

[58] A. A. Winder, "Underwater Sound - A Review: II. SKonar System Technology," *IEEE Trans. Sonics and Ultrason.,* Vol. SU-22, No. 5, September 1975, pp. 291–332.

[59] E. A. Massa, "Deep Water trnasducers for Ocean Engineering Applications," *Proceedings IEEE,* 1966 Ocean Electronics Symposium, August 29–31, Honolulu, HI, pp. 31–39.

[60] C. H. Sherman and J. L. Butler, *Transducers and Arrays for Underwater Sound,* New York, Springer, 2007.

[61] T. F. Hueter, "Twenty Years in Underwater Acoustics: Generation and Reception," *J. Acoust. Soc. Am.,* Vol. 51, No 3 (Part 2), March 1972, pp. 1025–1040.

[62] J. S. Hickman, "Trends in Modern Sonar Transducer Design," *Proceedings of the Twenty-Second National Electronics Conference,* October 1966, Chicago, Ill.

[63] E. L. Carson, G. E. Martin, G. W. Benthien, and J. S. Hickman, "Control of Element Velocity Distributions in Sonar Projection Arrays," *Proceedings of the Seventh Navy Science Symposium,* May 1963, Pensacola, FL.

[64] A. Parthy, C. Jin, and A. van Schaik, "Acoustic Holography with a Concentric Rigid and Open Spherical Microphone Array," *IEEE International Conference on Acoustics,* Speech and Signal Processing, 2009, pp. 2173–2176.

[65] J. F. Bartram, R. R. Ramseyer, and J. M Heines, "Fifth Generation Digital Sonar Signal Processing," *IEEE EASCON 1976 Record, IEEE Electronics and Aerospace Systems Convention,* September 26–29, Washington, D.C., pp. 91A–91G.

[66] T. F. Hueter, J. C. Preble, and G. D. Marshall, "Distributed Architecture for Sonar Systems and Their Impact on Operability," *IEEE EASCON 1976 Record, IEEE Electronics and Aerospace Convention,* September 16–29, Washington, D.C., pp. 96A–96F.

[67] V. C. Anderson, "The First Twenty Years of Acoustic Signal Processing," *J. Acoust. Soc. Am.,* Vol. 51, No. 3 (Part 2), March 1972, pp. 1062–1063.

[68] M. Federici, "On the Improvement of Detection and Precision Capabilities of Sonar Systems," *Proceedings of the Symposium on Sonar Systems*, July 9–12, 1962, University of Birmingham, pp. 535–540, Reprinted in J. Br. I.R.E., Vol. 25, No.6, June 1963.

[69] G. G. Rassweiler, M. R. Williams, L. M. Payne, and G. P. Martin, "A Miniaturized Lightweight Wideband Null Steerer," *IEEE Trans. Ant. & Prop.,* Vol. AP-24, September 1976, pp. 749–754.

[70] W. R. Jones, W. K. Masenten, and T. W. Miller, "UHF Adaptive Array Processor Development for Naval Communications," *Digest AP-S Int. Symp., IEEE Ant. & Prop Soc.,* June 2–22, 1977, Stanford, CA., IEEE Cat. No. 77CH1217-9 AP, pp. 394–397.

[71] G. Wade, "Acoustic Imaging with Holography and Lenses," *IEEE Trans. Sonics & Ultrason.,* Vol. SU-22, No. 6, November 1975, pp 385–394.

PART I

Adaptive Array Fundamental Principles: System Uses, System Elements, Basic Concepts, and Optimum Array Processing

CHAPTER

1 Introduction

Chapter Outline

An array of sensor elements has long been an attractive solution for severe reception problems that commonly involve signal detection and estimation. The basic reason for this attractiveness is that an array offers a means of overcoming the directivity and sensitivity limitations of a single sensor, offering higher gain and narrower beamwidth than that experienced with a single element. In addition, an array has the ability to control its response based on changing conditions of the signal environment, such as direction of arrival, polarization, power level, and frequency. The advent of highly compact, inexpensive digital computers has made it possible to exploit well-known results from signal processing and control theory to provide optimization algorithms that automatically adjust the response of an adaptive array and has given rise to a new domain called "smart arrays." This self-adjusting capability renders the operation of such systems more flexible and reliable and (more importantly) offers improved reception performance that would be difficult to achieve in any other way. This revised edition acquaints the reader with the historical background of the field and presents important new developments that have occurred over the last quarter century that have improved the utility and applicability of this exciting field.

1.1 MOTIVATION FOR USING ADAPTIVE ARRAYS

An array consists of two or more sensors in which the signals are coherently combined in a way that increases the antenna's performance. Arrays have the following advantages over a single sensor:

1. Higher gain. The gain is higher, because the array gain is on the order of the number of elements in the array. Higher resolution or narrower main beam follows from the larger aperture size.
2. Electronic beam scanning. Moving large antennas to steer the main beam is slow. Arrays with phase shifters at each element are able to steer the beam without mechanical motion, because the signals are made to add in phase at the beam steering angle.
3. Low sidelobes. If the desired signal enters the main beam while interfering signals enter the sidelobes, then lowering the sidelobes relative to the main beam improves the signal to interference ratio.
4. Multiple beams. Certain array feeds allow simultaneous multiple main beams.
5. Adaptive nulling. Adaptive arrays automatically move nulls in the directions of signals over the sidelobe region.

On the other hand, these advantages are countered by the significant disadvantages of increased cost and complexity.

Conventional signal reception systems are susceptible to degradation in signal-to-noise ratio (SNR) [or more generally, the signal-to-interference plus noise ratio (SINR)] performance because of the inevitable presence in the signal environment of undesired "noise" signals that enter the sidelobes or main beam of an array. Signals that interfere with the desired signal include deliberate electronic countermeasures (ECMs), nonhostile radiofrequency interference (RFI), clutter, multipath, and natural noise sources. The resulting SNR degradation is further aggravated by array motion, poor siting, sensor failures, and a nonstationary interference environment. As traffic in the spectrum increases, the suppression of interference becomes even more paramount.

Adaptive arrays improve the reception of desired signals in the presence of interference signals in radar, sonar, seismic, and communications systems. They automatically sense the presence of interference and suppress them while simultaneously enhancing desired signal reception without prior knowledge of the signal–interference environment. Adaptive arrays are designed to complement other interference suppression techniques, such as low sidelobes, spread-spectrum techniques, and high directivity.

An adaptive array has a computer algorithm that controls the signal levels at the elements until a measure of the quality of the array performance improves. It adjusts its pattern to form nulls, to modify gain, to lower sidelobes, or to do whatever it takes to improve its performance. An adaptive array offers enhanced reliability compared with that of a conventional array. When a single sensor element in a conventional array fails, the sidelobe structure of the array pattern degrades. With an adaptive array, however, the remaining operational sensors in the array automatically adjust to restore the pattern. Adaptive arrays are more reliable than conventional arrays, because they fail gracefully. The reception pattern of an array in place on an aircraft or ship is often quite different from the array pattern measured in isolation (in an anechoic chamber) as a result of signal scattering that occurs from vehicle structures located in the vicinity of the antenna. An adaptive array often yields successful operation even when antenna patterns are severely

distorted by near-field effects. The adaptive capability overcomes any distortions that occur in the near field (i.e., at distances from the radiating antenna closer than $\lambda/2\pi$ where λ is the wavelength) and merely responds to the signal environment that results from any such distortion. Likewise, in the far field (at distances from the radiating antenna greater than 2λ) the adaptive antenna is oblivious to the absence of any distortion.

An adaptive array improves the SNR by preserving the main beam that points at the desired signal at the same time that it places nulls in the pattern to suppress interference signals. Very strong interference suppression is possible by forming pattern nulls over a narrow bandwidth. This exceptional interference suppression capability is a principal advantage of adaptive arrays compared to waveform processing techniques, which generally require a large spectrum-spreading factor to obtain comparable levels of interference suppression. Sensor arrays possessing this key automatic response capability are sometimes referred to as "smart" arrays, since they respond to far more of the signal information available at the sensor outputs than do more conventional array systems.

The capabilities provided by the adaptive array techniques to be discussed in this book offer practical solutions to the previously mentioned realistic interference problems by virtue of their ability to sort out and distinguish the various signals in the spatial domain, in the frequency domain, and in polarization. At the present time, adaptive nulling is considered to be the principal benefit of the adaptive techniques employed by adaptive array systems, and automatic cancellation of sidelobe jamming provides a valuable electronic counter–countermeasure (ECCM) capability for radar systems. Adaptive arrays are designed to incorporate more traditional capabilities such as self-focusing on receive and retrodirective transmit. In addition to automatic interference nulling and beam steering, adaptive imaging arrays may also be designed to obtain microwave images having high angular resolution. It is useful to call self-phasing or retrodirective arrays *adaptive transmitting arrays* to distinguish the principal function of such systems from an *adaptive receiving array*, the latter being the focus of this book.

1.2 HISTORICAL PERSPECTIVE

The term *adaptive antenna* was first used by Van Atta [1] and others [2] to describe a self-phasing antenna system that automatically reradiates a signal in the direction from which it was received. This "retrodirective" system works without prior knowledge of the signal direction. Retrodirective arrays in point-to-point satellite communications systems automatically return a strong pilot signal to overcome the usual beamwidth (and consequent directivity) limitations.

The development of the phase-lock loop was another major step that made possible the self-steering (or self-phasing) type of adaptive array [3]. A self-phased array has each of the array elements independently phased, based on information obtained from the received signals. For example, several large-aperture receiving antennas with slaved steering can be self-phased on received signals from satellites or space vehicles so the effective receiving aperture is the sum of the individual apertures of all participating antennas.

In the early 1960s the key capability of adaptive interference nulling was recognized and developed by Howells [4,5]. Subsequently, Applebaum established the control law associated with the Howells adaptive nulling scheme by analyzing an algorithm that maximizes a generalized SNR [6]. Concurrently, the capability of self-training or self-optimizing control was applied to adaptive arrays by Widrow and others [7-9]. The

self-optimizing control work established the least mean square (LMS) error algorithm that was based on the method of steepest descent. The Applebaum and the Widrow algorithms are very similar, and both converge toward the optimum Wiener solution.

The use of sensor arrays for sonar and radar signal reception had long been common practice by the time the early adaptive algorithm work of Applebaum and Widrow was completed [10,11]. Early work in array processing concentrated on synthesizing a "desirable" pattern. Later, attention shifted to the problem of obtaining an improvement in the SNR [12–14]. Seismic array development commenced about the same period, so papers describing applications of seismic arrays to detect remote seismic events appeared during the late 1960s [15–17].

The major area of current interest in adaptive arrays is their application to problems arising in radar and communications systems, where the designer almost invariably faces the problem of interference suppression [18]. A second example of the use of adaptive arrays is that of direction finding in severe interference environments [19,20]. Another area in which adaptive arrays are proving useful is for systems that require adaptive beamforming and scanning in situations where the array sensor elements must be organized without accurate knowledge of element location [21]. Furthermore, large, unstructured antenna array systems may employ adaptive array techniques for high angular resolution imaging [22,23]. Adaptive antennas are a subset of smart antennas and include topics such as multiple input, multiple output (MIMO) [24], element failure compensation [25], reconfigurable antennas [26], and beam switching [27].

1.3 PRINCIPAL SYSTEM ELEMENTS

Figure 1-1 shows a diagram of an adaptive array. It consists of the sensor array, the beamforming network, and the adaptive processor that adjusts the variable weights in the beamforming network. The array design depends on the propagation medium in which the array operates, the frequency spectrum of interest, and the user's knowledge of the operational signal environment.

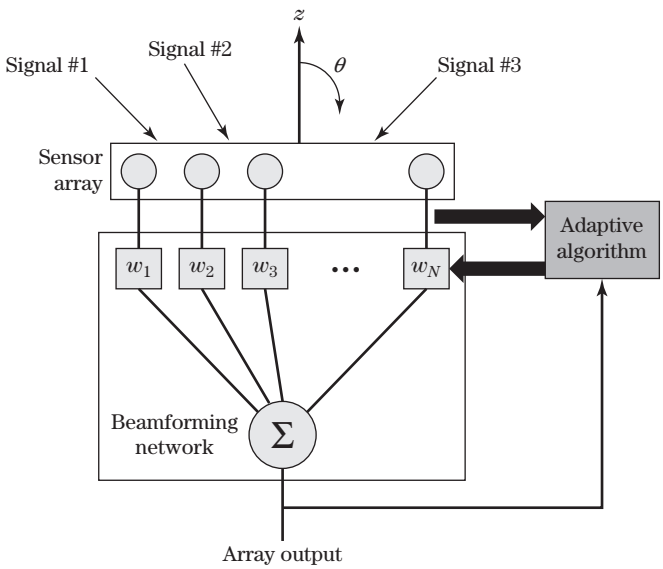


FIGURE 1-1 ■ Functional diagram of an *N*-element adaptive array.

The array consists of N sensors designed to receive (and transmit) signals in the propagation medium. The sensors are arranged to give adequate coverage (pattern gain) over a desired spatial region. The selection of the sensor elements and their physical arrangement place fundamental limitations on the ultimate capability of the adaptive array system. The output of each of the N elements goes to the beamforming network, where the output of each sensor element is first multiplied by a complex weight (having both amplitude and phase) and then summed with all other weighted sensor element outputs to form the overall adaptive array output signal. The weight values within the beamforming network (in conjunction with the sensor elements and their physical arrangement) then determine the overall array pattern. It is the ability to shape this overall array pattern that in turn determines how well the specified system requirements can be met for a given signal environment.

The exact structure of the adaptive algorithm depends on the degree of detailed information about the operational signal environment that is available to the array. As the amount of a priori knowledge (e.g., desired signal location, jammer power levels) concerning the signal environment decreases, the adaptive algorithm selected becomes critical to a successful design. Since the precise nature and direction of all signals present as well as the characteristics of the sensor elements are not known in practice, the adaptive algorithm must automatically respond to whatever signal environment (within broad limits) confronts it. If any signal environment limits are known or can reasonably be construed, such bounds are helpful in determining the adaptive processor algorithm used.

1.4 ADAPTIVE ARRAY PROBLEM STATEMENT

The fundamental problem facing the adaptive array designer is to improve the reception of a desired signal in the presence of undesired interfering signals. The terms *desired signal* and *interfering signals* imply that the characteristics of these two signal classes are different in some respect and that this difference provides the key to improving the desired signal reception. For example, if the direction of arrival of the desired signal is known (or can be deduced), then any signals arriving from different directions are suppressed by forming array pattern nulls in those directions. Likewise, if the interference signals are outside the desired bandwidth, then the interference signals are eliminated by band-pass filtering. Certain characteristics of the desired signal distinguish it from interference signals, so it is reasonable to assume that the nature of the desired signal is known even though certain signal parameters (e.g., direction of arrival, amplitude, phase) must be estimated. If the designer were solely concerned with suppressing interfering signals, then desired signal reception might suffer. Likewise, if desired signal enhancement were the sole focus of attention, then interference signal reception might also be enhanced. Therefore, the twin (and sometimes conflicting) objectives of desired signal enhancement and interference signal suppression are sought so that the overall desired signal reception performance is improved. In many cases, the overall reception performance is best measured by the output SNR. For passive sensor systems, however, the basic problem is that of determining whether a desired signal is present in a background of ambient noise and interfering signals. Determining signal presence or absence requires a decision that is not provided simply by maximizing the output SNR, and statistical decision theory provides solutions to problems of this kind that minimize the risk associated with incorrect decisions. The optimum processors prescribed by statistical decision theory are closely related to those

obtained by maximizing the output SNR, so there is an underlying unity to problems that initially appear to be quite different.

An adaptive array design includes the sensor array configuration, beamforming network implementation, signal processor, and adaptive algorithm that enables the system to meet several different requirements on its resulting performance in as simple and inexpensive a manner as possible. The system performance requirements are conveniently divided into two types: transient response and steady-state response. Transient response refers to the time required for the adaptive array to successfully adjust from the time it is turned on until reaching steady-state conditions or successfully adjusting to a change in the signal environment. Steady-state response refers to the long-term response after the weights are done changing. Steady-state measures include the shape of the overall array pattern and the output signal-to-interference plus noise ratio. Several popular performance measures are considered in detail in Chapter 3. The response speed of an adaptive array depends on the type of algorithm selected and the nature of the operational signal environment. The steady-state array response, however, can easily be formulated in terms of the complex weight settings, the signal environment, and the sensor array structure.

A fundamental trade-off exists between the rapidity of change in a nonstationary noise field and the steady-state performance of an adaptive system: generally speaking, the slower the variations in the noise environment, the better the steady-state performance of the adaptive array. Any adaptive array design needs to optimize the trade-off between the speed of adaptation and the accuracy of adaptation.

System requirements place limits on the transient response speed. In an aircraft communication system, for example, the signal modulation rate limits the fastest response speed (since if the response is too fast, the adaptive weights interact with the desired signal modulation). Responding fast enough to compensate for aircraft motion limits the slowest speed.

The weights in an adaptive array may be controlled by any one of a variety of different algorithms. The "best" algorithm for a given application is chosen on the basis of a host of factors including the signal structures, the a priori information available to the adaptive processor, the performance characteristics to be optimized, the required speed of response of the processor, the allowable circuit complexity, any device or other technological limitations, and cost-effectiveness.

Referring to Figure 1-1, the received signal impinges on the sensor array and arrives at each sensor at different times as determined by the direction of arrival of the signal and the spacing of the sensor elements. The actual received signal for many applications consists of a modulated carrier whose information-carrying component consists only of the complex envelope. If $s(t)$ denotes the modulated carrier signal, then $\tilde{s}(t)$ is commonly used to denote the complex envelope of $s(t)$ (as explained in Appendix B) and is the only quantity that conveys information. Rather than adopt complex envelope notation, however, it is simpler to assume that all signals are represented by their complex envelopes so the common carrier reference never appears explicitly. It is therefore seen that each of the N channel signals $x_k(t)$ represents the complex envelope of the output of the element of a sensor array that is composed of a signal component and a noise component, that is,

$$x_k(t) = s_k(t) + n_k(t), \quad k = 1, 2, \ldots, N \tag{1.1}$$

In a linear sensor array having equally spaced elements and assuming ideal propagation conditions, the $s_k(t)$ are determined by the direction of the desired signal. For example, if

the desired signal direction is located at an angle θ from mechanical boresight, then (for a narrowband signal)

$$s_k(t) = s(t) \exp\left\{ j \frac{2\pi k d}{\lambda} \sin\theta \right\} \tag{1.2}$$

where d is the element spacing, λ is the wavelength of the incident planar wavefront, and it is presumed that each of the sensor elements is identical.

For the beamforming network of Figure 1-1, the adaptive array output signal is written as

$$y(t) = \sum_{k=1}^{N} w_k x_k(t) \tag{1.3}$$

Equation (1.3) can be conveniently expressed in matrix notation as

$$y(t) = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w} \tag{1.4}$$

where the superscript T denotes transpose, and the vectors $\mathbf{w}$ and $\mathbf{x}$ are given by

$$\mathbf{w}^T = [w_1 w_2 \ldots w_N] \tag{1.5}$$

$$\mathbf{x}^T = [x_1 x_2 \ldots x_N] \tag{1.6}$$

Throughout this book the boldface lowercase symbol (e.g., $\mathbf{a}$) denotes a vector, and a boldface uppercase symbol (e.g., $\mathbf{A}$) denotes a matrix.

The adaptive processor must select the complex weights, w_k, to optimize a stipulated performance criterion. The performance criterion that governs the operation of the adaptive processor is chosen to reflect the steady-state performance characteristics that are of concern. The most popular performance measures that have been employed include the mean square error [9,28–31]; SNR ratio [6,14,32–34]; output noise power [35]; maximum array gain [36,37]; minimum signal distortion [38,39]; and variations of these criteria that introduce various constraints into the performance index [16,40–43]. In Chapter 3, selected performance measures are formulated in terms of the signal characterizations of (1.1)–(1.4). Solutions are found that determine the optimum choice for the complex weight vector and the corresponding optimum value of the performance measure. The operational signal environment plays a crucial role in determining the effectiveness of the adaptive array to operate. Since the array configuration has pronounced effects on the resulting system performance, it is useful to consider sensor spacing effects before proceeding with an analysis using an implicit description of such effects. The consideration of array configuration is undertaken in Chapter 2.

1.5 EXISTING TECHNOLOGY

In any echo-ranging system, the maximum detection range $R_{\max}$ determines the minimum period between consecutive pulses $T_{\min}$ [and hence the pulse repetition frequency (PRF)], according to

$$T_{\min} = \frac{2R_{\max}}{\mathfrak{v}} \tag{1.7}$$

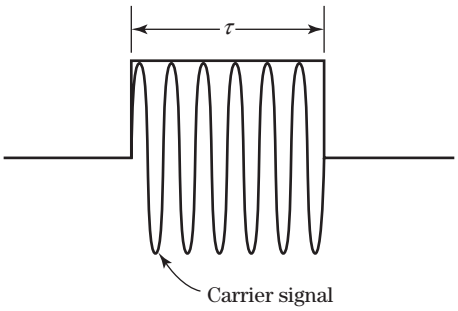


FIGURE 1-2 ■ Pulse modulated carrier signal.

where $\mathfrak{v}$ is the velocity of propagation of the transmitted signal. For underwater applications the velocity of sound in water varies widely with temperature, although a nominal value of 1,500 m/sec can be used for rough calculations. The velocity of electromagnetic wave propagation in the atmosphere can be taken approximately to be the speed of light or 3×10^8 m/sec.

If the range discrimination capability between targets is to be r_d, then the maximum pulse length $t_{\max}$ (in the absence of pulse compression) is given by

$$t_{\max} = \frac{2r_d}{\mathfrak{v}} \tag{1.8}$$

It will be noted that r_d also corresponds to the "blind range"— that is, the range within which target detection is not possible. Since the signal bandwidth $\cong 1/\text{pulse length}$, the range discrimination capability determines the necessary bandwidth of the transducers and their associated electrical channels.

The transmitted pulses form a pulse train in which each pulse modulates a carrier frequency as shown in Figure 1-2. The carrier frequency f_0 in turn determines the wavelength of the propagated wavefront since

$$\lambda_0 = \frac{\mathfrak{v}}{f_0} \tag{1.9}$$

where λ_0 is the wavelength. For sonar systems, frequencies in the range 100–100,000 Hz are commonly employed [44], whereas for radar systems the range can extend from a few megahertz up into the optical and ultraviolet regions, although most equipment is designed for microwave bands between 1 and 40 GHz. The wavelength of the propagated wavefront is important because the array element spacing (in units of λ) is an important parameter in determining the array pattern.

1.5.1 Radar Technology

There has been a steady increase in the demand for increased radar system performance and additional capability for both military and civilian purposes, and the vast number of applications of modern radar technology precludes anything more than the briefest mention of the major areas in which radar systems are found [45]. Military applications may very well involve a number of requirements that in the past would have involved a separate radar system for each different requirement. For example, a fire control system radar may be required to search large volumes of space, to detect and track both high- and low-speed targets ranging from very low to extremely high altitudes, to provide fire

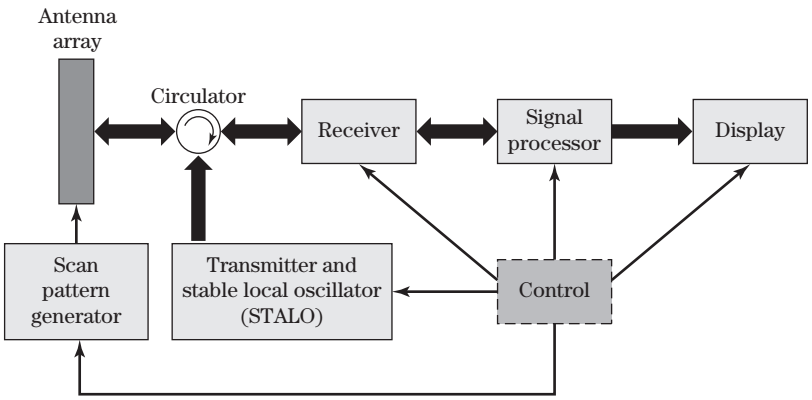


FIGURE 1-3 ■ Simplified block diagram of a typical radar system.

control for both missiles and guns against both airborne and ground (or sea) targets, and additionally to provide navigational aid and perform reconnaissance. Current civil aeronautical needs include air traffic control, collision avoidance, instrument approach systems, weather sensing, and navigational aids. Additional applications in the fields of law enforcement, transportation, and Earth resources are just beginning to grow to sizable proportions [48].

Figure 1-3 is a block diagram of a typical radar system. These major blocks and their corresponding functions are described in Table 1-1 [46]. The antenna, receiver, and signal processing blocks are of primary interest for our purposes, and these are now each briefly discussed in turn.

1.5.1.1 Radiating Elements and Antenna Arrays

The vast frequency range and power over which modern radar systems operate have led to an astonishing variety of radiator elements ranging from parabolic reflectors to horns, dipoles, bow ties [47], multiturn loops [48], spirals [49], log periodics [50], microstrip patches [51], and Vivaldis [52]. Large antenna apertures result in narrow beamwidths that are required for long-range detection and high resolution if targets close to one another are to be distinguished. Microwave frequencies are by far the most popular for radar applications since antenna apertures of relatively small physical size (but large in terms of wavelengths) are reasonable to build.

The antenna type selected for a radar application usually differs from one selected for a communications system. Shaped beam patterns that can be scanned are most popular for radar uses, whereas most communication applications require beams designed for omnidirectional coverage or for fixed point-to-point transmission. The earliest radars (developed

TABLE 1-1 ■ Functions of the Radar Blocks in Figure 1-3

Block	Function
Transmitter	Generates high power RF waveform
Antenna	Determines direction and shape of transmit-and-receive beam
Receiver	Provides frequency conversion and low-noise amplification
Signal processing	Provides target detections, target and clutter tracking, and target trajectory estimates
Display	Converts processed signals into meaningful tactical information

FIGURE 1-4 ■ FuG 65 Wurzburg Riese radar antenna (Courtesy of the National Electronics Museum).

during World War II) operated in the very high frequency (VHF) and ultra high frequency (UHF) bands. Sometimes, a parabolic dish was used, such as the FuG 65 Wurzburg Riese radar antenna in Figure 1-4. Its 3 m parabolic dish operated at 560 MHz and was used to guide German intercept fighters during WWII [53]. Arrays, such as the SCR-270 in Figure 1-5, were used by the United States in WWII for air defense [54]. It has four rows of eight dipoles that operate at 110 MHz. After WWII, radar and communications systems began operating at higher frequencies. Different types of antennas were tried for various systems. A microwave lens was used in the Nike AJAX MPA-4 radar shown in Figure 1-6 [55]. In time, phased array antennas became small enough to place in the nose of fighter airplanes. In the 1980s, the AN/APG-68 (Figure 1-7) was used in the F-16 fighter [56]. The array is a planar waveguide with slots for elements. Active electronically scanned arrays (AESA) provide fast wide angle scanning in azimuth and elevation and include advanced transmit/receive modules [57]. An example is the AN/APG-77 array for the F-22 fighter shown in Figure 1-8.

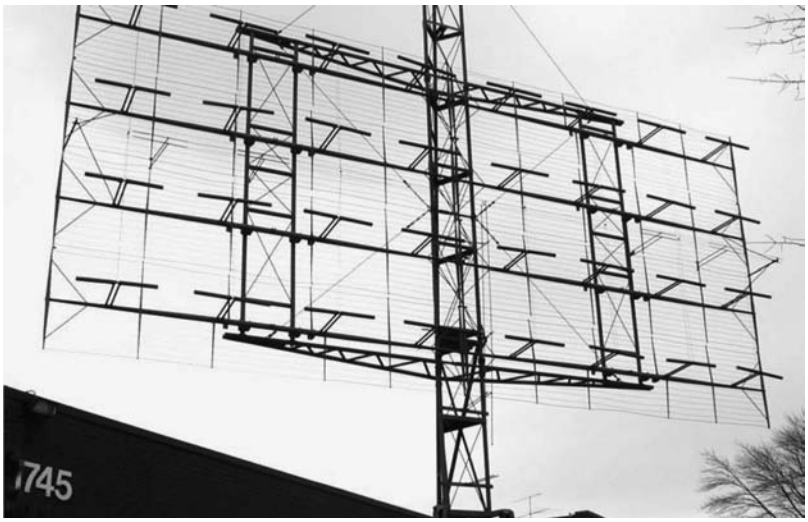


FIGURE 1-5 ■ SCR-270 antenna array (Courtesy of the National Electronics Museum).

FIGURE 1-6 ■ Waveguide lens antenna for the Nike AJAX MPA-4 radar (Courtesy of the National Electronics Museum).

The two most common forms of antenna arrays for radar applications are the linear array and the planar array. A linear array consists of antenna elements arranged in a straight line. A planar array, on the other hand, is a two-dimensional configuration in which the antenna elements are arranged to lie in a plane. Conformal arrays lie on a nonplanar surface. The linear array generates a fan beam that has a broad beamwidth in one plane and a narrow beamwidth in the orthogonal plane. The planar array is most frequently used in radar applications where a pencil beam is needed. A fan-shaped beam is easily produced by a rectangular-shaped aperture. A pencil beam may easily be generated by a square- or circular-shaped aperture. With proper weighting, an array can be made to simultaneously generate multiple search or tracking beams with the same aperture.

Array beam scanning requires a linear phase shift across the elements in the array. The phase shift is accomplished by either software in a digital beamformer or by hardware phase shifters. A phase shifter is often incorporated in a transmit/receive module. Common phase-shifter technology includes ferroelectrics, monolithic microwave integrated

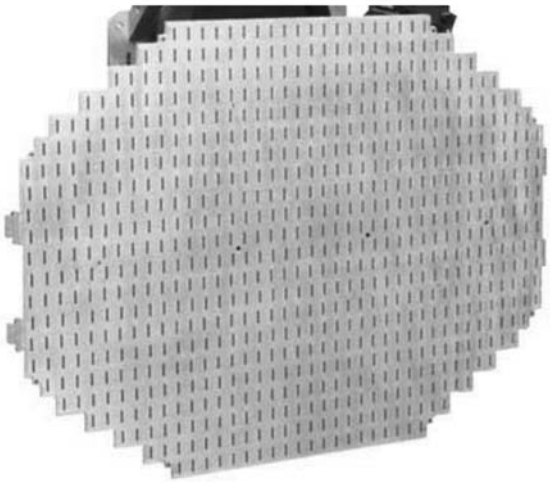


FIGURE 1-7 ■ AN/APG-68 array (Courtesy of Northrop Grumman and available at the National Electronics Museum).

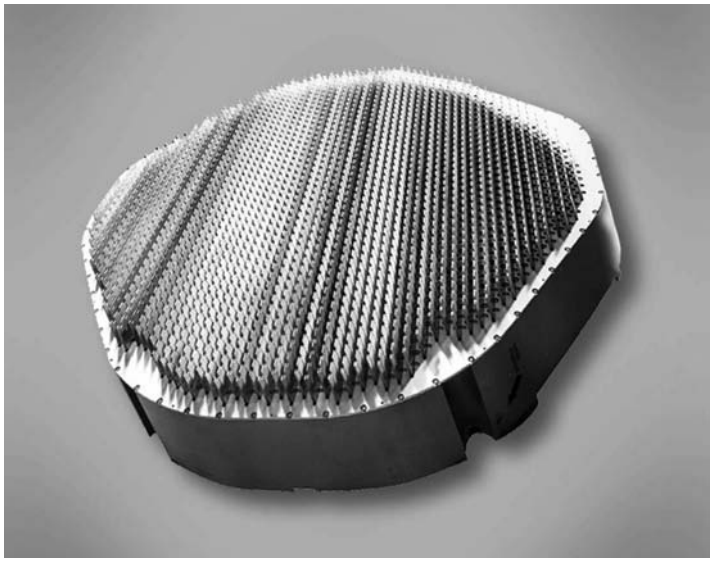


FIGURE 1-8 ■ AN/APG-77 array (Courtesy of Northrop Grumman and available at the National Electronics Museum).

circuit (MMIC), ferrites, and micro-electro-mechanical systems (MEMS). Some of their characteristics are shown in Table 1.2.

1.5.1.2 Receivers

A receiver design based on a matched filter or a cross-correlator maximizes the SNR in the linear portion of the receiver. Different types of receivers that have been employed in radar applications include the superheterodyne, superregenerative, crystal video, and tuned radio frequency (TRF) [58]. The most popular and widely applied receiver type is the superheterodyne, which is useful in applications where simplicity and compactness are especially important. A received signal enters the system through the antenna, then passes through the circulator and is amplified by a low-noise RF amplifier. Following RF amplification, a mixer stage is entered to translate the RF to a lower intermediate frequency (IF) where the necessary gain is easier to obtain and filtering is easier to synthesize. The gain and filtering are then accomplished in an IF amplifier section.

1.5.1.3 Signal Processing

Having maximized the SNR in the receiver section, the next step is to perform two basic operations by signal processing as follows: (1) detection of the presence of any targets, and

TABLE 1-2 ■ Phase-Shifter Characteristics [57]

Type	Ferroelectric	MMIC	Ferrite	MEMS
Cost	Low	High	Very high	Low
Reliability	Good	Very good	Excellent	Good
Power handling	> 1 W	> 10 W	kW	< 50 mW
Switch speed	ns	ns	10 to 100 μs	10 to 100 μs
Direct current power consumption	Low	low	High	Negligible
Size	Small	Small	Large	Small

(2) extraction of information from the received waveform to obtain target trajectory data such as position and velocity. Detecting a signal imbedded in a noise field is treated by means of statistical decision theory. Similarly, the problem of the extraction of information from radar signals can be regarded as a problem concerning the statistical estimation of parameters.

1.5.2 Sonar Technology

Operational active sonar systems may be classified as (1) search light sonar, (2) scanning sonar, or (3) rotational directional transmission (RDT) sonar [59]. Searchlight sonar has very narrow transmit and receive beams and provides an azimuth search capability by mechanically rotating the directional hydrophone array. Since the array is mechanically scanned and aimed, the data rate is correspondingly low, and the system does not provide a multiple-target detection and tracking capability. The requirement for mechanical directional training also limits the array size, so that operational frequencies are usually greater than 15 kHz, thereby increasing attenuation loss.

Scanning sonar systems overcome the data rate limitation of searchlight sonars by transmitting an omnidirectional, short-duration pulse and using electronic means to rapidly rotate a narrow receive beam continuously over a 360° azimuthal sector. The receiving beam output is presented to a panoramic display called a plan position indicator (PPI) that is used extensively in both radar and sonar systems. Scanning-type sonar systems thereby provide a multiple-target detection and tracking capability, and lower operating frequencies can be used, thereby decreasing attenuation losses. The scan speed of the receive beam is a compromise between the desired target resolution and the maximum receiver bandwidth (or minimum input SNR) that is permissible.

An RDT sonar system is characterized by RDT and a scanned preformed beam (PFB) receiver. Consequently, high transmitting directivity and a high data rate are accompanied by a low operational frequency. An RDT system combines the best features of searchlight and scanning sonars. A PFB receiver can have a smaller bandwidth than a scanning receiver, thereby improving the SNR. Furthermore, a PFB receiver can be corrected for Doppler due to own ship's motion employing a method called own Doppler nullifying (ODN), whereas a scanning receiver cannot.

The principal elements of a sonar receiver (Figure 1-9) are the hydrophone array, the beamformer, the signal processor, and the information processor. Each of these principal elements (except for the information processor, which involves display formatting and other command and control functions) is briefly discussed in turn.

1.5.2.1 Sonar Transducers and Hydrophones

A *hydrophone* produces an output voltage proportional to the acoustic signals incident on it; whereas a transducer generates and receives sound. For underwater applications, a very wide frequency range is involved—from about 10 Hz to more than 1 MHz [60].

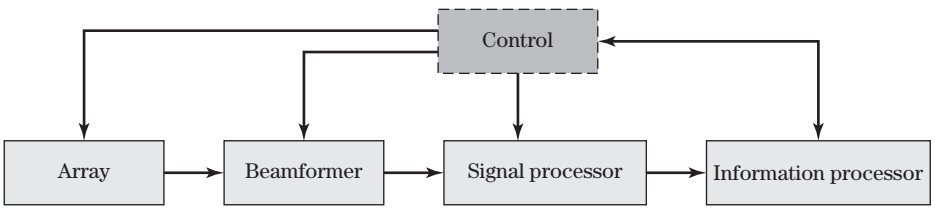


FIGURE 1-9 ■ Sonar receiver block diagram.

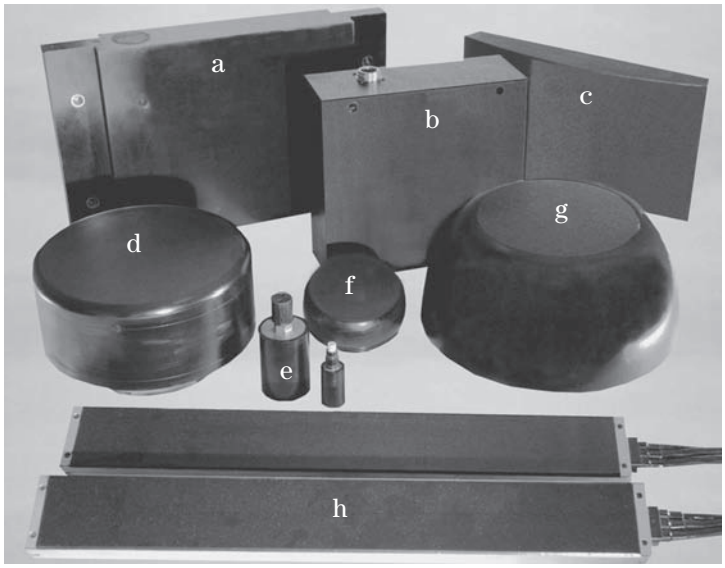


FIGURE 1-10 ■ Various acoustic transducers. a: Seabed mapping—20 kHz multibeam receive array module; 8 shaded elements per module; 10 modules per array. b: Subbottom profiling (parametric)—200 kHz primary frequency; 25 kHz secondary frequency. c: Port and harbor Security—curved 100 kHz transmit/receive array. d. Obstacle avoidance—10 × 10 planar receive array with curved transmitter. e. ACOMMS—Broadband piezocomposite transducers for wideband communication signals. f. AUV FLS—high-frequency, forward-looking sonar array. g. Mine hunting—10 × 10 transmit/receive broadband array; available with center frequencies between 20 kHz to 1MHz h. Side scan—multibeam transmit/receive array (Courtesy of Materials Systems Inc.).

A transmitting power ranging from a few acoustic watts up to several thousand acoustic watts at ocean depths up to 20,000 ft can be achieved [61]. Figure 1-10 shows a sampling of different acoustic transducers manufactured by Materials Systems Inc. for various applications.

The basic physical mechanisms most widely used in transducer technology include the following [62]:

1. Moving coil. This is long familiar from use as a loudspeaker in music reproduction systems and used extensively in water for applications requiring very low frequencies.
2. Magnetorestrictive. Magnetic materials vibrate in response to a changing magnetic field. Magnetorestrictive materials are rugged and easily handled, and magnetorestrictive transducers were highly developed and widely used during World War II.
3. Piezoelectric. The crystalline structure of certain materials results in mechanical vibration when subjected to an alternating current or an oscillating electric field. The relationship between mechanical strain and electric field is linear. Certain ceramic materials also exhibit a similar effect and have outstanding electromechanical properties. Consequently, over the last decade the great majority of underwater sound transducers have been piezoceramic devices that can operate over a wide frequency band and have both high sensitivity and high efficiency.
4. Electrostrictive. Similar to piezoelectric but has a nonlinear relationship between mechanical strain and electric field.

5. Electrostatic. These capacitive transducers use the change in force between two charged parallel plates due to mechanical movement. These have found use with MEMS but not in underwater acoustics.
6. Variable reluctance and hydroacoustic transducers have also been used for certain experimental and sonar development work, but these devices have not challenged the dominance of piezoceramic transducers for underwater sound applications [59].

1.5.2.2 Sonar Arrays

Sonar transducer arrays have low sidelobes and beams that scan over wide angular sectors. Acoustic array configurations include linear, planar, cylindrical, spherical, conformal, volumetric, reflector, and acoustic lenses [63]. These different array types lend themselves to towing, conformal mounting on hulls (where the array surface conforms to the shape of the underwater hull, so that no appendage is required), beam steering, and side-looking sonar and synthetic-aperture applications. Figure 1-11 is an example of a 10×10 acoustic planar array. A circular acoustic array is shown in Figure 1-12. Sometimes, the array must be conformal to the surface on which it is mounted. Figure 1-13 shows a towfish with an acoustic communications (ACOMMS) receive array that operates from 10 to 40 kHz and a sidescan transducer that transmits and receives at 900 kHz. A towfish is a sidescan sonar that is towed underwater by a boat.

A simple 2 ft $\times$ 4 ft planar array having more than 500 sensor elements for deep submergence applications is shown in the diagram of Figure 1-14. In the quest for larger power and lower frequency (with attendant lower attenuation losses), some arrays are very large. In one case a 35 ft $\times$ 50 ft low-frequency planar array weighs 150 tons and transmits close to 10^6 watts [63]. Large arrays with many closely spaced elements develop "hot spots" from mutual coupling. Consequently, the concept of "velocity control" was developed [64,65] to protect individual transducer elements against extreme impedance variations.

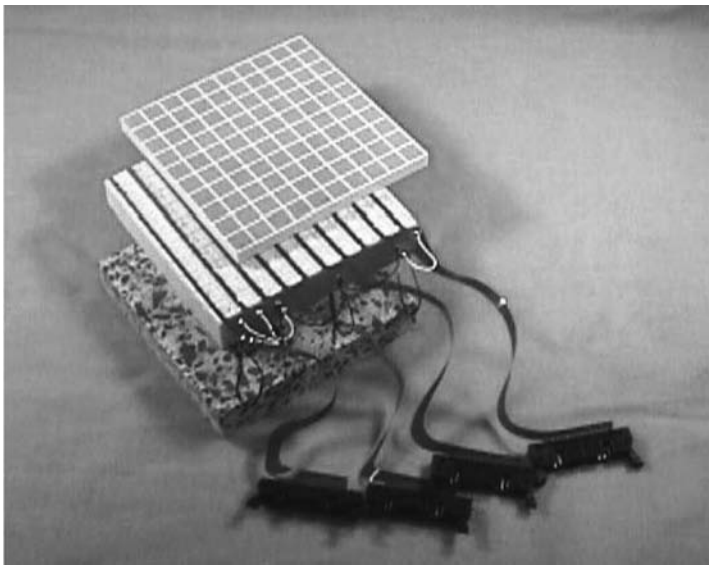


FIGURE 1-11 ■ Picture of a 100-element receive acoustic array manufactured in four layers: matching layer, piezocomposite, flex circuit, and absorbing back (Courtesy of Materials Systems Inc.).

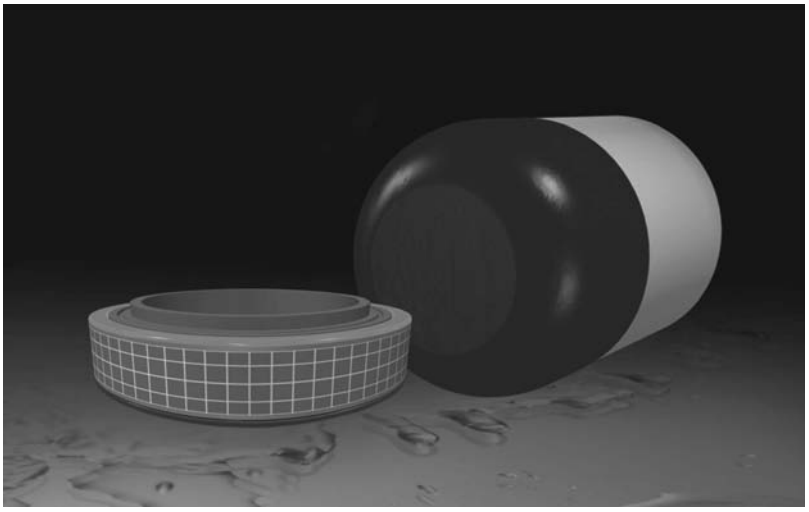


FIGURE 1-12 ■ Left—Port and harbor surveillance piezocomposite array; 100 kHz transmit/receive array. Right—Forward-looking piezocomposite sonar for AUV; piezocomposite facilitates broad bandwidth, high element count arrays, and curved geometries (Courtesy of Materials Systems Inc.).

Surface ships often use a bubble-shaped bow dome in which a cylindrical array is placed like that shown in Figure 1-15. This array uses longitudinal vibrator-type elements composed of a radiating front end of light weight and a heavy back mass, with a spring having active ceramic rings or disks in the middle [63]. The axial symmetry of a cylindrical array renders beam steering fairly simple with the azimuth direction in which the beam is formed, because the symmetry allows identical electronic equipment for the phasing and time delays required to form the beam. Planar arrays do not have this advantage,

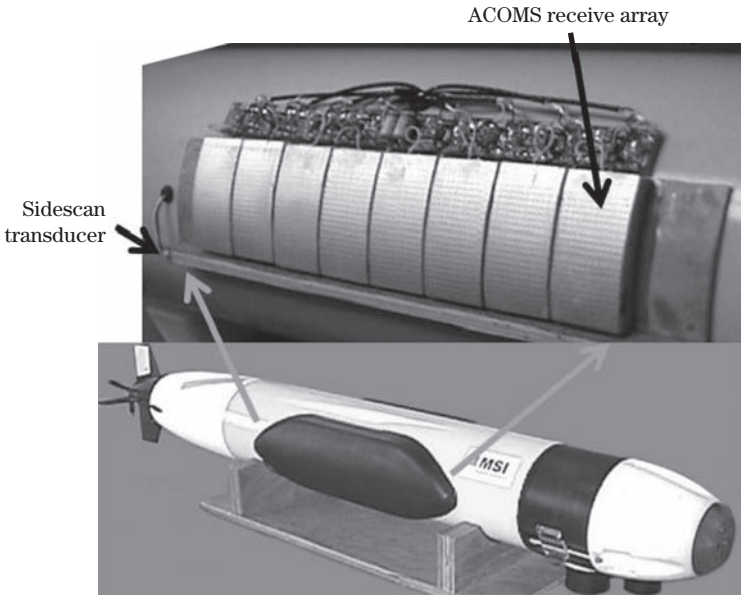


FIGURE 1-13 ■ The acoustic communications (ACOMMS) receive array operates from 10 to 40 kHz, and the sidescan transducer transmits and receives at 900 kHz (Courtesy of Materials Systems Inc.).

FIGURE 1-14 ■ Rectangular planar array for searchlight type sonar system having more than 500 elements.

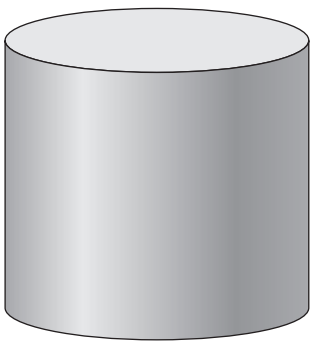


FIGURE 1-15 ■ Cylindrical sonar array used in bow-mounted dome.

since each new direction in space (whether in azimuth or in elevation) requires a new combination of electronic equipment to achieve the desired pointing.

A spherical array is the ideal shape for the broadest array coverage in all directions. Spherical arrays like that shown in the diagram of Figure 1-16 have been built with a diameter of 15 ft and more than 1,000 transducer elements. This spherical arrangement can be integrated into the bow of a submarine by means of an acoustically transparent dome that provides minimum beam distortion. For instance, the bow dome of a Virginia class submarine is a 25 ton hydrodynamically shaped composite structure that houses a sonar transducer sphere. The bow dome is 21 feet tall and has a maximum diameter of 26 feet. A two-inch thick, single-piece rubber boot is bonded to the dome to enhance acoustic performance. Minimal sound energy absorption and reflection properties inherent in the rubber material minimally reflect and absorb acoustic signals. Figure 1-17 shows a spherical microphone array that was constructed by placing a rigid spherical array at the center of a larger open spherical array [66]. Both arrays have 32 omnidirectional microphones and a relatively constant directivity from about 900 Hz to 16 kHz.

FIGURE 1-16 ■ Spherical array having 15 ft diameter and more than 1,000 transducer elements.

FIGURE 1-17 ■ A dual, concentric SMA. (A. Parthy, C. Jin, and A. van Schaik, "Acoustic holography with a concentric rigid and open spherical microphone array," IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 2173–2176.)

1.5.2.3 Beamformer

Beamforming ordinarily involves forming multiple beams from multielement arrays through the use of appropriate delay and weighting matrices. Such beams may be directionally fixed or steerable. After that, sonar systems of the 1950s and 1960s consisted largely of independent sonar sets for each transducer array. More recently, the sophisticated use of multiple sensors and advances in computer technology have led to integrated sonar systems that allow the interaction of data from different sensor arrays [67,68]. Such integrated sonar systems have software delays and weighting matrices, thereby generalizing the structure of digital time domain beamformers. Consequently several units of a single (programmable) beamformer design may be used for all the arrays in an integrated system. Furthermore, programmable beamformer matrices make it possible to adapt the receive pattern to the changing structure of the masking noise background.

1.5.2.4 Signal Processor

Signal processing involves filtering, spectral analysis, correlation, and the related operations of dynamic range compression and normalization (DRCN) that "match" the received array signals to the display/decision functions contained in the information processing block of Figure 1-9 [59,69]. The DRCN removes some of the spatial and temporal characteristics of the acoustic channel that are impressed on the propagating signal. Whether a sufficient degree of DRCN is achieved usually determines whether a system design will succeed or fail since it specifies the integration of a theoretically designed optimum signal processor to its system interfaces.

1.6 ORGANIZATION OF THE BOOK

1.6.1 Part 1

Chapter 2 introduces the adaptive array concept by first considering the nature of the signal environment in which an adaptive array is expected to improve the overall reception of a desired signal. The question of array architecture is considered, and the impact of the architecture selection on the ultimate system performance that can be realized is discussed. The potential of an array to enhance the output SNR performance by adjusting the beamforming network is demonstrated.

An adaptive array manipulates the received signals to meet a defined performance measure. This performance measure is an objective function to be maximized (or minimized). Several widely used performance measures for both narrowband and broadband applications are presented in Chapter 3 that are formulated in terms of the elements found within the adaptive array functional model.

Any adaptive array system must have performance limits:

1. Imposed by the array physical structure.
2. Resulting from the nature of the signal environment

The consideration of performance limits leads to a discussion of the array performance that results after the automatic adaptation process has been permitted to operate long enough to reach a steady-state solution. A steady-state analytic solution to the adaptive array control problem can be found for each performance measure, which enables the designer to determine ultimate system performance limits. The mathematical foundation required to develop the steady-state analytic solution for various performance measures is laid, and the relationships among the solutions obtained to the optimal "Wiener solution" are presented.

1.6.2 Part 2

The heart of the adaptive capability within an adaptive array system is the adaptive algorithm that adjusts the array pattern in response to the signal information found at the sensor element outputs. Part 2, including Chapters 4 through 8, introduces different classes of adaptation algorithms. In some cases adaptation algorithms are selected according to the kind of signal information available to the receiver:

1. The desired signal is known.
2. The desired signal is unknown, but its direction of arrival is known.

3. The desired signal is unknown and its direction of arrival is known.
4. No signal information is available at the outset, but as array operation progresses such information must be "learned" by the adaptive processor. These algorithms are called blind adaptive algorithms.

The selection of an adaptive algorithm has important consequences for the system transient performance.

The convergence properties of the various adaptation algorithms are analyzed and performance comparisons are made. Furthermore, the shortcomings of the algorithms under certain conditions are discussed. These results are summarized for convenience and provide the designer with a means for assessing which candidate algorithms are most appropriate for the signal conditions and system requirements.

1.6.3 Part 3

The adaptive array operating conditions considered so far were nonideal only in that interference signals were present with which the array had to contend. In actual practice, however, the effects of several other nonideal operating conditions often result in unacceptable degradation of array performance unless compensation of such effects is undertaken. Such nonideal operating conditions include processing of broadband signals, multipath effects, channel mismatching, and array propagation delay effects. Compensation for these factors by means of tapped delay-line processing is considered, and the question of how to design a tapped delay line to achieve a desired degree of compensation is addressed. Finally, current trends in adaptive array research that provide an indication of the direction that future developments are likely to take are discussed.

1.7 SUMMARY AND CONCLUSIONS

The motivation for and actual use of adaptive array systems are presented. The principal elements of an adaptive array system are defined, and the fundamental problems facing an adaptive array designer are given. Adaptive array design is a compromise among such factors as [70]:

1. Hardware complexity and cost
2. Data rate
3. Maximum range of detection (for radar and sonar)
4. Resolution in angle (and range and Doppler for radar and sonar)
5. Precision in the measurement of range, bearing, and Doppler (for radar and sonar)
6. Ability of the adaptive array to meet both transient and steady-state system performance requirements

A suboptimal acoustical array processor known as the DICANNE processor operated in sea tests against ship-generated interferences and consistently formed cancellation nulls 10–15 dB deep [59]. Use of an optimal wideband processor based on the minimum signal distortion performance measure in a computer-simulated sonar experiment resulted in effectively suppressing a strong coherent interfering signal by forming cancellation nulls 50 dB deep [59]. Such deep cancellation nulls were found, however, to be quite sensitive to

(1) small changes in interference signal bearing, (2) small errors in the adaptive weight values, and (3) statistical fluctuations of measured correlations due to finite integration time.

A lightweight four-element adaptive array using hybrid microwave integrated circuitry and weighing only 1 pound, intended for communication applications, was built and tested [71]. This unit employed a null-steering algorithm appropriate for a coherent sidelobe canceller and succeeded in forming broadband nulls over a 60–100 MHz bandwidth having a cancellation depth of 25–30 dB under weak desired signal and strong interference signal conditions. To attain this degree of interference signal cancellation, it was essential that the element channel circuitry be very well matched over a 20% bandwidth.

Another experimental four-element adaptive array system for eliminating interference in a communication system was also tested [48]. Pattern nulls of 10–20 db for suppressing interference signals over a 200–400 MHz band were easily achieved so long as the desired signal and interference signal had sufficient spatial separation (greater than the resolution capability of the antenna array), assuming the array has no way to distinguish between signals on the basis of polarization. Exploiting polarization differences between desired and interference signals by allowing full polarization flexibility in the array, an interference signal located at the same angle as the desired signal can be suppressed without degrading the reception of the desired signal. Yet another system employing digital control was developed for UHF communications channels and found capable of suppressing jammers by 20–32 dB [72].

In summary, interference suppression levels of 10–20 dB are consistently achieved in practice. It is more difficult but nevertheless practicable to achieve suppression levels of 20–35 dB and usually very difficult to form cancellation nulls greater than 35 dB in a practical operating system.

The rapid development of digital technology is presently having the greatest impact on signal reception systems. The full adaptation of digital techniques into the processing and interpretation of received signals is making possible the realization of practical signal reception systems whose performance approaches that predicted by theoretical limits. Digital processors and their associated memories have made possible the rapid digestion, correlation, and classification of data from larger search volumes, and new concepts in the spatial manipulation of signals have been developed. Adaptive array techniques started out with limited numbers of elements in the arrays, and the gradual increase in the numbers of elements and in the sophistication of the signal processing will likely result in an encounter with techniques employed in optical and acoustical holography [69,73]. Holography techniques are approaching such an encounter from the other direction, since they start out with a nearly continuous set of spatial samples (as in optical holography) and move down to a finite number of samples (in the case of acoustic holography).

1.8 PROBLEMS

1. **Radar Pulse Waveform Design** Suppose it is desired to design a radar pulse waveform that would permit two Ping-Pong balls to be distinguished when placed only 6.3 cm apart in range up to a maximum range from the radar antenna of 10 m.

   (a) What is the maximum PRF of the resulting pulse train?

   (b) What bandwidth is required for the radar receiver channel?

   (c) If it is desired to maintain an array element spacing of $d = 2$ cm where $d = \lambda_0/2$, what pulse carrier frequency should the system be designed for?

2. **Sonar Pulse Carrier Frequency Selection** In the design of an actual sonar system many factors must be considered—all the sonar parameters (e.g., source level, target strength) and the environment parameters. The effect of environmental parameters depends largely on frequency. Suppose in a highly oversimplified example that only the factors of transmission loss (due to attenuation) and ambient noise are of concern. Let the attenuation coefficient α be given by

$$\log_{10}(\alpha) = \frac{1}{4}[-21 + 5\log_{10}(f)]$$

Furthermore, let the ambient noise spectrum level N_0 be given by

$$10\log_{10}(N_0) = \frac{1}{3}[20 - 50\log_{10}(f)]$$

If the cost to system performance is given by $J = C_1\alpha + C_2N_0$ where C_1 and C_2 denote the relative costs of attenuation and noise to the system, what value of pulse carrier frequency f should be selected to optimize the system performance?

1.9 REFERENCES

[1] L. C. Van Atta, "Electromagnetic Reflection," U.S. Patent 2908002, October 6, 1959.

[2] *IEEE Trans. Antennas Propag.* (Special Issue on Active and Adaptive Antennas), Vol. AP-12, March 1964.

[3] D. L. Margerum, "Self-Phased Arrays," in *Microwave Scanning Antennas*, Vol. 3, *Array Systems*, edited by R. C. Hansen, Academic Press, New York, 1966, Ch. 5.

[4] P. W. Howells, "Intermediate Frequency Sidelobe Canceller," U.S. Patent 3202990, August 24, 1965.

[5] P. W. Howells, "Explorations in Fixed and Adaptive Resolution at GE and SURC," *IEEE Trans. Antennas Propag.*, Special Issue on Adaptive Antennas, Vol. AP-24, No. 5, pp. 575–584, September 1976.

[6] S. P. Applebaum, "Adaptive Arrays," Syracuse University Research Corporation, Rep. SPL TR66-1, August 1966.

[7] B. Widrow, "Adaptive Filters I: Fundamentals," Stanford University Electronics Laboratories, System Theory Laboratory, Center for Systems Research, Rep. SU-SEL-66-12, Tech. Rep. 6764-6, December 1966.

[8] B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, edited by R. E. Kalman and N. DeClaris, Holt, Rinehard and Winston, New York, 1971, Ch. 5.

[9] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, No. 12, p. 2143–2159, December 1967.

[10] L. Spitzer, Jr., "Basic Problems of Underwater Acoustic Research," National Research Council, Committee on Undersea Warfare, Report of Panel on Underwater Acoustics, September 1, 1948, NRC CUW 0027.

[11] R. H. Bolt and T. F. Burke, "A Survey Report on Basic Problems of Acoustics Research," Panel on Undersea Warfare, National Research Council, 1950.

[12] H. Mermoz, "Filtrage adapté et utilisation optimale d'une antenne," Proceedings NATO Advanced Study Institute, September 14–26, 1964. "Signal Processing with Emphasis on Underwater Acoustics" (Centre d'Etude des Phénomenès Aléatoires de Grenoble, Grenoble, France), 1964, pp. 163–294.

[13] F. Bryn, "Optimum Signal Processing of Three-Dimensional Arrays Operating on Gaussian Signals and Noise," *J. Acoust. Soc. Am.*, Vol. 34, March 1962, pp. 289–297.

[14] S. W. W. Shor, "Adaptive Technique to Discriminate Against Coherent Noise in a Narrowband System," *J. Acoust. Soc. Am.*, Vol. 39, January 1966, pp. 74–78.

[15] P. E. Green, Jr., R. A. Frosch, and C. F. Romney, "Principles of an Experimental Large Aperture Seismic Array (LASA)," *Proc. IEEE*, Vol. 53, December 1965, pp. 1821–1833.

[16] R. T. Lacoss, "Adaptive Combining of Wideband Array Data for Optimal Reception," *IEEE Trans. Geosci. Electron.*, Vol. GE-6, May 1968, pp. 78–86.

[17] B. D. Steinberg, "On Teleseismic Beam Formation by Very Large Arrays," *Bull. Seismol. Soc. Am.*, August 1971.

[18] *IEEE Trans. Antennas Propag.* (Special Issue on Adaptive Antennas), Vol. AP-24, September 1965.

[19] A. J. Berni, "Angle of Arrival Estimation Using an Adaptive Antenna Array," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-11, No. 2, March 1975, pp. 278–284; see errata *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-11, May and July 1975.

[20] R. C. Davis, L. E. Brennan, and I. S. Reed, "Angle Estimation with Adaptive Arrays in External Noise Fields," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-12, No. 2, March 1976, pp. 179–186.

[21] B. D. Steinberg, *Principles of Aperture and Array System Design*, Wiley, New York, 1976, Ch. 11.

[22] B. D. Steinberg, "Design Approach for a High-Resolution Microwave Imaging Radio Camera," *J. Franklin Inst.*, Vol. 296, No. 6, December 1973.

[23] E. N. Powers and B. D. Steinberg, "Valley Forge Research Center Adaptive Array," USNC/URSI Annual Meeting, Boulder, CO, October 1975.

[24] E. Biglieri, *MIMO Wireless Communications*, Cambridge, Cambridge University Press, 2007.

[25] M. H. Er, and S. K. Hui, "Beamforming in Presence of Element Failure," *Electronics Letters,* vol. 27, no. 3, pp. 273–275, 1991.

[26] J. T. Aberle, O. Sung-Hoon, D. T. Auckland, et al., "Reconfigurable Antennas for Wireless Devices," *Antennas and Propagation Magazine, IEEE,* vol. 45, no. 6, pp. 148–154, 2003.

[27] C.C. Chang, R.H. Lee, and T.Y. Shih, "Design of a beam switching/steering Butler matrix for phased array system," *IEEE Transactions on Antennas and Propagation,* vol. 58, no. 2, pp. 367–374, Feb. 2010.

[28] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. IEEE*, Vol. 57, No. 10, pp. 1696–1704, October 1969.

[29] J. Chang and F. Tuteur, "Optimum Adaptive Array Processor," Proceedings of the Symposium on Computer Processing in Communications, April 8–10, 1969, Polytechnic Institute of Brooklyn, pp. 695–710.

[30] O. L. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proc. IEEE*, Vol. 60, No. 8, pp. 926–935, August 1972.

[31] C. A. Baird, "Recursive Processing for Adaptive Arrays," *Proceedings of Adaptive Antenna Systems Workshop,* Vol. 1, Naval Research Laboratory, Washington, DC, September 27, 1974, pp. 163–180.

[32] R. T. Adams, "An Adaptive Antenna System for Maximizing Signal-to-Noise Ratio," Wescon Conference Proceedings, Session 24, 1966, pp. 1–4.

[33] L. J. Griffiths, "Signal Extraction Using Real-Time Adaptation of a Linear Multichannel Filter," SEL-68-017, Tech. Rep. No. 6788-1, System Theory Laboratory, Stanford University, February 1968.

[34] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, March 1971, pp. 254–262.

[35] C. A. Baird, Jr. and J. T. Rickard, "Recursive Estimation in Array Processing," Proceedings of the Fifth Asilomar Conference on Circuits and Systems, 1971, Pacific Grove, CA, pp. 509–513.

[36] J. J. Faran and R. Hills, Jr., "Wide-Band Directivity of Receiving Arrays," Harvard University Acoustics Research Laboratory, Tech. Memo 31, May 1, 1953.

[37] H. H. Az-Khatib and R. T. Compton, Jr., "A Gain Optimizing Algorithm for Adaptive Arrays," *IEEE Trans. Antennas Propag.*, Vol. AP-26, No. 2, March 1978, pp. 228–235.

[38] N. Wiener, *Extrapolation, Interpretation, and Smoothing of Stationary Time Series*, New York, MIT Press, 1949.

[39] H. L. Van Trees, "Optimum Processing for Passive Sonar Arrays," *IEEE 1966 Ocean Electronics Symposium,* August 29–31, Honolulu, HI, pp. 41–65.

[40] K. Takao, M. Fujita, and T. Nishi, "An Adaptive Antenna Array Under Directional Constraint," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 662–669.

[41] J. Capon, R. J. Greenfield, and R. J. Kolker, "Multi-Dimensional Maximum-Likelihood Processing of a Large Aperture Seismic Array," *Proc. IEEE*, Vol. 55, February 1967, pp. 192–217.

[42] J. F. Clearbout, "A Summary, by Illustration, of Least-Square Filter With Constraints," *IEEE Trans. Inf. Theory*, Vol. IT-14, March 1968, pp. 269–272.

[43] A. Booker and C. Y. Ong, "Multiple-Constraint Adaptive Filtering," *Geophysics*, Vol. 36, June 1971, pp. 498–509.

[44] R. L. Deavenport, "The Influence of the Ocean Environment . . . ," *EASCON 1975 Record, IEEE Electronics and Aerospace Systems Convention,* September 29–October 1, Washington, DC, pp. 66. A - 66E.

[45] E. Brookner, "Now: Phased-array Radars: Past, Astounding Breakthroughs and Future Trends, January 2008," *Microwave Journal,* Vol. 51, no. 1, pp. 31–48, 2008.

[46] W. G. Bruner, W. N. Case, and R. C. Leedom, "Digital Control in Modern Radar," *EASCON 1971 Record, IEEE Electronics and Aerospace Systems Convention,* October 6-8, 1971, Washington, DC, pp. 200–207.

[47] R. T. Compton, Jr., "An Experimental Four-Element Adaptive Array," *IEEE Trans. Antennas & Prop.,* Vol AP-25, No. 5, September 1965, pp. 697–706.

[48] W. L. Curtis, "Spiral Antennas," *IRE Trans. Antennas & Prop*, Vol AP-8, May, 1960, pp. 298–306.

[49] R. H. DuHamel and D. G. Berry, "Logarithmically Periodic Antenna Arrays," *IRE WESCON Convention Record,* 1958, Vol. 2, Pt. 1, pp. 161–174.

[50] R. Garg, P. Bhartia, I. Bahl, and A. Ittipiboon, *Microstrip Antenna Design Handbook,* Norwood, MA, Artech House, 2001.

[51] P. J. Gibson, "the Vivaldi Aerial," *9th European Microwave Conf. Proc.,* 19779, pp. 103–105.

[52] National Electronics Museum, Wurzburg Riese Radar Antenna, 2009, Display Information.

[53] National Electronics Museum, SCR-270 Antenna, 2009, Display Information.

[54] National Electronics Museum, Nike AJAX MPA-4 Antenna, 2009, Display Information.

[55] National Electronics Museum, AN/APG-68 Antenna, 2009, Display Information.

[56] M. I. Skolnik, *Introduction to Radar Systems,* McGraw-Hill, New York, 1962, Ch. 2.

[57] C. H. Sherman, "Underwater Sound - A Review: I. Underwater Sound Transducers," *IEEE Trans. Sonics and Ultrason.,* Vol. SU-22, No. 5, September 1975, pp. 281–290.

[58] A. A. Winder, "Underwater Sound - A Review: II. SKonar System Technology," *IEEE Trans. Sonics and Ultrason.,* Vol. SU-22, No. 5, September 1975, pp. 291–332.

[59] E. A. Massa, "Deep Water trnasducers for Ocean Engineering Applications," *Proceedings IEEE,* 1966 Ocean Electronics Symposium, August 29–31, Honolulu, HI, pp. 31–39.

[60] C. H. Sherman and J. L. Butler, *Transducers and Arrays for Underwater Sound,* New York, Springer, 2007.

[61] T. F. Hueter, "Twenty Years in Underwater Acoustics: Generation and Reception," *J. Acoust. Soc. Am.,* Vol. 51, No 3 (Part 2), March 1972, pp. 1025–1040.

[62] J. S. Hickman, "Trends in Modern Sonar Transducer Design," *Proceedings of the Twenty-Second National Electronics Conference,* October 1966, Chicago, Ill.

[63] E. L. Carson, G. E. Martin, G. W. Benthien, and J. S. Hickman, "Control of Element Velocity Distributions in Sonar Projection Arrays," *Proceedings of the Seventh Navy Science Symposium,* May 1963, Pensacola, FL.

[64] A. Parthy, C. Jin, and A. van Schaik, "Acoustic Holography with a Concentric Rigid and Open Spherical Microphone Array," *IEEE International Conference on Acoustics,* Speech and Signal Processing, 2009, pp. 2173–2176.

[65] J. F. Bartram, R. R. Ramseyer, and J. M Heines, "Fifth Generation Digital Sonar Signal Processing," *IEEE EASCON 1976 Record, IEEE Electronics and Aerospace Systems Convention,* September 26–29, Washington, D.C., pp. 91A–91G.

[66] T. F. Hueter, J. C. Preble, and G. D. Marshall, "Distributed Architecture for Sonar Systems and Their Impact on Operability," *IEEE EASCON 1976 Record, IEEE Electronics and Aerospace Convention,* September 16–29, Washington, D.C., pp. 96A–96F.

[67] V. C. Anderson, "The First Twenty Years of Acoustic Signal Processing," *J. Acoust. Soc. Am.,* Vol. 51, No. 3 (Part 2), March 1972, pp. 1062–1063.

[68] M. Federici, "On the Improvement of Detection and Precision Capabilities of Sonar Systems," *Proceedings of the Symposium on Sonar Systems*, July 9–12, 1962, University of Birmingham, pp. 535–540, Reprinted in J. Br. I.R.E., Vol. 25, No.6, June 1963.

[69] G. G. Rassweiler, M. R. Williams, L. M. Payne, and G. P. Martin, "A Miniaturized Lightweight Wideband Null Steerer," *IEEE Trans. Ant. & Prop.,* Vol. AP-24, September 1976, pp. 749–754.

[70] W. R. Jones, W. K. Masenten, and T. W. Miller, "UHF Adaptive Array Processor Development for Naval Communications," *Digest AP-S Int. Symp., IEEE Ant. & Prop Soc.,* June 2–22, 1977, Stanford, CA., IEEE Cat. No. 77CH1217-9 AP, pp. 394–397.

[71] G. Wade, "Acoustic Imaging with Holography and Lenses," *IEEE Trans. Sonics & Ultrason.,* Vol. SU-22, No. 6, November 1975, pp 385–394.

CHAPTER 2

Adaptive Array Concept

Chapter Outline

To understand why an array of sensor elements has the potential to improve the reception of a desired signal in an environment having several sources of interference, it is necessary to understand the nature of the signals as well as the properties of an array of sensor elements. Furthermore, the types of elements and their arrangement impact the adaptive array performance. To gain this understanding the desired signal characteristics, interference characteristics, and signal propagation effects are first discussed. The properties of sensor arrays are then introduced, and the possibility of adjusting the array response to enhance the desired signal reception is demonstrated. Trade-offs for linear and planar arrays are presented to aid the designer in finding an economical array configuration.

In arriving at an adaptive array design, it is necessary to consider the system constraints imposed by the nature of the array, the associated system elements with which the designer has to work, and the system requirements the design is expected to satisfy. Adaptive array requirements may be classified as either (1) steady-state or (2) transient depending on whether it is assumed the array weights have reached their steady-state values (assuming a stationary signal environment) or are being adjusted in response to a change in the signal environment. If the system requirements are to be realistic, they must not exceed the predicted theoretical performance limits for the adaptive array system being considered. Formulation of the constraints imposed by the sensor array is addressed in this chapter. Steady-state performance limits are considered in Chapter 3. The formulation of transient performance limits, which is considerably more involved, is addressed in Part 2. For the performance limits of adaptive array systems to be analyzed, it is necessary to develop a generic analytic model for the system. The development of such an analytic model will be concerned with the signal characteristics and the subsequent processing necessary to obtain the desired system response.

2.1 SIGNAL ENVIRONMENT

Our goal is to extract useful information from a desired signal received by an array of sensors. The adaptive array designer must exploit significant differences between the desired signal and any interference signals to distinguish the desired signal from all other signals received by the array. The signal parameters that may be exploited include direction of arrival, amplitude, phase, spectral characteristics (e.g., frequency and power), modulation characteristics, and polarization. It is therefore worthwhile to consider the desired signal characteristics and the nature of spurious interfering signals in different contexts.

2.1.1 Signals in Active and Passive Sensors

Active sensing devices, such as radar and sonar systems, generate a known pulse (or pulse train) that propagates through a transmission medium and reflects from a target back to the original sender. During most of the listening time interval the desired signal is absent in contrast with communication systems where the desired signal is usually present. Often, the signal structure and the direction of arrival are known, so the desired signal is easily recognized when it is present. In addition to the desired signal echo, noise due to clutter and multipath may also be present [1]. For radar systems, diffuse scattering of multipath gives rise to spurious signals, and a jammer may deliberately generate an interference signal. For active sonar systems, the two main types of interference signals come from ambient noise and reverberation return [2].

Reverberation is analogous to clutter in radar systems and is defined as signal-generated noise that results when a sonar signal transmitted into the ocean encounters many kinds of diffuse scattering objects that reradiate part of the signal back to the receiver. Reverberation returns are classified as surface, bottom, or volume reverberation depending on whether the unwanted reflections originate from the ocean surface, the ocean surface, the ocean bottom, or some point in between. Furthermore, multipath signals arise from signals that reflect from nondiffuse reflectors located at different reflecting angles and impinge on the receiving array. Other propagation effects that cause sonar signal distortion are geometric spreading, attenuation, multiple propagation paths, Doppler frequency shift, finite amplitude, medium coherency, and time dispersion [3].

In the case of passive sensing systems, the target or event of interest generates the desired signal and in many cases is present for most of the listening interval. A passive system has difficulty distinguishing the desired target signal from the background noise [4]. In contrast to active sensing devices, however, the direction of arrival and structure of the desired signal may not be known beforehand. The bandwidth of the desired signal is the most common means for distinguishing it from an interference signal. In some cases, the power level of the desired signal may be known and used as a distinguishing characteristic. Spread spectrum communication systems commonly employ a known pseudo noise (PN) code to modulate the transmitted waveform, and this code then provides a convenient means for distinguishing the desired signal.

2.1.2 Signal Models

Signals in passive sonar systems arise from a variety of sources such as engine noise or propeller noise [5] and are treated as random processes. Likewise, an unknown communications signal may often be regarded as random, although (as we will see) a simple

signal model possessing some of the signal characteristics of the unknown signal can be adopted in certain circumstances. Communications signals pass through environments that randomly add scattering and noise to the desired signal. Thermal sensor noise, ambient noise, and interference signal sources are also random in nature. These noises typically arise from the combined effect of many small independent sources, and application of the central limit theorem of statistics [6] permits the designer to model the resulting noise signal as a Gaussian (and usually stationary) random process. Quite frequently, the physical phenomena responsible for the randomness in the signals of concern are such that it is plausible to assume a Gaussian random process. The statistical properties of Gaussian signals are particularly convenient because the first- and second-moment characteristics of the process provide a complete characterization of the random signal.

The statistical properties of the signal are not always known, so a selected deterministic signal is used instead. This deterministic signal does not have to be a perfect replica of the desired signal. It needs only to be reasonably well correlated with the desired signal and uncorrelated with the interference signals.

Sometimes the desired signal is known, as in the case of a coherent radar with a target of known range and character. In other cases, the desired signal is known except for the presence of uncertain parameters such as phase and signal energy. The case of a signal known except for phase occurs with an ordinary pulse radar having no integration and with a target of known range and character. Likewise, the case of a signal known except for phase and signal energy occurs with a pulse radar operating without integration and with a target of unknown range and known character. The frequency and bandwidth of communication signals are typically known, and such signals may be given a signature by introducing a pilot signal.

For a receive array composed of N sensors, the received waveforms correspond to N outputs, $x_1(t), x_2(t), \ldots, x_N(t)$, which are placed in the received signal vector $\mathbf{x}(t)$ where

$$\mathbf{x}(t) \triangleq \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} \quad \text{for} \quad 0 \leq t \leq T \tag{2.1}$$

over the observation time interval. The received signal vector is the sum of the desired signal vector, $\mathbf{s}(t)$, and the noise component, $\mathbf{n}(t)$.

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t) \quad \text{for} \quad 0 \leq t \leq T \tag{2.2}$$

where the desired signal vector is represented by

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \quad \text{for} \quad 0 \leq t \leq T \tag{2.3}$$

The various signal components either may be known exactly, known only to a rough approximation, or known only in a statistical sense.

Interference noise fields at best are stationary and unknown, but they usually exhibit variations with time. Adaptive array processors that automatically respond to the interference environment must cope with such time variations. Most adaptive processing

techniques assume slowly varying Gaussian ambient noise fields. Adaptive processors designed for Gaussian noise are distinguished by the pleasant fact that they depend only on second-order noise moments. Consequently, when non-Gaussian noise fields must be dealt with, the most convenient approach is to design a Gaussian-equivalent suboptimum adaptive system based on the second-order moments of the non-Gaussian noise field. In general, an adaptive system works best when the variations that occur in the noise environment are slow.

2.1.3 Ideal Propagation Model

The signal vector, $\mathbf{s}(t)$, relates to a scalar signal, $s(t)$, generated at a point source in space by [7]

$$\mathbf{s}(t) = \int \mathbf{m}(t-\tau)s(\tau)\,d\tau \tag{2.4}$$

where the ith component of $\mathbf{m}(t)$ is $m_i(t)$ and represents the propagation effects from the source to the ith sensor as well as the response of the ith sensor. For the ideal case of nondispersive propagation and distortion-free sensors, then $m_i(t)$ is a simple time delay $\delta(t-\tau_i)$, and the desired signal component at each sensor element is identical except for a time delay so that (2.4) can be written as

$$\mathbf{s}(t) = \begin{bmatrix} s(t-\tau_1) \\ s(t-\tau_2) \\ \vdots \\ s(t-\tau_N) \end{bmatrix} \tag{2.5}$$

When the array is far from the source, then the signal is represented by a plane wave from the direction $\boldsymbol{\alpha}$ as shown in Figure 2-1 (where $\boldsymbol{\alpha}$ is taken to be a unit vector). In this case,

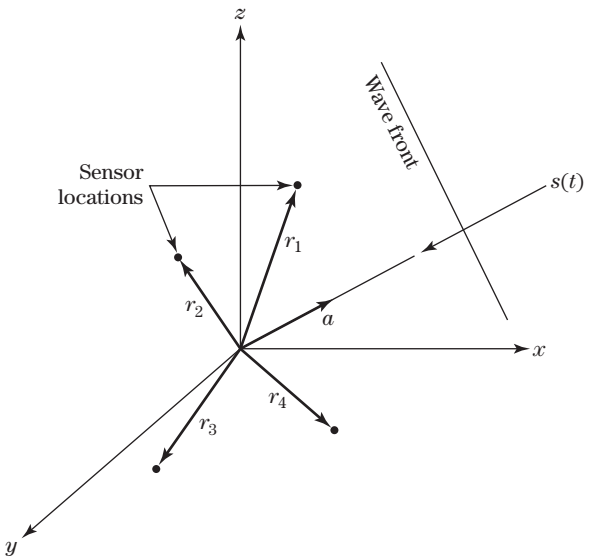


FIGURE 2-1 ■ Three-dimensional array with plane wave signal propagation.

the various time delays are simply given by [8]

$$\tau_i = \frac{\boldsymbol{\alpha} \cdot \mathbf{r}_i}{\mathfrak{v}} \tag{2.6}$$

where $\mathfrak{v}$ is the propagation velocity, each sensor coordinate is given by the vector $\mathbf{r}_i$, and $\boldsymbol{\alpha}\, \mathbf{r}_i$ denotes the dot product

$$\boldsymbol{\alpha} \cdot \mathbf{r}_i \triangleq \alpha^T \mathbf{r}_i \tag{2.7}$$

and T denotes transpose.

Measuring the relative time delays experienced at each sensor element therefore provides a means of determining the unknown direction of arrival of the desired signal $s(t)$. In the following sections on array properties, the previously described plane wave propagation effects play a fundamental role in determining the nature of the array response. Furthermore, it should be noted that the relative time-delay properties suggested by (2.5) and (2.6) work for random and nonrandom signals.

2.2 | ARRAY ELEMENT SPACING CONSIDERATIONS

It was noted earlier that an array of sensors increases the gain and decreases the bandwidth of a single sensor as well as changes the shape of the array pattern. To demonstrate, a single pair of identical sensor elements is first considered. The discussion then proceeds to consider the one-dimensional linear array of equally spaced elements and finally extends to planar and conformal arrays.

After considering the effects of sensor element arrangement within the array, the impact on array performance limits that are determined by a selected array configuration are examined. The position of the elements within the array determines array resolution and interferometer (grating lobe) [9] effects. In general, resolution increases as the array dimension (or separation between elements) increases. High array resolution improves the maximum output signal-to-noise ratio (SNR) when the angular separation between the desired and the undesired signals is small. High-resolution capability also implies sharp array pattern nulls, however, thereby reducing the array ability to place a broad null on clustered interference sources. An N-element linear array can have up to $N - 1$ degrees of freedom so that up to $N - 1$ array beam pattern nulls can be independently adjusted for array operation.

Typically, there is a limited amount of space over which to distribute antenna elements. Unfortunately, closely spaced elements yield fields that interact, which results in mutual coupling between elements, and it is well known [10] that mutual coupling has deleterious effects on array signal-to-interference plus noise ratio (SINR) performance and transient response. A discussion of how to account for mutual coupling effects is taken up in Chapter 10.

2.2.1 Pair of Identical Sensors

Consider the pair ($N = 2$) of identical isotropic point sources shown in Figure 2-2 spaced apart by a distance d. An isotropic point source radiates equally in all directions. Let a

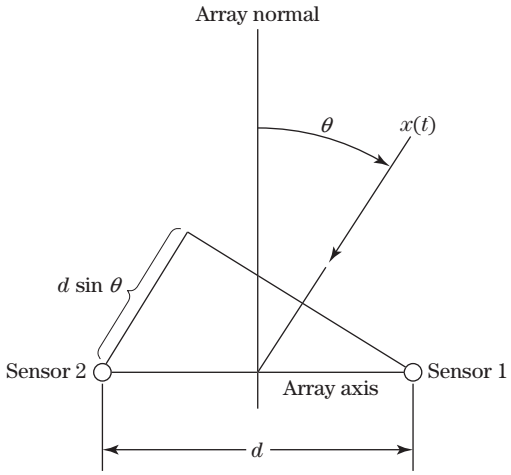


FIGURE 2-2 ■ Pair of isotropic point sources.

signal $x(t)$ impinge on the two sensor elements in a plane containing the two elements and the signal source from a direction θ with respect to the array normal. Figure 2-2 shows that element 2 receives the signal τ after element 1.

$$\tau = \frac{d \sin\theta}{\mathfrak{v}} \tag{2.8}$$

Let the array output signal $y(t)$ be given by the sum of the two sensor element signals so that

$$y(t) = x(t) + x(t - \tau) \tag{2.9}$$

If $x(t)$ is a narrowband signal having center frequency f_0, then the time delay τ corresponds to a phase shift of $2\pi(d/\lambda_0)\sin\theta$ radians, where λ_0 is the wavelength corresponding to the center frequency,

$$\lambda_0 = \frac{\mathfrak{v}}{f_0} \tag{2.10}$$

The overall array response is the sum of the signal contributions from the two array elements. That is,

$$y(t) = \sum_{i=1}^{2} x(t) e^{j(i-1)\psi} \tag{2.11}$$

where

$$\psi = 2\pi(d/\lambda_0)\sin\theta \tag{2.12}$$

The directional pattern or array factor of the array (sensitivity to signals vs. angle at a specified frequency) may be found by considering only the term

$$AF(\theta) = \sum_{i=1}^{2} e^{j(i-1)\psi} \tag{2.13}$$

The normalized array factor in decibels for the two elements is then given by

$$AF(\theta)(\text{decibels}) = 10\log_{10}\left(\frac{|AF(\theta)^2|}{2^2}\right) \tag{2.14}$$

so that the peak value is unity or 0 dB. A plot of the normalized $AF(\theta)$ for this two-element example is given in Figure 2-3 for $d/\lambda_0 = 0.5$, 1.0, and 1.5. From Figure 2-3a it is seen that for $d/\lambda_0 = 0.5$ there is one principal lobe (or main beam) having a 3 dB beamwidth of 60° and nulls at $\theta = \pm 90°$ off broadside. The nulls at $\theta = \pm 90°$ occur because at that direction of arrival the signal wavefront must travel exactly $\lambda_0/2$ between

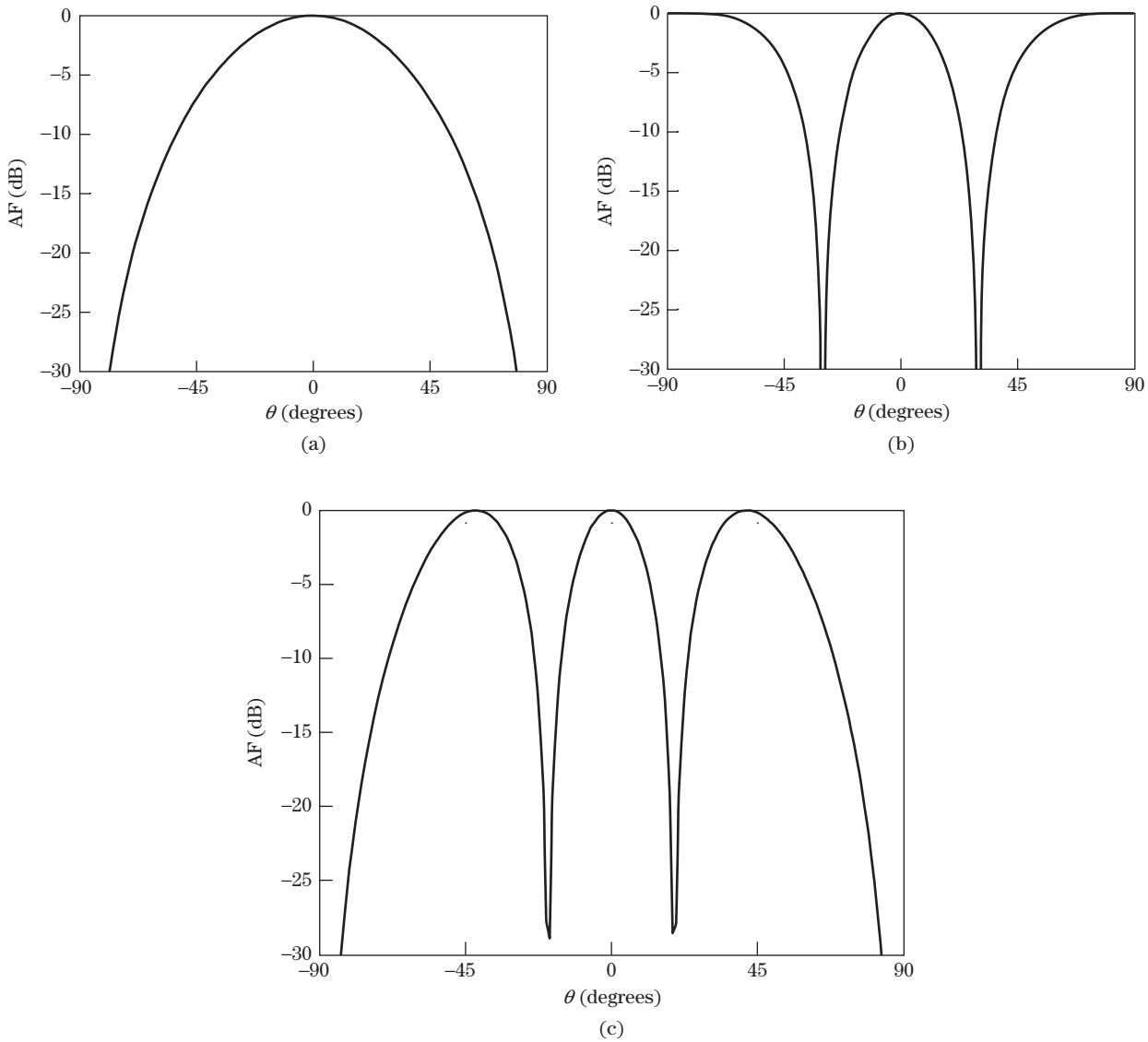


FIGURE 2-3 ■ Array beam patterns for two-element example. (a) $d/\lambda_0 = 0.5$. (b) $d/\lambda_0 = 1$. (c) $d/\lambda_0 = 1.5$.

the two sensors, which corresponds to a phase shift of 180° between the signals appearing at the two sensors and therefore yields exact cancellation of the resulting phasor sum. If the element spacing is less than $0.5\lambda_0$, then exact cancellation at $\theta = \pm 90°$ does not result, and in the limit as the element spacing approaches zero (ignoring mutual coupling effects) the directional pattern becomes an isotropic point source pattern. There is very little difference in the directional pattern between a single element and two closely spaced elements (less than $\lambda/4$ apart); consequently, arrays employing many elements very closely spaced are considered "inefficient" if it is desired to use as few array elements as possible for a specified sidelobe level and beamwidth. If the element spacing increases to greater than $0.5\lambda_0$, the two pattern nulls migrate in from $\theta = \pm 90°$, occurring at $\theta = \pm 30°$ when $d = \lambda_0$, as illustrated in Figure 2-3b. The nulls at $\theta = \pm 30°$ in Figure 2-3b occur because at that angle of arrival the phase path difference between the two sensors is once again 180°, and exact cancellation results from the phasor sum. Two sidelobes at $\theta = \pm 90°$ having an amplitude equal to the principal lobe at $\theta = 0°$. They appear because the phase path difference between the two sensors is then 360°, two phasors exactly align, and the array response is the same as for broadside angle of arrival. As the element spacing increases to $1.5\lambda_0$, the main lobe beamwidth decreases still further, thereby improving resolution, the two pattern nulls migrate further in, and two new nulls appear at ±90°, as illustrated in Figure 2-3c. Further increasing the interelement spacing results in the appearance of even more pattern nulls and sidelobes and a further decrease in the main lobe beamwidth.

When $N > 2$, the array factor for N isotropic point sources becomes

$$AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}[x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi + z_n\cos\theta]} \tag{2.15}$$

where

$(x_n, y_n, z_n) =$ position of sensor n in the array
$N =$ number of sensors in the array
$(\theta, \phi) =$ direction from array phase center

Three common examples of multiple element arrays used for adaptive nulling are [11]

$$\text{Linear array along the x-axis: } AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}x_n \sin\theta\cos\phi} \tag{2.16}$$

$$\text{Planar array in the x-y plane: } AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}[x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi]} \tag{2.17}$$

$$\text{3-dimensional array: } AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}[x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi + z_n\cos\theta]} \tag{2.18}$$

The sensors take samples of the signals incident on the array. As long as the sensors take two samples in one period or wavelength, then the received signal is adequately reproduced.

2.2.2 Linear Arrays

For a linear array of N equispaced sensor elements, the overall array response may again be found by considering the sum of signal contributions from each array element as in (2.11):

$$y(t) = \sum_{i=1}^{N} w_i x(t) e^{j(i-1)\psi} \tag{2.19}$$

The directional pattern in a plane containing the array may therefore be found by considering the array factor

$$AF(\theta) = \sum_{i=1}^{N} w_i e^{j(i-1)\psi} \tag{2.20}$$

When $w_i = 1.0$, the array is called a "uniform array." The uniform array has an array factor given by

$$AF(\theta) = \frac{\sin\left(\frac{N\psi}{2}\right)}{\sin\left(\frac{\psi}{2}\right)} \tag{2.21}$$

Nulls occur when the numerator is zero or

$$\theta_m = \sin^{-1}\left(\pm\frac{m\lambda}{dN}\right), \quad m = 1, 2, \ldots \tag{2.22}$$

Sidelobe peaks are approximately halfway between two adjacent nulls

$$\theta_m = \sin^{-1}\left(\pm\frac{(2m+1)\,\lambda}{2dN}\right), \quad m = 1, 2, \ldots \tag{2.23}$$

The first sidelobe is approximately 13.5 dB below the peak of the main beam.

For nonisotropic sensor elements, it is necessary to introduce an additional factor $F_i(f_0, \theta)$ in (2.20) to include the pattern introduced by each sensor element.

$$AF(\theta) = \sum_{i=1}^{N} F_i(f_0, \theta) w_i e^{j(i-1)\psi} \tag{2.24}$$

If mutual coupling is ignored or if the element patterns are averaged, the individual element patterns are all the same, so the array pattern is the product of an element pattern times the array factor.

$$AF(\theta) = F(f_0, \theta) \sum_{i=1}^{N} w_i e^{j(i-1)\psi} \tag{2.25}$$

Alternatively, the response of an arbitrary configuration of identical elements having the same orientation can be derived by using the principle of pattern multiplication [9]. The directional pattern of an array of identical spatial elements may be found by

1. Replacing each of the elements by a point source
2. Determining the directional array pattern of the resulting array of point sources

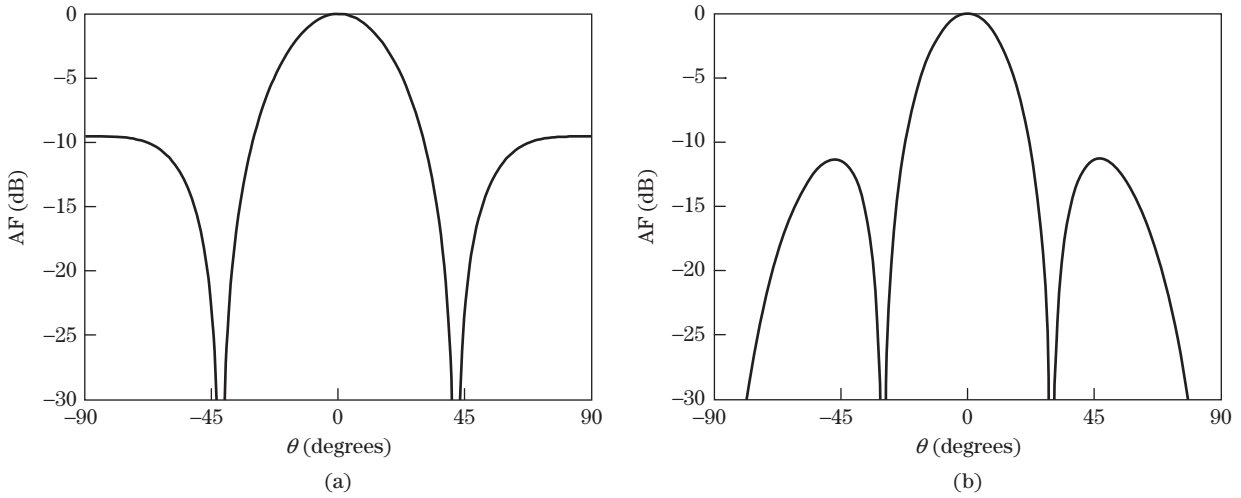


FIGURE 2-4 ■ inear array beam patterns for $d/\lambda_0 = 0.5$. (a) Three-element array. (b) Four-element array.

3. Multiplying the array pattern resulting from step 2 by the beam pattern of the individual elements of the original array

Maintaining the interelement spacing at $d/\lambda_0 = 0.5$ and increasing the number of point sources, the normalized array directional (or beam) pattern may be found from (2.21), and the results are shown in Figure 2-4 for three and four elements. It is seen that as the number of elements increases the main lobe beamwidth decreases, and the number of sidelobes and pattern nulls increases.

To illustrate how element spacing affects the directional pattern for a seven-element linear array, Figures 2-5a through 2.5d show the directional pattern in the azimuth plane for values of d/λ_0 ranging from 0.1 to 1.0. So long as d/λ_0 is less than $\frac{1}{7}$, the beam pattern has no exact nulls as the -8.5 dB null occurring at $\theta = \pm 90^\circ$ for $d/\lambda_0 = 0.1$ illustrates. As the interelement spacing increases beyond $d/\lambda_0 = \frac{1}{7}$, pattern nulls and sidelobes (and grating lobes) begin to appear, with more lobes and nulls appearing as d/λ_0 increases and producing an interferometer pattern. When $d/\lambda_0 = 1$ the endfire sidelobes at $\theta = \pm 90^\circ$ have a gain equal to the main lobe since the seven signal phasors now align exactly and add coherently.

Suppose for the linear array of Figure 2-6 that a phase shift (or an equivalent time delay) of δ is inserted in the second element of the array, a phase shift of 2δ in the third element, and a phase shift of $(n-1)\delta$ in each succeeding nth element. The insertion of this sequence of phase shifts has the effect of shifting the principal lobe (or main beam) by

$$\theta_s = -sin^{-1}\left[\frac{1}{2\pi}\left(\frac{\lambda_0}{d}\right)\delta\right] \tag{2.26}$$

so the overall directional pattern has in effect been "steered" by insertion of the phase-shift sequence. Figure 2-7 is the array factor in Figure 2-5c steered to $\theta_s = -30^\circ$ using $\delta = -90^\circ$.

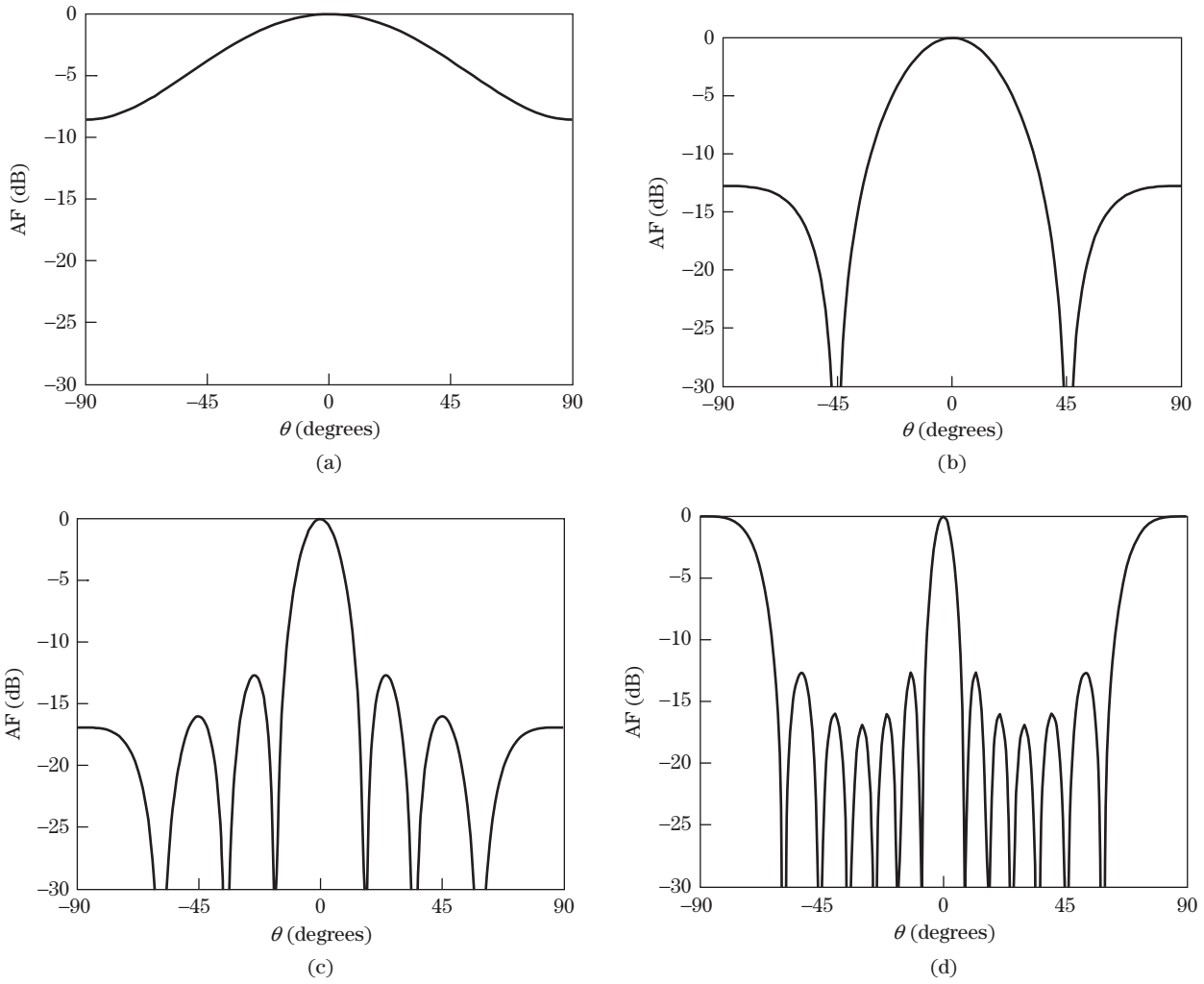


FIGURE 2-5 ■ Seven-element linear array factors. (a) $d/\lambda_0 = 0.1$. (b) $d/\lambda_0 = 0.2$. (c) $d/\lambda_0 = 0.5$. (d) $d/\lambda_0 = 1.0$.

The gain of an array factor determines how much the signal entering the main beam is magnified by the array. For a radar/sonar system the received power is given by [12]

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{2.27}$$

where

P_t = power transmitted
G = array gain
σ = target cross section
R = distance from array to target

and for a communications system is given by Friis transmission formula [13]

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2} \tag{2.28}$$

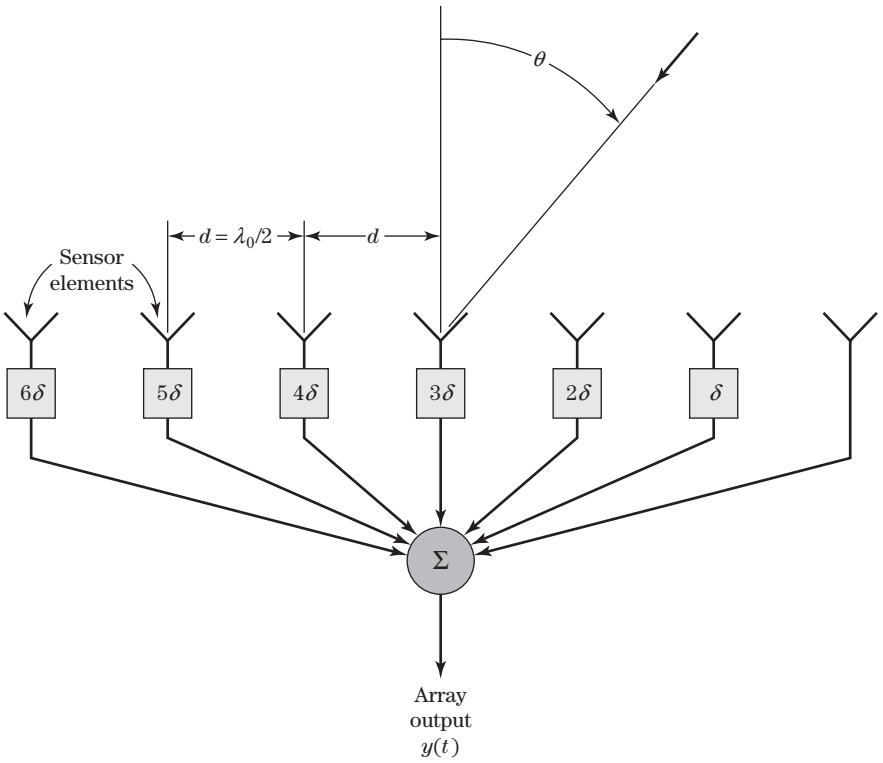


FIGURE 2-6 ■ Seven-element linear array steered with phase-shift elements.

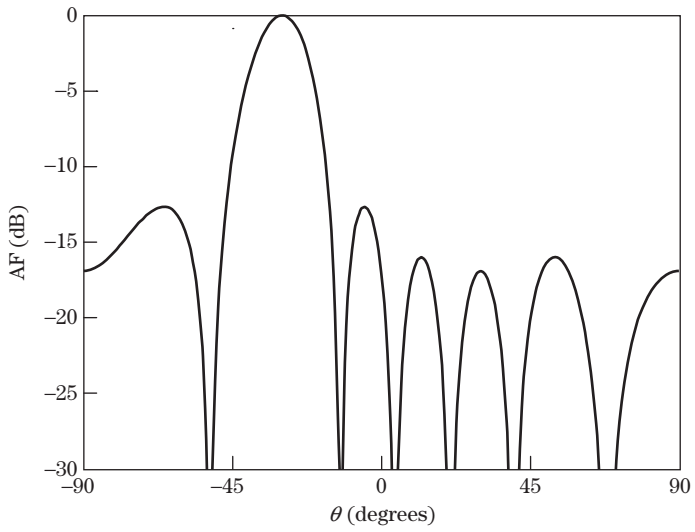


FIGURE 2-7 ■ Seven-element linear array factor steered to $\theta_s = 30^\circ$ with $\delta = -90^\circ$.

The array gain is the ratio of the power radiated in one direction to the power delivered to the array. Directivity is similar to gain but does not include losses in the array. As a result, directivity is greater than or equal to gain. Realized gain includes the mismatch between the array and the transmission line feeding the array. If gain is not written as a function of angle, then G is the maximum gain of the array. Gain is usually expressed

in dB as

$$G_{dB} = 10\log_{10} G = 10\log G \tag{2.29}$$

The directivity of an array is found by solving

$$D = \frac{4\pi |AF_{\max}|^2}{\int_0^{2\pi}\int_0^{\pi} |AF(\theta,\phi)|^2 \sin\theta d\theta d\phi} \tag{2.30}$$

If the elements are spaced 0.5λ apart in a linear array, then the directivity formula simplifies to

$$D = \frac{\left|\sum_{n=1}^{N} w_n\right|^2}{\sum_{n=1}^{N} |w_n|^2} \tag{2.31}$$

The z-transform converts the linear array factor into a polynomial using the substitution

$$z = e^{j\psi} \tag{2.32}$$

Substituting z into (2.16) yields a polynomial in z

$$AF = \sum_{n=1}^{N} w_n z^{(n-1)} = w_1 + w_2 z + \cdots + w_N z^{N-1} \tag{2.33}$$

that can be factored into the form

$$AF = w_N (z - z_1)(z - z_2)\cdots(z - z_{N-1}) \tag{2.34}$$

Roots of the array factor polynomial ($z = z_n$) correspond to nulls in the array factor. Roots have a magnitude of 1 and phase of ψ_n. The roots of (2.34) have a magnitude of 1 and a phase of ψ_n and hence can be graphed on a circle in the z-plane as shown in Figure 2-8

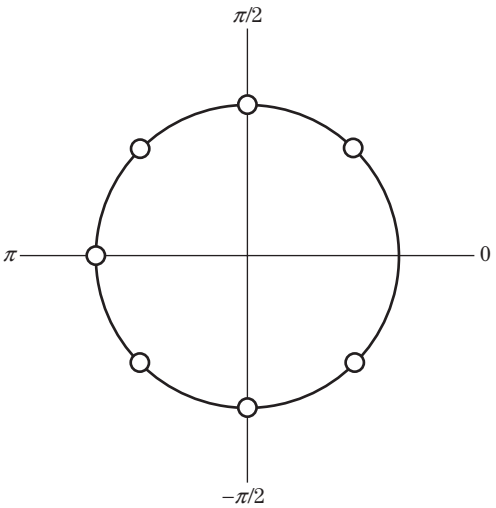


FIGURE 2-8 ■ Unit circle representation of an eight-element uniform array.

for the case of an eight-element uniform array with $d = 0.5\lambda$:

$$z^7 + z^6 + z^5 + z^4 + z^3 + z^2 + z + 1$$
$$= \left(z + e^{j\pi/4}\right)\left(z + e^{-j\pi/4}\right)\left(z + e^{j\pi/2}\right)\left(z + e^{-j\pi/2}\right)\left(z + e^{j3\pi/4}\right)\left(z + e^{j3\pi/4}\right)\left(z + e^{j\pi}\right) \tag{2.35}$$

2.2.3 Planar Arrays

Consider the rectangular-shaped planar array of discrete sensor elements arranged in the x–y plane as shown in Figure 2-9, where the coordinate origin is chosen at the central element. With N_x elements in each x-axis parallel column and uniform spacing d_x, and N_y elements in each y axis parallel row with uniform spacing d_y, the entire array has $N_x \times N_y$ elements.

The sum of signal contributions from each array element in a single column is the same as that for a linear array and is therefore given by

$$y(t) = \sum_{i=1}^{N_x} w_i x(t) e^{j(i-1)\psi_x} \tag{2.36}$$

whereas for a single-row array the output is

$$y(t) = \sum_{k=1}^{N_y} w_k x(t) e^{j(k-1)\psi_y} \tag{2.37}$$

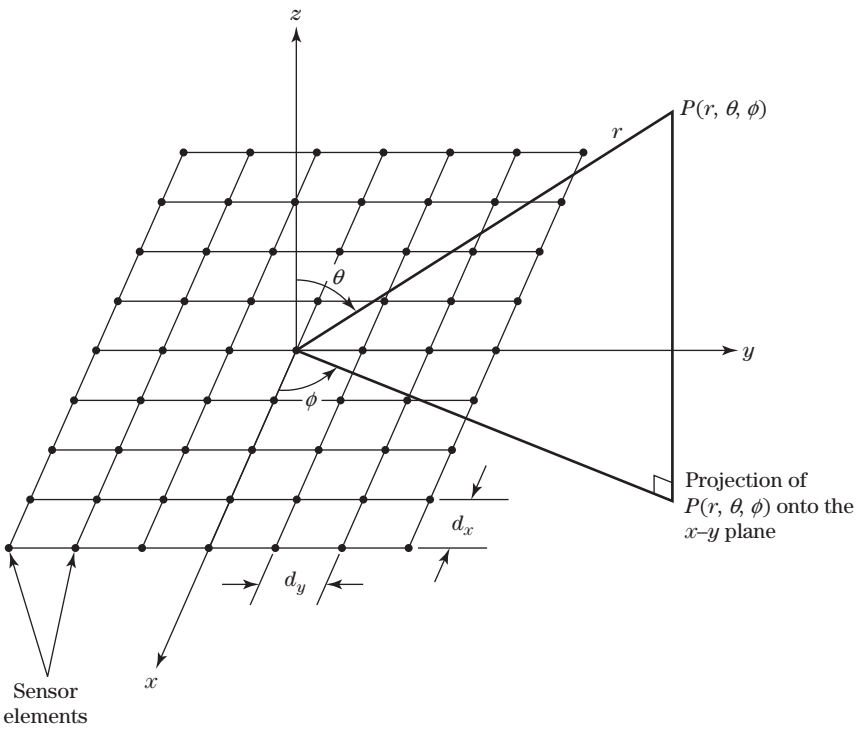


FIGURE 2-9 ■ Rectangular-shaped planar array.

where

$$\psi_x = 2\pi \left(\frac{d_x}{\lambda_0}\right) \sin\theta \cos\phi \quad \text{and} \quad \psi_y = 2\pi \left(\frac{d_y}{\lambda_0}\right) \sin\theta \sin\phi \tag{2.38}$$

The output signal now depends on both the projected azimuth angle, ϕ, and the elevation angle, θ. The total sum of signal contributions from all array elements is given by

$$y(t) = \sum_{i=1}^{N_x} \sum_{k=1}^{N_y} w_{i,k} x(t) e^{j(i-1)\psi_x} e^{j(k-1)\psi_y} \tag{2.39}$$

If the weights are separable, $w_{i,k} = w_i \times w_k$, then the planar array factor is

$$AF(\theta, \phi) = AF_x(\theta, \phi) AF_y(\theta, \phi) \tag{2.40}$$

where

and

$$\left.\begin{aligned} AF_x(\theta, \phi) &= \sum_{i=1}^{N_x} w_i e^{j(i-1)\psi_x} \\ AF_y(\theta, \phi) &= \sum_{k=1}^{N_y} w_k e^{j(k-1)\psi_y} \end{aligned}\right\} \tag{2.41}$$

Elements in a planar array are usually arranged in either a rectangular lattice or an equilateral triangle lattice. The y-spacing in an equilateral triangle lattice is found from the x-spacing by

$$d_y = d_x \sqrt{3}/2 \tag{2.42}$$

If $w_n = 1.0$ for all n, then the array factor for rectangular spacing is

$$AF = \frac{\sin\left(\frac{N_x \psi_x}{2}\right)}{N_x \sin\left(\frac{\psi_x}{2}\right)} \frac{\sin\left(\frac{N_y \psi_y}{2}\right)}{N_y \sin\left(\frac{\psi_y}{2}\right)} \tag{2.43}$$

where

$$\psi_x = \frac{2\pi}{\lambda} d_x \sin\theta \cos\phi$$

$$\psi_y = \frac{2\pi}{\lambda} d_y \sin\theta \sin\phi$$

Usually, the beamwidth of a planar array is defined for two orthogonal planes. For example, the beamwidth is usually defined in θ for $\phi = 0^\circ$ and $\phi = 90^\circ$. As with a linear array, nulls in the array factor of a planar array are found by setting the array factor equal to zero. Unlike linear arrays, the nulls are not single points.

The directivity of a planar array can be found by numerically integrating (2.24). A reasonable estimate for the 3 dB beamwidth in orthogonal planes is given by

$$D = \frac{32400}{\theta^\circ_{3dB\phi=0^\circ} \theta^\circ_{3dB\phi=90^\circ}} \tag{2.44}$$

where

$\theta^{\circ}_{3dB\phi=0^{\circ}}$ = 3 dB beamwidth in degrees at $\phi = 0^{\circ}$

$\theta^{\circ}_{3dB\phi=90^{\circ}}$ = 3 dB beamwidth in degrees at $\phi = 90^{\circ}$

The array factor for a planar array with rectangular element spacing can be written as

$$AF = \sum_{m=1}^{N_y} \sum_{n=1}^{N_x} w_{mn} e^{j2\pi \left\{ (n-1)\frac{d_x}{\lambda}(u-u_s) + (m-1)\frac{d_y}{\lambda}(v-v_s) \right\}} \text{(w)} \tag{2.45}$$

where

$$\begin{aligned} u &= \sin\theta \cos\phi \\ v &= \sin\theta \sin\phi \\ u_s &= \sin\theta_s \cos\phi_s \\ v_s &= \sin\theta_s \sin\phi_s \end{aligned} \tag{2.46}$$

when the main beam is steered to (θ_s, ϕ_s).

2.2.4 Conformal Arrays

Array elements do not have to lie along a line or in a plane. Many times, it is more convenient if the elements lie on or conform to a surface. The surface may be the side of a building or the side of an airplane. In any event, the array factor now has element locations given by (x_n, y_n, z_n), which are points lying on a defined surface.

$$AF = \sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}(x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi + z_n\cos\theta)} \tag{2.47}$$

The array curvature causes phase errors that distort the array factor unless phase weights at the elements compensate. As an example, consider a 12-element linear array bent around a cylinder of radius $r = 3.6\lambda$ as shown in Figure 2-10. If no phase compensation is applied, then the array factor looks like the dashed line in Figure 2-11. Adding a phase delay of $y_n 2\pi/\lambda$ results in the array factor represented by the solid line in Figure 2-11. The compensation restores the main beam and lowers the sidelobe level.

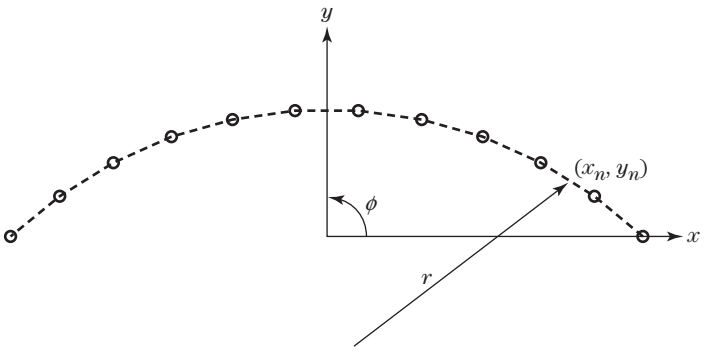


FIGURE 2-10 ■ Twelve-element linear array bent around a cylinder of radius $r = 3.6\lambda$.

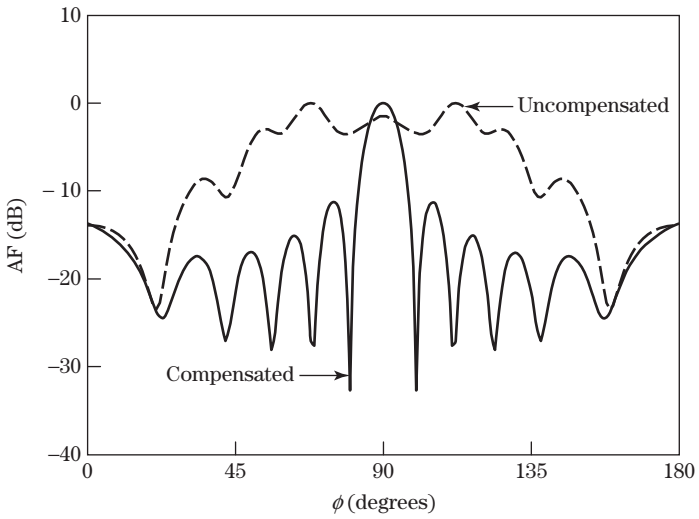


FIGURE 2-11 ■ Array factor for the 12-element conformal array (with and without phase compensation).

2.3 ARRAY PERFORMANCE

The ability to place nulls in the array pattern is highly dependent on the array architecture. Array designers start with the following requirements specified:

1. Resolution (gain)
2. Scan coverage
3. Sidelobe level
4. Cost (primarily driven by the number of elements)

These design specifications will limit the possible approaches to adaptive nulling. Control of the main beam gain, sidelobe level, and null placement is possible via the

1. Number of elements in the array
2. Element phase
3. Element amplitude
4. Element positions
5. Size of the array
6. Shape of the array

Trade-offs exist among array size, resolution, and cost. Increased control of the array pattern comes at a high cost. Increasing the number of elements in an array increases not only the gain but also the cost. The same is true for adaptive weights. More adaptive weights allows more control over null placement, but the cost and complexity for that control are high.

2.3.1 Enhanced Signal Reception by Adjustment of the Array Response

To demonstrate steering and modifying the array pattern to enhance desired signal reception and simultaneously suppress interference signals by complex weight selection,

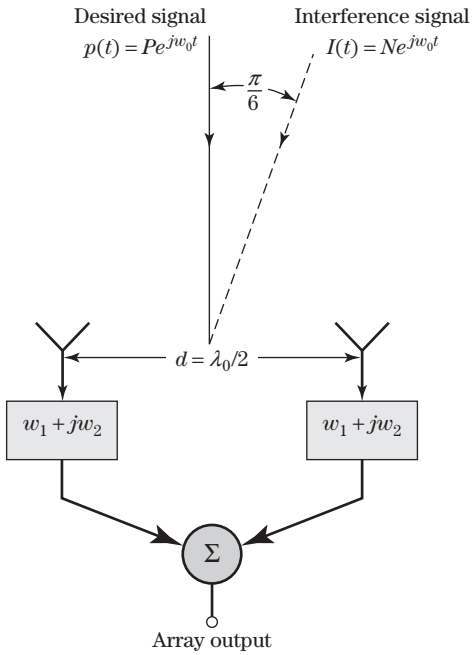


FIGURE 2-12 ■ Two-element array for interference suppression example.

consider the two element array of isotropic point sources in Figure 2-12 in which a desired signal arrives from the normal direction $\theta = 0^\circ$, and the interference signal arrives from the angle $\theta = 30^\circ$. For simplicity, both the interference signal and the desired pilot signal are assumed to be at the same frequency f_0. Furthermore, assume that at the point exactly midway between the array elements the desired signal and the interference are in phase (this assumption is not required but simplifies the development). The output signal from each element is input to a variable complex weight, and the complex weight outputs are then summed to form the array output.

Now consider how the complex weights can be adjusted to enhance the reception of $p(t)$ while rejecting $I(t)$. The array output due to the desired signal is

$$Pe^{j\omega_0 t}\{[w_1 + w_3] + j[w_2 + w_4]\} \tag{2.48}$$

For the output signal of (2.48) to be equal to $p(t) = Pe^{j\omega_0 t}$, it is necessary that

$$\left.\begin{aligned} w_1 + w_3 &= 1 \\ w_2 + w_4 &= 0 \end{aligned}\right\} \tag{2.49}$$

The incident interfering noise signal exhibits a phase lead with respect to the array midpoint when impinging on the element with complex weight $w_3 + jw_4$ of value $2\pi(\frac{1}{4})\sin(\pi/6) = \pi/4$ and a phase lag when striking the other element of value $-\pi/4$. Consequently, the array output due to the incident noise is given by

$$Ne^{j(\omega_0 t - \pi/4)}[w_1 + jw_2] + Ne^{j(\omega_0 t + \pi/4)}[w_3 + jw_4] \tag{2.50}$$

Now

$$e^{j(\omega_0 t - \pi/4)} = \frac{1}{\sqrt{2}}[e^{j\omega_0 t}(1 - j)]$$

and

$$e^{j(\omega_0 t+\pi/4)} = \frac{1}{\sqrt{2}}[e^{j\omega_0 t}(1+j)]$$

so for the array noise response to be zero it is necessary that

$$\left.\begin{aligned} w_1 + w_2 + w_3 - w_4 &= 0 \\ -w_1 + w_2 + w_3 + w_4 &= 0 \end{aligned}\right\} \tag{2.51}$$

Solving (2.49) and (2.51) simultaneously then yields

$$w_1 = \frac{1}{2}, w_2 = -\frac{1}{2}, w_3 = \frac{1}{2}, w_4 = \frac{1}{2} \tag{2.52}$$

With the previous weights, the array will accept the desired signal while simultaneously rejecting the interference.

While the complex weight selection yields an array pattern that achieves the desired system objectives, it is not a very practical way of approaching adaptive nulling. The method used in the previous example exploits the facts that there is only one directional interference source, that the signal and interference sources are monochromatic, and that a priori information concerning the frequency and the direction of arrival of each signal source is available. A practical processor must work without detailed information about the location, number, and nature of signal sources. Nevertheless, this example has demonstrated that an adaptive algorithm achieves certain performance objectives by adjusting the complex weights. The development of a practical adaptive array algorithm is undertaken in Part 2.

2.3.2 Interference Rejection

Improving the reception of a desired signal requires that the gain in the direction of the desired signal is maximized while the gain in the direction of the interfering signals is minimized. This goal is accomplished by appropriately modifying the signal magnitude or phase at all or some of the elements. Amplitude tapering or thinning an array produces low sidelobes at the requisite angles. An alternative approach is to synthesize nulls in the directions of the interfering signals.

2.3.2.1 Low Sidelobe Amplitude Tapers

The amplitude of interference entering the sidelobes is proportional to the sidelobe level. A binomial amplitude taper [14] applied to an array results in an array factor with no sidelobes. Unfortunately, the binomial array is not practical, because it is extremely inefficient and results in a low-gain main beam.

A better alternative is the Dolph-Chebyshev amplitude taper. All the sidelobes peaks are equal and are at a level determined by the array designer [15]. The following equation provides the zeros of the array factor that correspond to a sidelobe level in dB of *sll*:

$$\psi_n = 2\cos^{-1}\left\{\frac{\cos\left(\frac{(n-0.5)\pi}{N-1}\right)}{\cosh\left(\frac{\pi\zeta}{N-1}\right)}\right\} \tag{2.53}$$

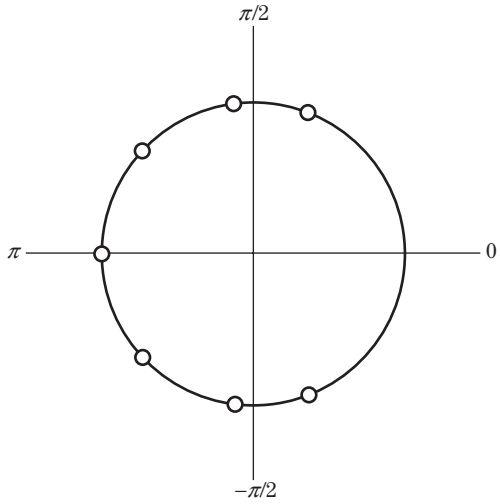


FIGURE 2-13 ■ Unit circle for the eight-element 30 dB Chebyshev taper.

where

$$\zeta = \frac{1}{\pi}\cosh^{-1}\left(10^{sll/20}\right) \tag{2.54}$$

The factored polynomial is found using ψ_n. Once the ψ_n are known, the polynomial of degree $N-1$ in factored form easily follows. The polynomial coefficients are the array amplitude weights.

As an example, designing an eight-element 30 dB Chebyshev array starts with calculating $\zeta = 1.1807$. Substituting into (2.53) results in angular locations (in radians) on the unit circle shown in Figure 2-13 and given by

$$\psi_n = 3.1416, \pm 2.389, \pm 1.1985, \pm 1.6941$$

Finally, the normalized amplitude weights are shown in Figure 2-14 with the corresponding array factor in Figure 2-15.

The Chebyshev taper is not practical for large arrays, because the amplitude weights are difficult to implement in a practical array. The Taylor taper [16] is widely used in the design of low sidelobe arrays. The first $\bar{n}-1$ sidelobes on either side of the main beam are sll, whereas the remaining sidelobes decrease as $1/\sin\theta$. The Taylor taper moves the first $\bar{n}-1$ nulls on either side of the main beam to

$$\theta_n = \begin{cases} \sin^{-1}\left(\frac{\pm\lambda\bar{n}}{Nd}\sqrt{\frac{\zeta^2+(n-0.5)^2}{\zeta^2+(\bar{n}-0.5)^2}}\right) & n < \bar{n} \\ \sin^{-1}\left(\frac{\pm\lambda n}{Nd}\right) & n \geq \bar{n} \end{cases} \tag{2.55}$$

As with the Chebyshev taper, ζ is first found from (2.54). Next, the array factor nulls are found using (2.55) and then substituting into ψ_n. Finally the factored array polynomial is multiplied together to get the polynomial coefficients, which are also the Taylor weights. A Taylor amplitude taper is also available for circular apertures as well [17].

The design of a 16-element 25 dB $\bar{n} = 5$ Taylor taper starts by calculating $\zeta = 1.1366$. Substituting into (2.55) results in angular locations (in radians) on the unit circle shown

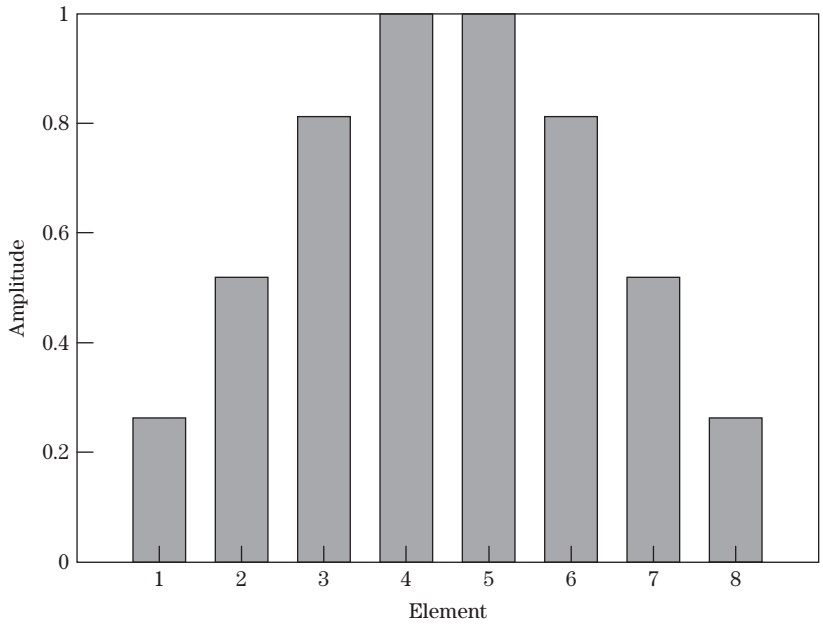


FIGURE 2-14 ■ Amplitude weights for the eight-element 30 dB Chebyshev taper.

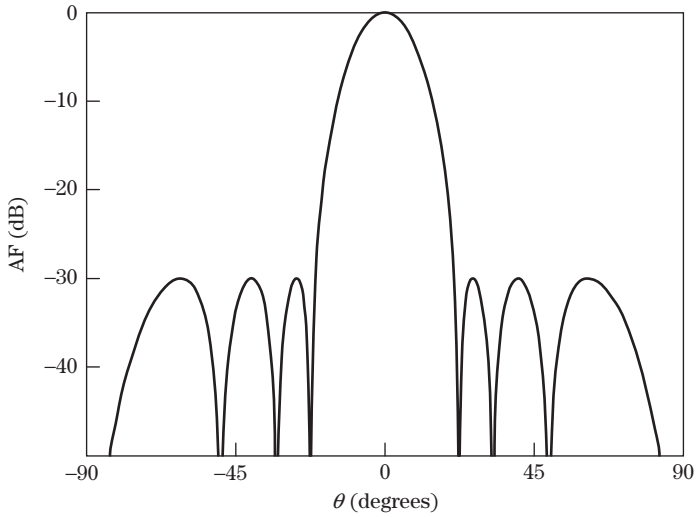


FIGURE 2-15 ■ Array factor for the eight-element 30 dB Chebyshev taper.

in Figure 2-16 and given by

$$\psi_n = 3.1416, \pm 2.7489, \pm 2.3562, \pm 1.9635, \pm 1.5582, \pm 1.1628, \pm 0.79576, \pm 0.52396$$

Finally, the normalized amplitude weights are shown in Figure 2-17 with the corresponding array factor in Figure 2-18.

2.3.2.2 Thinned Arrays

Not only can an amplitude taper lower the sidelobes of an array, but tapering the element density can also mimic an amplitude taper and lower the sidelobes as well. Aperiodic arrays have element spacings that are not periodic [18–24]. Nonuniformly spacing elements in

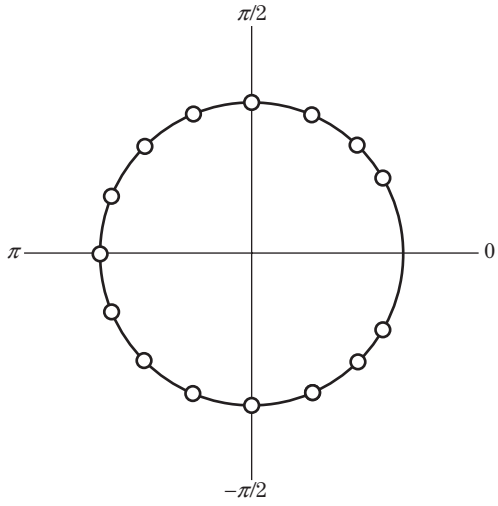


FIGURE 2-16 ■
Unit circle for 20 dB Taylor $\bar{n} = 5$ taper.

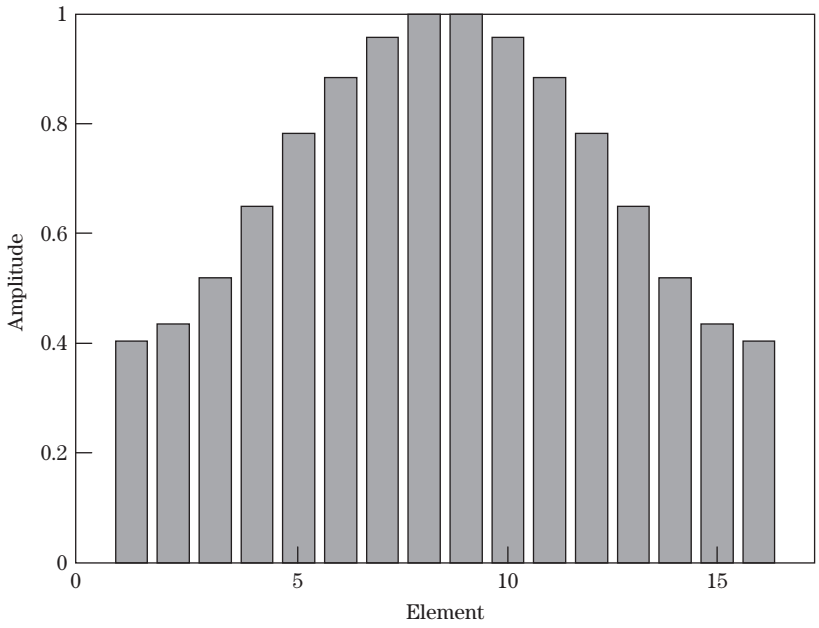


FIGURE 2-17 ■
Amplitude weights for Taylor 25 dB $\bar{n} = 5$ taper.

large arrays is impractical due to the complexity of the feed network and the inconsistency in the mutual coupling environment of the elements. As a result, density tapering in a large array is accomplished by "thinning" or removing active elements from the element lattice in the array aperture.

If the desired amplitude taper function is normalized, then it looks like a probability density function. A uniform random number generator assigns a random number to each element. If that random number exceeds the desired normalized amplitude taper for that element, then the element is passive in the array; otherwise, it is active. An active element is connected to the feed network, whereas an inactive element is not. The advantages of

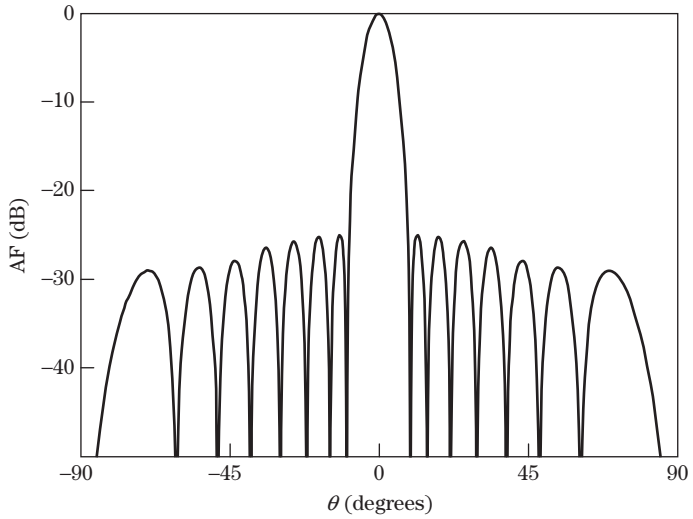


FIGURE 2-18 ■ Array factor for 20 dB Taylor $\bar{n} = 5$ taper.

thinning include the following:

- Simple feed network
- No amplitude weights needed
- Reduced number of active elements for the same beamwidth as a similar fully populated array

An expression for the root mean square (rms) sidelobe level of a statistically thinned array is given by [25]

$$\overline{sll^2} = \frac{1}{N_{active}} \tag{2.56}$$

where $\overline{sll^2}$ is the power level of the average sidelobe level, and N_{active} is the number of active elements out of N elements in the array. An expression for the peak sidelobe level of linear and planar arrays having half-wavelength spacing is found by assuming all the sidelobes are within three standard deviations of the rms sidelobe level and has the form [25]

$$P\left(\text{all sidelobes} < sll_p^2\right) \simeq \left(1 - e^{-sll_p^2/\overline{sll^2}}\right)^{N/2} \tag{2.57}$$

Statistical thinning was used until the 1990s when computer technology enabled engineers to numerically optimize the thinning configuration of an array to find the desired antenna pattern. Current approaches to array thinning include genetic algorithms [26] and particle swarm optimization [27]. In addition, array thinning with realistic elements, like dipoles [28], spirals [29], and thinned planar arrays [30], are used in place of point sources.

As an example, Figure 2-19 shows the array factor for a 100-element thinned linear array with half-wavelength spacing. The thinning was performed using a 25 dB Taylor $\bar{n} = 5$ amplitude taper as the probability density function. The array factor shown has a peak sidelobe level of −18.5 dB below the peak of the main beam and has 77% of the elements turned on. This array factor had the lowest relative sidelobe level of 20 independent runs. The genetic algorithm was used to find the thinning configuration that

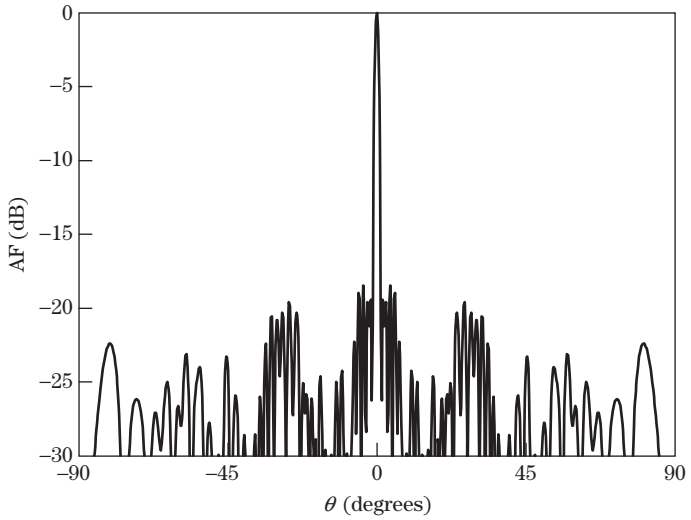


FIGURE 2-19 ■ Statistically thinned 100-element array.

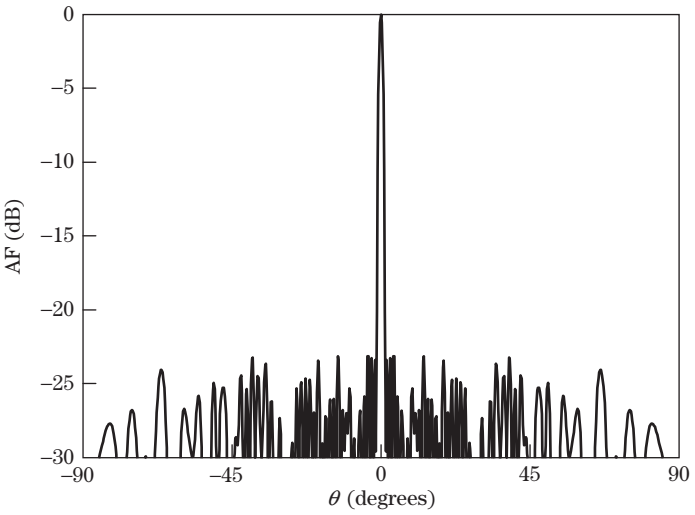


FIGURE 2-20 ■ Genetic algorithm thinned 100-element array.

yields the lowest relative sidelobe level. Figure 2-20 shows the optimized array factor with a -23.2 dB peak relative sidelobe level when 70% of the elements are turned on.

2.3.2.3 Null Synthesis Using the Unit Circle

Just as array factor nulls can be moved to synthesize low sidelobes, they can also be moved in certain directions where interference signals exist. For instance, a null can be placed at -38° of the eight-element uniform array with half-wavelength spacing by moving the zero at $-3\pi/4$ to $\pi \sin(-38^\circ)$ (Figure 2-21(a)), which equates to moving the null at -48.6° to -38° (Figure 2-21(b)). Moving this zero in (2.35) results in an array polynomial with complex weights:

$$\begin{aligned}&\left(z+e^{j\pi/4}\right)\left(z+e^{-j\pi/4}\right)\left(z+e^{j\pi/2}\right)\left(z+e^{-j\pi/2}\right)\left(z+e^{-j\pi\sin(38^\circ)}\right)\left(z+e^{j3\pi/4}\right)\left(z+e^{j\pi}\right)\\&\quad=z^7+(0.65+j0.23)\,z^6+(1.06+j0.32)\,z^5+(0.83-j0.04)\,z^4\\&\qquad+(0.74+j0.37)\,z^3+(1.09+j0.15)\,z^2+(0.68+j0.06)\,z+0.91+j0.41\end{aligned}\tag{2.58}$$

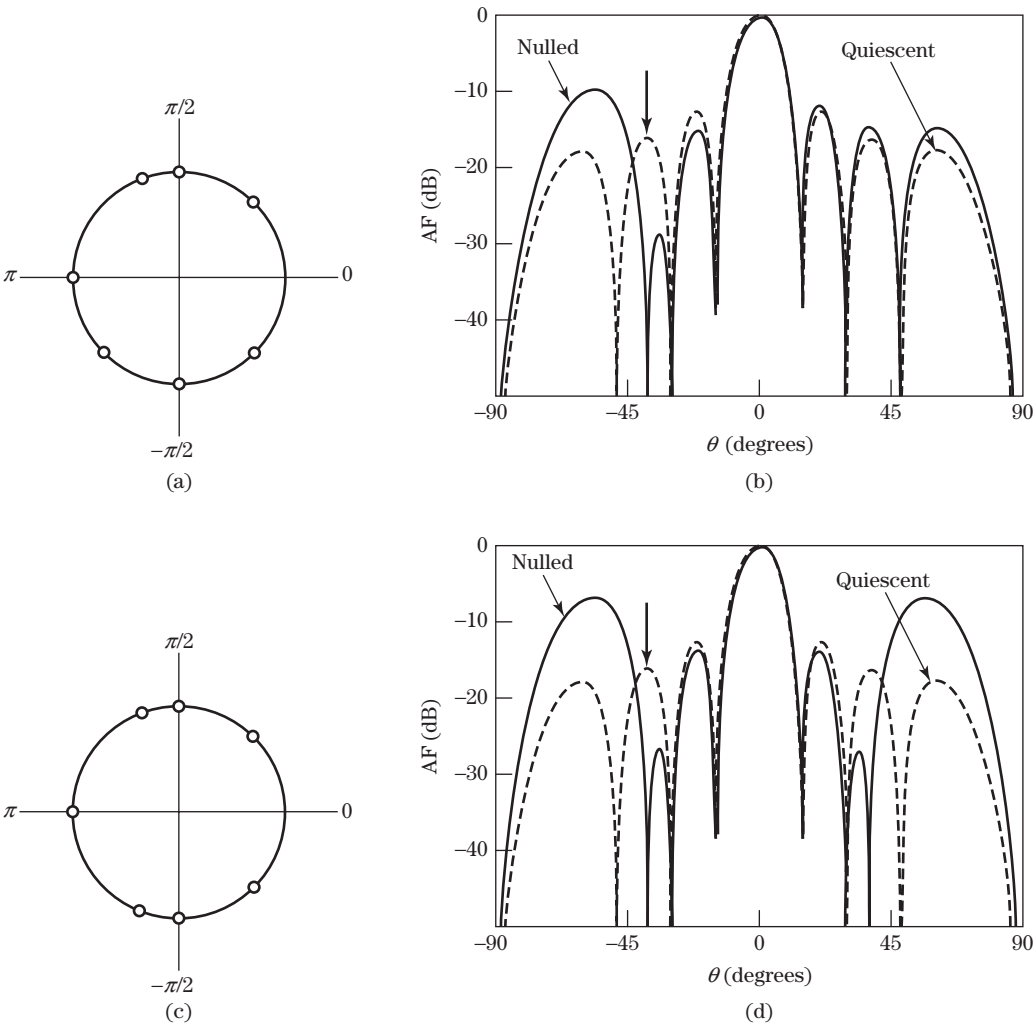


FIGURE 2-21 ■ Null synthesis with the unit circle. a: Moving one zero from $-3\pi/4$ to $\pi \sin(-38^\circ)$. b: Array factor nulls moves from -48.6° to -38°. c: Moving a second zero from $3\pi/4$ to $\pi \sin(38^\circ)$. d: Second-array factor null moves from 48.6° to 38°.

A null can be placed at -38° using real weights by also moving the null in the array factor at 48.6° to 38° to form complex conjugate pairs:

$$\begin{aligned}&\left(z + e^{j\pi/4}\right)\left(z + e^{-j\pi/4}\right)\left(z + e^{j\pi/2}\right)\left(z + e^{-j\pi/2}\right)\left(z + e^{-j\pi \sin(38^\circ)}\right)\left(z + e^{j\pi \sin(38^\circ)}\right)\left(z + e^{j\pi}\right)\\ &\quad = z^7 + 0.30z^6 + 1.29z^5 + 0.59z^4 + 0.59z^3 + 1.29z^2 + 0.30z + 1 \qquad (2.59)\end{aligned}$$

These examples of using null synthesis with the unit circle illustrate several points about adaptive nulling:

1. There are an infinite number of ways to place a null at -38°. After one of the zeros is placed at the desired angle, then the other zeros can be redistributed as well. Thus, placing a null at a given location does not require a unique set of weights unless further restrictions are applied.

2. Some weights produce greater distortion to the array factor than other weights. The weight selection impacts the gain of the array factor as well as the sidelobe levels and null positions.
3. An N element array with complex weights can uniquely place $N-1$ nulls. If the weights are real, then only $N/2-1$ nulls can be uniquely placed.
4. Adaptive algorithms maneuver the nulls on the unit circle indirectly by changing the element weights. The root music algorithm presented in Chapter 10 makes use of the zeros on the unit circle.

The next section shows an approach to null synthesis that results in small changes in the array weights and minimal distortion to the quiescent array factor.

2.3.2.4 Null Synthesis Using Cancellation Beams

Adaptive weights can be written as a perturbation to the quiescent weights

$$w_n = a_n(1+\Delta_n) \tag{2.59}$$

where a_n is the amplitude taper, and Δ_n is the complex weight perturbation that causes the nulls. Substituting (2.59) into the equation for a linear array factor results in a nulled array factor that is the sum of the quiescent array factor and a cancellation array factor [31].

$$\sum_{n=1}^{N} w_n e^{jkx_n u} = \underbrace{\sum_{n=1}^{N} a_n e^{jkx_n u}}_{\text{quiescent array factor}} + \underbrace{\sum_{n=1}^{N} \Delta_n a_n e^{jkx_n u}}_{\text{cancellation array factor}} \tag{2.60}$$

An infinite number of combinations of w_n place nulls in the M desired directions. Setting (2.60) equal to zero and writing the resulting equation in matrix form results in

$$\mathbf{A}\mathbf{W}^T = \mathbf{B}^T \tag{2.61}$$

where

$$\mathbf{A} = \begin{bmatrix} a_1 e^{jkx_1u_1} & a_2 e^{jkx_2u_1} & \cdots & a_N e^{jkx_Nu_1} \\ a_1 e^{jkx_1u_2} & a_2 e^{jkx_2u_2} & \cdots & a_N e^{jkx_Nu_2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 e^{jkx_1u_M} & a_2 e^{jkx_2u_M} & \cdots & a_N e^{jkx_Nu_M} \end{bmatrix}$$

$$\mathbf{W} = [\Delta_1\ \Delta_2 \cdots \Delta_N]$$

$$\mathbf{B} = -\left[\sum_{n=1}^{N} a_n e^{jkx_nu_1} \sum_{n=1}^{N} a_n e^{jkx_nu_2} \cdots \sum_{n=1}^{N} a_n e^{jkx_nu_M}\right]$$

Since (2.61) has more columns than rows, finding the weights requires a least squares solution in the form

$$\mathbf{W} = \mathbf{A}^{\dagger}(\mathbf{A}\mathbf{A}^{\dagger})^{-1}\mathbf{B} \tag{2.62}$$

Ideally, the quiescent pattern should be perturbed as little as possible when placing the nulls. The weights that produce the nulls are [32]

$$w_n = a_n - \sum_{m=1}^{M} \gamma_m c_n e^{-jnkdu_m} \tag{2.63}$$

where

γ_m = sidelobe level of the quiescent pattern at u_m
c_n = amplitude taper of the cancellation beam

When $c_n = a_n$, the cancellation pattern looks the same as the quiescent pattern. When $c_n = 1.0$, the cancellation pattern is a uniform array factor and is the constrained least mean square approximation to the quiescent pattern over one period of the pattern. The nulled array factor can now be written as

$$\sum_{n=1}^{N} w_n e^{jkx_n u} = \sum_{n=1}^{N} a_n e^{jkx_n u} - \sum_{m=1}^{M} \gamma_m \sum_{n=1}^{N} a_n e^{jkx_n(u-u_m)} \tag{2.64}$$

The sidelobes of the cancellation pattern in (2.64) produce small perturbations to the nulled array factor.

As an example, consider an eight-element 30 dB Chebyshev array with interference entering the sidelobe at $\theta = -40^\circ$. When $c_n = a_n$, the cancellation beam is a Chebyshev array factor as shown in Figure 2-22(a), whereas when $c_n = 1.0$, the cancellation beam is a uniform array factor as shown in Figure 2-22(b).

This procedure also extends to phase-only nulling by taking the phase of (2.63). When the phase shifts are small, then $e^{j\phi_n} \approx 1 + j\phi_n$, and the array factor can be written as [33]

$$\sum_{n=1}^{N} w_n e^{jkx_n u} = \sum_{n=1}^{N} a_n e^{j\delta_n} e^{jkx_n u} \approx \sum_{n=1}^{N} a_n (1 + j\delta_n) e^{jkx_n u} = \sum_{n=1}^{N} a_n e^{jkx_n u} + j \sum_{n=1}^{N} a_n \delta_n e^{jkx_n u} \tag{2.65}$$

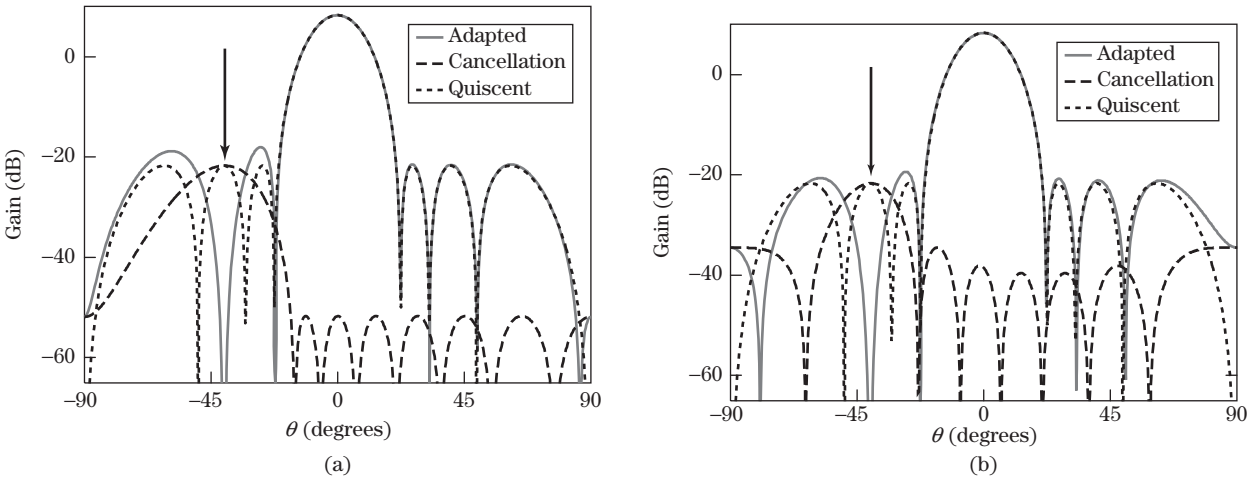


FIGURE 2-22 ■ Array factors for an eight-element array with a 30 dB Chebyshev taper when a null is synthesized at $\theta = -40^\circ$. (a) $c_n = a_n$. (b) $c_n = 1.0$.

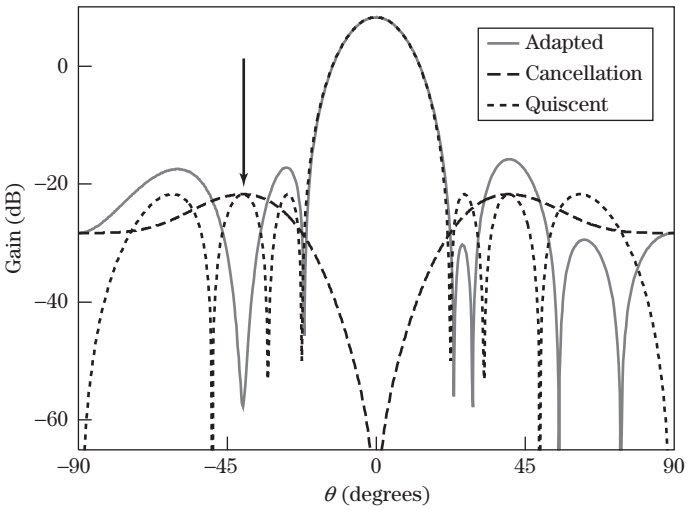


FIGURE 2-23 ■ Array factors for an eight-element array with a 30 dB Chebyshev taper when a phase-only null is synthesized at $\theta = -40^\circ$.

Using Euler's identity, (2.65) is written as

$$\sum_{n=1}^{N} a_n \delta_n \left[\cos\left(kx_n u_m\right) + j \sin\left(kx_n u_m\right)\right] e^{jkx_n u} = j \sum_{n=1}^{N} a_n \left[\cos\left(kx_n u_m\right) + j \sin\left(kx_n u_m\right)\right] \tag{2.66}$$

Equating the real and imaginary parts of (2.66) leads to

$$\sum_{n=1}^{N} a_n \delta_n \sin\left(kx_n u_m\right) e^{jkx_n u} = \sum_{n=1}^{N} a_n \cos(kx_n u_m) \tag{2.67}$$

$$\sum_{n=1}^{N} a_n \delta_n \cos\left(kx_n u_m\right) e^{jkx_n u} = -\sum_{n=1}^{N} a_n \sin(kx_n u_m) = 0 \tag{2.68}$$

Now, (2.67) is in the form that can be solved via (2.62).

Repeating the previous example of an eight-element 30 dB Chebyshev array with interference entering the sidelobe at $\theta = -40^\circ$results in the nulled array factor and associated cancellation beam shown in Figure 2-23. An interesting observation is that the sidelobe at $\theta = 40^\circ$increased. This phenomenon occurs in phase-only nulling unless large phase shifts are used to perform the nulling [34].

The adapted patterns in Figure 2-22 and Figure 2-23 are quite different. If the array polynomials associated with those patterns are factored, then the unit circle represents can be graphed as shown in Figure 2-24(a) and Figure 2-24(b). In both cases, a zero appears on the unit circle at $\psi = -115.7^\circ$. All of the other zeros differ. The zeros in Figure 2-24(a) are on the unit circle, whereas two of the zeros in Figure 2-24(b) are off the unit circle.

A monopulse phased array has separate sum and difference channels. The sum pattern has a peak, whereas the difference pattern has a null in the direction of the desired signal. The sum pattern peak detects the desired signal, and the difference pattern null accurately locates it. Figure 2-25 is a diagram of a monopulse array with sum and difference channels that are split at each element. Prior to the split, both channels share beam steering phase shifters (and amplitude weights) that can be used to simultaneously place nulls in the sum

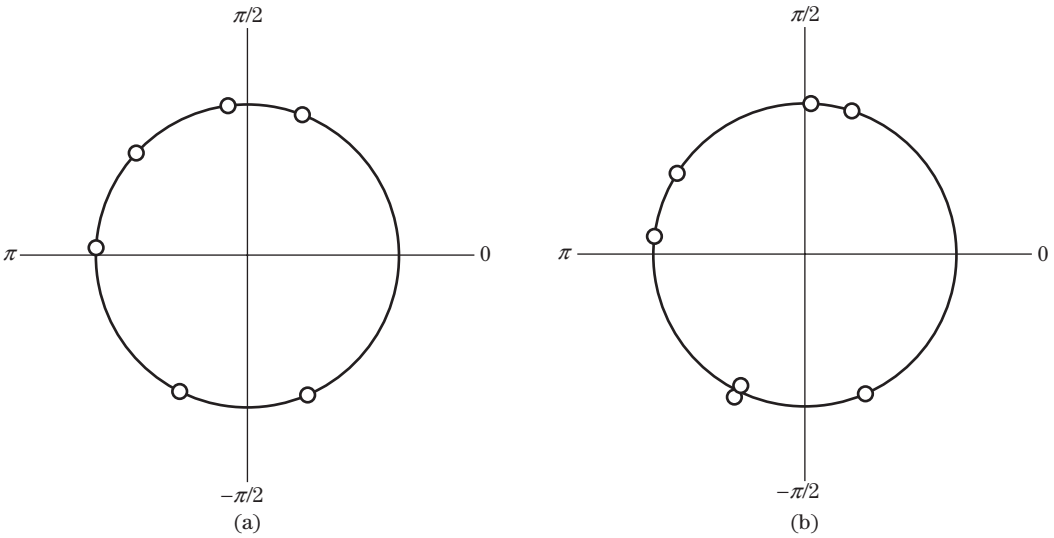


FIGURE 2-24 ■ Unit circle representations of the array factors in (a) Figure 2-22 (b) Figure 2-23.

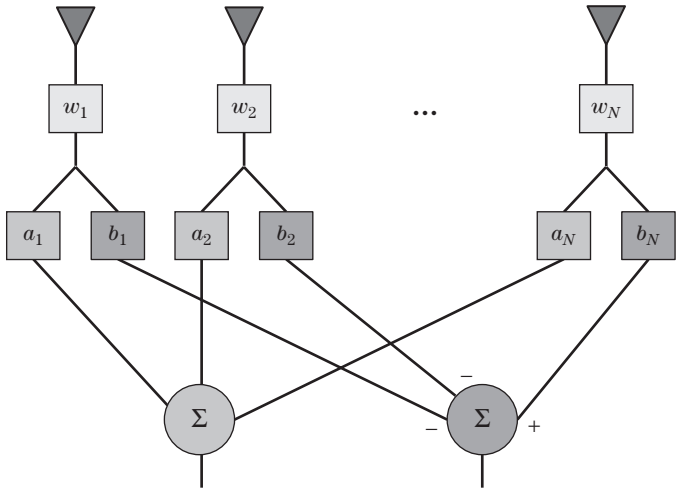


FIGURE 2-25 ■ Diagram of an array with low sidelobe sum and difference channels.

and difference patterns. The matrixes in (2.61) are now written as [35]

$$\mathbf{A} = \begin{bmatrix} a_1 e^{jkx_1u_1} & \cdots & a_N e^{jkx_1u_M} \\ \vdots & \ddots & \vdots \\ a_1 e^{jkx_Nu_1} & \cdots & a_N e^{jkx_Nu_M} \\ b_1 e^{jkx_1u_1} & \cdots & b_N e^{jkx_1u_M} \\ \vdots & \ddots & \vdots \\ b_1 e^{jkx_Nu_1} & \cdots & b_N e^{jkx_Nu_M} \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} \Delta_1 & \Delta_2 & \cdots & \Delta_N \end{bmatrix}$$

$$\mathbf{B} = -j \begin{bmatrix} \sum_{n=1}^{N} a_n e^{jkx_nu_1} & \cdots & \sum_{n=1}^{N} a_n e^{jkx_nu_M} & \sum_{n=1}^{N} b_n e^{jkx_nu_1} & \cdots & \sum_{n=1}^{N} b_n e^{jkx_nu_M} \end{bmatrix}$$

Phase-only nulling requires further simplification as noted in (2.66).

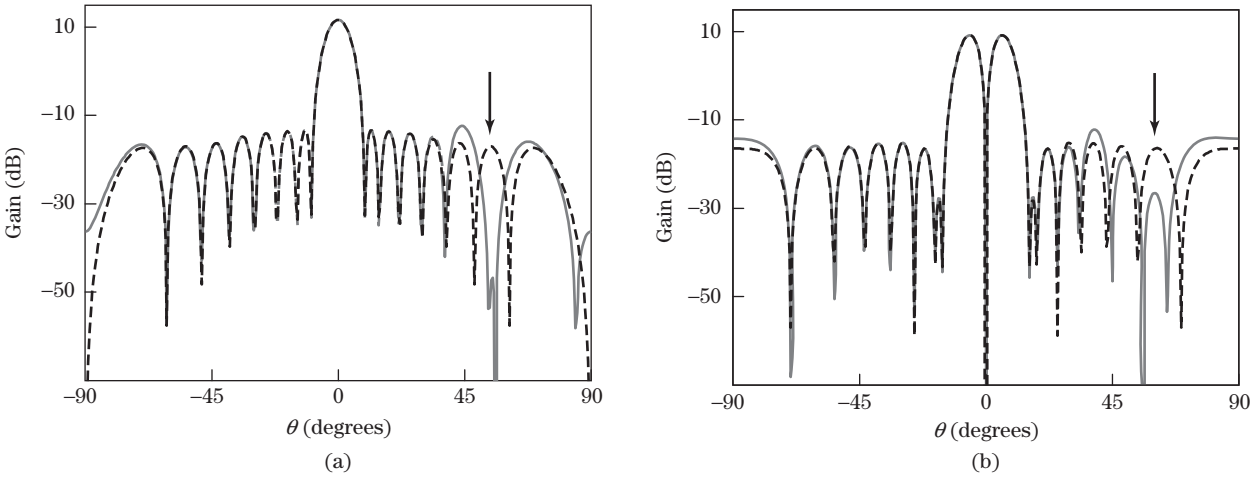


FIGURE 2-26 ■ Nulls are simultaneously placed at $\theta = -40^\circ$ in the sum and difference array factors for a 16-element monopulse array. (a) 25 dB $\bar{n} = 5$ Taylor sum pattern. (b) 25 dB $\bar{n} = 5$ Bayliss difference pattern. The dotted lines are the quiescent patterns.

Consider a 16-element array with a 25 dB $\bar{n} = 5$ Taylor sum pattern and a 25 dB $\bar{n} = 5$ Bayliss difference [11]. Assume that the interference enters the sidelobes at $\theta = 52^\circ$. After synthesizing the shared weights to place the nulls, the resulting sum and difference patterns are shown in Figure 2-26. The resulting patterns have the desired nulls.

2.4 | NULLING LIMITATIONS DUE TO MISCELLANEOUS ARRAY EFFECTS

A principal measure of the effectiveness of an adaptive array in cancelling an undesired interference signal is the ratio of the total output noise power, P_0, to internal noise power, P_N, as a function of frequency. To obtain this ratio for a simple example, consider the two-element adaptive array of Figure 2-27. If the interference signal arriving at element 1 from the direction θ is $s(t)$, then the signal appearing at element 2 is $s(t + \tau)$, where $\tau = (d/v) \sin\theta$, and v = wavefront propagation velocity.

The signals in the two element channels are weighted and summed to form the array output $y(t)$ so that

$$y(t) = w_1 s(t) + w_2 s(t + \tau) \tag{2.69}$$

which can be written in the frequency domain as

$$Y(\omega) = S(\omega)[w_1 + w_2 e^{-j\omega\tau}] \tag{2.70}$$

To exactly cancel the interference signal at the array output at a particular frequency, f_0 (termed the center frequency of the jamming signal), it is necessary that

$$w_2 = -w_1 e^{j\omega_0\tau} \tag{2.71}$$

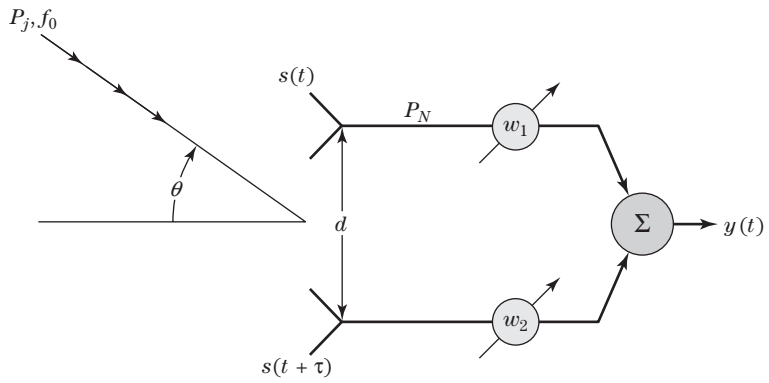


FIGURE 2-27 ■ Two-element adaptive array with jammer located at θ.

If we select the two adaptive weights to satisfy (2.69), it follows that the interference signal component of the array output at any frequency is given by

$$Y(\omega) = S(\omega)w_1[1 - e^{-j\tau(\omega-\omega_0)}] \tag{2.72}$$

and consequently the interference component of output power at any frequency may be expressed as (assuming momentarily for simplicity that $|S(\omega)|^2 = 1$)

$$|Y(\omega)|^2 = |w_1|^2\{2 - 2cos[\tau(\omega - \omega_0)]\} \tag{2.73}$$

Let $|S(\omega)|^2 = P_J$ and denote the constant spectral density of the internal noise power in each channel by P_N, it follows that the total output noise power spectral density, $P_o(\omega)$, can be written as

$$P_o(\omega) = 2|w_1|^2[1 - \cos(\tau(\omega - \omega_0))]P_J + 2|w_1|^2 P_N \tag{2.74}$$

If we recognize that the output thermal noise power spectral density is just $P_n = 2|w_1|^2 P_N$ and normalize the previous expression to P_n, it follows that the ratio of the total output interference plus noise power spectral density, P_0, to the output noise power spectral density, P_n, is then given by

$$\frac{P_0}{P_n}(\omega) = 1 + \frac{\{1 - \cos[\tau(\omega - \omega_0)]\}P_J}{P_N} \tag{2.75}$$

Where

P_J/P_N = jammer power spectral density to internal noise power spectral density per channel
$\tau = (d/v)\sin\theta$
d = sensor element spacing
θ = angular location of jammer from array broadside
ω_0 = center frequency of jamming signal

At the center frequency, a null is placed exactly in the direction of the jammer so that $P_0/P_n = 1$. If, however, the jamming signal has a nonzero bandwidth, then the other frequencies present in the jamming signal will not be completely attenuated. Figure 2-28 is a plot of (2.75) and shows the resulting output interference power density in decibels as

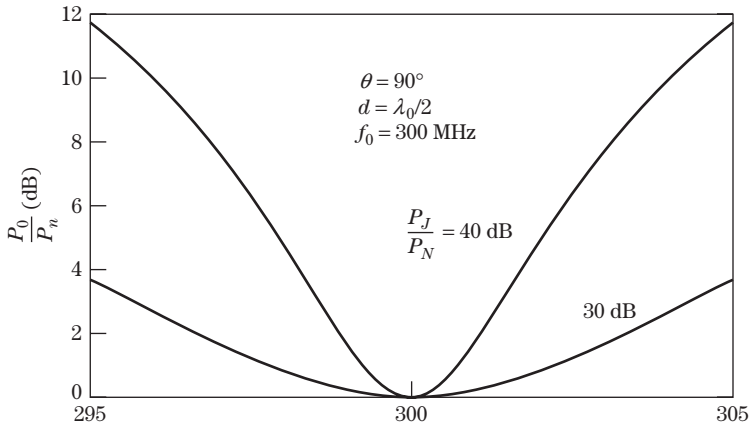


FIGURE 2-28 ■ Two-element cancellation performance: P_0/P_n versus frequency for $P_J/P_N = 30$ and 40 dB.

a function of frequency for a 10 MHz bandwidth jamming signal located at $\theta = 90°$ and an element spacing of $\lambda_0/2$ for two values of P_J/P_N. The results shown in Figure 2-28 indicate that for $P_J/P_N = 40$ dB about 12 dB of uncanceled output interference power remains at the edges of the 10 MHz band (with $f_0 = 300$ MHz). Consequently, the output residue power at that point is about $40 - 12 = 28$ dB below the interfering signal that would otherwise be present if no attempt at cancellation were made.

It can also be noted from (2.75) and Figure 2-28 that the null bandwidth decreases as the element spacing increases and the angle of the interference signal from array broadside increases. Specifically, the null bandwidth is inversely proportional to the element spacing and inversely proportional to the sine of the angle of the interference signal location.

Interference signal cancellation depends principally on three array characteristics: (1) element spacing; (2) interference signal bandwidth; and (3) frequency-dependent interelement channel mismatch across the cancellation bandwidth. The effect of sensor element spacing on the overall array sensitivity pattern has already been discussed. Yet another effect of element spacing is the propagation delay across the array aperture.

Assume that a single jammer having power P_J is located at θ and that the jamming signal has a flat power spectral density with (double-sided) bandwidth B Hz. It may be shown that the cancellation ratio P_0/P_J, where P_0 is the (canceled) output residue power in this case, is given by

$$\frac{P_0}{P_j} = 1 - \frac{\sin^2(\pi B\tau)}{(\pi B\tau)^2} \tag{2.76}$$

where τ is given by (2.8). Note that (2.76) is quite different from (2.75), a result that reflects the fact that the weight value that minimizes the output jammer power is different from the weight value that yields a null at the center frequency. Equation (2.76) is plotted in Figure 2-29, which shows how the interference signal cancellation decreases as the signal bandwidth–propagation delay product, $B\tau$, increases. It is seen from (2.76) that the cancellation ratio P_0/P_J is inversely proportional to the element spacing and the signal bandwidth.

An elementary interchannel amplitude and phase mismatch model for a two-element array is shown in Figure 2-30. Ignoring the effect of any propagation delay, the output

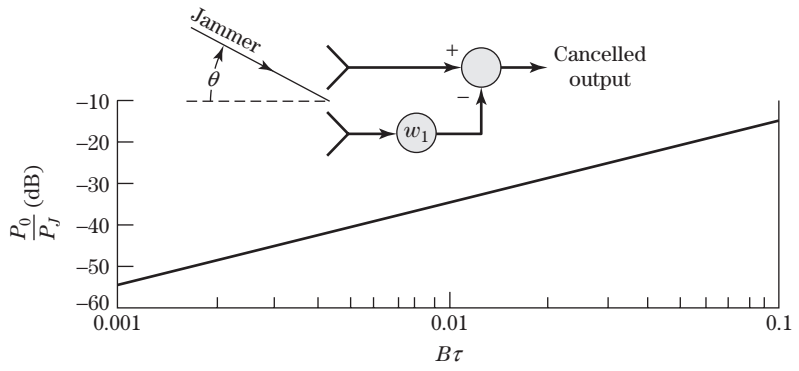


FIGURE 2-29 ■ Cancellation ratio versus jammer signal bandwidth–propagation delay product $B\tau$.

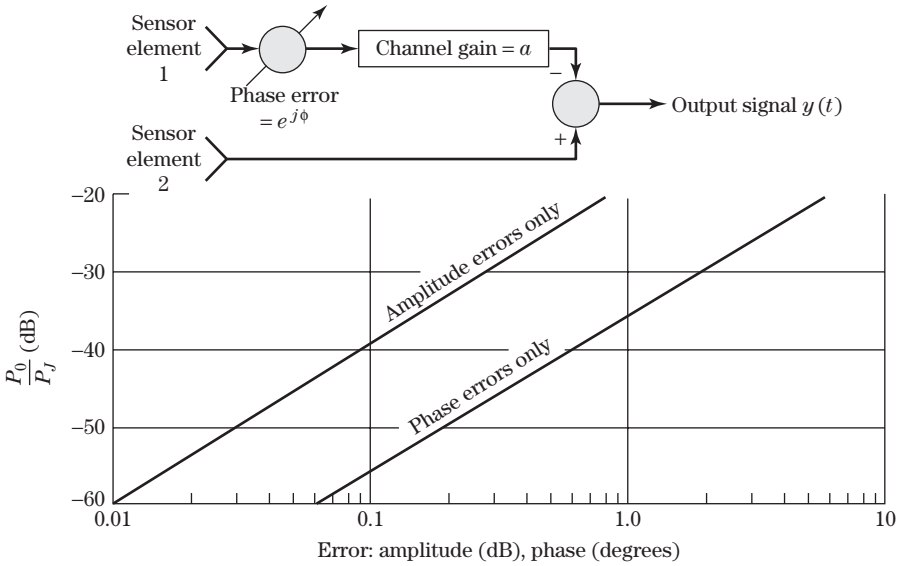


FIGURE 2-30 ■ Cancellation ratio P_0/P_J versus amplitude and phase channel mismatch for a two-element array.

signal may be expressed as

$$y(t) = 1 - ae^{j\phi} \tag{2.77}$$

From (2.77) it follows that the canceled output power is given by

$$|y(t)|^2 = |1 - ae^{j\phi}|^2 = 1 + a^2 - 2a\cos\phi \tag{2.78}$$

Figure 2-30 shows a plot of the cancellation ratio P_0/P_J based on (2.78) for amplitude errors only ($\phi = 0$) and phase errors only ($a = 1$). In Part 3, where adaptive array compensation is discussed, more realistic interchannel mismatch models are introduced, and means for compensating such interchannel mismatch effects are studied. The results obtained from Figure 2-30 indicates that the two-sensor channels must be matched to within about 0.5 dB in amplitude and to within approximately 2.8° in phase to obtain 25 dB of interference signal cancellation.

2.5 | NARROWBAND AND BROADBAND SIGNAL PROCESSING CONSIDERATIONS

We have seen that by adjusting the complex weights in the two-element array of Figure 2-12 it is possible to modify the overall array pattern to place a pattern null in the direction of an undesired interference signal. So long as the interference signal is "narrowband" (i.e., can be adequately characterized by a single frequency, ω_0), the resulting array pattern will succeed in suppressing the undesired interference signal. The most common to implement a complex weight is a quadrature hybrid circuit (Figure 2-31). Each individual sensor element is divided into an in-phase and quadrature-phase component by means of a 90° phase shifter. Each component is then applied to a variable weight whose output is summed to form the output signal. The resulting complex gain factor is then $Ae^{j\phi}$, where $\phi = -\tan^{-1}(w_2/w_1)$, and $A = \sqrt{w_1^2 + w_2^2}$. The weights w_1, w_2 assume a continuum of both positive and negative values, and the range of magnitude A is limited only by the range of limitations of the two individual weights. It should be recognized that it is not absolutely necessary to maintain a phase shift of exactly 90° to obtain quite satisfactory processing with quadrature hybrid circuitry.

In the event that the interference signal is broadband, then the complex weight selection appropriate for one frequency ω_1 will not be appropriate for a different frequency ω_2, since the array pattern nulls shift as the value of λ_0 changes. Figure 2-32 shows the array factor expanded around the null placed at 52° for the center frequency (f_0), 5% below the center frequency ($0.95f_0$), and 5% above the center frequency ($1.05f_0$). This example

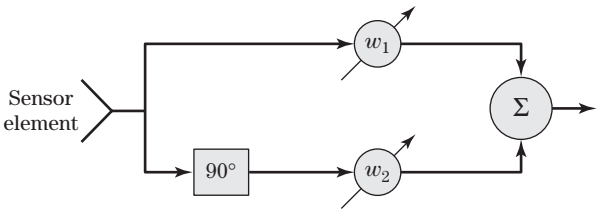


FIGURE 2-31 ■ Realization of a complex weight by means of a quadrature hybrid circuit.

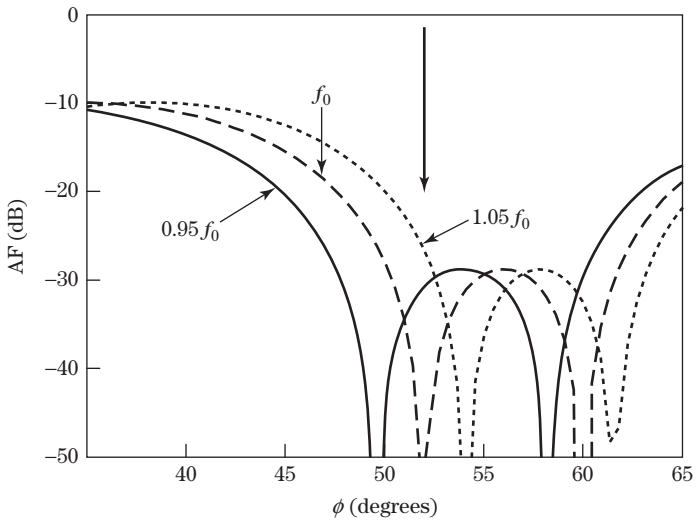


FIGURE 2-32 ■ The region around the null at 52° over a 10% bandwidth.

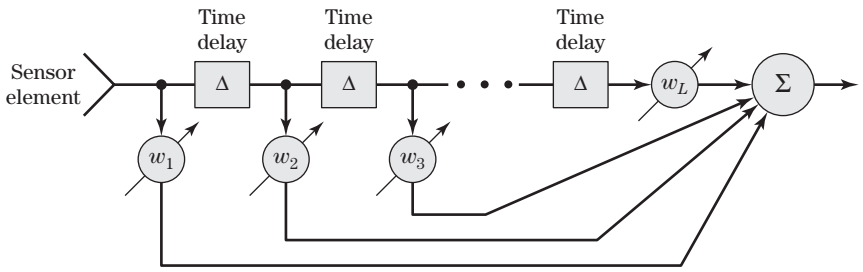


FIGURE 2-33 ■ Transversal filter realized with a tapped-delay line having L complex weights.

demonstrates that the desired null is narrowband, because it moves as the frequency changes. This fact leads to the conclusion that different complex weights are required at different frequencies if an array null is to be maintained in the same direction for all frequencies of interest. A simple and effective way of obtaining different amplitude and phase weightings at a number of frequencies over the band of interest is to replace the quadrature hybrid circuitry of Figure 2-31 by a transversal filter having the transfer function $\mathrm{h}(\omega)$. Such a transversal filter consists of a tapped-delay line having L complex weights as shown in Figure 2-33 [33,34]. A tapped-delay line has a transfer function that is periodic, repeating over consecutive filter bandwidths B_f, as shown in Appendix A. If the tap spacing is sufficiently close and the number of taps is large, this network approximates an ideal filter that allows exact control of gain and phase at each frequency within the band of interest. An upper limit on the tap spacing is given by the desired array cancellation bandwidth B_a, since $B_f \geq B_a$ and $B_f = 1/\Delta$ for uniform tap spacing. The transversal filter not only is useful for providing the desired adjustment of gain and phase over the frequency band of interest for wideband signals but is also well suited for providing array compensation for the effects of multipath, finite array propagation delay, and interchannel mismatch effects; these additional uses of transversal filters are explored further in Part 3. Transversal filters are also enhance the ability of an airborne moving target indication (AMTI) radar system to reject clutter, to compensate for platform motion, and to compensate for near-field scattering effects and element excitation errors [35].

To model a complete multichannel processor (in a manner analogous to that of Section 1.4 for the narrowband processing case), consider the tapped-delay line multichannel processor depicted in Figure 2-34. The multichannel wideband signal processor consists of N sensor element channels in which each channel contains one tapped-delay line like that of Figure 2-33 consisting of L tap points, $(L-1)$ time delays of Δ seconds each, and L complex weights. On comparing Figure 2-34 with Figure 1-1, it is seen that $x_1(t), x_2(t), \cdots x_N(t)$ in the former correspond exactly with $x_1(t), x_2(t) \ldots x_N(t)$ in the latter, which were defined in (1.6) to form the elements of the vector $\mathbf{x}(t)$. In like manner, therefore, define a complex vector $\mathbf{x}'_1(t)$ such that

$$(\mathbf{x}'_1)^T = (\mathbf{x}')^T = [x_1 x_2 \cdots x_N] \tag{2.79}$$

The signals appearing at the second tap point in all channels are merely a time-delayed version of the signals appearing at the first tap point, so define a second complex signal vector $\mathbf{x}'_2(t)$ where

$$\begin{aligned}[\mathbf{x}'_2(t)]^T &= [\mathbf{x}'(t-\Delta)]^T \\ &= [x_1(t-\Delta)x_2(t-\Delta)\cdots x_N(t-\Delta)]\end{aligned} \tag{2.80}$$

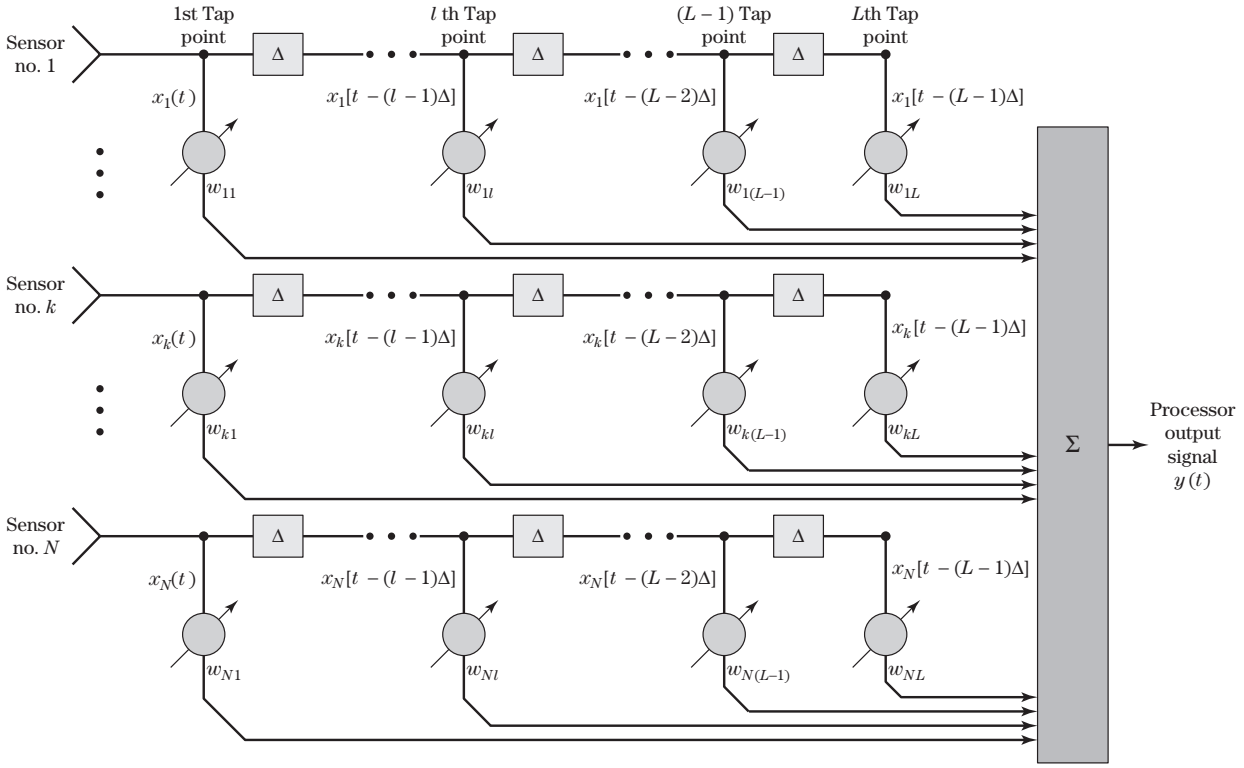


FIGURE 2-34 ■ Tapped-delay line multichannel processor for wideband signal processing.

Continuing in the previously given manner for all L tap points, a complete signal vector for the entire multichannel processor can be defined as

$$\mathbf{x}(t)\Delta = \begin{bmatrix} \mathbf{x}'_1(t) \\ \mathbf{x}'_2(t) \\ \vdots \\ \mathbf{x}'_L(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}'(t) \\ \mathbf{x}'(t-\Delta) \\ \vdots \\ \mathbf{x}'[t-(L-1)\Delta] \end{bmatrix} \tag{2.81}$$

It is seen that the signal vector $\mathbf{x}(t)$ contains L component vectors each having dimension N.

Define the weight vector

$$(\mathbf{w}'_1)^T \triangleq [w_{11} w_{21} \cdots w_{N1}] \tag{2.82}$$

The weight vector for the entire multichannel processor is then given by

$$\mathbf{w} \triangleq \begin{bmatrix} \mathbf{w}'_1 \\ \mathbf{w}'_2 \\ \vdots \\ \mathbf{w}'_L \end{bmatrix} \tag{2.83}$$

It is seen that the weight vector $\mathbf{w}$ so defined is a NL-dimensional vector containing L component vectors, each having dimension N.

As a consequence of the signal and weight vector definitions previously introduced, the output of the multichannel tapped-delay line processor can be written in the form

$$\begin{aligned} y(t) &= \sum_{l=1}^{L} (\mathbf{w}_l')^T \mathbf{x}_l'(t) = \sum_{l=1}^{L} (\mathbf{w}_l')^T \mathbf{x}'[t - (l-1)\Delta] \\ &= \mathbf{w}^T \mathbf{x}(t) \end{aligned} \tag{2.84}$$

which is the same form as (1.4). Yet another reason for expressing the signal vector in the form of (2.81) is that this construction leads to a Toeplitz form for the correlation matrix of the input signals, as shown in Chapter 3.

The array processors of Figures 2-33 and 2-34 are examples of classical time-domain processors that use complex weights and delay lines. Fast Fourier transform (FFT) techniques replace a conventional time-domain processor by an equivalent frequency domain processor using the frequency-domain equivalent of time delay [39–41]. When a tapped delay-line array and an FFT array use the same time delay between samples and the same number of samples, their performance is identical. FFT processing has two advantages compared with its time-domain equivalent: (1) the hardware problem can be alleviated; and (2) the computational burden can be reduced. The use of phase shifters or delay lines to form a directional beam in the time domain becomes cumbersome from a hardware standpoint as the number of delay elements increases, whereas using the frequency-domain equivalent permits beamforming to be accomplished with a digital computer, thereby simplifying the hardware. Likewise, taking FFTs tends to reduce the correlation between samples in different frequency subbands. When samples in different subbands are completely decorrelated, the signal covariance matrix has a block diagonal form, and the optimal weights can be computed separately in each subband, thus reducing the computational burden.

As more taps are added to each delay line, the bandwidth performance of the array improves. The bandwidth cutoff, B_c, of an array is defined as the maximum bandwidth such that the array SINR is within 1dB of its continuous wave (narrowband) value. It is better to divide the taps equally among the array elements rather than to employ an unequal distribution, although higher bandwidth performance can usually be realized by placing an extra tap behind the middle element in an array. A piecewise linear curve is plotted in Figure 2-35, giving the optimal number of taps per element (number of samples for FFT processing) versus signal fractional bandwidth, B, for 3 to 10 elements in a linear array ($B = BW/carrier\ frequency$). Since it is easiest to use 2^n samples in an FFT processor, where n is some integer, one would use the smallest value of n such that 2^n is at least as large as the value shown in Figure 2-35. The time delay between taps, T_o, can be described in terms of the parameter r, where r is the number of quarter wavelength delays found in T_o. For optimal SINR array performance, the value of r must be in the range $0 < r < 1/B$. If $r < 1/B$, the performance is optimal but results in large array weights. Therefore, the choice $r = 1/B$ is dictated by good engineering practice. The notion of the frequency-domain equivalent of time delay is introduced in the Problems section, and more detailed presentations of the techniques of digital beamforming may be found in [42–45].

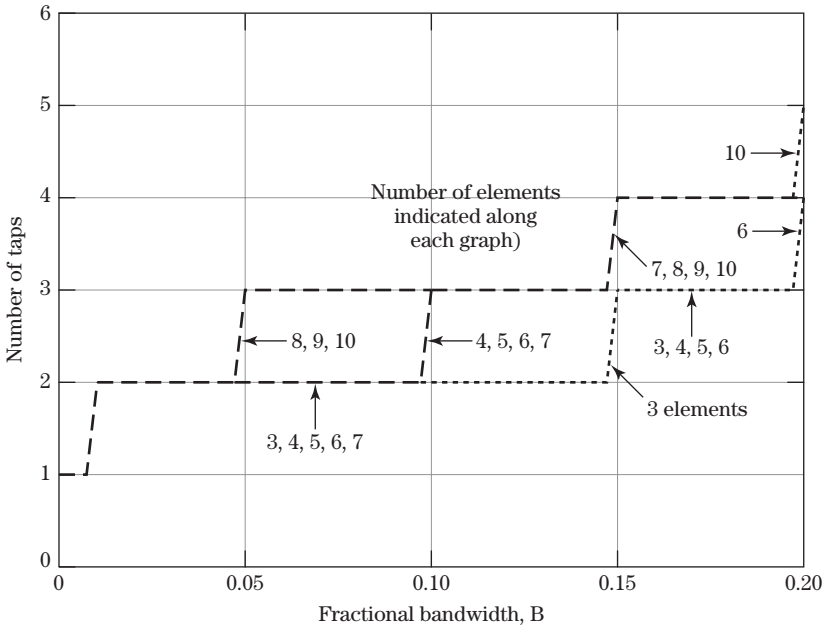


FIGURE 2-35 ■ Number of taps versus fractional bandwidth, *B*. From Vook and Compton, IEEE Trans. Aerosp. and Electron. Sys., Vol. 38, No. 3, July 1992.

2.6 | ADAPTIVE ARRAY PERFORMANCE MEASURE—COVERAGE IMPROVEMENT FACTOR (CIF) [44]

At any point over the array field of view, the value of the jammer-to-signal ratio can be computed as

$$\mathrm{J/S} = \frac{\sum_{i=1}^{n} G_i P_i + N_0}{G_s P_s} \tag{2.85}$$

where P_i is the ith jammer's power at the face of the array, G_i is the array gain toward the ith jammer, N_0 is the thermal noise power, P_s is the desired signal power at the array face, G_s is the array gain toward the desired signal, and n is the number of jammers present.

A typical J/S sky contour plot over a planar array field of view (in this case the upper hemisphere) is shown in Figure 2-36 The circle edges in this figure occur at an elevation angle of 0°, and the apex of the hemisphere (elevation angle of 90°) occurs at the circle center with other values of constant elevation angle forming concentric circles inside the outer circle. The contours of constant J/S values in this figure indicate that four jammers are present: three near the horizon at azimuth values of about 20°, 80°, and 110°, and one at about 60° elevation and 20° azimuth. Plotting the percentage of the sky for which J/S is less than a specified level then yields the J/S coverage statistic plot of Figure 2-37, which corresponds to the J/S contours of Figure 2-36 across the sky azimuth and elevation. Figure 2-37 shows that a J/S ratio of 46 dB is achieved over 72% of the sky.

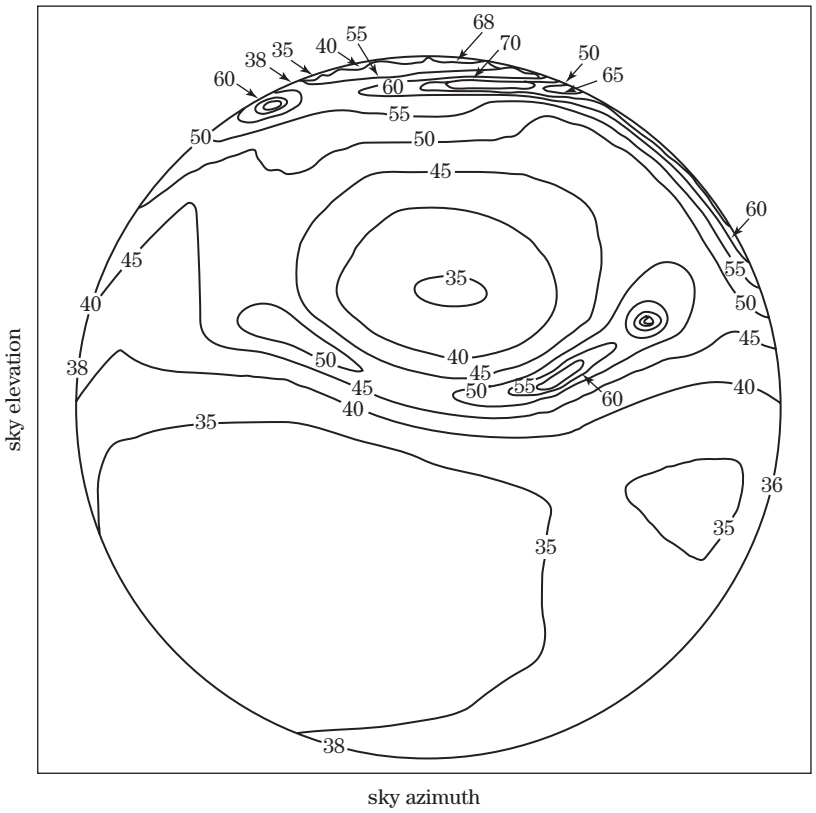


FIGURE 2-36 ■ J/S sky contour plot. From Weiss, Aerospace Corp. Technical Report, TOR-95(5474)-1, 20 December 1994

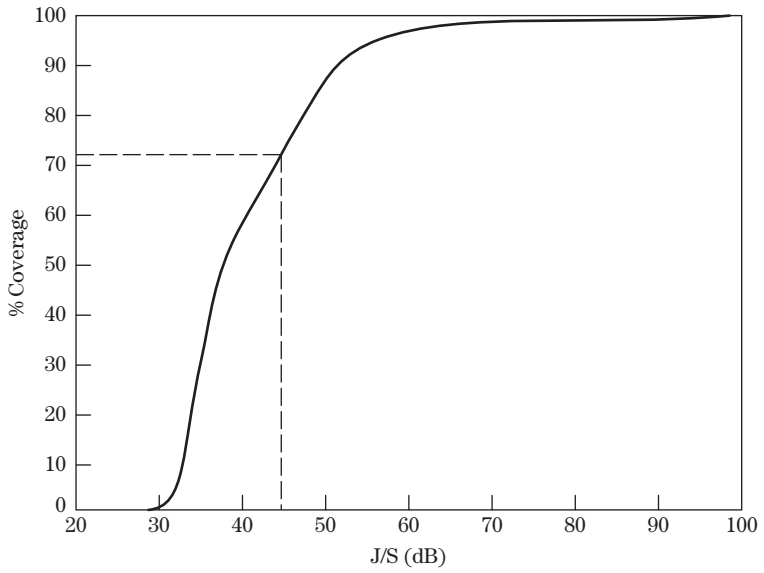


FIGURE 2-37 ■ J/S coverage statistics for Figure 2-36. From Weiss, Aerospace Corp. Technical Report, TOR-95(5474)-1, 20 December 1994.

The coverage improvement factor (CIF) of the array at each point in the field of view can be computed as

$$\text{CIF} = \frac{\left.\dfrac{\sum_{i=1}^{n} G_{iq} P_i + N_{0q}}{G_{sq} P_s}\right|_{\textit{before nulling}}}{\left.\dfrac{\sum_{i=1}^{n} G_i P_i + N_0}{G_s P_s}\right|_{\textit{after nulling}}} \tag{2.86}$$

where the numerator gains G_{iq} and G_{sq} denote the array gain toward the ith jammer and the desired signal, respectively, for the quiescent pattern case, and N_{0q} denotes the thermal noise power before nulling.

The coverage improvement factor statistic (CIFS) can then be obtained by plotting the percentage of the field of view where the nulling system provides X dB of protection compared with the quiescent system in the same manner as the J/S coverage statistic plot was obtained and yields a convenient way of assessing how effective the adaptive array is in yielding a performance advantage over a nonadaptive array.

2.7 SUMMARY AND CONCLUSIONS

The signal environment in which an adaptive array operates is introduced by discussing the different types of interference and modeling of the desired signal as either a random or nonrandom process. Signal propagation effects are also important because of the distortion introduced into the desired signal, and the rate of change of variations in the ambient noise environment directly affects the ability of an adaptive system to cope with a nonstationary signal environment by adjusting the pattern-forming network elements.

Sensor element spacing within an array directly affects its ability to respond appropriately to any given signal environment and to yield enhanced reception performance. The potential of an array to offer enhanced reception performance is demonstrated with a simple example. The presence of undesirable pattern grating lobes may be avoided either by using a "filled array" for systems having a small number of elements or by using a "thinned array" by exploiting the theory of randomly spaced elements for large arrays to achieve an efficient design. In addition to element spacing, interference signal nulling limitations may also derive from the following:

1. Jammer signal bandwidth
2. Signal propagation delay across the array aperture
3. Interchannel mismatch effects

Narrowband and broadband signal processing requirements differ in that narrowband processing can be achieved by means of a single constant complex weight in each element channel whereas a broadband signal requires frequency-dependent weighting that leads to the introduction of a transversal filter (tapped delay line). Time-domain processors can be replaced by equivalent frequency-domain processors whose functions are realized by a digital computer, thereby alleviating the need for cumbersome hardware.

A convenient way of assessing the effectiveness of an adaptive array compared with a nonadaptive array was discussed by introducing the concept of the CIF and its associated CIFS.

The best possible steady-state performance that can be achieved by an adaptive array system can be computed theoretically, without explicit consideration of the array factors affecting such performance. The theoretical performance limits for steady-state array operation are addressed in Chapter 3.

2.8 PROBLEMS

1. ***Circular Array.*** For the circular array of Figure 2-38 having N equally spaced elements, select as a phase reference the center of the circle. For a signal whose direction of arrival is θ with respect to the selected angle reference, define the angle ϕ_k for which $\phi_k = \theta - \psi_k$.

(a) Show that $\psi_k = (2\pi/N)(k-1)$ for $k = 1, 2, \ldots, N$.

(b) Show that the path length phase difference for any element with respect to the center of the circle is given by $u_k = \pi(R/\lambda)\cos\phi_k$.

(c) Show that an interelement spacing of $\lambda/2$ is maintained by choosing the radius of the array circle to be

$$R = \frac{\lambda}{4\sin(\pi/N)}$$

2. ***Linear Array with Spherical Signal Wavefront.*** Consider the linear array geometry of Figure 2-39 in which $\mathbf{v}$ is the vector from the array center to the signal source, θ is the angle between $\mathbf{v}$ and the array normal, and υ is the propagation velocity.

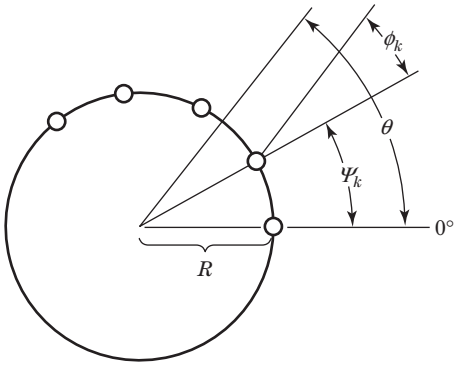


FIGURE 2-38 ■ Azimuth plane of circular array having N equally spaced elements.

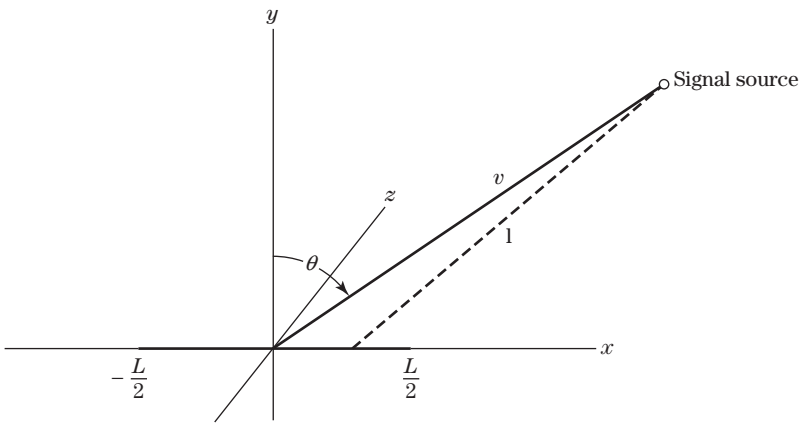


FIGURE 2-39 ■ Linear array geometry.

(a) Show that for a spherical signal wavefront the time delay experienced by an element located at position x in the array with respect to the array center is given by

$$\tau(x) = \frac{||\mathbf{v}||}{\mathfrak{v}} \left\{ 1 - \sqrt{1 + \left(\frac{x^2}{||\mathbf{v}||^2}\right) - 2\left(\frac{x}{||\mathbf{v}||}\right)\sin\theta} \right\}$$

(τ is positive for time advance and negative for time delay) where $||\mathbf{v}||$ denotes the length of the vector $\mathbf{v}$.

(b) Show that for a spherical signal wavefront the attenuation factor experienced by an element located at position x in the array with respect to the array center is given by

$$\rho(x) = \frac{||\mathbf{v}||}{||\mathbf{l}||} = \frac{1}{\sqrt{1 + \left(\frac{x^2}{||\mathbf{v}||^2}\right) - 2\left(\frac{x}{||\mathbf{v}||}\right)\sin\theta}}$$

where $||\mathbf{l}||$ denotes the distance between the signal source and the array element of concern.

3. ***Uniformly Spaced Linear Array [11].*** Show that for a uniformly spaced linear array having N elements the normalized array factor is given by

$$AF(u) = \frac{1}{N}\frac{\sin(Nu/2)}{\sin(u/2)}$$

where $u = 2\pi(d/\lambda)\sin\theta$.

4. ***Representation of Nonuniformly Spaced Arrays [20]***

(a) For nonuniformly spaced elements in a linear array, a convenient "base separation" d can be selected and the element position then represented by

$$d_i = \left(\frac{i}{2} + \varepsilon_i\right) d$$

where ε_i is the fractional change from the uniform spacing represented by d. The normalized field pattern for a nonuniformly spaced array is then given by

$$AF = \frac{1}{N}\sum_{i=1}^{N} \cos u_i = \frac{1}{N}\sum_{i=1}^{N} \cos\left[\left(\frac{i}{2} + \varepsilon_i\right) u\right]$$

where $u = 2\pi(d/\lambda)\sin\theta$. Show that AF can be put in the form

$$AF = AF_u - \frac{1}{N}\sum_i \left[\sin\varepsilon_i u \sin i\frac{u}{2} + (1 - \cos\varepsilon_i u)\cos i\frac{u}{2}\right]$$

where A_u is the pattern of a uniform array having element separation equal to the base separation and is given by the result obtained in Problem 3.

(b) Assuming all $\varepsilon_i u$ are small, show that the result obtained in part (a) reduces to

$$AF = AF_u - \frac{u}{N}\sum_i \varepsilon_i \sin i\left(\frac{u}{2}\right)$$

(c) The result obtained in part (b) can be rearranged to read

$$\sum_i \varepsilon_i \sin\left(i\frac{u}{2}\right) = \frac{N}{u}(AF_u - AF)$$

which may be regarded as a Fourier series representation of the quantity on the right-hand side of the previous equation. Consequently, the ε_i are given by the formula for Fourier coefficients:

$$\varepsilon_i = \frac{2N}{\pi}\int_{\pi}^{0}\frac{1}{u}(AF_u - AF)\sin\left(i\frac{u}{2}\right)du$$

Let

$$\frac{AF_u - AF}{u} = \frac{1}{u}\sum_{k=1}^{L} a_k\delta(u - u_k)$$

and show that

$$\varepsilon_i = 2\frac{N}{\pi}\sum_{k=1}^{L} a_k\frac{\sin[(i/2)u_k]}{u_k}$$

(d) For uniform arrays the pattern sidelobes have maxima at positions approximately given by

$$u_k = \frac{\pi}{N}(2k+1)$$

Consequently, u_k gives the positions of the necessary impulse functions in the representation of $(AF_u - AF)/u$ given in part (c). Furthermore, since the sidelobes adjacent to the main lobe drop-off as $1/u$, $(AF_u - AF)/u$ is now given by

$$\frac{AF_u - AF}{u} = AF\frac{1}{u^2}\sum_{k=1}^{L}(-1)^k\delta\left[u - \frac{\pi}{N}(2k+1)\right]$$

For this representation of $(AF_u - AF)/u$ show that

$$\varepsilon_i = 2AF\left(\frac{N}{\pi}\right)^3\sum_{k=1}^{L}(-1)^k\frac{\sin(i\pi/2N)(2k+1)}{(2k+1)^2}$$

5. ***Nonuniformly Spaced Arrays and the Equivalent Uniformly Spaced Array [19].*** One method of gaining insight into the gross behavior of a nonuniformly spaced array and making it amenable to a linear analysis is provided by the correspondence between a nonuniformly spaced array and its equivalent uniformly spaced array (EUA). The EUA provides a best mean square representation for the original nonuniformly spaced array.

(a) Define the uniformly spaced array factor $A(\theta)$ as the normalized phasor sum of the responses of each of the sensor elements to a narrowband signal in the uniform array of Figure 2-40, where the normalization is taken with respect to the response of the center

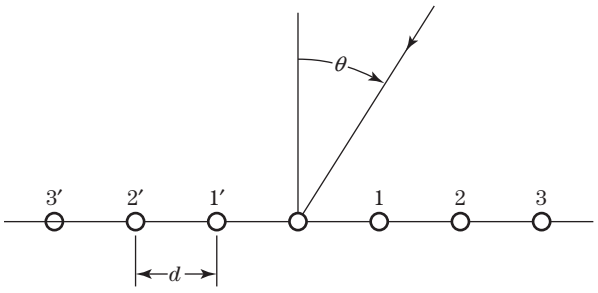


FIGURE 2-40 ■ Uniformly spaced array having an odd number of sensor elements.

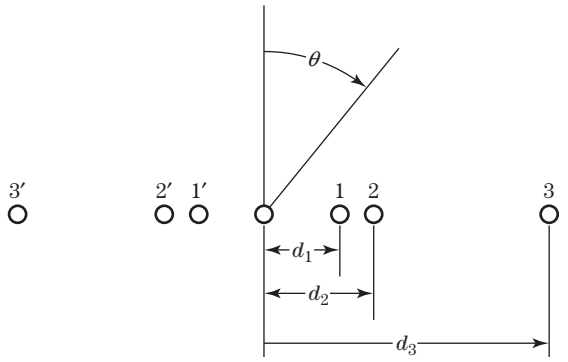


FIGURE 2-41 ■ Symmetrical nonuniformly spaced array having an odd number of sensor elements.

element. Show that

$$AF(\theta) = 1 + 2\sum_{i=1}^{(N-1)/2} \cos[2\pi i l \sin\theta]$$

where $l = d/\lambda$. Furthermore, show that when $d/\lambda = 2$ the array factor has its second principal maxima occurring at $\theta = \pi/6$.

(b) Show that the nonuniformly spaced array factor for the array of Figure 2-41 is given by

$$AF(\theta) = 1 + 2\sum_{i=1}^{(N-1)/2} \cos[2\pi l_i \sin\theta]$$

where $l_i = d_i/\lambda$.

(c) The result obtained in part (b) may be regarded as $AF(\theta) = 1 + 2\Sigma_i \cos\omega_i$ where

$$\omega_i = \left(\frac{2}{R} l_i\right) \underbrace{(\pi R \sin\theta)}_{\omega_1}, \quad R = \text{ arbitrary scale factor}$$

Regarding $\omega_1 = \pi g \sin\theta$ as the lowest nonzero superficial array frequency in the EUA, the scale factor R determines the element spacing in the EUA. For example, $R = 1$ corresponds to half-wavelength spacing, and $R = \frac{1}{2}$ corresponds to quarter-wavelength spacing. Each term in $AF(\theta)$ may be expanded (in a Fourier series representation for each higher harmonic) into an infinite number of uniformly spaced equivalent elements. Since the amplitude of the equivalent elements varies considerably across the array, only a few terms of the expansion need be considered for a reasonable equivalent representation in terms of EUAs. The mth term in $AF(\theta)$ is given by

$$\cos\omega_m = \cos\frac{2}{R} l_m \omega_1 = \cos\mu_m \omega_1$$

where $\mu_m = (2l_m/R)$

The mth term may consequently be expanded in an integral harmonic cosine series as follows:

$$\cos\mu_m\omega_1 = \sum_{v=0}^{\infty} a_{mv}\cos(v\omega_1)$$

where

$$a_{mv} = \frac{2}{\pi} \int_0^{\pi R} \cos \mu_m \omega_1 \cos v\omega_1 d\omega_1$$

Show that when $R_1 = 1$

$$a_{mv} = \frac{2}{\pi} \frac{\mu_m (-1)^v \sin \mu_m \pi}{\mu_m^2 - v^2} \quad \text{for } \mu_m \neq 0, \pm 1, \pm 2, \cdots$$

(d) Applying the harmonic cosine series expansion given in part (c) to each component $\cos \omega_i$, show that the resulting amplitude coefficients may be added term by term to give the EUA representation:

$$AF(\theta) = 1 + 2 \sum_{i=1}^{P} \left\{ \sum_{v=0}^{\infty} a_{mv} \cos(v\omega_1) \right\}$$
$$= 1 + 2 \sum_{r=0}^{P} A_r \cos r\omega_1$$

where

$$AF_0 = a_{10} + a_{20} + a_{30} + \cdots$$
$$AF_1 = a_{11} + a_{21} + a_{31} + \cdots$$

Note that the representation given by the previous result is not unique. Many choices are possible as a function of the scale factor R. From the least mean square error property of the Fourier expansion, each representation will be the best possible for a given value of ω_1.

6. ***Array Propagation Delay Effects***. Array propagation delay effects can be investigated by considering the simple two-element array model of Figure 2-42. It is desired to adjust the value of w_1 so the output signal power P_0 is minimized for a directional interference signal source.

(a) Since $x_2(\omega) = e^{-j\omega\tau} x_1(\omega)$ in the frequency domain, the task of w_1 may be regarded as one of providing the best estimate (in the mean square error sense) of $x_1(\omega)e^{-j\omega\tau}$, and the error in this estimate is given by $\varepsilon(\omega) = x_1(\omega)(e^{-j\omega\tau} - w_1)$. For the weight w_1 to be optimal, the error in the estimate must be orthogonal to the signal $x_1(\omega)$ so it is necessary that

$$E\{|x_1(\omega)|^2 [e^{-j\omega\tau} - w_1]\} = 0$$

where the expectation $E\{\cdot\}$ is taken over frequency. If $|x_1(\omega)|^2$ is a constant independent of frequency, it follows from the previously given orthogonality condition that w_1

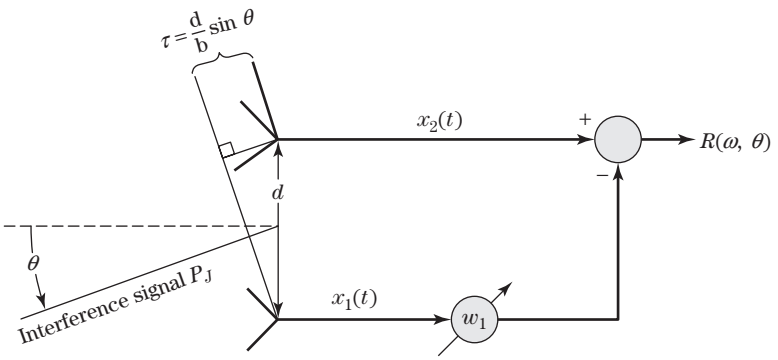


FIGURE 2-42 ■ Two-element linear array having a single complex weight with interelement propagation delay, τ.

must satisfy

$$E\{e^{-j\omega\tau} - w_1\} = 0$$

If the signal has a rectangular spectrum over the bandwidth $-\pi B \leq \omega \leq \pi B$, show that the previous equation results in

$$w_1 = \frac{\sin(\pi B\tau)}{\pi B\tau}$$

(b) The output power may be expressed as

$$P_0 = \int_{-\pi B}^{\pi B} \phi_{RR}(\omega, \theta) d\omega$$

where $\phi_{RR}(\omega, \theta)$ is the output power spectral density. The output power spectral density is in turn given by

$$\phi_{RR}(\omega, \theta) = |e^{-j\omega\tau} - w_1|^2 \phi_{ss}(\omega)$$

where $\phi_{ss}(\omega)$ is the power spectral density of the directional interference signal. Assuming that $\phi_{ss}(\omega)$ is unity over the signal bandwidth (so that $P_J = 2\pi B$), then from part (a) it follows that

$$P_0 = \int_{-\pi B}^{\pi B} \left| e^{-j\omega\tau} - \frac{\sin(\pi B\tau)}{\pi B\tau} \right|^2 dw$$

Show that P_0 can be expressed as

$$\frac{P_0}{P_J} = 1 - \left[\frac{\sin(\pi B\tau)}{\pi B\tau} \right]^2$$

where P_J is the directional interference signal power.

7. ***Open-Loop Scanning of Adaptively Formed Beams [47].*** To scan an adaptively formed beam about the direction of a received signal requires open-loop scanning. Open-loop scanning, however, requires knowledge of the array element locations. An airborne thinned array may contain elements distributed over the entire air frame. Although element coordinates may be measured accurately on the ground, an airframe is flexible; consequently, the relative element positions may vary considerably in flight, with rms deviations of 10^{-1} ft easily occurring.

For open-loop scanning, four theoretical results have been obtained as follows [47]:

1. The rms array element coordinate tolerance is

$$\sigma_x = \sigma_y = \sigma_z = \frac{\lambda}{4\pi\theta_s}$$

where λ is the radiated wavelength, and θ_s is the maximum scan angle from the initial pointing direction of the array (i.e., θ_s is half the field of view).

2. The rms tolerance in the estimate of initial pointing direction is

$$\sigma_\theta \cong \frac{\Delta\theta}{2\theta_s}$$

where $\Delta\theta$ is the beamwidth $\cong \lambda/D$, and D is the linear dimension of the array.

3. The distance beyond which far-field scanning may be employed is given by

$$R_m \approx \frac{\theta_s D^2}{2\lambda}$$

where D is the linear dimension of the array.

4. The rms tolerance in the range estimate of the target is

$$\sigma_{R_0} \cong \frac{R_0 \Delta\theta}{2\theta_s}$$

where $R_0 < R_m$ is the target range.

(a) Assuming $\lambda = \sigma_x = \sigma_y = \sigma_z = 10^{-1}$ ft, what field of view can successfully be scanned with open-loop techniques?

(b) What is the minimum range at which far-field scanning may be used?

(c) For the field of view found in part (a), determine the rms tolerance in the initial pointing error.

8. ***Frequency-Domain Equivalent of Time Delay [45].*** The conventional time-domain processor for beamforming involving sum and delay techniques can be replaced by an equivalent frequency-domain processor using the FFT.

The wavefront of a plane wave arriving at a line of sensors having a spacing d between elements will be delayed in time by an amount τ_1 between adjacent sensors given by

$$\tau_1 = \left(\frac{d}{\mathfrak{v}}\right) \sin\theta_1$$

where θ_1 is the angle between the direction of arrival and the array normal. Consequently, for a separation of n elements

$$\tau_{n1} = n\tau_1$$

In classical time-domain array processing, a spatial beam is formed at an angle θ_1 by introducing appropriate delay lines at the output of each sensor. Let $x_n(t)$ denote the input signal from the plane wave at the nth sensor, and let the time delay present in the nth sensor channel be τ_n. Since the array output is formed by coherent addition of each of the sensor outputs, the output $y(t)$ is given by

$$y(t) = \sum_{n=0}^{N-1} x_n(t - \tau_n)$$

for an array having N sensors

(a) Show the following time–frequency domain relationships based on the Fourier transformation:

Time Domain	Frequency Domain
$x_n(t)$	$X_n(\omega)$
$x_n(t - \tau_n)$	$X_n(\omega)e^{-j\phi_n}$
$\sum_{n=0}^{N-1} x_n(t - \tau_n)$	$\sum_{n=0}^{N-1} X_n(\omega)e^{-j\phi_n}$

Note that ϕ_n represents the phase shift equivalent to the time delay τ_n.

(b) The plane wave arriving at the array with arrival angle θ_1 can be represented by

$$x(t, z) = \cos\left[\omega\left(t - \frac{z}{\mathfrak{v}}\right)\right] = \cos[\omega t - kz]$$

where k is the wavenumber of the plane wave, and z is the axis of propagation at an angle θ_1 with respect to the array normal. Show that the phase shift ϕ_n associated with a time delay τ_n is given by

$$\phi_n = \omega\tau_n$$

where

$$\tau_n = n\left(\frac{d}{\mathfrak{v}}\right)\sin\theta_1$$

(c) It is convenient to index θ and ω to account for different combinations of the angle θ_l and frequency ω so that

$$\theta = \theta_l = l\Delta\theta, \quad l = 0, 1, \ldots, N-1$$
$$\omega = \omega_m = m\Delta\omega, m = 0, 1, \ldots, M-1$$

Furthermore, index the input time samples in i so that

$$t = t_i = i\Delta t, i = 0, 1, \ldots, M-1$$

Let $X_n(\omega)$ be represented at discrete frequencies (a discrete Fourier transform) by

$$X_n(m\Delta\omega) = X_n(\omega_m) = X_{mn}$$

Likewise, the sampled time waveform from the nth array element can be represented by

$$x_n(i\Delta t) = x_n(t_i) = x_{in}$$

The time–frequency domain equivalence between x_{in} and X_{mn} is then given by

Time Domain		Frequency Domain
$x_{in} = x_n(i\Delta t)$	$\leftrightarrow$	$X_n(m\Delta\omega) = X_{mn}$

Consequently, the summed expression from part (a) can be written as

$$\sum_{n=0}^{N-1} X_n(\omega)e^{-j\phi_n} = \sum_{n=0}^{N-1}(m\Delta\omega)e^{-j\phi_n}$$

Show that $\phi_{nl} = nkd\sin\theta_l$. Consequently, show that the previously summed expressions can be rewritten in terms of both frequency and angle as

$$X_m(1\Delta\theta) = \sum_{n=0}^{N-1} X_n(m\Delta\omega)e^{-j\phi_{nl}}$$

(d) The phase shift ϕ_{nl} can be expressed as

$$\phi_{nl} = \frac{nl}{N}$$

implying that $\Delta\mu = 1/\text{Nkd}$ where $\sin(1\Delta\theta) = 1\Delta\mu$, so that

$$X_{ml} = X(1\Delta\theta) = \sum_{n=0}^{N-1} N_n(m\Delta\omega)e^{-jnl/N}$$

which is of the form of a discrete Fourier transform (DFT).

Show that $X_{mn} = X_n(m\Delta\omega)$ can also be expressed as a DFT

$$X_{mn} = \sum_{i=0}^{M-1} x_n(i\Delta t)e^{-jim/N}$$

Now expressing $x_n(i\Delta t)$ as

$$x_n(i\Delta t) = x_{ni}$$

a double DFT can now be written as

$$X_{ml} = \sum_{n=0}^{N-1}\sum_{i=0}^{M-1} x_{ni}e^{-jim/N}e^{-jnl/N}$$

The array output has therefore been obtained in the form of spectral sample versus beam number, which results after a two-dimensional DFT transformation of array input. The required DFTs can then be implemented using the FFT algorithm.

9. ***Linear Array Directivity.*** Calculate the directivity of a 10-element uniform array with $d = 0.5\lambda$, and compare it with a 10-element uniform array with $d = 1.0\lambda$. Why are they different?

10. ***Moving Nulls on the Unit Circle.*** Plot the unit circle representations of a four-element uniform array. Now, move the null locations from $\psi = \pm 90^\circ$ to $\psi = \pm 120^\circ$.

11. ***Chebyshev Array.*** Plot the unit circle representation, the amplitude weights, and the array factor for a six-element 20 dB Chebyshev array.

12. ***Taylor Array.*** Plot the unit circle representation, the amplitude weights, and the array factor for a 20-element 20 dB $\bar{n} = 5$ Taylor array.

13. ***Taylor Array.*** Start with a Taylor $\bar{n} = 4$ sll = 25dB taper, and place a null at u = 0.25 when d = 0.5λ. Do not allow complex weights.

14. ***Thinned Array.*** A 5,000-element linear array has a taper efficiency of 50% and elements spaced a half wavelength apart. What are the average and peak sidelobe levels of this array?

15. ***Thinned Array with Taylor Taper.*** Start with a 100-element array with element spacing $d = 0.5$, and thin to a 20 dB $\bar{n} = 5$ Taylor taper.

16. ***Cancellation Beams.*** A 40-element array of isotropic point sources spaced $\lambda/2$ apart has a 30 dB $\bar{n} = 7$ Taylor taper. Plot the array factors and cancellation beam for a null at 61°.

17. ***Cancellation Beams.*** Repeat the previous example using both uniform and weighted cancellation beams.

18. ***Cancellation Beams.*** A 40-element array of isotropic point sources spaced $\lambda/2$ apart has a 40 dB $\bar{n} = 7$ Taylor taper. Plot the array factors and cancellation beams for nulls at 13° and 61°. Use phase-only nulling.

19. ***Simultaneous Nulling in the Sum and Difference Patterns.*** A 40-element array of isotropic point sources spaced $\lambda/2$ apart has a 30 dB $\bar{n} = 7$ Taylor taper and a 30 dB $\bar{n} = 7$ Bayliss taper. Plot the array factors and cancellation beams for nulls at 13° and 61°. Use simultaneous phase-only nulling in the sum and difference patterns.

20. ***Triangular Spacing in a Planar Array [11].*** Show that the uniform array factor for triangular spacing is

$$AF = \frac{\sin\left(N_{xe}\psi_x\right)}{N_{xe}\sin\left(\psi_x\right)}\frac{\sin\left(N_{ye}\psi_y\right)}{N_{ye}\sin\left(\psi_y\right)} + e^{-j\left(\psi_x+\psi_y\right)}\frac{\sin\left(N_{xo}\psi_x\right)}{N_{xo}\sin\left(\psi_x\right)}\frac{\sin\left(N_{yo}\psi_y\right)}{N_{yo}\sin\left(\psi_y\right)}$$

2.9 REFERENCES

[1] P. W. Howells, "Explorations in Fixed and Adaptive Resolution at GE and SURC," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 575–584.

[2] R. L. Deavenport, "The Influence of the Ocean Environment on Sonar System Design," *IEEE 1975 EASCON Record, Electronics and Aerospace Systems Convention,* September 29–October 1, pp. 66.A–66.E.

[3] A. A. Winder, "Underwater Sound—A Review: II. Sonar System Technology," *IEEE Trans. Sonics and Ultrason.*, Vol. SU-22, No. 5, September 1975, pp. 291–332.

[4] L. W. Nolte and W. S. Hodgkiss, "Directivity or Adaptivity?" *IEEE 1975 EASCON Record, Electronics and Aerospace Systems Convention,* September 29–October 1, pp. 35.A–35.H.

[5] G. M. Wenz, "Review of Underwater Acoustics Research: Noise," *J. Acoust. Soc. Am.*, Vol. 51, No. 3 (Part 2), March 1972, pp. 1010–1024.

[6] B. W. Lindgren, *Statistical Theory*, New York, MacMillan, 1962, Ch. 4.

[7] F. Bryn, "Optimum Structures of Sonar Systems Employing Spatially Distributed Receiving Elements," in *NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics,* Vol. 2, Paper No. 30, Enschede, The Netherlands, August 1968.

[8] H. Cox, "Optimum Arrays and the Schwartz Inequality," *J. Acoust. Soc. Am.*, Vol. 45, No. 1, January 1969, pp. 228–232.

[9] R. S. Elliott, "The Theory of Antenna Arrays," in *Microwave Scanning Antennas, Vol. 2, Array Theory and Practice*, edited by R. C. Hansen, New York, Academic Press, 1966, Ch. 1.

[10] I. J. Gupta and A. A. Ksienski, "Effect of Mutual Coupling on the Performance of Adaptive Arrays," *IEEE AP-S Trans.*, Vol. 31, No. 5, September 1983, pp. 785–791.

[11] R. L. Haupt, *Antenna Arrays: A Computational Approach*, New York, Wiley, 2010.

[12] M.I. Skolnik, *Introduction to Radar Systems*, New York, McGraw-Hill, 2002.

[13] H. T. Friis, "A Note on a Simple Transmission Formula," *IRE Proc.*, May 1946, pp. 254–256.

[14] J. S. Stone, United States Patent No. 1,643,323 and 1,715,433.

[15] C. L. Dolph, "A Current Distribution for Broadside Arrays which Optimizes the Relationship Between Beam Width and Side-Lobe Level," *Proceedings of the IRE,* Vol. 34, No. 6, June 1946, pp. 335–348.

[16] T. T. Taylor, "Design of Line Source Antennas for Narrow Beamwidth and Low Side Lobes," *IRE AP Trans.,* Vol. 3, 1955, pp. 16–28.

[17] T. Taylor, "Design of Circular Apertures for Narrow Beamwidth and Low Sidelobes," *IEEE AP-S Trans.*, Vol. 8, No. 1, January 1960, pp. 17–22.

[18] F. Hodjat and S. A. Hovanessian, "Nonuniformly Spaced Linear and Planar Array Antennas for Sidelobe Reduction," *IEEE Trans. Antennas Propag.*, Vol. AP-26, No. 2, March 1978, pp. 198–204.

[19] S. S. Sandler, "Some Equivalences Between Equally and Unequally Spaced Arrays," *IRE Trans. Antennas and Progag.*, Vol. AP-8, No. 5, September 1960, pp. 496–500.

[20] R. F. Harrington, "Sidelobe Reduction by Nonuniform Element Spacing," *IRE Trans. Antennas Propag.*, Vol. AP-9, No. 2, March 1961, pp. 187–192.

[21] D. D. King, R. F. Packard, and R. K. Thomas, "Unequally Spaced, Broadband Antenna Arrays," *IRE Trans. Antennas Propag.*, Vol. AP-8, No. 4, July 1960, pp. 380–385.

[22] M. I. Skolnik, "Nonuniform Arrays," in *Antenna Theory, Part I*, edited by R. E. Collin and F. J. Zucker, New York, McGraw-Hill, 1969, Ch. 6.

[23] Y. T. Lo, "A Mathematical Theory of Antenna Arrays with Randomly Spaced Elements," *IEEE Trans. Antennas Propag.*, Vol. AP-12, May 1964, pp. 257–268.

[24] Y. T. Lo and R. J. Simcoe, "An Experiment on Antenna Arrays with Randomly Spaced Elements," *IEEE Trans. Antennas Propag.*, Vol. AP-15, March 1967, pp. 231–235.

[25] E. Brookner, "Antenna Array Fundamentals—Part 1," in *Practical Phased Array Antenna Systems*, Norwood, MA, Artech House, 1991, pp. 2-1–2-37.

[26] R.L. Haupt, "Thinned Arrays Using Genetic Algorithms," *IEEE AP-S Transactions,* Vol. 42, No. 7, July 1994, pp. 993–999.

[27] R. L. Haupt and D. Werner, *Genetic Algorithms in Electromagnetics,* New York, Wiley, 2007.

[28] R. L. Haupt, "Interleaved Thinned Linear Arrays," *IEEE AP-S Trans.*, Vol. 53, September 2005, pp. 2858–2864.

[29] R. Guinvarc'h and R. L. Haupt, "Dual Polarization Interleaved Spiral Antenna Phased Array with an Octave Bandwidth" *IEEE Trans. Antennas Propagat.*, Vol. 58, No. 2, 2010, pp. 397–403.

[30] R. L. Haupt, "Optimized Element Spacing for Low Sidelobe Concentric Ring Arrays," *IEEE AP-S Trans.*, Vol. 56, No. 1, January 2008, pp. 266–268.

[31] H. Steyskal, "Synthesis of Antenna Patterns with Prescribed Nulls," *IEEE Transactions on Antennas and Propagation,* Vol. 30, no. 2, 1982, pp. 273–279.

[32] H. Steyskal, R. Shore, and R. Haupt, "Methods for Null Control and Their Effects on the Radiation Pattern," *IEEE Transactions on Antennas and Propagation,* Vol. 34, No. 3, 1986, pp. 404–409.

[33] H. Steyskal, "Simple Method for Pattern Nulling by Phase Perturbation," *IEEE Transactions on Antennas and Propagation,* Vol. 31, No. 1, 1983, pp. 163–166.

[34] R. Shore, "Nulling a Symmetric Pattern Location with Phase-Only Weight Control," *IEEE Transactions on Antennas and Propagation,* Vol. 32, No. 5, 1984, pp. 530–533.

[35] R. Haupt, "Simultaneous Nulling in the Sum and Difference Patterns of a Monopulse Antenna," *IEEE Transactions on Antennas and Propagation,* Vol. 32, No. 5, 1984, pp. 486–493.

[36] B. Widrow and M. E. Hoff, Jr., "Adaptive Switching Circuits," *IRE 1960 WESCON Convention Record,* Part 4, pp. 96–104.

[37] J. H. Chang and F. B. Tuteur, "A New Class of Adaptive Array Processors," *J. Acoust. Soc. Am.*, Vol. 49, No. 3, March 1971, pp. 639–649.

[38] L. E. Brennan, J. D. Mallet, and I. S. Reed, "Adaptive Arrays in Airborne MTI Radar," *IEEE Trans. Antennas Propag.*, Special Issue Adaptive Antennas, Vol. AP-24, No. 5, September 1976, pp. 607–615.

[39] R. T. Compton, Jr., "The Bandwidth Performance of a Two-Element Adaptive Array with Tapped Delay-Line Processing," *IEEE Trans. Ant. & Prop.,* Vol. AP-36, No.1, Jan. 1988, pp. 5–14.

[40] R. T. Compton, Jr., "The Relationship Between Tapped Delay-Line and FFT Processing in Adaptive Arrays," *IEEE Trans. Ant. & Prop.,* Vol. AP-36, No. 1, January 1988, pp. 15–26.

[41] F. W. Vook and R. T. Compton, Jr., "Bandwidth Performance of Linear Adaptive Arrays with Tapped Delay-Line Processing," *IEEE Trans. Aerosp. & Electron. Sys.,* Vol. AES-28, No. 3, July 1992, pp. 901–908.

[42] P. Rudnick, "Digital Beamforming in the Frequency Domain," *J. Acoust. Soc. Am.*, Vol. 46, No. 5, 1969.

[43] D. T. Deihl, "Digital Bandpass Beamforming with the Discrete Fourier Transform," Naval Research Laboratory Report 7359, AD-737-191, 1972.

[44] I. M. Weiss, "GPS Adaptive Antenna/Electronics—Proposed System Characterization Performance Measures," Aerospace Corp. Technical Report No. TOR-95 (544)-1, 20 December 1994.

[45] L. Armijo, K. W. Daniel, and W. M. Labuda, "Applications of the FFT to Antenna Array Beamforming," *EASCON 1974 Record, IEEE Electronics and Aerospace Systems Convention,* October 7–9, Washington, DC, pp. 381–383.

[46] A. M. Vural, "A Comparative Performance Study of Adaptive Array Processors," *Proceedings of the 1977 International Conference on Acoustics, Speech, and Signal Processing,* May, pp. 695–700.

[47] S. H. Taheri and B. D. Steinberg, "Tolerances in Self-Cohering Antenna Arrays of Arbitrary Geometry," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 733–739.

CHAPTER 3

Optimum Array Processing: Steady-State Performance Limits and the Wiener Solution

Chapter Outline

Optimum array processing is an optimum multichannel filtering problem [1–12]. The objective of array processing is to enhance the reception (or detection) of a desired signal that may be either random or deterministic in a signal environment containing numerous interference signals. The desired signal may also contain one or several uncertain parameters (e.g., spatial location, signal energy, phase) that it may be advantageous to estimate.

Optimum array processing techniques are broadly classified as (1) processing appropriate for ideal propagation conditions and (2) processing appropriate for perturbed propagation conditions. Ideal propagation implies an ideal nonrandom, nondispersive medium where the desired signal is a plane (or spherical) wave and the receiving sensors are distortionless. In this case the optimum processor is said to be matched to a plane wave signal. Any performance degradation resulting from deviation of the actual operating conditions from the assumed ideal conditions is minimized by the use of complementary methods, such as the introduction of constraints. When operating under the aforementioned ideal conditions, vector weighting of the input data succeeds in matching the desired signal.

When perturbations in either the propagating medium or the receiving mechanism occur, the plane wave signal assumption no longer holds, and vector weighting the input data will not match the desired signal. Matrix weighting of the input data is necessary [13] for a signal of arbitrary characteristics. The principal advantage of such an element space-matched array processor operates on noncoherent wavefront signals where matching can be performed in only a statistical sense.

Steady-state performance limits establish a performance measure for any selected design. Quite naturally, optimum array processing has mostly assumed ideal propagation

conditions, and various approaches to this problem have been proposed for both narrowband and broadband signal applications. By far, the most popular and widely reported approaches involve the adoption of an array performance measure that is optimized by appropriate selection of an optimum weight vector. Such performance measure optimization approaches have been widely used for radar, communication, and sonar systems and therefore are discussed at some length. Approaches to array processing that use the maximum entropy method (which is particularly applicable to the passive sonar problem) [14,15] and eigenvalue resolution techniques [16,17] have been reported.

To determine the optimal array weighting and its associated performance limits, some mathematical preliminaries are first discussed, and the signal descriptions for conventional and signal-aligned arrays are introduced. It is well known that when all elements in an array are uniformly weighted, then the maximum signal-to-noise ratio (SNR) is obtained if the noise contributions from the various element channels have equal power and are uncorrelated [18]. When there is any directional interference, however, the noise from the various element channels is correlated. Consequently, selecting an optimum set of weights involves attempting to cancel correlated noise components. Signal environment descriptions in terms of correlation matrices therefore play a fundamental role in determining the optimum solution for the complex weight vector.

The problem of formulating some popular array performance measures in terms of complex envelope signal characterizations is discussed. It is a remarkable fact that the different performance measures considered here all converge (to within a constant scalar factor) toward the same steady-state solution: the optimum Wiener solution. Passive detection systems face the problem of designing an array processor for optimum detection performance. Classical statistical detection theory yields an array processor based on a likelihood ratio test that leads to a canonical structure for the array processor. This canonical structure contains a weighting network that is closely related to the steady-state weighting solutions found for the selected array performance measures. It therefore turns out that what at first appear to be quite different optimization problems actually have a unified basis.

3.1 MATHEMATICAL PRELIMINARIES

Our derivation of the optimal weight settings represents the signal envelopes as well as the adaptive weights in their complex envelope form. The meaning of such complex representations therefore is briefly reviewed.

From Appendix B, the relationship between real signals and their corresponding complex envelope representations is

$$\text{actual signal} = \text{Re}\{(\text{complex envelope representation})e^{j\omega_0 t}\} \tag{3.1}$$

where Re{ } denotes "real part of." This result is further discussed in Appendix B. Applying this notation to the array output yields

$$\text{actual } y(t) = \text{Re}\{\mathbf{x}^T\mathbf{w}^* e^{j\omega_0 t}\} = \text{Re}\{\mathbf{w}^\dagger\mathbf{x}e^{j\omega_0 t}\} \tag{3.2}$$

where $\mathbf{w}^\dagger\mathbf{x}$ is the complex envelope representation of $y(t)$, * denotes complex conjugate, and † denotes complex conjugate transpose $[(\ )^*]^T$. (Boldface lowercase symbols denote vectors; boldface uppercase symbols denote matrices.) A closely related complex

representation is the analytic signal $\psi(t)$ for which $y(t) \triangleq \text{Re}\{\psi(t)\}$ and

$$\psi(t) \triangleq y(t) + j\check{y}(t) \tag{3.3}$$

where $\check{y}(t)$ denotes the Hilbert transform of $y(t)$. It follows that for a complex weight representation $w = w_1 + jw_2$ having an input signal with analytic representation $x_1 + jx_2$, the resulting actual output signal is given by

$$\text{actual ouput} = \text{Re}\{(w_1 - jw_2)(x_1(t) + jx_2(t))\} = w_1x_1(t) + w_2x_2(t) \tag{3.4}$$

where $x_2(t) = \check{x}_1(t)$. Note that (complex envelope) $e^{j\omega_0 t} = \psi(t)$. As a consequence of the previously given meaning of complex signal and weight representations, two alternate approaches to the optimization problems for the optimal weight solutions may be taken as follows:

1. Reformulate the problem involving complex quantities in terms of completely real quantities so that familiar mathematical concepts and operations are conveniently carried out.
2. Revise the definitions of certain concepts and operations (e.g., covariance matrices, gradients) so that all complex quantities are handled directly in an appropriate manner.

Numerous examples of both approaches are found in the adaptive array literature. It is therefore appropriate to consider how to use both approaches since there are no compelling advantages favoring one approach over the other.

3.1.1 Problem Formulation in Terms of Real Variables

Let z_1 be the first component of a complex vector $\mathbf{z}$ having n components; $\mathbf{z}$ may represent, for example, a weight or signal vector. Furthermore, let the real part of z_1 be denoted by x_1 and the imaginary part of z_1 be denoted by x_2. Continuing in this manner, we have

$$z_k = x_{2k-1} + jx_{2k} \tag{3.5}$$

It follows that the n-component complex vector $\mathbf{z}$ is completely represented by the $2n$-component real vector $\mathbf{x}$. By representing all complex quantities in terms of corresponding real vectors, the adaptive array problem is solved using familiar definitions of mathematical concepts and procedures. If we carry out the foregoing procedure for the single complex weight $w_1 + jw_2$ and the complex signal $x_1 + jx_2$, there results

$$\mathbf{w}^T\mathbf{x} = [w_1 w_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = w_1x_1 + w_2x_2 \tag{3.6}$$

which is in agreement with the result expressed by (3.4). If the foregoing approach is adopted, then certain off-diagonal elements of the corresponding correlation matrix (defined in the following section) will be zero. This fact does not affect the results obtained in this chapter; the point is discussed more fully in connection with an example presented in Chapter 11.

3.1.2 Correlation Matrices for Real Signals

Once the signal and noise processes in an adaptive array system have been described in terms of statistical properties, the system performance is conveniently evaluated in terms of its average behavior. The evaluation of average behavior leads directly to an interest in quantities related to the second statistical moment such as autocorrelation and cross-correlation matrices. The degree of correlation that exists between the components of two random vectors is given by the elements of the correlation matrix between the two vectors [19]. For example, the cross-correlation matrix between the vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ having stationary statistical properties is defined by

$$\mathbf{R}_{xy}(\tau) \triangleq E\{\mathbf{x}(t)\mathbf{y}^T(t-\tau)\} \tag{3.7}$$

where $E\{\ \}$ denotes the expected value, and τ is a running time-delay variable. Likewise, the autocorrelation matrix for the vector $\mathbf{x}(t)$ is defined by

$$\mathbf{R}_{xx}(\tau) \triangleq E\{\mathbf{x}(t)\mathbf{x}^T(t-\tau)\} \tag{3.8}$$

If a signal vector $\mathbf{x}(t)$ consists of uncorrelated desired signals and noise components so that

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t) \tag{3.9}$$

then

$$\mathbf{R}_{xx}(\tau) = \mathbf{R}_{ss}(\tau) + \mathbf{R}_{nn}(\tau) \tag{3.10}$$

The correlation matrices of principal concern in the adaptive array analysis are those for which the time-delay variable τ is zero. Rather than write the correlation matrix argument explicitly as $\mathbf{R}_{xy}(0)$ and $\mathbf{R}_{xx}(0)$, we define

$$\mathbf{R}_{xx} = \mathbf{R}_{xx}(0) \tag{3.11}$$

and

$$\mathbf{R}_{xy} = \mathbf{R}_{xy}(0) \tag{3.12}$$

It follows that for an N-dimensional vector $\mathbf{x}(t)$, the autocorrelation matrix is simply written as $\mathbf{R}_{xx}$ where

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}\mathbf{x}^T\} = \begin{bmatrix} \overline{x_1(t)x_1(t)} & \overline{x_1(t)x_2(t)} & \dots & \overline{x_1(t)x_N(t)} \\ \overline{x_2(t)x_1(t)} & \overline{x_2(t)x_2(t)} & \dots & \\ \vdots & & & \\ \overline{x_N(t)x_1(t)} & \dots & & \overline{x_N(t)x_N(t)} \end{bmatrix} \tag{3.13}$$

where $\overline{x_i(t)x_k(t)}$ denotes $E\{x_i(t)x_k(t)\}$.

From the previous definitions, it immediately follows that

$$\mathbf{R}_{xx}^T = \mathbf{R}_{xx} \quad \text{and} \quad \mathbf{R}_{xy}^T = \mathbf{R}_{yx} \tag{3.14}$$

so the autocorrelation matrix is symmetric. In general, the autocorrelation matrix $\mathbf{R}_{xx}$ is only positive semidefinite, and the inverse matrix $\mathbf{R}_{xx}^{-1}$ may not exist. Since in practice only estimates of the signal environment correlation matrices are available and such estimates are based on distinct time samples, then a sufficient number of such time samples for a nonzero bandwidth signal process may guarantee positive definiteness so the inverse matrix will exist. The signal vectors $\mathbf{x}(t)$ and $\mathbf{n}(t)$ of (3.9), for example, often contain uncorrelated amplifier self-noise and therefore may be regarded as having positive definite correlation matrices. The desired signal vector $\mathbf{s}(t)$ may, however, contain correlated components, as when a uniform plane wave arrives simultaneously at two or more array sensor elements, thereby producing identical, in-phase signal components. Consequently, the autocorrelation matrix $\mathbf{R}_{ss}$ is only positive semidefinite, and its inverse $\mathbf{R}_{ss}^{-1}$ may not exist.

Now consider the correlation matrix that results for the tapped delay-line processor of Section 2.5 by examining the $(N \times N)$-dimensional autocorrelation matrix given by

$$\mathbf{R}'_{xx}(\tau) \triangleq E\{\mathbf{x}'(t)\mathbf{x}'^{T}(t-\tau)\} \tag{3.15}$$

where $\mathbf{x}'(t)$ is the (real) signal vector defined in (2.80) for a tapped delay-line multichannel processor. Likewise, for the NL-dimensional vector of all signals observed at the tap points, the autocorrelation matrix is given by

$$\mathbf{R}_{xx}(\tau) \triangleq E\{\mathbf{x}(t)\mathbf{x}^{T}(t-\tau)\} \tag{3.16}$$

Substituting (2.82) for $\mathbf{x}(t)$ into (3.16) then yields

$$\mathbf{R}_{xx}(\tau) = E\left\{ \begin{bmatrix} \mathbf{x}'(t) \\ \mathbf{x}'(t-\Delta) \\ \vdots \\ \mathbf{x}'[t-(L-1)\Delta] \end{bmatrix} \left[\mathbf{x}'^{T}(t-\tau)\mathbf{x}'^{T}(t-\tau-\Delta)\cdots\mathbf{x}'^{T}[t-\tau-(L-1)\Delta]\right] \right\} \tag{3.17}$$

and, using (3.15), $R_{xx}(\tau)$ then becomes

$$\mathbf{R}_{xx}(\tau) = \begin{bmatrix} \mathbf{R}'_{xx}(\tau) & \mathbf{R}'_{xx}(\tau+\Delta) & \cdots & \mathbf{R}'_{xx}[\tau+(L-1)\Delta] \\ \mathbf{R}'^{T}_{xx}(\tau-\Delta) & \mathbf{R}'_{xx}(\tau) & & \vdots \\ \vdots & & & \vdots \\ \mathbf{R}'^{T}_{xx}[\tau-(L-1)\Delta] & \cdots & & \mathbf{R}'_{xx}(\tau) \end{bmatrix} \tag{3.18}$$

The $(NL \times NL)$-dimensional matrix $\mathbf{R}_{xx}(\tau)$ given by (3.18) has the form of a Toeplitz matrix—a matrix having equal valued matrix elements along any diagonal [20]. The desirability of the Toeplitz form lies in the fact that the entire matrix is constructed from the first row of submatrices; that is, $\mathbf{R}'_{xx}(\tau)$, $\mathbf{R}'_{xx}(\tau+\Delta), \ldots, \mathbf{R}'_{xx}[\tau+(L-1)\Delta]$. Consequently, only an $(N \times NL)$-dimensional matrix need be stored to have all the information contained in $\mathbf{R}_{xx}(\tau)$.

Covariance matrices are closely related to correlation matrices since the covariance matrix between the vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ is defined by

$$\text{cov}[\mathbf{x}(t), \mathbf{y}(t)] \triangleq E\{(\mathbf{x}(t)-\overline{\mathbf{x}})(\mathbf{y}(t)-\overline{\mathbf{y}})^{T}\} \tag{3.19}$$

where

$$\overline{\mathbf{x}} = E\{\mathbf{x}(t)\} \quad \text{and} \quad \overline{\mathbf{y}} = E\{\mathbf{y}(t)\}$$

Thus, for zero-mean processes and with $\tau = 0$, correlation matrices and covariance matrices are identical, and the adaptive array literature frequently uses the two terms interchangeably.

Frequency-domain signal descriptions as well as time-domain signal descriptions are valuable in considering array processors for broadband applications. The frequency-domain equivalent of time-domain descriptions may be found by taking the Fourier transform of the time domain quantities, $\mathfrak{f}(\omega) = \mathfrak{F}\{f(t)\}$. The Fourier transform of a signal correlation matrix yields the signal cross-spectral density matrix

$$\mathbf{\Phi}_{xx}(\omega) = \mathfrak{F}\{\mathbf{R}_{xx}(\tau)\} \tag{3.20}$$

Cross-spectral density matrices therefore present the signal information contained in correlation matrices in the frequency domain.

3.1.3 Revised Definitions Required for Complex Vector Quantities

The optimization problems that must be solved to yield the most desirable complex weight vector choices for different performance measures involve the use of norms, gradients, and covariance (or correlation) matrices of complex vector quantities. It is therefore useful to consider the definitions of these terms for both real and complex vectors.

The norm of a vector in Hilbert space, denoted by $||\mathbf{x}||$, represents the length of the vector. For a real vector

$$||\mathbf{x}|| \triangleq \sqrt{\mathbf{x}^T\mathbf{x}} \tag{3.21}$$

whereas for a complex vector

$$||\mathbf{x}|| \triangleq \sqrt{\mathbf{x}^\dagger\mathbf{x}} \tag{3.22}$$

The gradient of a scalar function, ∇_y, consists of the partial derivatives of $f(\cdot)$ along each component direction of $\mathbf{y}$. In the case of real variables, the gradient operator is a vector operator given by

$$\nabla_y \triangleq \left[\frac{\partial}{\partial y_1} \cdots \frac{\partial}{\partial y_n}\right]^T \tag{3.23}$$

so that

$$\nabla_y f(\mathbf{y}) = \frac{\partial f}{\partial y_1}\mathbf{e}_1 + \frac{\partial f}{\partial y_2}\mathbf{e}_2 + \cdots + \frac{\partial f}{\partial y_n}\mathbf{e}_n \tag{3.24}$$

where $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ is the set of unit basis vectors for the vector $\mathbf{y}$. For a complex vector $\mathbf{y}$ each element y_k has a real and an imaginary component:

$$y_k = x_k + jz_k \tag{3.25}$$

Therefore, each partial derivative in (3.24) now has a real and an imaginary component so that [21]

$$\frac{\partial f}{\partial y_k} = \frac{\partial f}{\partial x_k} + \frac{(j)\partial f}{\partial z_k} \tag{3.26}$$

In the optimization problems encountered in this chapter, it is frequently desired to obtain the gradient with respect to the vector $\mathbf{x}$ of the scalar quantity $\mathbf{x}^T\mathbf{a} = \mathbf{a}^T\mathbf{x}$ and of the quadratic form $\mathbf{x}^T\mathbf{A}\mathbf{x}$ (which is also a scalar), where $\mathbf{A}$ is a symmetric matrix. Note that $\mathbf{x}^T\mathbf{a}$ is an inner product

$$(\mathbf{x}, \mathbf{a}) = \mathbf{x}^T\mathbf{a} = \mathbf{a}^T\mathbf{x} \tag{3.27}$$

If $\mathbf{A}$ has a dyadic structure, the quadratic form $\mathbf{x}^T\mathbf{A}\mathbf{x}$ is regarded as an inner product squared, or

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = (\mathbf{x}, \mathbf{v})^2 \tag{3.28}$$

where

$$\mathbf{A} = \mathbf{v}\mathbf{v}^T \tag{3.29}$$

The trace of a matrix product given by

$$\text{trace}[\mathbf{A}\mathbf{B}^T] = \sum_i \sum_k a_{ik} b_{ik} \tag{3.30}$$

has all the properties of an inner product, so formulas for the differentiation of the trace of various matrix products are of interest in obtaining solutions to optimization problems. A partial list of convenient differentiation formulas is given in Appendix C. From these formulas it follows for real variables that

$$_x(\mathbf{y}^T\mathbf{A}\mathbf{x}) = \mathbf{y}^T\mathbf{A} \tag{3.31}$$

and

$$_x(\mathbf{x}^T\mathbf{A}\mathbf{x}) = 2\mathbf{A}\mathbf{x} \tag{3.32}$$

whereas for complex variables the corresponding gradients are given by

$$_x(\mathbf{y}^\dagger\mathbf{A}\mathbf{x}) = \mathbf{y}^\dagger\mathbf{A} \tag{3.33}$$

and

$$_x(\mathbf{x}^\dagger\mathbf{A}\mathbf{x}) = 2\mathbf{A}\mathbf{x} \tag{3.34}$$

3.1.4 Correlation Matrices for Complex Signals

For complex vector quantities, the corresponding correlation matrix definitions must be revised from (3.7) and (3.8) so that

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^*\mathbf{x}^T\} \qquad \text{and} \qquad \mathbf{R}_{xy} \triangleq E\{\mathbf{x}^*\mathbf{y}^T\} \tag{3.35}$$

An alternative correlation matrix definition also found in the literature is given by

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}\mathbf{x}^{\dagger}\} \qquad \text{and} \qquad \mathbf{R}_{xy} \triangleq E\{\mathbf{x}\mathbf{y}^{\dagger}\} \tag{3.36}$$

The definition of (3.36) yields a matrix that is the complex conjugate (or the transpose) of the definition given by (3.35). So long as one adheres consistently to either one definition or the other, the results obtained (in terms of the selected performance measure) will turn out to be the same; therefore, it is immaterial which definition is used. With either of the aforementioned definitions, it immediately follows that

$$\mathbf{R}_{xx}^{\dagger} = \mathbf{R}_{xx} \qquad \text{and} \qquad \mathbf{R}_{xy}^{\dagger} = \mathbf{R}_{yx} \tag{3.37}$$

So the autocorrelation matrix $\mathbf{R}_{xx}$ is Hermitian, whereas the cross-correlation matrix $\mathbf{R}_{xy}$ is in general not Hermitian (since $\mathbf{R}_{yx}^{\dagger} \neq \mathbf{R}_{xy}$). Whether the autocorrelation matrix is positive definite or positive semidefinite once again is determined by whether the signal vector of concern is $\mathbf{x}(t)$, $\mathbf{s}(t)$, or $\mathbf{n}(t)$.

3.2 | SIGNAL DESCRIPTIONS FOR CONVENTIONAL AND SIGNAL ALIGNED ARRAYS

Two types of arrays have proven useful in different applications: (1) the conventional array illustrated in Figure 3-1; and (2) the signal aligned array illustrated in Figure 3-2. The signal aligned array is useful where the direction of arrival of the desired signal is known a priori,

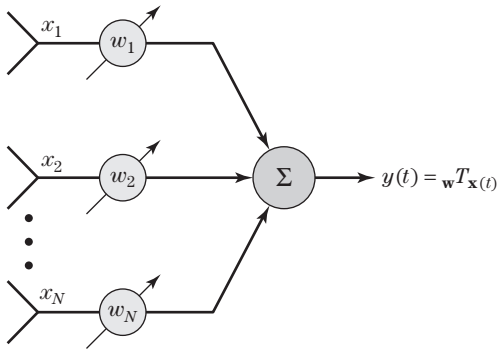


FIGURE 3-1 ■ Conventional narrowband array.

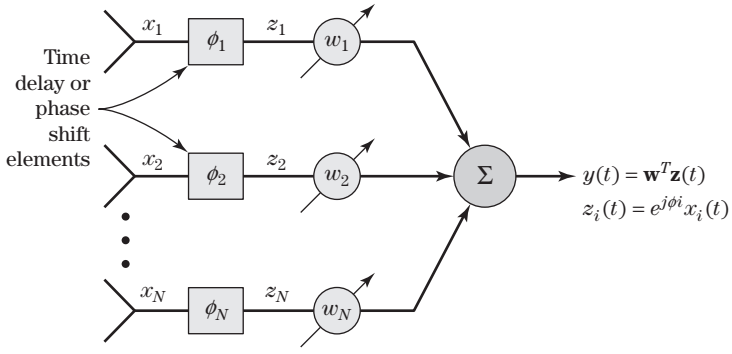


FIGURE 3-2 ■ Signal aligned narrowband array.

and this information is used to obtain time-coincident desired signals in each channel. One advantage of the signal aligned array structure is that a set of weights is found that is independent of the desired signal time structure, which provides a distortionless output for any waveform incident on the array from the (assumed) known desired signal direction [22]. Such a processor is useful for extracting impulse "bursts," which are present only during relatively short time intervals.

The outputs of the arrays illustrated in Figures 3-1 and 3-2 are expressed, respectively, as

$$y(t) = \mathbf{w}^{\dagger}\mathbf{x}(t) \qquad \text{and} \qquad y(t) = \mathbf{w}^{\dagger}\mathbf{z}(t) \tag{3.38}$$

where $\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t)$ is the vector of received signals that are complex valued functions. The noise vector $\mathbf{n}(t)$ may be assumed to be stationary and ergodic and to have both directional and thermal noise components that are independent of the signal. The signal vector $\mathbf{s}(t)$ induced at the sensor elements from a single directional signal source is assumed to be

$$s(t) = \sqrt{S}\, e^{j\omega_0 t} \tag{3.39}$$

where ω_0 is the (radian) carrier frequency, and S represents the signal power. Assuming identical antenna elements, the resulting signal component in each array element is just a phase-shifted version (due to propagation along the array) of the signal appearing at the first array element encountered by the directional signal source. It therefore follows that the signal vector $\mathbf{s}(t)$ is written as

$$\begin{aligned} \mathbf{s}^T(t) &= \left[\sqrt{S}\, e^{j\omega_0 t}, \sqrt{S}\, e^{j\omega_0 t+\theta_1}, \ldots, \sqrt{S}\, e^{j\omega_0 t+\theta_{N-1}}\right] \\ &= s(t)\mathbf{v}^T \end{aligned} \tag{3.40}$$

where $\mathbf{v}$ is defined to be the array propagation vector

$$\mathbf{v}^T = [1, e^{j\theta_1}, \ldots, e^{j\theta_{N-1}}] \tag{3.41}$$

Consequently, the received signal vector for the conventional array of Figure 3-1 is written as

$$\mathbf{x}(t) = s(t)\mathbf{v} + \mathbf{n}(t) \tag{3.42}$$

The received signal vector (after the time-delay or phase-shift elements) for the signal aligned array of Figure 3-2 is written as

$$\mathbf{z}(t) = s(t)\mathbf{1} + \mathbf{n}'(t) \tag{3.43}$$

where now $\mathbf{v}$ of (3.42) has been replaced by $\mathbf{1} = (1, 1, \ldots, 1)^T$ since the desired signal terms in each channel are time aligned and therefore identical. The components of the noise vector then become

$$n_i'(t) = n_i(t)e^{j\phi_i} \tag{3.44}$$

In developing the optimal solutions for selected performance measures, four correlation matrices will be required. These correlation matrices are defined as follows for

narrowband uncorrelated signal processes:

$$\mathbf{R}_{ss} \triangleq E\{\mathbf{s}^*(t)\mathbf{s}^T(t)\} = S\mathbf{v}^*\mathbf{v}^T \tag{3.45}$$

where S denotes the signal power

$$\mathbf{R}_{nn} \triangleq E\{\mathbf{n}^*(t)\mathbf{n}^T(t)\} \tag{3.46}$$

$$\mathbf{r}_{xs} \triangleq E\{\mathbf{x}^*(t)s(t)\} = S\mathbf{v}^* \tag{3.47}$$

and

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^*(t)\mathbf{x}^T(t)\} = \mathbf{R}_{ss} + \mathbf{R}_{nn} \tag{3.48}$$

3.3 | OPTIMUM ARRAY PROCESSING FOR NARROWBAND APPLICATIONS

As previously noted, several different performance measures are adopted to govern the operation of the adaptive processor that adjusts the weighting for each of the sensor element outputs of Figure 1-1. Now consider the problem of formulating four popular performance measures in terms of complex envelope signal characterizations and of determining the optimum steady-state solution for the adaptive weight vector.

The following performance measures are considered:

1. Mean square error (MSE) criterion
2. Signal-to-noise ratio criterion
3. Maximum likelihood (ML) criterion
4. Minimum noise variance (MV) criterion

These criteria form the basis for adaptive algorithms for narrowband systems. Narrowband adaptive arrays have complex weights in each channel. For wideband signals, however, the adaptive array has a set of linear filters that are usually approximated by tapped delay lines in each channel of the array. Thus, consideration of the narrowband processing case is slightly easier, although conceptually the wideband processor is regarded as a set of frequency-dependent complex weights.

The optimum solutions for the complex weight vector for each of the aforementioned four performance measures are derived first. Since it is desirable to be equally familiar with real and complex notation, the derivations are carried out using real notation; the corresponding complex solutions are then given so the student may develop the complex results using the derivations for real quantities but employing complex notation. Following the derivation of the desired results for each performance measure, it is shown that each solution is closely related to a single optimal solution by means of factorization of the results: this factorization provides the connection between the various solutions obtained and the form known as the Wiener solution.

3.3.1 The MSE Performance Measure

The MSE performance measure was considered by Widrow et al. [5] for the conventional array configuration, and additional procedures based on this criterion have been developed

and further extended [23–25]. Suppose the desired directional signal $s(t)$ is known and represented by a reference signal $d(t)$. This assumption is never strictly met in practice because a communication signal cannot possibly be known a priori if it is to convey information; hence, the desired signal must be unknown in some respect. Nevertheless, it turns out that in practice enough is usually known about the desired signal that a suitable reference signal $d(t)$ is obtained to approximate $s(t)$ in some sense by appropriately processing the array output signal. For example, when $s(t)$ is an amplitude modulated signal, it is possible to use the carrier component of $s(t)$ for $d(t)$ and still obtain suitable operation. Consequently, the desired or "reference" signal concept is a valuable tool, and one can proceed with the analysis as though the adaptive processor had a complete desired signal characterization.

The difference between the desired array response and the actual array output signal defines an error signal as shown in Figure 3-3:

$$\varepsilon(t) = d(t) - \mathbf{w}^T\mathbf{x}(t) \tag{3.49}$$

The squared error can therefore be written as

$$\varepsilon^2(t) = d^2(t) - 2d(t)\mathbf{w}^T\mathbf{x}(t) + \mathbf{w}^T\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{w} \tag{3.50}$$

Taking expected values of both sides of (3.50) then yields

$$E\{\varepsilon^2(t)\} = \overline{d^2(t)} - 2\mathbf{w}^T\mathbf{r}_{xd} + \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{3.51}$$

where

$$\mathbf{r}_{xd} = \begin{bmatrix} \overline{x_1(t)d(t)} \\ \overline{x_2(t)d(t)} \\ \vdots \\ \overline{x_N(t)d(t)} \end{bmatrix} \tag{3.52}$$

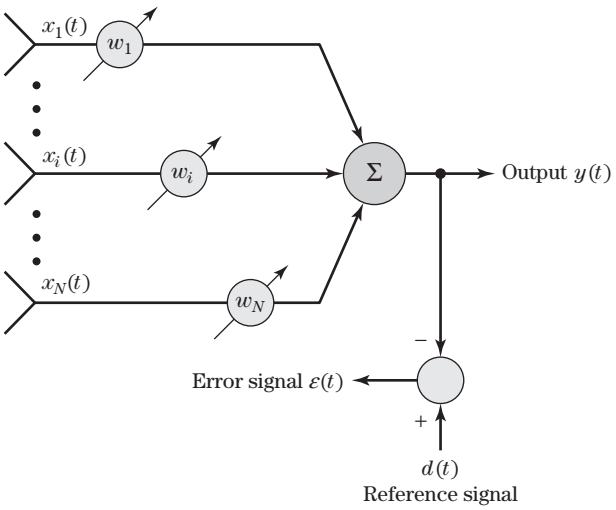


FIGURE 3-3 ■ Basic adaptive array structure with known desired signal.

Since $d(t) = s(t)$, from (3.39) it follows that $\overline{d^2(t)} = S$ so that

$$E\{\varepsilon^2(t)\} = S - 2\mathbf{w}^T\mathbf{r}_{xd} + \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{3.53}$$

Since (3.53) is a quadratic function of $\mathbf{w}$, its extremum is a minimum. Therefore, the value of $\mathbf{w}$ that minimizes of $E\{\varepsilon^2(t)\}$ is found by setting the gradient of (3.53) with respect to the weight vector equal to zero, that is,

$$_w\overline{(\varepsilon^2)} = 0 \tag{3.54}$$

Since

$$_w\overline{(\varepsilon^2)} = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}\mathbf{w} \tag{3.55}$$

it follows that the optimum choice for the weight vector must satisfy

$$\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} = \mathbf{r}_{xd} \qquad \text{or} \qquad \mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{3.56}$$

Equation (3.56) is the Wiener-Hopf equation in matrix form, and its solution, $\mathbf{w}_{\text{opt}}$, is consequently referred to as the optimum Wiener solution.

If we use $d(t) = s(t)$ and (3.39) and (3.42), it then follows that

$$\mathbf{r}_{xd} = E\{\mathbf{x}d\} = S\mathbf{v} \tag{3.57}$$

so that

$$\mathbf{w}_{\text{MSE}} = S\mathbf{R}_{xx}^{-1}\mathbf{v} \tag{3.58}$$

where it is assumed that $\mathbf{R}_{xx}$ is nonsingular so that $\mathbf{R}_{xx}^{-1}$ exists. Setting the weight vector equal to $\mathbf{w}_{MSE}$, the resulting minimum MSE is found from (3.53) to be

$$\overline{\varepsilon_{\text{min}}^2} = S - \mathbf{r}_{xd}^T\mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{3.59}$$

For complex quantities, (3.53), (3.58), and (3.59) become

$$\overline{|\varepsilon(t)|^2} = S - 2\text{Re}\{\mathbf{w}^\dagger\mathbf{r}_{xd}\} + \mathbf{w}^\dagger\mathbf{R}_{xx}\mathbf{w} \tag{3.60}$$

$$w_{\text{MSE}} = S\mathbf{R}_{xx}^{-1}\mathbf{v}^* \tag{3.61}$$

$$\overline{|\varepsilon|_{\text{min}}^2} = S - \mathbf{r}_{xd}^\dagger\mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{3.62}$$

3.3.2 The SNR Performance Measure

Adaptive algorithms based on maximizing the SNR have been developed for the conventional array configuration in communication and detection systems [26–28]. The output signal from the adaptive array of Figure 3-1 is written as

$$y(t) = \mathbf{w}^T\mathbf{x}(t) \tag{3.63}$$

where the input signal vector may be regarded as composed of a signal component $\mathbf{s}(t)$ and a noise component $\mathbf{n}(t)$ so that

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t) \tag{3.64}$$

The signal and noise components of the array output signal may therefore be written as

$$y_s(t) = \mathbf{w}^T \mathbf{s}(t) = \mathbf{s}^T(t)\mathbf{w} \tag{3.65}$$

and

$$y_n(t) = \mathbf{w}^T \mathbf{n}(t) = \mathbf{n}^T(t)\mathbf{w} \tag{3.66}$$

where

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \quad \text{and} \quad \mathbf{n}(t) = \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_N(t) \end{bmatrix} \tag{3.67}$$

Consequently, the output signal power may be written as

$$E\{|y_s(t)|^2\} = |\overline{\mathbf{w}^T \mathbf{s}}|^2 \tag{3.68}$$

and the output noise power is

$$E\{|y_n(t)|^2\} = |\overline{\mathbf{w}^T \mathbf{n}}|^2 \tag{3.69}$$

The output SNR is therefore given by

$$\left(\frac{s}{n}\right) = \frac{|\overline{\mathbf{w}^T \mathbf{s}}|^2}{|\overline{\mathbf{w}^T \mathbf{n}}|^2} = \frac{\mathbf{w}^T \left[\overline{\mathbf{s}\mathbf{s}^T}\right] \mathbf{w}}{\mathbf{w}^T \left[\overline{\mathbf{n}\mathbf{n}^T}\right] \mathbf{w}} = \frac{\mathbf{w}^T \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{nn} \mathbf{w}} \tag{3.70}$$

The ratio given by (3.70) is rewritten as

$$\left(\frac{s}{n}\right) = \frac{\mathbf{z}^T \mathbf{R}_{nn}^{-1/2} \mathbf{R}_{ss} \mathbf{R}_{nn}^{-1/2} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \tag{3.71}$$

where

$$\mathbf{z} \triangleq \mathbf{R}_{nn}^{1/2} \mathbf{w} \tag{3.72}$$

Equation (3.70) may be recognized as a standard quadratic form and is bounded by the minimum and maximum eigenvalues of the symmetric matrix $\mathbf{R}_{nn}^{-1/2} \mathbf{R}_{ss} \mathbf{R}_{nn}^{-1/2}$ (or, more conveniently, $\mathbf{R}_{nn}^{-1} \mathbf{R}_{ss}$) [29]. The optimization of (3.70) by appropriately selecting the weight vector $\mathbf{w}$ consequently results in an eigenvalue problem where the ratio (s/n) must satisfy the relationship [30]

$$\mathbf{R}_{ss} \mathbf{w} = \left(\frac{s}{n}\right) \mathbf{R}_{nn} \mathbf{w} \tag{3.73}$$

in which (s/n) now represents an eigenvalue of the symmetric matrix noted already. The maximum eigenvalue satisfying (3.73) is denoted by $(s/n)_{\text{opt}}$. Corresponding to $(s/n)_{\text{opt}}$, a unique eigenvector $\mathbf{w}_{\text{opt}}$ represents the optimum element weights. Therefore

$$\mathbf{R}_{ss} \mathbf{w}_{\text{opt}} = \left(\frac{s}{n}\right)_{\text{opt}} \mathbf{R}_{nn} \mathbf{w}_{\text{opt}} \tag{3.74}$$

Substitution of (3.70) corresponding to $(s/n)_{\text{opt}}$ into (3.74) yields

$$\mathbf{R}_{ss}\mathbf{w}_{\text{opt}} = \frac{\mathbf{w}_{\text{opt}}^T \mathbf{R}_{ss}\mathbf{w}_{\text{opt}}}{\mathbf{w}_{\text{opt}}^T \mathbf{R}_{nn}\mathbf{w}_{\text{opt}}}\mathbf{R}_{nn}\mathbf{w}_{\text{opt}} \tag{3.75}$$

Substituting $\mathbf{R}_{ss} = [\overline{\mathbf{s}\mathbf{s}^T}]$ and noting that $\mathbf{s}^T\mathbf{w}_{\text{opt}}$ is a scalar quantity occurring on both sides of (3.75) that may be canceled, we get the result

$$\mathbf{s} = \frac{\mathbf{w}_{\text{opt}}^T \mathbf{s}}{\mathbf{w}_{\text{opt}}^T \mathbf{R}_{nn}\mathbf{w}_{\text{opt}}} \cdot \mathbf{R}_{nn}\mathbf{w}_{\text{opt}} \tag{3.76}$$

The ratio $\left(\mathbf{w}_{\text{opt}}^T \mathbf{s}/\mathbf{w}_{\text{opt}}^T \mathbf{R}_{nn}\mathbf{w}_{\text{opt}}\right)$ may be seen as just a complex (scalar) number, denoted here by c. It therefore follows that

$$\mathbf{w}_{\text{opt}} = \left(\frac{1}{c}\right)\mathbf{R}_{nn}^{-1}\mathbf{s} \tag{3.77}$$

Since from (3.39) the envelope of $\mathbf{s}$ is just $\sqrt{S}\,\mathbf{v}$, it follows that

$$\mathbf{w}_{\text{SNR}} = \alpha\mathbf{R}_{nn}^{-1}\mathbf{v} \tag{3.78}$$

where

$$\alpha = \frac{\sqrt{S}}{c}$$

The maximum possible value that $(s/n)_{\text{opt}}$ is derived by converting the original system into orthonormal system variables. Since $\mathbf{R}_{nn}$ is a positive definite Hermitian matrix, it is diagonalized by a nonsingular coordinate transformation as shown in Figure 3-4. Such a transformation is selected so that all element channels have equal noise power components that are uncorrelated. Denote the transformation matrix that accomplishes this diagonalization by $\mathbf{A}$ so that

$$\mathbf{s}' = \mathbf{A}\mathbf{s} \tag{3.79}$$

and

$$\mathbf{n}' = \mathbf{A}\mathbf{n} \tag{3.80}$$

where a prime ($'$) denotes quantities after the transformation.

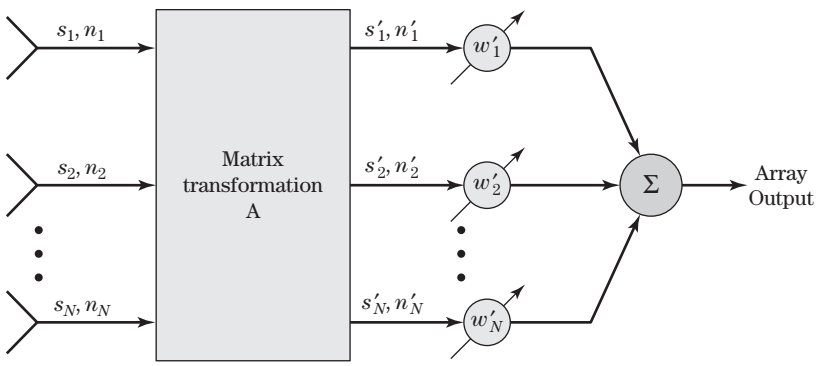


FIGURE 3-4 ■ Functional representation of orthonormal transformation and adaptive weight combiner equivalent to system of Figure 3-1.

The signal component of the array output now becomes

$$y_s = \mathbf{w}'^T \mathbf{s}' = \mathbf{w}'^T \mathbf{A}\mathbf{s} \tag{3.81}$$

and the noise component becomes

$$y_n = \mathbf{w}'^T \mathbf{n}' = \mathbf{w}'^T \mathbf{A}\mathbf{n} \tag{3.82}$$

For the array output of the system in Figure 3-4 to be equivalent to the output of the system in Figure 3-1, it is necessary that

$$\mathbf{w} = \mathbf{A}^T \mathbf{w}' \tag{3.83}$$

The output noise power of the orthonormal system is given by

$$E\{|y_n(t)|^2\} = E\{|\mathbf{w}'^T \mathbf{n}'|^2 = \mathbf{w}'^T E\{\mathbf{n}'\mathbf{n}'^T\}\mathbf{w}' \tag{3.84}$$

Since the transformation matrix $\mathbf{A}$ decorrelates the various noise components and equalizes their powers, the covariance matrix of the noise process $\mathbf{n}'(t)$ is just the identity matrix, that is,

$$E\{\mathbf{n}'\mathbf{n}'^T\} = \mathbf{I}_N \tag{3.85}$$

It immediately follows from (3.84) and (3.85) that

$$E\{|y_n(t)|^2\} = \mathbf{w}'^T \mathbf{w}' = ||\mathbf{w}'||^2 \tag{3.86}$$

The output noise power of the original system of Figure 3-1 is given by

$$E\{|y_n(t)|^2\} = \mathbf{w}^T \mathbf{R}_{nn} \mathbf{w} \tag{3.87}$$

Substituting (3.83) into (3.87), it follows that

$$E\{|y_n(t)|^2\} = \mathbf{w}'^T \mathbf{A}\mathbf{R}_{nn}\mathbf{A}^T \mathbf{w}' \tag{3.88}$$

For the output noise power of the two systems to be equivalent, it is necessary that

$$\mathbf{A}\mathbf{R}_{nn}\mathbf{A}^T = \mathbf{I}_N \tag{3.89}$$

or

$$\mathbf{R}_{nn} = [\mathbf{A}^T \mathbf{A}]^{-1} \tag{3.90}$$

Equation (3.90) simply expresses the fact that the transformation $\mathbf{A}$ diagonalizes and normalizes the matrix $\mathbf{R}_{nn}$.

The signal output of the orthonormal array system is given by (3.81). Applying the Cauchy-Schwartz inequality (see Appendix D and [4]) to this expression immediately yields an upper bound on the array output signal power as

$$|y_S(t)|^2 \leq ||\mathbf{w}'||^2 ||\mathbf{s}'||^2 \tag{3.91}$$

where

$$||\mathbf{s}'||^2 = \mathbf{s}'^T \mathbf{s}' \qquad \text{and} \qquad ||\mathbf{w}'||^2 = \mathbf{w}'^T \mathbf{w}' \tag{3.92}$$

From (3.86) and (3.91) it follows that the maximum possible value of the SNR is given by

$$\mathrm{SNR}_{\max} = ||\mathbf{s}'||^2 \tag{3.93}$$

Substituting (3.79) into (3.83) and using (3.93) and (3.90), we find it then follows that

$$\mathrm{SNR}_{\mathrm{opt}} = \mathbf{s}^T \mathbf{R}_{nn}^{-1} \mathbf{s} \tag{3.94}$$

For complex quantities, (3.70) becomes

$$\left(\frac{s}{n}\right) = \frac{\mathbf{w}^\dagger \left[\overline{\mathbf{s}^* \mathbf{s}^T}\right] \mathbf{w}}{\mathbf{w}^\dagger \left[\overline{\mathbf{n}^* \mathbf{n}^T}\right] \mathbf{w}} = \frac{\mathbf{w}^\dagger \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{R}_{nn} \mathbf{w}} \tag{3.95}$$

Equation (3.78) is replaced by

$$\mathbf{w}_{\mathrm{SNR}} = \alpha \mathbf{R}_{nn}^{-1} \mathbf{v}^* \tag{3.96}$$

(3.90) is now

$$\mathbf{R}_{nn} = [\mathbf{A}^T \mathbf{A}^*]^{-1} \tag{3.97}$$

whereas (3.94) becomes

$$\mathrm{SNR}_{\mathrm{opt}} = \mathbf{s}^T \mathbf{R}_{nn}^{-1} \mathbf{s}^* \tag{3.98}$$

Designing the adaptive processor so that the weights satisfy $\mathbf{R}_{nn}\mathbf{w} = \alpha \mathbf{v}^*$ means that the output SNR is the governing performance criterion, even for the quiescent environment (when no jamming signal and no desired signal are present). It is usually desirable, however, to compromise the output SNR to exercise some control over the array beam pattern sidelobe levels. An alternative performance measure that yields more flexibility in beam shaping is introduced in the manner described in the following material.

Suppose in the normal quiescent signal environment that the most desirable array weight vector selection is given by $\mathbf{w}_q$ (where now $\mathbf{w}_q$ represents the designer's most desirable compromise among, for example, gain or sidelobe levels). For this quiescent environment the signal covariance matrix is $\mathbf{R}_{nn_q}$. Define the column vector $\mathbf{t}$ by

$$\mathbf{R}_{nn_q} \mathbf{w}_q = \alpha \mathbf{t}^* \tag{3.99}$$

On comparing (3.99) with (3.96), we see that the ratio being maximized is no longer (3.95) but is the modified ratio given by

$$\frac{|\mathbf{w}^\dagger \mathbf{t}|^2}{\mathbf{w}^\dagger \mathbf{R}_{nn} \mathbf{w}} \tag{3.100}$$

This ratio is a more general criterion than the SNR and is often used for practical applications. The vector $\mathbf{t}$ is referred to as a generalized signal vector, and the ratio (3.100) is called the generalized signal-to-noise ratio (GSNR). For tutorial purposes, the ordinary SNR is usually employed, although the GSNR is often used in practice.

The adaptive array of Figure 3-1 is a generalization of a coherent sidelobe canceller (CSLC). As an illustration of the application of the foregoing SNR performance measure

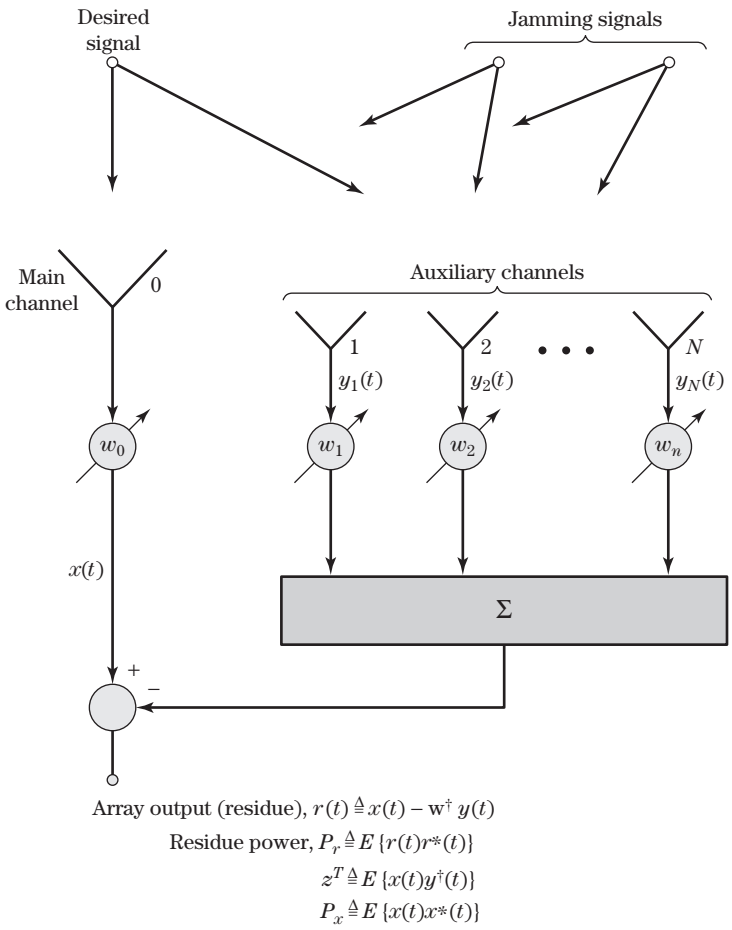


FIGURE 3-5 ■ Coherent sidelobe cancellation (CSLC) system.

concepts, it is useful to show how sidelobe cancellation may be regarded as a special case of an adaptive array.

The functional diagram of a standard sidelobe cancellation system is shown in Figure 3-5. A sidelobe canceller consists of a main antenna with high gain designated as channel *o* and *N* auxiliary antenna elements with their associated channels. The auxiliary antennas have a gain approximately equal to the average sidelobe level of the main antenna gain pattern [18,31]. If the gain of the auxiliary antenna is greater than the highest sidelobe, then the weights in the auxiliary channel are less than one. A properly designed auxiliary antenna provides replicas of the jamming signals appearing in the sidelobes of the main antenna pattern. These replica jamming signals provide coherent cancellation in the main channel output signal, thereby providing an interference-free array output response from the sidelobes. Jamming signals entering the main beam cannot be canceled.

In applying the GSNR performance measure, it is first necessary to select a $\mathbf{t}$ column vector. Since any desired signal component contributed by the auxiliary channels to the total array desired signal output is negligible compared with the main channel contribution, and the main antenna has a carefully designed pattern, a reasonable choice for $\mathbf{t}$ is the $N + 1$

component vector

$$\mathbf{t} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.101}$$

which preserves the main channel response signal. From (3.99) the optimum weight vector must satisfy

$$\mathbf{R}'_{nn}\mathbf{w}' = \alpha\mathbf{t} \tag{3.102}$$

where $\mathbf{R}'_{nn}$ is the $(N+1) \times (N+1)$ covariance matrix of all channel input signals (in the absence of the desired signal), and $\mathbf{w}'$ is the $N+1$ column vector of all channel weights.

Let the $N \times N$ covariance matrix of the auxiliary channel input signals (again in the absence of the desired signal) be denoted by $\mathbf{R}_{nn}$, and let $\mathbf{w}$ be the N-component column vector of the auxiliary channel weights. Equation (3.102) may now be partitioned to yield

$$\left[\begin{array}{c|c} P_0 & \Psi^\dagger \\ \hline \Psi & \mathbf{R}_{nn} \end{array}\right] \left[\begin{array}{c} w_0 \\ \hline \mathbf{w} \end{array}\right] = \left[\begin{array}{c} \alpha \\ \hline \mathbf{0} \end{array}\right] \tag{3.103}$$

where $P_0 =$ noise power output of the main channel and

$$\mathbf{\Psi} \triangleq E\begin{bmatrix} x_1^* x_0 \\ x_2^* x_0 \\ \vdots \\ x_N^* x_0 \end{bmatrix} = \begin{array}{l} \text{cross-correlation vector between the} \\ \text{main channel output and the output of} \\ \text{each auxiliary channel} \end{array}$$

Equation (3.103) in turn may be written as two separate equations

$$\text{(scalar equation) } P_0 w_0 + \mathbf{\Psi}^\dagger \mathbf{w} = \alpha \tag{3.104}$$

$$\text{(matrix equation) } \mathbf{R}_{nn}\mathbf{w} = -w_0\mathbf{\Psi} \tag{3.105}$$

The solution represented by (3.102) is implemented using the $N+1$ weights w_0, $w_1, \ldots, w_N$ shown in Figure 3-5. It is also possible to attain the same optimum SNR using only the N weights in the auxiliary channels, however. To see this, note that, if $\mathbf{w}'$ is optimum for a given noise environment (a given $\mathbf{R}'_{nn}$), then any scalar multiple of $\mathbf{w}'$ will also yield the optimum SNR since the output SNR does not depend on the absolute level of $\mathbf{w}'$. Consequently, the scalar weight w_0 in (3.104) is fixed at any convenient nonzero value. Note that since $(\mathbf{w}')^T\mathbf{t} = w_0$, the GSNR represented by (3.100) will never be optimized by $w_0 = 0$. Only the weights in the auxiliary channels need adjusted when w_0 is fixed at some convenient value $\hat{w}_0$, so that

$$\mathbf{R}_{nn}\mathbf{w}_{\text{opt}} = -\hat{w}_0\mathbf{\Psi} \tag{3.106}$$

Since $\hat{w}_0$ is fixed and (3.106) optimizes the GSNR ratio of $\hat{w}_0$ to the output noise power, it follows that $\mathbf{w}_{\text{opt}}$ must be minimizing the output noise power resulting from the sidelobe response.

3.3.3 The ML Performance Measure

When the desired signal waveform is completely unknown (e.g., with seismic waves), then the desired signal is a time function that must be estimated. The derivation of the maximum likelihood estimator of the desired signal requires the assumption that the noise components have a multidimensional Gaussian distribution [22,32].

The input signal vector may once again be written as

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t) \tag{3.107}$$

where

$$\mathbf{s}(t) = s(t)\mathbf{v} \tag{3.108}$$

for the conventional array of Figure 3-1, and an estimate of $s(t)$ is desired. Define the likelihood function of the input signal vector as

$$\mathfrak{L}[\mathbf{x}(t)] = -\ln[P\{\mathbf{x}(t)/\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t)\}] \tag{3.109}$$

where $P\{z/y\}$ is the probability density function for z given the event y. Thus, the likelihood function defined by (3.109) is the negative natural logarithm of the probability density function for the input signal vector $\mathbf{x}(t)$ given that $\mathbf{x}(t)$ contains both the desired signal and interfering noise.

Now assume that the noise vector $\mathbf{n}(t)$ is a stationary, zero-mean Gaussian random vector with a covariance matrix $\mathbf{R}_{nn}$. Furthermore, assume that $\mathbf{x}(t)$ is also a stationary Gaussian random vector having the mean $s(t)\mathbf{v}$, where $s(t)$ is a deterministic but unknown quantity. With these assumptions, the likelihood function is written as

$$\mathfrak{L}[\mathbf{x}(t)] = c[\mathbf{x}(t) - s(t)\mathbf{v}]^T \mathbf{R}_{nn}^{-1}[\mathbf{x}(t) - s(t)\mathbf{v}] \tag{3.110}$$

where c is a scalar constant independent of $\mathbf{x}(t)$ and $s(t)$.

The maximum likelihood processor is obtained by solving for the estimate of $s(t)$, denoted by $\hat{s}(t)$, which maximizes (3.110). Taking the partial derivative of $\mathfrak{L}[\mathbf{x}(t)]$ with respect to $s(t)$ and setting the result equal to zero yields

$$0 = \frac{\partial \mathfrak{L}[\mathbf{x}(t)]}{\partial s(t)} = -2\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{x} + 2\hat{s}(t)\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{v} \tag{3.111}$$

It immediately follows that the estimate $\hat{s}(t)$ that maximizes $\mathfrak{L}[\mathbf{x}(t)]$ is given by

$$\hat{s}(t)\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{v} = \mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.112}$$

Since the quantity $\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{v}$ is a scalar, (3.112) is rewritten as

$$\hat{s}(t) = \frac{\mathbf{v}^T \mathbf{R}_{nn}^{-1}}{\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{v}}\mathbf{x}(t) \tag{3.113}$$

which is of the form $\hat{s}(t) = \mathbf{w}_{\mathrm{ML}}^T \mathbf{x}(t)$. Consequently, the maximum likelihood weight vector is given by

$$\mathbf{w}_{\mathrm{ML}} = \frac{\mathbf{R}_{nn}^{-1}\mathbf{v}}{\mathbf{v}^T \mathbf{R}_{nn}^{-1}\mathbf{v}} \tag{3.114}$$

For complex quantities (3.110) becomes

$$\Re\,[\mathbf{x}(t)] = c[\mathbf{x}(t) - s(t)\mathbf{v}]^{\dagger}\mathbf{R}_{nn}^{-1}[\mathbf{x}(t) - s(t)\mathbf{v}] \tag{3.115}$$

Likewise (3.112) is replaced by

$$\hat{s}(t)\mathbf{v}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{v} = \mathbf{v}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}(t) \tag{3.116}$$

and the result expressed by (3.114) then becomes

$$\mathbf{w}_{\mathrm{ML}} = \frac{\mathbf{R}_{nn}^{-1}\mathbf{v}}{\mathbf{v}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{v}} \tag{3.117}$$

3.3.4 The MV Performance Measure

When both the desired signal $s(t)$ and the desired signal direction are known (as with a signal aligned array), then minimizing the output noise variance provides a means of ensuring good signal reception, and methods based on this performance measure have been developed [33,34]. For the signal aligned array of Figure 3-2, the array output is given by

$$y(t) = \mathbf{w}^{T}\mathbf{z}(t) = s(t)\sum_{i=1}^{N} w_i + \sum_{i=1}^{N} w_i n_i' \tag{3.118}$$

where the n_i' represent the noise components after the signal aligning phase shifts. When we constrain the sum of the array weights to be unity, then the output signal becomes

$$y(t) = s(t) + \mathbf{w}^{T}\mathbf{n}'(t) \tag{3.119}$$

which represents an unbiased output signal since

$$E\{y(t)\} = s(t) \tag{3.120}$$

The variance of the array output may therefore be expressed as

$$\mathrm{var}[y(t)] = E\left\{\mathbf{w}^{T}\mathbf{n}'(t)\mathbf{n}'^{T}(t)\mathbf{w}\right\} = \mathbf{w}^{T}\mathbf{R}_{n'n'}\mathbf{w} \tag{3.121}$$

The relationship between $\mathbf{n}(t)$, the noise vector appearing before the signal aligning phase shifts, and $\mathbf{n}'(t)$ is given by

$$\mathbf{n}'(t) = \mathbf{\Phi}\mathbf{n}(t) \tag{3.122}$$

where $\mathbf{\Phi}$ is the diagonal unitary transformation described by

$$\mathbf{\Phi} = \begin{bmatrix} e^{j\phi_1} & & 0 \\ & e^{j\phi_2} & \\ & \ddots & \\ 0 & & e^{j\phi_N} \end{bmatrix} \tag{3.123}$$

The variance of the array output remains unaffected by such a unitary transformation so that

$$\text{var}[y(t)] = \mathbf{w}^T \mathbf{R}_{n'n'} \mathbf{w} = \mathbf{w}^T \mathbf{R}_{nn} \mathbf{w} \tag{3.124}$$

It is now desired to minimize (3.124) subject to the constraint

$$\mathbf{w}^T \mathbf{1} = 1 \tag{3.125}$$

where

$$\mathbf{1} = [1, 1, \ldots, 1]^T \tag{3.126}$$

To solve this minimization problem, form the modified performance criterion

$$\mathfrak{P}_{\text{MV}} = \frac{1}{2} \mathbf{w}^T \mathbf{R}_{nn} \mathbf{w} + \lambda [1 - \mathbf{w}^T \mathbf{1}] \tag{3.127}$$

where the factor λ is a Lagrange multiplier. Since $\mathfrak{P}_{\text{MV}}$ is a quadratic function of $\mathbf{w}$, it follows that the optimal choice for $\mathbf{w}$ may be found by setting the gradient $_w\mathfrak{P}_{\text{MV}}$ equal to zero. The gradient is given by

$$_w\mathfrak{P}_{\text{MV}} = \mathbf{R}_{nn} \mathbf{w} - \lambda \mathbf{1} \tag{3.128}$$

so that

$$\mathbf{w}_{\text{MV}} = \mathbf{R}_{nn}^{-1} \mathbf{1} \lambda \tag{3.129}$$

The optimum solution $\mathbf{w}_{\text{MV}}$ satisfies the constraint condition so that

$$\mathbf{w}_{\text{MV}}^T \mathbf{1} = 1 \tag{3.130}$$

and on substituting (3.130) into (3.129) there results

$$\lambda = \frac{1}{\mathbf{1}^T \mathbf{R}_{nn}^{-1} \mathbf{1}} \tag{3.131}$$

It follows immediately that

$$\mathbf{w}_{\text{MV}} = \frac{\mathbf{R}_{nn}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{R}_{nn}^{-1} \mathbf{1}} \tag{3.132}$$

where $\mathbf{w}_{\text{MV}}$ satisfies (3.130).

On substituting (3.132) into (3.124), the minimum value of the output noise variance is found to be

$$\text{var}_{\min}[y(t)] = \frac{\mathbf{1}}{\mathbf{1}^T \mathbf{R}_{nn}^{-1} \mathbf{1}} \tag{3.133}$$

If complex quantities are introduced, all the foregoing expressions remain unchanged, except the definition of the covariance matrix $\mathbf{R}_{nn}$ must be appropriate for complex vectors.

3.3.5 Factorization of the Optimum Solutions

The solutions obtained in the preceding sections all applied to the conventional array, with the single exception of the minimum variance weights, where it was necessary to use the signal aligned array to define a well-posed problem. All of the solutions are closely related to one another since (as will be shown) they differ only by a scalar gain factor. Hence, the different solutions possess identical output SNRs. This relationship is shown by factoring the various solutions into a linear matrix filter followed by a scalar processor as described in the following paragraphs [35].

The optimum weight vector obtained for the minimum MSE performance measure is written from (3.61) as

$$\mathbf{w}_{\text{MSE}} = \mathbf{R}_{xx}^{-1} S\mathbf{v}^* = [S\mathbf{v}^*\mathbf{v}^T + \mathbf{R}_{nn}]^{-1} S\mathbf{v}^* \tag{3.134}$$

Applying the matrix identity (D.10) of Appendix D to (3.134) results in

$$\begin{aligned}\mathbf{w}_{\text{MSE}} &= \left[S\mathbf{R}_{nn}^{-1} - \frac{S^2\mathbf{R}_{nn}^{-1}\mathbf{v}^*\mathbf{v}^T\mathbf{R}_{nn}^{-1}}{1 + S\mathbf{v}^T\mathbf{R}_{nn}^{-1}\mathbf{v}^*}\right]\mathbf{v}^* \\ &= \left[\frac{S}{1 + S\mathbf{v}^T\mathbf{R}_{nn}^{-1}\mathbf{v}^*}\right]\mathbf{R}_{nn}^{-1}\mathbf{v}^*\end{aligned} \tag{3.135}$$

From (3.135), the minimum MSE weights are the product of a matrix filter $\mathbf{R}_{nn}^{-1}\mathbf{v}^*$ (which is also common to the other weight vector solutions) and a scalar factor. Since the MV solution is applied only to a signal aligned array, for the other solutions to pertain to the signal aligned array it is necessary to replace the $\mathbf{v}$ vector only wherever it occurs by the $\mathbf{1}$ vector.

Now consider the noise power N_0 and the signal power S_0 that appear at the output of the linear matrix filter corresponding to $\mathbf{w} = \mathbf{R}_{nn}^{-1}\mathbf{v}^*$ as follows:

$$N_0 = \mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w} = \mathbf{v}^T\mathbf{R}_{nn}^{-1}\mathbf{v}^* \tag{3.136}$$

and

$$S_0 = \mathbf{w}^\dagger\mathbf{R}_{ss}\mathbf{w} = SN_0^2 \tag{3.137}$$

The optimal weight vector solution given in Sections 3.3.1 and 3.3.3 for the MSE and the ML ratio performance measures are now written in terms of the previous quantities as

$$\mathbf{w}_{\text{MSE}} = \frac{1}{N_0} \cdot \frac{S_0}{N_0 + S_0} \cdot \mathbf{R}_{nn}^{-1}\mathbf{v}^* \tag{3.138}$$

and

$$\mathbf{w}_{\text{ML}} = \frac{1}{N_0} \cdot \mathbf{R}_{nn}^{-1}\mathbf{v}^* \tag{3.139}$$

Likewise, for the signal aligned array (where $\mathbf{v} = \mathbf{1}$) the ML weights reduce to the unbiased, MV weights, that is,

$$\mathbf{w}_{\text{ML}}|_{\mathbf{v}=\mathbf{1}} = \frac{\mathbf{R}_{nn}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{nn}^{-1}\mathbf{1}} = \mathbf{w}_{\text{MV}} \tag{3.140}$$

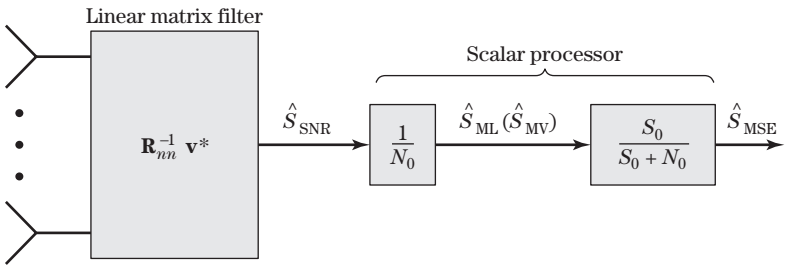


FIGURE 3-6 ■ The factored minimum MSE processor showing relationships to the maximum SNR, ML, and MV processors.

The previous expressions show that the minimum MSE processor can be factored into a linear matrix filter followed by a scalar processor that contains the estimates corresponding to the other performance measures, as shown in Figure 3-6. The different optimum weight vector previously derived be expressed by

$$\mathbf{w} = \beta \mathbf{R}_{nn}^{-1} \mathbf{v}^* \tag{3.141}$$

where β is an appropriate scalar gain; hence, they all yield the same SNR, which can then be expressed as

$$\left(\frac{s}{n}\right) = \frac{\mathbf{w}^\dagger \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{R}_{nn} \mathbf{w}} = \frac{\beta^2 S \mathbf{v}^T \left(\mathbf{R}_{nn}^{-1}\right) \mathbf{v}^* \mathbf{v}^T \mathbf{R}_{nn}^{-1} \mathbf{v}^*}{\beta^2 \mathbf{v}^T \left(\mathbf{R}_{nn}^{-1}\right) \mathbf{v}^*} = S \mathbf{v}^T R_{nn}^{-1} \mathbf{v}^* \tag{3.142}$$

For the case of a wideband processor, it can similarly be shown that the solutions to various estimation and detection problems are factored into a linear matrix filter followed by appropriate scalar processing. This development is undertaken in the next section.

The fact that the optimum weight vector solutions for an adaptive array using the different performance criteria indicated in the preceding section are all given (to within a constant factor) by the Wiener solution underscores the fundamental importance of the Wiener-Hopf equation in establishing theoretical adaptive array steady-state performance limits. These theoretical performance limits provide the designer with a standard for determining how much any improvement in array implementation can result in enhanced array steady-state performance, and they are a valuable tool for judging the merit of alternate designs.

3.4 | OPTIMUM ARRAY PROCESSING FOR BROADBAND APPLICATIONS

An array processor for passive sensing devices must decide whether the random processes observed at the array element outputs consist of a desired signal obscured by noise or noise alone. Performance measures like maximum output SNR or minimum MS, are not adequate for deciding whether a signal is present. It is well known from statistical detection theory, however, that decisions based on the likelihood ratio test minimize the risk associated with an incorrect decision. In addition, the likelihood ratio test yields decisions that are optimum for a wide range of performance criteria [36–38]. This section establishes the relationship between the likelihood ratio test and some other popular performance measures for broadband signal applications.

Let the observation vector $\mathbf{x}$ consist of elements representing the outputs from each of the array sensors $x_i(t)$, $i = 1, 2, \ldots, N$. The likelihood ratio is then given by the ratio of conditional probability density functions [39]

$$\Lambda(\mathbf{x}) \triangleq \frac{p[\mathbf{x}/\text{signal present}]}{p[\mathbf{x}/\text{signal absent}]} \tag{3.143}$$

If $\Lambda(\mathbf{x})$ exceeds a certain threshold η then the signal is assumed present, whereas if $\Lambda(\mathbf{x})$ is less than this threshold the signal is assumed absent. The ratio (3.143) therefore represents the likelihood that the sample $\mathbf{x}$ was observed, given that the signal is present relative to the likelihood that it was observed given that the signal is absent. Such an approach certainly comes far closer to determining the "best" system for a given class of decisions, since it assumes at the outset that the system makes such decisions and obtains the processor design accordingly.

It is worthwhile to mention briefly some extensions of the likelihood ratio test represented by (3.143). In many practical cases, one or several signal parameters (e.g., spatial location, phase, or signal energy) are uncertain. When uncertain signal parameters (denoted by $\boldsymbol{\theta}$) are present, one intuitively appealing approach is to explicitly estimate $\boldsymbol{\theta}$ (denoted by $\hat{\boldsymbol{\theta}}$) and use this estimate to form a classical generalized likelihood ratio (GLR) [40]

$$\Lambda_G(\mathbf{x}/\hat{\boldsymbol{\theta}}) \triangleq \frac{p[\mathbf{x}/\hat{\boldsymbol{\theta}}, \text{signal present}]}{p[\mathbf{x}/\text{signal absent}]} \tag{3.144}$$

Uncertain signal parameters can also be modeled as random variables with any prior information about them summarized in the form of an a priori density function $p(\boldsymbol{\theta})$. The likelihood ratio can then be written as the ratio of marginal density functions and is referred to as the Bayes likelihood ratio [41]

$$\Lambda_B(\mathbf{x}) \triangleq \frac{\int_\theta p[\mathbf{x}/\boldsymbol{\theta}, \text{signal present}]p(\boldsymbol{\theta})d\boldsymbol{\theta}}{p[\mathbf{x}/\text{signal absent}]} \tag{3.145}$$

where $\boldsymbol{\theta} \in \theta$.

It is well known that likelihood ratio tests are regarded as completely equivalent to matched filter processing in which the output SNR is maximized at a certain observation time $t = T$ [41–43]. Another goal of this section is to determine the role that matched filtering plays in optimum array processing.

An additional important concept in likelihood ratio tests is that of a "sufficient statistic" [44]. A brute force approach to the tests represented by the ratios (3.143) to (3.145) is to process the observation vector $\mathbf{x}$ to actually construct the ratio $\Lambda(\mathbf{x})$ and compare this ratio with a threshold η to make a decision. In many cases, although the observation vector is N-dimensional, the threshold decision occurs along a one-dimensional coordinate in the N-dimensional space, so that choosing a sufficient statistic simply amounts to picking a coordinate system in which one coordinate contains all the information required to make a decision. If such a sufficient statistic is available and is denoted by the scalar $l(\mathbf{x})$, then it is not necessary to construct an unwieldy ratio $\Lambda(\mathbf{x})$, since only the relatively simple quantity $l(\mathbf{x})$ needs be considered in arriving at the required decision.

Detection performance is commonly measured in terms of the receiver operating characteristic (ROC). The ROC is a figure displaying the probability of detection (given by $\Pr[\Lambda(\mathbf{x}) > \eta/\text{signal present}]$) versus the probability of false alarm (given by $\Pr[\Lambda(\mathbf{x}) > \eta/\text{signal absent}]$) with the SNR as an independent parameter [38,44]. Solutions for the

optimum detection processor may also be obtained very simply by working instead with the "detection index" d at a specified observation time T where

$$d \triangleq \frac{E\{y(T)/\text{signal present}\} - E\{y(T)/\text{signal absent}\}}{\{\text{var}[y(T)/\text{signal absent}]\}^{1/2}} \tag{3.146}$$

and $y(T)$ is the processor output at $t = T$. By maximizing the detection index, the detection performance is also optimized, since this corresponds to obtaining the greatest possible expected normalized difference of the processor output between the two signal conditions it is desired to discriminate.

The unification of a variety of problems occurring in estimation and detection theory is achieved by considering an observation vector described by

$$\mathbf{x}(t) = \mathbf{M}\mathbf{s}(t) + \mathbf{n}(t) \tag{3.147}$$

where $\mathbf{x}$ = observation vector
$\mathbf{n}$ = additive random noise vector having zero mean and noise covariance matrix $\mathbf{R}_{nn}$
$\mathbf{s}$ = signal vector (known, unknown, or random)
$\mathbf{M}$ = known transformation matrix

The treatment of (3.147) for different classes of problems given here generally follows the development presented by Cox [10]. In most cases, results are obtained using arguments only by analogy, which sacrifice the mathematical rigor of techniques that transform continuous time functions into discrete functions and use arguments based on Fourier series expansions of stationary random processes but indicate how important results come about with very little effort. Equation (3.147) is interpreted as a complex equation, where a complex random vector

$$\mathbf{z} = \zeta + j\gamma \tag{3.148}$$

is required to have the following two properties, which remain invariant under any linear transformation:

1. The real part $\boldsymbol{\zeta}$ and the imaginary part $\boldsymbol{\gamma}$ are both real random vectors having the same covariance matrix.
2. All components of $\boldsymbol{\zeta}$ and $\boldsymbol{\gamma}$ satisfy

$$E\{\zeta_l \gamma_m\} = -E\{\zeta_m \gamma_l\} \qquad \text{for all } l \text{ and } m \tag{3.149}$$

The development that follows involves some useful matrix properties and generalizations of the Schwartz inequality that are summarized for convenience in Appendix D. Gaussian random vectors are important in the subsequent analysis, so some useful properties of both real and complex Gaussian random vectors are given in Appendix E.

3.4.1 Estimation of a Random Signal

Assuming that $\mathbf{s}$ in (3.147) is random, then the problem is posed of estimating $\mathbf{s}$ given the observation vector $\mathbf{x}$. Let the mean value of $\mathbf{s}$ be given by

$$E\{\mathbf{s}\} = \mathbf{u} \tag{3.150}$$

and the associated covariance matrix be given by

$$E\{(\mathbf{s}-\mathbf{u})(\mathbf{s}-\mathbf{u})^{\dagger}\} = \mathbf{R}_{ss} \tag{3.151}$$

where $\mathbf{u}$ and $\mathbf{R}_{ss}$ are both known. The best estimate of $\mathbf{s}$, given $\mathbf{x}$, for a quadratic cost criterion, is just the conditional mean $E\{\mathbf{s}/\mathbf{x}\}$.

3.4.1.1 Gaussian Random Signal

When $\mathbf{s}$ and $\mathbf{n}$ are both Gaussian and independent, then $\mathbf{x}$ and $\mathbf{s}$ are jointly Gaussian, and the conditional mean $E\{\mathbf{s}/\mathbf{x}\}$ is obtained by considering the vector formed by combining $\mathbf{s}$ and $\mathbf{x}$ into a single vector. From (3.147), it follows that

$$E\left\{\begin{bmatrix}\mathbf{s}\\ \mathbf{x}\end{bmatrix}\right\} = \begin{bmatrix}\mathbf{u}\\ \mathbf{M}\mathbf{u}\end{bmatrix} \tag{3.152}$$

and

$$\operatorname{cov}\begin{bmatrix}\mathbf{s}\\ \mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{ss} & \mathbf{R}_{ss}\mathbf{M}^{\dagger}\\ \mathbf{M}\mathbf{R}_{ss} & \mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\end{bmatrix} \tag{3.153}$$

Applying (E.14) or (E.42) from Appendix E it follows immediately that

$$\hat{\mathbf{s}} = E\{\mathbf{s}/\mathbf{x}\} = \mathbf{u} + \mathbf{R}_{ss}\mathbf{M}^{\dagger}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right]^{-1}(\mathbf{x}-\mathbf{M}\mathbf{u}) \tag{3.154}$$

The associated covariance matrix of $\hat{\mathbf{s}}$ is obtained from (E.15) or (E.43) from Appendix E so that

$$\operatorname{cov}(\hat{\mathbf{s}}) = \mathbf{R}_{ss} - \mathbf{R}_{ss}\mathbf{M}^{\dagger}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right]^{-1}\mathbf{M}\mathbf{R}_{ss} \tag{3.155}$$

Applying the matrix identities (D.10) and (D.11) of Appendix D to (3.154) and (3.155) yields

$$\hat{\mathbf{s}} = \mathbf{u} + \left[\mathbf{R}_{ss}^{-1}+\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{M}\mathbf{u}) \tag{3.156}$$

or

$$\hat{\mathbf{s}} = \left[\mathbf{I}+\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\left[\mathbf{u}+\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}\right] \tag{3.157}$$

and

$$\operatorname{cov}(\hat{\mathbf{s}}) = \left[\mathbf{R}_{ss}^{-1}+\mathbf{M}^{\dagger}\mathbf{R}_{nn}\mathbf{M}\right]^{-1} \tag{3.158}$$

or

$$\operatorname{cov}(\hat{\mathbf{s}}) = \left[\mathbf{I}+\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{R}_{ss} \tag{3.159}$$

Letting $\mathbf{u} = \mathbf{0}$ yields the result for the interesting and practical zero-mean case, then (3.156) yields

$$\hat{\mathbf{s}} = \left[\mathbf{R}_{ss}^{-1}+\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.160}$$

which will be used later.

Since the mean, the mode, and the maximum likelihood of a Gaussian density function are equal, the best estimate of $\mathbf{s}$ corresponds to the maximum of the a posteriori density

function. In other words, the value of $\mathbf{s}$ is chosen that maximizes

$$p(\mathbf{s}/\mathbf{x}) = \frac{p(\mathbf{x}/\mathbf{s})p(\mathbf{s})}{p(\mathbf{x})} \tag{3.161}$$

From (3.147) it is seen that $p(\mathbf{x}/\mathbf{s})$ is Gaussian with mean $\mathbf{Ms}$ and covariance matrix $\mathbf{R}_{nn}$. Also, $p(\mathbf{s})$ is Gaussian with mean $\mathbf{u}$ and covariance matrix $\mathbf{R}_{ss}$. Consequently, from (3.161) and (E.49) of Appendix E it follows that

$$p(\mathbf{s}/\mathbf{x}) = (\text{const.}) \exp\left\{-\frac{\alpha}{2}\left[(\mathbf{x}-\mathbf{Ms})^\dagger \mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{Ms}) + (\mathbf{s}-\mathbf{u})^\dagger \mathbf{R}_{ss}^{-1}(\mathbf{s}-\mathbf{u})\right]\right\} \tag{3.162}$$

where $\alpha = 2$ for a complex random vector, $\alpha = 1$ for a real random vector, and "const." denotes a constant of proportionality. Choosing $\mathbf{s}$ to maximize (3.162) is equivalent to choosing $\mathbf{s}$ to minimize the following part of the exponent of (3.162)

$$J = (\mathbf{x}-\mathbf{Ms})^\dagger \mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{Ms}) + (\mathbf{s}-\mathbf{u})^\dagger \mathbf{R}_{ss}^{-1}(\mathbf{s}-\mathbf{u}) \tag{3.163}$$

Either maximizing (3.162) or minimizing (3.163) leads to (3.154) as the appropriate equation for the estimate $\hat{\mathbf{s}}$. Minimizing the exponent of (3.162) is sometimes used in nonlinear estimation problems, where the quantity $\mathbf{Ms}$ is replaced by a nonlinear function $\mathbf{m}(\mathbf{s})$ that has a conditional mean that is difficult to compute.

3.4.1.2 Non-Gaussian Random Signal

In the event that $\mathbf{s}$ and $\mathbf{n}$ are non-Gaussian random vectors, the conditional mean $E\{\mathbf{s}/\mathbf{x}\}$ remains the best estimate for a quadratic cost criterion. In many cases, convenient expressions for computing the conditional mean do not exist, and only second-order statistics are available. Sometimes the estimate given by (3.154) or its equivalent forms are used even though such an estimate no longer represents the conditional mean of $p(\mathbf{s}/\mathbf{x})$. Another approach often used in non-Gaussian problems is to find the linear estimate that minimizes the MSE. A linear estimate is one that has the form

$$\hat{\mathbf{s}} = \mathbf{a} + \mathbf{Bx} \tag{3.164}$$

The values assigned to $\mathbf{a}$ and $\mathbf{B}$ are determined by minimizing the expression

$$e = \text{trace}\{E[(\mathbf{s}-\hat{\mathbf{s}})(\mathbf{s}-\hat{\mathbf{s}})^\dagger]\} = E\{(\mathbf{s}-\hat{\mathbf{s}})^\dagger(\mathbf{s}-\hat{\mathbf{s}})\} \tag{3.165}$$

On combining (3.147) and (3.164) it follows that

$$(\mathbf{s}-\hat{\mathbf{s}}) = [\mathbf{I}-\mathbf{BM}](\mathbf{s}-\mathbf{u}) + \{[\mathbf{I}-\mathbf{BM}]\mathbf{u}-\mathbf{a}\} - \mathbf{Bn} \tag{3.166}$$

Consequently,

$$\begin{aligned} E\{(\mathbf{s}-\hat{\mathbf{s}})(\mathbf{s}-\hat{\mathbf{s}})^\dagger\} = {} & [\mathbf{I}-\mathbf{BM}]\mathbf{R}_{ss}[\mathbf{I}-\mathbf{M}^\dagger\mathbf{B}^\dagger] + \mathbf{B}\mathbf{R}_{nn}\mathbf{B}^\dagger \\ & + \{[\mathbf{I}-\mathbf{BM}]\mathbf{u}-\mathbf{a}\}\{[\mathbf{I}-\mathbf{BM}]\mathbf{u}-\mathbf{a}\}^\dagger \end{aligned} \tag{3.167}$$

By setting the gradient of (3.167) with respect to $\mathbf{a}$ equal to zero, the value of $\mathbf{a}$ that minimizes (3.165) is given by

$$\mathbf{a} = [\mathbf{I}-\mathbf{BM}]\mathbf{u} \tag{3.168}$$

After we substitute (3.168) into (3.167) and complete the square in the manner of (D.9) of Appendix D, it follows that

$$E\{(\mathbf{s}-\hat{\mathbf{s}})(\mathbf{s}-\hat{\mathbf{s}})^{\dagger}\} = \mathbf{R}_{ss} - \mathbf{R}_{ss}\mathbf{M}^{\dagger}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right]^{-1}\mathbf{M}\mathbf{R}_{ss} + \left\{\mathbf{B}-\mathbf{R}_{ss}\mathbf{M}^{\dagger}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right]^{-1}\right\}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right] \cdot \{\mathbf{B}^{\dagger}-[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}]^{-1}\mathbf{M}\mathbf{R}_{ss}\} \tag{3.169}$$

The value of $\mathbf{B}$ that minimizes (3.165) may easily be found from (3.169)

$$\mathbf{B} = \mathbf{R}_{ss}\mathbf{M}^{\dagger}\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}+\mathbf{R}_{nn}\right]^{-1} \tag{3.170}$$

By substituting the results from (3.168) and (3.170) into (3.164), the expression for $\hat{\mathbf{s}}$ corresponding to (3.154) is once again obtained. Likewise, by substituting (3.170) into (3.169) the same expression for the error covariance matrix as appeared in (3.155) also results. In the Gaussian case the estimate $\hat{s}$ given by (3.154) is the conditional mean, whereas in the non-Gaussian case the same estimate is the "best" linear estimate in the sense that it minimizes the MSE.

3.4.1.3 Application of Random Signal Estimation Results to Optimum Array Processing

The signal vector received by an array consisting of N sensors is expressed as

$$\mathbf{x}(t) = \int_{-\infty}^{t} \mathbf{m}(t-v)s(v)dv + \mathbf{v}(t) \tag{3.171}$$

where $\mathbf{m}(t)$ is a linear transformation that represents propagation effects and any signal distortion occurring in the sensor. In the case of ideal (nondispersive) propagation and distortion-free electronics, the elements of $\mathbf{m}(t)$ are time delays, $\delta(t-T_i)$, whereas the scalar function $s(t)$ is the desired signal.

When the signal and noise processes are stationary and the observation interval is long ($t \to \infty$), then the convolution in (3.171) is circumvented by working in the frequency domain using Fourier transform techniques. Taking the Fourier transform of (3.171) yields

$$\mathfrak{x}(\omega) = \mathfrak{m}(\omega)\mathfrak{s}(\omega) + \mathfrak{n}(\omega) \tag{3.172}$$

where the German quantities denote Fourier transformed variables. Note that (3.172) is the same form as (3.147); however, it is a frequency-domain equation, whereas (3.147) is a time-domain equation. The fact that (3.172) is a frequency-domain equation implies that cross-spectral density matrices (which are the Fourier transforms of covariance matrices) now fill the role that covariance matrices formerly played for (3.147).

Now apply the frequency-domain equivalent of (3.160) to obtain the solution

$$\hat{\mathfrak{s}}(\omega) = \left[\phi_{ss}^{-1}(\omega) + \mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)\right]^{-1}\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega) \tag{3.173}$$

Note that the quantity $\phi_{\mathbf{ss}}^{-1}(\omega) + \mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)$ is just a scalar so that (3.173) is rewritten as

$$\mathfrak{s}(\omega) = |\mathfrak{a}(\omega)|^2\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega) \tag{3.174}$$

where

$$|\mathfrak{a}(\omega)|^2 = \frac{\phi_{ss}(\omega)}{1 + \phi_{ss}(\omega)\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)} \tag{3.175}$$

and $\phi_{ss}(\omega)$ is simply the power of $s(t)$ appearing at the frequency ω. In general, the frequency response given by (3.174) is not realizable, and it is necessary to introduce a time delay to obtain a good approximation to the corresponding time waveform $\hat{s}(t)$. Taking the inverse Fourier transform of (3.174) to obtain $\hat{s}(t)$ then yields

$$\hat{s}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} |\mathfrak{a}(\omega)|^2 \mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)e^{j\omega t}d\omega \tag{3.176}$$

Note that (3.174) is rewritten as

$$\hat{\mathfrak{s}}(\omega) = |\mathfrak{a}(\omega)|^2 \mathfrak{h}(\omega)\mathfrak{x}(\omega) \tag{3.177}$$

where

$$\mathfrak{h}(\omega) = \mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega) \tag{3.178}$$

is a $1 \times n$ row vector of individual filters. The single filter $\mathfrak{h}_j(\omega)$ operates on the received signal component $\mathfrak{x}_j(\omega)$ so that it performs the operation of spatial prewhitening and then matching to the known propagation and distortion effects imprinted on the structure of the signal. The processor obtained for this estimation problem therefore employs the principle: first prewhiten, then match. A block diagram of the processor corresponding to (3.174) is shown in Figure 3-7.

3.4.2 Estimation of an Unknown, Nonrandom Signal

Now assume that **s** in (3.147) is a vector of unknown parameters rather than a vector of random variables. The problem, once again, is estimating **s** given the observation vector **x**. An estimate having desirable optimal properties for nonrandom parameters is the maximum likelihood estimate that corresponds to the value of **s** that maximizes the conditional density function $p(\mathbf{x/s})$ for a particular observation **x**.

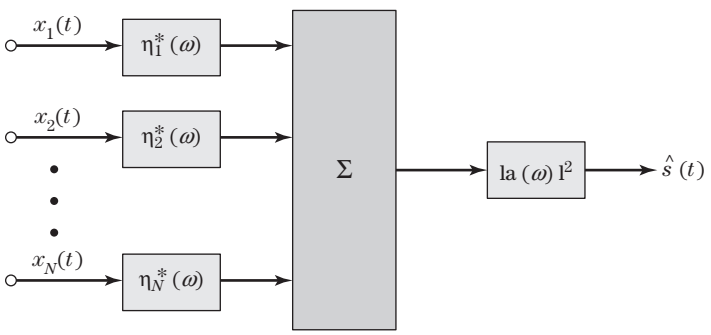


FIGURE 3-7 ■ Optimum array processor for estimation of a random signal.

3.4.2.1 Gaussian Noise Case

When the noise vector $\mathbf{n}$ is Gaussian, the conditional density function takes the form

$$p(\mathbf{x}/\mathbf{s}) = \text{const.}\exp\left\{-\frac{1}{2}\alpha(\mathbf{x}-\mathbf{M}\mathbf{s})^{\dagger}\mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{M}\mathbf{s})\right\} \tag{3.179}$$

Maximizing (3.179) corresponds to minimizing the exponent, and the estimate that minimizes this exponent is easily shown to be given by

$$\hat{\mathbf{s}} = \left[\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.180}$$

It is seen from (3.180) that $\hat{\mathbf{s}}$ is obtained from a linear transformation on $\mathbf{x}$, so the distribution of $\hat{\mathbf{s}}$ is also Gaussian. From (3.180) it immediately follows that since $E\{\mathbf{x}\} = \mathbf{M}\mathbf{s}$, then

$$E\{\hat{\mathbf{s}}\} = \mathbf{s} \tag{3.181}$$

and

$$\text{cov}(\hat{\mathbf{s}}) = \left[\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1} \tag{3.182}$$

It is interesting to note that (3.180) is reached in the limit in (3.156) as $\mathbf{u} \to \mathbf{0}$ and $\mathbf{R}_{ss}^{-1} \to \mathbf{0}$. In other words, the ML estimate corresponds to the estimate obtained with an a priori Gaussian distribution $p(\mathbf{s})$ having zero mean and infinite variance.

3.4.2.2 Non-Gaussian Noise Case

When $\mathbf{n}$ is not Gaussian, the likelihood function will not have the simple form of (3.179); thus, the ML estimate may not be so easily obtained. Furthermore, the available information is limited to second-order statistics. Consequently, a weighted least square estimate is popular to use in these circumstances, and it is obtained by minimizing

$$J = (\mathbf{x}-\mathbf{M}\mathbf{s})^{\dagger}\mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{M}\mathbf{s}) \tag{3.183}$$

The value of $\mathbf{s}$ that minimizes (3.183) is given by (3.180).

Yet another approach uses the minimum variance unbiased linear estimate. A linear estimate has the form

$$\hat{\mathbf{s}} = \mathbf{a} + \mathbf{B}\mathbf{x} \tag{3.184}$$

and an unbiased estimate requires that

$$E\{\hat{\mathbf{s}}\} = \mathbf{s} \tag{3.185}$$

If an estimate is to have minimum variance, then

$$\text{trace}\{E\left[(\hat{\mathbf{s}}-\mathbf{s})(\hat{\mathbf{s}}-\mathbf{s})^{\dagger}\right]\} = E\left\{(\hat{\mathbf{s}}-\mathbf{s})^{\dagger}(\hat{\mathbf{s}}-\mathbf{s})\right\} \tag{3.186}$$

must be minimized. Combining (3.147) and (3.184) results in

$$\hat{\mathbf{s}} = \mathbf{a} + \mathbf{B}\mathbf{M}\mathbf{s} + \mathbf{B}\mathbf{n} \tag{3.187}$$

For (3.185) to hold, it is necessary that

$$\mathbf{a} = \mathbf{0} \tag{3.188}$$

and

$$\mathbf{BM} = \mathbf{I} \tag{3.189}$$

From (3.187) and (3.189), it follows that

$$E\{\hat{\mathbf{s}}\hat{\mathbf{s}}^\dagger\} = \mathbf{s}\mathbf{s}^\dagger + \mathbf{B}\mathbf{R}_{nn}\mathbf{B}^\dagger \tag{3.190}$$

and (3.186) is minimized by choosing $\mathbf{B}$ to minimize the quantity trace $(\mathbf{B}\mathbf{R}_{nn}\mathbf{B}^\dagger)$ subject to the constraint (3.189). This constrained minimization problem is solved by introducing a matrix Lagrange multiplier $\boldsymbol{\lambda}$ and minimizing the quantity

$$J = \text{trace}\{\mathbf{B}\mathbf{R}_{nn}\mathbf{B}^\dagger + [\mathbf{I} - \mathbf{B}\mathbf{M}]\boldsymbol{\lambda} + \boldsymbol{\lambda}^\dagger[\mathbf{I} - \mathbf{M}^\dagger\mathbf{B}^\dagger]\} \tag{3.191}$$

On completing the square of (3.191) using the formula (D.9) in Appendix D, the previous expression is rewritten as

$$\begin{aligned} J = \text{trace}\{&\boldsymbol{\lambda} + \boldsymbol{\lambda}^\dagger - \boldsymbol{\lambda}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\boldsymbol{\lambda} \\ &+ [\mathbf{B} - \boldsymbol{\lambda}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}]\mathbf{R}_{nn}[\mathbf{B}^\dagger - \mathbf{R}_{nn}^{-1}\mathbf{M}\boldsymbol{\lambda}]\} \end{aligned} \tag{3.192}$$

From (3.192), it follows that the minimizing value of $\mathbf{B}$ is given by

$$\mathbf{B} = \boldsymbol{\lambda}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1} \tag{3.193}$$

To eliminate $\boldsymbol{\lambda}$, use (3.189) so that

$$\boldsymbol{\lambda}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M} = \mathbf{I} \tag{3.194}$$

or

$$\boldsymbol{\lambda}^\dagger = [\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}]^{-1} \tag{3.195}$$

Hence

$$\mathbf{B} = [\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}]^{-1}\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1} \tag{3.196}$$

Consequently, the estimate (3.184) once again reduces to (3.180), and the mean and covariance of the estimate are given by (3.181) and (3.182), respectively.

3.4.2.3 Application of Unknown, Nonrandom Signal Estimation Results to Optimum Array Processing

Again, using the frequency-domain equation (3.172) it follows by analogy with (3.147) and (3.180) that the optimum frequency-domain estimator is given by

$$\mathfrak{s}(\omega) = [\mathfrak{m}^\dagger(\omega)\boldsymbol{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)]^{-1}\mathfrak{m}^\dagger(\omega)\boldsymbol{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega) \tag{3.197}$$

Nothing that $\mathfrak{m}^\dagger(\omega)\boldsymbol{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)$ is a scalar, (3.197) is written as

$$\mathfrak{s}(\omega) = \left[\frac{1}{\phi_{ss}(\omega)\mathfrak{m}^\dagger(\omega)\boldsymbol{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)} + 1\right] \cdot |\mathfrak{a}(\omega)|^2\mathfrak{h}(\omega)\mathfrak{x}(\omega) \tag{3.198}$$

where $|\mathfrak{a}(\omega)|^2$ and $\mathfrak{h}(\omega)$ are given by (3.175) and (3.178), respectively. Therefore, the only difference between the optimum array processor for random signal estimation and

for unknown, nonrandom signal estimation is the presence of an additional scalar transfer function. The characteristic prewhitening and matching operation represented by $\mathfrak{h}(\omega) = \mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)$ is still required.

3.4.3 Detection of a Known Signal

The binary detection problem (i.e., whether the signal is present) leads to a likelihood ratio test in which the likelihood ratio given by (3.143) is compared with a threshold η.

3.4.3.1 Gaussian Noise Case

When the noise is Gaussian, then the likelihood ratio (3.143) is written as

$$\Lambda(\mathbf{x}) = \frac{\exp\left\{-\frac{1}{2}\alpha(\mathbf{x}-\mathbf{Ms})^{\dagger}\mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{Ms})\right\}}{\exp\left\{-\frac{1}{2}\alpha\left(\mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}\right)\right\}} \tag{3.199}$$

where $\alpha = 2$ for complex $\mathbf{x}$ and $\alpha = 1$ for real $\mathbf{x}$. Clearly (3.199) is written in terms of a single exponential function so that

$$\Lambda(\mathbf{x}) = \exp\left\{-\frac{1}{2}\alpha\left[\mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms} - \mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} - \mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms}\right]\right\} \tag{3.200}$$

Since the term $\mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms}$ does not depend on any observation of $\mathbf{x}$, a sufficient test statistic for making a decision is the variable

$$y = \frac{1}{2}\left(\mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} + \mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms}\right) = \mathrm{Re}\left\{\mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}\right\} \tag{3.201}$$

The factor α is assumed to be incorporated into the threshold level setting. The distribution of the sufficient statistic y is Gaussian, since it results from a linear operation on $\mathbf{x}$, which in turn is Gaussian both when the signal is present and when it is absent.

When the signal is absent, then

$$E\{y\} = 0 \tag{3.202}$$

and from (E.7) and (E.41) of Appendix E, it follows that

$$\mathrm{var}(y) = \frac{\mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms}}{\alpha} \tag{3.203}$$

Likewise, when the signal is present, then

$$E\{y\} = \mathbf{s}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{Ms} \tag{3.204}$$

and the variance of y is the same as when the signal is absent.

3.4.3.2 Non-Gaussian Noise Case

In the event that the noise vector $\mathbf{n}$ is non-Gaussian and only second-order statistics are available, a linear transformation of the observation vector $\mathbf{x}$ is commonly sought that is of the form

$$y = \frac{1}{2}[\mathbf{k}^{\dagger}\mathbf{x} + \mathbf{x}^{\dagger}\mathbf{k}] = \mathrm{Re}\{\mathbf{k}^{\dagger}\mathbf{x}\} \tag{3.205}$$

where the vector $\mathbf{k}$ is selected so the output SNR is maximized. The ratio given by

$$r_0 = \frac{\text{change in mean-squared output due to signal presence}}{\text{mean-squared output for noise alone}}$$

or equivalently

$$r_0 = \frac{E\{y^2/\text{signal present}\} - E\{y^2/\text{signal absent}\}}{E\{y^2/\text{signal absent}\}} \tag{3.206}$$

is referred to as the signal-to-noise ratio. Using (3.205) and (E.54) of Appendix E, the ratio is written as

$$r_0 = \frac{\alpha[\text{Re}\{\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\}]^2}{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}} \leq \frac{\alpha\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{k}}{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}} \tag{3.207}$$

It is convenient to factor $\mathbf{R}_{nn}$ by introducing

$$\mathbf{R}_{nn} = \mathbf{T}^\dagger\mathbf{T} \tag{3.208}$$

in which case the upper bound for r_0 given in (3.207) is written as

$$\frac{\alpha\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{k}}{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}} = \frac{\alpha\mathbf{k}^\dagger\mathbf{T}^\dagger(\mathbf{T}^\dagger)^{-1}\mathbf{M}\mathbf{s}\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{T}^{-1}\mathbf{T}\mathbf{k}}{\mathbf{k}^\dagger\mathbf{T}^\dagger\mathbf{T}\mathbf{k}} \tag{3.209}$$

With the upper bound written as in (3.209), (D.16) of Appendix D is applied and the resulting terms rearranged to yield

$$r_0 = \frac{\alpha[\text{Re}\{\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\}]^2}{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}} \leq \alpha\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{s} \tag{3.210}$$

where equality results when

$$\mathbf{k}^\dagger = \mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1} \tag{3.211}$$

On substituting $\mathbf{k}^\dagger$ of (3.211) into (3.205), it is found that the test statistic is once again given by (3.201).

A different approach to the problem of detecting a known signal embedded in non-Gaussian noise is to maximize the detection index given by (3.146). For y given by (3.205), it is easily shown that

$$d = \sqrt{\alpha}\,\frac{\text{Re}\{\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\}}{\sqrt{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}}} \leq \frac{\sqrt{\alpha\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{k}}}{\sqrt{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}}} \tag{3.212}$$

Now applying (D.17) of Appendix D to the upper bound in (3.212) in the same manner as (D.16) was applied to the upper bound in (3.209), the detection index is shown to satisfy

$$d = \sqrt{\alpha}\,\frac{\text{Re}\{\mathbf{k}^\dagger\mathbf{M}\mathbf{s}\}}{\sqrt{\mathbf{k}^\dagger\mathbf{R}_{nn}\mathbf{k}}} \leq \sqrt{\alpha\mathbf{s}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{s}} \tag{3.213}$$

Equality results in (3.213) when (3.211) is satisfied, so (3.201) once again results for the test statistic.

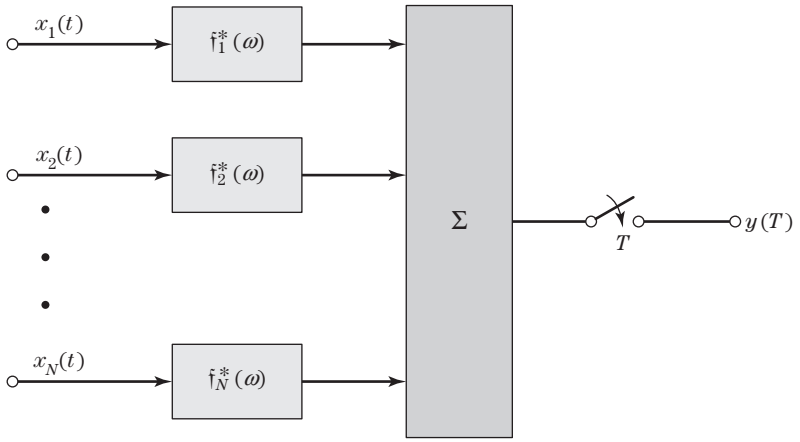


FIGURE 3-8 ■ Linear processor structure for known signal detection problem.

3.4.3.3 Application of Known Signal Detection Results to Optimum Array Processing

When the received signal is known and expressed by (3.171), maximizing the detection index at a specified observation time, T, is best done using the frequency domain in (3.172) and applying an appropriate form of the Schwartz inequality to the detection index. With the linear processor specified in Figure 3-8, then

$$E\{y(T)/\text{signal present}\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathfrak{m}(\omega)\mathfrak{s}(\omega)e^{j\omega T}d\omega \tag{3.214}$$

and

$$\text{var}[y(T)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathbf{\Phi}_{nn}(\omega)\mathfrak{f}(\omega)d\omega \tag{3.215}$$

Since $E\{y(T)/\text{signal absent}\} = 0$, the detection index becomes

$$d = \frac{1/\sqrt{2\pi}\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathfrak{m}(\omega)\mathfrak{s}(\omega)e^{j\omega T}d\omega}{\left[\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathbf{\Phi}_{nn}(\omega)\mathfrak{f}(\omega)d\omega\right]^{1/2}} \tag{3.216}$$

Now let $\mathbf{\Phi}_{nn}(\omega) = \mathfrak{T}^{\dagger}(\omega)\mathfrak{T}(\omega)$, so (3.216) is rewritten as

$$d\sqrt{2\pi} = \frac{\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathfrak{T}^{\dagger}(\omega)[\mathfrak{T}^{\dagger}(\omega)]^{-1}\mathfrak{m}(\omega)\mathfrak{s}(\omega)e^{j\omega T}d\omega}{\left[\int_{-\infty}^{\infty} \mathfrak{f}^{\dagger}(\omega)\mathfrak{T}^{\dagger}(\omega)\mathfrak{T}(\omega)\mathfrak{f}(\omega)d\omega\right]^{1/2}} \tag{3.217}$$

The Schwartz inequality in the form of (D.14) of Appendix D may be applied to (3.217) by letting $\mathfrak{f}^{\dagger}(\omega)\mathfrak{T}^{\dagger}(\omega)$ play the role of $\mathbf{f}^{\dagger}$ and $[\mathfrak{T}^{\dagger}(\omega)]^{-1}\mathfrak{m}(\omega)\mathfrak{s}(\omega)e^{j\omega T}$ play the role of $\mathbf{g}$. It then follows that

$$d\sqrt{2\pi} \le \left[\int_{-\infty}^{\infty} \mathfrak{s}^{*}(\omega)\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)\mathfrak{s}(\omega)d\omega\right]^{1/2} \tag{3.218}$$

where equality occurs if and only if

$$\mathfrak{f}^{\dagger}(\omega) = e^{-j\omega T}\mathfrak{s}*(\omega)\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega) \tag{3.219}$$

Once again, the usual spatial prewhitening and matching operation represented by $\mathfrak{m}^\dagger(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)$ appears in the optimum processor.

3.4.4 Detection of a Random Signal

When $\mathbf{s}$ is a random signal vector it will again be assumed that the mean value $\mathbf{u}$ and the covariance matrix $\mathbf{R}_{ss}$ are both known. The optimum processor for the binary detection problem again leads to a likelihood ratio test.

3.4.4.1 Gaussian Case

When the vectors $\mathbf{s}$ and $\mathbf{n}$ are both Gaussian, the likelihood ratio (3.143) is written as

$$\Lambda(\mathbf{x}) = \text{const.}\frac{\exp\{-(\alpha/2)(\mathbf{x}-\mathbf{Mu})^\dagger\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^\dagger + \mathbf{R}_{nn}\right]^{-1}(\mathbf{x}-\mathbf{Mu})\}}{\exp\{-(\alpha/2)\mathbf{x}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{x}\}} \tag{3.220}$$

where "const." represents a constant of proportionality. Expanding (3.220) by carrying out indicated multiplications and taking logarithms yields the following sufficient test statistic:

$$\begin{aligned} y = {} & \frac{1}{2}(\mathbf{x}-\mathbf{Mu})^\dagger\{\mathbf{R}_{nn}^{-1} - \left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^\dagger + \mathbf{R}_{nn}\right]^{-1}\}(\mathbf{x}-\mathbf{Mu}) \\ & + \frac{1}{2}\mathbf{u}^\dagger\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{u} \end{aligned} \tag{3.221}$$

where the factor α has again been incorporated into the threshold level setting.

Thinking of the known mean value $\mathbf{u}$ as the deterministic part of $\mathbf{s}$ and the deviation of the observation from its mean $\mathbf{x}-\mathbf{Mu}$ as the random part of the observation, we then see that the test statistic (3.221) is composed of a linear term corresponding to the test statistic in the known signal case [represented by (3.201)] and a quadratic term involving only the random part of the observation. Since the known signal case was treated in Section 3.4.3, only the random part of the problem will be considered here by assuming that the mean has been subtracted out or, equivalently, by assuming that $\mathbf{s}$ has zero mean.

When $\mathbf{u} = 0$, then (3.221) reduces to

$$y = \frac{1}{2}\mathbf{x}^\dagger\{\mathbf{R}_{nn}^{-1} - \left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^\dagger + \mathbf{R}_{nn}\right]^{-1}\}\mathbf{x} \tag{3.222}$$

Applying the matrix identities (D.11) and (D.12) from Appendix D transforms (3.222) as follows:

$$y = \frac{1}{2}\mathbf{x}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\left[\mathbf{R}_{ss}^{-1} + \mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.223}$$

or

$$y = \frac{1}{2}\mathbf{x}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^\dagger\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^\dagger + \mathbf{R}_{nn}\right]^{-1}\mathbf{x} \tag{3.224}$$

A special case of some practical importance arises when $\mathbf{s}$ is only a scalar s and $\mathbf{M}$ is a column vector $\mathbf{m}$. Then (3.223) becomes

$$2y = \frac{\left|\mathbf{x}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{m}\right|^2}{\phi_{ss} + \mathbf{m}^\dagger\mathbf{R}_{nn}^{-1}\mathbf{m}} \tag{3.225}$$

For small signals

$$\left[\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger} + \mathbf{R}_{nn}\right]^{-1} \approx \mathbf{R}_{nn}^{-1} \tag{3.226}$$

In this case, (3.224) becomes

$$y = \frac{1}{2}\mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.227}$$

3.4.4.2 Gaussian Noise, Non-Gaussian Signal

Whereas the sufficient test statistic for the known signal detection problem involved only a linear operation on the observation vector, the random signal detection problem of the preceding section involved a test statistic that was quadratic in the observation vector. Likewise, when **s** is non-Gaussian and only second-order statistics are available, then the "best" quadratic processor is found (in the sense that the detection index is maximized). When we assume **s** has zero mean, the test statistic is presumed to have the form

$$y = \mathbf{x}^{\dagger}\mathbf{K}\mathbf{K}^{\dagger}\mathbf{x} \tag{3.228}$$

where **K** maximizes the detection index defined by (3.146). Note that since y is quadratic in **x** and the variance of y in the denominator of (3.146) involves $E\{y^2/\text{signal absent}\}$, then fourth-order moments of the distribution for **x** are involved. By assuming the noise field is Gaussian, the required fourth-order moments are expressed in terms of the covariance matrix by applying (E.51) of Appendix E.

The numerator of (3.146) when y is given by (3.228) is written as

$$E\{y/\text{signal present}\} - E\{y/\text{signal absent}\} = \text{trace}[\mathbf{K}^{\dagger}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{K}] \tag{3.229}$$

By use of (E.51), the denominator of (3.146) becomes

$$\sqrt{\text{var}(y/\text{signal absent})} = \sqrt{\text{trace}[(\mathbf{K}^{\dagger}\mathbf{R}_{nn}\mathbf{K})^2] \cdot \frac{2}{\alpha}} \tag{3.230}$$

Consequently

$$d = \frac{\sqrt{(\alpha/2)}\ \text{trace}(\mathbf{K}^{\dagger}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{K})}{\sqrt{\text{trace}[(\mathbf{K}^{\dagger}\mathbf{R}_{nn}\mathbf{K})^2]}} \tag{3.231}$$

It is now desired to maximize d by applying the Schwartz inequality to obtain an upper bound. It is again convenient to introduce $\mathbf{R}_{nn} = \mathbf{T}^{\dagger}\mathbf{T}$ before applying (D.16) so that

$$\begin{aligned} d &= \frac{\sqrt{(\alpha/2)}\ \text{trace}\left[(\mathbf{T}^{\dagger})^{-1}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{T}^{-1}\mathbf{T}\mathbf{K}\mathbf{K}^{\dagger}\mathbf{T}^{\dagger}\right]}{\sqrt{\text{trace}\{(\mathbf{T}\mathbf{K}\mathbf{K}^{\dagger}\mathbf{T}^{\dagger})^2\}}} \\ &\le \sqrt{(\alpha/2)\ \text{trace}\left\{\left(\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\right)^2\right\}} \end{aligned} \tag{3.232}$$

Equality obtains in (3.232) when

$$\mathbf{K}^{\dagger} = \mathbf{A}^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1} \tag{3.233}$$

where

$$\mathbf{R}_{ss} = \mathbf{A}\mathbf{A}^{\dagger} \tag{3.234}$$

Substituting (3.233) into (3.228) then yields the test statistic

$$y = \mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\mathbf{R}_{ss}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.235}$$

which is identical to the test statistic (3.227) obtained from the likelihood ratio in the small signal case.

3.4.4.3 Application of Random Signal Detection Results to Optimum Array Processing

Starting with the frequency-domain equation (3.172), (3.223) leads to a sufficient test statistic is given by

$$\begin{aligned}\mathfrak{y}(\omega) &= \frac{1}{2}|\mathfrak{a}(\omega)|^2\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right]^{\dagger}\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right] \\ &= \frac{1}{2}\left|\mathfrak{a}(\omega)\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right|^2\end{aligned} \tag{3.236}$$

where $|\mathfrak{a}(\omega)|^2$ is given by (3.175). The structure of the optimum processor corresponding to (3.176) is shown in Figure 3-9, where Parseval's theorem was invoked to write (3.236) in the time domain

$$y(T) = \frac{1}{2T}\int_0^T |z(t)|^2 dt \tag{3.237}$$

If the small signal assumption is applicable, then using (3.227) instead of (3.223) leads to

$$\mathfrak{y}(\omega) = \frac{1}{2}\phi_{ss}(\omega)\left|\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right|^2 \tag{3.238}$$

The result expressed by (3.228) is different from (3.236) only in that the $\phi_{ss}(\omega)$ has replaced the scalar factor $|\mathfrak{a}(\omega)|^2$ in (3.236).

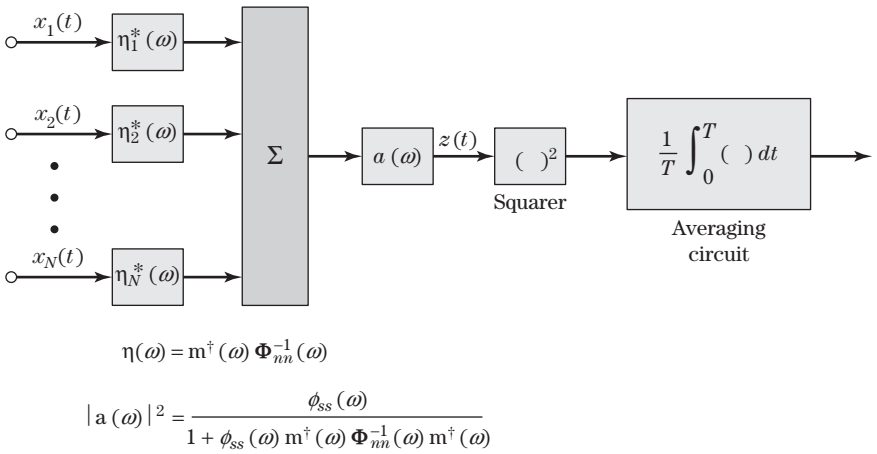


FIGURE 3-9 ■ Optimum array processor for detection of a random signal.

3.4.5 Detection of an Unknown, Nonrandom Signal

When $\mathbf{s}$ is unknown but nonrandom, good results are obtained from the GLR test. The procedure is to form the ratio

$$\Lambda_G(\mathbf{x}) = \frac{p(\mathbf{x}/\hat{\mathbf{s}}, \text{signal present})}{p(\mathbf{x}/\text{signal absent})} \tag{3.239}$$

where $\hat{\mathbf{s}}$ is selected to maximize the conditional density function $p(\mathbf{x}/\mathbf{s}, \text{signal present})$ and is the maximum likelihood estimate of $\mathbf{s}$.

3.4.5.1 Gaussian Noise Case

When $\mathbf{n}$ is Gaussian the likelihood ratio is given by

$$\frac{p(\mathbf{x}/\mathbf{s}, \text{signal present})}{p(\mathbf{x}/\text{signal absent})} = \frac{\exp\left\{-(\alpha/2)(\mathbf{x}-\mathbf{M}\mathbf{s})^{\dagger}\mathbf{R}_{nn}^{-1}(\mathbf{x}-\mathbf{M}\mathbf{s})\right\}}{\exp\left\{-(\alpha/2)\mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}\right\}} \tag{3.240}$$

Substituting the likelihood estimate of $\hat{\mathbf{s}}$ in (3.180) into (3.240) yields the generalized likelihood ratio

$$\Lambda_G(\mathbf{x}) = \exp\left\{\frac{\alpha}{2}\mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\left[\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x}\right\} \tag{3.241}$$

The sufficient test statistic for (3.241) is obviously

$$y = \frac{1}{2}\mathbf{x}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\left[\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.242}$$

or

$$y = \frac{1}{2}\hat{\mathbf{s}}_1^{\dagger}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.243}$$

where

$$\hat{\mathbf{s}}_1 = \left[\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^{\dagger}\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.244}$$

Setting $\mathbf{R}_{ss}^{-1} = \mathbf{0}$ in (3.223) for the case of a random signal vector reduces (3.223) to (3.242).

3.4.5.2 Application of Unknown, Nonrandom Signal Detection Results to Optimum Array Processing

Again, using the frequency-domain equation (3.172), it follows from (3.242) that a sufficient test statistic is given by

$$\begin{aligned}\mathfrak{y}(\omega) &= \frac{1}{2}\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)\right]^{-1}\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right]^{\dagger}\cdot\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right] \\ &= \frac{1}{2}\left[\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{m}(\omega)\right]^{-1}\left|\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)\mathfrak{x}(\omega)\right|^2\end{aligned} \tag{3.245}$$

The result expressed by (3.245) reveals the characteristic prewhitening and matching operator $\mathfrak{m}^{\dagger}(\omega)\mathbf{\Phi}_{nn}^{-1}(\omega)$.

3.4.6 Array Gain

The array gain is defined as the ratio of the output signal-to-noise spectral ratio to the input signal-to-noise spectral ratio. The linear processor structure of Figure 3-8 consists of a set of filters (one for each sensor), followed by a summation device for which the array gain is given by [17]

$$G(\omega) = \frac{\mathfrak{f}^{\dagger}(\omega)\tilde{\mathbf{\Phi}}_{ss}(\omega)\mathfrak{f}(\omega)}{\mathfrak{f}^{\dagger}(\omega)\tilde{\mathbf{\Phi}}_{nn}(\omega)\mathfrak{f}(\omega)} \tag{3.246}$$

where $\tilde{\mathbf{\Phi}}_{ss}(\omega)$ and $\tilde{\mathbf{\Phi}}_{nn}(\omega)$ represent the normalized cross-spectral density matrices of the signal vector and noise vector, that is,

$$\mathbf{\Phi}_{ss}(\omega) = \sigma_s^2(\omega)\tilde{\mathbf{\Phi}}_{ss}(\omega) \tag{3.247}$$

and

$$\mathbf{\Phi}_{nn}(\omega) = \sigma_n^2(\omega)\tilde{\mathbf{\Phi}}_{nn}(\omega) \tag{3.248}$$

where $\sigma_n^2(\omega)$ is the noise power spectral density averaged over the N sensors so that

$$\sigma_n^2(\omega) = \frac{1}{N}\text{trace}[\mathbf{\Phi}_{nn}(\omega)] \tag{3.249}$$

and

$$\sigma_s^2(\omega) = \frac{1}{N}\text{trace}[\mathbf{\Phi}_{ss}(\omega)] \tag{3.250}$$

The array gain corresponding to (3.246) is the ratio of the output signal-to-noise spectral ratio to the input signal-to-noise spectral ratio.

Whenever the signal vector $\mathbf{s}(t)$ is related to a scalar signal $s(t)$ by a known transformation $\mathbf{m}(t)$ such that

$$\mathbf{s}(t) = \int_{-\infty}^{t} \mathbf{m}(t-\tau)s(\tau)d\tau \tag{3.251}$$

Then $\tilde{\mathbf{\Phi}}_{ss}(\omega)$ is simply given by the dyad [17]

$$\tilde{\mathbf{\Phi}}_{ss}(\omega) = \mathfrak{m}(\omega)\mathfrak{m}^{\dagger}(\omega) \tag{3.252}$$

where $\mathfrak{m}(\omega)$ denotes the normalized Fourier transform of $\mathbf{m}(t)$ so that

$$\mathfrak{m}^{\dagger}(\omega)\mathfrak{m}(\omega) = N \tag{3.253}$$

When $\tilde{\mathbf{\Phi}}_{ss}(\omega)$ is given by (3.252), then the array gain becomes

$$G(\omega) = \frac{|\mathfrak{f}^{\dagger}(\omega)\mathfrak{m}(\omega)|^2}{\mathfrak{f}^{\dagger}(\omega)\tilde{\mathbf{\Phi}}_{nn}(\omega)\mathfrak{f}(\omega)} \tag{3.254}$$

The quantity $\mathfrak{f}^{\dagger}(\omega)\mathfrak{m}(\omega)$ may be regarded as the inner product of the beam steering vector $\mathfrak{f}$ and the signal direction $\mathfrak{m}$ for plane wave propagation. Define a generalized angle γ

between $\mathbf{f}$ and $\mathbf{m}$ as described in Appendix F such that

$$\cos^2(\gamma) = \frac{|\mathbf{f}^\dagger(\omega)\mathbf{m}(\omega)|^2}{(\mathbf{f}^\dagger(\omega)\mathbf{f}(\omega))(\mathbf{m}^\dagger(\omega)\mathbf{m}(\omega))} \tag{3.255}$$

In a conventional beamformer the vector $\mathbf{f}$ is chosen to be proportional to $\mathbf{m}$, thus making γ equal to zero and "matching to the signal direction." This operation also maximizes the array gain against spatially white noise as shown subsequently.

Substituting (3.252) into (3.246) and using $\mathbf{g} = \tilde{\mathbf{\Phi}}_{nn}^{\frac{1}{2}}\mathbf{f}$ yields

$$G(\omega) = \frac{\left|\mathbf{g}^\dagger(\omega)\tilde{\mathbf{\Phi}}_{nn}^{-\frac{1}{2}}(\omega)\mathbf{m}(\omega)\right|^2}{\mathbf{g}^\dagger(\omega)\mathbf{g}(\omega)} \tag{3.256}$$

Applying the Schwartz inequality (D.14) to (3.256) then gives

$$G(\omega) \leq \mathbf{m}^\dagger(\omega)\tilde{\mathbf{\Phi}}_{nn}^{-1}(\omega)\mathbf{m}(\omega) \tag{3.257}$$

where equality results when the vector $\mathbf{f}^\dagger$ be a scalar multiple of $\mathbf{m}^\dagger\tilde{\mathbf{\Phi}}_{nn}^{-1}$. Therefore, maximizing the array gain with no constraints yields the same prewhitening and matching operation found in the preceding sections for a variety of detection and estimation problems. Note also that this processor reduces to the conventional direction matching beamformer when the noise field is uncorrelated from sensor to sensor so that $\tilde{\mathbf{\Phi}}_{nn} = \mathbf{I}$, the identity matrix [45].

Maximizing the array gain is closely related to minimizing the array output signal variance under a desired signal response constraint, because this is completely equivalent to minimizing the denominator of (3.246) subject to a constraint on the numerator. Since $G(\omega)$ is not changed by any scaling of the vector $\mathbf{f}$, introducing a constraint like $\mathbf{f}^\dagger\mathbf{m} = \beta$ does not affect $G(\omega)$ and merely determines the scalar multiple used in selecting $\mathbf{f}$. Consequently, selecting $\mathbf{f} = \beta\tilde{\mathbf{\Phi}}_{nn}^{-1}\mathbf{m}/(\mathbf{m}^\dagger\tilde{\mathbf{\Phi}}_{nn}^{-1}\mathbf{m})$ maximizes $G(\omega)$ as well as satisfies the constraint $\mathbf{f}^\dagger\mathbf{m} = \beta$. Maximizing the array gain and minimizing the output signal distortion yield exactly the same filters for monochromatic signals [3]. The filter that yields the maximum likelihood estimate and the Wiener filter are different by only a scalar transfer function [46]. Furthermore, the likelihood ratio processor, the maximum SNR filter, and the Wiener filter are known to be equivalent in the case of a narrowband signal that is known except for phase and corrupted by additive, Gaussian noise [47].

3.4.7 Criterion Invariant Array Processor

Various detection and estimation techniques and performance measures related to optimum array processing were treated in the preceding sections by means of a unified theory. The likelihood ratio test for optimum signal detection, several signal estimation problems, and various performance criteria are all related by a prewhitening and matching operation that defines a vector operator for the received signal vector that has a scalar output. This scalar output is a single waveform that is then processed by different scalar operators depending on the problem of concern.

The results obtained for the various classes of problems may be conveniently summarized in the criterion invariant processor shown in Figure 3-10. This figure illustrates that the principle "*first prewhiten, then match*," represented by the operator $\mathbf{m}^\dagger\mathbf{\Phi}_{nn}^{-1}$, is common for a wide variety of different optimization problems. Since the optimum

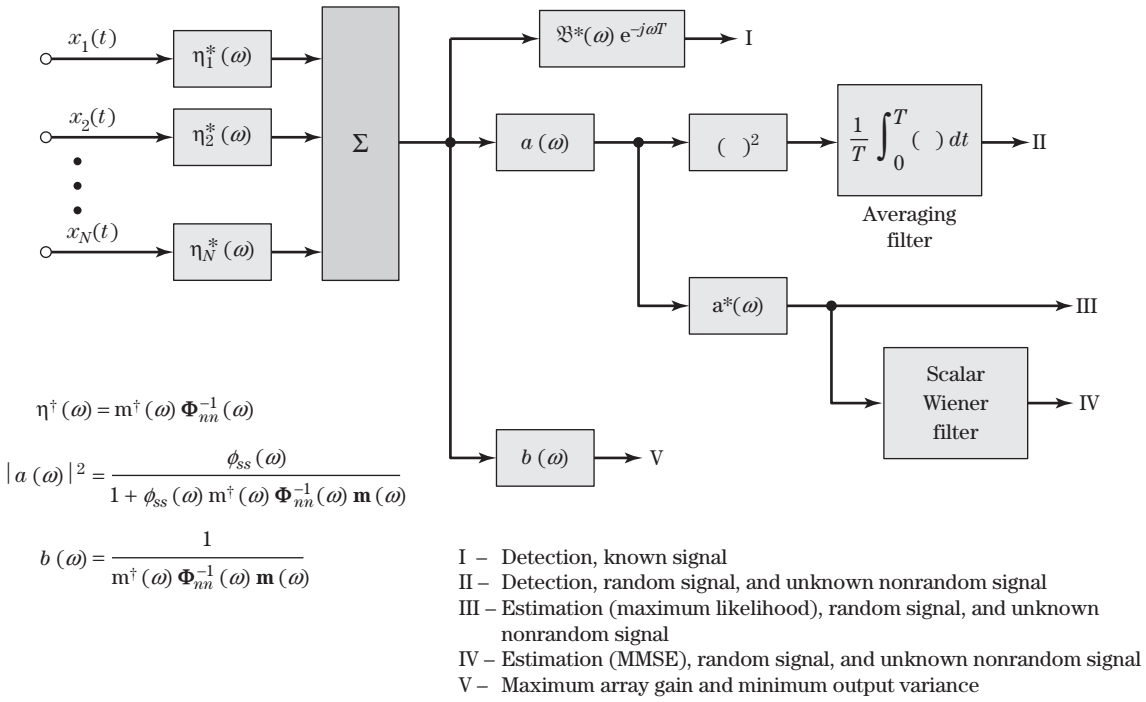


FIGURE 3-10 ■ Criterion invariant array processor for broadband applications.

processor depends on the inverse of the noise cross-spectral density matrix, but in practice the only measurable quantity is the cross-spectral matrix of the sensor outputs (which in general contains desired signal-plus-noise components), the use of the signal-plus-noise spectral matrix may result in performance degradation unless provision is made to obtain a signal-free estimate of the noise cross-spectral matrix or the use of the signal-plus-noise spectral matrix is specifically provided for. The consequences involved for providing an inexact prewhitening and matching operation are discussed in reference [45]. It may be further noted that the minimum mean square error (MMSE) signal estimate is different from the maximum likelihood (distortionless) estimate (or any other estimate) only by a scalar Wiener filter. A Wiener processor is therefore regarded as forming the undistorted signal estimate for observational purposes before introducing the signal distortion resulting from the scalar Wiener filter. The nature of the scalar Wiener filter is further considered in the Problems section.

3.5 | OPTIMUM ARRAY PROCESSING FOR PERTURBED PROPAGATION CONDITIONS

The processor structure of Figure 3-8 is optimum only when the signal cross-spectral density matrix is a simple dyad [17]

$$\mathbf{\Phi}_{ss}(\omega) = \mathfrak{m}(\omega)\mathfrak{m}^\dagger(\omega) \tag{3.258}$$

Perturbations in the propagation process destroys the dyad nature of $\mathbf{\Phi}_{ss}(\omega)$, so the processor structure must be more general than that of Figure 3-8. To determine the nature of a more general processor structure, the optimum processor for a noncoherent narrowband signal will be found.

Consider the problem of detecting a signal that is narrowband with unknown amplitude and phase. Such conditions frequently occur when the received signal undergoes unknown amplitude and phase changes during propagation. When the signal is present, the received waveform is expressed as

$$x(t) = \text{Re}\{ae^{j\theta}r(t)e^{j\phi(t)}e^{j\omega_0 t}\} + n(t) \tag{3.259}$$

where

a = unknown amplitude
θ = unknown phase
$r(t)$ = known amplitude modulation
$\phi(t)$ = known phase modulation
ω_0 = known carrier frequency

Using real notation we can rewrite (3.259) as

$$x(t) = m_1(t)s_1 + m_2(t)s_2 + n(t) \tag{3.260}$$

where

$$s_1 = a\cos\theta$$
$$s_2 = -a\sin\theta$$
$$m_1(t) = r(t)\cos[\omega_0 t + \phi(t)] = f(t)\cos(\omega_0 t) - g(t)\sin(\omega_0 t) \tag{3.261}$$
$$m_2(t) = r(t)\sin[\omega_0 t + \phi(t)] = f(t)\sin(\omega_0 t) + g(t)\cos(\omega_0 t) \tag{3.262}$$

and

$$f(t) = r(t)\cos[\phi(t)] \tag{3.263}$$
$$g(t) = r(t)\sin[\phi(t)] \tag{3.264}$$

where

$$r(t) = \sqrt{f^2(t) + g^2(t)}$$
$$\phi(t) = \tan^{-1}\left\{\frac{g(t)}{f(t)}\right\}$$

Equation (3.259) may now be rewritten as

$$x = \mathbf{ms} + n \tag{3.265}$$

where

$$\mathbf{m} = [m_1 m_2], \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \quad \text{and } \text{var}(n) = \sigma_n^2$$

The results of Section 3.4.5 for the detection of an unknown, nonrandom signal may now be applied, for which the sufficient test statistic is given by (3.242). For real variables (3.242) becomes

$$y = \frac{1}{2}\mathbf{x}^T\mathbf{R}_{nn}^{-1}\mathbf{M}\left[\mathbf{M}^T\mathbf{R}_{nn}^{-1}\mathbf{M}\right]^{-1}\mathbf{M}^T\mathbf{R}_{nn}^{-1}\mathbf{x} \tag{3.266}$$

Note that $m_1(t)$ and $m_2(t)$ are orthogonal functions having equal energy so that $|m_1|^2 = |m_2|^2$, and $m_1(t) \cdot m_2(t) = 0$. Consequently, when the functions are sampled in time to form $\mathbf{x}^T = [x(t_1), x(t_2), \dots]$, $\mathbf{m}_i^T = [m_i(t_1), m_i(t_2), \dots]$, then (3.266) for the test statistic is rewritten as

$$\begin{aligned} y &= \frac{1}{2}\mathbf{x}^2\left[\mathbf{m}_1^T\mathbf{m}_2^T\right]\begin{bmatrix}\mathbf{m}_1\\ \mathbf{m}_2\end{bmatrix}\left(|\mathbf{m}_1|^2\sigma_n^2\right)^{-1} \\ &= \frac{1}{2}\left(\sigma_n^2|\mathbf{m}_1|^2\right)^{-1}\left\{\left(\mathbf{m}_1^T\mathbf{x}\right)^2 + \left(\mathbf{m}_2^T\mathbf{x}\right)^2\right\} \end{aligned} \tag{3.267}$$

Since the scalar factor $\frac{1}{2}\left(\sigma_n^2|\mathbf{m}_1|^2\right)^{-1}$ in (3.267) is incorporated into the threshold setting for the likelihood ratio test, a suitable test statistic is

$$z = \left(\mathbf{m}_1^T\mathbf{x}\right)^2 + \left(\mathbf{m}_2^T\mathbf{x}\right)^2 \tag{3.268}$$

For a continuous time observation, the test statistic given by (3.268) is rewritten as

$$z = \left[\int_0^T m_1(t)x(t)dt\right]^2 + \left[\int_0^T m_2(t)x(t)dt\right]^2 \tag{3.269}$$

The test statistic represented by (3.269) is conveniently expressed in terms of the "sine" and "cosine" components, $f(t)$ and $g(t)$. Using (3.261) and (3.262), we then can rewrite (3.269) as shown.

$$\begin{aligned} z = &\left[\int_0^T f(t)\cos(\omega_0 t)x(t)dt - \int_0^T g(t)\sin(\omega_0 t)x(t)dt\right]^2 \\ &+ \left[\int_0^T f(t)\sin(\omega_0 t)x(t)dt + \int_0^T g(t)\cos(\omega_0 t)x(t)dt\right]^2 \end{aligned} \tag{3.270}$$

The previous test statistic suggests the processor shown in Figure 3-11.

The optimum detector structure for a noncoherent signal shown in Figure 3-11 leads to the more general array processor structure illustrated in Figure 3-12. This more general processor structure is appropriate when propagating medium or receiving mechanism perturbations cause the plane wave desired signal assumption to hold no longer or when it is desired to match the processor to a signal of arbitrary covariance structure. A matched array processor like that of Figure 3-12 using matrix weighting is referred to as an element space-matched array processor [48].

Regarding the quantity **Ms** in (3.147) as the signal having arbitrary characteristics, then it is appropriate to define an array gain in terms of the detection index at the output of a general quadratic processor [17]. For Gaussian noise, the results summarized in (3.231) may be used to give

$$G = \frac{\text{trace}\left[\mathfrak{R}^\dagger(\omega)\mathbf{\Phi}_{ss}(\omega)\mathfrak{R}(\omega)\right]}{\left\{\text{trace}\left[\left(\mathfrak{R}^\dagger(\omega)\mathbf{\Phi}_{nn}(\omega)\mathfrak{R}(\omega)\right)^2\right]\right\}^{1/2}} \tag{3.271}$$

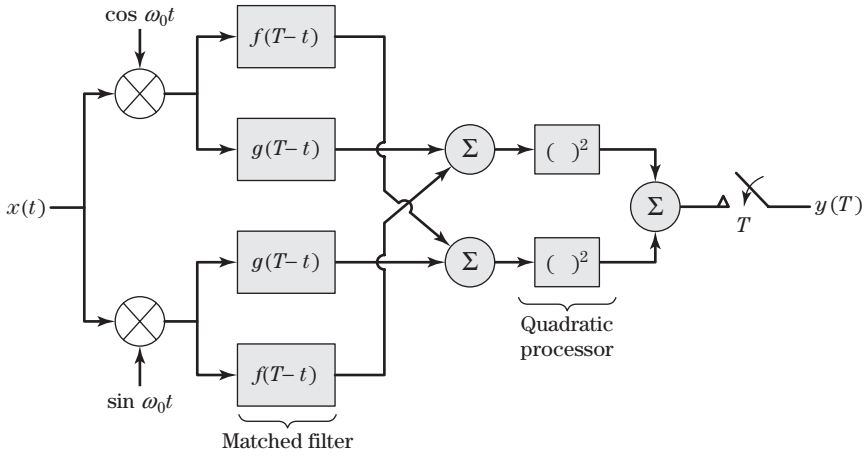


FIGURE 3-11 ■ Quadrature matched filter and envelope detector for received waveform having unknown amplitude and phase.

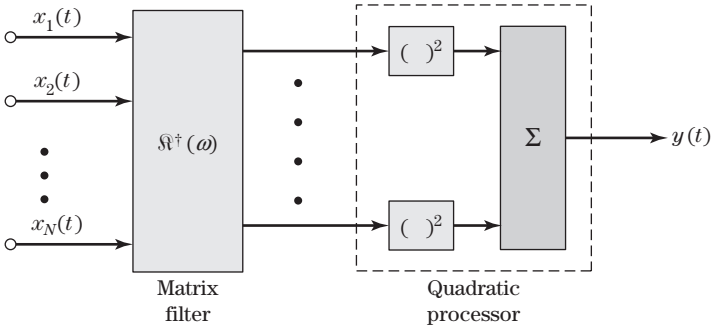


FIGURE 3-12 ■ Structure for general array processor.

It may be seen that (3.271) reduces to (3.246) when $\mathfrak{R}$ is a column vector. Under perturbed propagation conditions, $\mathbf{\Phi}_{ss}$ is given by [13]

$$\mathbf{\Phi}_{ss}(\omega) = \mathfrak{A}(\omega)\mathfrak{A}^\dagger(\omega) \tag{3.272}$$

where the matrix $\mathfrak{A}$ has N rows and r columns, and r denotes the rank of the matrix $\mathbf{\Phi}_{ss}$. For a plane wave signal, the cross-spectral density matrix $\mathbf{\Phi}_{ss}$ has rank one, and the dyad structure of (3.258) holds. The array gain given by (3.271) may be maximized in the same manner as (3.246), now using (D.16) instead of (D.14), to yield [13]

$$G \leq \left\{ \text{trace} \left(\left[\mathbf{\Phi}_{ss}(\omega) \mathbf{\Phi}_{nn}^{-1}(\omega) \right]^2 \right) \right\}^{1/2} \tag{3.273}$$

where equality obtains when the matrix $\mathfrak{R}$ is chosen to be a scalar multiple of $\mathfrak{A}^\dagger \mathbf{\Phi}_{nn}^{-1}$. It follows that the maximum gain cannot be achieved by any $\mathfrak{R}$ having less than r columns.

3.6 | POLARIZATION SENSITIVE ARRAYS

For radio applications (in contrast to sonar applications), if an array receives and uses more than one polarization (by using polarization sensitive elements), then its performance is far superior to one that does not [34]. When two signals arrive from the same direction, it is possible to null one signal and not the other if their polarizations are different. How

"different" must two polarizations be to be handled differently by an antenna array? This question concerns the separation between signal polarizations, which is treated by consideration of the Poincare sphere.

Antennas are designed for maximum directivity and polarization match when the antennas point at each other. Since the gain and polarization of an antenna change with angle, moving the antennas in a communication system causes a directivity loss coupled with a polarization mismatch, which reduces the received power. This polarization alignment problem is the reason that circular polarization is preferred in mobile communications systems.

To improve the link budget, the polarization and directivity of the receive or transmit antennas are adaptively modified as the positions of the two antennas change. Adapting the polarization requires an antenna, such as crossed dipoles, that can modify the major and minor axes of its polarization ellipse of the transmitted wave. Some applications that used adaptive crossed dipoles are given in [49–54].

Consider a four-element adaptive array consisting of two pairs of crossed dipoles as shown in Figure 3-13. The polarization properties of a transverse electromagnetic (TEM) wave are defined using the orthogonal electric fields shown in Figure 3-14, where that E_ϕ represents the horizontal field component, and E_θ represents the vertical field component. As time progresses, the wave propagates, and the E_ϕ and E_θ components traverse the polarization ellipse. The major axis of the ellipse has a tilt given by β (where $0 \leq \beta < \pi$

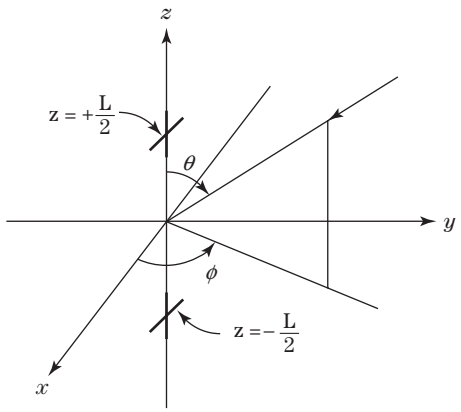


FIGURE 3-13 ■ Crossed Dipole Array Geometry. From Compton, IEEE Trans. Ant. & Prop., Sept. 1981.

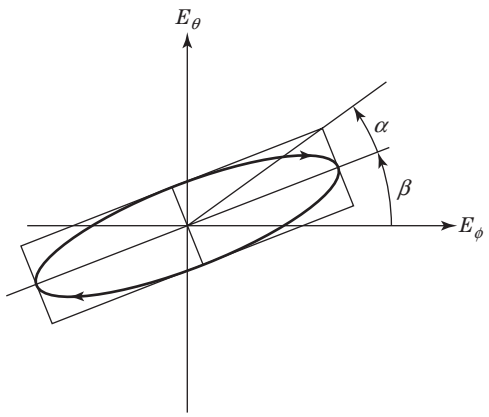


FIGURE 3-14 ■ Polarization Ellipse. From Compton, IEEE Trans. Ant. & Prop., Sept. 1981

to avoid ambiguity). The ellipticity angle, α, is then given by

$$\alpha = \tan^{-1}(\mathrm{r}) \tag{3.274}$$

where r is the axial ratio

$$\mathrm{r} = \frac{\text{minor axis}}{\text{major axis}} \tag{3.275}$$

the angle α is positive when the electric field vector rotates clockwise, and negative when it rotates counterclockwise (α is always in the range $-\pi/4 \leq \alpha \leq \pi/4$). Figure 3-14 depicts the situation where α is positive.

The electric field at any instant is given by the vector

$$\vec{\mathrm{E}} = \mathrm{E}_\phi \vec{\phi} + \mathrm{E}_\theta \vec{\theta} \tag{3.276}$$

The electric field components for a given state of polarization are related to the polarization ellipse and are given by (aside from a common phase factor)

$$\mathrm{E}_\phi = \mathrm{A}\cos\gamma \tag{3.277a}$$
$$\mathrm{E}_\theta = \mathrm{A}\sin\gamma\, \mathrm{e}^{\mathrm{j}\eta} \tag{3.277b}$$

where "A" denotes amplitude, and γ and η are related to α and β by

$$\cos(2\gamma) = \cos(2\alpha)\cos(2\beta) \tag{3.278a}$$
$$\tan(\eta) = \tan(2\alpha)\csc(2\beta) \tag{3.278b}$$

The previous equations relating the four variables α, β, γ, and η have a geometrical relationship shown on the Poincare sphere in Figure 3-15. For a given point on the sphere, M, the quantities 2γ, 2β, and 2α form the sides of a right spherical triangle. 2γ is the side of the triangle between the point M and the point labeled H (H is the point on the sphere representing horizontal polarization. The point V correspondingly represents vertical polarization and lies 180° removed from H in the equatorial plane of the sphere). The side 2β lies along the equator and extends to the point where it meets the perpendicular projection of M onto the equator. The angle η then lies between the sides 2γ and 2β. The special case $\alpha = 0$ corresponds to linear polarization in which case the point M lies on the equator: if in addition $\beta = 0$ then M lies at the point H, and only E_ϕ is nonzero so we have

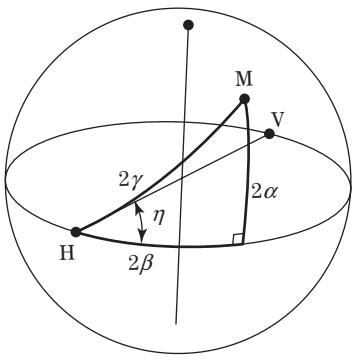


FIGURE 3-15 ■ Poincaré Sphere. From Compton, IEEE Trans. Ant. & Prop., Sept. 1981

horizontal polarization; if, however, $\beta = {}^{\pi}/_{2}$, then M lies at the point V, and we have vertical polarization. The poles of the sphere correspond to circular polarization ($\alpha = \pm 45^{\circ}$), with clockwise circular polarization ($\alpha = +45^{\circ}$) at the upper pole. We may conclude that five parameters characterize the polarization of an electric field: (θ, ϕ, α, β, and A).

The electric field in Figure 3-16 has an E_ϕ component with an x-component of $-E_\phi \sin\phi$ and a y-component of $E_\phi \cos\phi$. The component of E_θ lying in the x-y plane is $E_\theta \cos\theta$, while the z-component is $-E_\theta \sin\theta$. It immediately follows that the x and y components of E_θ are given by $E_\theta \cos\theta \cos\phi$ and $E_\theta \cos\theta \sin\phi$. The conversion of the electric field from spherical to rectangular coordinates is given by

$$\vec{E} = (E_\theta \cos\theta \cos\phi - E_\phi \sin\phi)\hat{x} + (E_\theta \cos\theta \sin\phi + E_\phi \cos\phi)\hat{y} - (E_\theta \sin\theta)\hat{z} \tag{3.279}$$

Substituting (3.277) into (3.279) then yields

$$\vec{E} = A[(\sin\gamma \cos\theta \cos\phi \; e^{j\eta} - \cos\gamma \sin\phi)\hat{x} + (\sin\gamma \cos\theta \cos\phi \; e^{j\eta} + \cos\gamma \cos\phi)\hat{y} - (\sin\gamma \sin\theta \; e^{j\eta})\hat{z}] \tag{3.280}$$

A crossed dipole antenna has two or three orthogonal dipoles that can excite two or three orthogonal linear polarizations. Adaptive crossed dipoles control the currents fed to the dipoles to change the polarization and radiation pattern. The array geometry is shown in Figure 3-16, where the top dipole consists of elements x_1(vertical) and x_2(horizontal along the x-axis) and the bottom dipole consists of elements x_3(vertical) and x_4(horizontal along the x-axis). We may then write the response of the four linear elements to the incoming electric field of Figure 3-16 (now including time and space phase factors) in vector form as

$$\mathbf{X} = Ae^{j(\omega t + \psi)}\mathbf{U} \tag{3.281}$$

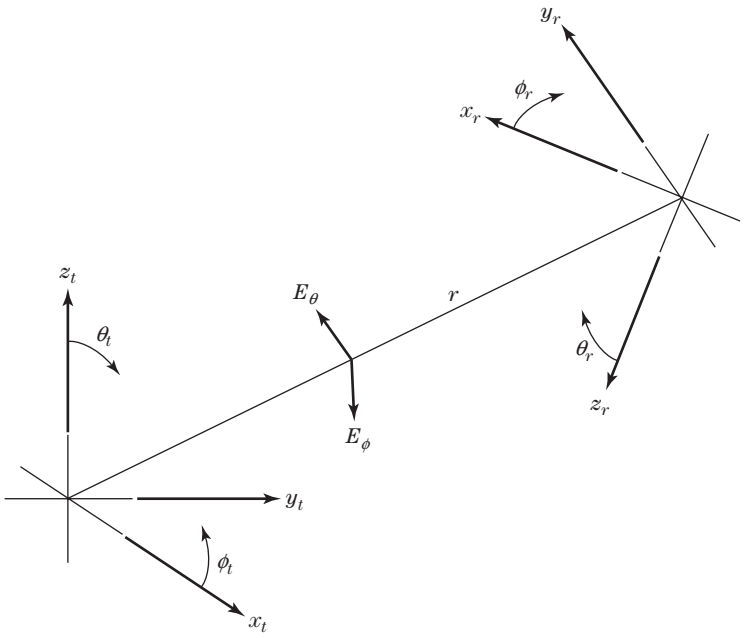


FIGURE 3-16 ■ Diagram of an adaptive crossed dipole communications system.

where

$$\mathbf{U} = \begin{bmatrix} (-\sin\gamma\sin\theta \quad e^{j\eta})e^{jp} \\ (\sin\gamma\cos\theta\cos\phi \quad e^{j\eta} - \cos\gamma\sin\phi)e^{jp} \\ (-\sin\gamma\sin\theta \quad e^{j\eta})e^{-jp} \\ (\sin\gamma\cos\theta\cos\phi \quad e^{j\eta} - \cos\gamma\sin\phi)e^{-jp} \end{bmatrix} \tag{3.282}$$

and ω is the carrier signal frequency, ψ is the carrier phase of the signal at the coordinate origin when t = 0, and p is the phase shift of the signals at the two dipoles with respect to the origin as a result of spatial delay where

$$\mathrm{p} = \frac{\pi L}{\lambda}\cos\theta \tag{3.283}$$

Now suppose there is a desired signal specified by ($\mathrm{A_d}$, θ_d, ϕ_d, α_d, β_d) and an interference signal by ($\mathrm{A_i}$, θ_i, ϕ_i, α_i, β_i). If thermal noise is present on each signal then the total signal vector is given by

$$\mathbf{X} = \mathbf{X}_\mathrm{d} + \mathbf{X}_\mathrm{i} + \mathbf{X}_\mathrm{n} \tag{3.284a}$$

$$= A_d e^{j(\varpi t+\varphi_d)}\mathbf{U}_\mathrm{d} + A_i e^{j(\varpi t+\varphi_i)}\mathbf{U}_\mathrm{i} + \mathbf{X}_\mathrm{n} \tag{3.284b}$$

where $\underline{\mathrm{U}}_\mathrm{d}$, $\underline{\mathrm{U}}_\mathrm{i}$ are given by (3.282). The corresponding signal covariance matrix is then given by

$$\mathbf{\Phi}_{\mathrm{xx}} = \mathbf{\Phi}_\mathrm{d} + \mathbf{\Phi}_\mathrm{i} + \mathbf{\Phi}_\mathrm{n} \tag{3.285a}$$

where

$$\mathbf{\Phi}_\mathrm{d} = E\{\mathbf{X}_d\mathbf{X}_d^H\} = \mathrm{A_d^2}\,\mathbf{U}_\mathrm{d}\,\mathbf{U}_\mathrm{d}^\mathrm{H} \tag{3.285b}$$

$$\mathbf{\Phi}_\mathrm{i} = E\{\mathbf{X}_i\mathbf{X}_\mathrm{i}^\mathrm{H}\} = \mathrm{A_i^2}\,\mathbf{U}_\mathrm{i}\,\mathbf{U}_\mathrm{i}^\mathrm{H} \tag{3.285c}$$

and

$$\mathbf{\Phi}_\mathrm{n} = \sigma^2\mathbf{I} \tag{3.285d}$$

A small difference in polarization between two received signals (measured in terms of the angular separation between two points on the Poincare sphere) is enough for an adaptive polarized array to provide substantial protection against the interference signal. Compton [34] shows that the signal-to-noise plus interference ratio (SINR) is nearly proportional to $\cos^2\left(\frac{\mathrm{M}_d\mathrm{M}_i}{2}\right)$, where $\mathrm{M}_d\mathrm{M}_i$ represents the (radian) distance on the sphere between the two polarizations.

The currents fed to the crossed dipoles in an adaptive communications system can be adjusted to optimize the received signal [54]. Figure 3-16 shows the transmit antenna is at an angle of (θ_r, φ_r) from the receive antenna, and the receive antenna is at an angle of (θ_t, φ_t). Maximum power transfer occurs when $\theta_t = 0^\circ$ and $\theta_r = 0^\circ$ or when the main beams point at each other. Controlling the complex signal weighting at the dipoles of the transmit and receive antennas modifies the directivity and polarization of both antennas. If the crossed dipole has three orthogonal dipoles, then the transmitted electric field is written as

$$E = -j\frac{\omega\mu e^{-jkr}}{4\pi r}\left[\left(I_xL_x\cos\theta\cos\varphi + I_yL_y\cos\theta\sin\varphi - I_zL_z\sin\theta\right)\hat{\theta}\right.$$
$$\left.+\left(-I_xL_x\sin\varphi + I_yL_y\cos\varphi\right)\hat{\phi}\right]$$

where

r = distance from the origin to the field point at *(x,y,z)*,
$L_{x,y,z}$ = dipole length in the x, y, and z directions
ω = radial frequency
k = wave number
μ = permeability
$I_{x,y,z}$ = constant current in the x, y, or z direction

The directivity and polarization loss factors are given by

$$D(\theta,\varphi) = 4\pi \frac{|E_\theta(\theta,\varphi)|^2 + |E_\varphi(\theta,\varphi)|^2}{\int_0^{2\pi}\int_0^{\pi}\left[|E_\theta(\theta,\varphi)|^2 + |E_\varphi(\theta,\varphi)|^2\right]\sin\theta d\theta d\varphi} \tag{3.286}$$

$$PLF = \frac{E_{\theta t}}{\sqrt{E_{\theta t}^2 + E_{\varphi t}^2}}\frac{E_{\theta r}}{\sqrt{E_{\theta r}^2 + E_{\varphi r}^2}} + \frac{E_{\varphi t}}{\sqrt{E_{\theta t}^2 + E_{\varphi t}^2}}\frac{E_{\varphi r}}{\sqrt{E_{\theta r}^2 + E_{\varphi r}^2}} \tag{3.287}$$

where $0 \le PLF \le 1$ with $PLF = 1$ a perfect match. The t and r subscripts represent transmit and receive, respectively.

When the transmitting antenna is a pair of orthogonal crossed dipoles in the x-y plane that emits a circularly polarized field in the z-direction, then increasing θ_t causes the transmitted electric field to transition from circular polarization through elliptical until linear polarization results at $\theta_t = 90^\circ$. Assume the transmit antenna is a ground station that tracks a satellite and has currents given by $I_x = 1$, $I_y = j$, and $I_z = 0$. The transmit antenna delivers a circularly polarized signal at maximum directivity to the moving receive antenna. If the receive antenna has two crossed dipoles, then the maximum receive power transfer occurs when the receive antenna is directly overhead of the transmit antenna. If the receive antenna remains circularly polarized as it moves, then the power received decreases, because the two antennas are no longer polarization matched. The loss in power transfer comes from a reduction in the directivity and the PLF. If the currents at each dipole are optimally weighted using an adaptive algorithm, then the power loss is compensated. The curve in Figure 3-17 results from finding the optimum weights for a range of angles. The link improvement over the nonadaptive antenna is as much as 3dB at $\theta_r = 90^\circ$.

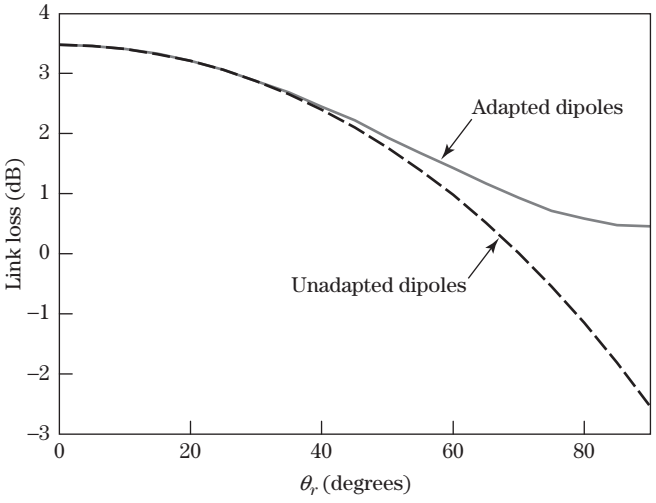


FIGURE 3-17 ■ Power loss as a function of receive angle when two dipoles are adaptive.

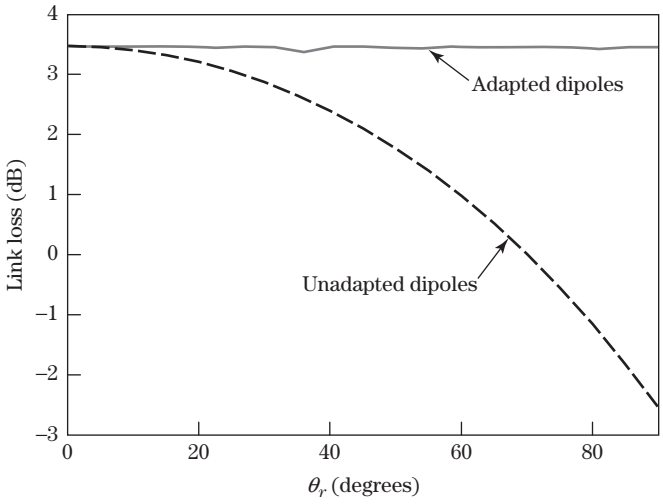


FIGURE 3-18 ■ Power loss as a function of receive angle when three dipoles are adaptive.

A third orthogonal dipole adds another degree of freedom to the receive antenna as shown in Figure 3-18. Since a tri-dipole antenna is able to produce any θ or φ polarization, the receive antenna adapts to maximize the received signal in any direction, and there is no change in the link budget as a function of θ_r. The three orthogonal dipoles compensate for changes in the receive antenna directivity and polarization as it moves and produces up to 6 dB improvement in the link budget at $\theta_r = 90^\circ$.

3.7 SUMMARY AND CONCLUSIONS

Optimum array processing techniques for several classes of detection, estimation, and array performance optimization problems have been shown to be closely related to one another and, in fact, are different only by virtue of some scalar processing that follows a common matrix filter and combiner operator. This matrix filter operator embodies the principle of "first prewhiten, then match" in its realization. For narrowband signals, the matching operation provides for time-delay (or phase) steering to obtain an undistorted representation of the signal. For broadband signals it is necessary to provide for signal spectral matching, which is not equivalent to time-delay steering. Perturbed signal propagation conditions resulting in noncoherent wavefront signals require element space-matched processing in which the signal matching can be performed only in a statistical sense. The optimum processors derived in this chapter for the most part have involved solving unconstrained optimization problems. Various classes of constrained optimum processors are introduced in the Problems section of this chapter.

The optimum array processor solutions resulting from the prewhitening and matching operation are intimately related to the optimal Wiener solution, which thereby provides a convenient theoretical performance limit against which to compare actual system performance. Direct implementation of an optimum processor solution is impractical because the signal environment statistics are usually unknown. Furthermore, even if the signal environment statistics were known, an optimal processor designed for a particular noise field would have limited utility; consequently, adaptive techniques for realizing an optimum

processor are extremely important. Since the spatial properties of the steady-state solutions resulting from various different performance measures are either identical or very similar, the question of which array performance measure to select for a given application is usually not very significant; rather, it is the temporal properties of the adaptive control algorithm to be used to reach the steady-state solutions that are of principal concern to the designer. The characteristics of adaptive control algorithms to be used for controlling the weights in the array pattern-forming network are therefore highly important, and it is to these concerns that Part 2 of this book is addressed.

Finally, the concept of a polarization sensitive array using polarization sensitive elements was introduced. Such an array possesses more degrees of freedom than a conventional array and permits signals arriving from the same direction to be distinguished from one another. The notion of polarization "distance" is introduced using the Poincare sphere. The polarization and gain of crossed dipoles can be adapted to maximize a dynamically changing communications link. Adaptively adjusting the currents on the dipoles in a crossed dipole system can significantly improve the link budget.

3.8 PROBLEMS

1. ***Use of the Maximum SNR Performance Measure in a Single Jammer Encounter [18]***
The behavior of a linear adaptive array controlled by the maximum SNR performance measure when the noise environment consists of a single jammer added to the quiescent environment is of some practical interest. Consider a linear, uniformly spaced array whose weights are determined according to (3.96). Assume that the quiescent noise environment is characterized by the covariance matrix

$$\mathbf{R}_{nn_q} = \begin{bmatrix} p_q & & & 0 \\ & p_q & & \\ & & \ddots & \\ 0 & & & p_q \end{bmatrix} = p_q \mathbf{I}_k$$

where p_q = noise power output of each of the K array elements
$\mathbf{I}_k$ = identity matrix of order K

In general, the various weight amplitudes are not equal, and the array beam pattern is given by

$$G_q(\beta) = \sum_{k=1}^{K} a_k e^{j(k-1)(\beta-\beta_s)}$$

where

$$\beta = \frac{2\pi d}{\lambda} \sin\theta$$

or, in matrix notation, $G_q(\beta) = \mathbf{b}^T \mathbf{w}_q$, where

$$\mathbf{b} = \begin{bmatrix} 1 \\ e^{j\beta} \\ e^{j2\beta} \\ \vdots \\ e^{j(K-1)\beta} \end{bmatrix}, \qquad \mathbf{w}_q = \begin{bmatrix} a_1 \\ a_2 e^{-j\beta_s} \\ a_3 e^{-j2\beta_s} \\ \vdots \\ a_k e^{-j(K-1)\beta_s} \end{bmatrix}$$

In view of the expressions for $\mathbf{w}_q$ and $G_q(\beta)$ previously given, it follows that

$$\mathbf{UHw}_q = G_q(\beta_J) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

where $\mathbf{U}$ and $\mathbf{H}$ are described subsequently.

Consider a single narrowband jammer located at an angle θ_J from mechanical boresight. Denote the jamming signal appearing in the first element channel by $J(t)$; then the jamming signal appearing in the kth channel will be $J(t)\exp[-j(k-1)\beta_J]$, where $\beta_J = (2\pi d/\lambda)\sin\theta_J$. Let p_J represent the envelope jamming power in each channel; then the covariance of the jamming signals in the kth and lth channels is $p_J \exp[-j(k-l)\beta_J]$, which represents the klth element of the jammer covariance matrix $\mathbf{R}_{JJ}$.

(a) Show that $\mathbf{R}_{JJ}$ is expressed as

$$\mathbf{R}_{JJ} = p_J \mathbf{H}^* \mathbf{UH}$$

where $\mathbf{H}$ is the diagonal matrix

$$\mathbf{H} = \begin{bmatrix} 1 & & & & \\ & e^{j\beta_J} & & & 0 \\ & & e^{j2\beta_J} & & \\ & & & \ddots & \\ 0 & & & & e^{j(K-1)\beta_J} \end{bmatrix}$$

and $\mathbf{U}$ is a $K \times K$ matrix of ones.

(b) By use of the control law (3.99), it follows that the weight vector for the quiescent noise environment should satisfy $\mathbf{R}_{nn_q}\mathbf{w}_q = \mu\mathbf{t}^*$, where $\mathbf{R}_{nn_q}\mathbf{w}_q = p_q\mathbf{w}_q$. If the noise environment changes so the noise covariance matrix is now $\mathbf{R}_{nn}$ instead of $\mathbf{R}_{nn_q}$, it follows that the optimum weight vector should then satisfy $\mathbf{R}_{nn}\mathbf{w} = \mu\mathbf{t}^*$ or $\mathbf{w} = p_q\mathbf{R}_{nn}^{-1}\mathbf{w}_q$. The covariance matrix corresponding to the total noise environment (consisting of the quiescent noise plus jammer) is the sum $\mathbf{R}_{nn} = \mathbf{R}_{nn_q} + \mathbf{R}_{JJ}$ or $\mathbf{R}_{nn} = p_q\mathbf{I}_k + p_J\mathbf{H}^*\mathbf{UH}$. To obtain the optimum weight vector for the total noise environment requires the inverse of $\mathbf{R}_{nn}$. Using the fact that $\mathbf{H}^* = \mathbf{H}^{-1}$, show that

$$\mathbf{R}_{nn}^{-1} = \frac{1}{p_q}\left\{\mathbf{I}_k - \left(\frac{P_J}{p_q + Kp_J}\right)\mathbf{H}^*\mathbf{UH}\right\}$$

(c) Using the result from part (b), show that the optimum weight vector is given by

$$\mathbf{w} = \mathbf{w}_q - \left(\frac{p_J}{p_q + Kp_J}\right)\mathbf{H}^*\mathbf{UHw}_q$$

(d) Let the optimum quiescent weight vector for a desired signal located in the direction θ_s from mechanical boresight be expressed as

$$\mathbf{w}_q = \begin{bmatrix} a_1 \\ a_2 e^{-j\beta_s} \\ a_3 e^{-j2\beta_s} \\ \vdots \\ a_K e^{-j(K-1)\beta_s} \end{bmatrix}$$

where

$$\beta_s = \frac{2\pi d}{\lambda} \sin\theta_s$$

and the a_k represent weight amplitudes. If the various a_k are all equal, the resulting array beam pattern is of the form $(\sin Kx)/(\sin x)$.

It then follows from the definition of $\mathbf{H}$ in part (a) that

$$\mathbf{H}\mathbf{w}_q = \begin{bmatrix} a_1 \\ a_2 e^{j(\beta_J - \beta_s)} \\ \vdots \\ a_K e^{j(K-1)(\beta_J - \beta_s)} \end{bmatrix}$$

From the foregoing expressions and the results of part (c), show that the pattern obtained with the optimum weight vector may be expressed as

$$G(\beta) = \mathbf{b}^T\mathbf{w} = \mathbf{b}^T\mathbf{w}_q - \left(\frac{P_J}{P_q + KP_J}\right) G_q(\beta_J)\mathbf{b}_J^*$$

where $\mathbf{b}_J$ is just $\mathbf{b}$ with the variable β_J replacing β.

(e) Recalling from part (d) that $\mathbf{b}^T\mathbf{w}_q = G_q(\beta)$, show that $\mathbf{b}^T\mathbf{b}_J^* = C(\beta - \beta_J)$ where

$$C(x) = \exp\left\{ j\left[\frac{(K-1)x}{2}\right] \frac{\sin Kx/2}{\sin x/2} \right\}$$

(f) Using the fact that $\mathbf{b}^T\mathbf{w}_q = G_q(\beta)$ and the results of parts (d) and (e), show that

$$G(\beta) = G_q(\beta) - \left(\frac{P_J}{P_q + KP_J}\right) G_q(\beta_J)C(\beta - \beta_J)$$

This result expresses the fact that the array beam pattern of an adaptively controlled linear array in the presence of one jammer consists of two parts. The first part is the quiescent pattern $G_q(\beta)$, and the second part (which is subtracted from the first part) is a $(\sin Kx)/(\sin x)$-shaped cancellation beam centered on the jammer.

(g) From the results of part (e) it may be seen that $C(x)|_{x=0} = K$. Using this fact show that the gain of the array in the direction of the jammer is given by

$$G(\beta_J) = \left(\frac{P_q}{P_q + KP_J}\right) G_q(\beta_J)$$

With the array weights fixed at $\mathbf{w}_q$, the array pattern gain in the direction of the jammer would be $G_q(\beta_J)$. The foregoing result therefore shows that the adaptive control reduces the gain in the direction of the jammer by the factor

$$\frac{P_q}{P_q + KP_J} = \frac{1}{1 + K(P_J/P_q)}$$

(h) The proper measure of performance improvement against the jammer realized by the introduction of adaptively controlled weights is the cancellation ratio Γ defined by

$$\Gamma \triangleq \frac{\gamma}{1 - \gamma \cdot (J/N)_q}$$

where

$$\gamma \triangleq \frac{\mathbf{w}_q^\dagger \mathbf{R}_{nn_q} \mathbf{R}_{nn}^{-1} \mathbf{R}_{JJ} \mathbf{w}_q}{\mathbf{w}_q^\dagger \mathbf{R}_{JJ} \mathbf{w}_q}$$

and $(J/N)_q$ denotes the ratio of the jammer noise output to the quiescent noise output when the weights are fixed at $\mathbf{w}_q$, that is,

$$\left(\frac{J}{N}\right)_q \triangleq \frac{\mathbf{w}_q^\dagger \mathbf{R}_{JJ} \mathbf{w}_q}{\mathbf{w}_q^\dagger \mathbf{R}_{nn_q} \mathbf{w}_q}$$

It is now desired to evaluate γ and Γ for the single jammer environment. First, show that

$$\mathbf{R}_{nn_q} \mathbf{R}_{nn}^{-1} = I_k - \left(\frac{1}{P_q + KP_J}\right) \mathbf{R}_{JJ}$$

Then show that

$$\mathbf{R}_{JJ} \mathbf{R}_{JJ} = KP_J \mathbf{R}_{JJ}$$

to obtain the result

$$\mathbf{R}_{nn_q} \mathbf{R}_{nn}^{-1} \mathbf{R}_{JJ} = \left(\frac{P_q}{P_q + KP_J}\right) \mathbf{R}_{JJ}$$

Substituting these results into the expression for γ immediately yields

$$\gamma = \frac{1}{1 + (KP_J/P_q)}$$

For a main beam of the form $(\sin Kx)/(\sin x)$, the maximum possible value of output jammer noise power is Kp_J, which will occur only at the peak of the main beam. Consequently, the maximum possible value of $(J/N)_q$ is KP_J/P_q, in which case $\Gamma = 1$ (thereby indicating no performance improvement against the jammer). For jammers located in the sidelobe region, however, $(J/N)_q \ll KP_J/P_q$ so that $\gamma \cdot (J/N)_q \ll 1$; hence, $\Gamma \approx \gamma$. For a jammer located in the sidelobe region, the adaptive control "cancels" the jammer power by approximately the jammer-to-noise ratio in the cancellation beam.

2. ***Effect of Adaptive Weight Errors [55]***

(a) For the CSLC system depicted in Figure 3-5 with $w_0 = 1$, show that $P_r = P_x - \mathbf{w}^\dagger \mathbf{z} - \mathbf{z}^\dagger \mathbf{w} + \mathbf{w}^\dagger \mathbf{R}_{yy} \mathbf{w}$.

(b) Since $\mathbf{w}_{\text{opt}} = \mathbf{R}_{yy}^{-1} \mathbf{z}^*$ and $P_{r_{\min}} = P_x - \mathbf{z}^\dagger \mathbf{R}_{yy}^{-1} \mathbf{z}$, show that when $\mathbf{w} = \mathbf{w}_{\text{opt}} + \Delta$, then $P_r = P_{r_{\min}} + P_{\text{add}}$ where $P_{\text{add}} = \Delta^\dagger \mathbf{R}_{yy} \Delta$. Note that since P_{add} is given by a Hermitian quadratic form, it is bounded by the smallest and largest eigenvalues of $\mathbf{R}_{yy}$ as $\lambda_{\min} ||\Delta||^2 \leq P_{\text{add}} \leq \lambda_{\max} ||\Delta||^2$ where $||\Delta||^2 \triangleq \Delta^\dagger \Delta$.

(c) Since $\lambda_{\min}$ and $\lambda_{\max}$ are both geometry dependent, it is instructive to use a worst-case design approach. Assume each auxiliary array element receives the same interference power P_{I_a}. Furthermore, assume that the receiver of each array element has the same effective received

noise power P_n. Under these conditions, each diagonal entry of $\mathbf{R}_{yy}$ is equal to $P_n + P_{I_a}$, and hence trace $(\mathbf{R}_{yy}) = N(P_n + P_{I_a})$. Since the largest eigenvalue of $\mathbf{R}_{yy}$ is less than the trace of $\mathbf{R}_{yy}$, it follows that $P_{\text{add}} \leq N(P_n + P_{I_a})||\Delta||^2$. Assume that the weight errors are due to quantization errors where the quanta size of the in-phase and quadrature channel is q. Under worst-case conditions, each complex weight component quantization error is identical and equal to $(q/2)(i \pm j)$. Hence, show that

$$P_{\text{add}} \leq \frac{N^2 q^2}{2}(P_n + P_{I_a})$$

(d) The interference-to-noise ratio $(P_I/P_n)_{\text{main}}$ for the main element is related to the ratio $(P_{I_a}/P_n)_{\text{aux}}$ for each auxiliary element by

$$\left(\frac{P_I}{P_n}\right)_{\text{main}} = |\alpha|^2 \left(\frac{P_{I_a}}{P_n}\right)_{\text{aux}}$$

where α is the average voltage gain of the main antenna in the sidelobe region. Assume that α is given by

$$\alpha = q \cdot 2^{B-1}$$

where B represents the number of bits available to implement the quantizer. Show that by assuming $(P_I/P_n)_{\text{main}} \gg 1$ and using the results of part (c), then

$$\frac{P_{\text{add}}}{P_{r_{\min}}} = \left(\frac{N^2}{2^{2B-1}}\right) R_0$$

where

$$R_0 \triangleq \frac{P_{r_{\min}}}{P_n}$$

(e) The principle use of a worst-case analysis is to determine the minimum number of bits required to avoid any degradation of the SIR performance. When the number of available quantization bits is significantly less than the minimum number predicted by worst-case analysis, a better prediction of SIR performance is obtained using an MSE analysis. The only change required for an MSE analysis is to treat $||\Delta||^2$ in an appropriate manner. Assuming the weight errors of the in-phase and quadrature channels are independent and uniformly distributed, show that

$$||\Delta||^2_{\text{average}} = \frac{1}{6}||\Delta||^2_{\max}$$

and develop corresponding expressions for P_{add} and $P_{\text{add}}/P_{r_{\min}}$.

3. ***Wiener Linear MMSE Filtering for Broadband Signals [56]***
Consider the scalar signal $x(t)$

$$x(t) = s(t) + n(t)$$

where the desired signal $s(x)$ and the noise $n(t)$ are uncorrelated. The signal $x(t)$ is to be passed through a linear filter $h(t)$ so that the output $y(t)$ will best approximate $s(t)$ in the MMSE sense. The error in the estimate $y(t)$ is given by

$$e(t) = y(t) - s(t)$$

and it is desired to minimize $E\{e^2(t)\}$ where

$$y(t) = \int_{-\infty}^{\infty} h(\tau)s(t-\tau)d\tau = \underbrace{\int_{-\infty}^{\infty} h(\tau)s(t-\tau)dt}_{\triangleq s'(t)} + \underbrace{\int_{-\infty}^{\infty} h(\tau)n(t-\tau)d\tau}_{\triangleq n'(t)}$$

(a) Show that

$$E\{e^2(t)\} = E\underbrace{\left\{[s'(t)-s(t)]^2\right.}_{\substack{\text{signal}\\ \text{distortion}\\ \text{component}}} \underbrace{\left.+n'^2(t)\right\}}_{\substack{\text{noise}\\ \text{component}}}$$

(b) Show that

$$E\{n'^2(t)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} r_{nn}(v)h(\tau)h(\tau+v)dvd\tau$$

where $r_{nn}(v) =$ autocorrelation function of $n(t)$.

(c) By not imposing realizability or finite memory requirements on $h(t)$, then $H(\omega)$ may be introduced using

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t}dt = \Im\{h(t)\}$$

show that

$$E\{n'^2(t)\} = \int_{-\infty}^{\infty} H(\omega)H^*(\omega)\phi_{nn}(\omega)d\omega$$

where $\phi_{nn}(\omega) = \Im\{r_{nn}(\tau)\}$, the spectral density function of $n(t)$. Likewise, show that

$$E\{[s'(t)-s(t)]^2\} = \int_{-\infty}^{\infty}[1-H(\omega)][1-H^*(\omega)]\cdot\phi_{ss}(\omega)d\omega$$

so that

$$E\{e^2(t)\} = \int_{-\infty}^{\infty}\left\{H(\omega)H^*(\omega)\phi_{nn}(\omega)+[1-H(\omega)][1-H^*(\omega)]\phi_{ss}(\omega)\right\}d\omega$$

(d) It is desired to minimize $E\{e^2(t)\}$ by appropriately choosing $H(\omega)$. Since the integrand of the expression for $E\{e^2(t)\}$ appearing in part (c) is positive for all choices of $H(\omega)$, it is necessary only to minimize the integrand by choosing $H(\omega)$. Show that by setting the gradient of the integrand of $E\{e^2(t)\}$ with respect to $H(\omega)$ equal to zero, there results

$$H_{\text{opt}}(\omega) = \frac{\phi_{ss}(\omega)}{\phi_{ss}(\omega)+\phi_{nn}(\omega)}$$

which is the optimum scalar Wiener filter. Therefore, to obtain the scalar Wiener filter, it is necessary to know only the signal spectral density and the noise spectral density at the

point in question. For scalar processes, the foregoing result may be used to obtain the MMSE signal estimate by introducing the appropriate scalar Wiener filter at the point where a MMSE signal estimate is desired. It is easy to show that the corresponding result for vector processes is given by

$$\mathfrak{H}_{\text{opt}}(\omega) = [\mathbf{\Phi}_{ss}(\omega) + \mathbf{\Phi}_{nn}(\omega)]^{-1}\mathbf{\Phi}_{ss}(\omega)$$

and when $\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t)$ where $\mathbf{s}(t) = \mathbf{v}d(t)$ then $\mathfrak{H}_{\text{opt}}(\omega)$ is expressed as

$$\mathfrak{H}_{\text{opt}}(\omega) = \mathbf{\Phi}_{xx}^{-1}(\omega)\phi_{xd}(\omega)$$

which represents the broadband signal generalization of (3.56).

(e) Determine the appropriate scalar Wiener filter to insert at point III in Figure 3-10 to obtain the MMSE estimate of $s(t)$. Likewise, determine the appropriate scalar Wiener filter to insert at point V to obtain the MMSE estimate of $s(t)$.

4. ***The Use of Multiple Linear Constraints in Array Gain Maximization [17]***
It was found that in the process of maximizing array gain with a single linear constraint of the form, $\mathbf{k}^\dagger\mathbf{m} = \beta$, the constraint could be handled by using the degree of freedom in specifying the magnitude of $\mathbf{k}$. In the event that a null constraint of the form $\mathbf{k}^\dagger\mathbf{d} = 0$ is introduced, however, the situation must be handled as a special case of the more general problem involving multiple linear constraints. Multiple linear constraints may be introduced to keep the array gain relatively constant over a range of signal perturbations, thereby reducing any sensitivity to such perturbations. Likewise, multiple linear constraints may be used to control sidelobes in a specific neighborhood of directions. The basic problem is stated as that of minimizing the output power z subject to a constraint of the form $\mathbf{H}^\dagger\mathbf{k} = \mathbf{g}$ where each row vector of the constraint matrix $\mathbf{H}^\dagger$ imposes a constraint of the form $\mathbf{h}_i^\dagger\mathbf{k} = g_i$. Clearly, then, the constraint matrix $\mathbf{H}^\dagger$ has a row for each constraint, and the total number of rows must be less than the number of sensors, or the problem will be overspecified. Introducing the multiple linear constraints into the expression for output power by using a vector Lagrange multiplier $\boldsymbol{\lambda}$, it is therefore desired to minimize

$$z = \left[\mathbf{k}^\dagger + \lambda^\dagger\mathbf{H}^\dagger{\Phi_{\text{xx}}}^{-1}\right]\Phi_{\text{xx}}\left[{\Phi_{\text{xx}}}^{-1}\mathbf{H}\lambda + \mathbf{k}\right] - \lambda^\dagger\mathrm{H}^\dagger{\Phi_{\text{xx}}}^{-1}\mathbf{H}\lambda - \lambda^\dagger\mathbf{g} - \mathbf{g}^\dagger\lambda$$

(a) Complete the square of the above expression for ???z to obtain the equivalent result

$$z = \left[\mathbf{k}^\dagger + \boldsymbol{\lambda}^\dagger\mathbf{H}^\dagger\mathbf{\Phi}_{xx}^{-1}\right]\mathbf{\Phi}_{xx}\left[\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\boldsymbol{\lambda} + \mathbf{k}\right] - \boldsymbol{\lambda}^\dagger\mathbf{H}_{xx}^{\dagger-1}\mathbf{H}\boldsymbol{\lambda} - \boldsymbol{\lambda}^\dagger\mathbf{g} - \mathbf{g}^\dagger\boldsymbol{\lambda}$$

Since $\mathbf{k}$ appears only in the previously given first quadratic term, the minimizing value of $\mathbf{k}$ is obviously that for which the quadratic term is zero or

$$\mathbf{k}_{\text{opt}} = -\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\boldsymbol{\lambda}$$

(b) Use the constraint equation $\mathbf{H}^\dagger\mathbf{k} = \mathbf{g}$ to eliminate $\boldsymbol{\lambda}$ from the result for $\mathbf{k}_{\text{opt}}$ obtained in part (a) and thereby show that

$$\mathbf{k}_{\text{opt}} = \mathbf{\Phi}_{xx}^{-1}\mathbf{H}\left[\mathbf{H}^\dagger\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\right]^{-1}\mathbf{g}$$

With this value of $\mathbf{k}$ the output power then becomes

$$z = \mathbf{g}^\dagger\left[\mathbf{H}\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\right]^{-1}\mathbf{g}$$

(c) For the general array processor of Figure 3-12, the output power is given by $z = \text{trace}(\Re^\dagger\mathbf{\Phi}_{xx}\Re)$, and the problem of minimizing z subject to multiple linear constraints of the form $\mathbf{H}^\dagger\Re = \mathbf{L}$ is handled by using a matrix Lagrange multiplier, $\boldsymbol{\Lambda}$, and considering

$$z = \text{trace}\left(\Re^\dagger\mathbf{\Phi}_{xx}\Re + \boldsymbol{\Lambda}^\dagger[\mathbf{H}^\dagger\Re - \mathbf{L}] + [\Re^\dagger\mathbf{H} - \mathbf{L}^\dagger]\boldsymbol{\Lambda}\right)$$

Complete the square of the previous expression to show that the optimum solution is

$$\Re_{\text{opt}} = \mathbf{\Phi}_{xx}^{-1}\mathbf{H}\left[\mathbf{H}^{\dagger}\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\right]^{-1}\mathbf{L}$$

for which $\Re_{\text{opt}}^{\dagger}\mathbf{\Phi}_{xx}\Re_{\text{opt}} = \mathbf{L}^{\dagger}\left[\mathbf{H}^{\dagger}\mathbf{\Phi}_{xx}^{-1}\mathbf{H}\right]^{-1}\mathbf{L}$

5. ***Introduction of Linear Constraints to the MSE Performance Measure [15]***
The theoretically optimum array processor structure for maximizing (or minimizing) some performance measure may be too complex or costly to fully implement. This fact leads to the consideration of suboptimal array processors for which the processor structure is properly constrained within the context of the signal and interference environment.

The K-component weight vector $\mathbf{w}$ is said to be linearly constrained if

$$f = \mathbf{c}^{\dagger}\mathbf{w}$$

The number of linear, orthonormal constraints on $\mathbf{w}$ must be less than K if any remaining degrees of freedom are to be available for adaptation.

(a) The MSE is expressed as

$$E\{|y - y_A|^2\}$$

where $y_A = \mathbf{w}^{\dagger}\mathbf{x}$.

Consequently

$$E\{|y - y_A|^2\} = E\{|y|^2\} - 2\,\text{Re}\{\mathbf{w}^{\dagger}\mathbf{r}_{xy}\} + \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}$$

Append the constraint equation to the MSE by means of a complex Lagrange multiplier to form

$$J = E\{|y|^2\} - 2\,\text{Re}\{\mathbf{w}^{\dagger}\mathbf{r}_{xy}\} + \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} + \lambda[f - \mathbf{c}^{\dagger}\mathbf{w}] + [f^* - \mathbf{w}^{\dagger}\mathbf{c}]\lambda^*$$

Take the gradient of the foregoing expression with respect to $\mathbf{w}$ and set the result equal to zero to obtain

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\left[\mathbf{r}_{xy} + \lambda^*\mathbf{c}\right]$$

(b) Apply the constraint $f = \mathbf{c}^{\dagger}\mathbf{w}$ to the result in part (a) thereby obtaining a solution for λ

$$\lambda = \frac{\left(f^* - \mathbf{r}_{xy}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{c}\right)}{\left(\mathbf{c}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{c}\right)}$$

This solution may be substituted into the result of part (a) to obtain the resulting solution for the constrained suboptimal array processor.

6. ***Introduction of Multiple Linear Constraints [15]***
A multiple linear constraint on the array processor takes the form

$$\mathbf{f} = \mathbf{C}^{\dagger}\mathbf{w}$$

where the matrix $\mathbf{C}$ has the vector $\mathbf{c}_i$ as its ith column, and the set $\{\mathbf{c}_i / i = 1, 2, \ldots, M\}$ must be a set of orthonormal constraint vectors.

Append the multiple linear constraint to the expected output power of the array output signal to determine the optimum constrained solution for $\mathbf{w}$.

7. ***Introduction of Quadratic Constraints to an Expected Output Power Performance Measure [15]***
The weight vector $\mathbf{w}$ is said to be quadratically constrained if

$$g = \mathbf{w}^{\dagger}\mathbf{Q}\mathbf{w}$$

where $\mathbf{Q}$ is a $K \times K$ Hermitian matrix.

(a) The expected output power of an array output signal is

$$E\{|y|^2\} = E\{\mathbf{w}^{\dagger}\mathbf{x}\mathbf{x}^{\dagger}\mathbf{w}\} = \mathbf{w}^{\dagger}E\{\mathbf{x}\mathbf{x}^{\dagger}\}\mathbf{w} = \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}$$

Appending the constraint equation to the expected output power with a complex Lagrange multiplier yields

$$J = \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} + \lambda[g - \mathbf{w}^{\dagger}\mathbf{Q}\mathbf{w}]$$

Take the gradient of the previous expression with respect to $\mathbf{w}$ and set the result equal to zero (to obtain the extremum value of $\mathbf{w}$) to obtain

$$\mathbf{R}_{xx}^{-1}\mathbf{Q}\mathbf{w} = \lambda^{-1}\mathbf{w}$$

(b) Note that the previous result for $\mathbf{w}$ is satisfied when $\mathbf{w}$ is an eigenvector of $\left(\mathbf{R}_{xx}^{-1}\mathbf{Q}\right)$ and λ is the corresponding eigenvalue. Therefore, maximizing (minimizing) J corresponds to selecting the largest (smallest) eigenvalue of $\left(\mathbf{R}_{xx}^{-1}\mathbf{Q}\right)$. It follows that a quadratic constraint is regarded simply as a means of scaling the weight vector $\mathbf{w}$ in the array processor.

(c) Append both a multiple linear constraint and a quadratic constraint to the expected output power of the array to determine the optimum constrained solution for $\mathbf{w}$. Note that once again the quadratic constraint merely results in scaling the weight vector $\mathbf{w}$.

8. ***Introduction of Single-Point, Multiple-Point, and Derivative Constraints to a Minimum Power Output Criterion for a Signal-Aligned Array [49]***

(a) Consider the problem of minimizing the array output power given by

$$P_0 = \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}$$

subject to the constraint $\mathbf{C}^{\dagger}\mathbf{w} = \mathbf{f}$. Show that the solution to this problem is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{C}\left[\mathbf{C}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{C}\right]^{-1}\mathbf{f}$$

(b) A single-point constraint corresponds to the case when the following conditions hold:

$\mathbf{C} = 1$, the $N \times 1$ vector of one's where N is the number of array elements

so the constraint equation becomes

$$\mathbf{w}^{\dagger}\mathbf{1} = N$$

Show that the corresponding optimal weight vector is given by

$$\mathbf{w}_{\text{opt}} = \frac{N\mathbf{R}_{xx}^{-1}\mathbf{1}}{\left(\mathbf{1}^{T}\mathbf{R}_{xx}^{-1}\mathbf{1}\right)}$$

Under the single-point constraint, the weight vector minimizes the output power in all directions except in the look (presteered) direction.

(c) A multiple-point constraint is introduced by specifying that $\mathbf{C}$ is a matrix of dimension $N \times 3$ (for a three-point constraint) given by

$$\mathbf{C} = \mathbf{P} = [\mathbf{e}_1, \mathbf{e}, \mathbf{e}_2]$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are direction vectors referenced to the beam axis on either side, and

$$\mathbf{f} = \mathbf{P}^\dagger \mathbf{e} = \begin{bmatrix} \mathbf{e}_1^\dagger \mathbf{e} \\ N \\ \mathbf{e}_2^\dagger \mathbf{e} \end{bmatrix}$$

where $\mathbf{C}^\dagger \mathbf{w} = \mathbf{f}$. Show that the corresponding optimal constrained weight vector is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1} \mathbf{P} \left[\mathbf{P}^\dagger \mathbf{R}_{xx}^{-1} \mathbf{P} \right]^{-1} \mathbf{P}^\dagger \mathbf{e}$$

(d) Derivative constraints are used to maintain a broader region of the main lobe by specifying both the response on the beam axis and the derivatives of the response on the beam axis. The constraint matrix now has dimension $N \times k$ and is given by

$$\mathbf{C} = \mathbf{D} = \left[\mathbf{e}_0, \mathbf{e}_0', \mathbf{e}_0'', \ldots \right]$$

where $\mathbf{e}_0$ = the array steering vector
$\mathbf{e}_0'$ = the derivative of $\mathbf{e}_0$ with respect to sin θ_0 where θ_0 is the look direction
$\mathbf{e}_0''$ = the second derivative of $\mathbf{e}_0$ with respect to sin θ_0
and $\mathbf{f}^T = \delta_{10}^T = [N, 0, 0, \ldots]$ has dimension $1 \times k$. Show that the corresponding optimal weight vector is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1} \mathbf{D}^\dagger \left[\mathbf{D} \mathbf{R}_{xx}^{-1} \mathbf{D} \right]^{-1} \boldsymbol{\delta}_{10}$$

9. ***Maximum Likelihood (ML) Estimates of Target Range and Bearing [39]***

Optimal array processors designed for the detection and estimation of desired signals may only partially satisfy the requirements placed on the system signal processor since the second stage of the signal processing problem often involves the extraction of information concerning parameters such as target range and bearing.

A maximum likelihood estimator selects the parameter α, which maximizes the conditional probability density function (or "likelihood function") $p(\mathbf{x}/\alpha)$. Since it is usually preferable to work with $\ln p(\cdot)$, solutions are sought to the equation

$$\frac{\partial}{\partial \alpha} \ln p(\mathbf{x}/\alpha) = 0$$

By forming the likelihood functional

$$y(\alpha) = \frac{\partial}{\partial \alpha} \ln p(\mathbf{x}/\alpha)$$

the maximum likelihood estimate $\hat{\alpha}_{\text{ML}}$ is found by requiring $y(\alpha)$ to be equal to zero, and the statistics of $\hat{\alpha}_{\text{ML}}$ is related to those of $y(\alpha)$.

When the components of the received signal vector $\mathbf{x}$ (using a Fourier coefficients representation) are complex Gaussian random processes, then

$$p(\mathbf{x}/\alpha) = \frac{1}{Det(\pi \mathbf{M})} \exp(-\mathbf{x}^\dagger M^{-1} x)$$

where $\mathbf{M} = \Phi_{ss}(\omega) + \Phi_{nn}(\omega)$.

(a) Using the expression for $p(\mathbf{x}/\alpha)$, show that the likelihood functional is expressed as

$$y(\alpha) = \mathbf{x}^{\dagger}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \alpha}\mathbf{M}^{-1}\mathbf{x} - \text{trace}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \alpha}\right)$$

Note that $\Phi_{ss}(\omega)$ is a function of α, whereas $\Phi_{nn}(\omega)$ is not.

(b) It is shown [40] that the variance associated with the ML estimate $\hat{\alpha}_{\text{ML}}$ for the case of spatially incoherent noise is proportional to the quantity

$$\text{var}(\hat{\alpha}_{\text{ML}}) \propto [\text{trace}(\mathbf{T}\mathbf{T}^{\dagger})]^{-1}$$

where $\mathbf{T}$ is a weighting matrix that incorporates all geometrical properties of the array. In particular, for linear arrays the $\mathbf{T}$ matrix has elements given by the following:

$$\text{For bearing estimation: } t_{ij} = \frac{\sin\theta}{\mathfrak{v}}(z_i - z_j)$$

$$\text{For range estimation: } t_{ij} = -\frac{\sin^2\theta}{2\mathfrak{v}r^2}\left(z_i^2 - z_j^2\right)$$

where θ = signal bearing with respect to the array normal
z_n = position of nth sensor along the array axis (the z-axis)
$\mathfrak{v}$ = velocity of signal propagation
r = true target range

Show that for an array having length L, and with $K \gg 1$ equally spaced sensors then

$$\text{For bearing estimation: } [\text{trace}(\mathbf{T}\mathbf{T}^{\dagger})]^{-1} = \frac{6\mathfrak{v}^2}{K^2L^2\sin^2\theta}$$

$$\text{For range estimation: } [\text{trace}(\mathbf{T}\mathbf{T}^{\dagger})]^{-1} = \frac{45\mathfrak{v}^2r^4}{2L^4K^2\sin^4\theta}$$

The foregoing results show that the range estimate accuracy is critically dependent on the true range, whereas the bearing estimate is not (except for the range dependence of the SNR). The range estimate is also more critically dependent on the array aperture L than the bearing estimate.

10. ***Suboptimal Bayes Estimate of Target Angular Location [9]***

The signal processing task of extracting information concerning parameters such as target angular location can also be carried out by means of Bayes estimation. A Bayes estimator is just the expectation of the variable being estimated conditioned on the observed data, that is,

$$\hat{u} = E\{u_k/\mathbf{x}\} = \int_{-\infty}^{\infty} u_k\, p(u_k/\mathbf{x})du$$

where $u_k = \sin\theta$ denotes the angular location of the kth target, $\mathbf{x}$ denotes the observed data vector, and the a posteriori probability density function $p(u/\mathbf{x})$ is rewritten by application of Bayes's rule as

$$p(u/\mathbf{x}) = \frac{p(\mathbf{x}/u)\,p(u)}{p(\mathbf{x})}$$

The optimum estimators that result using the foregoing approach are quite complex, requiring the evaluation of multiple integrals that may well be too lengthy for many portable applications. Consequently, the development of a simple suboptimum estimator that approximates the optimum Bayes estimator is of practical importance.

(a) The lowest-order nonlinear approximation to the optimum Bayes location estimator is given by [9]

$$\hat{\mathbf{u}} = \mathbf{x}^T\mathbf{B}$$

where the matrix of N target locations is a $1 \times N$ row vector

$$\hat{\mathbf{u}} = [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N]$$

Assuming a $(2K + 1)$ element array where each element output is amplified and detected with synchronous quadrature detectors, then the output of the mth pair of quadrature detectors is

$$x_{ym}(t) = s_{ym}(t) + n_{ym}(t)$$

and

$$x_{zm}(t) = s_{zm}(t) + n_{zm}(t)$$

The data vector $\mathbf{x}$ can then be defined by a $2K(2K + 1) \times 1$ column vector

$$\mathbf{x} = [x_{ym}x_{zn}]$$

in which each index pair m, n occupies a separate row. Define a $2K(2K + 1) \times 1$ column vector of coefficients

$$\mathbf{b}_k = \left[b_{mn}^{(k)}\right], \qquad m \neq n$$

where, again, each index pair m, n occupies a separate row. The full matrix of coefficients $\mathbf{B}$ is then given by the $2K(2K + 1) \times N$ matrix

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$$

Show that, by choosing the coefficient matrix $\mathbf{B}$ so that the resulting estimator is orthogonal to the estimation error, that is, $E[\hat{\mathbf{u}}^T(\mathbf{u} - \hat{\mathbf{u}})] = 0$, then $\mathbf{B}$ is given by

$$\mathbf{B} = [E\{\mathbf{x}\mathbf{x}^T\}]^{-1}E\{\mathbf{x}\mathbf{u}\}$$

With $\mathbf{B}$ selected as indicated, the MSE in the location estimator for the kth target is then given by

$$\mathfrak{E}_k = E[u_k(u_k - \hat{u}_k)] = E\left\{u_k^2\right\} - [E\{\mathbf{x}u_k\}]^T \cdot [E\{\mathbf{x}\mathbf{x}^T\}]^{-1} \cdot [E\{\mathbf{x}u_k\}]$$

(b) Consider a one-target location estimation problem using a two-element array. From the results of part (a), it follows that

$$\hat{u} = b_{12}x_{y1}x_{z2} + b_{21}x_{y2}x_{y1}$$

Let the signal components of the quadrature detector outputs be given by

$$s_{ym} = \alpha \cos m\pi u + \beta \sin m\pi u$$

and

$$s_{zm} = \beta \cos m\pi u - \alpha \sin m\pi u$$

where the joint probability density function (PDF) for α, β is given by

$$p(\alpha, \beta) = \frac{1}{2\pi\sigma^2}\exp\left\{-\frac{(\alpha^2+\beta^2)}{2\sigma^2}\right\}$$

Furthermore, assume the quadrature noise components are independent so the PDF for the noise originating at the mth antenna element is

$$p(n_{ym}, n_{zm}) = \frac{1}{2\pi\sigma_n^2}\exp\left\{-\frac{\left(n_{ym}^2+n_{zm}^2\right)}{2\sigma_n^2}\right\}$$

The aforementioned signal and noise models correspond to a Rayleigh fading environment and additive noise due to scintillating clutter. Show that the optimum coefficients for determining $\hat{u}$ are given by

$$b_{12} = -b_{21} = \frac{(1/\sigma^2)E\{u\sin\pi u\}}{4E\{\sin^2\pi u\} + 2/\gamma + 1/\gamma^2}$$

where $\gamma \triangleq \sigma^2/\sigma_n^2$, the SNR. Finally, show that the MSE for this estimator is given by

$$\text{MSE} = E\{u^2\} - \frac{E^2\{u\sin\pi u\}}{2E\{\sin^2\pi u\} + 1/\gamma + 2/2\gamma^2}$$

and therefore depends only on the SNR and $p(u)$.

11. ***Constrained Minimum Power Criterion for Element Space Matched Array Processing in Narrowband Application [48]***

A minimum variance estimate of the signal power P_s for an element space-matched array processor may be obtained by solving

$$\text{minimize} : \text{var}[\mathbf{x}^\dagger\mathbf{K}\mathbf{x}] = \text{tr}[(\mathbf{K}\mathbf{R}_{xx})^2]$$

subject to the constraint $\text{tr}[\mathbf{K}\mathbf{R}_{ss}] = 1$.

(a) Show that the solution of the problem is given by

$$\mathbf{K}_{\text{opt}} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{R}_{ss}\mathbf{R}_{xx}^{-1}}{\text{tr}\left[\left(\mathbf{R}_{xx}^{-1}\mathbf{R}_{ss}\right)^2\right]}$$

so the expected value of the processor output power is then

$$P_0 = \frac{\text{tr}\left[\mathbf{R}_{xx}^{-1}\mathbf{R}_{ss}\right]}{\text{tr}\left[\left(\mathbf{R}_{xx}^{-1}\mathbf{R}_{ss}\right)^2\right]}$$

(b) Show that when $\mathbf{R}_{ss}$ has rank one and is consequently given by the dyad $\mathbf{R}_{ss} = \mathbf{v}\mathbf{v}^\dagger$, then

$$\mathbf{K}_{\text{opt}} = \mathbf{h}(\omega)\mathbf{h}^\dagger(\omega)$$

where

$$\mathbf{h} = \text{scalar}\cdot\mathbf{v}^\dagger\mathbf{R}_{nn}^{-1}$$

This result demonstrates that under plane wave signal assumptions the element space-matched array processor degenerates into a plane wave matched processor with a quadratic detector.

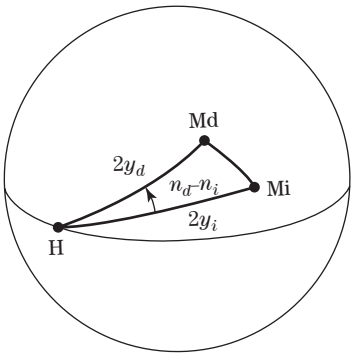


FIGURE 3-19 ■ Points M_d and M_i on the Poincare Sphere From Compton, IEEE Trans. On Ant. and Prop, September, 1981.

12. ***Signal-to-Interference–plus-Noise Ratio (SINR) for a Two Element Polarized Array [49]***
Let M_d and M_i be points on the Poincare sphere representing the polarizations of the desired and interference signals, respectively as shown in Figure 3-20. From the figure it is seen that $2\gamma_d$, $2\gamma_i$, and the arc M_dM_i form the sides of a spherical triangle. The angle $\eta_d - \eta_i$ is the angle opposite side M_dM_i. Using a well-known spherical trigonometric identity, the sides on the spherical triangle are related by $\cos 2\gamma_d \cos 2\gamma_i + \sin 2\gamma_d \sin 2\gamma_i \cos(\eta_d - \eta_i) = \cos(M_dM_i)$.

(a) For the case where both the desired and interference signals arrive from broadside, and $\theta_d = \phi_d = \theta_i = \phi_i = 90^\circ$, show that

$$|\mathbf{U}_d\mathbf{U}_i|^2 = 2[1 + \cos(M_dM_i)] = 4\cos^2\left(\frac{M_dM_i}{2}\right)$$

(b) Using the fact that $\text{SINR} = \dfrac{P_d}{P_i + P_n}$, where $P_d = \dfrac{A_d^2}{2}\left|\mathbf{U}_d^T\mathbf{w}\right|^2$, $P_i = \dfrac{A_i^2}{2}\left|\mathbf{U}_i^T\mathbf{w}\right|^2$, and $P_n = \dfrac{\sigma^2}{2}|\mathbf{w}|^2$ where $\mathbf{w}$ denotes the weight vector, use the matrix inversion lemma (m.i.l.) to show that the SINR is written as

$$\text{SINR} = \xi_d\left[\mathbf{U}_d^H\mathbf{U}_d - \frac{\left|\mathbf{U}_d^H\mathbf{U}_i\right|^2}{\xi_i^{-1} + \mathbf{U}_i^H\mathbf{U}_i}\right]$$

where $\xi_d = \dfrac{A_d^2}{\sigma^2}$ is the desired signal-to-noise ratio and similarly for ξ_i.

(c) Using the result of part (a) in the SINR of part (b) show that

$$\text{SINR} = \xi_d\left[2 - \frac{4\cos^2\left(\frac{M_dM_i}{2}\right)}{\xi_i^{-1} + 2}\right]$$

13. ***Development of the SINR [57]***
The signal covariance matrix is given by

$$\mathbf{\Phi} = \sigma^2\mathbf{I} + A_d^2\mathbf{U}_d\mathbf{U}_d^H + A_i^2\mathbf{U}_i\mathbf{U}_i^H,$$

where the $\mathbf{U}$ denote the steering vectors for the desired and interference signals, respectively

It is first desired to find $\mathbf{\Phi}^{-1}$.

(a) Use the matrix inversion lemma,

$$[\mathbf{B} - \beta\mathbf{Z}\mathbf{Z}^H]^{-1} = \mathbf{B}^{-1} - \tau\mathbf{B}^{-1}\mathbf{Z}^H\mathbf{B}^{-1}$$

where $\mathbf{B}$ is a nonsingular N × N matrix, $\mathbf{Z}$ is an N × 1 column vector, and β and τ are scalars related by

$$\tau^{-1} + \beta^{-1} = \mathbf{Z}\mathbf{B}^{-1}\mathbf{Z}^{\mathrm{H}}$$

Applying the m.i.l. to the quantity $\sigma^2\mathbf{I} + \mathrm{A}_i^2\mathbf{U}_i\mathbf{U}_i^{\mathrm{H}}$, where $\mathbf{B} = \sigma^2\mathbf{I}$, $\mathbf{Z} = \mathbf{U}_i$, and $\beta = -\mathrm{A}_i^2$, show that

$$\tau = (\mathbf{Z}\mathbf{B}^{-1}\mathbf{Z}^{\mathrm{H}} - \beta^{-1})^{-1} = \left(\frac{1}{\sigma^2}\mathbf{U}_i\mathbf{U}_i^{\mathrm{H}} + \frac{1}{\mathrm{A}_i^2}\right)^{-1}$$

and hence

$$\left[\sigma^2\mathbf{I} + \mathrm{A}_i^2\mathbf{U}_i\mathbf{U}_i^{\mathrm{H}}\right]^{-1} = \frac{1}{\sigma^2}\left[\mathbf{I} - \frac{\mathbf{U}_i\mathbf{U}_i^{\mathrm{H}}}{\xi_i^{-1} + \mathbf{U}_i\mathbf{U}_i^{\mathrm{H}}}\right]$$

where ξ_i is defined to be A_i^2/σ^2

(b) Use the m.i.l again, this time letting $\mathbf{B} = \sigma^2\mathbf{I} + \mathrm{A}_i^2\mathbf{U}_i\,\mathbf{U}_i^{\mathrm{H}}$, $\mathbf{Z} = \mathbf{U}_d$, and $\beta = -\mathrm{A}_d^2$ to show that

$$\mathbf{\Phi}^{-1} = \frac{1}{\sigma^2}\mathbf{I} - \frac{\tau}{\sigma^4}\mathbf{U}_d\mathbf{U}_d^{\mathrm{H}} - \frac{1}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\left[\frac{1}{\sigma^2} + \frac{\tau}{\sigma^2}\frac{\left(\mathbf{U}_d\mathbf{U}_i^{\mathrm{H}}\right)\left(\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}\right)}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right]\mathbf{U}_i\mathbf{U}_i^{\mathrm{H}}$$

$$+ \frac{\tau}{\sigma^4}\frac{\left(\mathbf{U}_d\mathbf{U}_d^{\mathrm{H}}\right)}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\mathbf{U}_d\mathbf{U}_i^{\mathrm{H}} + \frac{\tau}{\sigma^4}\frac{\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}$$

where

$$\tau^{-1} = \frac{1}{A_d^2} + \frac{1}{\sigma^2}\mathbf{U}_d\mathbf{U}_d^{\mathrm{H}} - \frac{1}{\sigma^2}\left[\frac{\left(\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}\right)\left(\mathbf{U}_d\mathbf{U}_i^{\mathrm{H}}\right)}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right]$$

(c) Use the fact that $\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{S}$ where $\mathbf{S} = \mathrm{E}\{\mathbf{X}^*\mathbf{R}(t)\} = A_rA_d\mathbf{U}_d^*$ along with the result of part (b) to show that

$$\mathbf{w} = \frac{A_rA_d}{\sigma^2}\left(1 - \frac{\tau}{\sigma^2}\frac{\gamma}{A_d^2}\right)\left(\mathbf{U}_d^* - \frac{\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\mathbf{U}_i^*\right).$$

where A_r denotes reference signal amplitude, and γ is given by

$$\gamma = A_d^2\left[\mathbf{U}_d\mathbf{U}_d^{\mathrm{H}} - \frac{\left(\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}\right)\left(\mathbf{U}_d\mathbf{U}_i^{\mathrm{H}}\right)}{\xi_i^{-1} + \mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right]$$

(d) Use the result of part (c) to show that

$$\mathbf{U}_d^{\mathrm{T}}\mathbf{w} = \frac{A_rA_d}{\sigma^2}\left(1 - \frac{\tau}{\sigma^2}\frac{\gamma}{A_d^2}\right)\frac{\gamma}{A_d^2}$$

(e) Use the result of part (d) along with the fact that τ from part (a) is expressed as

$$\tau = A_d^2\left(1 + \frac{\gamma}{\sigma^2}\right)^{-1}$$

to show that

$$\mathbf{U}_d^{\mathrm{T}}\mathbf{w} = \frac{A_r}{A_d}\left(\frac{\gamma}{\sigma^2 + \gamma}\right).$$

(f) Use the result of part (e) along with $\mathrm{P}_d = \frac{A_d^2}{2}\left|\mathbf{U}_d^T\mathbf{w}\right|^2$ to show that

$$\mathrm{P}_d = \frac{A_r^2}{2}\left(\frac{\gamma}{\sigma^2+\gamma}\right)^2$$

(g) In a manner similar to part (d) show that

$$\mathbf{U}_i^{\mathrm{T}}\mathbf{w} = \frac{A_r A_d}{\sigma^2}\left(1-\frac{\tau}{\sigma^2}\frac{\gamma}{A_d^2}\right)\mathbf{U}_i\mathbf{U}_d^{\mathrm{H}}\left(\frac{\xi_i^{-1}}{\xi_i^{-1}+\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right)$$

(h) Use the result of part (g) along with $\mathrm{P}_i = \frac{A_i^2}{2}\left|\mathbf{U}_i^T\mathbf{w}\right|^2$ to show that

$$\mathrm{P}_i = \frac{A_r^2A_d^2A_i^2}{2}\left(\frac{1}{\sigma^2+\gamma}\right)^2\left(\frac{\xi_i^{-1}}{\xi_i^{-1}+\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right)^2\left(\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_d\right)\left(\mathbf{U}_d^{\mathrm{H}}\mathbf{U}_i\right)$$

(i) Finally, from part (c) show that

$$|\mathbf{w}|^2 = A_r^2A_d^2\left(\frac{1}{\sigma^2+\gamma}\right)^2\left[\frac{\gamma}{A_d^2}-\left(\mathbf{U}_d^{\mathrm{H}}\mathbf{U}_i\right)\left(\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_d\right)\frac{\xi_i^{-1}}{\left(\xi_i^{-1}+\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i\right)^2}\right]$$

(j) From the fact that $\mathrm{P}_n = \frac{\sigma^2}{2}|\mathbf{w}|^2$, use the result of part (i) to show that

$$\mathrm{P}_n = \frac{A_r^2A_d^2A_i^2}{2}\left(\frac{1}{\sigma^2+\gamma}\right)^2\left\{\frac{\gamma\xi_i^{-1}}{A_d^2}-\left(\mathbf{U}_d^{\mathrm{H}}\mathbf{U}_i\right)\left(\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_d\right)\left(\frac{\xi_i^{-1}}{\xi_i^{-1}+\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right)^2\right\}$$

(k) Combining the results of part (h) with part (j), show that

$$\mathrm{P}_i+\mathrm{P}_n = \frac{A_r^2A_i^2}{2}\left(\frac{1}{\sigma^2+\gamma}\right)^2\gamma\xi_i^{-1}$$

(l) Using the results of part (f) for P_d, show that the SINR reduces to the form

$$\mathrm{SINR} = \frac{\mathrm{P}_d}{\mathrm{P}_i+\mathrm{P}_n} = \frac{\gamma}{\sigma^2}.$$

Using the definition of γ given in part (c), it follows that an alternative expression for the SINR is given by

$$\mathrm{SINR} = \xi_d\left[\mathbf{U}_d^{\mathrm{H}}\mathbf{U}_d-\frac{\left|\mathbf{U}_d^{\mathrm{H}}\mathbf{U}_i\right|^2}{\xi_i^{-1}+\mathbf{U}_i^{\mathrm{H}}\mathbf{U}_i}\right]$$

3.9 REFERENCES

[1] P. L. Stocklin, "Space-Time Sampling and Likelihood Ratio Processing in Acoustic Pressure Fields," *J. Br. IRE*, July 1963, pp. 79–90.

[2] F. Bryn, "Optimum Signal Processing of Three-Dimensional Arrays Operating on Gaussian Signals and Noise," *J. Acoust. Soc. Am.*, Vol. 34, No. 3, March 1962, pp. 289–297.

[3] D. J. Edelblute, J. M. Fisk, and G. L. Kinneson, "Criteria for Optimum-Signal-Detection Theory for Arrays," *J. Acoust. Soc. Am.*, Vol. 41, January 1967, pp. 199–206.

[4] H. Cox, "Optimum Arrays and the Schwartz Inequality," *J. Acoust. Soc. Am.*, Vol. 45, No. 1, January 1969, pp. 228–232.

[5] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, December 1967, pp. 2143–2159.

[6] L. J. Griffiths, "A Simple Algorithm for Real-Time Processing in Antenna Arrays," *Proc. IEEE*, Vol. 57, October 1969, pp. 1696–1707.

[7] N. Owsley, "Source Location with an Adaptive Antenna Array," Naval Underwater Systems Center, Rept. NL-3015, January 1971.

[8] A. H. Nuttall and D. W. Hyde, "A Unified Approach to Optimum and Suboptimum Processing for Arrays," *U.S. Navy Underwater Sound Laboratory Report 992,* April 1969, pp. 64–68.

[9] G. W. Young and J. E. Howard, "Applications of Spacetime Decision and Estimation Theory to Antenna Processing System Design," *Proc. IEEE*, Vol. 58, May 1970, pp. 771–778.

[10] H. Cox, "Interrelated Problems in Estimation and Detection I and II," Vol. 2, *NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics,* Enchede, The Netherlands, August 12–23, 1968, pp. 23-1–23-64.

[11] N. T. Gaarder, "The Design of Point Detector Arrays," *IEEE Trans. Inf. Theory*, part 1, Vol. IT-13, January 1967, pp. 42–50; part 2, Vol. IT-12, April 1966, pp. 112–120.

[12] J. Capon, "Applications of Detection and Estimation Theory to Large Array Seismology," *Proc. IEEE*, Vol. 58, May 1970, pp. 760–770.

[13] H. Cox, "Sensitivity Considerations in Adaptive Beamforming," *Proceedings of NATO Advanced Study Institute on Signal Processing,* Loughborough, England, August 1972, pp. 621–644.

[14] A. M. Vural, "An Overview of Adaptive Array Processing for Sonar Applications," *IEEE 1975 EASCON Record, Electronics and Aerospace Systems Convention,* September 29–October 1, pp. 34.A–34 M.

[15] J. P. Burg, "Maximum Entropy Spectral Analysis," *Proceedings of NATO Advanced Study Institute on Signal Processing,* August 1968.

[16] N. L. Owsley, "A Recent Trend in Adaptive Spatial Processing for Sensor Arrays: Constrained Adaptation," *Proceedings of NATO Advanced Study Institute on Signal Processing,* Loughborough, England, August 1972, pp. 591–603.

[17] R. R. Kneiper, et al., "An Eigenvector Interpretation of an Array's Bearing Response Pattern," *Naval Underwater Systems Center Report No. 1098,* May 1970.

[18] S. P. Applebaum, "Adaptive Arrays," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 585–598.

[19] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York, McGraw-Hill, 1965, Ch. 8.

[20] R. A. Wiggins and E. A. Robinson, "Recursive Solution to the Multichannel Filtering Problem," *J. Geophys. Res.*, Vol. 70, No. 8, April 1965, pp. 1885–1891.

[21] R. V. Churchill, *Introduction to Complex Variables and Applications*, New York, McGraw-Hill, 1948, Ch. 8.

[22] M. J. Levin, "Maximum-Likelihood Array Processing," Lincoln Laboratories, Massachusetts Institute of Technology, Lexington, MA, Semiannual Technical Summary Report on Seismic Discrimination, December 31, 1964.

[23] J. Chang and F. Tuteur, *Symposium on Common Practices in Communications*, Brooklyn Polytechnic Institute, 1969.

[24] P. E. Mantey and L. J. Griffiths, "Iterative Least-Squares Algorithms for Signal Extraction," *Proceedings of the 2nd Hawaii Conference on System Sciences,* 1969, pp. 767–770.

[25] R. L. Riegler and R. T. Compton, Jr., "An Adaptive Array for Interference Rejection," *Proc. IEEE*, Vol. 61, No. 6, June 1973, pp. 748–758.

[26] S. P. Applebaum, Syracuse University Research Corporation Report SPL TR 66-1, Syracuse, NY, August 1966.

[27] S. W. W. Shor, "Adaptive Technique to Discriminate Against Coherent Noise in a Narrow-Band System," *J. Acous. Soc. Am.*, Vol. 34, No. 1, pp. 74–78.

[28] R. T. Adams, "An Adaptive Antenna System for Maximizing Signal-to-Noise Ratio," *WESCON Conference Proceedings,* Session 24, 1966, pp. 1–4.

[29] R. Bellman, *Introduction to Matrix Analysis*, New York, McGraw-Hill, 1960.

[30] R. F. Harrington, *Field Computation by Moment Methods*, New York, Macmillan, 1968, Ch. 10.

[31] B. D. Steinberg, *Principles of Aperture and Array System Design*, New York, Wiley, 1976, Ch. 12.

[32] L. J. Griffiths, "Signal Extraction Using Real-Time Adaptation of a Linear Multichannel Filter," SEL-68-017, Technical Report No. 6788-1, System Theory Laboratory, Stanford University, February 1968.

[33] R. T. Lacoss, "Adaptive Combining of Wideband Array Data for Optimal Reception," *IEEE Trans. Geosci. Electron.*, Vol. GE-6, No. 2, May 1968, pp. 78–86.

[34] C. A. Baird, Jr., and J. T. Rickard, "Recursive Estimation in Array Processing," *Proceedings of the Fifth Asilomar Conference on Circuits and Systems,* 1971, pp. 509–513.

[35] C. A. Baird, Jr., and C. L. Zahm, "Performance Criteria for Narrowband Array Processing," *IEEE Conference on Decision and Control,* December 15–17, 1971, Miami Beach, FL, pp. 564–565.

[36] W. W. Peterson, T. G. Birdsall, and W. C. Fox, "The Theory of Signal Detectability," *IRE Trans.*, PGIT-4, 1954, pp. 171–211.

[37] D. Middleton and D. Van Meter, "Modern Statistical Approaches to Reception in Communication Theory," *IRE Trans.*, PGIT-4, 1954, pp. 119–145.

[38] T. G. Birdsall, "The Theory of Signal Detectability: ROC Curves and Their Character," Technical Report No. 177, Cooley Electronics Laboratory, University of Michigan, Ann Arbor, MI, 1973.

[39] W. J. Bangs and P. M. Schultheiss, "Space-Time Processing for Optimal Parameter Estimation," *Proceedings of NATO Advanced Study Institute on Signal Processing,* Loughborough, England, August 1972, pp. 577–589.

[40] L. W. Nolte, "Adaptive Processing: Time-Varying Parameters," *Proceedings of NATO Advanced Study Institute on Signal Processing,* Loughborough, England, August 1972, pp. 647–655.

[41] D. O. North, "Analysis of the Factors which Determine Signal/Noise Discrimination in Radar," Report PTR-6C, RCA Laboratories, 1943.

[42] L. A. Zadeh and J. R. Ragazzini, "Optimum Filters for the Detection of Signals in Noise," *IRE Proc.*, Vol. 40, 1952, pp. 1223–1231.

[43] G. L. Turin, "An Introduction to Matched Filters," *IRE Trans.*, Vol. IT-6, June 1960, pp. 311–330.

[44] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, New York, Wiley, 1968, Ch. 2.

[45] H. Cox, "Resolving Power and Sensitivity to Mismatch of Optimum Array Processors," *J. Acoust. Soc. Am.*, Vol., 54, No. 3, September 1973, pp. 771–785.

[46] H. L. Van Trees, "Optimum Processing for Passive Sonar Arrays," *IEEE 1966 Ocean Electronics Symposium,* August, Hawaii, pp. 41–65, 1966.

[47] L. W. Brooks and I. S. Reed, "Equivalence of the Likelihood Ratio Processor, the Maximum Signal-to-Noise Ratio Filter, and the Wiener Filter," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-8, No. 5, September 1972, pp. 690–691.

[48] A. M. Vural, "Effects of Perturbations on the Performance of Optimum/Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst.,* Vol. AES-15, No. 1, January 1979, pp. 76–87.

[49] R. T. Compton, "On the Performance of a Polarization Sensitive Adaptive Array," *IEEE Trans. Ant. & Prop.,* Vol. AP-29, September 1981, pp. 718–725.

[50] A. Singer, "Space vs. Polarization Diversity," *Wireless Review*, February 15, 1998, pp. 164–168.

[51] M. R. Andrews, P. P. Mitra, and R. deCarvalho, "Tripling the Capacity of Wireless Communications Using Electromagnetic Polarization," *Nature*, Vol. 409, January 18, 2001, pp. 316–318.

[52] R. E. Marshall and C. W. Bostian, "An Adaptive Polarization Correction Scheme Using Circular Polarization," *IEEE International Antennas and Propagation Society Symposium,* Atlanta, GA, June 1974, pp. 395–397.

[53] R. T. Compton, "The Tripole Antenna: An Adaptive Array with Full Polarization Flexibility," *IEEE Trans. Ant. & Prop.,* Vol. AP-29, November 1981, pp. 944–952.

[54] R. L. Haupt, "Adaptive Crossed Dipole Antennas Using a Genetic Algorithm," *IEEE Trans. Ant. & Prop.,* Vol. AP-52, August 2004, pp. 1976–1982.

[55] R. Nitzberg, "Effect of Errors in Adaptive Weights," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-12, No. 3, May 1976, pp. 369–373.

[56] J. P. Burg, "Three-Dimensional Filtering with an Array of Seismometers," *Geophysics*, Vol. 29, No. 5, October 1964, pp. 693–713.

[57] A. Ishide and R.T. Compton, Jr., "On grating Nulls in Adaptive Arrays," *IEEE Trans. on Ant. & Prop.,* Vol. AP-28, No. 4, July 1980, pp. 467–478.

PART II

Adaptive Algorithms

Chapter 3 showed that a variety of popular performance measures led to closely related optimum weight vector solutions. Consequently, the choice of a specific performance measure is not as important as the adaptive algorithm that adjusts the array weights, since the algorithm influences the speed of the array transient response and the hardware complexity. Part 2 presents a survey of algorithms and discusses the important performance characteristics of each one. In some cases, algorithms are tailored to particular signal conditions, whereas in other cases they handle a variety of signal environments. These algorithm characteristics provide the designer with a means for picking the algorithm based on convergence speed, operational signal environment, and hardware complexity.

CHAPTER

4

Gradient-Based Algorithms

Chapter Outline

Gradient algorithms are popular, because they are simple, easy to understand, and solve a large class of problems. The performance, $\Re(\mathbf{w})$, and adaptive weights determine the nature of the performance surface. When $\Re(\mathbf{w})$ is a quadratic function of the weight settings, then it is a bowl-shaped surface with a minimum at the "bottom of the bowl." In this case, local optimization methods, such as gradient methods, can find the bottom. In the event that the performance surface is irregular, having several relative optima or saddle points, then the transient response of the gradient-based minimum-seeking algorithms get stuck in a local minimum. The gradient-based algorithms considered in this chapter are as follows:

1. Least mean square (LMS)
2. Howells–Applebaum loop
3. Differential steepest descent (DSD)
4. Accelerated gradient (AG)
5. Steepest descent for power minimization

Variations of these algorithms come from introducing constraints into the adjustment rule, and one section develops the procedure for deriving such variations. Finally, changes in the modes of adaptation are discussed, illustrating how two-mode adaptation enhances the convergence.

4.1 INTRODUCTORY CONCEPTS

The method of steepest descent is the most common gradient algorithm applied to quadratic performance surfaces. Any quadratic performance surface has a unique minimum point that can be found by using a gradient-based algorithm.

4.1.1 The Quadratic Performance Surface

A bowl-shaped quadratic performance surface, like the mean square error (MSE) performance measure for the adaptive array of Figure 3-3, is the ideal playing field for a gradient-based algorithm. Recall from Chapter 3 that the array output signal is given by

$$y(t) = \mathbf{w}^T(t)\mathbf{x}(t) \tag{4.1}$$

Denoting the desired array response by $d(t)$, we may express the error signal as

$$e(t) = d(t) - y(t) = d(t) - \mathbf{w}^T(t)\mathbf{x}(t) \tag{4.2}$$

The square of the foregoing error signal is then

$$e^2(t) = d^2(t) - 2d(t)\mathbf{x}^T(t)\mathbf{w}(t) + \mathbf{w}^T(t)\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{w}(t) \tag{4.3}$$

The MSE is just the expected value of $e^2(t)$, or

$$\begin{aligned} E\{e^2(t)\} &= \xi[\mathbf{w}(t)] \\ &= \overline{d}^2(t) - 2\mathbf{r}_{xd}^T(t)\mathbf{w}(t) + \mathbf{w}^T(t)\mathbf{R}_{xx}(t)\mathbf{w}(t) \end{aligned} \tag{4.4}$$

where the overbar denotes expected value, $\mathbf{r}_{xd}(t)$ is given by (3.52), and $\mathbf{R}_{xx}(t)$ is given by (3.13). When the input signals are statistically stationary, then $\mathbf{r}_{xd}$ and $\mathbf{R}_{xx}$ are also stationary, and there is no need to write these quantities as a function of time. In nonstationary signal environments, however, the notation $\mathbf{r}_{xd}(t)$ and $\mathbf{R}_{xx}(t)$ is required.

The MSE in (4.4) is a quadratic function of the weight vector $\mathbf{w}(t)$. In the nonstationary case, the bottom of the bowl as well as its curvature and orientation change. The analysis of time-varying adaptive performance signal statistics is beyond the scope of this book.

4.1.2 The Method of Steepest Descent

If the statistics of the signal environment are perfectly known, then the gradient at any point on the performance surface can be calculated. The gradient of (4.4) with respect to the weight vector is [1]

$$\{\xi[\mathbf{w}(t)]\} = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}\mathbf{w}(t) \tag{4.5}$$

It was shown in Chapter 3 that the minimum is the Wiener solution

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{4.6}$$

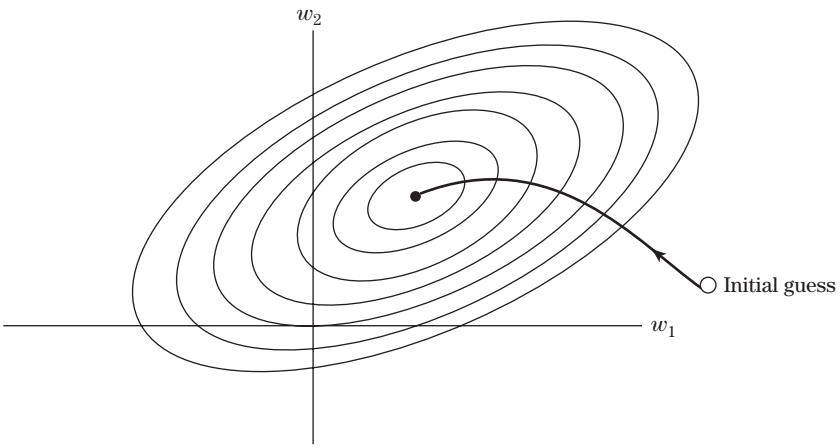


FIGURE 4-1 ■ Steepest descent with very small step size (overdamped case).

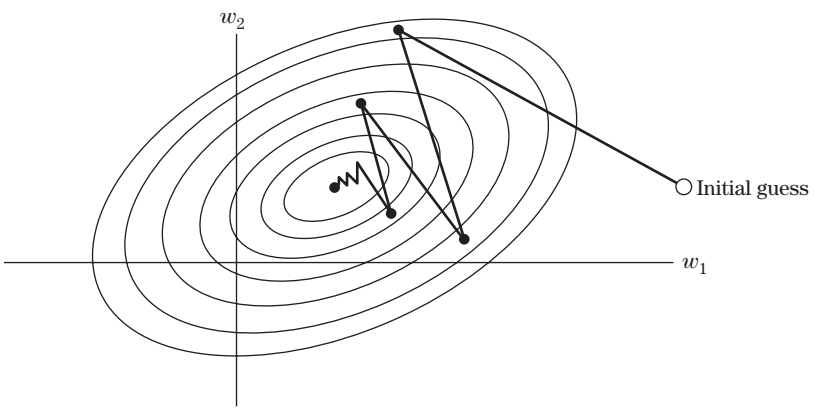


FIGURE 4-2 ■ Steepest descent with large step size (underdamped case).

On substituting (4.6) into (4.4), the minimum MSE is then found to be

$$\xi_{\min} = \overline{d}^2(t) - \mathbf{w}_{\text{opt}}^T \mathbf{r}_{xd} \tag{4.7}$$

The method of steepest descent begins with an initial guess of the weight vector components. Having selected a starting point, we then calculate the gradient vector and perturb the weight vector in the opposite direction (i.e., in the direction of the steepest downward slope). Contour plots of a quadratic performance surface (corresponding to a two-weight adjustment problem) are shown in Figures 4-1 and 4-2. In these figures the MSE is measured along a coordinate normal to the plane of the paper. The ellipses in these figures are contours of constant MSE. The gradient is orthogonal to these constant value contours (pointing in the steepest direction) at every point on the performance surface. If the steepest descent uses small steps, it is "overdamped," and the path taken to the bottom appears continuous as shown in Figure 4-1. If the steepest descent uses large steps, it is "underdamped," and each step is normal to the error contour as shown in Figure 4-2.

The discrete form of the method of steepest descent is [1]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta_s \quad \left[\overline{e}^2(k)\right] \tag{4.8}$$

where

$\mathbf{w}(k)$ = old weight vector guess at time kT
$\mathbf{w}(k+1)$ = new weight vector guess at time $(k+1)T$
$\overline{[e^2(k)]}$ = gradient vector of the MSE determining the direction in which to move from $\mathbf{w}(k)$
Δ_s = step size

Substituting the gradient of (4.5) into (4.8) then yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\Delta_s(\mathbf{R}_{xx}\mathbf{w}(k) - \mathbf{r}_{xd}) \tag{4.9}$$

4.1.3 Feedback Model of Steepest Descent

The transient behavior of the method of steepest descent yields valuable insight into the behavior of the LMS algorithm. The only difference between the two weight adjustment algorithms is that with steepest descent the signal environment statistics are perfectly known (so the gradient at any point can be exactly determined), whereas the LMS algorithm signal statistics are unknown (although here they are assumed to be stationary) and therefore must be estimated. The first step in determining the transient behavior of the method of steepest descent is to formulate a feedback model of the weight adjustment relationship.

Figure 4-3 is the feedback flow graph of (4.8) and (4.9). The symbol Z^{-1} is the Z-transform representation [2–5] of a unit (one iteration cycle) time delay, and $Z^{-1}\mathbf{I}$ is the matrix transfer function of a unit delay branch. This flow graph represents a first-order multidimensional sampled-data control loop.

By setting the initial weight vector $\mathbf{w}(0)$ equal to the initial guess in the flow graph, the resulting sequence of $\mathbf{w}(k)$ behaves exactly as in the local minimization weight adjustment algorithm. Since the "output" of the flow graph model is the current weight vector $\mathbf{w}(k)$, the flow graph model determines the transient behavior of the weight adjustment algorithm.

Each transfer function appearing in the flow graph of Figure 4-3 is a diagonal matrix except for the feedback branch denoted by $2\mathbf{R}_{xx}$. This branch matrix in general has off-diagonal elements, since the input signals are usually mutually correlated. Consequently, transients cross-couple from one component of the weight vector to the next, thereby complicating the study of transient behavior. A remedy is to diagonalize the flow graph and eliminate such cross-coupling effects from consideration: the diagonalization then enables one to consider the natural modes of behavior of the flow graph by merely introducing a coordinate transformation.

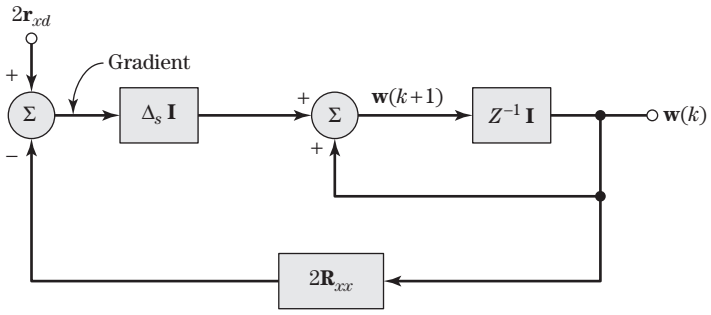


FIGURE 4-3 ■ Feedback model of steepest descent.

To diagonalize the flow graph of Figure 4-3, consider the expression for the MSE given by (4.4). Using $\mathbf{w}_{\text{opt}}$ and $\xi_{\min}$ in (4.6) and (4.7), the MSE becomes

$$E\{e^2(k)\} = \xi(k) = \xi_{\min} + [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}]^T \mathbf{R}_{xx} [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}] \tag{4.10}$$

Since the matrix $\mathbf{R}_{xx}$ is real, symmetric, and positive definite (for real variables), it is diagonalized by means of a unitary transformation matrix $\mathbf{Q}$ so that

$$\mathbf{R}_{xx} = \mathbf{Q}^{-1} \mathbf{\Lambda} \mathbf{Q} \tag{4.11}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, and $\mathbf{Q}$ is the modal square matrix of eigenvectors. If $\mathbf{Q}$ is constructed from normalized eigenvectors, then it is orthonormal so that $\mathbf{Q}^{-1} = \mathbf{Q}^T$, and the MSE becomes

$$\xi(k) = \xi_{\min} + [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}]^T \mathbf{Q}^T \mathbf{\Lambda} \mathbf{Q} [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}] \tag{4.12}$$

Now define

$$\mathbf{Q}\mathbf{w}(k) \triangleq \mathbf{w}'(k) \tag{4.13}$$

$$\mathbf{Q}\mathbf{w}_{\text{opt}} \triangleq \mathbf{w}'_{\text{opt}} \tag{4.14}$$

Equation (4.12) can then be rewritten as

$$\xi(k) = \xi_{\min} + [\mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}]^T \mathbf{\Lambda} [\mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}] \tag{4.15}$$

$\mathbf{Q}$ projects $\mathbf{w}(k)$ into the primed coordinates [resulting in $\mathbf{w}'(k)$]. Since $\mathbf{\Lambda}$ is a diagonal matrix and $\xi(k)$ is a quadratic performance surface, the primed coordinates comprise the principal axes of the quadratic performance surface. The feedback model of Figure 4-3 expresses all quantities in terms of the primed coordinate system. The resulting equivalent feedback model of Figure 4-4 eliminates all cross-couplings that previously existed within the feedback paths.

The steepest descent algorithm in Figure 4-4 is composed of the natural modes of the flow graph. The transients of each mode are isolated (since each of the primed coordinates has its own natural mode), and the natural behavior of steepest descent is completely explored by considering the behavior of a single primed coordinate.

An isolated one-dimensional feedback model for the pth normal coordinate is shown in Figure 4-5. The pulse transfer function of this closed-loop feedback system is [1]

$$\frac{\mathbf{w}'_p(z)}{r'_p(z)} = \frac{\Delta_s Z^{-1}}{1 + (1 - 2\Delta_s \lambda_p) Z^{-1}} \tag{4.16}$$

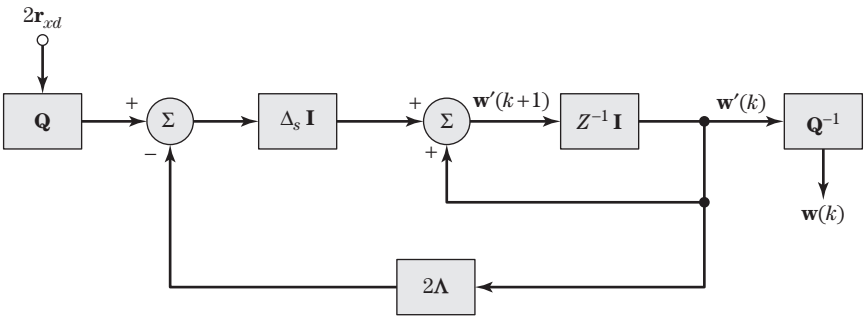


FIGURE 4-4 ■ Diagonalized version of feedback model of steepest descent using normal (primed) coordinates.

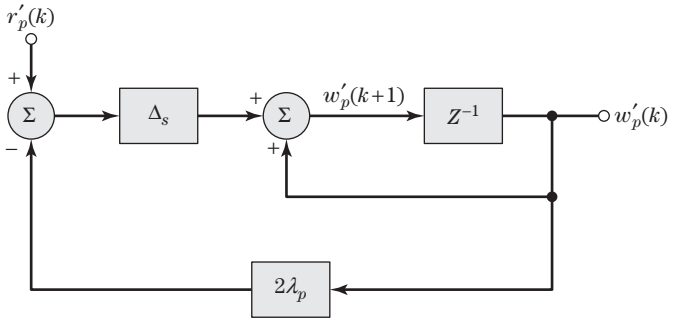


FIGURE 4-5 ■ One-dimensional feedback model for the *p*th normal coordinate of steepest descent.

where λ_p is the pth eigenvalue of $\mathbf{R}_{xx}$. The impulse response of (4.16) is found by letting $r'_p(z) = 1$ and taking the inverse Z-transform of the resulting output $\mathcal{Z}^{-1}\{w'_p(z)\}$. It follows that the impulse response is of the form

$$w'_p(kT) = \text{constant} \times e^{-\alpha_p(kT)}$$

where

$$\alpha_p = -\frac{1}{T} ln(1 - 2\Delta_s\lambda_p) \tag{4.17}$$

and T = one iteration period. The time response of (4.17) is a stable system when

$$|1 - 2\Delta_s\lambda_p| < 1 \tag{4.18}$$

Equation (4.18) is satisfied when

$$\Delta_s > 0 \quad \text{and} \quad |\Delta_s\lambda_p| < 1 \tag{4.19}$$

Since $\mathbf{R}_{xx}$ is positive definite, $\lambda_p > 0$ for all p. Consequently, the stability of the multidimensional flow graph of Figure 4-4 is guaranteed if and only if $\lambda_p = \lambda_{\max}$ in (4.19) and

$$\Delta_s > 0 \quad \text{and} \quad |\Delta_s\lambda_{\max}| < 1 \tag{4.20}$$

The stability of the steepest descent adaptation process is therefore guaranteed so long as

$$\frac{1}{\lambda_{\max}} > \Delta_s > 0 \tag{4.21}$$

4.2 | THE LMS ALGORITHM

When the operational environment signal statistics are stationary but unknown (a typical situation), then the gradient of the performance surface at any point must be estimated. The LMS algorithm introduced by Widrow has proven particularly useful for a quadratic performance function [6–10]. It is worthwhile noting that the LMS algorithm requires a reference signal, $d(t)$, to generate the error signal given by (4.2). The desired signal in a communications system is usually present, so the actual signal is used as the reference signal. In systems where the desired signal is usually not present (as in radar or

sonar systems), it is pointless to try to generate a fictitious desired signal. Thus, the LMS algorithm described here is usually employed to improve communications system performance. The LMS algorithm is exactly like the method of steepest descent except that now changes in the weight vector are made in the direction given by an estimated gradient vector instead of the actual gradient vector. In other words, changes in the weight vector are expressed as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta_s \, \hat{} \, [\xi(k)] \tag{4.22}$$

where

$\mathbf{w}(k) =$ weight vector before adaptation step
$\mathbf{w}(k+1) =$ weight vector after adaptation step
$\Delta_s =$ step size that controls rate of convergence and stability
$\hat{} \, [\xi(k)] =$ estimated gradient vector of ξ with respect to $\mathbf{w}$

The adaptation process described by (4.22) attempts to find a solution as close as possible to the Wiener solution given by (4.6). It is tempting to try to solve (4.6) directly, but such an approach has several drawbacks:

1. Computing and inverting an $N \times N$ matrix when the number of weights N is large becomes more challenging as input data rates increase.
2. This method may require up to $[N(N+3)]/2$ autocorrelation and cross-correlation measurements to find the elements of $\mathbf{R}_{xx}$ and $\mathbf{r}_{xd}$. In many practical situations, such measurements must be repeated whenever the input signal statistics change.
3. Implementing a direct solution requires setting weight values with high accuracy in open loop fashion, whereas a feedback approach provides self-correction of inaccurate settings, thereby giving tolerance to hardware errors.

To obtain the estimated gradient of the MSE performance measure, take the gradient of a single time sample of the squared error as follows:

$$\hat{} \, _k = \quad [\xi(k)] = 2e(k) \quad [e(k)] \tag{4.23}$$

Since

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w} \tag{4.24}$$

it follows that

$$[e(k)] = \quad [d(k) - \mathbf{x}^T(k)\mathbf{w}] = -\mathbf{x}(k) \tag{4.25}$$

so that

$$\hat{} \, _k = -2e(k)\mathbf{x}(k) \tag{4.26}$$

It is easy to show that the gradient estimate given by (4.26) is unbiased by considering the expected value of the estimate and comparing it with the gradient of the actual MSE. The expected value of the estimate is given by

$$E\{\hat{} \, _k\} = -2E\{\mathbf{x}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\} \tag{4.27}$$

$$= -2[\mathbf{r}_{xd}(k) - \mathbf{R}_{xx}(k)\mathbf{w}(k)] \tag{4.28}$$

Now consider the MSE

$$\xi[\mathbf{x}(k)] = E[d^2(k)] + \mathbf{w}^T \mathbf{R}_{xx}(k)\mathbf{w} - 2\mathbf{w}^T \mathbf{r}_{xd}(k) \tag{4.29}$$

Differentiating (4.29) with respect to $\mathbf{w}$ yields the gradient $\{\xi[\mathbf{w}(k)]\}$ as

$$\{\xi[\mathbf{w}(k)]\} = 2\mathbf{R}_{xx}(k)\mathbf{w}(k) - 2\mathbf{r}_{xd}(k) \tag{4.30}$$

Comparing (4.28) and (4.30) reveals that

$$E\{\hat{\ }_k\} = \{\xi[\mathbf{w}(k)]\} \tag{4.31}$$

so the expected value of the estimated gradient equals the true value of the gradient of the MSE.

Substituting the estimated gradient of (4.26) into the weight adjustment rule of (4.22) then yields the weight control rule

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\Delta_s e(k)\mathbf{x}(k) \tag{4.32}$$

The LMS algorithm given by (4.32) can be rewritten for complex quantities as

$$\frac{\mathbf{w}(k+1) - \mathbf{w}(k)}{\Delta t} = 2k_s e(k)\mathbf{x}^*(k) \tag{4.33}$$

where Δt is the elapsed time between successive iterations, and $\Delta_s = k_s \Delta t$. In the limit as $\Delta t \to 0$, (4.33) yields an equivalent differential equation representation of the LMS algorithm that is appropriate for use in continuous systems as

$$\frac{d\mathbf{w}(t)}{dt} = 2k_s e(t)\mathbf{x}^*(t) \tag{4.34}$$

Equation (4.34) can also be written as

$$\mathbf{w}(t) = 2k_s \int_0^t e(\tau)\mathbf{x}^*(\tau)d\tau + \mathbf{w}(0) \tag{4.35}$$

A block diagram representation of the weight adjustment rule represented by (4.35) is shown in Figure 4-6.

The discrete version of (4.34) is given by (4.33) and is more commonly written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2k_s \Delta t e(k)\mathbf{x}^*(k) \tag{4.36}$$

A block diagram representation of the weight adjustment rule represented by (4.35) is illustrated in Figure 4-7.

4.2.1 Convergence to the Wiener Solution

Assume that the time between successive iterations of the LMS algorithm is long enough so that the signal vectors $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$ are uncorrelated. From (4.32) it follows that $\mathbf{w}(k)$ is a function of only $\mathbf{x}(k-1), \mathbf{x}(k-2), \ldots, \mathbf{x}(0)$, and $\mathbf{w}(0)$, where the successive input signal vectors are uncorrelated so that $\mathbf{w}(k)$ is independent of $\mathbf{x}(k)$. It will now be shown that for a stationary input signal process meeting these conditions, the expected value of the weight vector $E\{\mathbf{w}(k)\}$ converges to the Wiener solution given by (4.6).

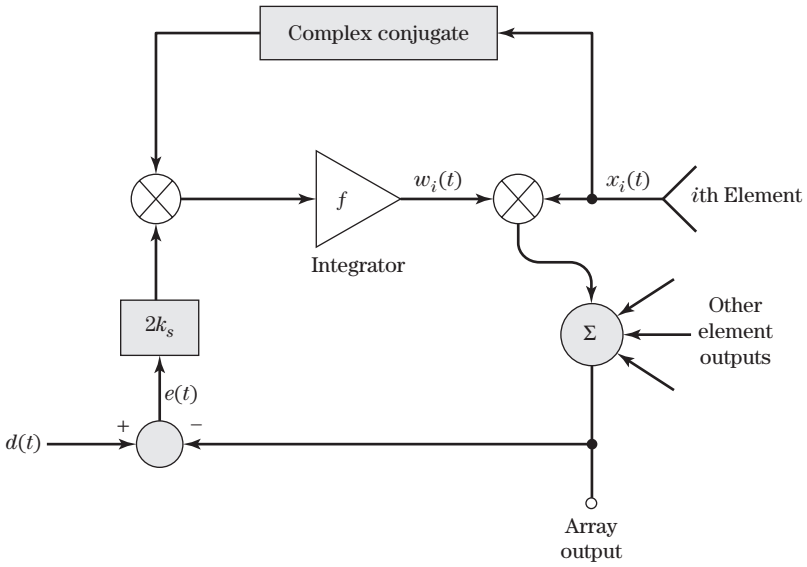


FIGURE 4-6 ■ Analog realization of the LMS weight adjustment algorithm.

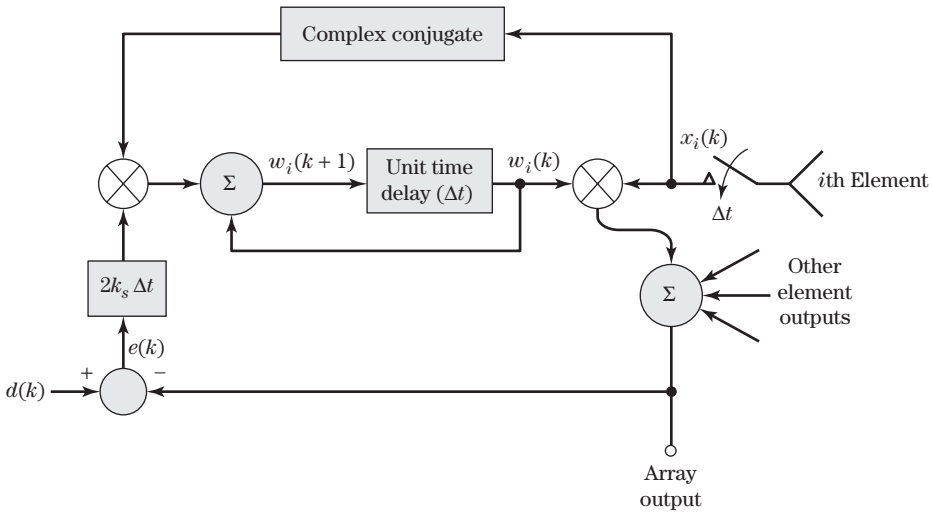


FIGURE 4-7 ■ Digital realization of the LMS weight adjustment algorithm.

Taking the expected value of both sides of (4.36), there results

$$E\{\mathbf{w}(k+1)\} = E\{\mathbf{w}(k)\} + 2k_s\Delta t \cdot E\{\mathbf{x}^*(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\} \tag{4.37}$$

Now let

$$E\{\mathbf{x}^*(k)d(k)\} = \mathbf{r}_{xd} \tag{4.38}$$

$$E\{\mathbf{x}^*(k)\mathbf{x}^T(k)\} = \mathbf{R}_{xx} \tag{4.39}$$

Consequently, (4.37) is rewritten as

$$\begin{aligned} E\{\mathbf{w}(k+1)\} &= E\{\mathbf{w}(k)\} - 2k_s\Delta t\mathbf{R}_{xx}E\{\mathbf{w}(k)\} + 2k_s\Delta t\mathbf{r}_{xd} \\ &= [\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]E\{\mathbf{w}(k)\} + 2k_s\Delta t\mathbf{r}_{xd} \end{aligned} \tag{4.40}$$

Starting with an initial guess $\mathbf{w}(0)$, the $(k+1)$th iteration of (4.40) yields

$$E\{\mathbf{w}(k+1)\} = [\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]^{(k+1)}\mathbf{w}(0) + 2k_s\Delta t\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]^i\mathbf{r}_{xd} \tag{4.41}$$

Diagonalizing (4.41) by using (4.11) to obtain the normal form results in

$$E\{\mathbf{w}(k+1)\} = \mathbf{Q}^{-1}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^{(k+1)}\mathbf{Q}\mathbf{w}(0) + 2k_s\Delta t\mathbf{Q}^{-1}\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^i\mathbf{Q}\mathbf{r}_{xd} \tag{4.42}$$

When the magnitude of all the terms in the diagonal matrix $[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]$ are less than one, then

$$\lim_{k\to\infty}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^{(k+1)} \to 0 \tag{4.43}$$

Therefore, the first term of (4.42) vanishes after a sufficient number of iterations, and the summation factor in the second term of (4.42) becomes

$$\lim_{k\to\infty}\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^i = \frac{1}{2k_s\Delta t}\mathbf{\Lambda}^{-1} \tag{4.44}$$

Therefore, after a sufficient number of iterations, (4.42) yields

$$\begin{aligned}\lim_{k\to\infty}E\{\mathbf{w}(k+1)\} &= 2k_s\Delta t\mathbf{Q}^{-1}\left(\frac{1}{2k_s\Delta t}\mathbf{\Lambda}^{-1}\right)\mathbf{Q}\mathbf{r}_{xd}\\ &= \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}\end{aligned} \tag{4.45}$$

This result shows that the expected value of the weight vector in the LMS algorithm does converge to the Wiener solution after a sufficient number of iterations.

Since all the eigenvalues in $\mathbf{\Lambda}$ are positive, it follows that all the terms in the aforementioned diagonal matrix, $\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}$, have a magnitude less than one provided that

$$\begin{aligned}|1 - 2k_s\Delta t\lambda_{\max}| &< 1\\ \frac{1}{\lambda_{\max}} &> k_s\Delta t > 0\end{aligned} \tag{4.46}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{R}_{xx}$. The convergence condition (4.46) is exactly the same as the stability condition (4.21) for the noise-free steepest descent feedback model.

The foregoing condition on k_s for convergence of the mean value of the LMS algorithm relates to the total input signal power P_{IN}, as described henceforth. Since $\lambda_{\max}$ satisfies the inequality

$$\lambda_{\max} \le \text{trace}[\mathbf{R}_{xx}] \tag{4.47}$$

where

$$\text{trace}[\mathbf{R}_{xx}] = E\{\mathbf{x}^{\dagger}(k)\mathbf{x}(k)\} = \sum_{i=1}^{N} E\{|x_i|^2\} \triangleq P_{\text{IN}} \tag{4.48}$$

then the convergence condition (4.46) is assured if

$$\frac{1}{P_{\text{IN}}} > k_s \Delta t > 0 \tag{4.49}$$

These LMS algorithm convergence results assumed that successive input signal samples are independent. This independence assumption is probably overly restrictive, since Griffiths [11] presented experimental results that show that adaptation using highly correlated successive samples also converges to the Wiener solution, although the resulting steady-state MSE is slightly higher than what results for statistically independent successive samples. For some applications, mean squared convergence and its associated stability properties may be of concern, in which case more stringent conditions on k_s must be satisfied [12].

4.2.2 Transient Response Characteristics for LMS Adaptation

In normal coordinates, the adaptive weight transients consist of sums of exponentials with time constants given by

$$\tau_p = \frac{1}{2(k_s \Delta t)\lambda_p}, \quad p = 1, 2, \ldots, N \tag{4.50}$$

where λ_p is the pth eigenvalue of the correlation matrix $\mathbf{R}_{xx}$. Since τ_p is inversely proportional to λ_p, the transient response is fastest for strong signals (large λ_p) and slowest for weak signals (small λ_p). Thus, the LMS algorithm convergence time depends on the eigenvalue spread in $\mathbf{R}_{xx}$. The exponential having the longest time constant (corresponding to the smallest normal coordinate signal power) determines the transient response of the LMS algorithm. Often there is no choice for the value of the constant k_s that represents a good compromise between the various eigenvalues that will yield a desirably short transient period of operation.

Figure 4-8 shows a contour plot of a quadratic performance surface corresponding to two widely diverse eigenvalues. The highly elongated MSE contours in Figure 4-8 result in many adaptive iterations before weight values become acceptably close to the desired Wiener solution. In the event that all the eigenvalues are equal, then all the time constants are equal, and

$$\tau = \frac{1}{2(k_s \Delta t)\lambda} \tag{4.51}$$

A "learning curve" that plots the expected value of the performance measure at each stage of the learning process as a function of the number of adaptation iterations provides a convenient way of monitoring the progress of an adaptive process. It has just been shown that the underlying transient behavior of the adaptive weights has an exponential nature. Since the MSE is a quadratic function of the weight values, the transients in the MSE function are also exponential.

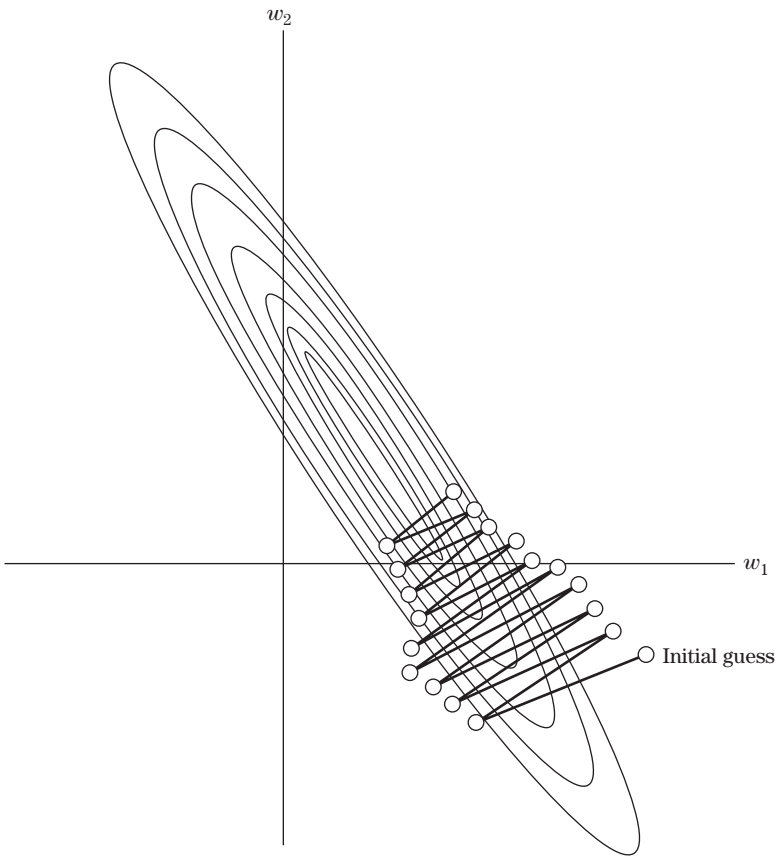


FIGURE 4-8 ■ Steepest descent transient response with widely diverse eigenvalues.

Since the square of an exponential function is an exponential having half the time constant of the original exponential function, it follows that when all the time constants are equal the MSE learning curve is an exponential having the time constant

$$\tau_{\text{MSE}} = \frac{\tau}{2} = \frac{1}{4(k_s \Delta t)\lambda} \tag{4.52}$$

In general, of course, the eigenvalues of $\mathbf{R}_{xx}$ are unequal so that

$$\tau_{p_{\text{MSE}}} = \frac{\tau_p}{2} = \frac{1}{4(k_s \Delta t)\lambda_p} \tag{4.53}$$

where $\tau_{p_{\text{MSE}}}$ is the time constant for the MSE learning curve, τ_p is the time constant in the weights, and λ_p is the eigenvalue of the pth normal mode. The adaptive process uses one signal data sample/iteration, so the time constant expressed in terms of the number of data samples is

$$T_{p_{\text{MSE}}} = \tau_{p_{\text{MSE}}} \tag{4.54}$$

Plots of actual experimental learning curves look like noisy exponentials—an effect due to the inherent noise that is present in the adaptation process. A slower adaptation rate (i.e., the smaller the magnitude of k_s) has a smaller noise amplitude that corrupts the learning curve.

4.2.3 Weight Misadjustment During LMS Adaptation

Adaptation speed of the LMS algorithm depends on two factors: (1) the weight adjustment step size and (2) the statistical averages sample size. If a large step size is selected for the weight adjustment process, then the excursions in successive weight values will be large, resulting in transient behavior like the underdamped case shown in Figure 4-2. If a small number of samples is used to estimate the statistical averages, then the time elapsed in obtaining such averages is small, but the quality of the resulting estimates is low. In general, a fast adaptive algorithm has a poor steady-state performance.

Since the input signal statistics are not known a priori, the minimum MSE is not the Wiener solution. The "misadjustment" M compares the actual MSE with the optimum Wiener solution

$$M \triangleq \frac{[\xi_{\text{actual}} - \xi_{\min}]}{\xi_{\min}} \tag{4.55}$$

where $\xi = E\{e^2\}$. The LMS algorithm misadjustment can be evaluated for a specified value of $k_s \Delta t$ by considering the noise associated with the gradient-estimation process.

Assume that the adaptive process converged to a steady state in the neighborhood of the MSE surface minimum point. The gradient-estimation noise of the adaptive algorithm at the minimum point (where the true gradient is zero) is just the gradient estimate itself. Therefore, the gradient noise vector $\mathbf{g}$ is given by

$$\mathrm{g}(k) = \hat{} \ (k) = -2e(k)\mathbf{x}(k) \tag{4.56}$$

The covariance of this estimation noise is given by

$$\text{cov}[\mathbf{g}(k)] = E\{\mathbf{g}(k)\mathbf{g}^T(k)\} = 4E\{e^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\} \tag{4.57}$$

When the weight vector is optimized ($\mathbf{w}(k) = \mathbf{w}_{\text{opt}}$), then the error $e(k)$ is uncorrelated with the input vector $\mathbf{x}(k)$. If $e(k)$ and $\mathbf{x}(k)$ are Gaussian processes, then not only are they uncorrelated at the minimum point of the MSE surface, but they are also statistically independent. With these conditions (4.57) becomes

$$\text{cov}[\mathbf{g}(k)] = 4E\{e^2(k)\}E\{\mathbf{x}(k)\mathbf{x}^T(k)\} = 4\xi_{\min}\mathbf{R}_{xx} \tag{4.58}$$

In the primed normal coordinates, the previous covariance can be written as

$$\text{cov}[\mathbf{g}'(k)] = \mathbf{Q}\text{cov}[\mathbf{g}(k)]\mathbf{Q}^{-1} = 4\xi_{\min}\mathbf{\Lambda} \tag{4.59}$$

Adaptation based on noisy gradient estimates results in noise in the weight vector. Recall that the noise-free method of steepest descent is described by the iterative relation

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta_s[- \quad (k)] \tag{4.60}$$

where Δ_s is the constant that controls stability and rate of convergence, and $\quad(k)$ is the gradient at the point on the performance surface corresponding to $\mathbf{w} = \mathbf{w}(k)$. Following Widrow and McCool [13], subtract $\mathbf{w}_{\text{opt}}$ from both sides of (4.60), and define $\mathbf{v}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_{\text{opt}}$ to obtain

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s[- \quad (k)] \tag{4.61}$$

With estimated gradients instead of exact gradients, (4.61) can be rewritten as

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s(-\hat{\ }(k)) = \mathbf{v}(k) + \Delta_s[- \quad (k) - \mathbf{g}(k)] \tag{4.62}$$

Now since $\quad(k)$ is given by (4.5), it follows that

$$(k) = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}[\mathbf{w}_{\text{opt}} + \mathbf{v}(k)] = 2\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.63}$$

Consequently, (4.62) can be written as

$$\mathbf{v}(k+1) = (\mathbf{I} - 2\Delta_s\mathbf{R}_{xx})\mathbf{v}(k) - \Delta_s\mathbf{g}(k) \tag{4.64}$$

which represents a first-order vector difference equation with a stochastic driving function—$\Delta_s\mathbf{g}(k)$. Multiplying (4.64) by $\mathbf{Q}$ produces

$$\mathbf{v}'(k+1) = (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k) - \Delta_s\mathbf{g}'(k) \tag{4.65}$$

After initial transients have died out and the steady state is reached, $\mathbf{v}'(k)$ responds to the stationary driving function $-\Delta_s\mathbf{g}'(k)$ in the manner of a stationary random process. The absence of any cross-coupling in the primed normal coordinate system means that the components of both $\mathbf{g}'(k)$ and $\mathbf{v}'(k)$ are mutually uncorrelated, and the covariance matrix of $\mathbf{g}'(k)$ is therefore diagonal. To find the covariance matrix of $\mathbf{v}'(k)$ consider

$$\begin{aligned}\mathbf{v}'(k+1)\mathbf{v}'^T(k+1) = {} & (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s\mathbf{\Lambda}) \\ & + \Delta_s^2\mathbf{g}'(k)\mathbf{g}'^T(k) - \Delta_s(\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k)\mathbf{g}'^T(k) \\ & - \Delta_s\mathbf{g}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\end{aligned} \tag{4.66}$$

Taking expected values of both sides of (4.66) (and noting that $\mathbf{v}'(k)$ and $\mathbf{g}'(k)$ are uncorrelated since $\mathbf{v}'(k)$ is affected only by gradient noise from previous iterations), we find

$$\begin{aligned}\text{cov}\,[\mathbf{v}'(k)] &= (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\text{cov}\,[\mathbf{v}'(k)](\mathbf{I} - 2\Delta_s\mathbf{\Lambda}) + \Delta_s^2\text{cov}\,[\mathbf{g}'(k)] \\ &= \Delta_s^2\left[4\Delta_s\mathbf{\Lambda} - 4\Delta_s^2\mathbf{\Lambda}^2\right]^{-1}\text{cov}\,[\mathbf{g}'(k)]\end{aligned} \tag{4.67}$$

In practical applications, the LMS algorithm uses a small value for Δ_s, so that

$$\Delta_s\mathbf{\Lambda} \ll \mathbf{I} \tag{4.68}$$

With (4.68) satisfied, the squared terms involving $\Delta_s\mathbf{\Lambda}$ in (4.67) may be neglected, so

$$\text{cov}\,[\mathbf{v}'(k)] = \frac{\Delta_s}{4}\mathbf{\Lambda}^{-1}\text{cov}\,[\mathbf{g}'(k)] \tag{4.69}$$

Using (4.59), we find

$$\text{cov}\,[\mathbf{v}'(k)] = \frac{\Delta_s}{4}\mathbf{\Lambda}^{-1}(4\xi_{\min}\mathbf{\Lambda}) = \Delta_s\xi_{\min}\mathbf{I} \tag{4.70}$$

Therefore, the covariance of the steady-state noise in the weight vector (near the minimum point of the MSE surface) is

$$\text{cov}\,[\mathbf{v}(k)] = \Delta_s\xi_{\min}\mathbf{I} \tag{4.71}$$

Without noise in the weight vector, the actual MSE experienced would be $\xi_{\min}$. The presence of noise in the weight vector causes the steady-state weight vector solution to randomly meander about the minimum point. This random meandering results in an "excess" MSE— that is, an MSE that is greater than $\xi_{\min}$. Since

$$\xi(k) = \overline{d}^2(k) - 2\mathbf{r}_{xd}^T\mathbf{w}(k) + \mathbf{w}^T(k)\mathbf{R}_{xx}\mathbf{w}(k) \tag{4.72}$$

where

$$\xi_{\min} = \overline{d}^2(k) - \mathbf{w}_{\text{opt}}^T\mathbf{r}_{xd} \tag{4.73}$$

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{4.74}$$

It follows that (4.72) can be rewritten as (also see (4.10))

$$\xi(k) = \xi_{\min} + \mathbf{v}^T(k)\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.75}$$

In terms of the primed normal coordinates, (4.75) can be rewritten as

$$\xi(k) = \xi_{\min} + \mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k) \tag{4.76}$$

It immediately follows from (4.76) that the average excess MSE is

$$E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} = \sum_{p=1}^{N}\lambda_p E\left\{\left[\mathbf{v}'_p(k)\right]^2\right\} \tag{4.77}$$

Using (4.70) to recognize that $E\{[\mathbf{v}'_p(k)]^2\}$ is just $\Delta_s\xi_{\min}$ for each p, we see it then follows that

$$\begin{aligned} E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} &= \Delta_s\xi_{\min}\sum_{p=1}^{N}\lambda_p \\ &= \Delta_s\xi_{\min}tr(\mathbf{R}_{xx}) \end{aligned} \tag{4.78}$$

The misadjustment in the LMS algorithm is therefore given by

$$M = \frac{E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\}}{\xi_{\min}} = \Delta_s tr(\mathbf{R}_{xx}) \tag{4.79}$$

Since $\Delta_s = k_s\Delta t$, (4.79) emphasizes the fact that the degree of misadjustment experienced with the LMS algorithm can be controlled merely by adjusting k_s. When the step size is decreased, the time required to reach the steady-state condition increases, so there is a trade-off between the misadjustment and the adaptation speed.

The LMS algorithm misadjustment can also be expressed in a manner that gives insight into the relationship between misadjustment and adaptation speed. From (4.53) it follows that

$$\Delta_s\lambda_p = \frac{1}{4\tau_{p_{\text{MSE}}}} \tag{4.80}$$

Furthermore

$$\Delta_s tr(\mathbf{R}_{xx}) = \Delta_s \sum_{p=1}^{N} \lambda_p = \sum_{p=1}^{N} \left(\frac{1}{4\tau_{p_{\text{MSE}}}} \right) = \frac{N}{4} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} \tag{4.81}$$

where

$$\left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} \triangleq \frac{1}{N} \sum_{p=1}^{N} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right) \tag{4.82}$$

Consequently, the misadjustment can be written as

$$M = \frac{N}{4} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} = \frac{N}{4} \left(\frac{1}{T_{p_{\text{MSE}}}} \right)_{\text{av}} \tag{4.83}$$

where $T_{p_{\text{MSE}}}$ is the learning curve time constant in units of the number of data samples.

4.2.4 Practical Considerations for LMS Adaptation

Generation of the error signal in LMS adaptation requires an appropriate desired signal. If the desired signal is the signal itself, then the adaptive array output reproduces the signal in the best MSE sense and nearly eliminates the noise. As a practical matter, the signal is not available for adaptation purposes—indeed, if it were available there would be no need for a receiver and a receiving array.

An LMS algorithm artificially injects a known signal termed the *reference signal* or *pilot signal* for the desired signal. The pilot signal has the same (or similar) directional and spectral characteristics as those of the desired signal. These directional and spectral characteristics may sometimes be known a priori, but usually only estimates of these parameters are available. Many practical communication systems derive the reference signal from the array output—a practice that requires a high degree of compatibility between the signaling waveforms and the adaptive array. In general, it is not feasible to simply put an adaptive array in any arbitrary communication system because of the following reasons [14]:

1. The adaptive array weights are random processes that modulate the desired signal; consequently, either the desired signal waveforms or the adaptive algorithm must be chosen so this modulation does not impair the communication system effectiveness.
2. The desired signal and interference signal waveforms must be different in some respect, so this known difference can be exploited to enable the adaptive array to distinguish these two signal classes.
3. A practical method for reference-signal generation must be available.

The reference signal needs to satisfy only the following criteria [14]:

1. The reference signal must be highly correlated with the desired signal at the array output.
2. The reference signal must be uncorrelated with any interference signal components appearing at the array output.

If these two correlation properties are satisfied, then the adaptive array behaves in the desired manner, since only the correlation between the reference signal and the element signals $x_i(t)$ affects the adaptive weights. The impact of any phase shift occurring in

the network responsible for generating the reference signal (when the reference signal is derived from the array output) is discussed in [15].

LMS algorithm adaptation with an injected pilot signal causes the array to form a beam in the pilot-signal direction. This array beam has a flat spectra response and linear phase shift characteristic within the passband defined by the spectral characteristic of the pilot signal. Furthermore, directional noise incident on the array manifests as correlated noise components that the array will respond by producing beam pattern nulls in the noise direction within the array passband.

Since injection of the pilot signal could "block" the receiver (by rendering it insensitive to the actual signal of interest), mode-dependent adaptation schemes have been devised to overcome this difficulty. Two such adaptation algorithms are discussed in the following section.

4.2.5 One-Mode and Two-Mode LMS Adaptation

Figure 4-9 illustrates a practical two-mode method [10] for providing a pilot signal to form the array beam and then switching the pilot signal off to adapt to the inputs to eliminate noise. The ideal time delays $\delta_1, \delta_2, \ldots, \delta_N$ are selected to produce a set of input signals that appear to be a radiated plane wave from the desired direction. The adaptive processor inputs are connected either to the actual sensor element outputs (during adaptation to eliminate noise) or to the set of delayed signals obtained from the pilot signal generator and the selected time-delay elements (to preserve the main lobe in the desired direction).

During adaptation, all signals delivered to the adaptive processor are sensor element outputs derived from the actual noise field. The adaptation process in this mode tends to eliminate all received signals since the desired response signal has been set to zero.

To preserve the main beam in the desired direction during adaptation, the input signals to the adaptive processor are derived from the pilot signal. For example, if a sinusoidal pilot signal having frequency f_0 is used, then minimizing the MSE forces the array gain

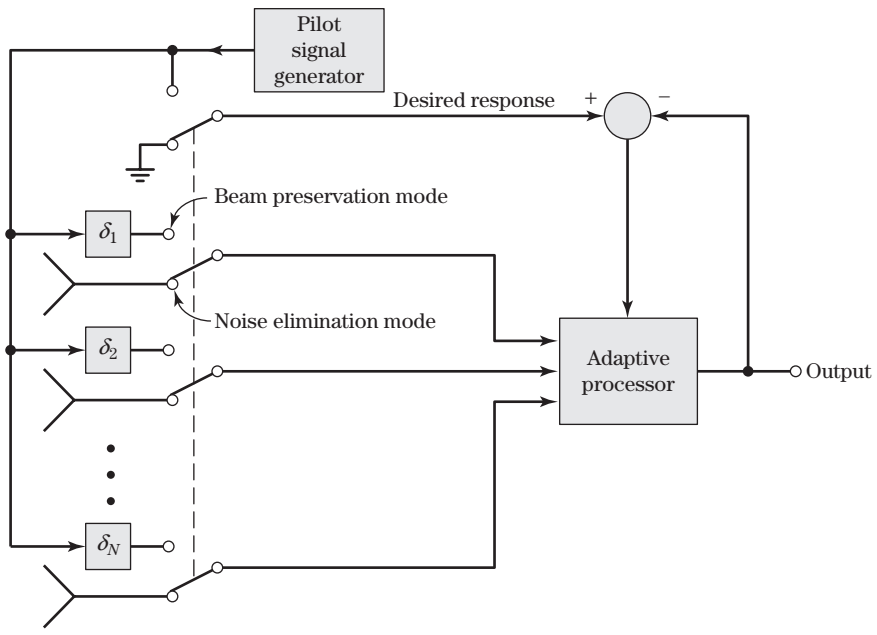


FIGURE 4-9 ■ Two-mode LMS adaptation for beam preservation and noise elimination.

in the desired look direction to have a specific amplitude and phase shift at that frequency. On the other hand, if the pilot signal is chosen to be the sum of several sinusoids having different frequencies, then the adaptation process forces the array gain and phase in the desired look direction to have specific values at each one of the pilot-signal frequencies. Finally, if several pilot signals corresponding to different look directions are added together, then the array gain is simultaneously constrained at the various frequencies and angles corresponding to the different pilot signals selected. In summary, the two-mode adaptation process minimizes the total power of all signals received that are uncorrelated with the pilot signals while constraining the gain and phase of the array beam to values corresponding to the frequencies and angles dictated by the pilot-signal components.

Figure 4-10 illustrates a practical one-mode method for simultaneously eliminating all noises uncorrelated with the pilot signal and forming a desired array beam. The circuitry of Figure 4-10 circumvents the difficulty of being unable to receive the actual signal, while the processor is connected to the pilot-signal generator by introducing an auxiliary adaptive processor. For the auxiliary adaptive processor, the desired response is the pilot signal, and both the pilot signal and the actual received signals enter the processor. A second processor performs no adaptation (its weights are slaved to the weights of the adaptive processor) and generates the actual array output signal. The slaved processor inputs do not contain the pilot signal and can therefore receive the transmitted signal at all times.

In the one-mode adaptation method, the pilot signal is on continuously so the adaptive processor that minimizes the MSE forces the adaptive processor output to closely reproduce the pilot signal while rejecting all signals uncorrelated with the pilot signal.

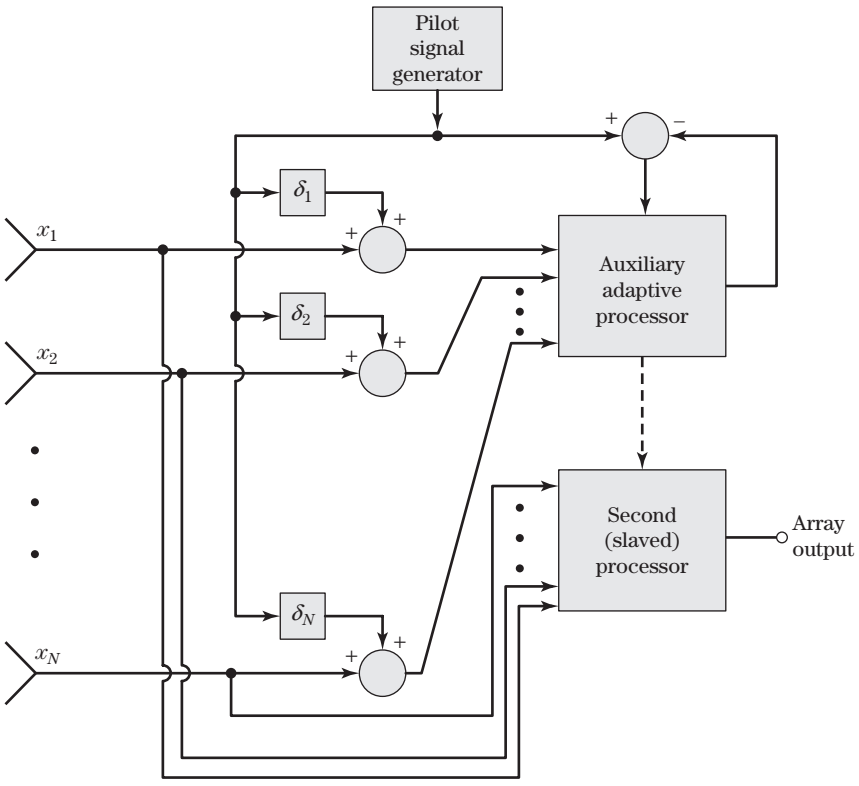


FIGURE 4-10 ■ One-mode LMS adaptation for simultaneous beam preservation and noise elimination.

The adaptive processor therefore preserves the desired array directivity in the look direction (over the pilot-signal passband) while placing nulls in the directions of noise sources (over the noise frequency bands).

4.3 | THE HOWELLS–APPLEBAUM ADAPTIVE PROCESSOR

The key capability of adaptive nulling was developed for an intermediate frequency (IF) radar sidelobe canceller as represented by the patent of Howells [16]. An analysis of this approach by Applebaum [17] established the control-law theory governing the operation of an adaptive control loop for each array element. The Applebaum algorithm maximizes a generalized signal-to-noise ratio (SNR) with the assumptions that the desired signal is absent most of the time (as in a pulsed radar or sonar system) and the direction of arrival of the desired signal is known. Because the Howells–Applebaum processor is practical to implement, it has been applied extensively to the problem of clutter and interference rejection in radar systems [18–21]. Unless otherwise noted, the analysis in this section follows the treatment of this subject given by Gabriel [22].

A six-element linear array with Howells–Applebaum loops at each element is shown in Figure 4-11. This figure shows the close relationship between an LMS loop and a

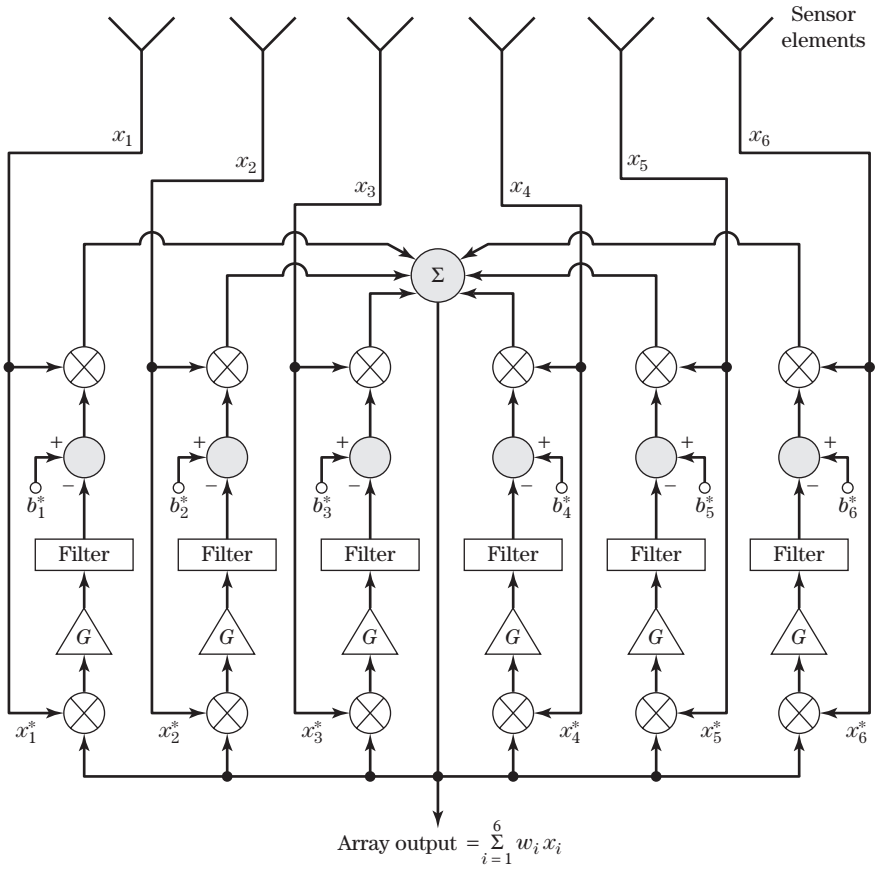


FIGURE 4-11 ■ Six-element linear adaptive array having six Howells–Applebaum control loops with beam steering signals. From Gabriel, *Proc. IEEE*, February 1976.

Howells–Applebaum loop. The beam steering vector, $\mathbf{b}^*$, in the Howells–Applebaum loop plays the same role as the cross-correlation vector, $\mathbf{r}_{xd}$, in the LMS loop. Define an element signal vector $\mathbf{x}$ in which the kth component, x_k, consists of the quiescent receiver channel noise voltage n_k and a summation of voltage terms associated with I external, narrowband interference sources:

$$\mathbf{x}^T = [x_1, x_2, \ldots, x_N] \tag{4.84}$$

where

$$x_k = n_k + \sum_{i=1}^{I} J_i e^{j\psi_i(2k-N-1)/2} \tag{4.85}$$

and

$$\psi_i = \left(\frac{2\pi d}{\lambda}\right) \sin\theta_i \tag{4.86}$$

The interference sources are assumed to be statistically independent where J_i is the element channel voltage amplitude associated with the ith source at θ_i.

Beam steering signals steer the receive beam to θ_s. For quiescent conditions (when only receiver noise is present), the adaptive weights settle to steady-state values denoted by the quiescent weight vector $\mathbf{w}_q$, where

$$\mathbf{w}_q^T = [w_{q_1}, w_{q_2}, \ldots, w_{q_N}] \tag{4.87}$$

and

$$w_{q_k} = a_k e^{-j\psi_0(2k-N-1)/2} \tag{4.88}$$

$$\psi_0 = \left(\frac{\pi d}{\lambda}\right) \sin\phi_0 \tag{4.89}$$

where a_k are the element amplitude weights. The resulting quiescent array factor is expressed as

$$AF_q(\theta) = (\mathbf{s}^T \mathbf{w}_q) = \sum_{k=1}^{N} a_k \exp[j(\psi - \psi_s)(2k - N - 1)/2] \tag{4.90}$$

where

$$\mathbf{s}^T = [s_1, s_2, \ldots, s_N] = \text{ element signals} \tag{4.91}$$

$$s_k = \exp[j\psi(2k - N - 1)/2] \tag{4.92}$$

$$\psi = \frac{2\pi d}{\lambda} \sin\theta \tag{4.93}$$

The components of the input beam steering vector $\mathbf{b}^*$

$$\mathbf{b}^{*T} = \left[b_1^*, b_2^*, \ldots, b_N^*\right] \tag{4.94}$$

are directly related to the components of $\mathbf{w}_q$ by the relation

$$b_k^* = c_k w_{q_k} \tag{4.95}$$

where the constants c_k are evaluated in the section immediately following.

4.3.1 Adaptive Weight Equations

The weight w_k associated with the kth sensor element is given by

$$w_k = b_k^* - z_k \tag{4.96}$$

where z_k = output voltage from the kth integrating RC filter. Each correlation mixer voltage is given by the product of the signal v_k^* with the summed array output

$$v_k = k^2 \left(x_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.97}$$

where

$$\tau_0 \frac{dz_k}{dt} + z_k = \gamma \left(x_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.98}$$

$$\gamma = k^2 G \tag{4.99}$$

The constant γ represents a conversion-factor gain constant that is assumed to be the same for all the loops. It is convenient to use (4.96) to convert from z_k to w_k, so that (4.98) now becomes

$$\tau_0 \frac{dw_k}{dt} + w_k = b_k^* - \gamma \left[x_k^* \sum_{i=1}^{N} w_i x_i \right] \tag{4.100}$$

Using matrix notation, we may write the complete set of N differential equations corresponding to (4.100) as

$$\tau_0 \frac{d\mathbf{w}}{dt} + \mathbf{w} = \mathbf{b}^* - \gamma \left[\mathbf{x}^* \mathbf{w}^T \mathbf{x} \right] \tag{4.101}$$

Since $(\mathbf{w}^T \mathbf{x}) = (\mathbf{x}^T \mathbf{w}) = \sum_{i=1}^{N} w_i x_i$, the bracketed term in (4.101) can be rewritten as

$$[\mathbf{x}^* \mathbf{w}^T \mathbf{x}] = [\mathbf{x}^* \mathbf{x}^T] \mathbf{w} \tag{4.102}$$

The expected (averaged) value of $\mathbf{x}^* \mathbf{x}^T$ yields the input signal correlation matrix

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^* \mathbf{x}^T\} \tag{4.103}$$

The averaged values of the correlation components forming the elements of $\mathbf{R}_{xx}$ are given by

$$\overline{x_k^* x_l} = \begin{cases} \displaystyle\sum_{i=1}^{I} |\overline{J}_i|^2 \exp[j\psi_i (l-k)] & l \neq k \quad (4.104) \\ |\overline{x}_k|^2 = |\overline{n}_k|^2 + \displaystyle\sum_{i=1}^{I} |\overline{J}_i|^2 & l = k \quad (4.105) \end{cases}$$

Since the correlation matrix in the absence of the desired signal is the sum of the quiescent receiver noise matrix $\mathbf{R}_{nn_q}$ and the individual interference source matrixes $\mathbf{R}_{nn_i}$, it follows that

$$\mathbf{R}_{nn} = \mathbf{R}_{nn_q} + \sum_{i=1}^{I} \mathbf{R}_{nn_i} \tag{4.106}$$

where $\mathbf{R}_{nn_q}$ can be expressed as

$$\mathbf{R}_{nn_q} = \begin{bmatrix} |\overline{n}_1|^2 & 0 & 0 & \cdots \\ 0 & |\overline{n}_2|^2 & 0 & \cdots \\ & & \ddots & \\ 0 & \cdots & \cdots & |\overline{n}_N|^2 \end{bmatrix} \tag{4.107}$$

and

$$\mathbf{R}_{nn_i} = |\overline{J}_i|^2 \begin{bmatrix} 1 & e^{j\psi_i} & e^{j2\psi_i} & \cdots \\ e^{-j\psi_i} & 1 & e^{j\psi_i} & \cdots \\ e^{-j2\psi_i} & e^{-j\psi_i} & 1 & \cdots \\ & & & \ddots \\ & & & 1 \end{bmatrix} \tag{4.108}$$

Substituting $\mathbf{R}_{nn}$ of (4.106) into (4.101) and rearranging terms, the final expression for the adaptive weight matrix differential equation becomes

$$\tau_0 \frac{d\mathbf{w}}{dt} + [\mathbf{I} + \gamma \mathbf{R}_{nn}]\mathbf{w} = \mathbf{b}^* \tag{4.109}$$

where $\mathbf{I}$ is the identity matrix.

In general, $\mathbf{R}_{nn}$ is not diagonal, so multiplying $\mathbf{R}_{nn}$ by a nonsingular orthonormal model matrix, Q, results in a simple transformation of coordinates that diagonalizes $\mathbf{R}_{nn}$. The resulting diagonalized matrix has diagonal elements that are the eigenvalues of the matrix $\mathbf{R}_{nn}$. The eigenvalues of $\mathbf{R}_{nn}$ are given by the solutions of the equation

$$|\mathbf{R}_{nn} - \lambda_i \mathbf{I}| = 0, \quad i = 1, 2, \ldots, N \tag{4.110}$$

Corresponding to each eigenvalue there is an associated eigenvector $\mathbf{e}_i$ that satisfies

$$\mathbf{R}_{nn}\mathbf{e}_i = \lambda_i \mathbf{e}_i \tag{4.111}$$

These eigenvectors (which are normalized to unit length and are orthogonal to one another) make up the rows of the transformation matrix $\mathbf{Q}$, that is,

$$\mathbf{Q} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \cdots \\ e_{21} & e_{22} & e_{23} & \cdots \\ e_{31} & e_{32} & e_{33} & \cdots \\ \vdots & & & \\ e_{N1} & e_{N2} & e_{N3} & \cdots \end{bmatrix}, \quad \text{where } \mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{iN} \end{bmatrix} \tag{4.112}$$

Once $\mathbf{R}_{nn}$ is diagonalized by the $\mathbf{Q}$-matrix transformation, there results

$$[\mathbf{Q}^*\mathbf{R}_{nn}\mathbf{Q}^T] = \begin{bmatrix} \lambda_1 & 0 & & 0 \\ 0 & \lambda_2 & & 0 \\ 0 & 0 & \lambda_3 & 0 \\ \vdots & & \ddots & \\ \cdots & \cdots & \cdot & \lambda_N \end{bmatrix} \tag{4.113}$$

Now since $\mathbf{R}_{nn} = E\{\mathbf{x}^*\mathbf{x}^T\}$, it follows that (4.113) may be written as

$$[\mathbf{Q}^*\mathbf{R}_{nn}\mathbf{Q}^T] = [\overline{\mathbf{Q}^*\mathbf{x}^*\mathbf{x}^T\mathbf{Q}^T}] = \overline{[\mathbf{x}'^*\mathbf{x}'^T]} = \mathbf{\Lambda} \tag{4.114}$$

where

$$\mathbf{x}' = \mathbf{Q}\mathbf{x} \tag{4.115}$$

The $\mathbf{Q}$ matrix transforms the real signal vector $\mathbf{x}$ into the orthonormal signal vector $\mathbf{x}'$. Furthermore, the components of $\mathbf{x}'$ are determined by the eigenvectors of $\mathbf{R}_{nn}$, that is,

$$x_k' = (\mathbf{e}_k^T\mathbf{x}) \tag{4.116}$$

Note that the orthonormal signal vector components x_k' have two special characteristics:

1. They are uncorrelated so

$$E\{x_k'^* x_l'\} = 0 \quad \text{for } l \neq k \tag{4.117}$$

2. Their amplitudes are given by the square root of the corresponding eigenvalue so that

$$E\{x_k'^* x_k'\} = \lambda_k \tag{4.118}$$

The transformation matrix $\mathbf{Q}$ yields the same signal components as an appropriately selected orthogonal beamforming network.

Just as the signal vector $\mathbf{x}$ was transformed into $\mathbf{x}'$ by (4.115), the beam steering vector $\mathbf{b}^*$ may likewise be transformed to define a new beam steering vector $\mathbf{b}'^*$ as

$$\mathbf{b}' = \mathbf{Q}\mathbf{b} \tag{4.119}$$

where the kth component of $\mathbf{b}'$ is determined by the kth eigenvector appearing in $\mathbf{Q}$.

The Q-coordinate transformation operating on both $\mathbf{x}$ and $\mathbf{b}^*$ suggests an equivalent circuit representation for the system that is illustrated in Figure 4-12b, where an equivalent "orthonormal adaptive array" system is shown alongside a simplified representation of the real system in Figure 4-12a. There are a set of weights forming the weight vector $\mathbf{w}'$ in the orthonormal system, and the adaptive weight matrix equation for the equivalent system is

$$\tau_0 \frac{d\mathbf{w}'}{dt} + [\mathbf{I} + \gamma\mathbf{R}_{nn}']\mathbf{w}' = b'^* \tag{4.120}$$

where

$$\mathbf{R}_{nn}' = E\{\mathbf{x}'^*\mathbf{x}'^T\} = \mathbf{\Lambda} \tag{4.121}$$

This diagonalization results in an orthonormal system, a set of independent linear differential equations, each of which has a solution when the eigenvalues are known. Each of the orthonormal servo loops in the equivalent system responds independently of the other loops, because the x_k' input signals are orthogonalized and are therefore completely uncorrelated with one another. The weight equation for the kth orthonormal servo loop can therefore be written as

$$\tau_0 \frac{dw_k'}{dt} + (1 + \gamma\lambda_k)w_k' = b_k'^* \tag{4.122}$$

Note that the equivalent servo gain factor can be defined from (4.122) as

$$\mu_k = \gamma\lambda_k \tag{4.123}$$

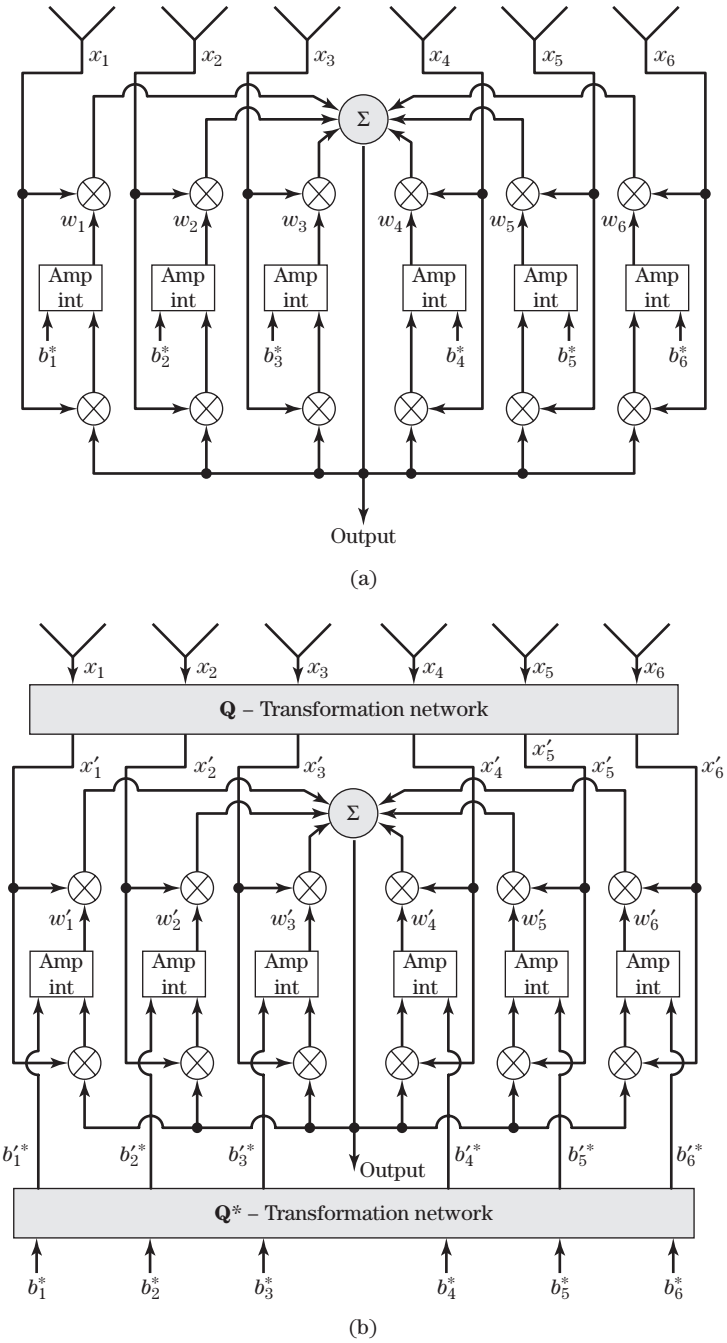


FIGURE 4-12 ■ Equivalent circuit representations for a six-element adaptive array system. a: Real adaptive array system. b: Equivalent orthonormal adaptive array system. From Gabriel, *Proc. IEEE*, February 1976.

so the equivalent servo gain factors for the various orthonormal loops are now determined by the eigenvalues of the input signal covariance matrix. The positive, real eigenvalues λ_k correspond to the square of a signal voltage amplitude, and any given eigenvalue is proportional to the power appearing at the orthonormal network output port.

For the input beam steering vector $\mathbf{b}^*$, the output desired signal power is given by

$$P_s = |\mathbf{w}^T \mathbf{b}|^2 \tag{4.124}$$

Likewise, the array output noise power can be written as

$$P_n = |\overline{\mathbf{w}^T\mathbf{x}}|^2 \tag{4.125}$$

where the signal vector $\mathbf{x}$ is assumed to be composed only of quiescent receiver channel noise plus the directional noise signal components due to external sources of interference. The signal-to-noise performance measure is therefore just a ratio of the aforementioned two quadratic forms

$$\left(\frac{s}{n}\right) = \frac{|\mathbf{w}^T\mathbf{b}|^2}{|\overline{\mathbf{w}^T\mathbf{x}}|^2} = \frac{\mathbf{w}^\dagger[\mathbf{b}^*\mathbf{b}^T]\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}} \tag{4.126}$$

The optimum weight vector (see Chapter 3) that yields the maximum SNR for (4.126) is

$$\mathbf{w}_{\text{opt}} = \frac{1}{(\text{constant})}\mathbf{R}_{nn}^{-1}\mathbf{b}^* \tag{4.127}$$

On comparing (4.127) with (4.45), it is seen that both the LMS and maximum SNR algorithms yield precisely the same weight vector solution (to within a multiplicative constant when the desired signal is absent) provided that $\mathbf{r}_{xd} = \mathbf{b}^*$, since these two vectors play exactly the same role in determining the optimum weight vector solution. Consequently, adopting a specific vector $\mathbf{r}_{xd}$ for the LMS algorithm is equivalent to selecting $\mathbf{b}^*$ for the maximum SNR algorithm, which represents direction of arrival information—this provides the relation between a reference signal and a beam steering signal for the LMS and maximum SNR algorithms to yield equivalent solutions.

From the foregoing discussion, it follows that the optimum orthonormal weight is

$$w'_{k_{\text{opt}}} = \left(\frac{1}{\mu_k}\right) b'^*_k \tag{4.128}$$

Substitute (4.123) and (4.128) into (4.122) results in

$$\tau_0\frac{dw'_k}{dt} + (1+\mu_k)w'_k = \mu_k w'_{k_{\text{opt}}} \tag{4.129}$$

For a step-function change in the input signal the solution may be written as follows:

$$w'_k(t) = \left[w'_k(0) - w'_k(\infty)\right]e^{-\alpha_k t} + w'_k(\infty) \tag{4.130}$$

where

$$w'_k(\infty) = \left(\frac{\mu_k}{1+\mu_k}\right) w'_{k_{\text{opt}}} \tag{4.131}$$

$$\alpha_k = \left(\frac{1+\mu_k}{\tau_0}\right) \tag{4.132}$$

In the foregoing equations $w'_k(\infty)$ represents the steady-state weight, $w'_k(0)$ is the initial weight value, and α_k is the transient decay factor. The adaptive weight transient responses can now be determined by the eigenvalues. The kth orthonormal servo loop may be represented by the simple type-0 position servo illustrated in Figure 4-13.

To relate the orthonormal system weights w'_k to the actual weights w_k note that the two systems shown in Figure 4-12 must be exactly equivalent so that

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}'^T\mathbf{x}' = \mathbf{w}'^T\mathbf{Q}\mathbf{x} \tag{4.133}$$

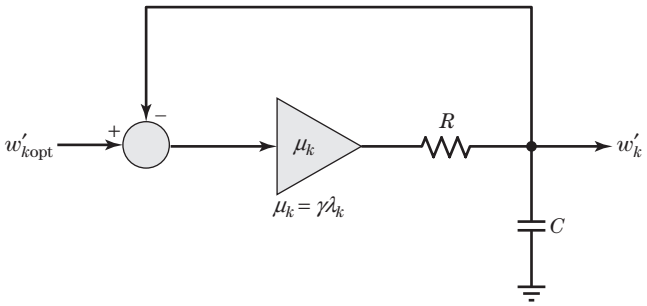


FIGURE 4-13 ■ Type-O servo model for *k*th orthonormal adaptive control loop. From Gabriel, *Proc. IEEE*, February 1976.

Consequently

$$\mathbf{w} = \mathbf{Q}^T \mathbf{w}' \tag{4.134}$$

From (4.134) it follows that the solution for the *k*th actual weight can be written as

$$w_k = \left(e_{1_k} w'_1 + e_{2_k} w'_2 + \cdots + e_{N_k} w'_N\right) \tag{4.135}$$

where e_{n_k} is the *k*th element of the *n*th eigenvector.

In the quiescent state, only receiver noise is present in any channel so that various channel signals are uncorrelated, and $\mathbf{Q}$ is an identity matrix, and there is no difference between $\mathbf{w}'$ and $\mathbf{w}$. With $\mathbf{Q}_q = \mathbf{I}$, the quiescent noise covariance matrix $\mathbf{R}_{nn_q}$ is already diagonalized, and if it is further assumed that the receiver noise power in all channels is equal (and denoted by $|\overline{n}_0|^2$), then from (4.113) it follows that

$$\mathbf{Q}_q^* \mathbf{R}_{nn_q} \mathbf{Q}_q^T = [\lambda_0 \delta_{ij}] \tag{4.136}$$

where

$$\lambda_0 = |\overline{n}_0|^2 \tag{4.137}$$

so the smallest eigenvalue is simply equal to the receiver channel noise power. This smallest eigenvalue then defines the minimum servo gain factor $\mu_{\min}$ as

$$\mu_{\min} = \gamma \lambda_0 \tag{4.138}$$

Since the quiescent steady-state weight $\mathbf{w}(\infty)$ must by definition be equal to $\mathbf{w}_q$, (4.131), (4.132), and (4.95) can be applied to yield

$$w_{q_k} = \frac{1}{1 + \mu_{\min}} b_k^* = \left(\frac{c_k}{1 + \mu_{\min}}\right) w_{q_k}$$

or

$$c_k = (1 + \mu_{\min}) \tag{4.139}$$

From (4.130)–(4.131) and (4.123), it follows that the effective time constant with which the *k*th component of $\mathbf{w}'$ converges to its optimum value is $\tau_0/(1 + \gamma\lambda_k)$. In effect, $\lambda_{\min}$ determines how rapidly the adaptive array follows changes in the noise environment. Equation (4.135) shows that each actual weight can be expressed as a weighted sum of exponentials, and the component that converges most slowly is the $\lambda_{\min}$ component.

When the adaptive array in Figure 4-11 operates with a distributed external noise field, the loop convergence is very slow for some angular noise distributions [23]. Furthermore, if γ is increased or τ_0 is decreased to speed the weight convergence, the loop becomes "noisy." Slow weight convergence occurs whenever trace($\mathbf{R}_{nn}$)/$\lambda_{\min}$ is large, and in these cases there is no choice of γ and τ_0 that yields rapid convergence without excessive loop noise. These facts suggest that the effects of noise on the solutions represented by (4.128)–(4.132) are important.

Griffiths provides a discrete form of the Howells–Applebaum weight update formula given by [24]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \gamma\left[\mu\mathbf{b}^* - \mathbf{x}^*(k)\mathbf{x}^\dagger(k)\mathbf{w}(k)\right] \tag{4.140}$$

where γ and μ are constants. The weights converge if γ is less than one over the largest eigenvalue. Compton shows that this is equivalent to [25]

$$0 < \gamma < \frac{1}{P_{IN}} \tag{4.141}$$

where P_{IN} is the total received power in (4.48). If γ is close to $1/P_{\text{IN}}$ then convergence is fast, but weight jitter is large. The weight jitter causes SNR fluctuations of several dB at steady state. If γ is small, then weight jitter is small, but the convergence is slow. A gain constant of [26]

$$\gamma = \frac{1}{2.5P_{IN}} \tag{4.142}$$

was found to provide a reasonably stable steady-state weights and rapid conversion.

4.3.2 Loop Noise Considerations

The random variations in the adaptive element weights of a Howells–Applebaum control loop result in an additional noise component in the array output signal. In this section expressions are given for the variance of the element weights and for the resulting additional noise in the array output [23].

Let $\boldsymbol{\xi}$ denote the noise component of the adaptive weight vector $\mathbf{w}$, and let $\mathfrak{N}$ denote the random component of $\mathbf{R}_{nn}$, so that

$$\mathbf{w} = \overline{\mathbf{w}} + \boldsymbol{\xi} \tag{4.143}$$

$$\mathbf{R}_{nn} = \overline{\mathbf{R}}_{nn} + \mathfrak{N} \tag{4.144}$$

where now $\overline{\mathbf{w}}$ and $\overline{\mathbf{R}}_{nn}$ denote average values. The adaptive weights must satisfy

$$\tau_0\frac{d\mathbf{w}}{dt} + (\mathbf{I} + \gamma\mathbf{R}_{nn})\mathbf{w} = \mathbf{b}^* \tag{4.145}$$

Substitute the values $\overline{\mathbf{w}}$ and $\overline{\mathbf{R}}_{nn}$ into (4.145) and subtract the result from the equation resulting with (4.143) and (4.144) substituted into (4.145) to give

$$\tau_0\frac{d\boldsymbol{\xi}}{dt} + \left(\mathbf{I} + \gamma\overline{\mathbf{R}}_{nn}\right)\boldsymbol{\xi} = -\gamma\mathfrak{N}\mathbf{w} \tag{4.146}$$

Premultiplying (4.146) by the transformation matrix $\mathbf{Q}^*$ and using the fact that $\mathbf{Q}^*\mathbf{Q}^T = \mathbf{I}$, then

$$\frac{d\boldsymbol{\zeta}}{dt} + \frac{1}{\tau_0}(\mathbf{I} + \gamma\boldsymbol{\Lambda})\boldsymbol{\zeta} = -\beta\mathbf{Q}^*\Re\mathbf{w} = \mathbf{u} \tag{4.147}$$

where

$$\boldsymbol{\zeta} = \mathbf{Q}^*\boldsymbol{\xi} \tag{4.148}$$

$$\beta = \frac{\gamma}{\tau_0} \tag{4.149}$$

Equation (4.147) represents a system of N independent linear differential equations of which the nth component can be written as

$$d\zeta_n + \sigma_n\zeta_n dt = u_n dt \tag{4.150}$$

where

$$\sigma_n = \frac{1 + \gamma\lambda_n}{\tau_0} \tag{4.151}$$

$$u_n = u_n(\tau, \overline{\mathbf{w}}, \boldsymbol{\zeta}) = (-\beta\mathbf{Q}^*\Re\mathbf{w})_n \tag{4.152}$$

Multiplying (4.150) by the factor $e^{\sigma_n t}$ and integrating each term from t_0 to t then yields

$$\begin{aligned}\zeta_n(t) = \zeta_n(t_0)\exp[-\sigma_n(t - t_0)] + \textstyle\int_{t_0}^{t} e^{-\sigma_n(r-\tau)} \\ \cdot u_n(\tau, \overline{\mathbf{w}}, \boldsymbol{\zeta})d\tau\end{aligned} \tag{4.153}$$

If only the steady-state case is considered, then the weights are near their mean steady-state values. The steady-state solution for variations in the element weights can be obtained from (4.153) by setting $t_0 = -\infty$ and ignoring any effect of the initial value $\zeta_n(t_0)$ to give

$$\zeta_n(t) = \int_0^\infty e^{-\sigma_n\tau}u_n(t - \tau)d\tau \tag{4.154}$$

One important measure of the noise present in the adaptive loops is the variance of the weight vector denoted by var($\mathbf{w}$):

$$\text{var}(\mathbf{w}) = E\left\{\sum_{n=1}^{N}|\mathbf{w}_n - \overline{\mathbf{w}}_n|^2\right\} = E\{\boldsymbol{\xi}^\dagger\boldsymbol{\xi}\} \tag{4.155}$$

where N is the dimension of the weight vector (or the number of degrees of freedom in the adaptive array system). Now since $\boldsymbol{\zeta} = \mathbf{Q}^*\boldsymbol{\xi}$, (4.155) becomes

$$\text{var}(\mathbf{w}) = E\{\boldsymbol{\zeta}^\dagger\boldsymbol{\zeta}\} \tag{4.156}$$

The elements of the covariance matrix of $\boldsymbol{\zeta}(t)$ in (4.156) are obtained from (4.154) and the definition of u_n

$$\begin{aligned}E\{\zeta_j^*\zeta_k\} = \beta^2\int_0^\infty d\tau_1\int_0^\infty E\{[\mathbf{Q}^*\Re(t - \tau_1)\overline{\mathbf{w}}]_j^* \\ \cdot \exp(-\sigma_j\tau_1 - \sigma_k\tau_2)[Q^*\Re(t - \tau_2)\overline{w}]_k\}\,d\tau_2 \\ + \beta^2\int_0^\infty d\tau_1\int_0^\infty E\{[\mathbf{Q}^*\Re(t - \tau_1)\boldsymbol{\xi}(t - \tau_1)]_j^* \\ \cdot \exp(-\sigma_j\tau_1 - \sigma_k\tau_2)[\mathbf{Q}^*\Re(t - \tau_2)\boldsymbol{\xi}(t - \tau_2)]_k\}\,d\tau_2\end{aligned} \tag{4.157}$$

where the cross-product terms do not appear since $E\{\boldsymbol{\xi}(t)\} = 0$ and $\Re(t)$ and $\boldsymbol{\xi}(t)$ are independent noise processes. A useful lower bound for (5.157) is given by [23].

$$\text{var}(\mathbf{w}) \geq \frac{\beta^2 \Delta}{2} \sum_{n=1}^{N} \frac{1}{\sigma_n} E\left\{ |(\mathbf{Q}^* \Re \overline{\mathbf{w}})_n|^2 \right\} \tag{4.158}$$

where Δ represents the time interval between successive independent samples of the input signal vector. For a pulse radar, Δ is approximately the same as the pulse width. For a communications system, Δ is approximately $1/B$, where B is the signal bandwidth.

The bound in (4.158) is useful in selecting parameter values for the Howells–Applebaum servo loops. If this bound is not small, then the noise fluctuations at the output of the adaptive loops are correspondingly large. For cases of practical interest [when var($\mathbf{w}$) is small compared with $\overline{\mathbf{w}}^\dagger \overline{\mathbf{w}}$], the right-hand side of (4.158) is an accurate estimate of var($\mathbf{w}$). Equation (4.158) simplifies (after considerable effort) to yield the expression

$$\text{var}(\mathbf{w}) \geq \left[\frac{\Re \beta \Delta}{2} - \frac{\beta \Delta}{2\gamma} \sum_{n=1}^{N} \frac{1}{\lambda_n + 1/\gamma} \right] \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \overline{\mathbf{w}} \tag{4.159}$$

where λ_n represents the nth eigenvalue of $\overline{\mathbf{R}}_{nn}$

$$\Re = \frac{N}{\beta} - \frac{1}{\gamma} \text{trace}(\boldsymbol{\mu}) \tag{4.160}$$

$$\boldsymbol{\mu} = (\beta \mathbf{H})^{-1} \tag{4.161}$$

$$\mathbf{H} = \overline{\Re}_{nn} \frac{\mathbf{I}}{\gamma} \tag{4.162}$$

Since the total output noise power is the noise power without noisy weights $\overline{\mathbf{w}}^\dagger \mathbf{R}_{nn} \overline{\mathbf{w}}$ plus the additional noise due to the random weight components, it can be shown that the total output noise power is given by

$$\begin{aligned} E\left\{ |\mathbf{w}^T \mathbf{x}|^2 \right\} &= \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \mathbf{w} + E\left\{ \boldsymbol{\xi}^\dagger \overline{\mathbf{R}}_{nn} \boldsymbol{\xi} \right\} \\ &\cong \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \overline{\mathbf{w}} \left[1 + \frac{\beta \Delta}{2} \sum_{n=1}^{N} \lambda_n \right] \end{aligned} \tag{4.163}$$

when $\gamma \lambda_n \gg 1$ for $n = 1, 2, \ldots, N$. The quantity $\beta \Delta$ occurs both in (4.163) and in (4.159) is the ratio $\gamma \Delta / \tau_0$, which is the gain divided by the loop time constant where the time constant is measured in intervals of the independent-sample rate of the system.

When loop noise is present in the system, the total noise power output increases by the factor $(1 + K_n)$, where from (4.163)

$$K_n \geq \frac{\gamma \Delta}{2\tau_0} \sum_{n=1}^{N} \lambda_n = \frac{\gamma}{4B\tau_0} \text{trace}(\overline{\mathbf{R}}_{nn}) \tag{4.164}$$

where $\Delta = 1/2B$ (i.e., B is the bandwidth of the input signal process), so that K_n is a direct measure of algorithm misadjustment due to noise in the weight vector. Recalling the solution to (4.129), we see that the effective time constant of the normal weight component $w'_{k'}$ having the slowest convergence rate is

$$\tau_{\text{eff}} = \frac{\tau_0}{1 + \gamma \lambda_{\min}} \cong \frac{\tau_0}{\gamma \lambda_{\min}} \tag{4.165}$$

where $\gamma\lambda_{\min} \geq 1$ to avoid a steady-state bias error in the solution. On combining (4.164) and (4.165) there results

$$\frac{\tau_{\text{eff}}}{\Delta} \geq \frac{1}{2K_n\lambda_{\min}} \sum_{n=1}^{N} \lambda_n = \frac{\text{trace}(\overline{\mathbf{R}}_{nn})}{2K_n\lambda_{\min}} \tag{4.166}$$

Equation (4.166) shows that, when the smallest eigenvalue $\lambda_{\min}$ is small compared with trace($\overline{\mathbf{R}}_{nn}$), many independent samples of the input signal are required before the adaptive array settles to a near-optimum set of weights without excessive loop noise; no set of loop parameters yields both low loop noise and rapid convergence in this case. Berni [27] gives an analysis of steady-state weight jitter in Howells–Applebaum control loops when there is no statistical independence between the input signal and weight processes. Steady-state weight jitter is closely related to the statistical dependence between the weight and signal processes.

4.3.3 Adaptive Array Behavior in Terms of Eigenvector Beams

The **Q**-matrix transformation defined by (4.112) is composed of normalized and mutually orthogonal eigenvectors. The components of these eigenvectors are interpreted as array element weights, associated with normalized orthogonal eigenvector beams. The kth eigenvector beam is expressed as

$$g_k(\theta) = (\mathbf{s}^T\mathbf{e}_k) = \sum_{i=1}^{N} e_{ki}s_i \tag{4.167}$$

where **s** and its components s_i for a linear N-element array are defined by

$$\mathbf{s}^T = [s_2, s_2, \ldots, s_N] \tag{4.168}$$

$$s_i = e^{j\psi(2i-N-1)/2} \tag{4.169}$$

By defining the variable z related to the spatial angle θ as

$$z \triangleq e^{j\psi} \tag{4.170}$$

then the eigenvector beam may be conveniently rewritten as

$$g_k(\theta) = \left(\frac{1}{\sqrt{z}}\right)^{N-1} \left[e_{k1} + e_{k2}z + e_{k3}z^2 + \cdots + e_{kN}z^{N-1}\right] \tag{4.171}$$

As shown in Chapter 2, the array polynomial can also be expressed in the factored form

$$g_k(\theta) = \left(\frac{1}{\sqrt{z}}\right)^{N-1} [a_{N-1}(z - z_1)(z - z_2)\cdots(z - z_{N-1})] \tag{4.172}$$

where the roots $z_1, z_2, \ldots, z_{N-1}$ are nulls in the eigenvector beam pattern.

If one narrowband interference source is at θ_1, then $\mathbf{R}_{nn}$ contains one unique eigenvalue and the corresponding unique eigenvector that produces a retrodirective eigenvector beam centered on the source at θ_1 as illustrated in Figure 4-14. The $\mathbf{R}_{nn}$ matrix in this case also contains nonunique eigenvalues having arbitrary nonunique eigenvectors; these arbitrary nonunique eigenvector beams are not essential to array operation, and array pattern performance is characterized solely in terms of the unique retrodirective eigenvector beams.

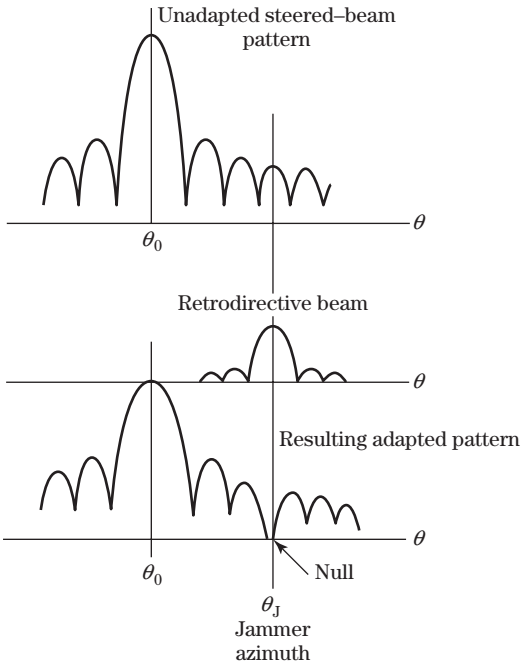


FIGURE 4-14 ■ Retrodirective beam principle illustrating subtraction of retrodirective beam from unadapted pattern to obtain adapted pattern with one interference source. From Gabriel, *Proc. IEEE,* February 1976.

The overall array beam pattern is most easily derived by considering the output of the orthonormal system represented in Figure 4-12b for the input signal vector $\mathbf{s}$, defined in (4.169). Since the output for the real orthonormal systems are identical, it follows that

$$AF(\theta, t) = \sum_{i=1}^{N} w_i s_i = \sum_{i=1}^{N} w_i' s_i' = w'^T \mathbf{s}' \tag{4.173}$$

where

$$\mathbf{s}' = \mathbf{Q}\mathbf{s} \tag{4.174}$$

Now the ith component of $\mathbf{s}'$ is given by

$$s_i' = \left(\mathbf{e}_i^T \mathbf{s}\right) = \sum_{k=1}^{N} e_{ik} s_k \tag{4.175}$$

but this summation defines the ith eigenvector beam [as can be seen from (4.167)], so that

$$s_i' = \left(\mathbf{e}_i^T \mathbf{s}\right) = g_i(\theta) \tag{4.176}$$

Consequently, the overall array factor can be expressed as

$$AF(\theta, t) = \sum_{i=1}^{N} w_i' g_i(\theta) \tag{4.177}$$

which shows that the output array factor is the summation of the N eigenvector beams weighted by the orthonormal system adaptive weights.

Since the kth component of the quiescent orthonormal weight vector is given by

$$w_{q_k}' = \left(\mathbf{e}_k^{\dagger} \mathbf{w}_q\right) \tag{4.178}$$

the steady-state solution for the kth component of the orthonormal weight vector given by (4.131) can be rewritten using (4.132), (4.95), and (4.139) to yield

$$w'_k(\infty) = \left(\frac{1+\mu_{\min}}{1+\mu_k}\right) w'_{q_k} \tag{4.179}$$

Assume as before that quiescent signal conditions up to time $t = 0$ consist only of receiver noise and that the external interference sources are switched on at $t = 0$; then

$$w'_k(0) = w'_{q_k} \tag{4.180}$$

and the solution for w_k expressed by (4.131) is rewritten in the more convenient form

$$w'_k = w'_{q_k} - (1 - e^{-\alpha_k t}) \left[\frac{\mu_k - \mu_{\min}}{1+\mu_k}\right] w'_{q_k} \tag{4.181}$$

It is immediately apparent that at time $t = 0$ (4.177) results in

$$AF(\theta, 0) = \sum_{i=1}^{N} w'_{q_i} g_i(\theta) = \left(\mathbf{w}_q'^T \mathbf{s}'\right) = \left(\mathbf{w}_q'^T \mathbf{Q}\mathbf{s}\right) \tag{4.182}$$

From (4.134) it is seen that $\mathbf{w}_q^T = \mathbf{w}_q'^T \mathbf{Q}$ so that

$$AF(\theta, 0) = \left(\mathbf{w}_q^T \mathbf{s}\right) = AF_q(\theta) \tag{4.183}$$

where the quiescent pattern $AF_q(\theta)$ was previously defined by (4.90).

Finally, by substituting (4.183) and (4.181) into (4.177), there results

$$AF(\theta, t) = AF_q(\theta) - \sum_{i=1}^{N} (1 - e^{-\alpha_i t}) \left[\frac{\mu_i - \mu_{\min}}{1+\mu_i}\right] w'_{q_i} g_i(\theta) \tag{4.184}$$

where it will be recalled that

$$\alpha_i = \frac{1+\mu_i}{\tau_0}$$
$$\mu_i = \gamma \lambda_i$$
$$\mu_{\min} = \gamma \lambda_{\min}$$

The foregoing result emphasizes that the adaptive array factor consists of two parts:

1. The quiescent beam pattern $AF_q(\theta)$
2. The summation of weighted orthogonal eigenvector beams that is subtracted from $AF_q(\theta)$

Note also from (4.184) that the weighting associated with any eigenvector beams corresponding to eigenvalues equal to λ_0 (the quiescent eigenvalue) is zero since the numerator $(\mu_i - \mu_{\min})$ is zero for such eigenvalues. Consequently, any eigenvector beams associated with λ_0 is disregarded, leaving only unique eigenvector beams to influence the resulting pattern. The transient response time of (4.184) is determined by the value of α_i, which in turn is proportional to the eigenvalue. Therefore, a large eigenvalue yields a fast transient response for its associated eigenvector beam, whereas a small eigenvalue results in a slow transient response.

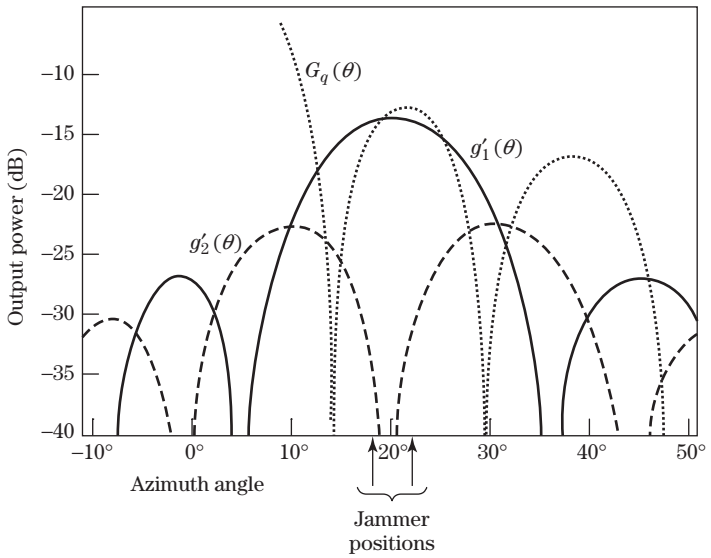


FIGURE 4-15 ■ Two eigenvector beam patterns $g_1'(\theta)$ and $g_2'(\theta)$ for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

The foregoing eigenvector beam interpretation of adaptive array behavior is illustrated by considering an eight-element linear array having $\lambda/4$ element spacing and two narrowband interference sources having nearly equal power ratios of $PR_1 = 1{,}250$ and $PR_2 = 1{,}200$ located at $\theta_1 = 18^\circ$ and $\theta_2 = 22^\circ$, respectively. Forming the covariance matrix using (4.106) for this case and solving for the eigenvalues yields two unique solutions: $\lambda_1 = 18{,}544.4$ and $\lambda_2 = 1{,}057.58$. These widely different eigenvalues result despite the nearly equal jammer powers, because the interference sources are close together compared with the array quiescent beamwidth.

Solving for the two (normalized) eigenvectors associated with the unique eigenvalues then permits the two eigenvector beam patterns to be found $g_1'(\theta)$ and $g_2'(\theta)$, which are both illustrated in Figure 4-15. Beam $g_1'(\theta)$ covers both interference sources in the same manner as a centered beam pattern, and its total output power is equal to the first eigenvalue

$$\left(\frac{\lambda_1}{\lambda_0}\right) = 1 + P_1 g_1'^2(\theta_1) + P_2 g_2'^2(\theta_2) = 18{,}544 \tag{4.185}$$

The second eigenvector beam $g_2'(\theta)$ splits the interference sources in the manner of a difference beam, and its total output power is equal to the second eigenvalue

$$\left(\frac{\lambda_2}{\lambda_0}\right) = 1 + P_1 g_2'^2(\theta_1) + P_2 g_2'^2(\theta_2) = 1057 \tag{4.186}$$

Although both eigenvector beams contain power from both sources, their respective output signals are decorrelated. The cross-correlation product of the two eigenvector beam outputs is

$$E\{x_1'^* x_2'\} = E\{|J_1|^2\} g_1'(\theta_1) g_2'(\theta_1) + E\{|J_2|^2\} g_1'(\theta_2) g_2'(\theta_2) \tag{4.187}$$

This cross-correlation product can be zero if the product $[g_1'(\theta) g_2'(\theta)]$ is positive when $\theta = \theta_1$ and negative when $\theta = \theta_2$, thereby resulting in decorrelation between the two eigenvector beam signals. Figure 4-16 shows the overall quiescent beam pattern and the resulting steady-state adapted pattern for this two-source example. Figure 4-17 illustrates the transient response (in terms of increase in output noise power) of the adaptive array for

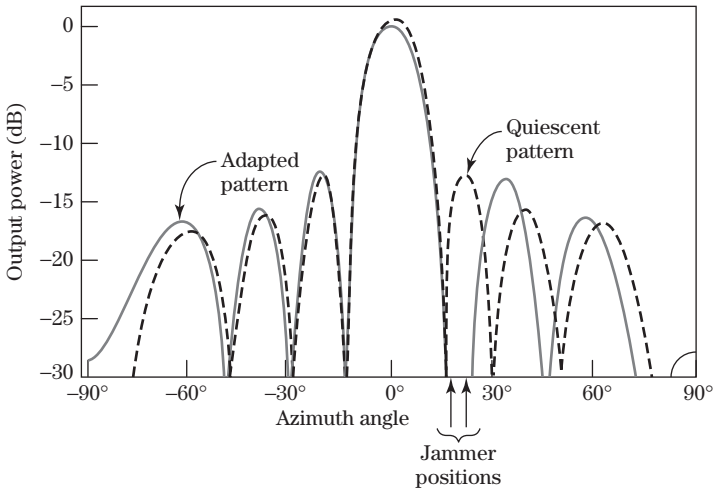


FIGURE 4-16 ■ Steady-state adapted array pattern and quiescent array pattern for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

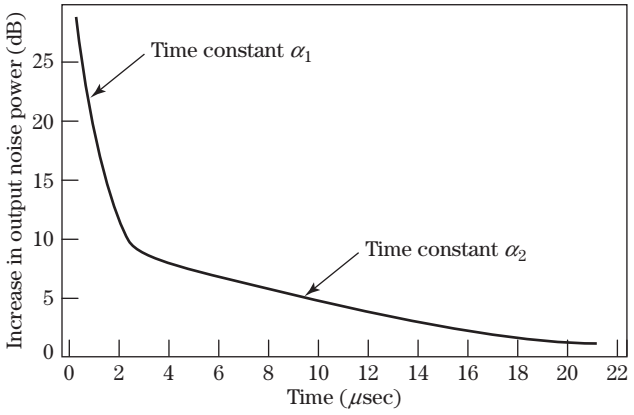


FIGURE 4-17 ■ Array transient response for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

this two-interference source example, where it is seen that the response has two distinct slopes associated with the two distinct (and widely different) eigenvalues.

4.3.4 Example of *N*-Element Linear Adaptive Array Performance

This section presents results from an eight-element linear array with Howells–Applebaum loops.

4.3.4.1 Assumptions and Initial Conditions

Only the quiescent receiver noise is present in each channel up to time $t = 0$, when all the external interference sources are switched on in a single step function. The quiescent RMS noise voltage in all channels equals the constant n_0, such that the quiescent eigenvalue λ_0 is $|n_0|^2$. This defines the quiescent servo gain factor $\mu_{\min}$ by way of (4.138). For the configuration of Figure 4-11, it is convenient to choose $\mu_{\min} = 1$, so the amplifier gains G are set accordingly. Once $\mu_{\min}$ is selected, it is convenient to express μ_i from (4.123) as a ratio of eigenvalues,

$$\frac{\mu_i}{\mu_{\min}} = \frac{\lambda_i}{\lambda_0} \tag{4.188}$$

The quiescent steered-beam pattern $AF_q(\theta)$ and its associated quiescent weight vector $\mathbf{w}_q$ are given by (4.87)–(4.90). The eight-element linear array has an element spacing $\lambda/2$, $\mu = \pi/2 \sin\theta$, and $a_k = 1$. The quiescent weights and array factor are given by

$$w_{q_k} = e^{-j\psi_0(2k-9)/2} \tag{4.189}$$

$$AF_q(\theta) = \frac{\sin[8(\psi - \psi_0)/2]}{\sin[(\psi - \psi_0)/2]} \tag{4.190}$$

The coefficients of the input beam steering vector $\mathbf{b}^*$ are found from (4.140) and (4.88)

$$c_k = (1 + \mu_{\min}) = 2 \tag{4.191}$$

$$b_k^* = c_k w_{q_k} = 2e^{-j\psi_0(2k-9)/2} \tag{4.192}$$

The maximum power condition for each of the orthonormal loops of Figure 4-12b is

$$\mu_{\max} = \mu_{\min}\left(\frac{\lambda_{\max}}{\lambda_0}\right) = \left(\frac{\pi B_c \tau_0}{10}\right) - 1 \tag{4.193}$$

where $\lambda_{\max}$ represents the maximum eigenvalue. The channel bandwidth B_c and filter time constant τ_0 are the same for all element channel servo loops. Solving for τ_0 from (4.193) yields

$$\tau_0 = \left(\frac{10}{\pi B_c}\right)\left[1 + \mu_{\min}\left(\frac{\lambda_{\max}}{\lambda_0}\right)\right] = \left(\frac{10}{\pi B_c}\right)\left[1 + \mu_{\min} + \mu_{\min}\sum_{r=1}^{R} P_r g_m^2(\theta_r)\right] \tag{4.194}$$

The maximum power (maximum eigenvalue) is much larger than the jammer-to-receiver-noise power ratios, because the various P_r are multiplied by the power gain of the eigenvector beams.

4.3.4.2 Output Noise Power and SNR Degradation

The output SNR of the adaptive array compared with the output SNR of a conventional array under the same interference conditions is the performance characteristic of ultimate interest. Instead of forming the actual SNR, it is sufficient to consider the output noise power by itself to illustrate the system transient behavior. Since the receiver noise and external interference sources are statistically independent, the total output noise power is the sum of the two separate output noise powers.

The receiver noise output power can be expressed as

$$|y_{0_n}(t)|^2 = \sum_{k=1}^{N} |w_k n_k|^2 = \sum_{i=1}^{N} |w_i' n_0|^2 \tag{4.195}$$

Substituting for w_i' from (4.181) then yields

$$|y_{0_n}(t)|^2 = |\overline{n}_0|^2 \sum_{i=1}^{k} [1 - A_i(t)]^2 |w_{q_i}'|^2 \tag{4.196}$$

where

$$A_i(t) = (1 - e^{-\alpha_i t})\left[\frac{\mu_i - \mu_0}{1 + \mu_i}\right] \tag{4.197}$$

From (4.197) it is seen that $A_i(t)$ is zero for $t = 0$ and for $\mu_i = \mu_0$ (for nonunique eigenvalues). Therefore, for quiescent conditions at $t = 0$, it follows that

$$|y_{0_n}(0)|^2 = |\overline{n}_0|^2 \sum_{i=1}^{N} |w'_{q_i}|^2 = |\overline{n}_0|^2 \sum_{k=1}^{N} |w_{q_k}|^2 \tag{4.198}$$

since the output noise power must be the same for either the real system or the equivalent orthonormal system. Consequently, (4.196) is rewritten as

$$|y_{0_n}(t)|^2 = |\overline{n}_0|^2 \sum_{k=1}^{N} |w_{q_k}|^2 - \sum_{i=1}^{N} [2 - A_i(t)] A_i(t) |w'_{q_i}|^2 \tag{4.199}$$

Equation (4.199) is a particularly convenient form because the w'_{q_i} associated with nonunique eigenvalues need not be evaluated since $A(t) = 0$ for such eigenvalues.

The output noise power contributed by R external interference sources is given by the sum of their output power pattern levels:

$$|y_{0_j}(t)|^2 = |\overline{n}_0|^2 \sum_{r=1}^{R} P_r AF^2(\theta_r, t) \tag{4.200}$$

where P_r is the rth source power ratio, θ_r is its angular location, and $AF(\theta_r, t)$ is given by (4.184).

The total output noise power is the sum of (4.199) and (4.200), and the increase in the output noise power (with interference sources turned on) is this sum over the quiescent noise (4.198).

$$\frac{|y_0(t)|^2}{|y_{0_n}(0)|^2} = 1 + \left\{ \frac{\sum_{r=1}^{R} P_r AF^2(\theta_r, t) - \sum_{i=1}^{N} [2 - A_i(t)] A_i(t) |w'_{q_i}|^2}{\sum_{k=1}^{N} |w_{q_k}|^2} \right\} \tag{4.201}$$

The output noise power increase in (4.201) indicates the system transient behavior. An increase in output noise power indicates the general magnitude of the adapted (steady-state) weights.

The degradation in the SNR, D_{sn}, enables one to normalize the effect of adapted-weight magnitude level. This degradation is the quiescent SNR divided by the adapted SNR.

$$D_{sn} = \left(\frac{AF_q^2(\theta_s)}{AF^2(\theta_s, t)} \right) \left(\frac{|y_0(t)|^2}{|y_{0_n}(0)|^2} \right) \tag{4.202}$$

where the ratio in the second factor is just (4.201), the increase in output noise power.

4.3.4.3 Eigenvalues and Eigenvectors of the Noise Covariance Matrix

For computational convenience, receiver noise is unity and all noise powers expressed as ratios to receiver noise power. Adopting this convention, the quiescent noise matrix $\mathbf{R}_{nn_q}$ is an identity matrix, and with R narrowband interference sources the noise covariance

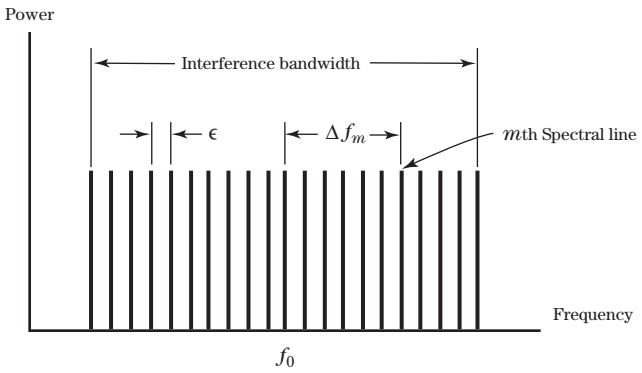


FIGURE 4-18 ■ Interference source power spectrum having uniform amplitude spectral lines spaced ε apart. From Gabriel, *Proc. IEEE,* February 1976.

matrix becomes

$$\mathbf{R}_{nn} = \mathbf{I} + \sum_{r=1}^{R} P_r \mathbf{M}_r \tag{4.203}$$

where $\mathbf{M}_r$ now represents the covariance matrix due to the rth interference source.

Wideband interference sources are represented by dividing the jammer power spectrum into a series of discrete spectral lines. A uniform amplitude spectrum of uncorrelated lines spaced apart by a constant frequency increment ε is once again assumed as illustrated in Figure 4-18. If P_r is the power ratio of the entire jammer power spectrum, then the power ratio of a single spectral line (assuming a total of L_r spectral lines) is

$$P_{rl} = \left(\frac{P_r}{L_r}\right) \tag{4.204}$$

Furthermore, if $B_r (B_r <$ element channel receiver bandwidth, $B_c)$ denotes the percent bandwidth of the jamming spectrum, then the frequency offset of the lth spectral line is

$$\frac{\Delta f_l}{f_0} = \left(\frac{B_r}{100}\right)\left[-\frac{1}{2} + \left(\frac{l-1}{L_r - 1}\right)\right] \tag{4.205}$$

The covariance matrix with R broadband interference sources is written as

$$\mathbf{R}_{nn} = \mathbf{I} + \sum_{r=1}^{R} \sum_{l=1}^{L_r} P_{rl} \mathbf{M}_{rl} \tag{4.206}$$

The mnth component (mth row and nth column) of the matrix $\mathbf{M}_{rl}$ is in turn given by

$$(\mathbf{M}_{rl})_{mn} = e^{j\psi_{rl}(n-m)} \tag{4.207}$$

where

$$\psi_{rl} = \left(\frac{f_l}{f_0}\right)\pi \sin\theta_r = \left(1 + \frac{\Delta f_l}{f_0}\right)\pi \sin\theta_r \tag{4.208}$$

4.3.4.4 Performance Characteristics for Various Signal Conditions

Four narrowband sources located in the sidelobe region of the quiescent beam pattern yield four distinct eigenvalues and require four degrees of freedom to provide the eigenvector

beams required to place nulls at the jammer locations. If the adaptive weight adjustments are large, there may be appreciable main beam distortion in the overall adapted pattern.

The Howells–Applebaum adaptive loop has one adaptive weight in each element channel of the array; this configuration works interference sources with a bandwidth of up to about 20%. Gabriel [22] gives two examples as follows: a 2% bandwidth source in the sidelobe region for which two degrees of freedom (two pattern nulls) are required to provide proper cancellation; and a 15% bandwidth source in the sidelobe region for which three degrees of freedom are required. Broadband interference sources require a transversal equalizer in each element channel (instead of a single adaptive weight) for proper compensation, with a Howells–Applebaum adaptive loop then required for every tap appearing in the tapped delay line.

The adapted pattern for main beam nulling exhibits severe distortions. For interference sources located in the main beam, the increase in output noise power is an unsatisfactory indication of array performance, because there is a net SNR degradation due to the resulting main beam distortion in the adapted pattern. Main beam constraints for such cases can be introduced.

4.3.5 Hard Limiter Modification for N Adaptive Loops

It was shown in Section 4.3.2 that the adaptive array performance depends on the external noise field as well as on the parameters of the adaptive control loops. The power level and angular location of the external noise field determine the noise covariance matrix and therefore its eigenvalues. The eigenvalues, in turn, directly affect the array performance, since both the transient response of the adaptive array and the control loop noise depend explicitly on these eigenvalues. For a nonstationary signal environment, wide variations in array performance may occur, ranging from excessive control loop noise (when the interference is strong) to very slow convergence (when the interference is weak). Introducing a hard limiter into the adaptive control loop reduces the effects of varying noise intensity, and the dynamic range of signals in the control loops are reduced without degrading array performance [28]. Figure 4-19 shows a six-element linear array with a hard limiter introduced in the conjugate signal branches.

With the signal envelopes hard limited, the input to the correlation mixers changes from x_k^* to $u_k^* = x_k^*/|x_k|$. This normalization removes amplitude variations in the conjugate signals but retains the phase variations. The correlation mixer voltage v_k is now given by

$$v_k' = k^2 \left(u_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.209}$$

On comparing (4.209) with (4.97), it is seen that u_k^* has simply replaced x_k^*, so the resulting adaptive weight matrix differential equation now becomes

$$\tau_0 \frac{d\mathbf{w}}{dt} + [\mathbf{I} + \gamma' \mathbf{M}]\mathbf{w} = \mathbf{b}^* \tag{4.210}$$

which is analogous to (4.109) with $\mathbf{M}$ replacing $\mathbf{R}_{nn}$ and $\gamma' = k^2 G'$ where $\mathbf{M}$ is the modified noise covariance with envelope limiting having elements given by

$$M_{ml} = E\left\{ \frac{x_m^* x_l}{|x_m|} \right\} \tag{4.211}$$

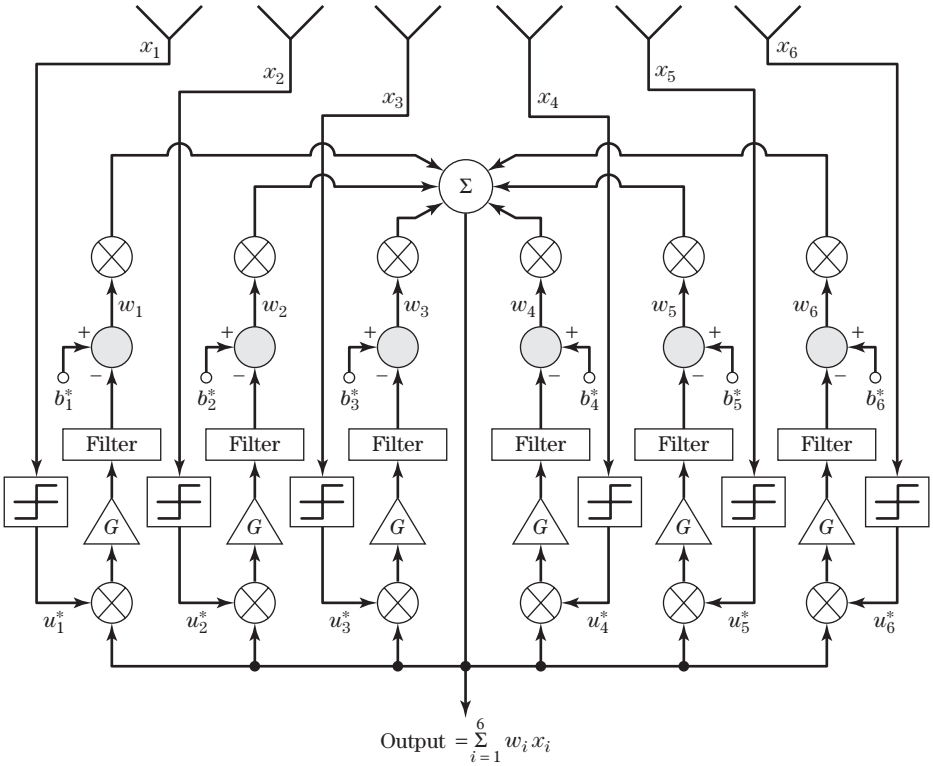


FIGURE 4-19 ■ Hard limiter modification of linear six-element adaptive array system. From Gabriel, *Proc. IEEE*, February 1976.

Assuming the quadrature components of each signal x_k are zero-mean Gaussian random variables having variance σ^2, we can then compute the elements of the covariance matrix $\mathbf{M}$ directly from the elements of $\mathbf{R}_{nn}$ by using the relation [28]

$$M_{ml} = \sqrt{\frac{\pi}{8}} \frac{1}{\sigma} (\mathbf{R}_{nn})_{ml} \tag{4.212}$$

It follows that the elements of $\mathbf{M}$ differ from the elements of $\mathbf{R}_{nn}$ by a common factor $(1/\sigma)\sqrt{(\pi/8)}$. Consequently, the effective time constants that determine the rate of convergence and control loop noise are changed by this same factor, thereby reducing the dependence of array performance on the strength of the external noise field.

It is worthwhile noting that limiting does not change the relative values of the signal covariance matrix elements or the relative eigenvalue magnitudes presuming identical channels. Thus, for widely different eigenvalues, limiting does reduce the eigenvalue spread to provide rapid transient response and low control loop noise. Nevertheless, limiting always reduces the dynamic range of signals in the control loops, thereby simplifying the loop implementation.

4.4 | INTRODUCTION OF MAIN BEAM CONSTRAINTS

As a result of introducing beam pattern nulls, the main beam may become distorted, resulting in a degradation in SNR performance. To prevent such array performance degradation from occurring, it is possible to introduce constraints so the adaptive processor maintains

desired mainlobe signals while realizing good cancellation of interference in the sidelobes. The constraint methods discussed here follow the development that is given by Applebaum and Chapman [29].

Techniques for applying main beam constraints to limit severe array pattern degradation include the following:

1. Time domain: The array adapts when the desired signal is not present in the main beam. These weights are kept until the next adaptation or sampling period. This approach does not protect against main beam distortion resulting from main beam jamming and is also vulnerable to blinking jammers.
2. Frequency domain: When the interference sources have much wider bandwidths than the desired signal, the adaptive processor is constrained to adapt to signals only outside the desired signal bandwidth. This approach somewhat degrades the cancellation capability and distorts the array factor.
3. Angle domain: Three angle domain techniques provide main beam constraints in the steady state: (a) pilot signals; (b) preadaptation spatial filters; and (c) control loop spatial filters. These techniques are also helpful for constraining the array response to short duration signals, since they slow down the transient response to main beam signals. The angle domain techniques provide the capability of introducing main beam constraints into the adaptive processor response.

4.4.1 Pilot Signals

To illustrate the use of pilot signal techniques, consider the multiple sidelobe canceller (MSLC) adaptive array configuration shown in Figure 4-20, where an integrator with feedback structure is taken to represent the integrating filter in the Howells–Applebaum

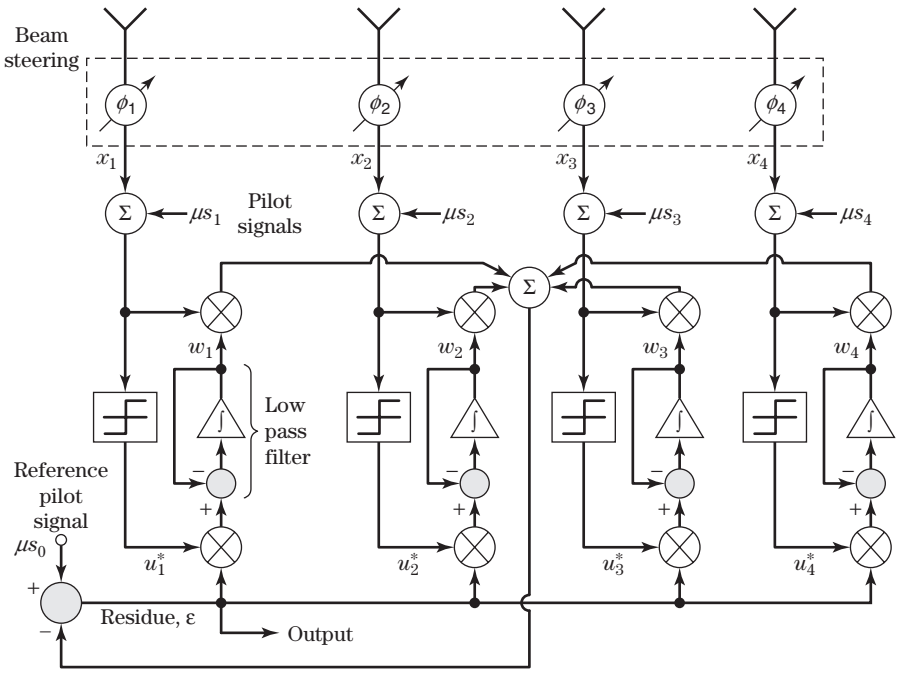


FIGURE 4-20 ■ Multiple sidelobe canceller (MSLC) adaptive array configuration with beam steering pilot signals and main beam control.

adaptive loop. The "pilot signals" shape the array beam and maintain the main beam gain (avoiding SNR degradation). The pilot signals are continuous wave (CW) tones injected into each element channel at a frequency that is easily filtered out of the signal bandwidth. It is not necessary to use the beam steering phase shifters shown in Figure 4-20, since if they are not present the pilot signals may be injected with the proper phase relationship corresponding to the desired main beam direction instead of in phase with each other as shown. The amplitudes and phases of the injected pilot signals $s_1, \ldots, s_4$ may be represented by the vector $\mu\mathbf{s}$, where $\mathbf{s}$ has unit length, and μ is a scalar amplitude factor. The reference channel (or main beam) signal is represented by the injected pilot signal s_0.

For the adaptive control loops shown in Figure 4-20, it follows that the vector differential equation for the weight vector is written as

$$\frac{d\mathbf{w}}{dt} = \mathbf{u}^*(t)\varepsilon(t) - \mathbf{w}(t) \tag{4.213}$$

Since $\varepsilon = \mu s_0 - \mathbf{x}^T\mathbf{w}$, it follows (4.213) and the results of Section 4.3.5 that

$$\frac{d\mathbf{w}}{dt} = g\mu\mathbf{r}_{xs_0} - [\mathbf{I} + g\mathbf{R}_{xx}]\mathbf{w} \tag{4.214}$$

where g is a gain factor representing the correlation mixer gain and the effect of the limiter. The steady-state solution of (4.214) is given by

$$\mathbf{w}_{ss} = [\mathbf{I} + g\mathbf{R}_{xx}]^{-1} g\mu\mathbf{r}_{xs_0} \tag{4.215}$$

In the absence of any desired signal, then

$$\mathbf{x} = \mathbf{n} + \mu\mathbf{s} \tag{4.216}$$

where $\mathbf{n}$ is the noise signal vector, and $\mu\mathbf{s}$ is the injected pilot signal vector. Consequently,

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^*\mathbf{x}^T\} = \mathbf{R}_{nn} + \mu\mathbf{s}^*\mathbf{s}^T \tag{4.217}$$

$$\mathbf{r}_{xs_0} \triangleq E\{\mathbf{x}^* s_0\} = \mu\mathbf{s}^* s_0 \tag{4.218}$$

On substituting (4.217) and (4.218) into (4.215) it can be shown that

$$\mathbf{w}_{ss} = \frac{\mathbf{K}^{-1}\mathbf{s}^* g\mu^2 s_0}{1 + g\mu^2\mathbf{s}^T\mathbf{K}^{-1}\mathbf{s}^*} \tag{4.219}$$

where $\mathbf{K} = \mathbf{I} + g\mathbf{R}_{nn}$. Substituting the expression for $\mathbf{K}^{-1}$ in (4.219) then yields

$$\mathbf{w}_{ss} = \frac{(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^* g\mu^2 s_0}{1 + g\mu^2\mathbf{s}^T(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*} \tag{4.220}$$

For large pilot signals $\mu^2 \to \infty$, and (4.220) becomes

$$\mathbf{w}_{ss} \cong \left[\frac{(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*}{\mathbf{s}^T(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*}\right] s_0 \tag{4.221}$$

If $\mathbf{s}$ has equal amplitude components, then the main beam response from (4.221) is

$$\mathbf{s}^T\mathbf{w}_{ss} \cong s_0 \tag{4.222}$$

which is a constant, independent of $\mathbf{R}_{nn}$ (and hence independent of any received waveforms).

The array configuration of Figure 4-20 uses one set of pilot signals for a single main beam constraint. Multiple constraints require multiple sets of pilot signals, with each set at a different frequency. Pilot signals are inserted close to the input of each element channel to compensate for any amplitude and phase errors. Strong pilot signals require channel elements with a large dynamic range, so they must be filtered to avoid interfering with the desired signal.

4.4.2 Preadaption Spatial Filters

Preadaption spatial filtering forms two beams following the beam steering phase shifters (Figure 4-21). The quiescent pattern main beam is formed with fixed weights $\mathbf{s}^*$ (forming a unit length weight vector). The second beam (termed a *cancellation beam*) is formed adaptively by an MSLC whose input channels are obtained from spatial filtering represented by the matrix transformation $\mathbf{A}$. $\mathbf{A}$ has one less output channel than the number of sensor elements and is selected to maintain a constant response in the main beam direction so that $\mathbf{As} = 0$.

From the signal vector definitions of Figure 4-21 it follows that

$$e_m = \mathbf{s}^\dagger \mathbf{x} = \mathbf{x}^T \mathbf{s}^* \tag{4.223}$$

$$e_0 = e_m - \mathbf{y}^T \mathbf{u} \tag{4.224}$$

$$\mathbf{u} = \mathbf{Ax} \tag{4.225}$$

Therefore

$$e_0 = \mathbf{s}^\dagger \mathbf{x} - \mathbf{y}^T \mathbf{Ax} = (\mathbf{s}^* - \mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{w}^T \mathbf{x} \tag{4.226}$$

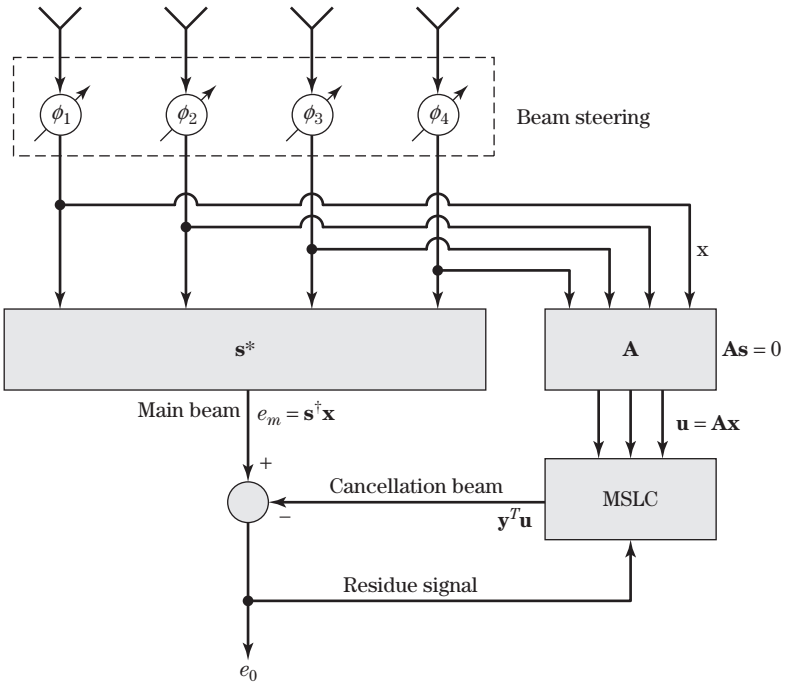


FIGURE 4-21 ■ General structure of preadaption spatial filtering.

The composite weight vector for the entire system can therefore be written as

$$\mathbf{w} = \mathbf{s}^* - \mathbf{A}^T\mathbf{y} \tag{4.227}$$

Consequently

$$\mathbf{s}^T\mathbf{w} = \mathbf{s}^T(\mathbf{s}^* - \mathbf{A}^T\mathbf{y}) = \|\mathbf{s}\|^2 - \mathbf{s}^T\mathbf{A}^T\mathbf{y} \tag{4.228}$$

Since $\mathbf{A}$ was selected so that $\mathbf{As} = 0$, it follows that

$$\mathbf{s}^T\mathbf{w} = \|\mathbf{s}\|^2 = 1 \tag{4.229}$$

Denote the covariance matrix associated with $\mathbf{u}$ by

$$\mathbf{R}_{uu} \triangleq E\{\mathbf{u}^*\mathbf{u}^T\} = E\{\mathbf{A}^*\mathbf{x}^*\mathbf{x}^T\mathbf{A}^T\} = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T \tag{4.230}$$

The MSLC unit generates a weight vector $\mathbf{y}$ that satisfies (4.213).

$$\frac{d\mathbf{y}}{dt} = g\mathbf{u}^*e_0 - \mathbf{y} \tag{4.231}$$

In the steady state [where $(d\mathbf{y}/dt) = 0$] it follows that

$$[\mathbf{I} + g\mathbf{R}_{uu}]\mathbf{y} = gE\{\mathbf{u}^*e_m\} \tag{4.232}$$

where g is a gain factor, and the right side of (4.232) represents the cross-correlation vector of e_m with each component of $\mathbf{u}$. Using (4.223), (4.225), and (4.230) in (4.232), we find that

$$\left(\mathbf{I} + g\mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T\right)\mathbf{y} = g\mathbf{A}^*\mathbf{R}_{xx}\mathbf{s}^* \tag{4.233}$$

Premultiply (4.233) by $\mathbf{A}^T$ and use (4.227); it then follows that the composite weight applied to the input signal vector satisfies the steady-state relation

$$\left(\mathbf{I} + g\mathbf{A}^T\mathbf{A}^*\mathbf{R}_{xx}\right)\mathbf{w}_{ss} = \mathbf{s}^* \tag{4.234}$$

when $\mathbf{x}$ does not contain a desired signal component, then $\mathbf{R}_{xx}$ may be replaced by $\mathbf{R}_{nn}$.

Now allow g to become very large so that (4.233) yields

$$\mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T\mathbf{y} = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{s}^* \tag{4.235}$$

$$\mathbf{A}^*\mathbf{R}_{xx}(\mathbf{s}^* - \mathbf{A}^T\mathbf{y}) = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{w}_{ss} = 0 \tag{4.236}$$

Since $\mathbf{As} = 0$ and the rank of the transformation matrix $\mathbf{A}$ is $N - 1$, (4.236) implies that $\mathbf{R}_{xx}\mathbf{w}$ is proportional to $\mathbf{s}^*$ so that

$$\mathbf{R}_{xx}\mathbf{w}_{ss} = \mu\mathbf{s}^* \tag{4.237}$$

$$\mathbf{w}_{ss} = \mu\mathbf{R}_{xx}^{-1}\mathbf{s}^* \tag{4.238}$$

where μ is a proportionality constant that may be evaluated using (4.229). Substituting $\mu = (\mathbf{s}^T\mathbf{R}_{xx}^{-1}\mathbf{s}^*)^{-1}$ in (4.238) then yields

$$\mathbf{w}_{ss} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{s}^*}{\mathbf{s}^T\mathbf{R}_{xx}^{-1}\mathbf{s}^*} \tag{4.239}$$

as the solution that the composite weight vector approaches when g becomes very large.

Preadaption spatial filtering avoids dynamic range problems, so it does require the implementation of multiple beams. The accuracy of the beam steering phase shifters limits the effectiveness of the constraints, but this limit is true of all three methods considered here. Two realizations of preadaption spatial filtering represented by Figure 4-21 include [29]: (1) the use of a Butler matrix to obtain orthogonal beams, one of which is regarded as the "main" beam; and (2) the use of an **A** matrix transformation obtained by fixed element-to-element subtraction.

4.4.3 Control Loop Spatial Filters

The Howells–Applebaum adaptive control loop with constraints applied directly in the loop by means of a spatial matrix filter is illustrated by the configuration of Figure 4-22. The spatial matrix filter removes any components of the signal vector **v** pointing in the direction of the unit length beam steering vector $\mathbf{b}^*$ by means of a projection operator. The successful removal of such signal components then constrains the array response in the direction of **b**.

The amplified output of the correlation mixer configuration of Figure 4-22 is given by

$$\mathbf{v}(t) = \mathbf{u}^*(t)\mathbf{x}^T(t)\mathbf{w}(t) \tag{4.240}$$

Taking expected values of (4.243) results in the steady-state values given by

$$\mathbf{v} = g\mathbf{R}_{xx}\mathbf{w}_{ss} \tag{4.241}$$

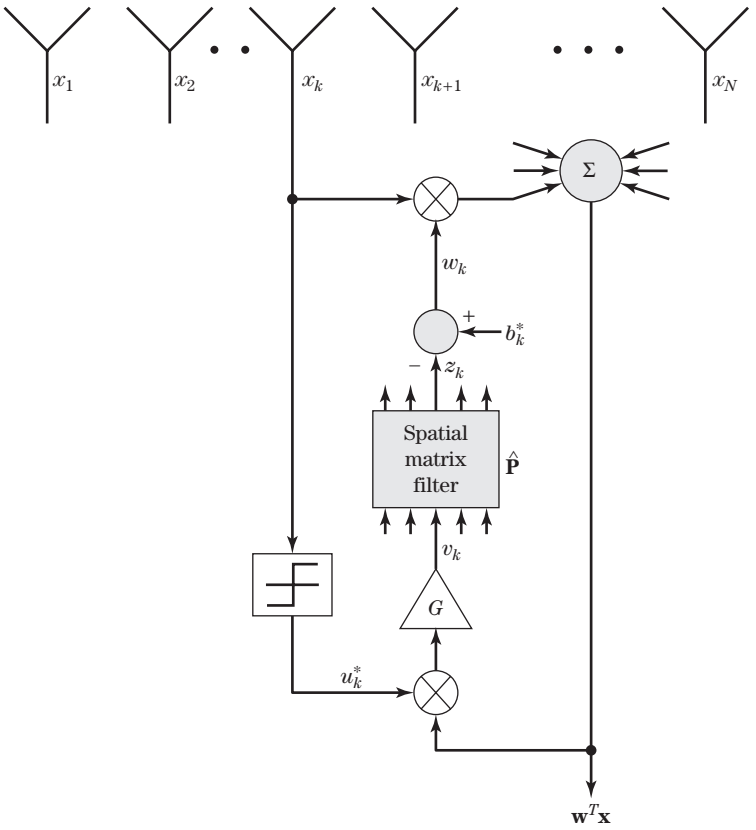


FIGURE 4-22 ■ Adaptive processor with control loop spatial filtering.

where g is a gain factor. Now

$$\mathbf{z} = \hat{\mathbf{P}}\mathbf{v} = g\hat{\mathbf{P}}\mathbf{R}_{xx}\mathbf{w}_{ss} \tag{4.242}$$

Substituting $\mathbf{w} = \mathbf{b}^* - \mathbf{z}$ into (4.245) means the steady-state weight values must satisfy

$$(\mathbf{I} + g\hat{\mathbf{P}}\mathbf{R}_{xx})\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.243}$$

When the beam steering vector is uniformly weighted, the projection performed by the spatial filter to remove signal components in the direction of $\mathbf{b}$ is

$$\hat{\mathbf{P}} = \mathbf{I} - \mathbf{b}^*\mathbf{b}^T \tag{4.244}$$

Substituting (4.244) into (4.243) results in

$$(\mathbf{I} + g\mathbf{R}_{xx} - g\mathbf{b}^*\mathbf{b}^T\mathbf{R}_{xx})\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.245}$$

$$[(\mathbf{I} + g\mathbf{R}_{xx})\mathbf{R}_{xx}^{-1} - g\mathbf{b}^*\mathbf{b}^T]\mathbf{R}_{xx}\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.246}$$

Rewrite (4.246) in the form

$$\mathbf{R}_{xx}\mathbf{w}_{ss} = [\mathbf{Q} - g\mathbf{b}^*\mathbf{b}^T]^{-1}\mathbf{b}^* \tag{4.247}$$

where $\mathbf{Q} \triangleq (\mathbf{I} + g\mathbf{R}_{xx})\mathbf{R}_{xx}^{-1}$, and apply a matrix inversion identity to obtain the result

$$[\mathbf{Q} - g\mathbf{b}^*\mathbf{b}^T]^{-1}\mathbf{b}^* = \frac{\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*}{1 - g\mathbf{b}^T\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*} \tag{4.248}$$

The denominator of (4.243) may be simplified as

$$\begin{aligned} g\mathbf{b}^T\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* &= \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx} - \mathbf{I})(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \\ &= \mathbf{b}^T\mathbf{b}^* - \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \\ &= 1 - \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \end{aligned} \tag{4.249}$$

Substituting (4.248) and (4.249) into (4.247) then yields

$$\mathbf{w}_{ss} = \frac{(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*}{\mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*} \tag{4.250}$$

The introduction of $\hat{\mathbf{P}} = \mathbf{I} - \mathbf{b}^*\mathbf{b}^T$ constrains the array response only in the direction of $\mathbf{b}$. Additional constraints are required to constrain the response over a finite region of the main lobe instead of only one direction. Likewise, when the steering vector is not uniform, an additional constraint is also required to obtain the desired quiescent pattern. It is furthermore desirable to transform the constraints to an orthogonal set, thereby minimizing the accuracy requirements of the spatial matrix filter. To illustrate a projection filter constructed from an orthogonal set, consider the representation of a projection filter with $M + 1$ orthogonal constraints:

$$\hat{\mathbf{P}} = \mathbf{I} - \sum_{m=0}^{M} \mathfrak{c}_m\mathfrak{c}_m^\dagger \tag{4.251}$$

where $\mathfrak{c}_m\mathfrak{c}_n = \delta_{mn}$, δ_{mn} = the Kronecker delta, and the $\mathfrak{c}_m$ are the constraint vectors.

The constraint that maintains the array response at the peak of the beam is the "zero-order" constraint. The weight vector solution obtained with a zero-order constraint differs from the unconstrained solution ($\hat{\mathbf{P}} = \mathbf{I}$) by a multiplicative scale factor. Multiple constraints are typically used to increase the beam constraint zone by controlling the first few derivatives of the pattern function in the direction of interest. A constraint that controls the mth derivative is referred to as an "mth-order" constraint.

To synthesize a $\mathfrak{c}_m$ constraint vector corresponding to the mth derivative of the pattern function, note that the pattern function of a linear array can be written as

$$AF(\theta) = \sum_{k=1}^{N} w_k e^{jk\theta} \tag{4.252}$$

The mth derivative of $AF(\theta)$ is

$$AF^m(\theta) = \sum_{k=1}^{N} (jk)^m w_k e^{jk\theta} \tag{4.253}$$

Consequently, the elements of $\mathfrak{c}_m$ (for $m = 0$, 1, and 2) are given by

$$\mathfrak{c}_{0i} = d_0 \tag{4.254}$$

$$\mathfrak{c}_{1i} = e_0 + e_1 i \tag{4.255}$$

$$\mathfrak{c}_{2i} = f_0 + f_1 i + f_2 i^2 \tag{4.256}$$

The constants defining the $\mathfrak{c}_m$ elements are made unit length and mutually orthogonal.

Consider how to establish a beam having nonuniform weighting as well as zero-, first-, and second-order constraints on the beam shape at the center of the main beam. First expand $\mathbf{w}_q$ in terms of the constraint vectors $\mathfrak{c}_m$ (for $m = 0$, 1, and 2) and a remainder vector $\mathfrak{c}_r$ as

$$\mathbf{w}_q = a_0\mathfrak{c}_0 + a_1\mathfrak{c}_1 + a_2\mathfrak{c}_2 + a_r\mathfrak{c}_r \tag{4.257}$$

where

$$a_i = \mathbf{w}_q^T \mathfrak{c}_i \quad \text{for } i = 0, 1, 2 \tag{4.258}$$

$$a_r\mathfrak{c}_r = w_q - \sum_{i=0}^{2} a_i\mathfrak{c}_i \tag{4.259}$$

Now, construct the complementary projection matrix filter according to

$$\hat{\mathbf{P}} = \mathbf{I} - \mathfrak{c}_0\mathfrak{c}_0^\dagger - \mathfrak{c}_1\mathfrak{c}_1^\dagger - \mathfrak{c}_2\mathfrak{c}_2^\dagger - \cdots - \mathfrak{c}_r\mathfrak{c}_r^\dagger \tag{4.260}$$

The subspace spanned by the constraint vectors in the N-dimension space of the adaptive processor is preserved by the foregoing construction. The spatial matrix filter constructed according to (4.263) results in a signal vector $\mathbf{z}$ containing no components in the direction of $\mathbf{w}_q$ or its first and second derivatives. The vector $\mathbf{w}_q$ is now added back in (at the point in Figure 4-21 where $\mathbf{b}^*$ is inserted) to form the final weight vector $\mathbf{w}$.

4.5 | CONSTRAINT FOR THE CASE OF KNOWN DESIRED SIGNAL POWER LEVEL

If there is no a priori knowledge of the desired signal structure or the direction of arrival, acquiring a weak desired signal in the presence of strong jamming or interference is accomplished by placing a constraint on the adaptation algorithm to prevent suppression of all signals (including interference) below a specified input power level. When a signal power threshold is greater than the desired signal input power, the weak desired signal will not be suppressed, whereas all interference signals above the threshold are suppressed.

The most common method of obtaining a power discrimination capability was formulated by Compton [30] and is based on the use of proportional feedback control in a Howells–Applebaum adaptive loop. To accomplish the same result Zahm [31] proposed another technique that uses a combination of a steering command vector and a bias signal. The weight adjustment control loops for the adaptive null-steering array in Figure 4-23 are governed by the differential equation

$$\frac{d\mathbf{w}}{dt} = \alpha\{\mathbf{x}(t)[x_0(t) - \mathbf{x}^{\dagger}(t)\mathbf{w}] - a\mathbf{w}\} \tag{4.261}$$

where a is a real scalar constant. The additional feedback path around the integrator provides the means for setting a power threshold.

For α sufficiently small, (4.261) can be approximated by expected values so that

$$\frac{d\mathbf{w}}{dt} = \alpha(\mathbf{r}_{xx_0} - [\mathbf{R}_{xx} + a\mathbf{I}]\mathbf{w}) \tag{4.262}$$

The steady-state weight vector is then given by

$$\mathbf{w} = [\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{r}_{xx_0} \tag{4.263}$$

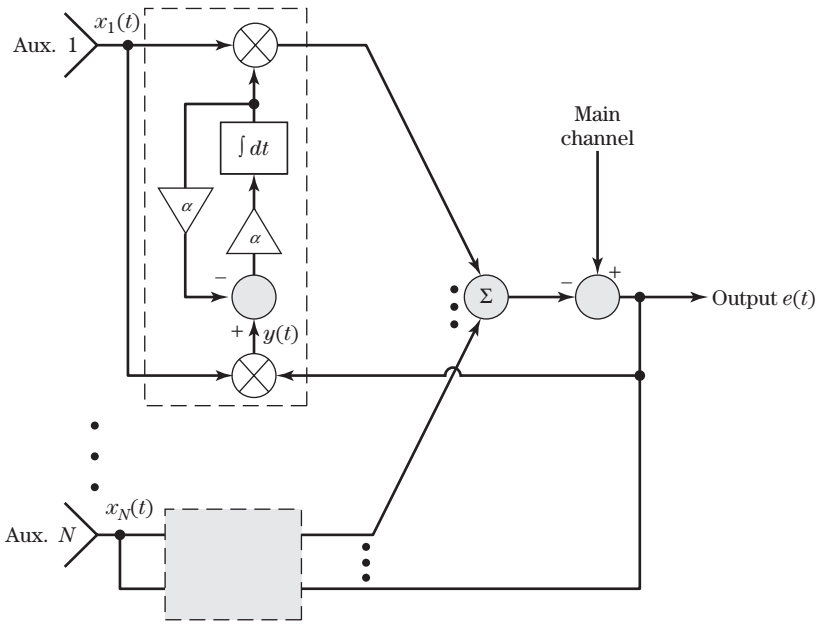


FIGURE 4-23 ■ Adaptive null steering with N auxiliary sensor elements. From Compton, Ohio State Univ. Quarterly Report 3234-1, December, 1971.

If a desired signal is present then the output signal-to-interference plus noise ratio (SINR) as

$$\text{SNR} = \frac{1}{(P_e/\overline{|s_0 - \mathbf{w}^\dagger \mathbf{s}|^2}) - 1} \tag{4.264}$$

where P_e represents the total output power, $\mathbf{s}$ is the desired signal direction vector, and s_0 is the main channel desired signal component. It will be convenient to define the parameter

$$SN' = \frac{\overline{|s_0 - \mathbf{w}^\dagger \mathbf{s}|^2}}{P_e} \tag{4.265}$$

so that

$$\text{SNR} = \frac{1}{(1/SN') - 1} \tag{4.266}$$

It can be shown that

$$SN' = \frac{\left| s_0 - \sum_{i=1}^{N} \frac{(\mathbf{Q}\mathbf{r}_{xx_0})_i^* (\mathbf{Q}\mathbf{s})_i}{\lambda_i + a} \right|^2}{a^2 \sum_{i=1}^{N} \frac{|(\mathbf{Q}\mathbf{r}_{xx_0})_i|^2}{\lambda_i (\lambda_i + a)^2} + P_{e_0}} \tag{4.267}$$

where $\mathbf{Q}$ is the unitary transformation that diagonalizes $\mathbf{R}_{xx}$, λ_i are the eigenvalues of R_{xx}, and P_{e_0} represents the minimum output power of $e(t)$ when $a = 0$. It can be shown from the previous results and (4.261) that a may be selected to prevent cancellation of the desired signal while suppressing high-level jammers. The output signal-to-interference plus noise ratio is therefore higher than when a pure integrator is used in the feedback loop.

To maximize the output SNR, the parameter a is selected to maximize SN'. To illustrate how a is selected, consider the case of one interfering jammer so that

$$\mathbf{R}_{xx} = J \mathbf{v}_J \mathbf{v}_J + P_s \mathbf{v}_s \mathbf{v}_s \tag{4.268}$$

$$\mathbf{r}_{xx_0} = \sqrt{J_0 J}\, \mathbf{v}_J e^{j\phi_J} + \sqrt{P_{s_0} P_s}\, \mathbf{v}_s e^{j\phi_s} \tag{4.269}$$

where

J_0 = main channel jammer power
J = auxiliary channel jammer power (assumed equal in all auxiliary channels)
$\mathbf{v}_J$ = jammer direction delay vector

P_{s_0}, P_s, and $\mathbf{v}_s$ are similarly defined for the desired signal. ϕ_J and ϕ_s represent the relative phase between the main and auxiliary channel signals for the jamming and desired signals, respectively.

If the desired signal and the interference signal angles of arrival are such that $\mathbf{v}_s$ and $\mathbf{v}_J$ are orthogonal (which simplifies the discussion for tutorial purposes), then (4.270) reduces to

$$SN' = \frac{P_{s_0} \left[\dfrac{\sigma^2 + a}{\sigma^2 + a + NP_s} \right]^2}{a^2 N \left[\dfrac{P_{s_0} P_s}{(NP_s + \sigma^2 + a)^2 (NP_s + \sigma^2)} + \dfrac{J_0 J}{(NJ + \sigma^2 + a)^2 (NJ + \sigma^2)} \right] + P_{e_0}} \tag{4.270}$$

where σ^2 = auxiliary channel thermal noise power. For $a = 0$ (which corresponds to the conventional LMS null-steering algorithm) SN' becomes

$$SN' = \frac{P_{s_0}\left[\dfrac{\sigma^2}{NP_s + \sigma^2}\right]^2}{P_{e_0}}; \quad a = 0 \tag{4.271}$$

This result shows that SN' decreases as the input desired signal power in the auxiliary channels NP_s increases above the thermal noise level σ^2. When $NP_s \gg \sigma^2$ in (4.271), SN' is inversely proportional to the input desired signal power, which is the power inversion characteristic of the minimum MSE performance criterion. When $a \gg NP_s + \sigma^2$ and $a \gg NJ + \sigma^2$, (4.270) becomes

$$SN' = \frac{P_{s_0}}{\dfrac{NP_s}{NP_s + \sigma^2}P_{s_0} + \dfrac{NJ}{NJ + \sigma^2}J_0 + P_{e_0}} \tag{4.272}$$

Suppression of the main channel signal is prevented by selecting a to be sufficiently large. However, a is too large in this example, because jammer suppression has also been prevented, as indicated by the presence of the term $NJJ_0/(NJ + \sigma^2)$ in the denominator of (4.272).

Next, assume that the main channel jammer power J_0 is nominally equal to the auxiliary channel jammer power, and choose $a = NP_s$. Then SN' becomes

$$SN' = \frac{0.25P_{s_0}}{\dfrac{P_{s_0}(NP_s - \sigma^2)^2}{4NP_s(NP_s + \sigma^2)} + \dfrac{(NP_s - \sigma^2)^2}{N[1 + (P_s/J)]^2(NJ + \sigma^2)} + P_{e_0}} \tag{4.273}$$

For $J \gg P_s$, $P_s \gg \sigma^2$,

$$SN' \approx \frac{0.25P_{s_0}}{0.25P_{s_0} + P_{e_0}}$$

An approximation for the output signal-to-interference plus noise ratio in (4.269) is

$$\text{SNR} \cong \frac{1}{4}\frac{P_{s_0}}{P_{e_0}} \tag{4.274}$$

Thus, the output signal-to-interference plus noise ratio is now proportional to the main channel signal power divided by the output residue power P_{e_0} (recall that P_{e_0} is the minimum output residue power obtained when $a = 0$). Equation (4.274) shows that when $J \gg P_s$ and $P_s \gg \sigma^2$, the output signal-to-interference plus noise ratio can be significantly improved by selecting the weight feedback gain as

$$a \approx NP_s \tag{4.275}$$

This value of a (when $J \gg P_s$ and $P_s \gg \sigma^2$) then prevents suppression of the relatively weak desired signal while strongly suppressing higher power level jamming signals.

4.6 THE DSD ALGORITHM

We have seen that, if there are perfect gradient measurements on each iteration, the adaptive weight vector converges to the optimal (Wiener) weight vector. In practice, gradient vector estimates are obtained from a limited statistical sample. The DSD algorithm

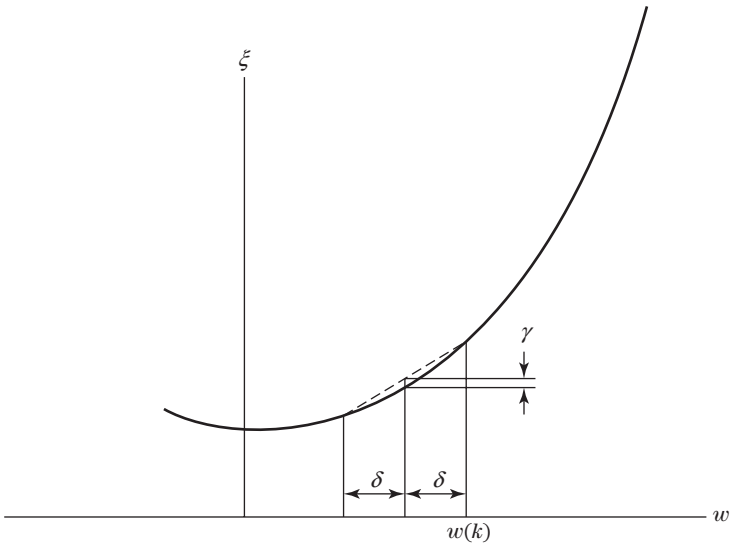


FIGURE 4-24 ■ One-dimensional gradient estimation by way of direct measurement.

obtains gradient vector estimates by direct measurement and is straightforward and easy to implement [13].

The parabolic performance surface representing the MSE function of a single variable w is defined by

$$\xi[w(k)] \triangleq \xi(k) = \xi_{\min} + \alpha w^2(k) \tag{4.276}$$

Figure 4-24 represents the parabolic performance surface as a function of a single component of the weight vector **w**. The first and second derivatives of the MSE are

$$\left[\frac{d\xi(k)}{dw}\right]_{w=w(k)} = 2\alpha w(k) \tag{4.277}$$

$$\left[\frac{d^2\xi(k)}{dw^2}\right]_{w=w(k)} = 2\alpha \tag{4.278}$$

These derivatives are numerically estimated by taking the "symmetric differences"

$$\left[\frac{d\xi(k)}{dw}\right]_{w=w(k)} = \frac{\xi[w(k)+\delta] - \xi[w(k)-\delta]}{2\delta} \tag{4.279}$$

$$\left[\frac{d^2\xi(k)}{dw^2}\right]_{w=w(k)} = \frac{\xi[w(k)+\delta] - 2\xi[w(k)] + \xi[w(k)-\delta]}{\delta^2} \tag{4.280}$$

The procedure for estimating the first derivative illustrated in Figure 4-24 requires that the weight adjustment be altered to two distinct settings while the gradient estimate is obtained. If K data samples are taken to estimate the MSE at the two weight settings $w(k)+\delta$ and $w(k)-\delta$, then the average MSE experienced (over both settings) is greater than the MSE at $w(k)$ by an amount γ. Consequently, a performance penalty is incurred that results from the weight alteration used to obtain the derivative estimate.

4.6.1 Performance Penalty Incurred by Gradient Measurement

Figure 4-24 shows that γ for the one-dimensional case is

$$\gamma = \frac{\alpha[w(k)+\delta]^2 + \alpha[w(k)-\delta]^2 + 2\xi_{\min}}{2} - \alpha w^2(k) - \xi_{\min} = \alpha\delta^2 \tag{4.281}$$

Thus, the value of γ depends only on α and not on $w(k)$. A dimensionless measure of the system perturbation each time the gradient is measured is defined by

$$P \triangleq \frac{\gamma}{\xi_{\min}} = \frac{\alpha\delta^2}{\xi_{\min}} \tag{4.282}$$

The perturbation is the average increase in the MSE normalized with respect to the minimum achievable MSE.

A two-dimensional gradient is needed for the input signal correlation matrix

$$\mathbf{R}_{xx} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{4.283}$$

The MSE corresponding to this correlation matrix is then

$$\xi = r_{11}w_1^2 + r_{22}w_2^2 + 2r_{12}w_1w_2 + \xi_{\min} \tag{4.284}$$

Measuring the partial derivative of the previous performance surface along the coordinate w_1 yields a perturbation

$$P = \frac{r_{11}\delta^2}{\xi_{\min}} \tag{4.285}$$

Likewise, the perturbation for the measured partial derivative along the coordinate w_2 is

$$P = \frac{r_{22}\delta^2}{\xi_{\min}} \tag{4.286}$$

If we allot equal time for the measurement of both partial derivatives (a total of $2K$ data samples are used for both measurements), the average perturbation experienced during the complete measurement process is given by

$$P_{\text{av}} = \frac{\delta^2}{\xi_{\min}} \cdot \frac{r_{11}+r_{22}}{2} \tag{4.287}$$

For N dimensions, define a general perturbation as the average of the perturbations experienced for each of the individual gradient component measurements so that

$$P = \frac{\delta^2}{\xi_{\min}} \cdot \frac{\text{tr}(\mathbf{R}_{xx})}{N} \tag{4.288}$$

where "tr" denotes trace, which is defined as the sum of the diagonal elements of the indicated matrix. When we convert the $\mathbf{R}_{xx}$ matrix to normal coordinates, the trace of $\mathbf{R}_{xx}$ is the sum of its eigenvalues. Since the sum of the eigenvalues divided by N is the average

of the eigenvalues (λ_{av}), the perturbation for the N-dimensional case is

$$P = \frac{\delta^2}{\xi_{min}}\lambda_{av} \tag{4.289}$$

Alternative means of measuring the gradient have also been used in practical systems. By perturbing or dithering a single weight sinusoidally, the cross-correlation between the weight value and the performance measure can be measured to determine the derivative of the performance surface. Likewise, all weights can be dithered simultaneously at distinct individual frequencies and the gradient components then obtained by cross-correlation. The procedure for determining the derivative illustrated in Figure 4-24 corresponds to square-wave dithering.

4.6.2 Gradient Measurement Noise and Misadjustment in the Weight Vector

Gradients measured by taking finite differences as in Figure 4-23 are noisy because the MSE measurements on which the differences are based are noisy. Each MSE measurement is an estimate $\hat{\xi}$ of the actual MSE ξ based on K data samples:

$$\hat{\xi} = \frac{1}{K}\sum_{k=1}^{K} e^2(k) \tag{4.290}$$

It is well known that the variance of a sample average estimate of the mean square obtained from K independent samples is given by the difference between the mean fourth and the square of the mean square all divided by K. Consequently the variance of $\hat{\xi}$ may be expressed as [32]

$$\text{var}[\hat{\xi}] = \frac{E\{e^4(k)\} - [E\{e^2(k)\}]^2}{K} \tag{4.291}$$

If the random variable $e(k)$ is normally distributed with zero mean and variance σ^2, then its mean fourth is $3\sigma^4$, and the square of its mean square is σ^4. Consequently, the variance in the estimate of ξ is given by

$$\text{var}[\hat{\xi}] = \frac{1}{K}(3\sigma^4 - \sigma^4) = \frac{2\sigma^4}{K} = \frac{2\xi^2}{K} \tag{4.292}$$

From (4.292) we find that the variance of $\hat{\xi}$ is proportional to the square of ξ and inversely proportional to the number of data samples. In general, the variance can be expressed as

$$\text{var}[\hat{\xi}] = \eta\frac{\xi^2}{K} \tag{4.293}$$

where η has the value of 2 for an unbiased Gaussian density function. In the event that the probability density function for $\hat{\xi}$ is not Gaussian, then the value of η is generally less than but close to 2. It is therefore convenient to assume that the final result expressed in (4.292) holds for the analysis that follows.

The derivatives required by the DSD algorithm are measured in accordance with (4.279). The measured derivative involves taking finite differences of two MSE estimates, so the error in the measured derivative involves the sum of two independent components

[since the error samples $e(k)$ are assumed to be independent]. The variance of each component to the derivative error is given by (4.292). Assume that we are attempting to measure the derivative at a point on the performance surface where the weight vector is near the minimum point of the MSE surface and that the perturbation P is small, then the two components of measured derivative error will have essentially the same variances. The total variance of the measured derivative error will then be the sum of the variances of the two components. From (4.279) and (4.292) it follows that the variance in the estimate of the derivative is given by

$$\text{var}\left[\frac{d\xi}{dw}\right]_{w=w(k)} = \frac{1}{4\delta^2}\left[\frac{2\xi^2[w(k)+\delta]}{K} + \frac{2\xi^2[w(k)-\delta]}{K}\right] \cong \frac{\xi_{\min}^2}{K\delta^2} \tag{4.294}$$

When an entire gradient vector is measured, then the errors in each component are independent. It is convenient to define a gradient noise vector $\mathbf{g}(k)$ in terms of the true gradient $\quad(k)$ and the estimated gradient $\hat{\ }(k)$:

$$\hat{\ }(k) \triangleq \quad(k) + \mathbf{g}(k) \tag{4.295}$$

where $\mathbf{g}(k)$ is the gradient noise vector. Under the previously assumed conditions, the covariance of the gradient noise vector can be expressed as

$$\text{cov}\,[\mathbf{g}(k)] = \frac{\xi_{\min}^2}{K\delta^2}\mathbf{I} \tag{4.296}$$

Transforming the gradient noise vector into normal coordinates, we have

$$\mathbf{g}'(k) = \mathbf{Q}\mathbf{g}(k) \tag{4.297}$$

We see from (4.296) that the covariance matrix of $\mathbf{g}(k)$ is a scalar multiplying the identity matrix, so projecting into normal coordinates through the orthonormal transformation $\mathbf{Q}$ yields the same covariance for $\mathbf{g}'(k)$:

$$\text{cov}\,[\mathbf{g}'(k)] = E\{\mathbf{Q}\mathbf{g}(k)\mathbf{g}^T(k)\mathbf{Q}^{-1}\} = \frac{\xi_{\min}^2}{K\delta^2}\mathbf{I} \tag{4.298}$$

This result merely emphasizes that near the minimum point of the performance surface the covariance of the gradient noise is essentially a constant and does not depend on $\mathbf{w}(k)$.

The fact that the gradient estimates are noisy means that weight adaptation based on these gradient estimates will also be noisy, and it is consequently of interest to determine the corresponding noise in the weight vector. Using estimated gradients, the method of steepest descent yields the vector difference equation

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s(-\hat{\ }(k)) = \mathbf{v}(k) + \Delta_s[-\quad(k) - \mathbf{g}(k)] \tag{4.299}$$

where $\mathbf{v}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_{\text{opt}}$. Since the true gradient from (4.63) is given by

$$(k) = 2\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.300}$$

(4.299) can be rewritten as

$$\mathbf{v}(k+1) = [\mathbf{I} - 2\Delta_s\mathbf{R}_{xx}]\mathbf{v}(k) - \Delta_s\mathbf{g}(k) \tag{4.301}$$

which is a first-order difference equation having a stochastic driving function $-\Delta_s \mathbf{g}(k)$. Projecting the previous difference equation into normal coordinates by premultiplying by $\mathbf{Q}$ then yields

$$\mathbf{v}'(k+1) = [\mathbf{I} - 2\Delta_s \mathbf{\Lambda}]\mathbf{v}'(k) - \Delta_s \mathbf{g}'(k) \tag{4.302}$$

After initial adaptive transients have died out and the steady state is reached, the weight vector $\mathbf{v}'(k)$ behaves like a stationary random process in response to the stochastic driving function $-\Delta_s \mathbf{g}'(k)$. In the normal coordinate system there is no cross-coupling between terms, and the components of $\mathbf{g}'(k)$ are uncorrelated; thus, the components of $\mathbf{v}'(k)$ are also mutually uncorrelated, and the covariance matrix of $\mathbf{g}'(k)$ is diagonal. The covariance matrix of $\mathbf{v}'(k)$ describes how noisy the weight vector will be in response to the stochastic driving function, and we now proceed to find this matrix. Since $\text{cov}[\mathbf{v}'(k)] \triangleq E\{\mathbf{v}'(k)\mathbf{v}'^T(k)\}$, it is of interest to determine the quantity $\mathbf{v}'(k+1)\mathbf{v}'^T(k+1)$ by way of (4.302) as follows:

$$\begin{aligned}\mathbf{v}'(k+1)\mathbf{v}'^T(k+1) = {} & (\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\mathbf{v}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s \mathbf{\Lambda}) \\ & + \Delta_s^2 \mathbf{g}'(k)\mathbf{g}'^T(k) - \Delta_s(\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\mathbf{v}'(k)\mathbf{g}'(k)\mathbf{v}'^T(k) \\ & - \Delta_s \mathbf{g}'(k)\mathbf{v}'^T(k)\end{aligned} \tag{4.303}$$

Taking expected values of both sides of (4.303) and noting that $\mathbf{v}'(k)$ and $\mathbf{g}'(k)$ are uncorrelated since $\mathbf{v}'(k)$ is affected only by gradient noise from previous adaptive cycles, we obtain for the steady state

$$\begin{aligned}\text{cov}[\mathbf{v}'(k)] &= (\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\text{cov}[\mathbf{v}'(k)](\mathbf{I} - 2\Delta_s \mathbf{\Lambda}) + \Delta_s^2 \text{cov}[\mathbf{g}'(k)] \\ &= \left(\mathbf{I} - 4\Delta_s \mathbf{\Lambda} + 4\Delta_s^2 \mathbf{\Lambda}^2\right)\text{cov}[\mathbf{v}'(k)] + \Delta_s^2 \text{cov}[\mathbf{g}'(k)]\end{aligned} \tag{4.304}$$

Combining like terms in (4.304) then yields

$$\text{cov}[\mathbf{v}'(k)] = \Delta_s^2 \left[4\Delta_s \mathbf{\Lambda} - 4\Delta_s^2 \mathbf{\Lambda}^2\right]^{-1} \text{cov}[\mathbf{g}'(k)] \tag{4.305}$$

In practice, the step size in the method of steepest descent is selected so that

$$\Delta_s \mathbf{\Lambda} \ll \mathbf{I} \tag{4.306}$$

As a result of (4.306), squared terms occurring in (4.305) can be neglected so that

$$\text{cov}[\mathbf{v}'(k)] \cong \frac{\Delta_s}{4} \mathbf{\Lambda}^{-1} \text{cov}[\mathbf{g}'(k)] \tag{4.307}$$

Since $\text{cov}[\mathbf{g}'(k)]$ is given by (4.298), we now have

$$\text{cov}[\mathbf{v}'(k)] \cong \frac{\Delta_s \xi_{\min}^2}{4K\delta^2} \mathbf{\Lambda}^{-1} \tag{4.308}$$

The covariance of the weight vector in the operational coordinate system can be obtained from (4.308) by recalling that $\mathbf{R}_{xx}^{-1} = \mathbf{Q}^{-1}\mathbf{\Lambda}^{-1}\mathbf{Q}$ and $\mathbf{v}' = \mathbf{Q}\mathbf{v}$ so that

$$\begin{aligned}\text{cov}[\mathbf{v}(k)] &= E\left\{\mathbf{Q}^{-1}\mathbf{v}'(k)\mathbf{v}'^T(k)\mathbf{Q}\right\} \\ &= \frac{\Delta_s \xi_{\min}^2}{4K\delta^2} \mathbf{R}_{xx}^{-1}\end{aligned} \tag{4.309}$$

Without any noise in the weight vector, the method of steepest descent converges to a steady-state solution at the minimum point of the MSE performance surface (the bottom of the bowl). The MSE would then be $\xi_{\min}$. The noise present in the weight vector causes the steady-state solution to randomly wander about the minimum point. The result of this wandering is a steady-state MSE that is greater than $\xi_{\min}$ and hence is said to have an "excess" MSE. We will now consider how severe this excess MSE is for the noise that is in the weight vector.

We have already seen in Section 4.1.3 that the MSE can be expressed as

$$\xi(k) = \xi_{\min} + \mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k) \tag{4.310}$$

where $\mathbf{v}'(k) = \mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}$. Consequently, the average excess MSE is

$$E\left\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\right\} = \sum_{p=1}^{N} \lambda_p E\left\{\left[v'_p(k)\right]^2\right\} \tag{4.311}$$

But from (4.308) we may write

$$E\left\{\left[v'_p(k)\right]^2\right\} = \frac{\Delta_s \xi_{\min}^2}{4k\delta^2}\left(\frac{1}{\lambda_p}\right) \tag{4.312}$$

Thus, (4.311) can be rewritten as

$$E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} = \frac{N\Delta_s \xi_{\min}^2}{4K\delta^2} \tag{4.313}$$

Recalling that the misadjustment M is defined as the average excess MSE divided by the minimum MSE there results for the DSD algorithm

$$M = \frac{N\Delta_s \xi_{\min}}{4K\delta^2} \tag{4.314}$$

The foregoing result is more usefully expressed in terms of time constants of the learning process and the perturbation of the gradient estimation process as developed next.

Each measurement to determine a gradient component uses $2K$ samples of data. Each adaptive weight iteration involves N gradient component measurements and therefore requires a total of $2KN$ data samples. From Section 4.2.3 it may be recalled that the MSE learning curve has a pth mode time constant given by

$$\tau_{p_{\text{MSE}}} = \frac{1}{4\Delta_s \lambda_p} = \frac{\tau_p}{2} \tag{4.315}$$

in time units of the number of iterations. It is useful to define a new time constant $T_{p_{\text{MSE}}}$ whose basic time unit is the data sample and whose value is expressed in terms of the number of data samples. It follows that for the DSD algorithm

$$T_{p_{\text{MSE}}} \triangleq 2KN\tau_{p_{\text{MSE}}} \tag{4.316}$$

The time constant $T_{p_{\text{MSE}}}$ relates to real time units (seconds) once the sampling rate is known.

By using the perturbation formula (4.282) to substitute for $\xi_{\min}$ in (4.314), the misadjustment for the DSD algorithm is rewritten as

$$M = \frac{N\Delta_s\lambda_{\text{av}}}{4KP} \tag{4.317}$$

The time constant defined by (4.316) is rewritten using (4.315) as

$$T_{p_{\text{MSE}}} = \frac{NK}{2\Delta_s\lambda_p} \tag{4.318}$$

from which one can conclude that

$$\lambda_p = \frac{NK}{2\Delta_s}\left(\frac{1}{T_{p_{\text{MSE}}}}\right) \tag{4.319}$$

so that

$$\lambda_{\text{av}} = \frac{NK}{2\Delta_s}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} \tag{4.320}$$

Combining (4.26) and (4.320) then yields the misadjustment as

$$M = \frac{N^2}{8P}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} \tag{4.321}$$

Equation (4.321) shows that for the DSD algorithm, misadjustment is proportional to the square of the number of weights and inversely proportional to the perturbation. In addition, the misadjustment is also inversely proportional to the speed of adaptation (fast adaptation results in high misadjustment). Since the DSD algorithm is based on steepest descent, it suffers from the disparate eigenvalue problem discussed in Section 4.2.2.

It is appropriate here to compare the misadjustment for the DSD algorithm given by (4.321) with the misadjustment for the LMS algorithm given by (4.83). With a specified level of misadjustment for the LMS algorithm, the adaptive time constants increase linearly with the number of weights rather than with the square of the number of weights as is the case with the DSD algorithm. Furthermore, with the LMS algorithm there is no perturbation. As a result, in typical circumstances much faster adaptation is possible with the LMS algorithm than with the DSD algorithm.

M is defined as a normalized performance penalty that results from noise in the weight vector. In an actual adaptive system employing the DSD algorithm, the weight vector is not only stochastically perturbed due to the presence of noise but in addition is deterministically perturbed so the gradient can be measured. As a consequence of the deterministic perturbation, another performance penalty accrues as measured by the perturbation P, which is also a normalized ratio of excess MSE. The total excess MSE is therefore the sum of the "stochastic" and "deterministic" perturbation components. The total misadjustment can be expressed as

$$M_{\text{tot}} = M + P \tag{4.322}$$

Adding the previous two components then yields

$$M_{\text{tot}} = \frac{N^2}{8P}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} + P \tag{4.323}$$

Since P is a design parameter given by (4.282), it can be selected by choosing the deterministic perturbation size δ. It is desirable to minimize the total misadjustment M_{tot} by appropriately selecting P. The result of such optimization is to make the two right-hand terms of (4.323) equal so that

$$P_{\text{opt}} = \frac{1}{2} M_{\text{tot}} \tag{4.324}$$

The minimum total misadjustment then becomes

$$(M_{\text{tot}})_{\min} = \frac{N^2}{4P_{\text{opt}}} \left(\frac{1}{T_{\text{MSE}}} \right)_{\text{av}} = \left[\frac{N^2}{2} \left(\frac{1}{T_{\text{MSE}}} \right)_{\text{av}} \right]^{1/2} \tag{4.325}$$

Unlike the LMS algorithm, the DSD algorithm is sensitive to any correlation that exists between successive samples of the error signal $e(k)$, since such correlation has the effect of making the effective statistical sample size less than the actual number of error samples in computing the estimated gradient vector. Because of such reduced effective sample size, the actual misadjustment experienced is greater than that predicted by (4.325), which was derived using the assumption of statistical independence between successive error samples.

4.7 | THE ACCELERATED GRADIENT APPROACH (AG)

Algorithms based on the steepest descent method exhibit an undesirable degree of sensitivity of the convergence speed to the eigenvalue spread in the input signal covariance matrix. The conjugate gradient method [33–39] is faster than the steepest descent approach. This reluctance to apply accelerated gradient methods to adaptive array problems is due principally to the following reasons:

1. There is increased hardware complexity associated with implementing the algorithm.
2. The enhanced convergence speed realized is not as fast as what can be obtained with other methods (to be discussed in later chapters).
3. The various accelerated gradient methods are all susceptible (although with different degrees) to signals that are noise corrupted.
4. There is a required increase in computation and memory space over steepest descent methods.

Despite these objections, it is worthwhile to apply one of the accelerated gradient methods to determine what improvement in convergence speed reasonably might be expected by recourse to such techniques. Powell's method [35] is tolerant of noise-induced errors, although other methods theoretically may yield faster convergence speeds. Powell's method assumes that if a performance measure $\Re(\mathbf{w})$ is quadratic in the independent variables, then any line that passes through the minimum point of the quadratic performance surface intersects the family of constant performance contours at equal angles. This property is illustrated in Figure 4-25 for two dimensions where it is seen that the line AC connecting point A with the minimum point C intersects the constant performance contours at equal angles. As a consequence of the equal angle property, the line

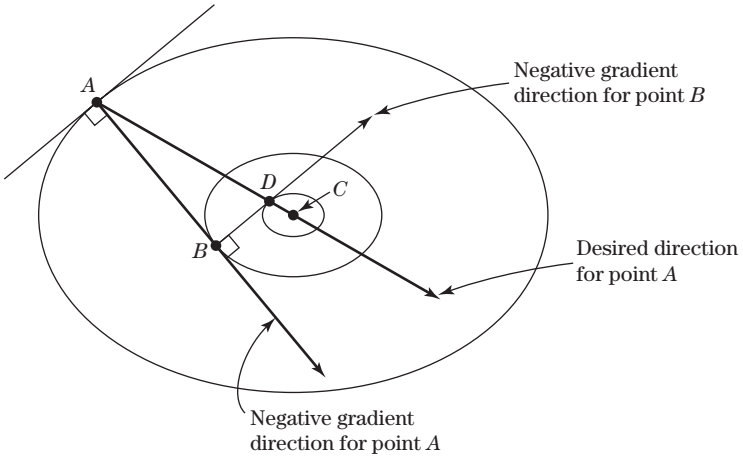


FIGURE 4-25 ■ Two-dimensional diagram showing directional relationships for the Powell descent method.

joining the point A with the point D in Figure 4-25 passes through the point C where the derivative of the performance measure $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line AD is zero.

Given an initial estimate $\mathbf{w}_0$ at point A, first find the gradient direction that is normal to the tangent of the constant performance measure contour. Proceed along the line defined by the negative gradient direction to the point B where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line is zero. The point B may in fact be any arbitrary point on the line that is a finite distance from A; however, by choosing it in the manner described the convergence of the method is assured.

Having found point B, the negative gradient direction that is parallel to the original tangent at $\mathfrak{P}(\mathbf{w}_0)$. Traveling in this new normal direction, we find a point where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line is zero (point D in Figure 4-25). The line passing through the points A and D also passes through the point C. The desired point C is the point where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line AD is zero.

The generalization of the foregoing procedure to an N-dimensional space can be obtained by recognizing that the directional relationships (which depend on the equal angle property) given in Figure 4-25 are valid only in a two-dimensional plane. The first step (moving from point A to point B) is accomplished by moving in the negative gradient direction in the N-dimensional space. Having found point B, we can construct $(N-1)$ planes between the original negative gradient direction and $(N-1)$ additional mutually orthogonal vectors, thereby defining points C, D, E, . . . , until $(N-1)$ additional points have been defined. The last three points in the N-dimensional space obtained in the foregoing manner may now be treated in the same fashion as points A, B, and D of Figure 4-25 by drawing a connecting line between the last point obtained and the point defined two steps earlier. Traveling along the connecting line one may then define a new point C of Figure 4-25. This new point may then be considered as point D in Figure 4-25, and a new connecting line may be drawn between the new point and the point obtained three steps earlier.

The steps corresponding to one complete Powell descent cycle for five dimensions are illustrated in Figure 4-26. The first step from A to B merely involves traveling in the negative gradient direction $\mathbf{v}_1$ with a step size α_1 chosen to satisfy the condition

$$\frac{d}{d\alpha_1}\{\mathfrak{P}[\mathbf{w}(0)+\alpha_1\mathbf{v}_1]\}=0 \tag{4.326}$$

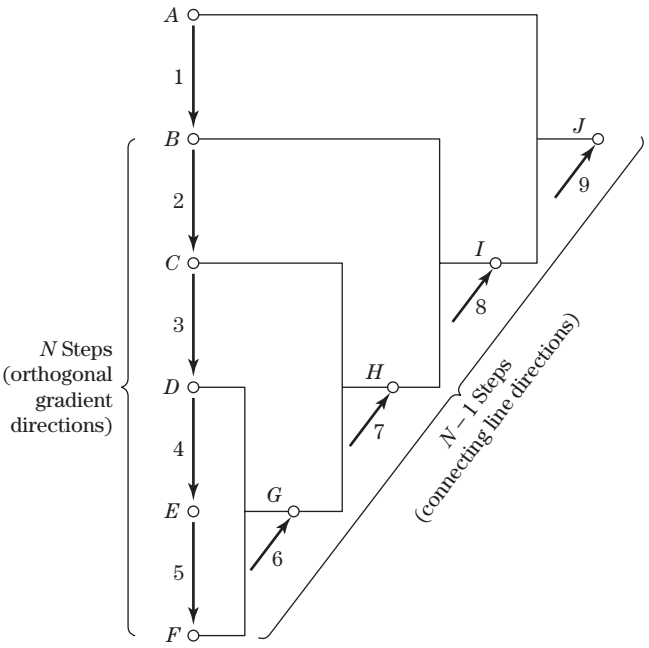


FIGURE 4-26 ■ Illustration of Powell descent method steps required in five dimensions for one complete cycle.

so that

$$\mathbf{w}(1) = \mathbf{w}(0) + \alpha_1 \mathbf{v}_1 \tag{4.327}$$

Having determined point B, we determine point C by traveling in the negative gradient direction $\mathbf{v}_2$ (a direction that is also orthogonal to $\mathbf{v}_1$) from point B with step size α_2 selected to satisfy

$$\frac{d}{d\alpha_2}\{\Re[\mathbf{w}(1) + \alpha_2 \mathbf{v}_2]\} = 0 \tag{4.328}$$

Point D is determined from point C by using the aforementioned procedure, and this process continues until a total of five points (B through F in Figure 4-26) are defined. A descent direction $\mathbf{v}_6$ is defined by drawing a connecting line between points D and F (analogous to points A and D of Figure 4-25) and traveling along this line with step size α_6 selected to satisfy

$$\frac{d}{d\alpha_6}\{\Re[\mathbf{w}(5) + \alpha_6 \mathbf{v}_6]\} = 0 \tag{4.329}$$

thereby arriving at point G. A new descent direction $\mathbf{v}_7$ is defined by drawing a connecting line between points C and G and traveling along this line with step size α_7 selected to satisfy

$$\frac{d}{d\alpha_7}\{\Re[\mathbf{w}(6) + \alpha_7 \mathbf{v}_7]\} = 0 \tag{4.330}$$

thereby arriving at point H. This process continues until the solution point J for the cycle is found from the descent direction $\mathbf{v}_9$ (defined by the connecting line between points A and I) and the step size α_9. In general, one Powell descent cycle in N-dimensional space

therefore requires $N + N - 1 = 2N - 1$ steps. On completion of one descent cycle, the entire cycle are repeated using the last weight vector obtained as the initial weight vector for the new cycle.

4.7.1 Algorithm Based on the Powell Accelerated Gradient Cycle

Each step in a Powell descent cycle involves traveling from a weight vector $\mathbf{w}$ along a direction $\mathbf{v}$ with step size α such that

$$\frac{d}{d\alpha}[\Re(\mathbf{w} + \alpha\mathbf{v})] = 0 \tag{4.331}$$

For complex weights the MSE performance measure is given by

$$\xi(\mathbf{w}) = E\{d^2\} + \mathbf{w}^\dagger \mathbf{r}_{xd} + \mathbf{r}_{xd}^\dagger \mathbf{w} + \mathbf{w}^\dagger \mathbf{R}_{xx} \mathbf{w} \tag{4.332}$$

The gradient of $\xi(\mathbf{w} + \alpha\mathbf{v})$ with respect to α is then given by

$$_\alpha[\xi(\mathbf{w} + \alpha\mathbf{v})] = \mathbf{v}^\dagger \mathbf{r}_{xd} + \mathbf{r}_{xd}^\dagger \mathbf{v} + \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{w} + \mathbf{w}^\dagger \mathbf{R}_{xx} \mathbf{v} + 2\alpha \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v} \tag{4.333}$$

and this gradient is equal to zero when the step size is

$$\alpha = -\frac{\mathbf{v}^\dagger \mathbf{r}_{xd} + \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{w}}{\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}} \tag{4.334}$$

Since $\mathbf{r}_{xd}$ and $\mathbf{R}_{xx}$ are unknown, some estimate of the numerator is employed to obtain an appropriate step size estimate. Noting that $\mathbf{r}_{xd} + \mathbf{R}_{xx}\mathbf{w}$ is one-half the gradient of $\xi(\mathbf{w})$, it follows that the numerator of (4.334) are approximated by $\mathbf{v}^\dagger \text{Av}\{e(k)\mathbf{x}(k)\}$. Note that the quantity $\mathbf{v}^\dagger \mathbf{x}$ is regarded as the output of a processor whose weights correspond to $\mathbf{v}$ and that $\text{Av}\{(\mathbf{v}^\dagger \mathbf{x})(\mathbf{x}^\dagger \mathbf{v})\}$ is an approximation of the quantity $\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}$, where the average Av{ } is taken over K data samples. The simultaneous generation of the estimates $\hat{}_w$ and $\text{Av}\{\mathbf{v}^\dagger \mathbf{x}\mathbf{x}^\dagger \mathbf{v}\}$ requires parallel processors: one processor with weight values equal to $\mathbf{w}(k)$ and another processor with weight values equal to $\mathbf{v}(k)$. Having described the procedure for determining the appropriate step size along a direction $\mathbf{v}$, we may now consider the steps required to implement an entire Powell descent cycle.

The steps required to generate one complete Powell descent cycle are as follows.

Step 1 Starting with the initial weight setting $\mathbf{w}(0)$, estimate the negative gradient direction $\mathbf{v}(0)$ using K data samples then travel in this direction with the appropriate step size to obtain $\mathbf{w}(1)$. The step size determination requires an additional K data samples to obtain by way of (4.334).

Steps 2 $\rightarrow N$ Estimate the negative gradient direction at $\mathbf{w}(k)$ using K data samples. If the gradient estimates and the preceding step size were error free, the current gradient is automatically orthogonal to the previous gradient directions. Since the gradient estimate is not error free, determine the new direction of travel $\mathbf{v}(k)$ by requiring it to be orthogonal to all previous directions $\mathbf{v}(0), \mathbf{v}(1), \ldots, \mathbf{v}(k-1)$ by employing the Schmidt orthogonalization process so that

$$\mathbf{v}(k) = \hat{}(k) - \sum_{i=0}^{k-1} \frac{[\mathbf{v}^\dagger(i) \hat{}(k)]}{[\mathbf{v}^\dagger(i)\mathbf{v}(i)]} \cdot \mathbf{v}(i) \tag{4.335}$$

Travel in the direction $-\mathbf{v}(k)$ using the appropriate step size (which requires an additional K data samples to obtain) to arrive at $\mathbf{w}(k+1)$.

Steps $N+1 \rightarrow 2N$ **1** Determine the new direction of travel at $\mathbf{w}(k)$ by forming

$$\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}[2(N-1)-k] \tag{4.336}$$

Travel in the direction $-\mathbf{v}(k)$ from $w[2(N-1)-k]$ using the appropriate step size to arrive at $\mathbf{w}(k+1)$. These steps require only K data samples since now the direction of travel does not require that a gradient estimate be obtained.

4.8 GRADIENT ALGORITHM WITH CONSTRAINTS

The early applications of sidelobe cancellation to radar antennas neglected the effects of signals in the main beam on the adapted response. Such neglect was amply justified because adaptive processors would not respond to low level reflected target signals, and the small number of degrees of freedom then available to the adaptive processor limited the response to large targets or extended clutter. Current adaptive arrays with large numbers of degrees of freedom are explicitly constrained to avoid degradation of the main beam.

Adaptive arrays having large numbers of degrees of freedom and fast response times operating with high-energy, long-duration waveforms may have reflected signal returns that are large enough to elicit undesirable responses from the adaptive processor. Such undesirable responses may produce signal cancellation and signal waveform distortion. Furthermore, jammer power level affects the array response in the main beam direction, thereby allowing blinking jammers to modulate the signal response and consequently degrade the performance of any subsequent coherent processing. The Frost [40] algorithm imposes constraints on the adaptive weights such that certain main beam properties are preserved. It turns out that the resulting constrained optimization system has two parts: (1) a preprocessing part called a "spatial correction filter," which compensates the signals for the misalignment between the plane wave front and the sensor array geometry; and (2) a signal processor that includes the adaptive weights and accounts for the primary function of the adaptive array.

The constrained LMS algorithm requires that the direction of arrival and a frequency band of interest be specified a priori for the appropriate constraint conditions to be imposed. Because the look direction frequency response relates to the adaptive weights, the algorithm maintains a selected frequency response in the look direction while simultaneously minimizing output noise power. If the look direction is perpendicular to the line of sensors, then identical signal components appear at the first taps [so $x_1(t) = x_2(t) = \cdots = x_N(t)$ in Figure 4-27] and propagate down the tapped delay lines following each sensor [so $x_{N+1}(t) = x_{N+2}(t) = \cdots = x_{2N}(t)$, and $x_{(J-1)N+1}(t) = x_{(J-1)N+2}(t) = \cdots = x_{NJ}(t)$]. Noise component waveforms arriving at the sensors from any direction other than the look direction will not usually produce equal voltage components at any vertical column of taps. Consequently, as far as the signal is concerned, the adaptive processor appears as an equivalent single tapped delay line in which each adaptive weight equals the sum of the weights in the corresponding vertical column of the original processor. These J summation weights in the equivalent tapped delay line are assigned a value to give the desired frequency response characteristic in the look direction, thereby giving rise to J constraint conditions. In the event that the look direction is other than that perpendicular

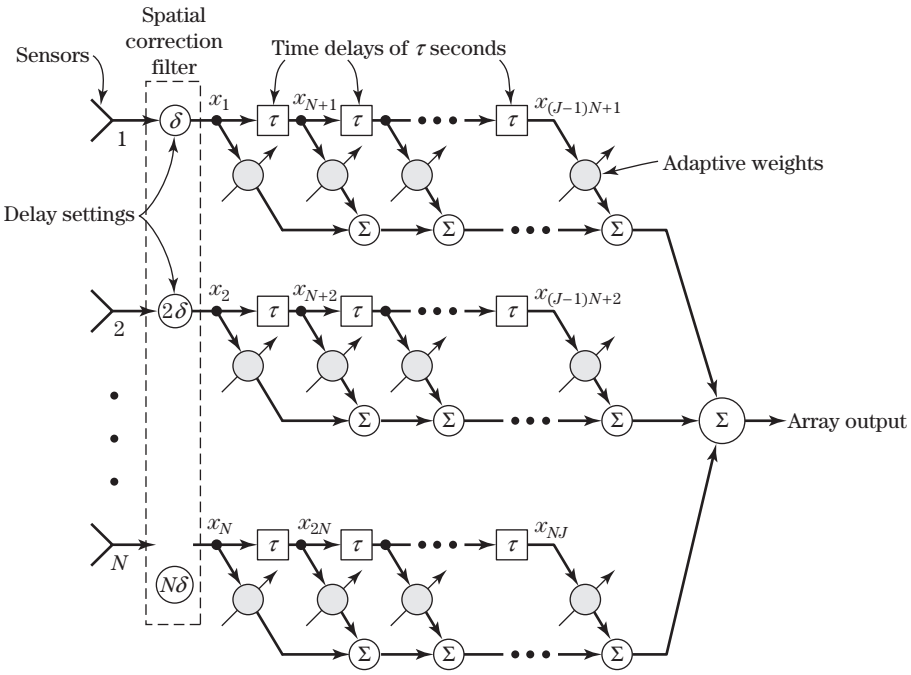


FIGURE 4-27 ■ Broadband adaptive array having N sensors and J adjustable weights per sensor.

to the line of sensors (as the previous discussion has assumed), then the time delays in the spatial correction filter are adjusted so the signal components of each channel at the output of the preprocessor are in phase.

The adaptive signal processor of Figure 4-27 has N sensors and J taps per sensor for a total of NJ adjustable weights. Using J constraints to determine the look direction frequency response leaves $NJ - J$ degrees of freedom to minimize the total array output power. Since the J constraints fix the look direction frequency response, minimizing the total output power is equivalent to minimizing the nonlook direction noise power (provided the signal voltages at the taps are uncorrelated with the corresponding noise voltages at these taps). If signal-correlated noise in the array is present, then part or all of the signal component of the array output may be cancelled. Although signal-correlated noise may not occur frequently, sources of such noise include multiple signal-propagation paths, and coherent radar or sonar "clutter."

It is desirable for proper noise cancellation that the noise voltages appearing at the adaptive processor taps be correlated among themselves (although uncorrelated with the signal voltages). Such noise sources may be generated by lightning, "jammers," noise from nearby vehicles, spatially localized incoherent clutter, and self-noise from the structure carrying the array. Noise voltages that are uncorrelated between taps (e.g., amplifier thermal noise) are partially rejected by the adaptive array either as the result of incoherent noise voltage addition at the array output or by reducing the weighting applied to any taps that may have a disproportionately large uncorrelated noise power.

4.8.1 Optimum Constrained Weight Vector Solution

The voltages appearing at each array tap in Figure 4-27 are sampled every Δ seconds (where Δ is a multiple of the delay τ between taps). The vector of tap voltages at the kth

sample is defined by

$$\mathbf{x}^T(k) \triangleq [x_1(k\Delta), x_2(k\Delta), \ldots, x_{NJ}(k\Delta)] \tag{4.337}$$

At any tap the voltages that appear may be regarded as the sums of voltages due to look direction signals s and nonlook direction noises n, so that

$$\mathbf{x}(k) = \mathbf{s}(k) + \mathbf{n}(k) \tag{4.338}$$

where the NJ-dimensional vector of look direction signals is defined by

$$\mathbf{s}(k) \triangleq \begin{bmatrix} s(k\Delta) \\ \vdots \\ s(k\Delta) \\ s(k\Delta - \tau) \\ \vdots \\ s(k\Delta - \tau) \\ \vdots \\ s[k\Delta - (J-1)\tau] \\ \vdots \\ s[k\Delta - (J-1)\tau] \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \end{matrix} \tag{4.339}$$

and the vector of nonlook direction noises is defined by

$$\mathbf{n}^T(k) \triangleq [n_1(k\Delta), n_2(k\Delta), \ldots, n_{NJ}(k\Delta)] \tag{4.340}$$

The weight vector appearing at each tap is denoted by $\mathbf{w}$, where

$$\mathbf{w}^T \triangleq [w_1, w_2, \ldots, w_{NJ}] \tag{4.341}$$

We assume that the signals and noises are zero-mean random processes with unknown second-order statistics. The covariance matrices of $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{n}$ are given by

$$E\{\mathbf{x}(k)\mathbf{x}^T(k)\} = \mathbf{R}_{xx} \tag{4.342}$$

$$E\{\mathbf{s}(k)\mathbf{s}^T(k)\} = \mathbf{R}_{ss} \tag{4.343}$$

$$E\{\mathbf{n}(k)\mathbf{n}^T(k)\} = \mathbf{R}_{nn} \tag{4.344}$$

Since the vector of look direction signals is assumed uncorrelated with the vector of nonlook direction noises

$$E\{\mathbf{n}(k)\mathbf{s}^T(k)\} = \mathbf{0} \tag{4.345}$$

Assume that the noise environment is such that $\mathbf{R}_{xx}$ and $\mathbf{R}_{nn}$ are positive definite and symmetric.

The adaptive array output (which forms the signal estimate) at the kth sample is given by

$$y(k) = \mathbf{w}^T\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{w} \tag{4.346}$$

From (4.346), it follows that the expected array output power is

$$E\{y^2(k)\} = E\{\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}\} = \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{4.347}$$

Suppose that the weights in the jth vertical column of taps sums to a selected number f_j. This constraint may be expressed by the relation

$$\mathfrak{c}_j^T\mathbf{w} = f_j, \quad j = 1, 2, \ldots, J \tag{4.348}$$

where the NJ-dimensional vector $\mathfrak{c}_j$ is given by

$$\mathfrak{c}_j = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \vdots \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \left.\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}\right\} j\text{th column of } N \text{ elements} \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \vdots \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \end{bmatrix} \tag{4.349}$$

Now consider the requirement of constraining the entire weight vector to satisfy all J equations given by (4.348). Define a $J \times NJ$ constraint matrix $\mathbf{C}$ having $\mathfrak{c}_j$ as elements.

$$\mathbf{C} \triangleq [\mathfrak{c}_1 \cdots \mathfrak{c}_j \cdots \mathfrak{c}_J] \tag{4.350}$$

Furthermore, define $\mathbf{f}$ as the J-dimensional vector of summed weight values for each of the j vertical columns that yield the desired frequency response characteristic in the look direction as

$$\mathbf{f} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_J \end{bmatrix} \tag{4.351}$$

It immediately follows by inspection that the full set of constraints (4.348) can be written in matrix form as

$$\mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{4.352}$$

Now that the look direction frequency response is fixed by the constraint equation (4.352), minimizing the nonlook direction noise power is equivalent to minimizing the total output power given by (4.347). The constrained optimization problem reduces to

$$\underset{\mathbf{w}}{\text{Minimize}}\ \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{4.353}$$

$$\text{subject to } \mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{4.354}$$

Lagrange multipliers are used to find $\mathbf{w}_{\text{opt}}$ that satisfy (4.353) and (4.354) [41]. Adjoining the constraint equation (4.354) to the cost function (4.353) by a J-dimensional vector $\boldsymbol{\lambda}$, whose elements are undetermined Lagrange multipliers (and including a factor of $\frac{1}{2}$ to simplify the arithmetic), then yields

$$\underset{\mathbf{w}}{\text{Minimize}}\ \Re(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{xx}}\mathbf{w} + \lambda^{\mathrm{T}}[\mathbf{C}^{\mathrm{T}}\mathbf{w} - \mathbf{f}] \tag{4.355}$$

The gradient of (4.355) with respect to $\mathbf{w}$ is given by

$$_{\mathbf{w}}\Re(\mathbf{w}) = \mathbf{R}_{xx}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} \tag{4.356}$$

A necessary condition for (4.355) to be minimized is that the gradient be equal to zero so that

$$\mathbf{R}_{xx}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} = \mathbf{0} \tag{4.357}$$

Therefore, the optimal weight vector is given by

$$\mathbf{w}_{\text{opt}} = -\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda} \tag{4.358}$$

where the vector $\boldsymbol{\lambda}$ remains to be determined. The vector of Lagrange multipliers may now be evaluated from the constraint equation

$$\mathbf{C}^T\mathbf{w}_{\text{opt}} = \mathbf{f} = \mathbf{C}^T\left[-\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda}\right] \tag{4.359}$$

It then follows that the vector $\boldsymbol{\lambda}$ is given by

$$\boldsymbol{\lambda} = -\left[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right]^{-1}\mathbf{f} \tag{4.360}$$

where the existence of $[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}]^{-1}$ is guaranteed by the fact that $\mathbf{R}_{xx}$ is positive definite and $\mathbf{C}$ has full rank. Combining (4.358) and (4.360) then yields the optimum constrained weight vector

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{C}\left[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right]^{-1}\mathbf{f} \tag{4.361}$$

If we substitute $\mathbf{w}_{\text{opt}}$ into (4.346), it follows that the constrained least squares estimate of the look direction signal provided by the array is

$$y_{\text{opt}}(k) = \mathbf{w}_{\text{opt}}^T\mathbf{x}(k) \tag{4.362}$$

If the vector of summed weight values $\mathbf{f}$ is selected so the frequency response characteristic in the look direction is all-pass and linear phase (distortionless), then the output of the constrained LMS signal processor is the maximum likelihood (ML) estimate of a stationary process in Gaussian noise (provided the angle of arrival is known) [42]. A variety of other optimal processors can also be obtained by a suitable choice of the vector $\mathbf{f}$ [43]. It is also worth noting that the solution (4.361) is sensitive to deviations of the actual signal direction from that specified by $\mathbf{C}$ and to various random errors in the array parameters [44].

4.8.2 The Adaptive Algorithm

Assume the adaptive processor must determine the correlation matrix $\mathbf{R}_{xx}$. It follows that in stationary environments during learning and in time-varying environments an estimate of the optimum adaptive processor weights must be periodically recomputed. The initial guess of an appropriate weight vector satisfies (4.354), so a good starting point is

$$\mathbf{w}(0) = \mathbf{C}[\mathbf{C}^T\mathbf{C}]^{-1}\mathbf{f} \tag{4.363}$$

where the quantity $\mathbf{C}[\mathbf{C}^T\mathbf{C}]^{-1}$ represents the pseudo-inverse of the singular matrix $\mathbf{C}^T$ [45]. For a gradient type algorithm, after the kth iteration the next weight vector is given by

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \Delta_s \quad {}_w\Re[\mathbf{w}(k)] \\ &= \mathbf{w}(k) - \Delta_s[\mathbf{R}_{xx}\mathbf{w}(k) + \mathbf{C}\lambda(k)]\end{aligned} \tag{4.364}$$

where Δ_s is the step size constant, and $\Re$ denotes the performance measure. Requiring $\mathbf{w}(k+1)$ to satisfy (4.352) then yields

$$\mathbf{f} = \mathbf{C}^T\mathbf{w}(k+1) = \mathbf{C}^T\{\mathbf{w}(k) - \Delta_s[\mathbf{R}_{xx}\mathbf{w}(k) + \mathbf{C}\lambda(k)]\} \tag{4.365}$$

Consequently, the Lagrange multipliers are given by

$$\begin{aligned}\lambda(k) = &-[\mathbf{C}^T\mathbf{C}]^{-1}\mathbf{C}^T\mathbf{R}_{xx}\mathbf{w}(k) - \frac{1}{\Delta_s}[\mathbf{C}^T\mathbf{C}]^{-1} \\ &\cdot[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)]\end{aligned} \tag{4.366}$$

Substituting (4.366) into (4.364) then gives the iterative relation

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) &- \Delta_s[I - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T]\mathbf{R}_{xx}\mathbf{w}(k) \\ &+ \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)]\end{aligned} \tag{4.367}$$

It is convenient to define the NJ-dimensional vector

$$\dagger \triangleq \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} \tag{4.368}$$

and the $NJ \times NJ$ matrix

$$\mathbf{P} \triangleq \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{4.369}$$

Then the iterative relation (4.369) may be rewritten as

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s\mathbf{R}_{xx}\mathbf{w}(k)] + \dagger \tag{4.370}$$

In the actual system the input correlation matrix is not known, and it is necessary to adopt some estimate of this matrix to insert in place of $\mathbf{R}_{xx}$ in the iterative weight adjustment equation. An approximation for $\mathbf{R}_{xx}$ at the kth iteration is merely the outer product of the tap voltage vector with itself: $\mathbf{x}(k)\mathbf{x}^T(k)$. Substituting this estimate of $\mathbf{R}_{xx}$ into (4.370) and recognizing that $y(k) = \mathbf{x}^T(k)\mathbf{w}(k)$ then yields the constrained LMS algorithm

$$\left.\begin{aligned} \mathbf{w}(0) &= \tilde{\mathbf{f}} \\ \mathbf{w}(k+1) &= \mathbf{P}[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)] + \tilde{\mathbf{f}} \end{aligned}\right\} \tag{4.371}$$

If it is merely desired to ensure that the complex response of the adaptive array system to a normalized signal input from the look direction is unity, then the spatial correction filter is dispensed with and the compensation for phase misalignment incorporated directly into the variable weight selection as suggested by Takao et al. [46]. Denote the complex response (amplitude and phase) of the array system by $Y(\theta)$, where θ is the angle measured from the normal direction to the array face. The appropriate conditions to impose on the adaptive weights are found by requiring that $\Re e\{Y(\theta)\} = 1$ and $Im\{Y(\theta)\} = 0$ when $\theta = \theta_c$, the look direction.

4.8.3 Conditions Ensuring Convergence to the Optimum Solution

The weight vector $\mathbf{w}(k)$ obtained by employing (4.371) is a random vector. Convergence of the mean value of the weight vector to the optimum is shown by considering the length of the difference vector between the mean of the actual weight vector and the optimum weight vector: convergence is assured if the length of the difference vector asymptotically approaches zero.

If we start with the weight adjustment equation

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s \mathbf{x}(k)y(k)] + \tilde{\mathbf{f}} \tag{4.372}$$

and recognize that $y(\mathbf{k}) = \mathbf{x}^T(k)\mathbf{w}(k)$, then taking the expected value of both sides of (4.372) yields

$$E[\mathbf{w}(k+1)] = \mathbf{P}\{E[\mathbf{w}(k)] - \Delta_s \mathbf{R}_{xx} E[\mathbf{w}(k)]\} + \tilde{\mathbf{f}} \tag{4.373}$$

Define the difference vector $\mathbf{v}(k+1)$ by

$$\mathbf{v}(k+1) \triangleq E\left[\mathbf{w}(k+1)\right] - \mathbf{w}_{\text{opt}} \tag{4.374}$$

Substitute (4.373) into (4.374) and use $\tilde{\mathbf{f}} = (\mathbf{I} - \mathbf{P})\mathbf{w}_{\text{opt}}$ and $\mathbf{P}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} = 0$ [which may be verified by direct substitution of (4.361) and (4.369)], then the difference vector satisfies

$$\mathbf{v}(k+1) = \mathbf{P}\mathbf{v}(k) - \Delta_s \mathbf{P}\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.375}$$

Note from (4.369) that $\mathbf{P}$ is idempotent (i.e., $\mathbf{P}^2 = \mathbf{P}$), then premultiplying (4.375) by $\mathbf{P}$ reveals that $\mathbf{P}\mathbf{v}(k+1) = \mathbf{v}(k+1)$ for all k, so (4.375) can be rewritten as

$$\begin{aligned} \mathbf{v}(k+1) &= [\mathbf{I} - {}_s\mathbf{P}\mathbf{R}_{xx}\mathbf{P}]\mathbf{v}(k) \\ &= [\mathbf{I} - {}_s\mathbf{P}\mathbf{R}_{xx}\mathbf{P}]^{(k+1)}\mathbf{v}(0) \end{aligned} \tag{4.376}$$

From (4.376) it follows that the matrix $\mathbf{P}\mathbf{R}_{xx}\mathbf{P}$ determines both the rate of convergence of the mean weight vector to the optimum solution and the steady-state variance of the weight

vector about the optimum. The matrix $\mathbf{PR}_{xx}\mathbf{P}$ has J zero eigenvalues (corresponding to the column vectors of the constraint matrix $\mathbf{C}$) and $NJ - J$ nonzero eigenvalues σ_i, $i = 1, 2, \ldots, NJ - J$ [48]. The values of the $NJ - J$ nonzero eigenvalues are bounded by the relation

$$\lambda_{\min} \leq \sigma_{\min} \leq \sigma_i \leq \sigma_{\max} \leq \lambda_{\max} \tag{4.377}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of R_{xx}, respectively, and $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest nonzero eigenvalues of $\mathbf{PR}_{xx}\mathbf{P}$, respectively.

The initial difference vector $\mathbf{v}(0) = \tilde{\mathbf{f}} - \mathbf{w}_{\text{opt}}$ can be expressed as a linear combination of the eigenvectors of $\mathbf{PR}_{xx}\mathbf{P}$ corresponding to the nonzero eigenvalues [47]. Consequently, if $\mathbf{v}(0)$ is equals an eigenvector $\mathbf{e}_i$ of $\mathbf{PR}_{xx}\mathbf{P}$ corresponding to the nonzero eigenvalue σ_i, then

$$\begin{aligned} \mathbf{v}(k+1) &= [\mathbf{I} - \Delta_s \mathbf{PR}_{xx}\mathbf{P}]^{(k+1)}\mathbf{e}_i \\ &= [1 - \Delta_s \sigma_i]^{(k+1)}\mathbf{e}_i \end{aligned} \tag{4.378}$$

From (4.378) it follows that along any eigenvector of $\mathbf{PR}_{xx}\mathbf{P}$ the mean weight vector converges to the optimum weight vector geometrically with the geometric ratio $(1 - \Delta_s \sigma_i)$. Consequently, the time required for the difference vector length to decay to $1/e$ of its initial value is given by the time constant

$$\begin{aligned} \tau_i &= \frac{\Delta t}{\ln(1 - \Delta_s \sigma_i)} \\ &\cong \frac{\Delta t}{\Delta_s \sigma_i} \quad \text{if } \Delta_s \sigma_i \ll 1 \end{aligned} \tag{4.379}$$

where Δt denotes the time interval corresponding to one iteration.

If the step size constant Δ_s is selected so that

$$0 < \Delta_s < \frac{1}{\sigma_{\max}} \tag{4.380}$$

then the length (given by the norm) of any difference vector is bounded by

$$\begin{aligned} (1 - \Delta_s \sigma_{\max})^{(k+1)} \|\mathbf{v}(0)\| &\leq \|\mathbf{v}(k+1)\| \\ &\leq (1 - \Delta_s \sigma_{\min})^{(k+1)} \|\mathbf{v}(0)\| \end{aligned} \tag{4.381}$$

It immediately follows that if the initial difference vector length is finite, then the mean weight vector converges to the optimum so that

$$\lim_{k \to \infty} \|E\{\mathbf{w}(k)\} - \mathbf{w}_{\text{opt}}\| = 0 \tag{4.382}$$

where the convergence occurs with the time constants given by (4.379).

The LMS algorithm is designed to cope with nonstationary noise environments by continually adapting the weights in the signal processor. In stationary environments, however, this adaptation results in the weight vector exhibiting an undesirable variance about the optimum solution thereby producing an additional (above the optimum) component of noise to appear at the adaptive array output.

The optimum (minimum) output power level is given by

$$E\{y_{\text{opt}}^2(k)\} = \mathbf{w}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} = \mathbf{f}^T\left(\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right)^{-1}\mathbf{f} \tag{4.383}$$

The additional noise caused by adaptively adjusting the weights can be compared with (4.383) to determine the penalty incurred by the adaptive algorithm. A direct measure of this penalty is the "misadjustment" M defined by (4.55). For a step size constant satisfying

$$0 < \Delta_s < \frac{1}{\sigma_{\max} + \frac{1}{2}\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})} \tag{4.384}$$

The steady-state misadjustment has been shown to be bounded by [48]

$$\frac{\Delta_s}{2} \cdot \frac{\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})}{1 - (\Delta_s/2)[\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P}) + 2\sigma_{\min}]} \leq M \leq \frac{\Delta_s}{2} \cdot \frac{\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})}{1 - (\Delta_s/2)[\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P}) + 2\sigma_{\max}]} \tag{4.385}$$

If Δ_s is chosen to satisfy

$$0 < \Delta_s < \frac{2}{3\text{tr}(\mathbf{R}_{xx})} \tag{4.386}$$

then it will automatically also satisfy (4.384). It is also worth noting that the upper bound in (4.383) can be easily calculated directly from observations since $\text{tr}(\mathbf{R}_{xx}) = E\{\mathbf{x}^T(k)\mathbf{x}(k)\}$, the sum of the powers of the tap voltages.

4.8.4 A Useful Geometrical Interpretation

The constrained LMS algorithm (4.372) has a simple geometrical interpretation [40] that is useful for visualizing the error correcting property that prevents the weight vector from deviating from the constraint condition. Even unavoidable computational errors due to roundoff, truncation, or quantization are prevented from accumulating by the error correcting property, which continuously corrects for any errors that may occur, whatever their source may be.

In an error-free algorithm, the successive values of the NJ-dimensional weight vector $\mathbf{w}$ all exactly satisfy the constraint equation (4.354) and therefore all lie on a constraint plane $\mathbf{\Lambda}$ defined by

$$\mathbf{\Lambda} = \{\mathbf{w} : \mathbf{C}^T\mathbf{w} = \mathbf{f}\} \tag{4.387}$$

This constraint plane [which is $(NJ - J)$-dimensional] may be indicated diagramatically as shown in Figure 4-28.

Any vectors that point in a direction normal to the constraint plane are linear combinations of the constraint matrix column vectors and therefore all have the form $\mathbf{Ca}$, where $\mathbf{a}$ is a constant vector whose components determine the linear combination. Consequently, the initial weight vector in the algorithm (4.371), $\mathbf{f} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$, points in a direction normal to the constraint plane. In addition, the initial weight vector terminates exactly on the constraint plane since $\mathbf{C}^T\mathbf{f} = \mathbf{f}$. As a result, $\mathbf{f}$ is the shortest vector that can terminate on the constraint plane, as illustrated in Figure 4-28.

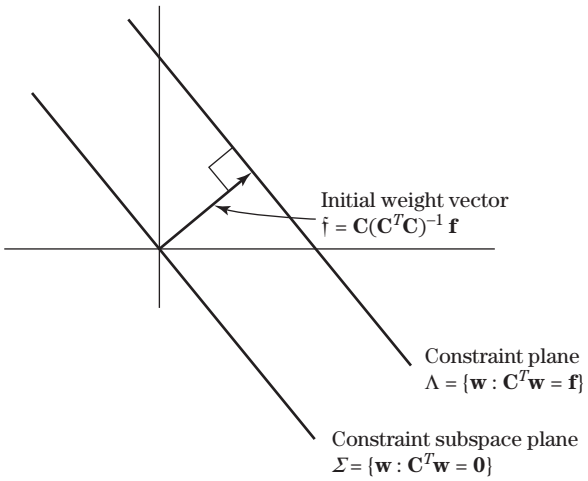


FIGURE 4-28 ■ Representation of the constraint plane, the constraint subspace plane, and initial weight vector in the **w**-space.

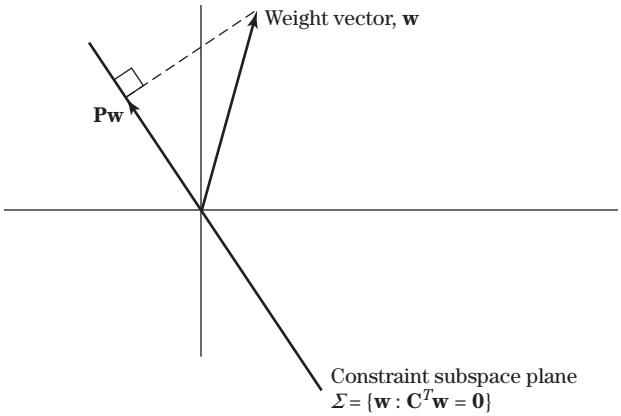


FIGURE 4-29 ■ Matrix **P** projects vectors onto the constraint subspace plane.

By setting the constraint weight vector **f** equal to zero, the homogeneous form of the constraint equation

$$\mathbf{C}^T\mathbf{w} = \mathbf{0} \tag{4.388}$$

defines a second plane [that is also $(NJ-J)$-dimensional] that passes through the coordinate space origin. This constraint subspace is depicted in Figure 4-28.

The constrained LMS algorithm (4.371) premultiplies a certain vector in the **W**-space by the matrix **P**, a projection operator. Premultiplication of any weight vector by the matrix **P** results in the elimination of any vector components perpendicular to the plane $\sum$, thereby projecting the original weight vector onto the constraint subspace plane as illustrated in Figure 4-29.

The only factor in (4.371) remaining to be discussed is the vector $y(k)\mathbf{x}(k)$, which is an estimate of the unconstrained gradient of the performance measure. Recall from (4.355) that the unconstrained performance measure is $\frac{1}{2}\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}$ and from (4.356) that the unconstrained gradient is given by $\mathbf{R}_{xx}\mathbf{w}$. Since the covariance matrix $\mathbf{R}_{xx}$ is unknown a priori, the estimate provided by $y(k)\mathbf{x}(k)$ is used in the algorithm.

The constrained optimization problem posed by (4.353) and (4.354) is illustrated diagramatically in the **w**-space as shown in Figure 4-30. The algorithm must succeed

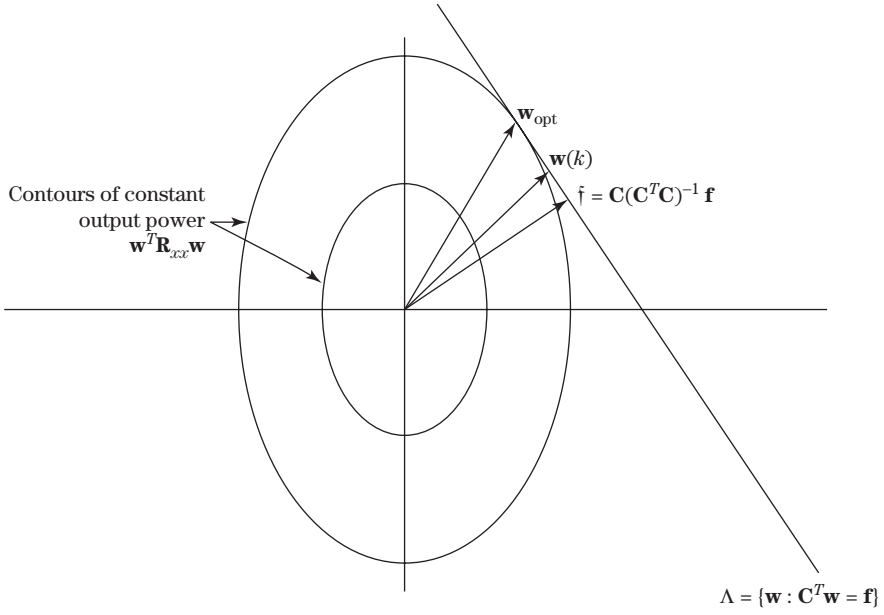


FIGURE 4-30 ■ Diagrammatic representation of the constrained optimization problem showing contours of constant output power, the constraint plane $\mathbf{\Lambda}$, the initial weight vector $\hat{\mathbf{f}}$, and the optimum constrained weight vector $\mathbf{w}_{\text{opt}}$ that minimizes the output power.

in moving from the initial weight vector $\hat{\mathbf{f}}$ to the optimum weight vector $\mathbf{w}_{\text{opt}}$ along the constraint plane $\mathbf{\Lambda}$. The operation of the constrained LMS algorithm (4.371) in solving the previously given constrained optimization problem is considered.

In Figure 4-31 the current value of the weight vector, $\mathbf{w}(k)$, is to be modified by taking the unconstrained negative gradient estimate $-y(k)\mathbf{x}(k)$, scaling it by Δ_s, and adding the result to $\mathbf{w}(k)$. In general, the resulting vector lies somewhere off the constraint plane. Premultiplying the vector $[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)]$ by the matrix $\mathbf{P}$, the projection onto the constraint subspace plane is obtained. Finally, adding $\hat{\mathbf{f}}$ to constraint subspace plane projection produces a new weight vector that lies on the constraint plane. This new

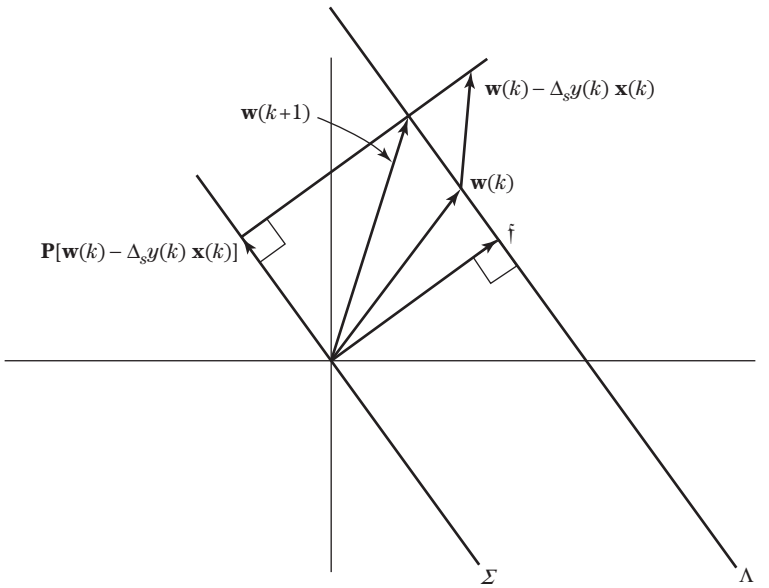


FIGURE 4-31 ■ Operation of the constrained LMS algorithm: $\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)] + \hat{\mathbf{f}}$.

weight vector $\mathbf{w}(k+1)$ satisfies the constraint to within the numerical accuracy of the computations. This error-correcting feature of the constrained LMS algorithm prevents any computational errors from accumulating.

The convergence properties of the constrained LMS algorithm are closely related to those for the unconstrained LMS algorithm and have been previously discussed. Likewise, the same procedures that increased convergence speed for the LMS algorithm also work for the constrained LMS algorithm.

4.9 SIMULATION RESULTS

The fundamental misadjustment versus speed of adaptation trade-off is less favorable for the DSD algorithm than for the LMS algorithm [13]. Thus, it remains to determine the improvement in this fundamental trade-off that can be realized using the Powell accelerated gradient (PAG) algorithm compared with the LMS algorithm where eigenvalue spread in the input signal covariance matrix is present.

Figure 4-32 depicts a four-element Y array having $d = 0.787\lambda$ element spacing with the desired signal located at 0° and three distinct narrowband Gaussian jamming signals located at 15°, 90°, and 165°. The received signal covariance matrix is therefore given by

$$\frac{1}{n}\mathbf{R}_{xx} = \frac{s}{n}(\mathbf{u}\mathbf{u}^\dagger) + \sum_{i=1}^{3}\frac{J_i}{n}\left(\mathbf{v}_i\mathbf{v}_i^\dagger\right) + \mathbf{I} \tag{4.389}$$

where n denotes the thermal noise power (taken to be unity), s/n denotes the signal-to-thermal noise ratio, and J_i/n denotes the jammer-to-thermal noise ratios for each of the three jammers ($i = 1, 2, 3$). The elements of the signal steering vector $\mathbf{u}$ and the jammer steering vectors $\mathbf{v}_i$ are easily defined from the array geometry and the signal arrival angles. The desired signal is a biphase modulated signal having a phase angle of either 0° or 180° with equal probability at each sample.

Two signal conditions were simulated corresponding to two values of eigenvalue spread in the received signal covariance matrix. The first condition represents a respectable

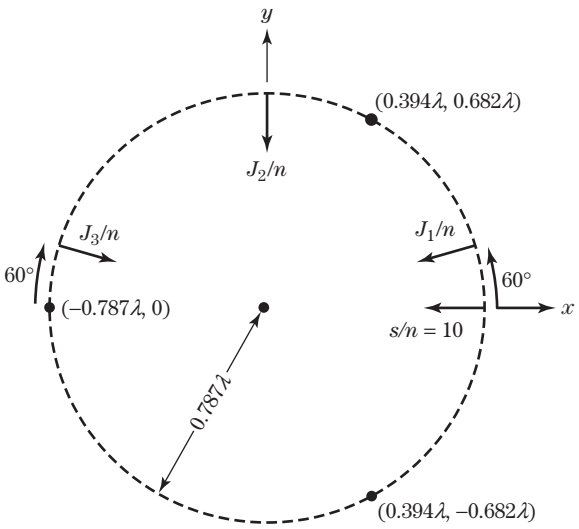


FIGURE 4-32 ■ Four-element Y-array geometry with signal and jammer locations for selected example.

eigenvalue spread of $\lambda_{max}/\lambda_{min} = 2440$, whereas the second condition represents a more extreme eigenvalue spread of $\lambda_{max}/\lambda_{min} = 16,700$. Choosing the jammer-to-thermal noise ratios to be $J_1/n = 500$, $J_2/n = 40$, and $J_3/n = 200$ together with $s/n = 10$ yields the corresponding eigenvalues $\lambda_1 = 2.44 \times 10^3$, $\lambda_2 = 4.94 \times 10^2$, $\lambda_3 = 25.62$, and $\lambda_4 = 1.0$ for which the optimum output SNR is $SNR_{opt} = 15.0$ (11.7 dB). Likewise, choosing the jammer-to-thermal noise ratios to be $J_1/n = 4000$, $J_2/n = 40$, and $J_3/n = 400$ along with $s/n = 10$ yields the eigenvalues $\lambda_1 = 1.67 \times 10^4$, $\lambda_2 = 10^3$, $\lambda_3 = 29$, and $\lambda_4 = 1.0$ for which the optimum output SNR is also $SNR_{opt} = 15.0$. In all cases the initial weight vector setting was taken to be $\mathbf{w}^T(0) = [0.1, 0, 0, 0]$. Figures 4-33 and 4-34 show the convergence results for the LMS and PAG algorithms, respectively, plotted as output SNR in decibels versus number of iterations for an eigenvalue spread of 2,440 (here output SNR means output signal-to-jammer plus thermal noise ratio). The expected value of the gradient and $\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}$ required by the PAG algorithm was taken over $K = 9$ data samples, and one iteration of the PAG algorithm occurred every nine data samples,

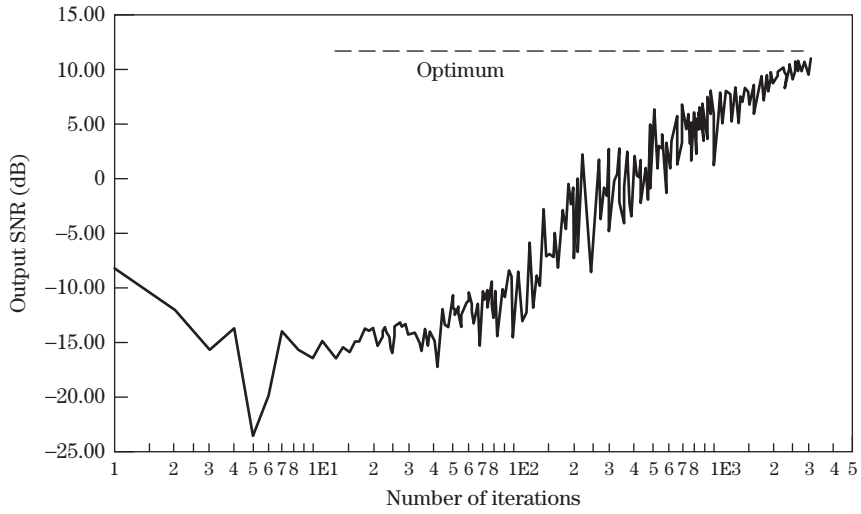


FIGURE 4-33 ■ Output SNR versus number of iterations for LMS algorithm with eigenvalue spread = 2440 and $\alpha_L = 0.1$.

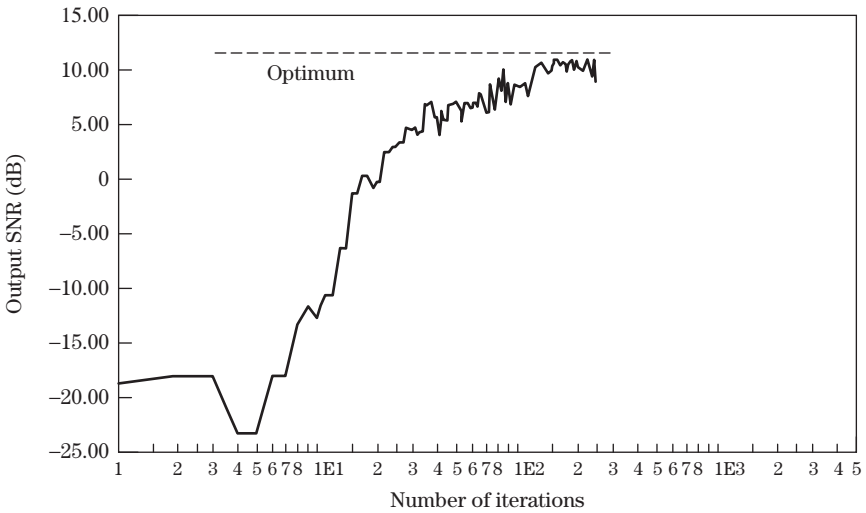


FIGURE 4-34 ■ Output SNR versus number of iterations for PAG algorithm with eigenvalue spread = 2,440 and $K = 9$.

even though a weight update does not occur on some iterations. The loop gain of the LMS loop was selected in accordance with (4.49), which requires that $\Delta_s \text{tr}(\mathbf{R}_{xx}) < 1$ for stability. Letting $\Delta_s \text{tr}(\mathbf{R}_{xx}) = \alpha_L$ and choosing $\alpha_L = 0.1$ therefore ensures stability while giving reasonably fast convergence with an acceptable degree of misadjustment error. As a consequence of the manner in which an iteration was defined for the PAG algorithm, the time scale for Figure 4-34 is nine times greater than the time scale for Figure 4-33. In Figure 4-34 the PAG algorithm is within 3 dB of the optimum after approximately 80 iterations (720 data samples), whereas in Figure 4-33 the LMS algorithm requires approximately 1500 data samples to reach the same point. Furthermore, it may be seen that the steady-state misadjustment for the two algorithms in these examples is very comparable so the PAG algorithm converges twice as fast as the LMS algorithm for a given level of misadjustment in this example.

Figures 4-35 and 4-36 show the convergence of the LMS and PAG algorithms for the same algorithm parameters as in Figures 4-33 and 4-34 but with the eigenvalue spread = 16,700. In Figure 4-36 the PAG algorithm is within 3 dB of the optimum after approx-

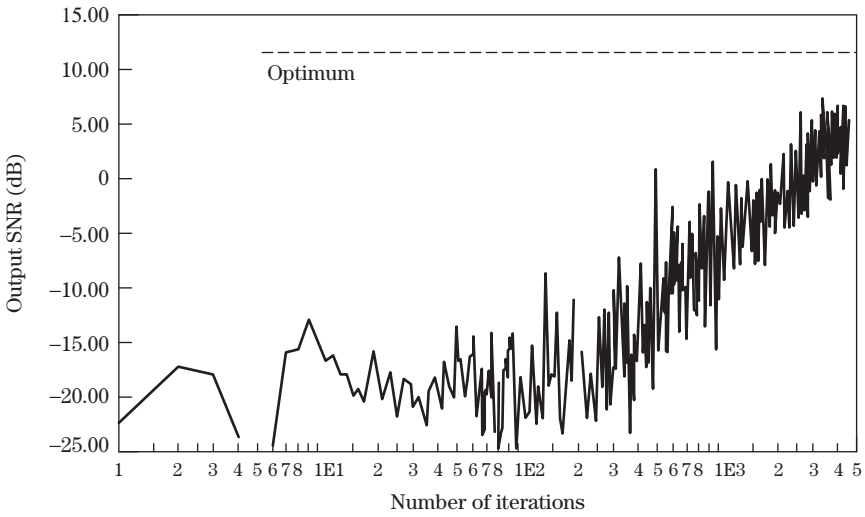


FIGURE 4-35 ■ Output SNR versus number of iterations for LMS algorithm with eigenvalue spread = 16,700 and $\alpha_L = 0.1$.

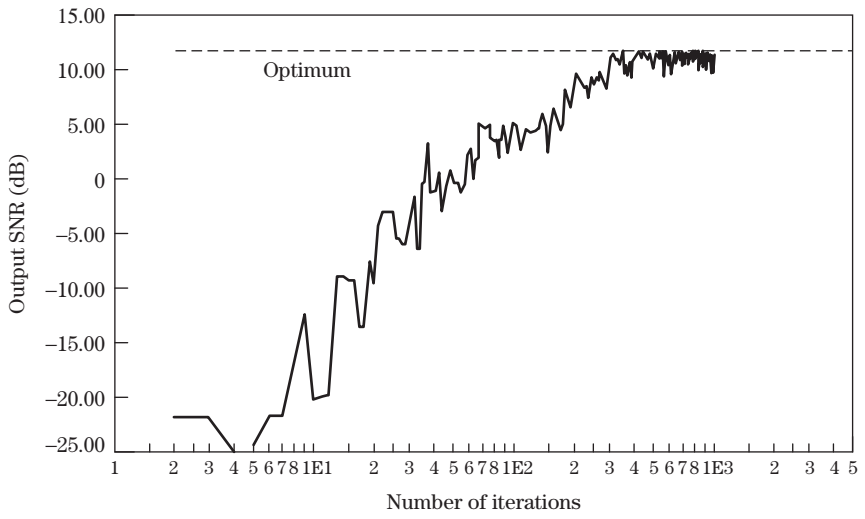


FIGURE 4-36 ■ Output SNR versus number of iterations for PAG algorithm with eigenvalue spread = 16,700 and $K = 9$.

imately 200 iterations (1,800 data samples), whereas the LMS algorithm in Figure 4-35 does not reach the same point even after 4,500 data samples. The degree of convergence speed improvement that is attainable therefore increases as the degree of eigenvalue spread increases.

A word of caution is needed concerning the expected convergence when using the PAG algorithm. The simulation results given here were compiled for an array having only four elements; as the number of array elements increases, the number of consecutive steps in orthogonal gradient directions also increases, thereby yielding significant direction errors in the later steps (since estimation errors accumulate over the consecutive step directions). Accordingly, for a given level of misadjustment the learning curve time constant does not increase linearly with N (as with LMS adaptation), but rather increases more rapidly. In fact, when $N > 10$, the PAG algorithm actually converges more slowly than the LMS algorithm.

4.10 | PHASE-ONLY ADAPTIVE NULLING USING STEEPEST DESCENT

A phased array may or may not have variable amplitude weights but always has phase shifters for beam steering and calibration. Since the phase shifters are already in place for beam steering purposes, they can also serve as adaptive elements to adjust the antenna pattern. The authors in [49] presented a beam-space algorithm for low sidelobe arrays that relies on small phase shifts. When the direction of arrival for all the interfering sources is known, then cancellation beams in the directions of the sources are subtracted from the original pattern. Adaptation consists of matching the peak of the cancellation beam with the culprit sidelobe and subtracting [50].

The steepest descent algorithm can also find phase settings that minimize the output power [51]. Multiplying the change in output power due to a change in phase at each element by a step size, μ, produces a gradient vector for the phase weights having components given by

$$\delta_n(\kappa + 1) = \delta_n(\kappa) + \mu \frac{P(\kappa) - P(\kappa - 1)}{\Delta(\kappa)} \tag{4.390}$$

where

$P(\kappa) =$ array output power at time step κ
$\delta_n(\kappa) =$ phase shift at element n
$\Delta(\kappa) =$ small phase increment

$$\mu = \frac{\Delta^2}{\sqrt{\sum_{n=1}^{N} [P(\kappa) - P(-1)]^2}}$$

This algorithm worked well for phase-only simultaneous nulling of the sum and difference patterns of an 80-element linear array of H plane sectoral horns [50]. A diagram of the array appears in Figure 2-25 of Chapter 2. The sum channel has a 30 dB low sidelobe Taylor taper and the difference channel has a 30 dB low sidelobe Bayliss taper. These channels share eight-bit beam steering phase shifters. Experiments used a CW signal incident on a sidelobe but no signal incident on the main beam. The cost function takes

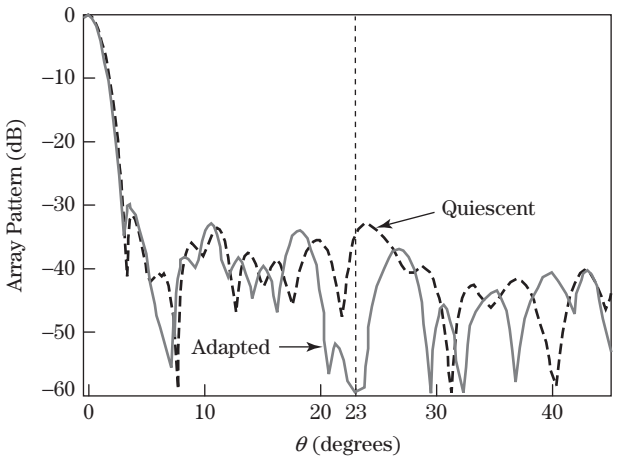


FIGURE 4-37 ■ Adapted sum pattern for simultaneous phase-only nulling in the sum and difference channels.

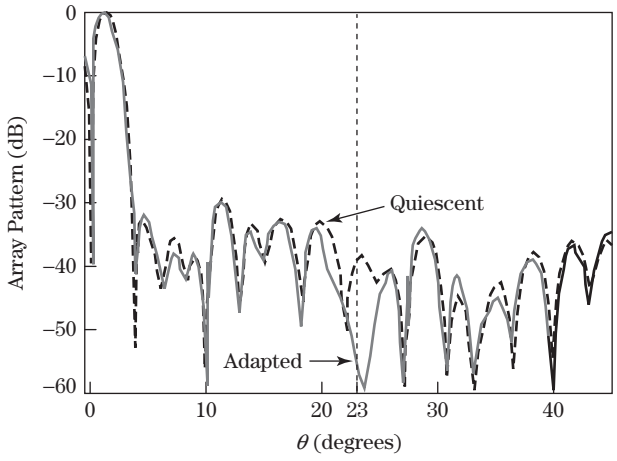


FIGURE 4-38 ■ Adapted difference pattern for simultaneous phase-only nulling in the sum and difference channels.

into consideration both the sum and difference channel output powers; otherwise, a null will not appear in both patterns. Minimizing the output power of both channels when an interfering signal appears at $\theta = 23^\circ$ results in the patterns shown in Figure 4-37 and Figure 4-38. The desired nulls are place with relatively small deviations from the quiescent patterns.

4.11 SUMMARY AND CONCLUSIONS

The LMS algorithm applies the method of steepest descent to the MSE performance measure to obtain a simple implementation that is particularly well suited to continuous signal communication systems. The LMS algorithm requires a reference signal that is compared with the array output to form an error signal. This technique is useful for adaptive arrays that are expected to distinguish between desired and undesired signals on the basis of differences in modulation characteristics. The heart of an LMS loop is the correlator (multiplier), which forms the product $e(t)x_i(t)$ that is required to obtain the estimated gradient. For an N-element array, N correlators are therefore required to implement the LMS algorithm to control each array element.

The Howells–Applebaum adaptive processor is generally employed in situations where the desired signal is usually absent (in contrast to the LMS algorithm, which requires that the desired signal be present) and makes use of a beam steering vector instead of a reference signal. We find that the Howells–Applebaum processor behavior is characterized by a quiescent mode (when the desired signal is absent) and an adapted mode (when the desired signal is present). The transient behavior of the algorithm is most easily described in terms of eigenvector beams, which can be analyzed by introduction of a transformation to a normal coordinate system that diagonalizes the signal covariance matrix. The processor exhibits the characteristic of sensitivity to eigenvalue spread, so that strong interference sources are canceled rapidly while weak interference sources are suppressed slowly. The dynamic range requirements of the circuitry used to implement the Howells–Applebaum processor can be reduced by introduction into the control loop of a hard limiter that modifies the effective signal covariance matrix but does not affect the eigenvalue spread.

Different methods for constraining the maximum SNR algorithm to maintain a favorable desired signal response are discussed:

1. The use of pilot signals
2. Preadaptation spatial filters
3. Control loop spatial filters
4. Discrimination constraint for known desired signal power level

The close relationship that exists between the Howells–Applebaum maximum SNR processor and the LMS algorithm makes the similar transient behavior characteristics of these two algorithms hardly surprising. The susceptibility of the algorithm performance to eigenvalue spread in the signal covariance matrix leads to a consideration of ways this susceptibility can be reduced. One way of reducing this susceptibility and maintaining fast transient response for all eigenvalue conditions is to employ a direct matrix inversion (DMI) algorithm, which is introduced in the next chapter.

For some practical applications it may be undesirable to require N correlators as the LMS algorithm does. In such cases, the alternative presented by the DSD algorithm, which requires only direct performance index measurements (error power measurements in the case of the MSE criterion), may be attractive. The DSD algorithm does not have as favorable a convergence speed versus misadjustment trade-off as the LMS algorithm, and both the DSD and LMS algorithms exhibit the same degree of convergence speed sensitivity to eigenvalue spread in the input signal covariance matrix.

One way of reducing the convergence speed sensitivity to eigenvalue spread is to employ an algorithm based on an accelerated gradient approach, provided the number of degrees of freedom of the array processor is not too high. An algorithm based on the Powell descent cycle was presented illustrating the improvement in the speed of convergence that can be realized. Accelerated gradient approaches have certain implementation drawbacks, however, and other methods (discussed in later chapters) may be preferred to obtain the desired reduction in convergence speed sensitivity to eigenvalue spread. In applications involving high energy, long-duration waveforms, it is often desirable to constrain the main beam of the array so that undesirable signal waveform distortion will not occur.

A nice summary of the LMS and Howells–Applebaum algorithms is given by [52]

$$\mathbf{w}_{n+1} = \begin{cases} (\mathbf{I} - \gamma \mathbf{R}_n)\mathbf{w}_n + \gamma \mu \mathbf{b}^* & \text{Howells–Applebaum} \\ (\mathbf{I} - \gamma \mathbf{R}_n)\mathbf{w}_n + \gamma \mathbf{r}_{xd} & \text{LMS} \end{cases} \tag{4.391}$$

4.12 PROBLEMS

1. ***Misadjustment-Speed of Adaptation Trade-off for the LMS and DSD Algorithms [13]*** For the LMS algorithm the total misadjustment in the steady state is given by (4.83), whereas the total (minimum) misadjustment for the DSD algorithm is given by (4.317).

(a) Assuming all eigenvalues are equal so that $(T_{p_{\text{MSE}}})_{\text{av}} = T_{\text{MSE}}$ and that $M = 10\%$ for the LMS algorithm, plot T_{MSE} versus N for $N = 2, 4, 8, \ldots, 512$.

(b) Assuming all eigenvalues are equal so that $(T_{p_{\text{MSE}}})_{\text{av}} = T_{\text{MSE}}$ and that $(M_{\text{tot}})_{\min} = 10\%$ for the DSD algorithm, plot T_{MSE} versus N for $N = 2, 4, 8, \ldots, 512$ and compare this result with the previous diagram obtained in part (a).

2. ***Reference Signal Generation for LMS Adaptation Using Polarization as a Desired Signal Discriminant [53]*** LMS adaptation requires a reference signal to be generated having properties sufficiently correlated either to the desired signal or the undesired signal to permit the adaptive system to preserve the desired signal in its output. Usually, the desired signal waveform properties (e.g., frequency, duration, type of modulation, signal format) are used to generate the reference signal, but if the signal and the interference can be distinguished by polarization, then polarization may be employed as a useful discriminant for reference signal generation.

Let s denote a linearly polarized desired signal having the known polarization angle θ, and let n denote a linearly polarized interference signal having the polarization angle α (where it is known only that $\alpha \neq \theta$). Assume that the desired signal and interference impinge on two linearly polarized antennas (A and B) as shown in Figure 4-39 where the antennas differ in orientation by the angle β. The two signals v_a and v_b may then be expressed as

$$v_a = s\cos\theta + n\cos\alpha$$
$$v_b = s\cos(\beta - \theta) + n\cos(\beta - \alpha)$$

(a) Show that by introducing the weight w_1 as illustrated in Figure 4-39, then the signal $v_b' = v_b - w_1 v_a$ can be made to be signal free (have zero desired signal content) by setting

$$w_1 = \frac{\cos(\beta - \theta)}{\cos\theta}$$

so that

$$v_b' = n\frac{\sin\beta}{\cos\theta}\sin(\alpha - \theta) = nf(\alpha, \beta, \theta)$$

(b) From the results of part (a), show that

$$v_0 = v_a - w_2 v_b' = s\cos\theta + n[\cos\alpha - w_2 f(\alpha, \beta, \theta)]$$

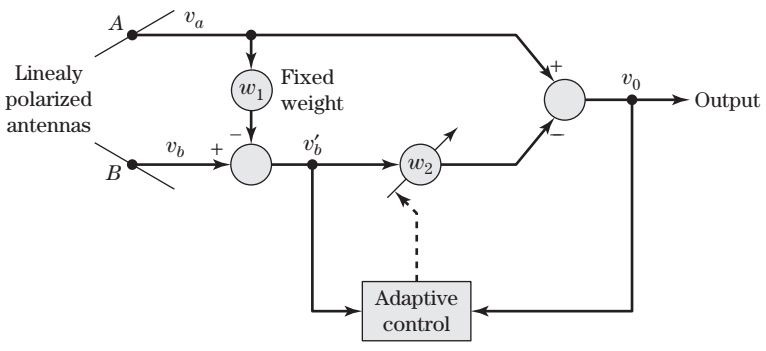


FIGURE 4-39 ■ Adaptive array configuration for interference rejection on the basis of polarization using LMS adaptation.

Since the output signal v_0 contains both desired signal and interference components, correlating it with the signal-free voltage v_b' yields a measure of the interference remaining in the output signal, and the adaptive weight w_2 can then be adjusted to reduce the interference content in the output.

(c) The error in the output signal v_0 is the interference signal component that is still present after $w_2 b_b'$ is subtracted from v_a. Assume that the interference and the desired signal are uncorrelated; then

$$E\left\{v_0^2\right\} = E\{s^2\}\cos^2\theta + E\{n^2\}[\cos\alpha - w_2 f(\alpha, \beta, \theta)]^2$$

If the rate of change of w_2 is proportional to $\partial E\left\{v_0^2\right\}/\partial w_2$, show that the final value of the weight occurs when $\partial E\left\{v_0^2\right\}/\partial w_2 = 0$ so that

$$w_2 = \frac{\cos\alpha}{f(\alpha, \beta, \theta)}$$

(d) With w_2 set to the final value determined in part (c), show that the steady-state system output is given by

$$v_0 = s\cos\theta$$

thereby showing that the system output is free of interference under steady-state conditions. The previous result assumes that (1) knowledge of θ and the setting of w_1 are error free; (2) the circuitry is noiseless; And (3) the number of input signals equals the number of antennas available. These ideal conditions are not met in practice, and [53] analyzes the system behavior under nonideal operating conditions.

3. ***Relative Sensitivity of the Constrained Look-Direction Response Processor to Perturbed Wavefronts [44]*** The solution to the problem of minimizing the expected output power of an array $\eta = E\{\mathbf{w}^\dagger \mathbf{x}\mathbf{x}^\dagger \mathbf{w}\}$ subject to $\mathbf{x}_0^\dagger \mathbf{w} = f$ (or equivalently, $\eta = f^2$) is given by (4.366). Since the look direction response is constrained by $\mathbf{x}_0^\dagger \mathbf{w} = f$ where $\mathbf{x}_0$ denotes a plane wave signal arriving from the angle θ_0, the rationale behind this constraint is to regard the processor as a filter that will pass plane waves from the angle θ_0 but attenuate plane waves from all other directions.

Let a perturbed plane wave be represented by $\mathbf{x}$, having components

$$\mathbf{x}_k = A_{k_0}(1 + \alpha_k)\exp[j(\phi_{k_0} + \xi_k)]$$

where α_k represents amplitude deviations, and ξ_k represents phase deviations from the nominal plane wave signal $\mathbf{x}_0$. Assume that α_k, ξ_k are all uncorrelated zero-mean Gaussian random variables with variances $\sigma_\alpha^2, \sigma_\xi^2$ at each sensor of the array.

(a) Using $\eta = \mathbf{w}^\dagger E\{\mathbf{x}\mathbf{x}^\dagger\}\mathbf{w}$ and the fact that

$$E\{x_i x_j^*\} = x_{i_0} x_{j_0}^* \exp\left(-\sigma_\xi^2\right) \quad \text{for } i \neq j$$

and

$$E\left\{x_i x_j^*\right\} = |x_{i_0}|^2\left(1 + \sigma_\alpha^2\right) \quad \text{for } i = j$$

show that

$$\eta = \exp\left(-\sigma_\xi^2\right)\mathbf{w}^\dagger \mathbf{x}_0 \mathbf{x}_0^\dagger \mathbf{w} + \left[1 - \exp\left(-\sigma_\xi^2\right) + \sigma_\alpha^2\right]\mathbf{w}^\dagger \mathbf{w}$$

or $\eta \cong f^2 + \left(\sigma_\xi^2 + \sigma_\alpha^2\right)\mathbf{w}^\dagger\mathbf{w}$ for small values of $\sigma_\xi^2, \sigma_\alpha^2$ assuming that $|x_{i_0}|^2 = 1$ (which is the case for a planar wave).

(b) The result in part (a) can be rewritten as

$$\eta = f^2\left[1 + \mathfrak{s}\left(\sigma_\xi^2 + \sigma_\alpha^2\right)\right]$$

where

$$\mathfrak{s} \triangleq \frac{\mathbf{w}^\dagger\mathbf{w}}{f^2}$$

Consequently, the ratio $\mathfrak{s}$ can be regarded as the relative sensitivity of the processor to the perturbations whose variances are $\sigma_\xi^2, \sigma_\alpha^2$. Using the weights given by (4.169), show that

$$\mathfrak{s} = \frac{\mathbf{x}_0^\dagger\mathbf{R}_{xx}^{-2}\mathbf{x}_0}{\left(\mathbf{x}_0^\dagger R_{xx}^{-1}\mathbf{x}_0\right)^2}$$

The previous relative sensitivity can become large if the eigenvalues of $\mathbf{R}_{xx}$ have a large spread, but if the eigenvalues of $\mathbf{R}_{xx}$ have a small spread then $\mathfrak{s}$ cannot become large.

4. ***MSLC Relationships [29]*** Show that (4.219) results from (4.215) by the following:

(a) Substitute (4.217) and (4.218) into (4.219).

(b) Let $\mathbf{K} = \mathbf{I} + g\mathbf{R}_{nn}$.

(c) Apply the matrix inversion lemma [(D.10) of Appendix D] to the resulting expression.

5. ***MSLC Relationships [29]*** Show that (4.234) follows from the steady-state relationship given by (4.232).

6. ***Control Loop Spatial Filter Relationships [29]*** Apply the matrix inversion identity

$$[\mathbf{Q} + \mathbf{e}\mathbf{f}^T]^{-1}\mathbf{e} = \frac{\mathbf{Q}^{-1}\mathbf{e}}{1 + \mathbf{f}^T\mathbf{Q}^{-1}\mathbf{e}}$$

where $\mathbf{Q}$ is a nonsingular $N \times N$ matrix and $\mathbf{e}$ and $\mathbf{f}$ are $\mathbf{N} \times 1$ vectors to (4.247), and show that (4.248) results.

7. ***Control Loop Spatial Filter Relationships [29]*** By substituting the relationships expressed by (4.248) and (4.249) into (4.247), show that the steady-state weight vector relationship given by (4.250) results.

8. ***Control Loop Spatial Filter Relationships [29]*** To show that (4.267) can be developed from (4.265), define the ratio

$$SN' \triangleq \frac{\mathbf{w}'^\dagger\mathbf{s}'\mathbf{s}'^\dagger\mathbf{w}'}{\mathbf{w}'^\dagger\mathbf{R}'_{xx}\mathbf{w}'}$$

where

$$\mathbf{w}' = \begin{bmatrix} -\mathbf{w} \\ \hline 1 \end{bmatrix}, \quad \mathbf{s}' = \begin{bmatrix} \mathbf{s} \\ \hline s_0 \end{bmatrix}$$

and

$$\mathbf{R}'_{xx} = \left[\begin{array}{c|c} \mathbf{R}_{xx} & \mathbf{r}_{xx_0} \\ \hline \mathbf{r}_{xx_0}^\dagger & P_0 \end{array}\right]$$

(a) Show that $|s_0 - \mathbf{w}^\dagger\mathbf{s}|^2 = \mathbf{w}'^\dagger\mathbf{s}'\mathbf{s}'^\dagger\mathbf{w}'$.

(b) Show that $P_e \triangleq \mathbf{w}'^{\dagger}\mathbf{R}'_{xx}\mathbf{w}' = \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} - \mathbf{w}^{\dagger}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}w + P_0$.

(c) Since $\mathbf{w} = [\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{r}_{xx_0}$ from (4.263) show that

$$\mathbf{w} = \mathbf{w}_{\text{opt}} + \Delta\mathbf{w}$$

where $\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0}$ (the Wiener solution), and $\Delta\mathbf{w} = -a\mathbf{R}^{-1}_{xx}\mathbf{w}$.

(d) Substitute $\mathbf{w} = \mathbf{w}_{\text{opt}} + \Delta\mathbf{w}$ into P_e from part (b) and show that

$$P_e = P_{e_0} + \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\Delta\mathbf{w}$$

where

$$\begin{aligned} P_{e_0} &= P_0 - \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}\mathbf{w}_{\text{opt}} + \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} \\ &= P_0 - \mathbf{r}^{\dagger}_{xx_0}\mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0} \\ &= P_0 - \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} \end{aligned}$$

Hint: Note that

$$\begin{aligned} &\Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} + \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\Delta\mathbf{w} \\ &\quad - \Delta\mathbf{w}^{\dagger}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}\Delta\mathbf{w} = 0 \end{aligned}$$

because

$$\begin{aligned} \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} &= \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0} \\ &= \Delta\mathbf{w}^{\dagger}\mathbf{r}_{xx_0} \end{aligned}$$

(e) Show that

$$\Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\Delta\mathbf{w} = a^2\sum_{i=1}^{N}\frac{|(\mathbf{Q}\mathbf{r}_{xx_0})_i|^2}{\lambda_i(\lambda_i + a)^2}$$

by using $\Delta\mathbf{w} = -a\mathbf{R}^{-1}_{xx}\mathbf{w}$.

Hint: Note that

$$\mathbf{r}^{\dagger}_{xx_0}\mathbf{Q}^{-1}\mathbf{Q}[\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{R}^{-1}_{xx}\mathbf{Q}\mathbf{Q}^{-1}\cdot[\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{r}_{xx_0}$$

is composed entirely of diagonalized matrices since

$$\mathbf{Q}\mathbf{R}^{-1}_{xx}\mathbf{Q}^{-1} = \mathbf{\Lambda} \text{ and } \mathbf{Q}\mathbf{Q}^{-1} = \mathbf{I}$$

9. ***Performance Degradation Due to Errors in the Assumed Direction of Signal Incidence [54]***
The received signal vector can be represented by

$$\mathbf{x}(t) = \mathbf{s}(t) + \sum_{i=2}^{m}\mathbf{g}_i(t) + \mathbf{n}(t)$$

where

$$\begin{aligned} \mathbf{s}(t) &= \text{desired signal vector} = s(t)\mathbf{v}_1 \\ \mathbf{g}_i(t) &= \text{directional noise sources} = g_i(t)\mathbf{v}_i \end{aligned}$$

and

$$\mathbf{n}(t) = \text{thermal noise vector comprised of narrowband Gaussian noise components independent from one sensor element to the next.}$$

The vectors $\mathbf{v}_i, i = 1, \ldots, m$ can be regarded as steering vectors where

$$\mathbf{v}_i^T = [\exp(-j\omega_c\tau_{i1}), \exp(-j\omega_c\tau_{i2}), \ldots, \exp(-j\omega_c\tau_{iN})]$$

and τ_{ik} represents the delay of the ith directional signal at the kth sensor relative to the geometric center of the array; ω_c is the carrier signal frequency.

The optimum weight vector should satisfy

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}$$

where $\mathbf{R}_{xx}$ is the received signal covariance matrix, and $\mathbf{r}_{xd}$ is the cross-correlation vector between the desired signal s and the received signal vector $\mathbf{x}$. Direction of arrival information is contained in $\mathbf{r}_{xd}$, and if the direction of incidence is assumed known, then $\mathbf{r}_{xd}$ can be specified and only $\mathbf{R}_{xx}^{-1}$ need be determined to find $\mathbf{w}_{\text{opt}}$. If the assumed direction of incidence is in error, however, then $\mathbf{w} = \mathbf{R}_{xx}^{-1}\tilde{\mathbf{r}}_{xd}$ where $\tilde{\mathbf{r}}_{xd}$ represents the cross-correlation vector computed using the errored signal steering vector $\tilde{\mathbf{v}}_1$.

(a) For the foregoing signal model, the optimum weight vector can be written as $\mathbf{w}_{\text{opt}} = [S\mathbf{v}_1\mathbf{v}_1^\dagger + \mathbf{R}_{nn}]^{-1} \cdot (S\mathbf{v}_1)$, where $\mathbf{R}_{nn}$ denotes the noise covariance matrix, and S denotes the desired signal power per sensor. If $\mathbf{v}_1$ is in error, then $\tilde{\mathbf{r}}_{xd} = (S\tilde{\mathbf{v}}_1)$. Show that the resulting weight vector computed using $\tilde{\mathbf{r}}_{xd}$ is given by

$$\mathbf{w} = \frac{S}{1 + S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1}\left[\left(1 + S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1\right)\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1 - S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1\mathbf{R}_{nn}^{-1}\mathbf{v}_1\right]$$

(b) Using the result obtained in part (a), show that the output signal-to-noise power ratio (when only the desired signal and thermal noise are present) from the array is given by

$$\left(\frac{S}{N}\right)_{\text{out}} = \frac{\mathbf{w}^\dagger E\{\mathbf{s}\mathbf{s}^\dagger\}\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}} = \frac{S\mathbf{w}^\dagger\left(\mathbf{v}_1\mathbf{v}_1^\dagger\right)\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}}$$

$$= \frac{S|\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2}{\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1 - 2S|\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2 + \mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1[S^2\{(\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1)^* \times (\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1) - |\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2\} + 2S\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1]}$$

(c) Use the fact that $\mathbf{R}_{nn} = \sigma^2\mathbf{I}$ and the result of part (b) to show that

$$\left(\frac{S}{N}\right)_{\text{out}} = \frac{S\left(\frac{N}{\sigma^2}\right)\frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}}{\left(1 + \frac{NS}{\sigma^2}\right)^2\left[1 - \frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}\right] + 2\left(\frac{NS}{\sigma^2}\right)\left[1 - \frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}\right]}$$

(d) Show for a uniform linear array that

$$\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right| = \frac{|\sin[(N\pi d/\lambda_c)\sin\tilde{\theta}]|^2}{|\sin[(\pi d/\lambda_c)\sin\tilde{\theta}]|^2}$$

where d represents the separation between sensors, and $\tilde{\theta}$ represents the angular uncertainty from boresight.

4.13 REFERENCES

[1] B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, edited by R. E. Kalman and N. DeClaris, New York, Holt, Rinehart and Winston, 1970, pp. 563–587.

[2] F. R. Ragazzini and G. F. Franklin, *Sampled-Data Control Systems*, New York, McGraw-Hill, 1958.

[3] E. I. Jury, *Sampled-Data Control Systems*, New York, Wiley, 1958.

[4] J. T. Tou, *Digital and Sampled-Data Control Systems*, New York, McGraw-Hill, 1959.

[5] B. C. Kuo, *Discrete-Data Control Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1970.

[6] B. Widrow, "Adaptive Filters I: Fundamentals," Stanford Electronics Laboratories, Stanford, CA, Rept. SEL-66-126 (Tech. Rept. 6764-6), December 1966.

[7] J. S. Koford and G. F. Groner, "The Use of an Adaptive Threshold Element to Design a Linear Optimal Pattern Classifier," *IEEE Trans. Inf. Theory*, Vol. IT-12, January 1966, pp. 42– 50.

[8] K. Steinbuch and B. Widrow, "A Critical Comparison of Two Kinds of Adaptive Classification Networks," *IEEE Trans. Electron. Comput. (Short Notes)*, Vol. EC-14, October 1965, pp. 737–740.

[9] F. W. Smith, "Design of Quasi-Optimal Minimum-Time Controllers," *IEEE Trans. Autom. Control*, Vol. AC-11, January 1966, pp. 71–77.

[10] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, No. 12, December 1967, pp. 2143–2159.

[11] L. J. Griffiths, "Signal Extraction Using Real-Time Adaptation of a Linear Multichannel Filter," Ph.D. Disseration, Stanford University, December 1967.

[12] K. D. Senne and L. L. Horowitz, "New Results on Convergence of the Discrete-Time LMS Algorithm Applied to Narrowband Adaptive Arrays," *Proceedings of the 17th IEEE Conference on Decision and Control,* San Diego, CA, January 10–12, 1979, pp. 1166–1167.

[13] B. Widrow and J. M. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 615–638.

[14] R. T. Compton, Jr., "An Adaptive Array in a Spread-Spectrum Communication System," *Proc. IEEE*, Vol. 66, No. 3, March 1978, pp. 289–298.

[15] D. M. DiCarlo and R. T. Compton, Jr., "Reference Loop Phase Shift in Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-14, No. 4, July 1978, pp. 599–607.

[16] P. W. Howells, "Intermediate Frequency Side-Lobe Canceller," U.S. Patent 3202990, August 24, 1965.

[17] S. P. Applebaum, "Adaptive Arrays," Syracuse University Research Corp., Report SPL TR 66-1, August 1966.

[18] L. E. Brennan and I. S. Reed, "Theory of Adaptive Radar," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-9, No. 2, March 1973, pp. 237–252.

[19] L. E. Brennan and I. S. Reed, "Adaptive Space-Time Processing in Airborne Radars," Technology Service Corporation, Report TSC-PD-061-2, Santa Monica, CA, February 24, 1971.

[20] A. L. McGuffin, "Adaptive Antenna Compatibility with Radar Signal Processing," in *Proceedings of the Array Antenna Conference,* February 1972, Naval Electronics Laboratory, San Diego, CA.

[21] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 2, March 1971, pp. 254–262.

[22] W. F. Gabriel, "Adaptive Arrays—An Introduction," *Proc. IEEE*, Vol. 64, No. 2, February 1976, pp. 239–272.

[23] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 2, March 1971, pp. 254–262.

[24] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. of IEEE,* Vol. 57, No. 10, October 1969, pp. 1696–1704.

[25] R. T. Compton, Jr., *Adaptive Antennnas Concepts and Performance,* Englewood Cliffs, NJ, Prentice-Hall, Inc., 1988.

[26] M. W. Ganz, "Rapid Convergence by Cascading Applebaum Adaptive Arrays," *IEEE Trans. AES,* Vol. 30, No. 2, April 1994, pp. 298–306.

[27] A. J. Berni, "Weight Jitter Phenomena in Adaptive Control Loops," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-14, No. 4, July 1977, pp. 355–361.

[28] L. E. Brennan and I. S. Reed, "Effect of Envelope Limiting in Adaptive Array Control Loops," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 4, July 1971, pp. 698–700.

[29] S. P. Applebaum and D. J. Chapman, "Adaptive Arrays with Main Beam Constraints," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 650– 662.

[30] R. T. Compton, Jr., "Adaptive Arrays: On Power Equalization with Proportional Control," Ohio State University, Columbus, Quarterly Report 3234-1, Contract N0019-71-C-0219, December 1971.

[31] C. L. Zahm, "Application of Adaptive Arrays to Suppress Strong Jammers in the Presence of Weak Signals," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-9, No. 2, March 1973, pp. 260–271.

[32] B. W. Lindgren, *Statistical Theory*, New York, Macmillan, 1960, Ch. 5.

[33] M. R. Hestenes and E. Stiefel, "Method of Conjugate Gradients for Solving Linear Systems," *J. Res. Natl. Bur. Stand.*, Vol. 29, 1952, p. 409.

[34] J. D. Powell, "An Iterative Method for Finding Stationary Values of a Function of Several Variables," *Comput. J. (Br.)*, Vol. 5, No. 2, July 1962, pp. 147–151.

[35] R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," *Comput. J. (Br.)*, Vol. 6, No. 2, July 1963, pp. 163–168.

[36] R. Fletcher and C. M. Reeves, "Functional Minimization by Conjugate Gradients," *Comput. J. (Br.)*, Vol. 7, No. 2, July 1964, pp. 149–154.

[37] D. G. Luenberger, *Optimization by Vector Space Methods*, New York, Wiley, 1969, Ch. 10.

[38] L. Hasdorff, *Gradient Optimization and Nonlinear Control*, New York, Wiley, 1976.

[39] L. S. Lasdon, S. K. Mitter, and A. D. Waren, "The Method of Conjugate Gradients for Optimal Control Problems," *IEEE Trans. Autom. Control*, Vol. AC-12, No. 2, April 1967, pp. 132–138.

[40] O. L. Frost, III, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proc. IEEE*, Vol. 60, No. 8, August 1972, pp. 926–935.

[41] A. E. Bryson, Jr. and Y. C. Ho, *Applied Optimal Control*, Waltham, MA, Blaisdell, 1969, Ch. 1.

[42] E. J. Kelly and M. J. Levin, "Signal Parameter Estimation for Seismometer Arrays," Massachusetts Institute of Technology, Lincoln Laboratories Technical Report 339, January 1964.

[43] A. H. Nuttall and D. W. Hyde, "A Unified Approach to Optimum and Suboptimum Processing for Arrays," U.S. Navy Underwater Sound Laboratory, New London, CT, USL Report 992, April 1969.

[44] R. N. McDonald, "Degrading Performance of Nonlinear Array Processors in the Presence of Data Modeling Errors," *J. Acoust. Soc. Am.*, Vol. 51, No. 4, April 1972, pp. 1186–1193.

[45] D. Q. Mayne, "On the Calculation of Pseudoinverses," *IEEE Trans. Autom. Control*, Vol. AC-14, No. 2, April 1969, pp. 204–205.

[46] K. Takao, M. Fujita, and T. Nishi, "An Adaptive Antenna Array Under Directional Constraint," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 662–669.

[47] O. L. Frost, III, "Adaptive Least Squares Optimization Subject to Linear Equality Constraints," Stanford Electronics Laboratories, Stanford, CA, DOC. SEL-70-053, Technical Report TR6796-2, August 1970.

[48] J. L. Moschner, "Adaptive Filtering with Clipped Input Data," Stanford Electronics Laboratories, Stanford, CA, Dec. SEL-70-053, Technical Report TR6796-1, June 1970.

[49] C. Baird, and G. Rassweiler, "Adaptive Sidelobe Nulling Using Digitally Controlled Phase-Shifters," *IEEE Transactions on Antennas and Propagation,* Vol. 24, No. 5, 1976, pp. 638–649.

[50] H. Steyskal, "Simple Method for Pattern Nulling by Phase Perturbation," *IEEE Transactions on Antennas and Propagation,* Vol. 31, No. 1, 1983, pp. 163–166.

[51] R. L. Haupt, "Adaptive Nulling in Monopulse Antennas," *IEEE Transactions on Antennas and Propagation,* Vol. 36, No. 2, 1988, pp. 202–208.

[52] D. S. De Lorenzo, J. Gautier, J. Rife, P. Enge, and D. Akos, "Adaptive Array Processing for GPS Interference Rejection," *Proc. ION GNSS 2005*, pp. 618–627.

[53] H. S. Lu, "Polarization Separation by an Adaptive Filter," *IEEE Trans. Aerosp. Electron. Sys.*, Vol. AES-9, No. 6, November 1973, pp. 954–956.

[54] C. L. Zahm, "Effects of Errors in the Direction of Incidence on the Performance of an Adaptive Array," *Proc. IEEE*, Vol. 60, No. 8, August 1972, pp. 1008–1009.

PART II

Adaptive Algorithms

Chapter 3 showed that a variety of popular performance measures led to closely related optimum weight vector solutions. Consequently, the choice of a specific performance measure is not as important as the adaptive algorithm that adjusts the array weights, since the algorithm influences the speed of the array transient response and the hardware complexity. Part 2 presents a survey of algorithms and discusses the important performance characteristics of each one. In some cases, algorithms are tailored to particular signal conditions, whereas in other cases they handle a variety of signal environments. These algorithm characteristics provide the designer with a means for picking the algorithm based on convergence speed, operational signal environment, and hardware complexity.

CHAPTER

4

Gradient-Based Algorithms

Chapter Outline

Gradient algorithms are popular, because they are simple, easy to understand, and solve a large class of problems. The performance, $\Re(\mathbf{w})$, and adaptive weights determine the nature of the performance surface. When $\Re(\mathbf{w})$ is a quadratic function of the weight settings, then it is a bowl-shaped surface with a minimum at the "bottom of the bowl." In this case, local optimization methods, such as gradient methods, can find the bottom. In the event that the performance surface is irregular, having several relative optima or saddle points, then the transient response of the gradient-based minimum-seeking algorithms get stuck in a local minimum. The gradient-based algorithms considered in this chapter are as follows:

1. Least mean square (LMS)
2. Howells–Applebaum loop
3. Differential steepest descent (DSD)
4. Accelerated gradient (AG)
5. Steepest descent for power minimization

Variations of these algorithms come from introducing constraints into the adjustment rule, and one section develops the procedure for deriving such variations. Finally, changes in the modes of adaptation are discussed, illustrating how two-mode adaptation enhances the convergence.

4.1 INTRODUCTORY CONCEPTS

The method of steepest descent is the most common gradient algorithm applied to quadratic performance surfaces. Any quadratic performance surface has a unique minimum point that can be found by using a gradient-based algorithm.

4.1.1 The Quadratic Performance Surface

A bowl-shaped quadratic performance surface, like the mean square error (MSE) performance measure for the adaptive array of Figure 3-3, is the ideal playing field for a gradient-based algorithm. Recall from Chapter 3 that the array output signal is given by

$$y(t) = \mathbf{w}^T(t)\mathbf{x}(t) \tag{4.1}$$

Denoting the desired array response by $d(t)$, we may express the error signal as

$$e(t) = d(t) - y(t) = d(t) - \mathbf{w}^T(t)\mathbf{x}(t) \tag{4.2}$$

The square of the foregoing error signal is then

$$e^2(t) = d^2(t) - 2d(t)\mathbf{x}^T(t)\mathbf{w}(t) + \mathbf{w}^T(t)\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{w}(t) \tag{4.3}$$

The MSE is just the expected value of $e^2(t)$, or

$$\begin{aligned} E\{e^2(t)\} &= \xi[\mathbf{w}(t)] \\ &= \overline{d}^2(t) - 2\mathbf{r}_{xd}^T(t)\mathbf{w}(t) + \mathbf{w}^T(t)\mathbf{R}_{xx}(t)\mathbf{w}(t) \end{aligned} \tag{4.4}$$

where the overbar denotes expected value, $\mathbf{r}_{xd}(t)$ is given by (3.52), and $\mathbf{R}_{xx}(t)$ is given by (3.13). When the input signals are statistically stationary, then $\mathbf{r}_{xd}$ and $\mathbf{R}_{xx}$ are also stationary, and there is no need to write these quantities as a function of time. In nonstationary signal environments, however, the notation $\mathbf{r}_{xd}(t)$ and $\mathbf{R}_{xx}(t)$ is required.

The MSE in (4.4) is a quadratic function of the weight vector $\mathbf{w}(t)$. In the nonstationary case, the bottom of the bowl as well as its curvature and orientation change. The analysis of time-varying adaptive performance signal statistics is beyond the scope of this book.

4.1.2 The Method of Steepest Descent

If the statistics of the signal environment are perfectly known, then the gradient at any point on the performance surface can be calculated. The gradient of (4.4) with respect to the weight vector is [1]

$$\{\xi[\mathbf{w}(t)]\} = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}\mathbf{w}(t) \tag{4.5}$$

It was shown in Chapter 3 that the minimum is the Wiener solution

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{4.6}$$

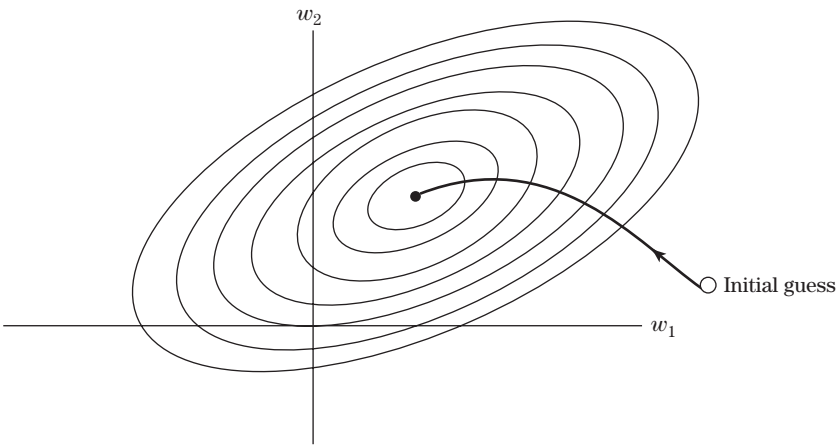


FIGURE 4-1 ■ Steepest descent with very small step size (overdamped case).

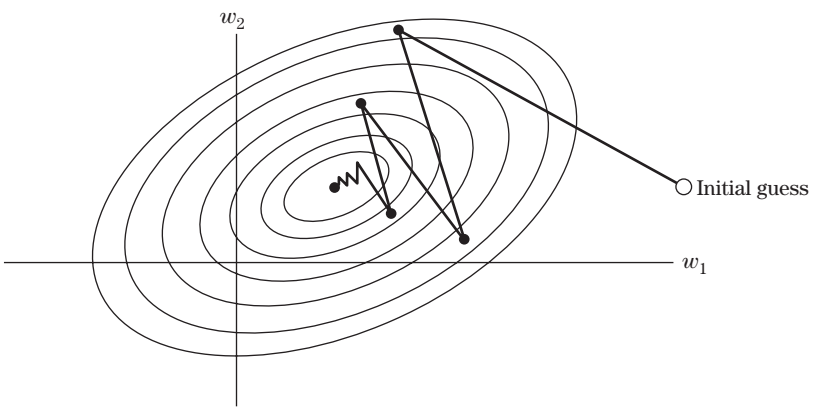


FIGURE 4-2 ■ Steepest descent with large step size (underdamped case).

On substituting (4.6) into (4.4), the minimum MSE is then found to be

$$\xi_{\min} = \overline{d}^2(t) - \mathbf{w}_{\text{opt}}^T \mathbf{r}_{xd} \tag{4.7}$$

The method of steepest descent begins with an initial guess of the weight vector components. Having selected a starting point, we then calculate the gradient vector and perturb the weight vector in the opposite direction (i.e., in the direction of the steepest downward slope). Contour plots of a quadratic performance surface (corresponding to a two-weight adjustment problem) are shown in Figures 4-1 and 4-2. In these figures the MSE is measured along a coordinate normal to the plane of the paper. The ellipses in these figures are contours of constant MSE. The gradient is orthogonal to these constant value contours (pointing in the steepest direction) at every point on the performance surface. If the steepest descent uses small steps, it is "overdamped," and the path taken to the bottom appears continuous as shown in Figure 4-1. If the steepest descent uses large steps, it is "underdamped," and each step is normal to the error contour as shown in Figure 4-2.

The discrete form of the method of steepest descent is [1]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta_s \quad \left[\overline{e}^2(k)\right] \tag{4.8}$$

where

$\mathbf{w}(k)$ = old weight vector guess at time kT
$\mathbf{w}(k+1)$ = new weight vector guess at time $(k+1)T$
$\overline{[e^2(k)]}$ = gradient vector of the MSE determining the direction in which to move from $\mathbf{w}(k)$
Δ_s = step size

Substituting the gradient of (4.5) into (4.8) then yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\Delta_s(\mathbf{R}_{xx}\mathbf{w}(k) - \mathbf{r}_{xd}) \tag{4.9}$$

4.1.3 Feedback Model of Steepest Descent

The transient behavior of the method of steepest descent yields valuable insight into the behavior of the LMS algorithm. The only difference between the two weight adjustment algorithms is that with steepest descent the signal environment statistics are perfectly known (so the gradient at any point can be exactly determined), whereas the LMS algorithm signal statistics are unknown (although here they are assumed to be stationary) and therefore must be estimated. The first step in determining the transient behavior of the method of steepest descent is to formulate a feedback model of the weight adjustment relationship.

Figure 4-3 is the feedback flow graph of (4.8) and (4.9). The symbol Z^{-1} is the Z-transform representation [2–5] of a unit (one iteration cycle) time delay, and $Z^{-1}\mathbf{I}$ is the matrix transfer function of a unit delay branch. This flow graph represents a first-order multidimensional sampled-data control loop.

By setting the initial weight vector $\mathbf{w}(0)$ equal to the initial guess in the flow graph, the resulting sequence of $\mathbf{w}(k)$ behaves exactly as in the local minimization weight adjustment algorithm. Since the "output" of the flow graph model is the current weight vector $\mathbf{w}(k)$, the flow graph model determines the transient behavior of the weight adjustment algorithm.

Each transfer function appearing in the flow graph of Figure 4-3 is a diagonal matrix except for the feedback branch denoted by $2\mathbf{R}_{xx}$. This branch matrix in general has off-diagonal elements, since the input signals are usually mutually correlated. Consequently, transients cross-couple from one component of the weight vector to the next, thereby complicating the study of transient behavior. A remedy is to diagonalize the flow graph and eliminate such cross-coupling effects from consideration: the diagonalization then enables one to consider the natural modes of behavior of the flow graph by merely introducing a coordinate transformation.

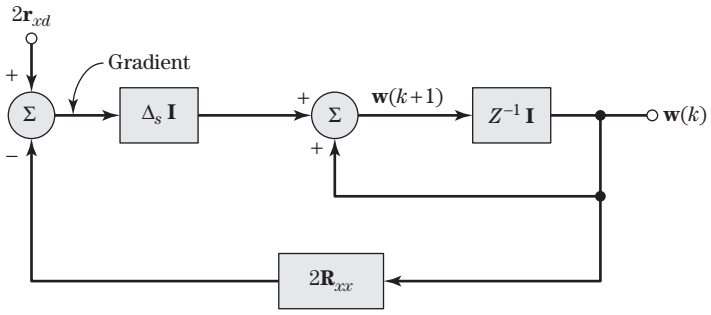


FIGURE 4-3 ■ Feedback model of steepest descent.

To diagonalize the flow graph of Figure 4-3, consider the expression for the MSE given by (4.4). Using $\mathbf{w}_{\text{opt}}$ and $\xi_{\min}$ in (4.6) and (4.7), the MSE becomes

$$E\{e^2(k)\} = \xi(k) = \xi_{\min} + [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}]^T \mathbf{R}_{xx} [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}] \tag{4.10}$$

Since the matrix $\mathbf{R}_{xx}$ is real, symmetric, and positive definite (for real variables), it is diagonalized by means of a unitary transformation matrix $\mathbf{Q}$ so that

$$\mathbf{R}_{xx} = \mathbf{Q}^{-1} \mathbf{\Lambda} \mathbf{Q} \tag{4.11}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, and $\mathbf{Q}$ is the modal square matrix of eigenvectors. If $\mathbf{Q}$ is constructed from normalized eigenvectors, then it is orthonormal so that $\mathbf{Q}^{-1} = \mathbf{Q}^T$, and the MSE becomes

$$\xi(k) = \xi_{\min} + [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}]^T \mathbf{Q}^T \mathbf{\Lambda} \mathbf{Q} [\mathbf{w}(k) - \mathbf{w}_{\text{opt}}] \tag{4.12}$$

Now define

$$\mathbf{Q}\mathbf{w}(k) \triangleq \mathbf{w}'(k) \tag{4.13}$$

$$\mathbf{Q}\mathbf{w}_{\text{opt}} \triangleq \mathbf{w}'_{\text{opt}} \tag{4.14}$$

Equation (4.12) can then be rewritten as

$$\xi(k) = \xi_{\min} + [\mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}]^T \mathbf{\Lambda} [\mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}] \tag{4.15}$$

$\mathbf{Q}$ projects $\mathbf{w}(k)$ into the primed coordinates [resulting in $\mathbf{w}'(k)$]. Since $\mathbf{\Lambda}$ is a diagonal matrix and $\xi(k)$ is a quadratic performance surface, the primed coordinates comprise the principal axes of the quadratic performance surface. The feedback model of Figure 4-3 expresses all quantities in terms of the primed coordinate system. The resulting equivalent feedback model of Figure 4-4 eliminates all cross-couplings that previously existed within the feedback paths.

The steepest descent algorithm in Figure 4-4 is composed of the natural modes of the flow graph. The transients of each mode are isolated (since each of the primed coordinates has its own natural mode), and the natural behavior of steepest descent is completely explored by considering the behavior of a single primed coordinate.

An isolated one-dimensional feedback model for the pth normal coordinate is shown in Figure 4-5. The pulse transfer function of this closed-loop feedback system is [1]

$$\frac{\mathbf{w}'_p(z)}{r'_p(z)} = \frac{\Delta_s Z^{-1}}{1 + (1 - 2\Delta_s \lambda_p) Z^{-1}} \tag{4.16}$$

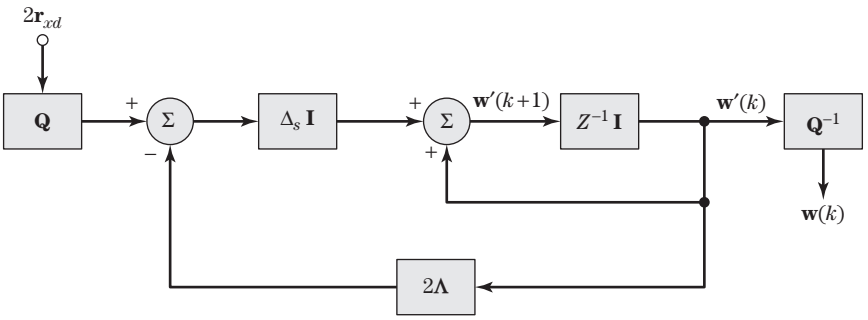


FIGURE 4-4 ■ Diagonalized version of feedback model of steepest descent using normal (primed) coordinates.

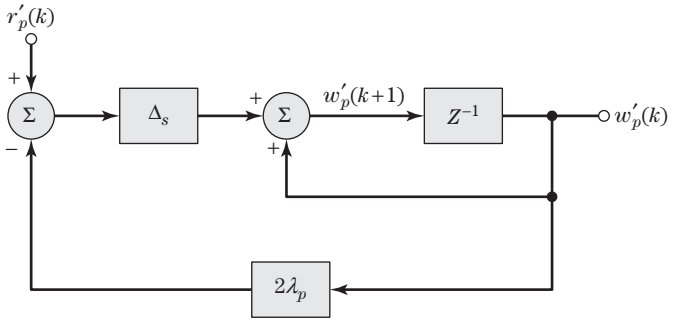


FIGURE 4-5 ■ One-dimensional feedback model for the *p*th normal coordinate of steepest descent.

where λ_p is the pth eigenvalue of $\mathbf{R}_{xx}$. The impulse response of (4.16) is found by letting $r'_p(z) = 1$ and taking the inverse Z-transform of the resulting output $\mathcal{Z}^{-1}\{w'_p(z)\}$. It follows that the impulse response is of the form

$$w'_p(kT) = \text{constant} \times e^{-\alpha_p(kT)}$$

where

$$\alpha_p = -\frac{1}{T} ln(1 - 2\Delta_s\lambda_p) \tag{4.17}$$

and T = one iteration period. The time response of (4.17) is a stable system when

$$|1 - 2\Delta_s\lambda_p| < 1 \tag{4.18}$$

Equation (4.18) is satisfied when

$$\Delta_s > 0 \quad \text{and} \quad |\Delta_s\lambda_p| < 1 \tag{4.19}$$

Since $\mathbf{R}_{xx}$ is positive definite, $\lambda_p > 0$ for all p. Consequently, the stability of the multidimensional flow graph of Figure 4-4 is guaranteed if and only if $\lambda_p = \lambda_{\max}$ in (4.19) and

$$\Delta_s > 0 \quad \text{and} \quad |\Delta_s\lambda_{\max}| < 1 \tag{4.20}$$

The stability of the steepest descent adaptation process is therefore guaranteed so long as

$$\frac{1}{\lambda_{\max}} > \Delta_s > 0 \tag{4.21}$$

4.2 | THE LMS ALGORITHM

When the operational environment signal statistics are stationary but unknown (a typical situation), then the gradient of the performance surface at any point must be estimated. The LMS algorithm introduced by Widrow has proven particularly useful for a quadratic performance function [6–10]. It is worthwhile noting that the LMS algorithm requires a reference signal, $d(t)$, to generate the error signal given by (4.2). The desired signal in a communications system is usually present, so the actual signal is used as the reference signal. In systems where the desired signal is usually not present (as in radar or

sonar systems), it is pointless to try to generate a fictitious desired signal. Thus, the LMS algorithm described here is usually employed to improve communications system performance. The LMS algorithm is exactly like the method of steepest descent except that now changes in the weight vector are made in the direction given by an estimated gradient vector instead of the actual gradient vector. In other words, changes in the weight vector are expressed as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \Delta_s \, \hat{} \, [\xi(k)] \tag{4.22}$$

where

$\mathbf{w}(k) =$ weight vector before adaptation step
$\mathbf{w}(k+1) =$ weight vector after adaptation step
$\Delta_s =$ step size that controls rate of convergence and stability
$\hat{} \, [\xi(k)] =$ estimated gradient vector of ξ with respect to $\mathbf{w}$

The adaptation process described by (4.22) attempts to find a solution as close as possible to the Wiener solution given by (4.6). It is tempting to try to solve (4.6) directly, but such an approach has several drawbacks:

1. Computing and inverting an $N \times N$ matrix when the number of weights N is large becomes more challenging as input data rates increase.
2. This method may require up to $[N(N+3)]/2$ autocorrelation and cross-correlation measurements to find the elements of $\mathbf{R}_{xx}$ and $\mathbf{r}_{xd}$. In many practical situations, such measurements must be repeated whenever the input signal statistics change.
3. Implementing a direct solution requires setting weight values with high accuracy in open loop fashion, whereas a feedback approach provides self-correction of inaccurate settings, thereby giving tolerance to hardware errors.

To obtain the estimated gradient of the MSE performance measure, take the gradient of a single time sample of the squared error as follows:

$$\hat{} \, _k = \quad [\xi(k)] = 2e(k) \quad [e(k)] \tag{4.23}$$

Since

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w} \tag{4.24}$$

it follows that

$$[e(k)] = \quad [d(k) - \mathbf{x}^T(k)\mathbf{w}] = -\mathbf{x}(k) \tag{4.25}$$

so that

$$\hat{} \, _k = -2e(k)\mathbf{x}(k) \tag{4.26}$$

It is easy to show that the gradient estimate given by (4.26) is unbiased by considering the expected value of the estimate and comparing it with the gradient of the actual MSE. The expected value of the estimate is given by

$$E\{\hat{} \, _k\} = -2E\{\mathbf{x}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\} \tag{4.27}$$

$$= -2[\mathbf{r}_{xd}(k) - \mathbf{R}_{xx}(k)\mathbf{w}(k)] \tag{4.28}$$

Now consider the MSE

$$\xi[\mathbf{x}(k)] = E[d^2(k)] + \mathbf{w}^T \mathbf{R}_{xx}(k)\mathbf{w} - 2\mathbf{w}^T \mathbf{r}_{xd}(k) \tag{4.29}$$

Differentiating (4.29) with respect to $\mathbf{w}$ yields the gradient $\{\xi[\mathbf{w}(k)]\}$ as

$$\{\xi[\mathbf{w}(k)]\} = 2\mathbf{R}_{xx}(k)\mathbf{w}(k) - 2\mathbf{r}_{xd}(k) \tag{4.30}$$

Comparing (4.28) and (4.30) reveals that

$$E\{\hat{\ }_k\} = \{\xi[\mathbf{w}(k)]\} \tag{4.31}$$

so the expected value of the estimated gradient equals the true value of the gradient of the MSE.

Substituting the estimated gradient of (4.26) into the weight adjustment rule of (4.22) then yields the weight control rule

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\Delta_s e(k)\mathbf{x}(k) \tag{4.32}$$

The LMS algorithm given by (4.32) can be rewritten for complex quantities as

$$\frac{\mathbf{w}(k+1) - \mathbf{w}(k)}{\Delta t} = 2k_s e(k)\mathbf{x}^*(k) \tag{4.33}$$

where Δt is the elapsed time between successive iterations, and $\Delta_s = k_s \Delta t$. In the limit as $\Delta t \to 0$, (4.33) yields an equivalent differential equation representation of the LMS algorithm that is appropriate for use in continuous systems as

$$\frac{d\mathbf{w}(t)}{dt} = 2k_s e(t)\mathbf{x}^*(t) \tag{4.34}$$

Equation (4.34) can also be written as

$$\mathbf{w}(t) = 2k_s \int_0^t e(\tau)\mathbf{x}^*(\tau)d\tau + \mathbf{w}(0) \tag{4.35}$$

A block diagram representation of the weight adjustment rule represented by (4.35) is shown in Figure 4-6.

The discrete version of (4.34) is given by (4.33) and is more commonly written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2k_s \Delta t e(k)\mathbf{x}^*(k) \tag{4.36}$$

A block diagram representation of the weight adjustment rule represented by (4.35) is illustrated in Figure 4-7.

4.2.1 Convergence to the Wiener Solution

Assume that the time between successive iterations of the LMS algorithm is long enough so that the signal vectors $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$ are uncorrelated. From (4.32) it follows that $\mathbf{w}(k)$ is a function of only $\mathbf{x}(k-1), \mathbf{x}(k-2), \ldots, \mathbf{x}(0)$, and $\mathbf{w}(0)$, where the successive input signal vectors are uncorrelated so that $\mathbf{w}(k)$ is independent of $\mathbf{x}(k)$. It will now be shown that for a stationary input signal process meeting these conditions, the expected value of the weight vector $E\{\mathbf{w}(k)\}$ converges to the Wiener solution given by (4.6).

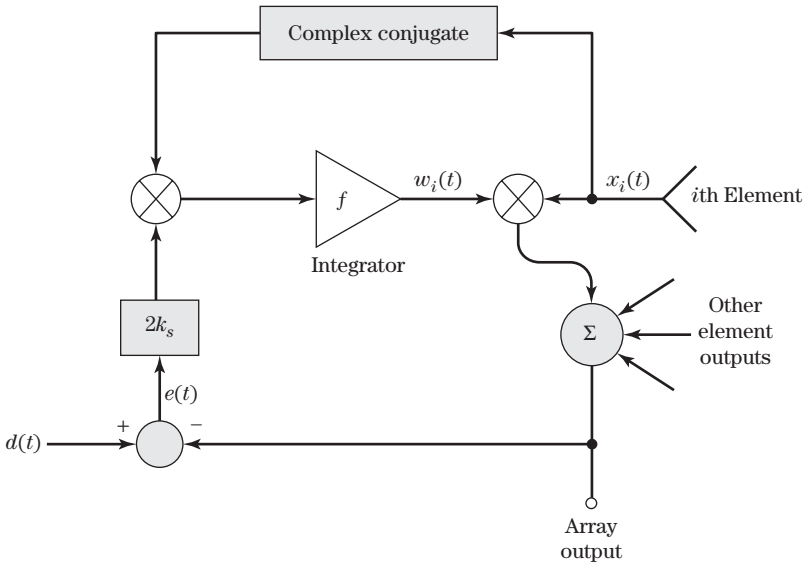


FIGURE 4-6 ■ Analog realization of the LMS weight adjustment algorithm.

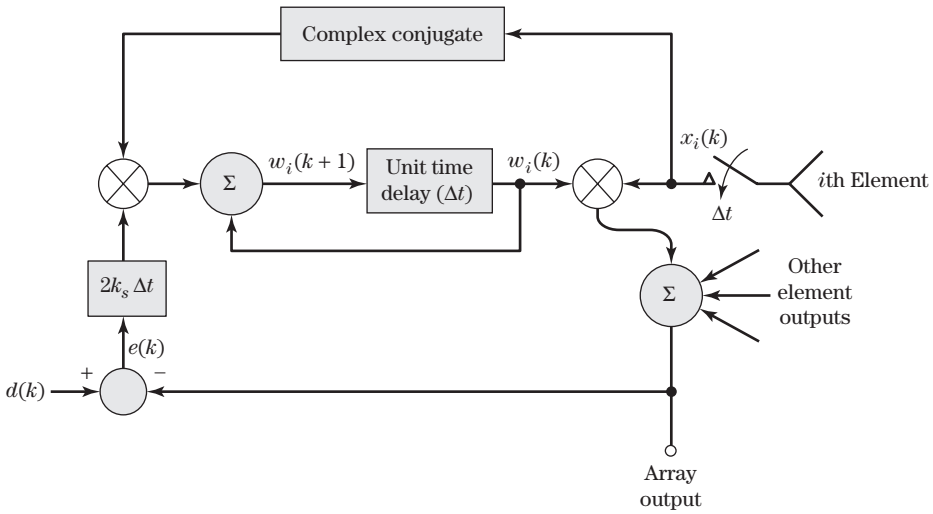


FIGURE 4-7 ■ Digital realization of the LMS weight adjustment algorithm.

Taking the expected value of both sides of (4.36), there results

$$E\{\mathbf{w}(k+1)\} = E\{\mathbf{w}(k)\} + 2k_s\Delta t \cdot E\{\mathbf{x}^*(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\} \tag{4.37}$$

Now let

$$E\{\mathbf{x}^*(k)d(k)\} = \mathbf{r}_{xd} \tag{4.38}$$

$$E\{\mathbf{x}^*(k)\mathbf{x}^T(k)\} = \mathbf{R}_{xx} \tag{4.39}$$

Consequently, (4.37) is rewritten as

$$\begin{aligned} E\{\mathbf{w}(k+1)\} &= E\{\mathbf{w}(k)\} - 2k_s\Delta t\mathbf{R}_{xx}E\{\mathbf{w}(k)\} + 2k_s\Delta t\mathbf{r}_{xd} \\ &= [\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]E\{\mathbf{w}(k)\} + 2k_s\Delta t\mathbf{r}_{xd} \end{aligned} \tag{4.40}$$

Starting with an initial guess $\mathbf{w}(0)$, the $(k+1)$th iteration of (4.40) yields

$$E\{\mathbf{w}(k+1)\} = [\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]^{(k+1)}\mathbf{w}(0) + 2k_s\Delta t\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{R}_{xx}]^i\mathbf{r}_{xd} \tag{4.41}$$

Diagonalizing (4.41) by using (4.11) to obtain the normal form results in

$$E\{\mathbf{w}(k+1)\} = \mathbf{Q}^{-1}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^{(k+1)}\mathbf{Q}\mathbf{w}(0) + 2k_s\Delta t\mathbf{Q}^{-1}\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^i\mathbf{Q}\mathbf{r}_{xd} \tag{4.42}$$

When the magnitude of all the terms in the diagonal matrix $[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]$ are less than one, then

$$\lim_{k\to\infty}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^{(k+1)} \to 0 \tag{4.43}$$

Therefore, the first term of (4.42) vanishes after a sufficient number of iterations, and the summation factor in the second term of (4.42) becomes

$$\lim_{k\to\infty}\sum_{i=0}^{k}[\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}]^i = \frac{1}{2k_s\Delta t}\mathbf{\Lambda}^{-1} \tag{4.44}$$

Therefore, after a sufficient number of iterations, (4.42) yields

$$\begin{aligned}\lim_{k\to\infty}E\{\mathbf{w}(k+1)\} &= 2k_s\Delta t\mathbf{Q}^{-1}\left(\frac{1}{2k_s\Delta t}\mathbf{\Lambda}^{-1}\right)\mathbf{Q}\mathbf{r}_{xd}\\ &= \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}\end{aligned} \tag{4.45}$$

This result shows that the expected value of the weight vector in the LMS algorithm does converge to the Wiener solution after a sufficient number of iterations.

Since all the eigenvalues in $\mathbf{\Lambda}$ are positive, it follows that all the terms in the aforementioned diagonal matrix, $\mathbf{I} - 2k_s\Delta t\mathbf{\Lambda}$, have a magnitude less than one provided that

$$\begin{aligned}|1 - 2k_s\Delta t\lambda_{\max}| &< 1\\ \frac{1}{\lambda_{\max}} &> k_s\Delta t > 0\end{aligned} \tag{4.46}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{R}_{xx}$. The convergence condition (4.46) is exactly the same as the stability condition (4.21) for the noise-free steepest descent feedback model.

The foregoing condition on k_s for convergence of the mean value of the LMS algorithm relates to the total input signal power P_{IN}, as described henceforth. Since $\lambda_{\max}$ satisfies the inequality

$$\lambda_{\max} \le \text{trace}[\mathbf{R}_{xx}] \tag{4.47}$$

where

$$\text{trace}[\mathbf{R}_{xx}] = E\{\mathbf{x}^{\dagger}(k)\mathbf{x}(k)\} = \sum_{i=1}^{N} E\{|x_i|^2\} \triangleq P_{\text{IN}} \tag{4.48}$$

then the convergence condition (4.46) is assured if

$$\frac{1}{P_{\text{IN}}} > k_s \Delta t > 0 \tag{4.49}$$

These LMS algorithm convergence results assumed that successive input signal samples are independent. This independence assumption is probably overly restrictive, since Griffiths [11] presented experimental results that show that adaptation using highly correlated successive samples also converges to the Wiener solution, although the resulting steady-state MSE is slightly higher than what results for statistically independent successive samples. For some applications, mean squared convergence and its associated stability properties may be of concern, in which case more stringent conditions on k_s must be satisfied [12].

4.2.2 Transient Response Characteristics for LMS Adaptation

In normal coordinates, the adaptive weight transients consist of sums of exponentials with time constants given by

$$\tau_p = \frac{1}{2(k_s \Delta t)\lambda_p}, \quad p = 1, 2, \ldots, N \tag{4.50}$$

where λ_p is the pth eigenvalue of the correlation matrix $\mathbf{R}_{xx}$. Since τ_p is inversely proportional to λ_p, the transient response is fastest for strong signals (large λ_p) and slowest for weak signals (small λ_p). Thus, the LMS algorithm convergence time depends on the eigenvalue spread in $\mathbf{R}_{xx}$. The exponential having the longest time constant (corresponding to the smallest normal coordinate signal power) determines the transient response of the LMS algorithm. Often there is no choice for the value of the constant k_s that represents a good compromise between the various eigenvalues that will yield a desirably short transient period of operation.

Figure 4-8 shows a contour plot of a quadratic performance surface corresponding to two widely diverse eigenvalues. The highly elongated MSE contours in Figure 4-8 result in many adaptive iterations before weight values become acceptably close to the desired Wiener solution. In the event that all the eigenvalues are equal, then all the time constants are equal, and

$$\tau = \frac{1}{2(k_s \Delta t)\lambda} \tag{4.51}$$

A "learning curve" that plots the expected value of the performance measure at each stage of the learning process as a function of the number of adaptation iterations provides a convenient way of monitoring the progress of an adaptive process. It has just been shown that the underlying transient behavior of the adaptive weights has an exponential nature. Since the MSE is a quadratic function of the weight values, the transients in the MSE function are also exponential.

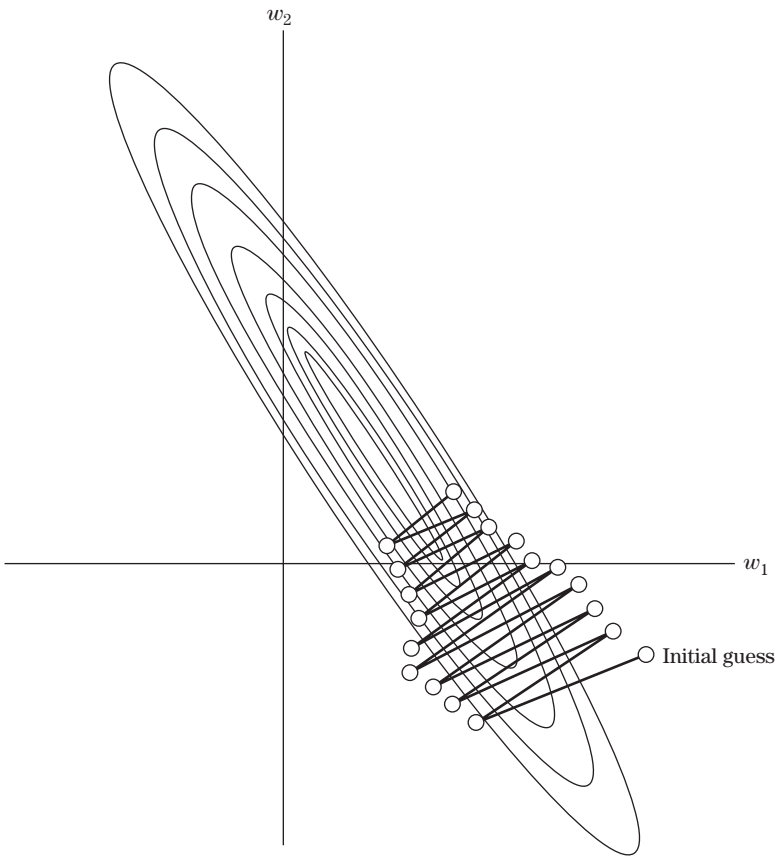


FIGURE 4-8 ■ Steepest descent transient response with widely diverse eigenvalues.

Since the square of an exponential function is an exponential having half the time constant of the original exponential function, it follows that when all the time constants are equal the MSE learning curve is an exponential having the time constant

$$\tau_{\text{MSE}} = \frac{\tau}{2} = \frac{1}{4(k_s \Delta t)\lambda} \tag{4.52}$$

In general, of course, the eigenvalues of $\mathbf{R}_{xx}$ are unequal so that

$$\tau_{p_{\text{MSE}}} = \frac{\tau_p}{2} = \frac{1}{4(k_s \Delta t)\lambda_p} \tag{4.53}$$

where $\tau_{p_{\text{MSE}}}$ is the time constant for the MSE learning curve, τ_p is the time constant in the weights, and λ_p is the eigenvalue of the pth normal mode. The adaptive process uses one signal data sample/iteration, so the time constant expressed in terms of the number of data samples is

$$T_{p_{\text{MSE}}} = \tau_{p_{\text{MSE}}} \tag{4.54}$$

Plots of actual experimental learning curves look like noisy exponentials—an effect due to the inherent noise that is present in the adaptation process. A slower adaptation rate (i.e., the smaller the magnitude of k_s) has a smaller noise amplitude that corrupts the learning curve.

4.2.3 Weight Misadjustment During LMS Adaptation

Adaptation speed of the LMS algorithm depends on two factors: (1) the weight adjustment step size and (2) the statistical averages sample size. If a large step size is selected for the weight adjustment process, then the excursions in successive weight values will be large, resulting in transient behavior like the underdamped case shown in Figure 4-2. If a small number of samples is used to estimate the statistical averages, then the time elapsed in obtaining such averages is small, but the quality of the resulting estimates is low. In general, a fast adaptive algorithm has a poor steady-state performance.

Since the input signal statistics are not known a priori, the minimum MSE is not the Wiener solution. The "misadjustment" M compares the actual MSE with the optimum Wiener solution

$$M \triangleq \frac{[\xi_{\text{actual}} - \xi_{\min}]}{\xi_{\min}} \tag{4.55}$$

where $\xi = E\{e^2\}$. The LMS algorithm misadjustment can be evaluated for a specified value of $k_s \Delta t$ by considering the noise associated with the gradient-estimation process.

Assume that the adaptive process converged to a steady state in the neighborhood of the MSE surface minimum point. The gradient-estimation noise of the adaptive algorithm at the minimum point (where the true gradient is zero) is just the gradient estimate itself. Therefore, the gradient noise vector $\mathbf{g}$ is given by

$$\mathrm{g}(k) = \hat{} \ (k) = -2e(k)\mathbf{x}(k) \tag{4.56}$$

The covariance of this estimation noise is given by

$$\text{cov}[\mathbf{g}(k)] = E\{\mathbf{g}(k)\mathbf{g}^T(k)\} = 4E\{e^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\} \tag{4.57}$$

When the weight vector is optimized ($\mathbf{w}(k) = \mathbf{w}_{\text{opt}}$), then the error $e(k)$ is uncorrelated with the input vector $\mathbf{x}(k)$. If $e(k)$ and $\mathbf{x}(k)$ are Gaussian processes, then not only are they uncorrelated at the minimum point of the MSE surface, but they are also statistically independent. With these conditions (4.57) becomes

$$\text{cov}[\mathbf{g}(k)] = 4E\{e^2(k)\}E\{\mathbf{x}(k)\mathbf{x}^T(k)\} = 4\xi_{\min}\mathbf{R}_{xx} \tag{4.58}$$

In the primed normal coordinates, the previous covariance can be written as

$$\text{cov}[\mathbf{g}'(k)] = \mathbf{Q}\text{cov}[\mathbf{g}(k)]\mathbf{Q}^{-1} = 4\xi_{\min}\mathbf{\Lambda} \tag{4.59}$$

Adaptation based on noisy gradient estimates results in noise in the weight vector. Recall that the noise-free method of steepest descent is described by the iterative relation

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta_s[- \quad (k)] \tag{4.60}$$

where Δ_s is the constant that controls stability and rate of convergence, and $\quad(k)$ is the gradient at the point on the performance surface corresponding to $\mathbf{w} = \mathbf{w}(k)$. Following Widrow and McCool [13], subtract $\mathbf{w}_{\text{opt}}$ from both sides of (4.60), and define $\mathbf{v}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_{\text{opt}}$ to obtain

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s[- \quad (k)] \tag{4.61}$$

With estimated gradients instead of exact gradients, (4.61) can be rewritten as

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s(-\hat{\ }(k)) = \mathbf{v}(k) + \Delta_s[- \quad (k) - \mathbf{g}(k)] \tag{4.62}$$

Now since $\quad(k)$ is given by (4.5), it follows that

$$(k) = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}[\mathbf{w}_{\text{opt}} + \mathbf{v}(k)] = 2\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.63}$$

Consequently, (4.62) can be written as

$$\mathbf{v}(k+1) = (\mathbf{I} - 2\Delta_s\mathbf{R}_{xx})\mathbf{v}(k) - \Delta_s\mathbf{g}(k) \tag{4.64}$$

which represents a first-order vector difference equation with a stochastic driving function—$\Delta_s\mathbf{g}(k)$. Multiplying (4.64) by $\mathbf{Q}$ produces

$$\mathbf{v}'(k+1) = (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k) - \Delta_s\mathbf{g}'(k) \tag{4.65}$$

After initial transients have died out and the steady state is reached, $\mathbf{v}'(k)$ responds to the stationary driving function $-\Delta_s\mathbf{g}'(k)$ in the manner of a stationary random process. The absence of any cross-coupling in the primed normal coordinate system means that the components of both $\mathbf{g}'(k)$ and $\mathbf{v}'(k)$ are mutually uncorrelated, and the covariance matrix of $\mathbf{g}'(k)$ is therefore diagonal. To find the covariance matrix of $\mathbf{v}'(k)$ consider

$$\begin{aligned}\mathbf{v}'(k+1)\mathbf{v}'^T(k+1) = {} & (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s\mathbf{\Lambda}) \\ & + \Delta_s^2\mathbf{g}'(k)\mathbf{g}'^T(k) - \Delta_s(\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\mathbf{v}'(k)\mathbf{g}'^T(k) \\ & - \Delta_s\mathbf{g}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\end{aligned} \tag{4.66}$$

Taking expected values of both sides of (4.66) (and noting that $\mathbf{v}'(k)$ and $\mathbf{g}'(k)$ are uncorrelated since $\mathbf{v}'(k)$ is affected only by gradient noise from previous iterations), we find

$$\begin{aligned}\text{cov}\,[\mathbf{v}'(k)] &= (\mathbf{I} - 2\Delta_s\mathbf{\Lambda})\text{cov}\,[\mathbf{v}'(k)](\mathbf{I} - 2\Delta_s\mathbf{\Lambda}) + \Delta_s^2\text{cov}\,[\mathbf{g}'(k)] \\ &= \Delta_s^2\left[4\Delta_s\mathbf{\Lambda} - 4\Delta_s^2\mathbf{\Lambda}^2\right]^{-1}\text{cov}\,[\mathbf{g}'(k)]\end{aligned} \tag{4.67}$$

In practical applications, the LMS algorithm uses a small value for Δ_s, so that

$$\Delta_s\mathbf{\Lambda} \ll \mathbf{I} \tag{4.68}$$

With (4.68) satisfied, the squared terms involving $\Delta_s\mathbf{\Lambda}$ in (4.67) may be neglected, so

$$\text{cov}\,[\mathbf{v}'(k)] = \frac{\Delta_s}{4}\mathbf{\Lambda}^{-1}\text{cov}\,[\mathbf{g}'(k)] \tag{4.69}$$

Using (4.59), we find

$$\text{cov}\,[\mathbf{v}'(k)] = \frac{\Delta_s}{4}\mathbf{\Lambda}^{-1}(4\xi_{\min}\mathbf{\Lambda}) = \Delta_s\xi_{\min}\mathbf{I} \tag{4.70}$$

Therefore, the covariance of the steady-state noise in the weight vector (near the minimum point of the MSE surface) is

$$\text{cov}\,[\mathbf{v}(k)] = \Delta_s\xi_{\min}\mathbf{I} \tag{4.71}$$

Without noise in the weight vector, the actual MSE experienced would be $\xi_{\min}$. The presence of noise in the weight vector causes the steady-state weight vector solution to randomly meander about the minimum point. This random meandering results in an "excess" MSE— that is, an MSE that is greater than $\xi_{\min}$. Since

$$\xi(k) = \overline{d}^2(k) - 2\mathbf{r}_{xd}^T\mathbf{w}(k) + \mathbf{w}^T(k)\mathbf{R}_{xx}\mathbf{w}(k) \tag{4.72}$$

where

$$\xi_{\min} = \overline{d}^2(k) - \mathbf{w}_{\text{opt}}^T\mathbf{r}_{xd} \tag{4.73}$$

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{4.74}$$

It follows that (4.72) can be rewritten as (also see (4.10))

$$\xi(k) = \xi_{\min} + \mathbf{v}^T(k)\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.75}$$

In terms of the primed normal coordinates, (4.75) can be rewritten as

$$\xi(k) = \xi_{\min} + \mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k) \tag{4.76}$$

It immediately follows from (4.76) that the average excess MSE is

$$E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} = \sum_{p=1}^{N}\lambda_p E\left\{\left[\mathbf{v}'_p(k)\right]^2\right\} \tag{4.77}$$

Using (4.70) to recognize that $E\{[\mathbf{v}'_p(k)]^2\}$ is just $\Delta_s\xi_{\min}$ for each p, we see it then follows that

$$\begin{aligned} E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} &= \Delta_s\xi_{\min}\sum_{p=1}^{N}\lambda_p \\ &= \Delta_s\xi_{\min}tr(\mathbf{R}_{xx}) \end{aligned} \tag{4.78}$$

The misadjustment in the LMS algorithm is therefore given by

$$M = \frac{E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\}}{\xi_{\min}} = \Delta_s tr(\mathbf{R}_{xx}) \tag{4.79}$$

Since $\Delta_s = k_s\Delta t$, (4.79) emphasizes the fact that the degree of misadjustment experienced with the LMS algorithm can be controlled merely by adjusting k_s. When the step size is decreased, the time required to reach the steady-state condition increases, so there is a trade-off between the misadjustment and the adaptation speed.

The LMS algorithm misadjustment can also be expressed in a manner that gives insight into the relationship between misadjustment and adaptation speed. From (4.53) it follows that

$$\Delta_s\lambda_p = \frac{1}{4\tau_{p_{\text{MSE}}}} \tag{4.80}$$

Furthermore

$$\Delta_s tr(\mathbf{R}_{xx}) = \Delta_s \sum_{p=1}^{N} \lambda_p = \sum_{p=1}^{N} \left(\frac{1}{4\tau_{p_{\text{MSE}}}} \right) = \frac{N}{4} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} \tag{4.81}$$

where

$$\left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} \triangleq \frac{1}{N} \sum_{p=1}^{N} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right) \tag{4.82}$$

Consequently, the misadjustment can be written as

$$M = \frac{N}{4} \left(\frac{1}{\tau_{p_{\text{MSE}}}} \right)_{\text{av}} = \frac{N}{4} \left(\frac{1}{T_{p_{\text{MSE}}}} \right)_{\text{av}} \tag{4.83}$$

where $T_{p_{\text{MSE}}}$ is the learning curve time constant in units of the number of data samples.

4.2.4 Practical Considerations for LMS Adaptation

Generation of the error signal in LMS adaptation requires an appropriate desired signal. If the desired signal is the signal itself, then the adaptive array output reproduces the signal in the best MSE sense and nearly eliminates the noise. As a practical matter, the signal is not available for adaptation purposes—indeed, if it were available there would be no need for a receiver and a receiving array.

An LMS algorithm artificially injects a known signal termed the *reference signal* or *pilot signal* for the desired signal. The pilot signal has the same (or similar) directional and spectral characteristics as those of the desired signal. These directional and spectral characteristics may sometimes be known a priori, but usually only estimates of these parameters are available. Many practical communication systems derive the reference signal from the array output—a practice that requires a high degree of compatibility between the signaling waveforms and the adaptive array. In general, it is not feasible to simply put an adaptive array in any arbitrary communication system because of the following reasons [14]:

1. The adaptive array weights are random processes that modulate the desired signal; consequently, either the desired signal waveforms or the adaptive algorithm must be chosen so this modulation does not impair the communication system effectiveness.
2. The desired signal and interference signal waveforms must be different in some respect, so this known difference can be exploited to enable the adaptive array to distinguish these two signal classes.
3. A practical method for reference-signal generation must be available.

The reference signal needs to satisfy only the following criteria [14]:

1. The reference signal must be highly correlated with the desired signal at the array output.
2. The reference signal must be uncorrelated with any interference signal components appearing at the array output.

If these two correlation properties are satisfied, then the adaptive array behaves in the desired manner, since only the correlation between the reference signal and the element signals $x_i(t)$ affects the adaptive weights. The impact of any phase shift occurring in

the network responsible for generating the reference signal (when the reference signal is derived from the array output) is discussed in [15].

LMS algorithm adaptation with an injected pilot signal causes the array to form a beam in the pilot-signal direction. This array beam has a flat spectra response and linear phase shift characteristic within the passband defined by the spectral characteristic of the pilot signal. Furthermore, directional noise incident on the array manifests as correlated noise components that the array will respond by producing beam pattern nulls in the noise direction within the array passband.

Since injection of the pilot signal could "block" the receiver (by rendering it insensitive to the actual signal of interest), mode-dependent adaptation schemes have been devised to overcome this difficulty. Two such adaptation algorithms are discussed in the following section.

4.2.5 One-Mode and Two-Mode LMS Adaptation

Figure 4-9 illustrates a practical two-mode method [10] for providing a pilot signal to form the array beam and then switching the pilot signal off to adapt to the inputs to eliminate noise. The ideal time delays $\delta_1, \delta_2, \ldots, \delta_N$ are selected to produce a set of input signals that appear to be a radiated plane wave from the desired direction. The adaptive processor inputs are connected either to the actual sensor element outputs (during adaptation to eliminate noise) or to the set of delayed signals obtained from the pilot signal generator and the selected time-delay elements (to preserve the main lobe in the desired direction).

During adaptation, all signals delivered to the adaptive processor are sensor element outputs derived from the actual noise field. The adaptation process in this mode tends to eliminate all received signals since the desired response signal has been set to zero.

To preserve the main beam in the desired direction during adaptation, the input signals to the adaptive processor are derived from the pilot signal. For example, if a sinusoidal pilot signal having frequency f_0 is used, then minimizing the MSE forces the array gain

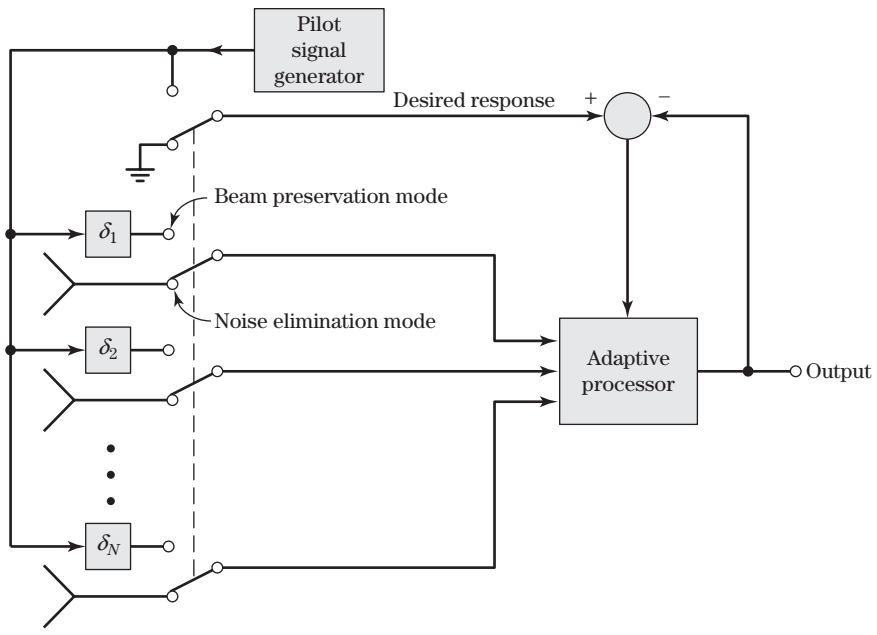


FIGURE 4-9 ■ Two-mode LMS adaptation for beam preservation and noise elimination.

in the desired look direction to have a specific amplitude and phase shift at that frequency. On the other hand, if the pilot signal is chosen to be the sum of several sinusoids having different frequencies, then the adaptation process forces the array gain and phase in the desired look direction to have specific values at each one of the pilot-signal frequencies. Finally, if several pilot signals corresponding to different look directions are added together, then the array gain is simultaneously constrained at the various frequencies and angles corresponding to the different pilot signals selected. In summary, the two-mode adaptation process minimizes the total power of all signals received that are uncorrelated with the pilot signals while constraining the gain and phase of the array beam to values corresponding to the frequencies and angles dictated by the pilot-signal components.

Figure 4-10 illustrates a practical one-mode method for simultaneously eliminating all noises uncorrelated with the pilot signal and forming a desired array beam. The circuitry of Figure 4-10 circumvents the difficulty of being unable to receive the actual signal, while the processor is connected to the pilot-signal generator by introducing an auxiliary adaptive processor. For the auxiliary adaptive processor, the desired response is the pilot signal, and both the pilot signal and the actual received signals enter the processor. A second processor performs no adaptation (its weights are slaved to the weights of the adaptive processor) and generates the actual array output signal. The slaved processor inputs do not contain the pilot signal and can therefore receive the transmitted signal at all times.

In the one-mode adaptation method, the pilot signal is on continuously so the adaptive processor that minimizes the MSE forces the adaptive processor output to closely reproduce the pilot signal while rejecting all signals uncorrelated with the pilot signal.

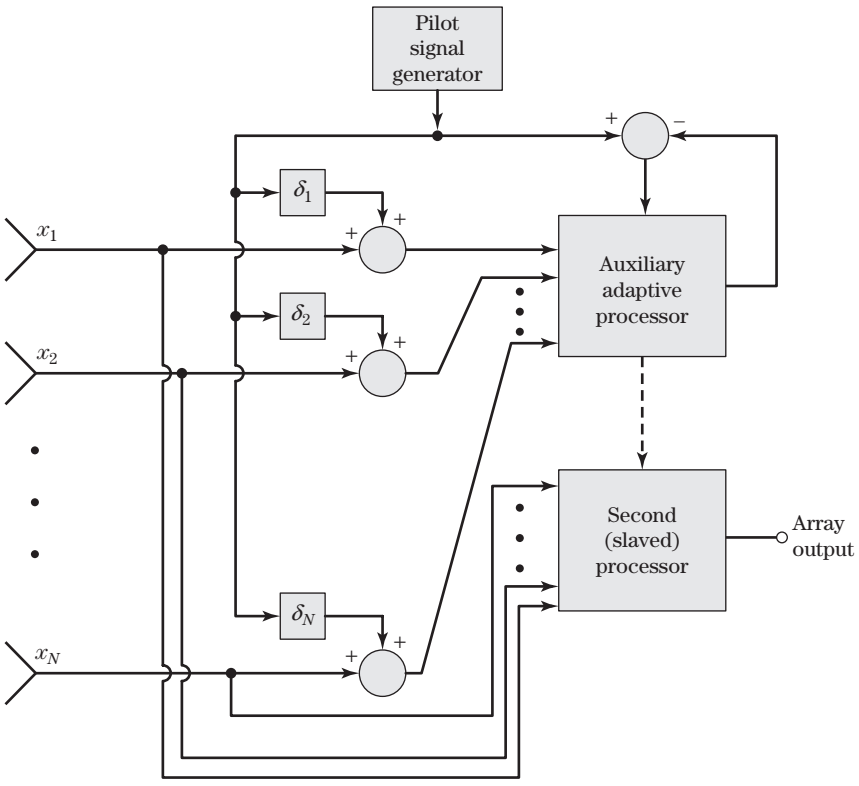


FIGURE 4-10 ■ One-mode LMS adaptation for simultaneous beam preservation and noise elimination.

The adaptive processor therefore preserves the desired array directivity in the look direction (over the pilot-signal passband) while placing nulls in the directions of noise sources (over the noise frequency bands).

4.3 | THE HOWELLS–APPLEBAUM ADAPTIVE PROCESSOR

The key capability of adaptive nulling was developed for an intermediate frequency (IF) radar sidelobe canceller as represented by the patent of Howells [16]. An analysis of this approach by Applebaum [17] established the control-law theory governing the operation of an adaptive control loop for each array element. The Applebaum algorithm maximizes a generalized signal-to-noise ratio (SNR) with the assumptions that the desired signal is absent most of the time (as in a pulsed radar or sonar system) and the direction of arrival of the desired signal is known. Because the Howells–Applebaum processor is practical to implement, it has been applied extensively to the problem of clutter and interference rejection in radar systems [18–21]. Unless otherwise noted, the analysis in this section follows the treatment of this subject given by Gabriel [22].

A six-element linear array with Howells–Applebaum loops at each element is shown in Figure 4-11. This figure shows the close relationship between an LMS loop and a

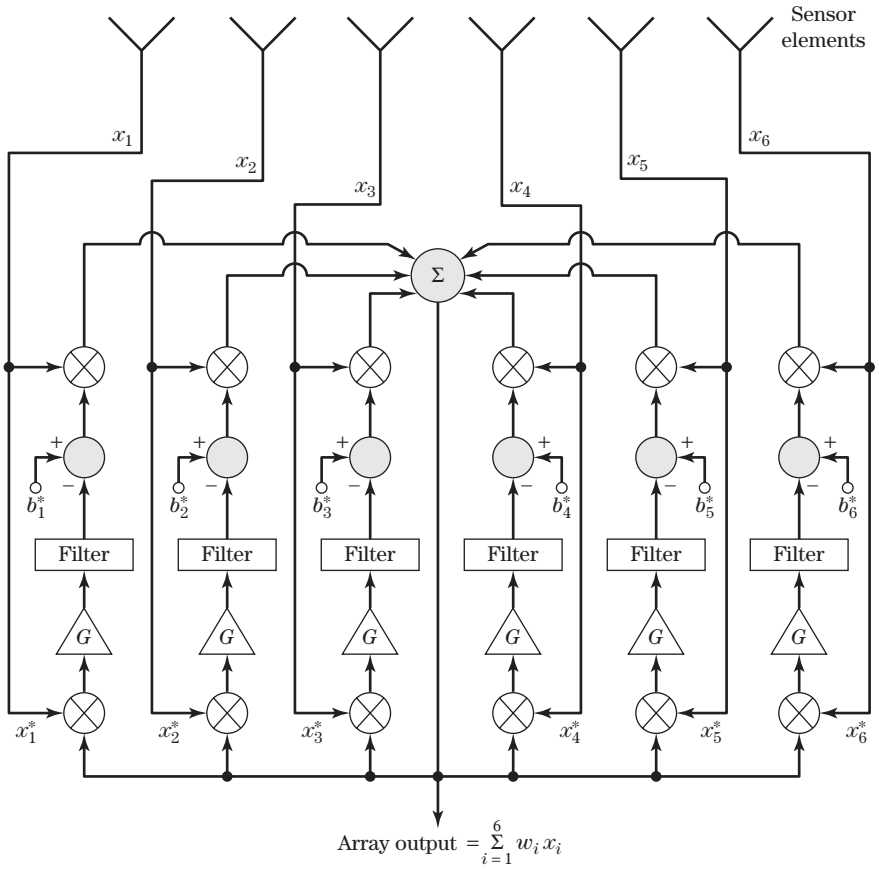


FIGURE 4-11 ■ Six-element linear adaptive array having six Howells–Applebaum control loops with beam steering signals. From Gabriel, *Proc. IEEE*, February 1976.

Howells–Applebaum loop. The beam steering vector, $\mathbf{b}^*$, in the Howells–Applebaum loop plays the same role as the cross-correlation vector, $\mathbf{r}_{xd}$, in the LMS loop. Define an element signal vector $\mathbf{x}$ in which the kth component, x_k, consists of the quiescent receiver channel noise voltage n_k and a summation of voltage terms associated with I external, narrowband interference sources:

$$\mathbf{x}^T = [x_1, x_2, \ldots, x_N] \tag{4.84}$$

where

$$x_k = n_k + \sum_{i=1}^{I} J_i e^{j\psi_i(2k-N-1)/2} \tag{4.85}$$

and

$$\psi_i = \left(\frac{2\pi d}{\lambda}\right) \sin\theta_i \tag{4.86}$$

The interference sources are assumed to be statistically independent where J_i is the element channel voltage amplitude associated with the ith source at θ_i.

Beam steering signals steer the receive beam to θ_s. For quiescent conditions (when only receiver noise is present), the adaptive weights settle to steady-state values denoted by the quiescent weight vector $\mathbf{w}_q$, where

$$\mathbf{w}_q^T = [w_{q_1}, w_{q_2}, \ldots, w_{q_N}] \tag{4.87}$$

and

$$w_{q_k} = a_k e^{-j\psi_0(2k-N-1)/2} \tag{4.88}$$

$$\psi_0 = \left(\frac{\pi d}{\lambda}\right) \sin\phi_0 \tag{4.89}$$

where a_k are the element amplitude weights. The resulting quiescent array factor is expressed as

$$AF_q(\theta) = (\mathbf{s}^T \mathbf{w}_q) = \sum_{k=1}^{N} a_k \exp[j(\psi - \psi_s)(2k - N - 1)/2] \tag{4.90}$$

where

$$\mathbf{s}^T = [s_1, s_2, \ldots, s_N] = \text{ element signals} \tag{4.91}$$

$$s_k = \exp[j\psi(2k - N - 1)/2] \tag{4.92}$$

$$\psi = \frac{2\pi d}{\lambda} \sin\theta \tag{4.93}$$

The components of the input beam steering vector $\mathbf{b}^*$

$$\mathbf{b}^{*T} = \left[b_1^*, b_2^*, \ldots, b_N^*\right] \tag{4.94}$$

are directly related to the components of $\mathbf{w}_q$ by the relation

$$b_k^* = c_k w_{q_k} \tag{4.95}$$

where the constants c_k are evaluated in the section immediately following.

4.3.1 Adaptive Weight Equations

The weight w_k associated with the kth sensor element is given by

$$w_k = b_k^* - z_k \tag{4.96}$$

where z_k = output voltage from the kth integrating RC filter. Each correlation mixer voltage is given by the product of the signal v_k^* with the summed array output

$$v_k = k^2 \left(x_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.97}$$

where

$$\tau_0 \frac{dz_k}{dt} + z_k = \gamma \left(x_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.98}$$

$$\gamma = k^2 G \tag{4.99}$$

The constant γ represents a conversion-factor gain constant that is assumed to be the same for all the loops. It is convenient to use (4.96) to convert from z_k to w_k, so that (4.98) now becomes

$$\tau_0 \frac{dw_k}{dt} + w_k = b_k^* - \gamma \left[x_k^* \sum_{i=1}^{N} w_i x_i \right] \tag{4.100}$$

Using matrix notation, we may write the complete set of N differential equations corresponding to (4.100) as

$$\tau_0 \frac{d\mathbf{w}}{dt} + \mathbf{w} = \mathbf{b}^* - \gamma \left[\mathbf{x}^* \mathbf{w}^T \mathbf{x} \right] \tag{4.101}$$

Since $(\mathbf{w}^T \mathbf{x}) = (\mathbf{x}^T \mathbf{w}) = \sum_{i=1}^{N} w_i x_i$, the bracketed term in (4.101) can be rewritten as

$$[\mathbf{x}^* \mathbf{w}^T \mathbf{x}] = [\mathbf{x}^* \mathbf{x}^T] \mathbf{w} \tag{4.102}$$

The expected (averaged) value of $\mathbf{x}^* \mathbf{x}^T$ yields the input signal correlation matrix

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^* \mathbf{x}^T\} \tag{4.103}$$

The averaged values of the correlation components forming the elements of $\mathbf{R}_{xx}$ are given by

$$\overline{x_k^* x_l} = \begin{cases} \displaystyle\sum_{i=1}^{I} |\overline{J}_i|^2 \exp[j\psi_i (l-k)] & l \neq k \quad (4.104) \\ |\overline{x}_k|^2 = |\overline{n}_k|^2 + \displaystyle\sum_{i=1}^{I} |\overline{J}_i|^2 & l = k \quad (4.105) \end{cases}$$

Since the correlation matrix in the absence of the desired signal is the sum of the quiescent receiver noise matrix $\mathbf{R}_{nn_q}$ and the individual interference source matrixes $\mathbf{R}_{nn_i}$, it follows that

$$\mathbf{R}_{nn} = \mathbf{R}_{nn_q} + \sum_{i=1}^{I} \mathbf{R}_{nn_i} \tag{4.106}$$

where $\mathbf{R}_{nn_q}$ can be expressed as

$$\mathbf{R}_{nn_q} = \begin{bmatrix} |\overline{n}_1|^2 & 0 & 0 & \cdots \\ 0 & |\overline{n}_2|^2 & 0 & \cdots \\ & & \ddots & \\ 0 & \cdots & \cdots & |\overline{n}_N|^2 \end{bmatrix} \tag{4.107}$$

and

$$\mathbf{R}_{nn_i} = |\overline{J}_i|^2 \begin{bmatrix} 1 & e^{j\psi_i} & e^{j2\psi_i} & \cdots \\ e^{-j\psi_i} & 1 & e^{j\psi_i} & \cdots \\ e^{-j2\psi_i} & e^{-j\psi_i} & 1 & \cdots \\ & & & \ddots \\ & & & 1 \end{bmatrix} \tag{4.108}$$

Substituting $\mathbf{R}_{nn}$ of (4.106) into (4.101) and rearranging terms, the final expression for the adaptive weight matrix differential equation becomes

$$\tau_0 \frac{d\mathbf{w}}{dt} + [\mathbf{I} + \gamma \mathbf{R}_{nn}]\mathbf{w} = \mathbf{b}^* \tag{4.109}$$

where $\mathbf{I}$ is the identity matrix.

In general, $\mathbf{R}_{nn}$ is not diagonal, so multiplying $\mathbf{R}_{nn}$ by a nonsingular orthonormal model matrix, Q, results in a simple transformation of coordinates that diagonalizes $\mathbf{R}_{nn}$. The resulting diagonalized matrix has diagonal elements that are the eigenvalues of the matrix $\mathbf{R}_{nn}$. The eigenvalues of $\mathbf{R}_{nn}$ are given by the solutions of the equation

$$|\mathbf{R}_{nn} - \lambda_i \mathbf{I}| = 0, \quad i = 1, 2, \ldots, N \tag{4.110}$$

Corresponding to each eigenvalue there is an associated eigenvector $\mathbf{e}_i$ that satisfies

$$\mathbf{R}_{nn}\mathbf{e}_i = \lambda_i \mathbf{e}_i \tag{4.111}$$

These eigenvectors (which are normalized to unit length and are orthogonal to one another) make up the rows of the transformation matrix $\mathbf{Q}$, that is,

$$\mathbf{Q} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \cdots \\ e_{21} & e_{22} & e_{23} & \cdots \\ e_{31} & e_{32} & e_{33} & \cdots \\ \vdots & & & \\ e_{N1} & e_{N2} & e_{N3} & \cdots \end{bmatrix}, \quad \text{where } \mathbf{e}_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{iN} \end{bmatrix} \tag{4.112}$$

Once $\mathbf{R}_{nn}$ is diagonalized by the $\mathbf{Q}$-matrix transformation, there results

$$[\mathbf{Q}^*\mathbf{R}_{nn}\mathbf{Q}^T] = \begin{bmatrix} \lambda_1 & 0 & & 0 \\ 0 & \lambda_2 & & 0 \\ 0 & 0 & \lambda_3 & 0 \\ \vdots & & \ddots & \\ \cdots & \cdots & \cdot & \lambda_N \end{bmatrix} \tag{4.113}$$

Now since $\mathbf{R}_{nn} = E\{\mathbf{x}^*\mathbf{x}^T\}$, it follows that (4.113) may be written as

$$[\mathbf{Q}^*\mathbf{R}_{nn}\mathbf{Q}^T] = [\overline{\mathbf{Q}^*\mathbf{x}^*\mathbf{x}^T\mathbf{Q}^T}] = \overline{[\mathbf{x}'^*\mathbf{x}'^T]} = \mathbf{\Lambda} \tag{4.114}$$

where

$$\mathbf{x}' = \mathbf{Q}\mathbf{x} \tag{4.115}$$

The $\mathbf{Q}$ matrix transforms the real signal vector $\mathbf{x}$ into the orthonormal signal vector $\mathbf{x}'$. Furthermore, the components of $\mathbf{x}'$ are determined by the eigenvectors of $\mathbf{R}_{nn}$, that is,

$$x_k' = (\mathbf{e}_k^T\mathbf{x}) \tag{4.116}$$

Note that the orthonormal signal vector components x_k' have two special characteristics:

1. They are uncorrelated so

$$E\{x_k'^* x_l'\} = 0 \quad \text{for } l \neq k \tag{4.117}$$

2. Their amplitudes are given by the square root of the corresponding eigenvalue so that

$$E\{x_k'^* x_k'\} = \lambda_k \tag{4.118}$$

The transformation matrix $\mathbf{Q}$ yields the same signal components as an appropriately selected orthogonal beamforming network.

Just as the signal vector $\mathbf{x}$ was transformed into $\mathbf{x}'$ by (4.115), the beam steering vector $\mathbf{b}^*$ may likewise be transformed to define a new beam steering vector $\mathbf{b}'^*$ as

$$\mathbf{b}' = \mathbf{Q}\mathbf{b} \tag{4.119}$$

where the kth component of $\mathbf{b}'$ is determined by the kth eigenvector appearing in $\mathbf{Q}$.

The Q-coordinate transformation operating on both $\mathbf{x}$ and $\mathbf{b}^*$ suggests an equivalent circuit representation for the system that is illustrated in Figure 4-12b, where an equivalent "orthonormal adaptive array" system is shown alongside a simplified representation of the real system in Figure 4-12a. There are a set of weights forming the weight vector $\mathbf{w}'$ in the orthonormal system, and the adaptive weight matrix equation for the equivalent system is

$$\tau_0 \frac{d\mathbf{w}'}{dt} + [\mathbf{I} + \gamma\mathbf{R}_{nn}']\mathbf{w}' = b'^* \tag{4.120}$$

where

$$\mathbf{R}_{nn}' = E\{\mathbf{x}'^*\mathbf{x}'^T\} = \mathbf{\Lambda} \tag{4.121}$$

This diagonalization results in an orthonormal system, a set of independent linear differential equations, each of which has a solution when the eigenvalues are known. Each of the orthonormal servo loops in the equivalent system responds independently of the other loops, because the x_k' input signals are orthogonalized and are therefore completely uncorrelated with one another. The weight equation for the kth orthonormal servo loop can therefore be written as

$$\tau_0 \frac{dw_k'}{dt} + (1 + \gamma\lambda_k)w_k' = b_k'^* \tag{4.122}$$

Note that the equivalent servo gain factor can be defined from (4.122) as

$$\mu_k = \gamma\lambda_k \tag{4.123}$$

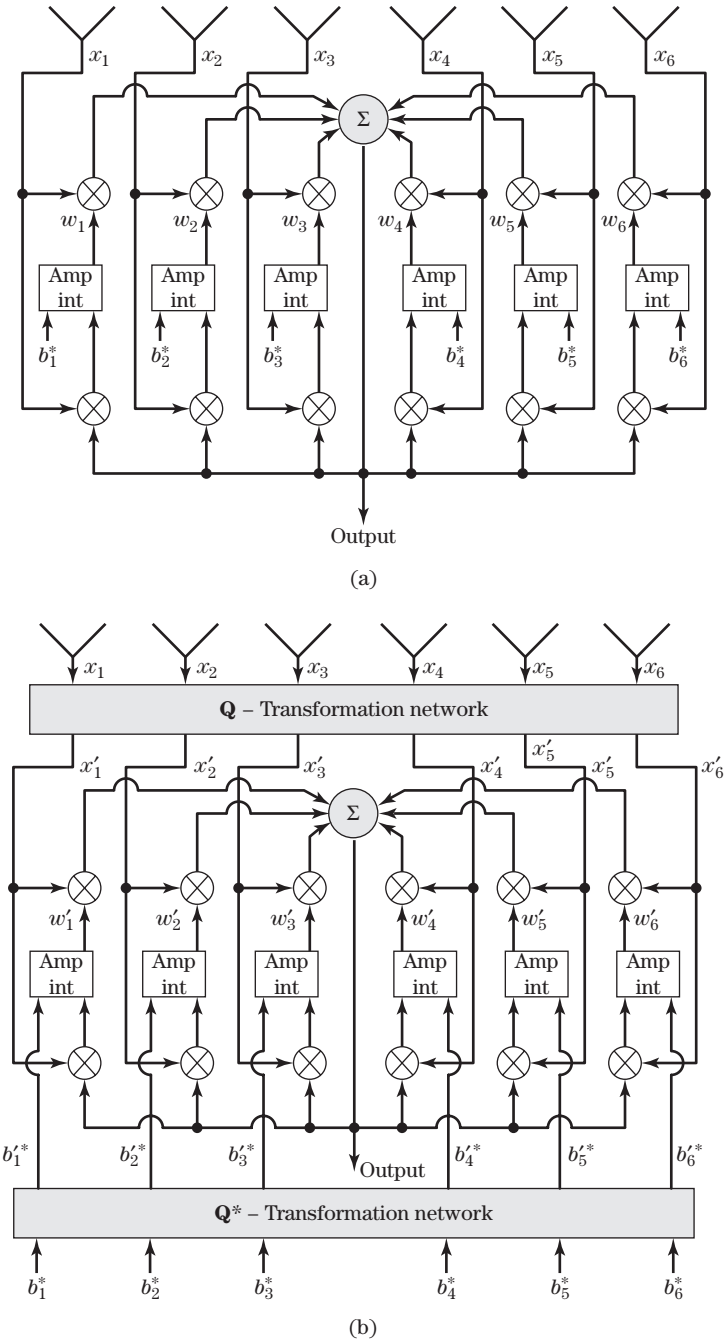


FIGURE 4-12 ■ Equivalent circuit representations for a six-element adaptive array system. a: Real adaptive array system. b: Equivalent orthonormal adaptive array system. From Gabriel, *Proc. IEEE*, February 1976.

so the equivalent servo gain factors for the various orthonormal loops are now determined by the eigenvalues of the input signal covariance matrix. The positive, real eigenvalues λ_k correspond to the square of a signal voltage amplitude, and any given eigenvalue is proportional to the power appearing at the orthonormal network output port.

For the input beam steering vector $\mathbf{b}^*$, the output desired signal power is given by

$$P_s = |\mathbf{w}^T \mathbf{b}|^2 \tag{4.124}$$

Likewise, the array output noise power can be written as

$$P_n = |\overline{\mathbf{w}^T\mathbf{x}}|^2 \tag{4.125}$$

where the signal vector $\mathbf{x}$ is assumed to be composed only of quiescent receiver channel noise plus the directional noise signal components due to external sources of interference. The signal-to-noise performance measure is therefore just a ratio of the aforementioned two quadratic forms

$$\left(\frac{s}{n}\right) = \frac{|\mathbf{w}^T\mathbf{b}|^2}{|\overline{\mathbf{w}^T\mathbf{x}}|^2} = \frac{\mathbf{w}^\dagger[\mathbf{b}^*\mathbf{b}^T]\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}} \tag{4.126}$$

The optimum weight vector (see Chapter 3) that yields the maximum SNR for (4.126) is

$$\mathbf{w}_{\text{opt}} = \frac{1}{(\text{constant})}\mathbf{R}_{nn}^{-1}\mathbf{b}^* \tag{4.127}$$

On comparing (4.127) with (4.45), it is seen that both the LMS and maximum SNR algorithms yield precisely the same weight vector solution (to within a multiplicative constant when the desired signal is absent) provided that $\mathbf{r}_{xd} = \mathbf{b}^*$, since these two vectors play exactly the same role in determining the optimum weight vector solution. Consequently, adopting a specific vector $\mathbf{r}_{xd}$ for the LMS algorithm is equivalent to selecting $\mathbf{b}^*$ for the maximum SNR algorithm, which represents direction of arrival information—this provides the relation between a reference signal and a beam steering signal for the LMS and maximum SNR algorithms to yield equivalent solutions.

From the foregoing discussion, it follows that the optimum orthonormal weight is

$$w'_{k_{\text{opt}}} = \left(\frac{1}{\mu_k}\right) b'^*_k \tag{4.128}$$

Substitute (4.123) and (4.128) into (4.122) results in

$$\tau_0\frac{dw'_k}{dt} + (1+\mu_k)w'_k = \mu_k w'_{k_{\text{opt}}} \tag{4.129}$$

For a step-function change in the input signal the solution may be written as follows:

$$w'_k(t) = \left[w'_k(0) - w'_k(\infty)\right]e^{-\alpha_k t} + w'_k(\infty) \tag{4.130}$$

where

$$w'_k(\infty) = \left(\frac{\mu_k}{1+\mu_k}\right) w'_{k_{\text{opt}}} \tag{4.131}$$

$$\alpha_k = \left(\frac{1+\mu_k}{\tau_0}\right) \tag{4.132}$$

In the foregoing equations $w'_k(\infty)$ represents the steady-state weight, $w'_k(0)$ is the initial weight value, and α_k is the transient decay factor. The adaptive weight transient responses can now be determined by the eigenvalues. The kth orthonormal servo loop may be represented by the simple type-0 position servo illustrated in Figure 4-13.

To relate the orthonormal system weights w'_k to the actual weights w_k note that the two systems shown in Figure 4-12 must be exactly equivalent so that

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}'^T\mathbf{x}' = \mathbf{w}'^T\mathbf{Q}\mathbf{x} \tag{4.133}$$

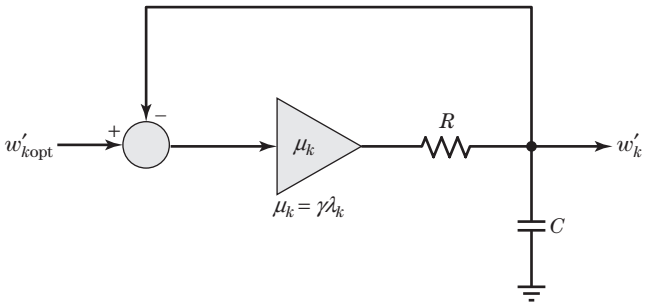


FIGURE 4-13 ■ Type-O servo model for *k*th orthonormal adaptive control loop. From Gabriel, *Proc. IEEE*, February 1976.

Consequently

$$\mathbf{w} = \mathbf{Q}^T \mathbf{w}' \tag{4.134}$$

From (4.134) it follows that the solution for the *k*th actual weight can be written as

$$w_k = \left(e_{1_k} w'_1 + e_{2_k} w'_2 + \cdots + e_{N_k} w'_N\right) \tag{4.135}$$

where e_{n_k} is the *k*th element of the *n*th eigenvector.

In the quiescent state, only receiver noise is present in any channel so that various channel signals are uncorrelated, and $\mathbf{Q}$ is an identity matrix, and there is no difference between $\mathbf{w}'$ and $\mathbf{w}$. With $\mathbf{Q}_q = \mathbf{I}$, the quiescent noise covariance matrix $\mathbf{R}_{nn_q}$ is already diagonalized, and if it is further assumed that the receiver noise power in all channels is equal (and denoted by $|\overline{n}_0|^2$), then from (4.113) it follows that

$$\mathbf{Q}_q^* \mathbf{R}_{nn_q} \mathbf{Q}_q^T = [\lambda_0 \delta_{ij}] \tag{4.136}$$

where

$$\lambda_0 = |\overline{n}_0|^2 \tag{4.137}$$

so the smallest eigenvalue is simply equal to the receiver channel noise power. This smallest eigenvalue then defines the minimum servo gain factor $\mu_{\min}$ as

$$\mu_{\min} = \gamma \lambda_0 \tag{4.138}$$

Since the quiescent steady-state weight $\mathbf{w}(\infty)$ must by definition be equal to $\mathbf{w}_q$, (4.131), (4.132), and (4.95) can be applied to yield

$$w_{q_k} = \frac{1}{1 + \mu_{\min}} b_k^* = \left(\frac{c_k}{1 + \mu_{\min}}\right) w_{q_k}$$

or

$$c_k = (1 + \mu_{\min}) \tag{4.139}$$

From (4.130)–(4.131) and (4.123), it follows that the effective time constant with which the *k*th component of $\mathbf{w}'$ converges to its optimum value is $\tau_0/(1 + \gamma\lambda_k)$. In effect, $\lambda_{\min}$ determines how rapidly the adaptive array follows changes in the noise environment. Equation (4.135) shows that each actual weight can be expressed as a weighted sum of exponentials, and the component that converges most slowly is the $\lambda_{\min}$ component.

When the adaptive array in Figure 4-11 operates with a distributed external noise field, the loop convergence is very slow for some angular noise distributions [23]. Furthermore, if γ is increased or τ_0 is decreased to speed the weight convergence, the loop becomes "noisy." Slow weight convergence occurs whenever trace($\mathbf{R}_{nn}$)/$\lambda_{\min}$ is large, and in these cases there is no choice of γ and τ_0 that yields rapid convergence without excessive loop noise. These facts suggest that the effects of noise on the solutions represented by (4.128)–(4.132) are important.

Griffiths provides a discrete form of the Howells–Applebaum weight update formula given by [24]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \gamma\left[\mu\mathbf{b}^* - \mathbf{x}^*(k)\mathbf{x}^\dagger(k)\mathbf{w}(k)\right] \tag{4.140}$$

where γ and μ are constants. The weights converge if γ is less than one over the largest eigenvalue. Compton shows that this is equivalent to [25]

$$0 < \gamma < \frac{1}{P_{IN}} \tag{4.141}$$

where P_{IN} is the total received power in (4.48). If γ is close to $1/P_{\text{IN}}$ then convergence is fast, but weight jitter is large. The weight jitter causes SNR fluctuations of several dB at steady state. If γ is small, then weight jitter is small, but the convergence is slow. A gain constant of [26]

$$\gamma = \frac{1}{2.5P_{IN}} \tag{4.142}$$

was found to provide a reasonably stable steady-state weights and rapid conversion.

4.3.2 Loop Noise Considerations

The random variations in the adaptive element weights of a Howells–Applebaum control loop result in an additional noise component in the array output signal. In this section expressions are given for the variance of the element weights and for the resulting additional noise in the array output [23].

Let $\boldsymbol{\xi}$ denote the noise component of the adaptive weight vector $\mathbf{w}$, and let $\mathfrak{N}$ denote the random component of $\mathbf{R}_{nn}$, so that

$$\mathbf{w} = \overline{\mathbf{w}} + \boldsymbol{\xi} \tag{4.143}$$

$$\mathbf{R}_{nn} = \overline{\mathbf{R}}_{nn} + \mathfrak{N} \tag{4.144}$$

where now $\overline{\mathbf{w}}$ and $\overline{\mathbf{R}}_{nn}$ denote average values. The adaptive weights must satisfy

$$\tau_0\frac{d\mathbf{w}}{dt} + (\mathbf{I} + \gamma\mathbf{R}_{nn})\mathbf{w} = \mathbf{b}^* \tag{4.145}$$

Substitute the values $\overline{\mathbf{w}}$ and $\overline{\mathbf{R}}_{nn}$ into (4.145) and subtract the result from the equation resulting with (4.143) and (4.144) substituted into (4.145) to give

$$\tau_0\frac{d\boldsymbol{\xi}}{dt} + \left(\mathbf{I} + \gamma\overline{\mathbf{R}}_{nn}\right)\boldsymbol{\xi} = -\gamma\mathfrak{N}\mathbf{w} \tag{4.146}$$

Premultiplying (4.146) by the transformation matrix $\mathbf{Q}^*$ and using the fact that $\mathbf{Q}^*\mathbf{Q}^T = \mathbf{I}$, then

$$\frac{d\boldsymbol{\zeta}}{dt} + \frac{1}{\tau_0}(\mathbf{I} + \gamma\boldsymbol{\Lambda})\boldsymbol{\zeta} = -\beta\mathbf{Q}^*\Re\mathbf{w} = \mathbf{u} \tag{4.147}$$

where

$$\boldsymbol{\zeta} = \mathbf{Q}^*\boldsymbol{\xi} \tag{4.148}$$

$$\beta = \frac{\gamma}{\tau_0} \tag{4.149}$$

Equation (4.147) represents a system of N independent linear differential equations of which the nth component can be written as

$$d\zeta_n + \sigma_n\zeta_n dt = u_n dt \tag{4.150}$$

where

$$\sigma_n = \frac{1 + \gamma\lambda_n}{\tau_0} \tag{4.151}$$

$$u_n = u_n(\tau, \overline{\mathbf{w}}, \boldsymbol{\zeta}) = (-\beta\mathbf{Q}^*\Re\mathbf{w})_n \tag{4.152}$$

Multiplying (4.150) by the factor $e^{\sigma_n t}$ and integrating each term from t_0 to t then yields

$$\begin{aligned}\zeta_n(t) = \zeta_n(t_0)\exp[-\sigma_n(t - t_0)] + \textstyle\int_{t_0}^{t} e^{-\sigma_n(r-\tau)} \\ \cdot u_n(\tau, \overline{\mathbf{w}}, \boldsymbol{\zeta})d\tau\end{aligned} \tag{4.153}$$

If only the steady-state case is considered, then the weights are near their mean steady-state values. The steady-state solution for variations in the element weights can be obtained from (4.153) by setting $t_0 = -\infty$ and ignoring any effect of the initial value $\zeta_n(t_0)$ to give

$$\zeta_n(t) = \int_0^\infty e^{-\sigma_n\tau}u_n(t - \tau)d\tau \tag{4.154}$$

One important measure of the noise present in the adaptive loops is the variance of the weight vector denoted by var($\mathbf{w}$):

$$\text{var}(\mathbf{w}) = E\left\{\sum_{n=1}^{N}|\mathbf{w}_n - \overline{\mathbf{w}}_n|^2\right\} = E\{\boldsymbol{\xi}^\dagger\boldsymbol{\xi}\} \tag{4.155}$$

where N is the dimension of the weight vector (or the number of degrees of freedom in the adaptive array system). Now since $\boldsymbol{\zeta} = \mathbf{Q}^*\boldsymbol{\xi}$, (4.155) becomes

$$\text{var}(\mathbf{w}) = E\{\boldsymbol{\zeta}^\dagger\boldsymbol{\zeta}\} \tag{4.156}$$

The elements of the covariance matrix of $\boldsymbol{\zeta}(t)$ in (4.156) are obtained from (4.154) and the definition of u_n

$$\begin{aligned}E\{\zeta_j^*\zeta_k\} = \beta^2\int_0^\infty d\tau_1\int_0^\infty E\{[\mathbf{Q}^*\Re(t - \tau_1)\overline{\mathbf{w}}]_j^* \\ \cdot \exp(-\sigma_j\tau_1 - \sigma_k\tau_2)[Q^*\Re(t - \tau_2)\overline{w}]_k\}\,d\tau_2 \\ + \beta^2\int_0^\infty d\tau_1\int_0^\infty E\{[\mathbf{Q}^*\Re(t - \tau_1)\boldsymbol{\xi}(t - \tau_1)]_j^* \\ \cdot \exp(-\sigma_j\tau_1 - \sigma_k\tau_2)[\mathbf{Q}^*\Re(t - \tau_2)\boldsymbol{\xi}(t - \tau_2)]_k\}\,d\tau_2\end{aligned} \tag{4.157}$$

where the cross-product terms do not appear since $E\{\boldsymbol{\xi}(t)\} = 0$ and $\Re(t)$ and $\boldsymbol{\xi}(t)$ are independent noise processes. A useful lower bound for (5.157) is given by [23].

$$\text{var}(\mathbf{w}) \geq \frac{\beta^2 \Delta}{2} \sum_{n=1}^{N} \frac{1}{\sigma_n} E\left\{ |(\mathbf{Q}^* \Re \overline{\mathbf{w}})_n|^2 \right\} \tag{4.158}$$

where Δ represents the time interval between successive independent samples of the input signal vector. For a pulse radar, Δ is approximately the same as the pulse width. For a communications system, Δ is approximately $1/B$, where B is the signal bandwidth.

The bound in (4.158) is useful in selecting parameter values for the Howells–Applebaum servo loops. If this bound is not small, then the noise fluctuations at the output of the adaptive loops are correspondingly large. For cases of practical interest [when var($\mathbf{w}$) is small compared with $\overline{\mathbf{w}}^\dagger \overline{\mathbf{w}}$], the right-hand side of (4.158) is an accurate estimate of var($\mathbf{w}$). Equation (4.158) simplifies (after considerable effort) to yield the expression

$$\text{var}(\mathbf{w}) \geq \left[\frac{\Re \beta \Delta}{2} - \frac{\beta \Delta}{2\gamma} \sum_{n=1}^{N} \frac{1}{\lambda_n + 1/\gamma} \right] \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \overline{\mathbf{w}} \tag{4.159}$$

where λ_n represents the nth eigenvalue of $\overline{\mathbf{R}}_{nn}$

$$\Re = \frac{N}{\beta} - \frac{1}{\gamma} \text{trace}(\boldsymbol{\mu}) \tag{4.160}$$

$$\boldsymbol{\mu} = (\beta \mathbf{H})^{-1} \tag{4.161}$$

$$\mathbf{H} = \overline{\Re}_{nn} \frac{\mathbf{I}}{\gamma} \tag{4.162}$$

Since the total output noise power is the noise power without noisy weights $\overline{\mathbf{w}}^\dagger \mathbf{R}_{nn} \overline{\mathbf{w}}$ plus the additional noise due to the random weight components, it can be shown that the total output noise power is given by

$$\begin{aligned} E\left\{ |\mathbf{w}^T \mathbf{x}|^2 \right\} &= \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \mathbf{w} + E\left\{ \boldsymbol{\xi}^\dagger \overline{\mathbf{R}}_{nn} \boldsymbol{\xi} \right\} \\ &\cong \overline{\mathbf{w}}^\dagger \overline{\mathbf{R}}_{nn} \overline{\mathbf{w}} \left[1 + \frac{\beta \Delta}{2} \sum_{n=1}^{N} \lambda_n \right] \end{aligned} \tag{4.163}$$

when $\gamma \lambda_n \gg 1$ for $n = 1, 2, \ldots, N$. The quantity $\beta \Delta$ occurs both in (4.163) and in (4.159) is the ratio $\gamma \Delta / \tau_0$, which is the gain divided by the loop time constant where the time constant is measured in intervals of the independent-sample rate of the system.

When loop noise is present in the system, the total noise power output increases by the factor $(1 + K_n)$, where from (4.163)

$$K_n \geq \frac{\gamma \Delta}{2\tau_0} \sum_{n=1}^{N} \lambda_n = \frac{\gamma}{4B\tau_0} \text{trace}(\overline{\mathbf{R}}_{nn}) \tag{4.164}$$

where $\Delta = 1/2B$ (i.e., B is the bandwidth of the input signal process), so that K_n is a direct measure of algorithm misadjustment due to noise in the weight vector. Recalling the solution to (4.129), we see that the effective time constant of the normal weight component $w'_{k'}$ having the slowest convergence rate is

$$\tau_{\text{eff}} = \frac{\tau_0}{1 + \gamma \lambda_{\min}} \cong \frac{\tau_0}{\gamma \lambda_{\min}} \tag{4.165}$$

where $\gamma\lambda_{\min} \geq 1$ to avoid a steady-state bias error in the solution. On combining (4.164) and (4.165) there results

$$\frac{\tau_{\text{eff}}}{\Delta} \geq \frac{1}{2K_n\lambda_{\min}} \sum_{n=1}^{N} \lambda_n = \frac{\text{trace}(\overline{\mathbf{R}}_{nn})}{2K_n\lambda_{\min}} \tag{4.166}$$

Equation (4.166) shows that, when the smallest eigenvalue $\lambda_{\min}$ is small compared with trace($\overline{\mathbf{R}}_{nn}$), many independent samples of the input signal are required before the adaptive array settles to a near-optimum set of weights without excessive loop noise; no set of loop parameters yields both low loop noise and rapid convergence in this case. Berni [27] gives an analysis of steady-state weight jitter in Howells–Applebaum control loops when there is no statistical independence between the input signal and weight processes. Steady-state weight jitter is closely related to the statistical dependence between the weight and signal processes.

4.3.3 Adaptive Array Behavior in Terms of Eigenvector Beams

The **Q**-matrix transformation defined by (4.112) is composed of normalized and mutually orthogonal eigenvectors. The components of these eigenvectors are interpreted as array element weights, associated with normalized orthogonal eigenvector beams. The kth eigenvector beam is expressed as

$$g_k(\theta) = (\mathbf{s}^T\mathbf{e}_k) = \sum_{i=1}^{N} e_{ki}s_i \tag{4.167}$$

where **s** and its components s_i for a linear N-element array are defined by

$$\mathbf{s}^T = [s_2, s_2, \ldots, s_N] \tag{4.168}$$

$$s_i = e^{j\psi(2i-N-1)/2} \tag{4.169}$$

By defining the variable z related to the spatial angle θ as

$$z \triangleq e^{j\psi} \tag{4.170}$$

then the eigenvector beam may be conveniently rewritten as

$$g_k(\theta) = \left(\frac{1}{\sqrt{z}}\right)^{N-1} \left[e_{k1} + e_{k2}z + e_{k3}z^2 + \cdots + e_{kN}z^{N-1}\right] \tag{4.171}$$

As shown in Chapter 2, the array polynomial can also be expressed in the factored form

$$g_k(\theta) = \left(\frac{1}{\sqrt{z}}\right)^{N-1} [a_{N-1}(z - z_1)(z - z_2)\cdots(z - z_{N-1})] \tag{4.172}$$

where the roots $z_1, z_2, \ldots, z_{N-1}$ are nulls in the eigenvector beam pattern.

If one narrowband interference source is at θ_1, then $\mathbf{R}_{nn}$ contains one unique eigenvalue and the corresponding unique eigenvector that produces a retrodirective eigenvector beam centered on the source at θ_1 as illustrated in Figure 4-14. The $\mathbf{R}_{nn}$ matrix in this case also contains nonunique eigenvalues having arbitrary nonunique eigenvectors; these arbitrary nonunique eigenvector beams are not essential to array operation, and array pattern performance is characterized solely in terms of the unique retrodirective eigenvector beams.

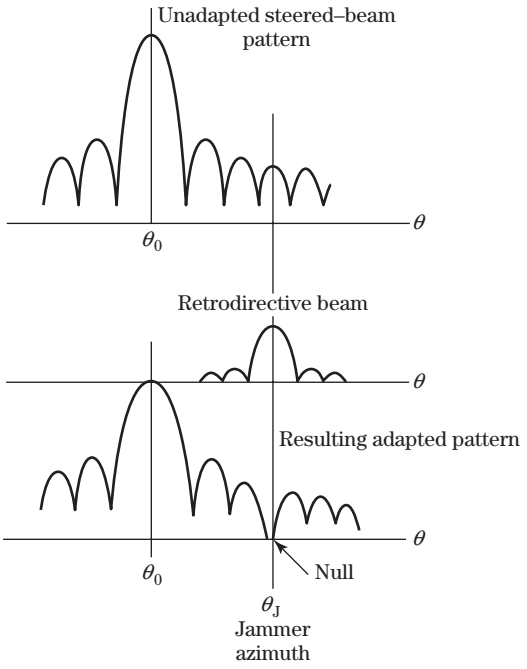


FIGURE 4-14 ■ Retrodirective beam principle illustrating subtraction of retrodirective beam from unadapted pattern to obtain adapted pattern with one interference source. From Gabriel, *Proc. IEEE,* February 1976.

The overall array beam pattern is most easily derived by considering the output of the orthonormal system represented in Figure 4-12b for the input signal vector $\mathbf{s}$, defined in (4.169). Since the output for the real orthonormal systems are identical, it follows that

$$AF(\theta, t) = \sum_{i=1}^{N} w_i s_i = \sum_{i=1}^{N} w_i' s_i' = w'^T \mathbf{s}' \tag{4.173}$$

where

$$\mathbf{s}' = \mathbf{Q}\mathbf{s} \tag{4.174}$$

Now the ith component of $\mathbf{s}'$ is given by

$$s_i' = \left(\mathbf{e}_i^T \mathbf{s}\right) = \sum_{k=1}^{N} e_{ik} s_k \tag{4.175}$$

but this summation defines the ith eigenvector beam [as can be seen from (4.167)], so that

$$s_i' = \left(\mathbf{e}_i^T \mathbf{s}\right) = g_i(\theta) \tag{4.176}$$

Consequently, the overall array factor can be expressed as

$$AF(\theta, t) = \sum_{i=1}^{N} w_i' g_i(\theta) \tag{4.177}$$

which shows that the output array factor is the summation of the N eigenvector beams weighted by the orthonormal system adaptive weights.

Since the kth component of the quiescent orthonormal weight vector is given by

$$w_{q_k}' = \left(\mathbf{e}_k^{\dagger} \mathbf{w}_q\right) \tag{4.178}$$

the steady-state solution for the kth component of the orthonormal weight vector given by (4.131) can be rewritten using (4.132), (4.95), and (4.139) to yield

$$w'_k(\infty) = \left(\frac{1+\mu_{\min}}{1+\mu_k}\right) w'_{q_k} \tag{4.179}$$

Assume as before that quiescent signal conditions up to time $t = 0$ consist only of receiver noise and that the external interference sources are switched on at $t = 0$; then

$$w'_k(0) = w'_{q_k} \tag{4.180}$$

and the solution for w_k expressed by (4.131) is rewritten in the more convenient form

$$w'_k = w'_{q_k} - (1 - e^{-\alpha_k t}) \left[\frac{\mu_k - \mu_{\min}}{1+\mu_k}\right] w'_{q_k} \tag{4.181}$$

It is immediately apparent that at time $t = 0$ (4.177) results in

$$AF(\theta, 0) = \sum_{i=1}^{N} w'_{q_i} g_i(\theta) = \left(\mathbf{w}_q'^T \mathbf{s}'\right) = \left(\mathbf{w}_q'^T \mathbf{Q}\mathbf{s}\right) \tag{4.182}$$

From (4.134) it is seen that $\mathbf{w}_q^T = \mathbf{w}_q'^T \mathbf{Q}$ so that

$$AF(\theta, 0) = \left(\mathbf{w}_q^T \mathbf{s}\right) = AF_q(\theta) \tag{4.183}$$

where the quiescent pattern $AF_q(\theta)$ was previously defined by (4.90).

Finally, by substituting (4.183) and (4.181) into (4.177), there results

$$AF(\theta, t) = AF_q(\theta) - \sum_{i=1}^{N} (1 - e^{-\alpha_i t}) \left[\frac{\mu_i - \mu_{\min}}{1+\mu_i}\right] w'_{q_i} g_i(\theta) \tag{4.184}$$

where it will be recalled that

$$\alpha_i = \frac{1+\mu_i}{\tau_0}$$
$$\mu_i = \gamma \lambda_i$$
$$\mu_{\min} = \gamma \lambda_{\min}$$

The foregoing result emphasizes that the adaptive array factor consists of two parts:

1. The quiescent beam pattern $AF_q(\theta)$
2. The summation of weighted orthogonal eigenvector beams that is subtracted from $AF_q(\theta)$

Note also from (4.184) that the weighting associated with any eigenvector beams corresponding to eigenvalues equal to λ_0 (the quiescent eigenvalue) is zero since the numerator $(\mu_i - \mu_{\min})$ is zero for such eigenvalues. Consequently, any eigenvector beams associated with λ_0 is disregarded, leaving only unique eigenvector beams to influence the resulting pattern. The transient response time of (4.184) is determined by the value of α_i, which in turn is proportional to the eigenvalue. Therefore, a large eigenvalue yields a fast transient response for its associated eigenvector beam, whereas a small eigenvalue results in a slow transient response.

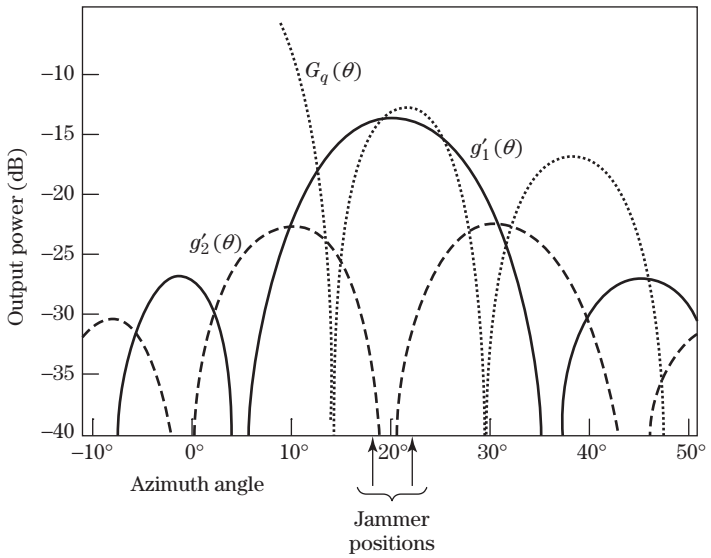


FIGURE 4-15 ■ Two eigenvector beam patterns $g_1'(\theta)$ and $g_2'(\theta)$ for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

The foregoing eigenvector beam interpretation of adaptive array behavior is illustrated by considering an eight-element linear array having $\lambda/4$ element spacing and two narrowband interference sources having nearly equal power ratios of $PR_1 = 1{,}250$ and $PR_2 = 1{,}200$ located at $\theta_1 = 18^\circ$ and $\theta_2 = 22^\circ$, respectively. Forming the covariance matrix using (4.106) for this case and solving for the eigenvalues yields two unique solutions: $\lambda_1 = 18{,}544.4$ and $\lambda_2 = 1{,}057.58$. These widely different eigenvalues result despite the nearly equal jammer powers, because the interference sources are close together compared with the array quiescent beamwidth.

Solving for the two (normalized) eigenvectors associated with the unique eigenvalues then permits the two eigenvector beam patterns to be found $g_1'(\theta)$ and $g_2'(\theta)$, which are both illustrated in Figure 4-15. Beam $g_1'(\theta)$ covers both interference sources in the same manner as a centered beam pattern, and its total output power is equal to the first eigenvalue

$$\left(\frac{\lambda_1}{\lambda_0}\right) = 1 + P_1 g_1'^2(\theta_1) + P_2 g_2'^2(\theta_2) = 18{,}544 \tag{4.185}$$

The second eigenvector beam $g_2'(\theta)$ splits the interference sources in the manner of a difference beam, and its total output power is equal to the second eigenvalue

$$\left(\frac{\lambda_2}{\lambda_0}\right) = 1 + P_1 g_2'^2(\theta_1) + P_2 g_2'^2(\theta_2) = 1057 \tag{4.186}$$

Although both eigenvector beams contain power from both sources, their respective output signals are decorrelated. The cross-correlation product of the two eigenvector beam outputs is

$$E\{x_1'^* x_2'\} = E\{|J_1|^2\} g_1'(\theta_1) g_2'(\theta_1) + E\{|J_2|^2\} g_1'(\theta_2) g_2'(\theta_2) \tag{4.187}$$

This cross-correlation product can be zero if the product $[g_1'(\theta) g_2'(\theta)]$ is positive when $\theta = \theta_1$ and negative when $\theta = \theta_2$, thereby resulting in decorrelation between the two eigenvector beam signals. Figure 4-16 shows the overall quiescent beam pattern and the resulting steady-state adapted pattern for this two-source example. Figure 4-17 illustrates the transient response (in terms of increase in output noise power) of the adaptive array for

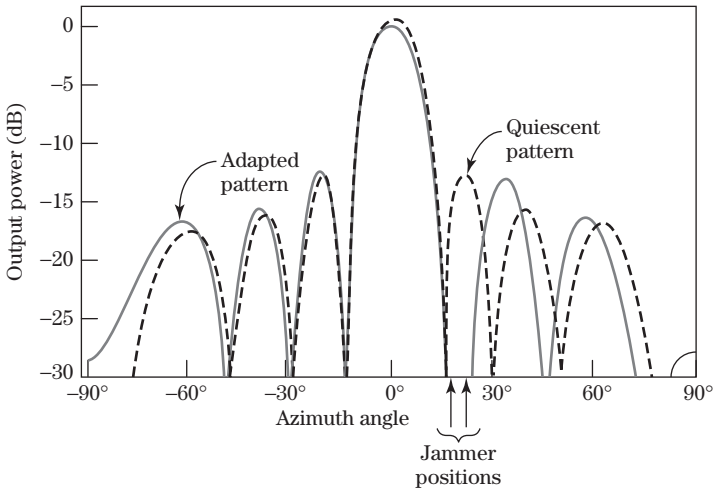


FIGURE 4-16 ■ Steady-state adapted array pattern and quiescent array pattern for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

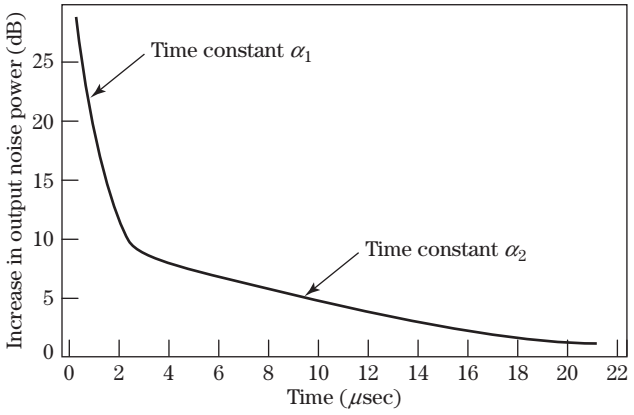


FIGURE 4-17 ■ Array transient response for two-jammer example. From Gabriel, *Proc. IEEE*, February 1976.

this two-interference source example, where it is seen that the response has two distinct slopes associated with the two distinct (and widely different) eigenvalues.

4.3.4 Example of *N*-Element Linear Adaptive Array Performance

This section presents results from an eight-element linear array with Howells–Applebaum loops.

4.3.4.1 Assumptions and Initial Conditions

Only the quiescent receiver noise is present in each channel up to time $t = 0$, when all the external interference sources are switched on in a single step function. The quiescent RMS noise voltage in all channels equals the constant n_0, such that the quiescent eigenvalue λ_0 is $|n_0|^2$. This defines the quiescent servo gain factor $\mu_{\min}$ by way of (4.138). For the configuration of Figure 4-11, it is convenient to choose $\mu_{\min} = 1$, so the amplifier gains G are set accordingly. Once $\mu_{\min}$ is selected, it is convenient to express μ_i from (4.123) as a ratio of eigenvalues,

$$\frac{\mu_i}{\mu_{\min}} = \frac{\lambda_i}{\lambda_0} \tag{4.188}$$

The quiescent steered-beam pattern $AF_q(\theta)$ and its associated quiescent weight vector $\mathbf{w}_q$ are given by (4.87)–(4.90). The eight-element linear array has an element spacing $\lambda/2$, $\mu = \pi/2 \sin\theta$, and $a_k = 1$. The quiescent weights and array factor are given by

$$w_{q_k} = e^{-j\psi_0(2k-9)/2} \tag{4.189}$$

$$AF_q(\theta) = \frac{\sin[8(\psi - \psi_0)/2]}{\sin[(\psi - \psi_0)/2]} \tag{4.190}$$

The coefficients of the input beam steering vector $\mathbf{b}^*$ are found from (4.140) and (4.88)

$$c_k = (1 + \mu_{\min}) = 2 \tag{4.191}$$

$$b_k^* = c_k w_{q_k} = 2e^{-j\psi_0(2k-9)/2} \tag{4.192}$$

The maximum power condition for each of the orthonormal loops of Figure 4-12b is

$$\mu_{\max} = \mu_{\min}\left(\frac{\lambda_{\max}}{\lambda_0}\right) = \left(\frac{\pi B_c \tau_0}{10}\right) - 1 \tag{4.193}$$

where $\lambda_{\max}$ represents the maximum eigenvalue. The channel bandwidth B_c and filter time constant τ_0 are the same for all element channel servo loops. Solving for τ_0 from (4.193) yields

$$\tau_0 = \left(\frac{10}{\pi B_c}\right)\left[1 + \mu_{\min}\left(\frac{\lambda_{\max}}{\lambda_0}\right)\right] = \left(\frac{10}{\pi B_c}\right)\left[1 + \mu_{\min} + \mu_{\min}\sum_{r=1}^{R} P_r g_m^2(\theta_r)\right] \tag{4.194}$$

The maximum power (maximum eigenvalue) is much larger than the jammer-to-receiver-noise power ratios, because the various P_r are multiplied by the power gain of the eigenvector beams.

4.3.4.2 Output Noise Power and SNR Degradation

The output SNR of the adaptive array compared with the output SNR of a conventional array under the same interference conditions is the performance characteristic of ultimate interest. Instead of forming the actual SNR, it is sufficient to consider the output noise power by itself to illustrate the system transient behavior. Since the receiver noise and external interference sources are statistically independent, the total output noise power is the sum of the two separate output noise powers.

The receiver noise output power can be expressed as

$$|y_{0_n}(t)|^2 = \sum_{k=1}^{N} |w_k n_k|^2 = \sum_{i=1}^{N} |w_i' n_0|^2 \tag{4.195}$$

Substituting for w_i' from (4.181) then yields

$$|y_{0_n}(t)|^2 = |\overline{n}_0|^2 \sum_{i=1}^{k} [1 - A_i(t)]^2 |w_{q_i}'|^2 \tag{4.196}$$

where

$$A_i(t) = (1 - e^{-\alpha_i t})\left[\frac{\mu_i - \mu_0}{1 + \mu_i}\right] \tag{4.197}$$

From (4.197) it is seen that $A_i(t)$ is zero for $t = 0$ and for $\mu_i = \mu_0$ (for nonunique eigenvalues). Therefore, for quiescent conditions at $t = 0$, it follows that

$$|y_{0_n}(0)|^2 = |\overline{n}_0|^2 \sum_{i=1}^{N} |w'_{q_i}|^2 = |\overline{n}_0|^2 \sum_{k=1}^{N} |w_{q_k}|^2 \tag{4.198}$$

since the output noise power must be the same for either the real system or the equivalent orthonormal system. Consequently, (4.196) is rewritten as

$$|y_{0_n}(t)|^2 = |\overline{n}_0|^2 \sum_{k=1}^{N} |w_{q_k}|^2 - \sum_{i=1}^{N} [2 - A_i(t)] A_i(t) |w'_{q_i}|^2 \tag{4.199}$$

Equation (4.199) is a particularly convenient form because the w'_{q_i} associated with nonunique eigenvalues need not be evaluated since $A(t) = 0$ for such eigenvalues.

The output noise power contributed by R external interference sources is given by the sum of their output power pattern levels:

$$|y_{0_j}(t)|^2 = |\overline{n}_0|^2 \sum_{r=1}^{R} P_r AF^2(\theta_r, t) \tag{4.200}$$

where P_r is the rth source power ratio, θ_r is its angular location, and $AF(\theta_r, t)$ is given by (4.184).

The total output noise power is the sum of (4.199) and (4.200), and the increase in the output noise power (with interference sources turned on) is this sum over the quiescent noise (4.198).

$$\frac{|y_0(t)|^2}{|y_{0_n}(0)|^2} = 1 + \left\{ \frac{\sum_{r=1}^{R} P_r AF^2(\theta_r, t) - \sum_{i=1}^{N} [2 - A_i(t)] A_i(t) |w'_{q_i}|^2}{\sum_{k=1}^{N} |w_{q_k}|^2} \right\} \tag{4.201}$$

The output noise power increase in (4.201) indicates the system transient behavior. An increase in output noise power indicates the general magnitude of the adapted (steady-state) weights.

The degradation in the SNR, D_{sn}, enables one to normalize the effect of adapted-weight magnitude level. This degradation is the quiescent SNR divided by the adapted SNR.

$$D_{sn} = \left(\frac{AF_q^2(\theta_s)}{AF^2(\theta_s, t)} \right) \left(\frac{|y_0(t)|^2}{|y_{0_n}(0)|^2} \right) \tag{4.202}$$

where the ratio in the second factor is just (4.201), the increase in output noise power.

4.3.4.3 Eigenvalues and Eigenvectors of the Noise Covariance Matrix

For computational convenience, receiver noise is unity and all noise powers expressed as ratios to receiver noise power. Adopting this convention, the quiescent noise matrix $\mathbf{R}_{nn_q}$ is an identity matrix, and with R narrowband interference sources the noise covariance

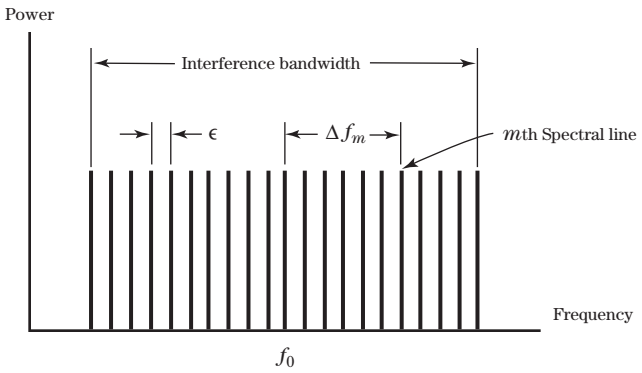


FIGURE 4-18 ■ Interference source power spectrum having uniform amplitude spectral lines spaced ε apart. From Gabriel, *Proc. IEEE,* February 1976.

matrix becomes

$$\mathbf{R}_{nn} = \mathbf{I} + \sum_{r=1}^{R} P_r \mathbf{M}_r \tag{4.203}$$

where $\mathbf{M}_r$ now represents the covariance matrix due to the rth interference source.

Wideband interference sources are represented by dividing the jammer power spectrum into a series of discrete spectral lines. A uniform amplitude spectrum of uncorrelated lines spaced apart by a constant frequency increment ε is once again assumed as illustrated in Figure 4-18. If P_r is the power ratio of the entire jammer power spectrum, then the power ratio of a single spectral line (assuming a total of L_r spectral lines) is

$$P_{rl} = \left(\frac{P_r}{L_r}\right) \tag{4.204}$$

Furthermore, if $B_r (B_r <$ element channel receiver bandwidth, $B_c)$ denotes the percent bandwidth of the jamming spectrum, then the frequency offset of the lth spectral line is

$$\frac{\Delta f_l}{f_0} = \left(\frac{B_r}{100}\right)\left[-\frac{1}{2} + \left(\frac{l-1}{L_r - 1}\right)\right] \tag{4.205}$$

The covariance matrix with R broadband interference sources is written as

$$\mathbf{R}_{nn} = \mathbf{I} + \sum_{r=1}^{R} \sum_{l=1}^{L_r} P_{rl} \mathbf{M}_{rl} \tag{4.206}$$

The mnth component (mth row and nth column) of the matrix $\mathbf{M}_{rl}$ is in turn given by

$$(\mathbf{M}_{rl})_{mn} = e^{j\psi_{rl}(n-m)} \tag{4.207}$$

where

$$\psi_{rl} = \left(\frac{f_l}{f_0}\right)\pi \sin\theta_r = \left(1 + \frac{\Delta f_l}{f_0}\right)\pi \sin\theta_r \tag{4.208}$$

4.3.4.4 Performance Characteristics for Various Signal Conditions

Four narrowband sources located in the sidelobe region of the quiescent beam pattern yield four distinct eigenvalues and require four degrees of freedom to provide the eigenvector

beams required to place nulls at the jammer locations. If the adaptive weight adjustments are large, there may be appreciable main beam distortion in the overall adapted pattern.

The Howells–Applebaum adaptive loop has one adaptive weight in each element channel of the array; this configuration works interference sources with a bandwidth of up to about 20%. Gabriel [22] gives two examples as follows: a 2% bandwidth source in the sidelobe region for which two degrees of freedom (two pattern nulls) are required to provide proper cancellation; and a 15% bandwidth source in the sidelobe region for which three degrees of freedom are required. Broadband interference sources require a transversal equalizer in each element channel (instead of a single adaptive weight) for proper compensation, with a Howells–Applebaum adaptive loop then required for every tap appearing in the tapped delay line.

The adapted pattern for main beam nulling exhibits severe distortions. For interference sources located in the main beam, the increase in output noise power is an unsatisfactory indication of array performance, because there is a net SNR degradation due to the resulting main beam distortion in the adapted pattern. Main beam constraints for such cases can be introduced.

4.3.5 Hard Limiter Modification for N Adaptive Loops

It was shown in Section 4.3.2 that the adaptive array performance depends on the external noise field as well as on the parameters of the adaptive control loops. The power level and angular location of the external noise field determine the noise covariance matrix and therefore its eigenvalues. The eigenvalues, in turn, directly affect the array performance, since both the transient response of the adaptive array and the control loop noise depend explicitly on these eigenvalues. For a nonstationary signal environment, wide variations in array performance may occur, ranging from excessive control loop noise (when the interference is strong) to very slow convergence (when the interference is weak). Introducing a hard limiter into the adaptive control loop reduces the effects of varying noise intensity, and the dynamic range of signals in the control loops are reduced without degrading array performance [28]. Figure 4-19 shows a six-element linear array with a hard limiter introduced in the conjugate signal branches.

With the signal envelopes hard limited, the input to the correlation mixers changes from x_k^* to $u_k^* = x_k^*/|x_k|$. This normalization removes amplitude variations in the conjugate signals but retains the phase variations. The correlation mixer voltage v_k is now given by

$$v_k' = k^2 \left(u_k^* \sum_{i=1}^{N} w_i x_i \right) \tag{4.209}$$

On comparing (4.209) with (4.97), it is seen that u_k^* has simply replaced x_k^*, so the resulting adaptive weight matrix differential equation now becomes

$$\tau_0 \frac{d\mathbf{w}}{dt} + [\mathbf{I} + \gamma' \mathbf{M}]\mathbf{w} = \mathbf{b}^* \tag{4.210}$$

which is analogous to (4.109) with $\mathbf{M}$ replacing $\mathbf{R}_{nn}$ and $\gamma' = k^2 G'$ where $\mathbf{M}$ is the modified noise covariance with envelope limiting having elements given by

$$M_{ml} = E\left\{ \frac{x_m^* x_l}{|x_m|} \right\} \tag{4.211}$$

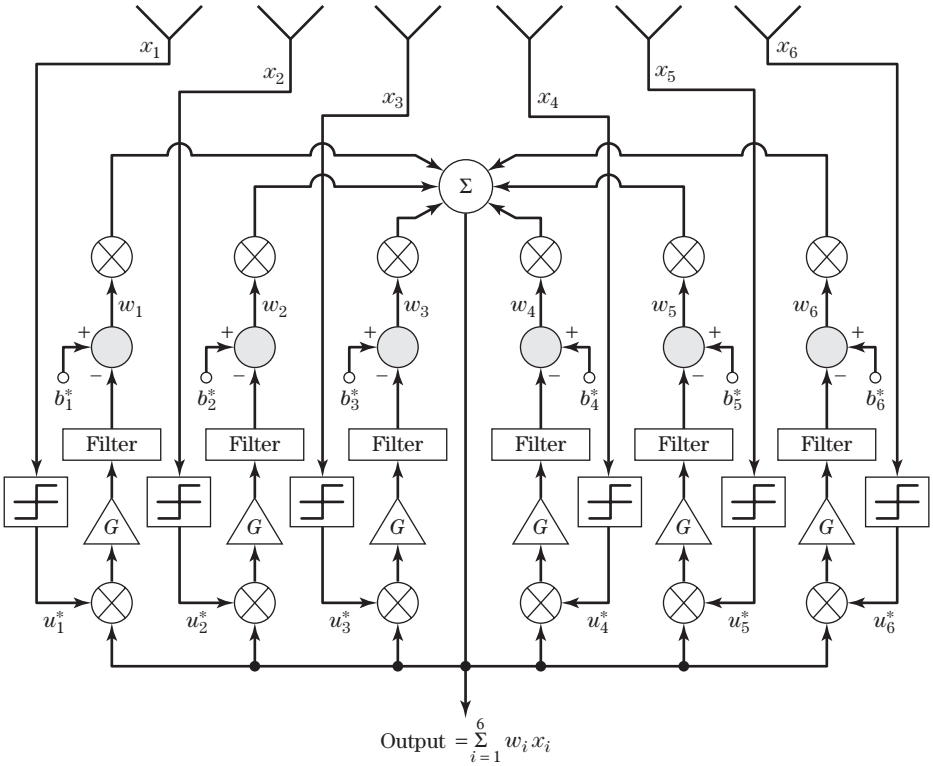


FIGURE 4-19 ■ Hard limiter modification of linear six-element adaptive array system. From Gabriel, *Proc. IEEE*, February 1976.

Assuming the quadrature components of each signal x_k are zero-mean Gaussian random variables having variance σ^2, we can then compute the elements of the covariance matrix $\mathbf{M}$ directly from the elements of $\mathbf{R}_{nn}$ by using the relation [28]

$$M_{ml} = \sqrt{\frac{\pi}{8}} \frac{1}{\sigma} (\mathbf{R}_{nn})_{ml} \tag{4.212}$$

It follows that the elements of $\mathbf{M}$ differ from the elements of $\mathbf{R}_{nn}$ by a common factor $(1/\sigma)\sqrt{(\pi/8)}$. Consequently, the effective time constants that determine the rate of convergence and control loop noise are changed by this same factor, thereby reducing the dependence of array performance on the strength of the external noise field.

It is worthwhile noting that limiting does not change the relative values of the signal covariance matrix elements or the relative eigenvalue magnitudes presuming identical channels. Thus, for widely different eigenvalues, limiting does reduce the eigenvalue spread to provide rapid transient response and low control loop noise. Nevertheless, limiting always reduces the dynamic range of signals in the control loops, thereby simplifying the loop implementation.

4.4 | INTRODUCTION OF MAIN BEAM CONSTRAINTS

As a result of introducing beam pattern nulls, the main beam may become distorted, resulting in a degradation in SNR performance. To prevent such array performance degradation from occurring, it is possible to introduce constraints so the adaptive processor maintains

desired mainlobe signals while realizing good cancellation of interference in the sidelobes. The constraint methods discussed here follow the development that is given by Applebaum and Chapman [29].

Techniques for applying main beam constraints to limit severe array pattern degradation include the following:

1. Time domain: The array adapts when the desired signal is not present in the main beam. These weights are kept until the next adaptation or sampling period. This approach does not protect against main beam distortion resulting from main beam jamming and is also vulnerable to blinking jammers.
2. Frequency domain: When the interference sources have much wider bandwidths than the desired signal, the adaptive processor is constrained to adapt to signals only outside the desired signal bandwidth. This approach somewhat degrades the cancellation capability and distorts the array factor.
3. Angle domain: Three angle domain techniques provide main beam constraints in the steady state: (a) pilot signals; (b) preadaptation spatial filters; and (c) control loop spatial filters. These techniques are also helpful for constraining the array response to short duration signals, since they slow down the transient response to main beam signals. The angle domain techniques provide the capability of introducing main beam constraints into the adaptive processor response.

4.4.1 Pilot Signals

To illustrate the use of pilot signal techniques, consider the multiple sidelobe canceller (MSLC) adaptive array configuration shown in Figure 4-20, where an integrator with feedback structure is taken to represent the integrating filter in the Howells–Applebaum

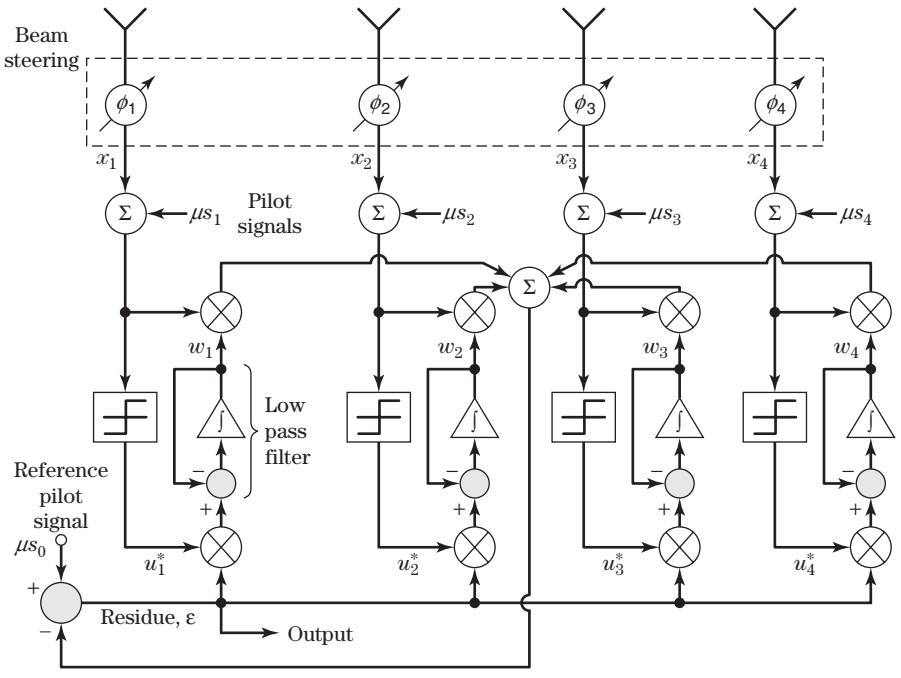


FIGURE 4-20 ■ Multiple sidelobe canceller (MSLC) adaptive array configuration with beam steering pilot signals and main beam control.

adaptive loop. The "pilot signals" shape the array beam and maintain the main beam gain (avoiding SNR degradation). The pilot signals are continuous wave (CW) tones injected into each element channel at a frequency that is easily filtered out of the signal bandwidth. It is not necessary to use the beam steering phase shifters shown in Figure 4-20, since if they are not present the pilot signals may be injected with the proper phase relationship corresponding to the desired main beam direction instead of in phase with each other as shown. The amplitudes and phases of the injected pilot signals $s_1, \ldots, s_4$ may be represented by the vector $\mu\mathbf{s}$, where $\mathbf{s}$ has unit length, and μ is a scalar amplitude factor. The reference channel (or main beam) signal is represented by the injected pilot signal s_0.

For the adaptive control loops shown in Figure 4-20, it follows that the vector differential equation for the weight vector is written as

$$\frac{d\mathbf{w}}{dt} = \mathbf{u}^*(t)\varepsilon(t) - \mathbf{w}(t) \tag{4.213}$$

Since $\varepsilon = \mu s_0 - \mathbf{x}^T\mathbf{w}$, it follows (4.213) and the results of Section 4.3.5 that

$$\frac{d\mathbf{w}}{dt} = g\mu\mathbf{r}_{xs_0} - [\mathbf{I} + g\mathbf{R}_{xx}]\mathbf{w} \tag{4.214}$$

where g is a gain factor representing the correlation mixer gain and the effect of the limiter. The steady-state solution of (4.214) is given by

$$\mathbf{w}_{ss} = [\mathbf{I} + g\mathbf{R}_{xx}]^{-1} g\mu\mathbf{r}_{xs_0} \tag{4.215}$$

In the absence of any desired signal, then

$$\mathbf{x} = \mathbf{n} + \mu\mathbf{s} \tag{4.216}$$

where $\mathbf{n}$ is the noise signal vector, and $\mu\mathbf{s}$ is the injected pilot signal vector. Consequently,

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^*\mathbf{x}^T\} = \mathbf{R}_{nn} + \mu\mathbf{s}^*\mathbf{s}^T \tag{4.217}$$

$$\mathbf{r}_{xs_0} \triangleq E\{\mathbf{x}^* s_0\} = \mu\mathbf{s}^* s_0 \tag{4.218}$$

On substituting (4.217) and (4.218) into (4.215) it can be shown that

$$\mathbf{w}_{ss} = \frac{\mathbf{K}^{-1}\mathbf{s}^* g\mu^2 s_0}{1 + g\mu^2\mathbf{s}^T\mathbf{K}^{-1}\mathbf{s}^*} \tag{4.219}$$

where $\mathbf{K} = \mathbf{I} + g\mathbf{R}_{nn}$. Substituting the expression for $\mathbf{K}^{-1}$ in (4.219) then yields

$$\mathbf{w}_{ss} = \frac{(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^* g\mu^2 s_0}{1 + g\mu^2\mathbf{s}^T(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*} \tag{4.220}$$

For large pilot signals $\mu^2 \to \infty$, and (4.220) becomes

$$\mathbf{w}_{ss} \cong \left[\frac{(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*}{\mathbf{s}^T(\mathbf{I} + g\mathbf{R}_{nn})^{-1}\mathbf{s}^*}\right] s_0 \tag{4.221}$$

If $\mathbf{s}$ has equal amplitude components, then the main beam response from (4.221) is

$$\mathbf{s}^T\mathbf{w}_{ss} \cong s_0 \tag{4.222}$$

which is a constant, independent of $\mathbf{R}_{nn}$ (and hence independent of any received waveforms).

The array configuration of Figure 4-20 uses one set of pilot signals for a single main beam constraint. Multiple constraints require multiple sets of pilot signals, with each set at a different frequency. Pilot signals are inserted close to the input of each element channel to compensate for any amplitude and phase errors. Strong pilot signals require channel elements with a large dynamic range, so they must be filtered to avoid interfering with the desired signal.

4.4.2 Preadaption Spatial Filters

Preadaption spatial filtering forms two beams following the beam steering phase shifters (Figure 4-21). The quiescent pattern main beam is formed with fixed weights $\mathbf{s}^*$ (forming a unit length weight vector). The second beam (termed a *cancellation beam*) is formed adaptively by an MSLC whose input channels are obtained from spatial filtering represented by the matrix transformation $\mathbf{A}$. $\mathbf{A}$ has one less output channel than the number of sensor elements and is selected to maintain a constant response in the main beam direction so that $\mathbf{As} = 0$.

From the signal vector definitions of Figure 4-21 it follows that

$$e_m = \mathbf{s}^\dagger \mathbf{x} = \mathbf{x}^T \mathbf{s}^* \tag{4.223}$$

$$e_0 = e_m - \mathbf{y}^T \mathbf{u} \tag{4.224}$$

$$\mathbf{u} = \mathbf{Ax} \tag{4.225}$$

Therefore

$$e_0 = \mathbf{s}^\dagger \mathbf{x} - \mathbf{y}^T \mathbf{Ax} = (\mathbf{s}^* - \mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{w}^T \mathbf{x} \tag{4.226}$$

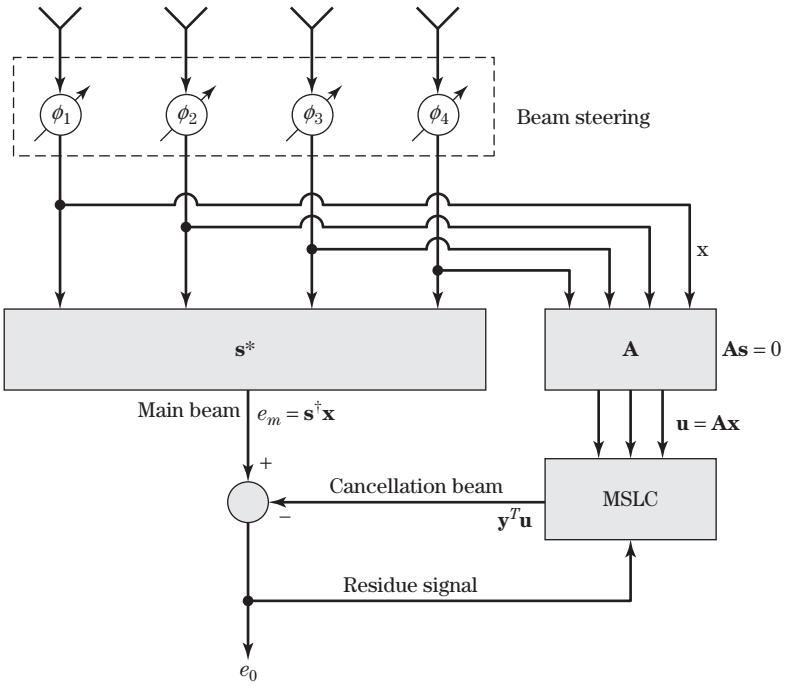


FIGURE 4-21 ■ General structure of preadaption spatial filtering.

The composite weight vector for the entire system can therefore be written as

$$\mathbf{w} = \mathbf{s}^* - \mathbf{A}^T\mathbf{y} \tag{4.227}$$

Consequently

$$\mathbf{s}^T\mathbf{w} = \mathbf{s}^T(\mathbf{s}^* - \mathbf{A}^T\mathbf{y}) = \|\mathbf{s}\|^2 - \mathbf{s}^T\mathbf{A}^T\mathbf{y} \tag{4.228}$$

Since $\mathbf{A}$ was selected so that $\mathbf{As} = 0$, it follows that

$$\mathbf{s}^T\mathbf{w} = \|\mathbf{s}\|^2 = 1 \tag{4.229}$$

Denote the covariance matrix associated with $\mathbf{u}$ by

$$\mathbf{R}_{uu} \triangleq E\{\mathbf{u}^*\mathbf{u}^T\} = E\{\mathbf{A}^*\mathbf{x}^*\mathbf{x}^T\mathbf{A}^T\} = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T \tag{4.230}$$

The MSLC unit generates a weight vector $\mathbf{y}$ that satisfies (4.213).

$$\frac{d\mathbf{y}}{dt} = g\mathbf{u}^*e_0 - \mathbf{y} \tag{4.231}$$

In the steady state [where $(d\mathbf{y}/dt) = 0$] it follows that

$$[\mathbf{I} + g\mathbf{R}_{uu}]\mathbf{y} = gE\{\mathbf{u}^*e_m\} \tag{4.232}$$

where g is a gain factor, and the right side of (4.232) represents the cross-correlation vector of e_m with each component of $\mathbf{u}$. Using (4.223), (4.225), and (4.230) in (4.232), we find that

$$\left(\mathbf{I} + g\mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T\right)\mathbf{y} = g\mathbf{A}^*\mathbf{R}_{xx}\mathbf{s}^* \tag{4.233}$$

Premultiply (4.233) by $\mathbf{A}^T$ and use (4.227); it then follows that the composite weight applied to the input signal vector satisfies the steady-state relation

$$\left(\mathbf{I} + g\mathbf{A}^T\mathbf{A}^*\mathbf{R}_{xx}\right)\mathbf{w}_{ss} = \mathbf{s}^* \tag{4.234}$$

when $\mathbf{x}$ does not contain a desired signal component, then $\mathbf{R}_{xx}$ may be replaced by $\mathbf{R}_{nn}$.

Now allow g to become very large so that (4.233) yields

$$\mathbf{A}^*\mathbf{R}_{xx}\mathbf{A}^T\mathbf{y} = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{s}^* \tag{4.235}$$

$$\mathbf{A}^*\mathbf{R}_{xx}(\mathbf{s}^* - \mathbf{A}^T\mathbf{y}) = \mathbf{A}^*\mathbf{R}_{xx}\mathbf{w}_{ss} = 0 \tag{4.236}$$

Since $\mathbf{As} = 0$ and the rank of the transformation matrix $\mathbf{A}$ is $N - 1$, (4.236) implies that $\mathbf{R}_{xx}\mathbf{w}$ is proportional to $\mathbf{s}^*$ so that

$$\mathbf{R}_{xx}\mathbf{w}_{ss} = \mu\mathbf{s}^* \tag{4.237}$$

$$\mathbf{w}_{ss} = \mu\mathbf{R}_{xx}^{-1}\mathbf{s}^* \tag{4.238}$$

where μ is a proportionality constant that may be evaluated using (4.229). Substituting $\mu = (\mathbf{s}^T\mathbf{R}_{xx}^{-1}\mathbf{s}^*)^{-1}$ in (4.238) then yields

$$\mathbf{w}_{ss} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{s}^*}{\mathbf{s}^T\mathbf{R}_{xx}^{-1}\mathbf{s}^*} \tag{4.239}$$

as the solution that the composite weight vector approaches when g becomes very large.

Preadaption spatial filtering avoids dynamic range problems, so it does require the implementation of multiple beams. The accuracy of the beam steering phase shifters limits the effectiveness of the constraints, but this limit is true of all three methods considered here. Two realizations of preadaption spatial filtering represented by Figure 4-21 include [29]: (1) the use of a Butler matrix to obtain orthogonal beams, one of which is regarded as the "main" beam; and (2) the use of an **A** matrix transformation obtained by fixed element-to-element subtraction.

4.4.3 Control Loop Spatial Filters

The Howells–Applebaum adaptive control loop with constraints applied directly in the loop by means of a spatial matrix filter is illustrated by the configuration of Figure 4-22. The spatial matrix filter removes any components of the signal vector **v** pointing in the direction of the unit length beam steering vector $\mathbf{b}^*$ by means of a projection operator. The successful removal of such signal components then constrains the array response in the direction of **b**.

The amplified output of the correlation mixer configuration of Figure 4-22 is given by

$$\mathbf{v}(t) = \mathbf{u}^*(t)\mathbf{x}^T(t)\mathbf{w}(t) \tag{4.240}$$

Taking expected values of (4.243) results in the steady-state values given by

$$\mathbf{v} = g\mathbf{R}_{xx}\mathbf{w}_{ss} \tag{4.241}$$

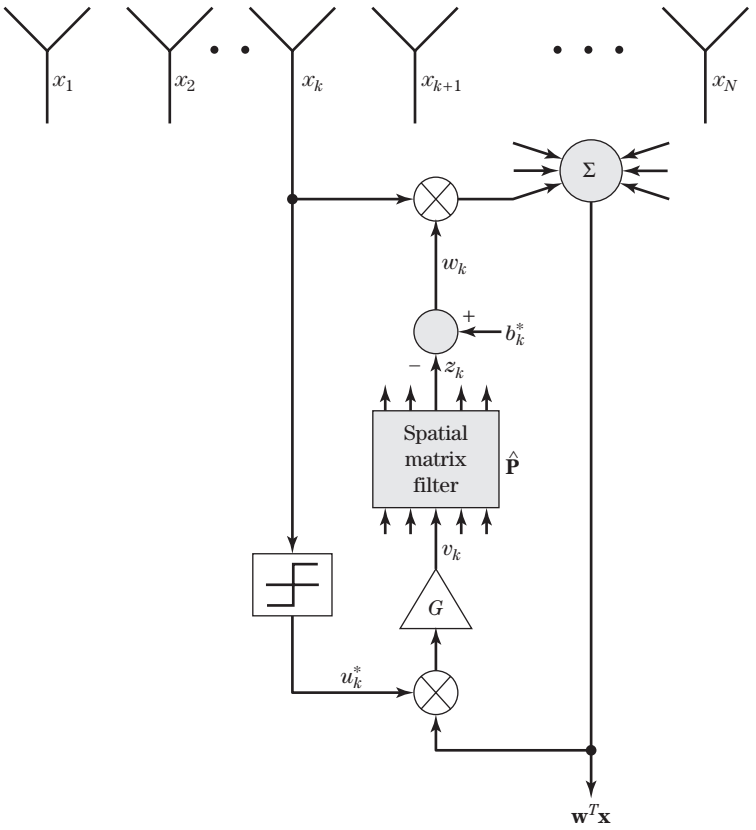


FIGURE 4-22 ■ Adaptive processor with control loop spatial filtering.

where g is a gain factor. Now

$$\mathbf{z} = \hat{\mathbf{P}}\mathbf{v} = g\hat{\mathbf{P}}\mathbf{R}_{xx}\mathbf{w}_{ss} \tag{4.242}$$

Substituting $\mathbf{w} = \mathbf{b}^* - \mathbf{z}$ into (4.245) means the steady-state weight values must satisfy

$$(\mathbf{I} + g\hat{\mathbf{P}}\mathbf{R}_{xx})\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.243}$$

When the beam steering vector is uniformly weighted, the projection performed by the spatial filter to remove signal components in the direction of $\mathbf{b}$ is

$$\hat{\mathbf{P}} = \mathbf{I} - \mathbf{b}^*\mathbf{b}^T \tag{4.244}$$

Substituting (4.244) into (4.243) results in

$$(\mathbf{I} + g\mathbf{R}_{xx} - g\mathbf{b}^*\mathbf{b}^T\mathbf{R}_{xx})\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.245}$$

$$[(\mathbf{I} + g\mathbf{R}_{xx})\mathbf{R}_{xx}^{-1} - g\mathbf{b}^*\mathbf{b}^T]\mathbf{R}_{xx}\mathbf{w}_{ss} = \mathbf{b}^* \tag{4.246}$$

Rewrite (4.246) in the form

$$\mathbf{R}_{xx}\mathbf{w}_{ss} = [\mathbf{Q} - g\mathbf{b}^*\mathbf{b}^T]^{-1}\mathbf{b}^* \tag{4.247}$$

where $\mathbf{Q} \triangleq (\mathbf{I} + g\mathbf{R}_{xx})\mathbf{R}_{xx}^{-1}$, and apply a matrix inversion identity to obtain the result

$$[\mathbf{Q} - g\mathbf{b}^*\mathbf{b}^T]^{-1}\mathbf{b}^* = \frac{\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*}{1 - g\mathbf{b}^T\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*} \tag{4.248}$$

The denominator of (4.243) may be simplified as

$$\begin{aligned} g\mathbf{b}^T\mathbf{R}_{xx}(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* &= \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx} - \mathbf{I})(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \\ &= \mathbf{b}^T\mathbf{b}^* - \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \\ &= 1 - \mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^* \end{aligned} \tag{4.249}$$

Substituting (4.248) and (4.249) into (4.247) then yields

$$\mathbf{w}_{ss} = \frac{(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*}{\mathbf{b}^T(\mathbf{I} + g\mathbf{R}_{xx})^{-1}\mathbf{b}^*} \tag{4.250}$$

The introduction of $\hat{\mathbf{P}} = \mathbf{I} - \mathbf{b}^*\mathbf{b}^T$ constrains the array response only in the direction of $\mathbf{b}$. Additional constraints are required to constrain the response over a finite region of the main lobe instead of only one direction. Likewise, when the steering vector is not uniform, an additional constraint is also required to obtain the desired quiescent pattern. It is furthermore desirable to transform the constraints to an orthogonal set, thereby minimizing the accuracy requirements of the spatial matrix filter. To illustrate a projection filter constructed from an orthogonal set, consider the representation of a projection filter with $M + 1$ orthogonal constraints:

$$\hat{\mathbf{P}} = \mathbf{I} - \sum_{m=0}^{M} \mathfrak{c}_m\mathfrak{c}_m^\dagger \tag{4.251}$$

where $\mathfrak{c}_m\mathfrak{c}_n = \delta_{mn}$, δ_{mn} = the Kronecker delta, and the $\mathfrak{c}_m$ are the constraint vectors.

The constraint that maintains the array response at the peak of the beam is the "zero-order" constraint. The weight vector solution obtained with a zero-order constraint differs from the unconstrained solution ($\hat{\mathbf{P}} = \mathbf{I}$) by a multiplicative scale factor. Multiple constraints are typically used to increase the beam constraint zone by controlling the first few derivatives of the pattern function in the direction of interest. A constraint that controls the mth derivative is referred to as an "mth-order" constraint.

To synthesize a $\mathfrak{c}_m$ constraint vector corresponding to the mth derivative of the pattern function, note that the pattern function of a linear array can be written as

$$AF(\theta) = \sum_{k=1}^{N} w_k e^{jk\theta} \tag{4.252}$$

The mth derivative of $AF(\theta)$ is

$$AF^m(\theta) = \sum_{k=1}^{N} (jk)^m w_k e^{jk\theta} \tag{4.253}$$

Consequently, the elements of $\mathfrak{c}_m$ (for $m = 0$, 1, and 2) are given by

$$\mathfrak{c}_{0i} = d_0 \tag{4.254}$$

$$\mathfrak{c}_{1i} = e_0 + e_1 i \tag{4.255}$$

$$\mathfrak{c}_{2i} = f_0 + f_1 i + f_2 i^2 \tag{4.256}$$

The constants defining the $\mathfrak{c}_m$ elements are made unit length and mutually orthogonal.

Consider how to establish a beam having nonuniform weighting as well as zero-, first-, and second-order constraints on the beam shape at the center of the main beam. First expand $\mathbf{w}_q$ in terms of the constraint vectors $\mathfrak{c}_m$ (for $m = 0$, 1, and 2) and a remainder vector $\mathfrak{c}_r$ as

$$\mathbf{w}_q = a_0\mathfrak{c}_0 + a_1\mathfrak{c}_1 + a_2\mathfrak{c}_2 + a_r\mathfrak{c}_r \tag{4.257}$$

where

$$a_i = \mathbf{w}_q^T \mathfrak{c}_i \quad \text{for } i = 0, 1, 2 \tag{4.258}$$

$$a_r\mathfrak{c}_r = w_q - \sum_{i=0}^{2} a_i\mathfrak{c}_i \tag{4.259}$$

Now, construct the complementary projection matrix filter according to

$$\hat{\mathbf{P}} = \mathbf{I} - \mathfrak{c}_0\mathfrak{c}_0^\dagger - \mathfrak{c}_1\mathfrak{c}_1^\dagger - \mathfrak{c}_2\mathfrak{c}_2^\dagger - \cdots - \mathfrak{c}_r\mathfrak{c}_r^\dagger \tag{4.260}$$

The subspace spanned by the constraint vectors in the N-dimension space of the adaptive processor is preserved by the foregoing construction. The spatial matrix filter constructed according to (4.263) results in a signal vector $\mathbf{z}$ containing no components in the direction of $\mathbf{w}_q$ or its first and second derivatives. The vector $\mathbf{w}_q$ is now added back in (at the point in Figure 4-21 where $\mathbf{b}^*$ is inserted) to form the final weight vector $\mathbf{w}$.

4.5 | CONSTRAINT FOR THE CASE OF KNOWN DESIRED SIGNAL POWER LEVEL

If there is no a priori knowledge of the desired signal structure or the direction of arrival, acquiring a weak desired signal in the presence of strong jamming or interference is accomplished by placing a constraint on the adaptation algorithm to prevent suppression of all signals (including interference) below a specified input power level. When a signal power threshold is greater than the desired signal input power, the weak desired signal will not be suppressed, whereas all interference signals above the threshold are suppressed.

The most common method of obtaining a power discrimination capability was formulated by Compton [30] and is based on the use of proportional feedback control in a Howells–Applebaum adaptive loop. To accomplish the same result Zahm [31] proposed another technique that uses a combination of a steering command vector and a bias signal. The weight adjustment control loops for the adaptive null-steering array in Figure 4-23 are governed by the differential equation

$$\frac{d\mathbf{w}}{dt} = \alpha\{\mathbf{x}(t)[x_0(t) - \mathbf{x}^{\dagger}(t)\mathbf{w}] - a\mathbf{w}\} \tag{4.261}$$

where a is a real scalar constant. The additional feedback path around the integrator provides the means for setting a power threshold.

For α sufficiently small, (4.261) can be approximated by expected values so that

$$\frac{d\mathbf{w}}{dt} = \alpha(\mathbf{r}_{xx_0} - [\mathbf{R}_{xx} + a\mathbf{I}]\mathbf{w}) \tag{4.262}$$

The steady-state weight vector is then given by

$$\mathbf{w} = [\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{r}_{xx_0} \tag{4.263}$$

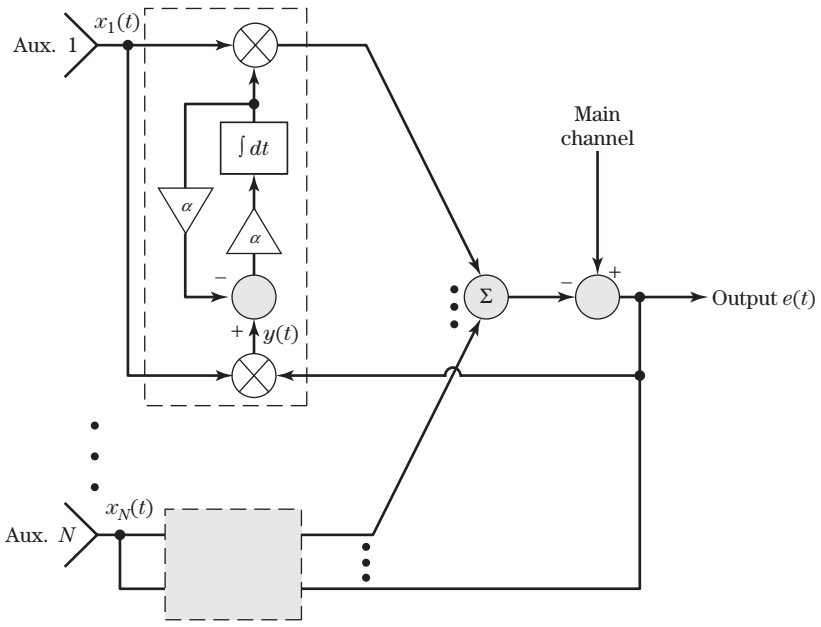


FIGURE 4-23 ■ Adaptive null steering with N auxiliary sensor elements. From Compton, Ohio State Univ. Quarterly Report 3234-1, December, 1971.

If a desired signal is present then the output signal-to-interference plus noise ratio (SINR) as

$$\text{SNR} = \frac{1}{(P_e/\overline{|s_0 - \mathbf{w}^\dagger \mathbf{s}|^2}) - 1} \tag{4.264}$$

where P_e represents the total output power, $\mathbf{s}$ is the desired signal direction vector, and s_0 is the main channel desired signal component. It will be convenient to define the parameter

$$SN' = \frac{\overline{|s_0 - \mathbf{w}^\dagger \mathbf{s}|^2}}{P_e} \tag{4.265}$$

so that

$$\text{SNR} = \frac{1}{(1/SN') - 1} \tag{4.266}$$

It can be shown that

$$SN' = \frac{\left| s_0 - \sum_{i=1}^{N} \frac{(\mathbf{Q}\mathbf{r}_{xx_0})_i^* (\mathbf{Q}\mathbf{s})_i}{\lambda_i + a} \right|^2}{a^2 \sum_{i=1}^{N} \frac{|(\mathbf{Q}\mathbf{r}_{xx_0})_i|^2}{\lambda_i (\lambda_i + a)^2} + P_{e_0}} \tag{4.267}$$

where $\mathbf{Q}$ is the unitary transformation that diagonalizes $\mathbf{R}_{xx}$, λ_i are the eigenvalues of R_{xx}, and P_{e_0} represents the minimum output power of $e(t)$ when $a = 0$. It can be shown from the previous results and (4.261) that a may be selected to prevent cancellation of the desired signal while suppressing high-level jammers. The output signal-to-interference plus noise ratio is therefore higher than when a pure integrator is used in the feedback loop.

To maximize the output SNR, the parameter a is selected to maximize SN'. To illustrate how a is selected, consider the case of one interfering jammer so that

$$\mathbf{R}_{xx} = J \mathbf{v}_J \mathbf{v}_J + P_s \mathbf{v}_s \mathbf{v}_s \tag{4.268}$$

$$\mathbf{r}_{xx_0} = \sqrt{J_0 J}\, \mathbf{v}_J e^{j\phi_J} + \sqrt{P_{s_0} P_s}\, \mathbf{v}_s e^{j\phi_s} \tag{4.269}$$

where

J_0 = main channel jammer power
J = auxiliary channel jammer power (assumed equal in all auxiliary channels)
$\mathbf{v}_J$ = jammer direction delay vector

P_{s_0}, P_s, and $\mathbf{v}_s$ are similarly defined for the desired signal. ϕ_J and ϕ_s represent the relative phase between the main and auxiliary channel signals for the jamming and desired signals, respectively.

If the desired signal and the interference signal angles of arrival are such that $\mathbf{v}_s$ and $\mathbf{v}_J$ are orthogonal (which simplifies the discussion for tutorial purposes), then (4.270) reduces to

$$SN' = \frac{P_{s_0} \left[\dfrac{\sigma^2 + a}{\sigma^2 + a + NP_s} \right]^2}{a^2 N \left[\dfrac{P_{s_0} P_s}{(NP_s + \sigma^2 + a)^2 (NP_s + \sigma^2)} + \dfrac{J_0 J}{(NJ + \sigma^2 + a)^2 (NJ + \sigma^2)} \right] + P_{e_0}} \tag{4.270}$$

where σ^2 = auxiliary channel thermal noise power. For $a = 0$ (which corresponds to the conventional LMS null-steering algorithm) SN' becomes

$$SN' = \frac{P_{s_0}\left[\dfrac{\sigma^2}{NP_s + \sigma^2}\right]^2}{P_{e_0}}; \quad a = 0 \tag{4.271}$$

This result shows that SN' decreases as the input desired signal power in the auxiliary channels NP_s increases above the thermal noise level σ^2. When $NP_s \gg \sigma^2$ in (4.271), SN' is inversely proportional to the input desired signal power, which is the power inversion characteristic of the minimum MSE performance criterion. When $a \gg NP_s + \sigma^2$ and $a \gg NJ + \sigma^2$, (4.270) becomes

$$SN' = \frac{P_{s_0}}{\dfrac{NP_s}{NP_s + \sigma^2}P_{s_0} + \dfrac{NJ}{NJ + \sigma^2}J_0 + P_{e_0}} \tag{4.272}$$

Suppression of the main channel signal is prevented by selecting a to be sufficiently large. However, a is too large in this example, because jammer suppression has also been prevented, as indicated by the presence of the term $NJJ_0/(NJ + \sigma^2)$ in the denominator of (4.272).

Next, assume that the main channel jammer power J_0 is nominally equal to the auxiliary channel jammer power, and choose $a = NP_s$. Then SN' becomes

$$SN' = \frac{0.25P_{s_0}}{\dfrac{P_{s_0}(NP_s - \sigma^2)^2}{4NP_s(NP_s + \sigma^2)} + \dfrac{(NP_s - \sigma^2)^2}{N[1 + (P_s/J)]^2(NJ + \sigma^2)} + P_{e_0}} \tag{4.273}$$

For $J \gg P_s$, $P_s \gg \sigma^2$,

$$SN' \approx \frac{0.25P_{s_0}}{0.25P_{s_0} + P_{e_0}}$$

An approximation for the output signal-to-interference plus noise ratio in (4.269) is

$$\text{SNR} \cong \frac{1}{4}\frac{P_{s_0}}{P_{e_0}} \tag{4.274}$$

Thus, the output signal-to-interference plus noise ratio is now proportional to the main channel signal power divided by the output residue power P_{e_0} (recall that P_{e_0} is the minimum output residue power obtained when $a = 0$). Equation (4.274) shows that when $J \gg P_s$ and $P_s \gg \sigma^2$, the output signal-to-interference plus noise ratio can be significantly improved by selecting the weight feedback gain as

$$a \approx NP_s \tag{4.275}$$

This value of a (when $J \gg P_s$ and $P_s \gg \sigma^2$) then prevents suppression of the relatively weak desired signal while strongly suppressing higher power level jamming signals.

4.6 THE DSD ALGORITHM

We have seen that, if there are perfect gradient measurements on each iteration, the adaptive weight vector converges to the optimal (Wiener) weight vector. In practice, gradient vector estimates are obtained from a limited statistical sample. The DSD algorithm

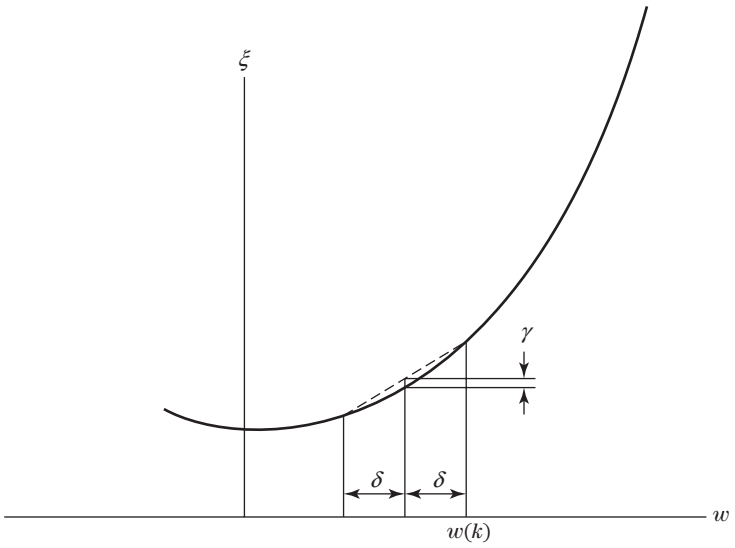


FIGURE 4-24 ■ One-dimensional gradient estimation by way of direct measurement.

obtains gradient vector estimates by direct measurement and is straightforward and easy to implement [13].

The parabolic performance surface representing the MSE function of a single variable w is defined by

$$\xi[w(k)] \triangleq \xi(k) = \xi_{\min} + \alpha w^2(k) \tag{4.276}$$

Figure 4-24 represents the parabolic performance surface as a function of a single component of the weight vector **w**. The first and second derivatives of the MSE are

$$\left[\frac{d\xi(k)}{dw}\right]_{w=w(k)} = 2\alpha w(k) \tag{4.277}$$

$$\left[\frac{d^2\xi(k)}{dw^2}\right]_{w=w(k)} = 2\alpha \tag{4.278}$$

These derivatives are numerically estimated by taking the "symmetric differences"

$$\left[\frac{d\xi(k)}{dw}\right]_{w=w(k)} = \frac{\xi[w(k)+\delta] - \xi[w(k)-\delta]}{2\delta} \tag{4.279}$$

$$\left[\frac{d^2\xi(k)}{dw^2}\right]_{w=w(k)} = \frac{\xi[w(k)+\delta] - 2\xi[w(k)] + \xi[w(k)-\delta]}{\delta^2} \tag{4.280}$$

The procedure for estimating the first derivative illustrated in Figure 4-24 requires that the weight adjustment be altered to two distinct settings while the gradient estimate is obtained. If K data samples are taken to estimate the MSE at the two weight settings $w(k)+\delta$ and $w(k)-\delta$, then the average MSE experienced (over both settings) is greater than the MSE at $w(k)$ by an amount γ. Consequently, a performance penalty is incurred that results from the weight alteration used to obtain the derivative estimate.

4.6.1 Performance Penalty Incurred by Gradient Measurement

Figure 4-24 shows that γ for the one-dimensional case is

$$\gamma = \frac{\alpha[w(k)+\delta]^2 + \alpha[w(k)-\delta]^2 + 2\xi_{\min}}{2} - \alpha w^2(k) - \xi_{\min} = \alpha\delta^2 \tag{4.281}$$

Thus, the value of γ depends only on α and not on $w(k)$. A dimensionless measure of the system perturbation each time the gradient is measured is defined by

$$P \triangleq \frac{\gamma}{\xi_{\min}} = \frac{\alpha\delta^2}{\xi_{\min}} \tag{4.282}$$

The perturbation is the average increase in the MSE normalized with respect to the minimum achievable MSE.

A two-dimensional gradient is needed for the input signal correlation matrix

$$\mathbf{R}_{xx} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{4.283}$$

The MSE corresponding to this correlation matrix is then

$$\xi = r_{11}w_1^2 + r_{22}w_2^2 + 2r_{12}w_1w_2 + \xi_{\min} \tag{4.284}$$

Measuring the partial derivative of the previous performance surface along the coordinate w_1 yields a perturbation

$$P = \frac{r_{11}\delta^2}{\xi_{\min}} \tag{4.285}$$

Likewise, the perturbation for the measured partial derivative along the coordinate w_2 is

$$P = \frac{r_{22}\delta^2}{\xi_{\min}} \tag{4.286}$$

If we allot equal time for the measurement of both partial derivatives (a total of $2K$ data samples are used for both measurements), the average perturbation experienced during the complete measurement process is given by

$$P_{\text{av}} = \frac{\delta^2}{\xi_{\min}} \cdot \frac{r_{11}+r_{22}}{2} \tag{4.287}$$

For N dimensions, define a general perturbation as the average of the perturbations experienced for each of the individual gradient component measurements so that

$$P = \frac{\delta^2}{\xi_{\min}} \cdot \frac{\text{tr}(\mathbf{R}_{xx})}{N} \tag{4.288}$$

where "tr" denotes trace, which is defined as the sum of the diagonal elements of the indicated matrix. When we convert the $\mathbf{R}_{xx}$ matrix to normal coordinates, the trace of $\mathbf{R}_{xx}$ is the sum of its eigenvalues. Since the sum of the eigenvalues divided by N is the average

of the eigenvalues (λ_{av}), the perturbation for the N-dimensional case is

$$P = \frac{\delta^2}{\xi_{min}}\lambda_{av} \tag{4.289}$$

Alternative means of measuring the gradient have also been used in practical systems. By perturbing or dithering a single weight sinusoidally, the cross-correlation between the weight value and the performance measure can be measured to determine the derivative of the performance surface. Likewise, all weights can be dithered simultaneously at distinct individual frequencies and the gradient components then obtained by cross-correlation. The procedure for determining the derivative illustrated in Figure 4-24 corresponds to square-wave dithering.

4.6.2 Gradient Measurement Noise and Misadjustment in the Weight Vector

Gradients measured by taking finite differences as in Figure 4-23 are noisy because the MSE measurements on which the differences are based are noisy. Each MSE measurement is an estimate $\hat{\xi}$ of the actual MSE ξ based on K data samples:

$$\hat{\xi} = \frac{1}{K}\sum_{k=1}^{K} e^2(k) \tag{4.290}$$

It is well known that the variance of a sample average estimate of the mean square obtained from K independent samples is given by the difference between the mean fourth and the square of the mean square all divided by K. Consequently the variance of $\hat{\xi}$ may be expressed as [32]

$$\text{var}[\hat{\xi}] = \frac{E\{e^4(k)\} - [E\{e^2(k)\}]^2}{K} \tag{4.291}$$

If the random variable $e(k)$ is normally distributed with zero mean and variance σ^2, then its mean fourth is $3\sigma^4$, and the square of its mean square is σ^4. Consequently, the variance in the estimate of ξ is given by

$$\text{var}[\hat{\xi}] = \frac{1}{K}(3\sigma^4 - \sigma^4) = \frac{2\sigma^4}{K} = \frac{2\xi^2}{K} \tag{4.292}$$

From (4.292) we find that the variance of $\hat{\xi}$ is proportional to the square of ξ and inversely proportional to the number of data samples. In general, the variance can be expressed as

$$\text{var}[\hat{\xi}] = \eta\frac{\xi^2}{K} \tag{4.293}$$

where η has the value of 2 for an unbiased Gaussian density function. In the event that the probability density function for $\hat{\xi}$ is not Gaussian, then the value of η is generally less than but close to 2. It is therefore convenient to assume that the final result expressed in (4.292) holds for the analysis that follows.

The derivatives required by the DSD algorithm are measured in accordance with (4.279). The measured derivative involves taking finite differences of two MSE estimates, so the error in the measured derivative involves the sum of two independent components

[since the error samples $e(k)$ are assumed to be independent]. The variance of each component to the derivative error is given by (4.292). Assume that we are attempting to measure the derivative at a point on the performance surface where the weight vector is near the minimum point of the MSE surface and that the perturbation P is small, then the two components of measured derivative error will have essentially the same variances. The total variance of the measured derivative error will then be the sum of the variances of the two components. From (4.279) and (4.292) it follows that the variance in the estimate of the derivative is given by

$$\text{var}\left[\frac{d\xi}{dw}\right]_{w=w(k)} = \frac{1}{4\delta^2}\left[\frac{2\xi^2[w(k)+\delta]}{K} + \frac{2\xi^2[w(k)-\delta]}{K}\right] \cong \frac{\xi_{\min}^2}{K\delta^2} \tag{4.294}$$

When an entire gradient vector is measured, then the errors in each component are independent. It is convenient to define a gradient noise vector $\mathbf{g}(k)$ in terms of the true gradient $\quad(k)$ and the estimated gradient $\hat{\ }(k)$:

$$\hat{\ }(k) \triangleq \quad(k) + \mathbf{g}(k) \tag{4.295}$$

where $\mathbf{g}(k)$ is the gradient noise vector. Under the previously assumed conditions, the covariance of the gradient noise vector can be expressed as

$$\text{cov}\,[\mathbf{g}(k)] = \frac{\xi_{\min}^2}{K\delta^2}\mathbf{I} \tag{4.296}$$

Transforming the gradient noise vector into normal coordinates, we have

$$\mathbf{g}'(k) = \mathbf{Q}\mathbf{g}(k) \tag{4.297}$$

We see from (4.296) that the covariance matrix of $\mathbf{g}(k)$ is a scalar multiplying the identity matrix, so projecting into normal coordinates through the orthonormal transformation $\mathbf{Q}$ yields the same covariance for $\mathbf{g}'(k)$:

$$\text{cov}\,[\mathbf{g}'(k)] = E\{\mathbf{Q}\mathbf{g}(k)\mathbf{g}^T(k)\mathbf{Q}^{-1}\} = \frac{\xi_{\min}^2}{K\delta^2}\mathbf{I} \tag{4.298}$$

This result merely emphasizes that near the minimum point of the performance surface the covariance of the gradient noise is essentially a constant and does not depend on $\mathbf{w}(k)$.

The fact that the gradient estimates are noisy means that weight adaptation based on these gradient estimates will also be noisy, and it is consequently of interest to determine the corresponding noise in the weight vector. Using estimated gradients, the method of steepest descent yields the vector difference equation

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \Delta_s(-\hat{\ }(k)) = \mathbf{v}(k) + \Delta_s[-\quad(k) - \mathbf{g}(k)] \tag{4.299}$$

where $\mathbf{v}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_{\text{opt}}$. Since the true gradient from (4.63) is given by

$$(k) = 2\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.300}$$

(4.299) can be rewritten as

$$\mathbf{v}(k+1) = [\mathbf{I} - 2\Delta_s\mathbf{R}_{xx}]\mathbf{v}(k) - \Delta_s\mathbf{g}(k) \tag{4.301}$$

which is a first-order difference equation having a stochastic driving function $-\Delta_s \mathbf{g}(k)$. Projecting the previous difference equation into normal coordinates by premultiplying by $\mathbf{Q}$ then yields

$$\mathbf{v}'(k+1) = [\mathbf{I} - 2\Delta_s \mathbf{\Lambda}]\mathbf{v}'(k) - \Delta_s \mathbf{g}'(k) \tag{4.302}$$

After initial adaptive transients have died out and the steady state is reached, the weight vector $\mathbf{v}'(k)$ behaves like a stationary random process in response to the stochastic driving function $-\Delta_s \mathbf{g}'(k)$. In the normal coordinate system there is no cross-coupling between terms, and the components of $\mathbf{g}'(k)$ are uncorrelated; thus, the components of $\mathbf{v}'(k)$ are also mutually uncorrelated, and the covariance matrix of $\mathbf{g}'(k)$ is diagonal. The covariance matrix of $\mathbf{v}'(k)$ describes how noisy the weight vector will be in response to the stochastic driving function, and we now proceed to find this matrix. Since $\text{cov}[\mathbf{v}'(k)] \triangleq E\{\mathbf{v}'(k)\mathbf{v}'^T(k)\}$, it is of interest to determine the quantity $\mathbf{v}'(k+1)\mathbf{v}'^T(k+1)$ by way of (4.302) as follows:

$$\begin{aligned}\mathbf{v}'(k+1)\mathbf{v}'^T(k+1) = {} & (\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\mathbf{v}'(k)\mathbf{v}'^T(k)(\mathbf{I} - 2\Delta_s \mathbf{\Lambda}) \\ & + \Delta_s^2 \mathbf{g}'(k)\mathbf{g}'^T(k) - \Delta_s(\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\mathbf{v}'(k)\mathbf{g}'(k)\mathbf{v}'^T(k) \\ & - \Delta_s \mathbf{g}'(k)\mathbf{v}'^T(k)\end{aligned} \tag{4.303}$$

Taking expected values of both sides of (4.303) and noting that $\mathbf{v}'(k)$ and $\mathbf{g}'(k)$ are uncorrelated since $\mathbf{v}'(k)$ is affected only by gradient noise from previous adaptive cycles, we obtain for the steady state

$$\begin{aligned}\text{cov}[\mathbf{v}'(k)] &= (\mathbf{I} - 2\Delta_s \mathbf{\Lambda})\text{cov}[\mathbf{v}'(k)](\mathbf{I} - 2\Delta_s \mathbf{\Lambda}) + \Delta_s^2 \text{cov}[\mathbf{g}'(k)] \\ &= \left(\mathbf{I} - 4\Delta_s \mathbf{\Lambda} + 4\Delta_s^2 \mathbf{\Lambda}^2\right)\text{cov}[\mathbf{v}'(k)] + \Delta_s^2 \text{cov}[\mathbf{g}'(k)]\end{aligned} \tag{4.304}$$

Combining like terms in (4.304) then yields

$$\text{cov}[\mathbf{v}'(k)] = \Delta_s^2 \left[4\Delta_s \mathbf{\Lambda} - 4\Delta_s^2 \mathbf{\Lambda}^2\right]^{-1} \text{cov}[\mathbf{g}'(k)] \tag{4.305}$$

In practice, the step size in the method of steepest descent is selected so that

$$\Delta_s \mathbf{\Lambda} \ll \mathbf{I} \tag{4.306}$$

As a result of (4.306), squared terms occurring in (4.305) can be neglected so that

$$\text{cov}[\mathbf{v}'(k)] \cong \frac{\Delta_s}{4} \mathbf{\Lambda}^{-1} \text{cov}[\mathbf{g}'(k)] \tag{4.307}$$

Since $\text{cov}[\mathbf{g}'(k)]$ is given by (4.298), we now have

$$\text{cov}[\mathbf{v}'(k)] \cong \frac{\Delta_s \xi_{\min}^2}{4K\delta^2} \mathbf{\Lambda}^{-1} \tag{4.308}$$

The covariance of the weight vector in the operational coordinate system can be obtained from (4.308) by recalling that $\mathbf{R}_{xx}^{-1} = \mathbf{Q}^{-1}\mathbf{\Lambda}^{-1}\mathbf{Q}$ and $\mathbf{v}' = \mathbf{Q}\mathbf{v}$ so that

$$\begin{aligned}\text{cov}[\mathbf{v}(k)] &= E\left\{\mathbf{Q}^{-1}\mathbf{v}'(k)\mathbf{v}'^T(k)\mathbf{Q}\right\} \\ &= \frac{\Delta_s \xi_{\min}^2}{4K\delta^2} \mathbf{R}_{xx}^{-1}\end{aligned} \tag{4.309}$$

Without any noise in the weight vector, the method of steepest descent converges to a steady-state solution at the minimum point of the MSE performance surface (the bottom of the bowl). The MSE would then be $\xi_{\min}$. The noise present in the weight vector causes the steady-state solution to randomly wander about the minimum point. The result of this wandering is a steady-state MSE that is greater than $\xi_{\min}$ and hence is said to have an "excess" MSE. We will now consider how severe this excess MSE is for the noise that is in the weight vector.

We have already seen in Section 4.1.3 that the MSE can be expressed as

$$\xi(k) = \xi_{\min} + \mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k) \tag{4.310}$$

where $\mathbf{v}'(k) = \mathbf{w}'(k) - \mathbf{w}'_{\text{opt}}$. Consequently, the average excess MSE is

$$E\left\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\right\} = \sum_{p=1}^{N} \lambda_p E\left\{\left[v'_p(k)\right]^2\right\} \tag{4.311}$$

But from (4.308) we may write

$$E\left\{\left[v'_p(k)\right]^2\right\} = \frac{\Delta_s \xi_{\min}^2}{4k\delta^2}\left(\frac{1}{\lambda_p}\right) \tag{4.312}$$

Thus, (4.311) can be rewritten as

$$E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} = \frac{N\Delta_s \xi_{\min}^2}{4K\delta^2} \tag{4.313}$$

Recalling that the misadjustment M is defined as the average excess MSE divided by the minimum MSE there results for the DSD algorithm

$$M = \frac{N\Delta_s \xi_{\min}}{4K\delta^2} \tag{4.314}$$

The foregoing result is more usefully expressed in terms of time constants of the learning process and the perturbation of the gradient estimation process as developed next.

Each measurement to determine a gradient component uses $2K$ samples of data. Each adaptive weight iteration involves N gradient component measurements and therefore requires a total of $2KN$ data samples. From Section 4.2.3 it may be recalled that the MSE learning curve has a pth mode time constant given by

$$\tau_{p_{\text{MSE}}} = \frac{1}{4\Delta_s \lambda_p} = \frac{\tau_p}{2} \tag{4.315}$$

in time units of the number of iterations. It is useful to define a new time constant $T_{p_{\text{MSE}}}$ whose basic time unit is the data sample and whose value is expressed in terms of the number of data samples. It follows that for the DSD algorithm

$$T_{p_{\text{MSE}}} \triangleq 2KN\tau_{p_{\text{MSE}}} \tag{4.316}$$

The time constant $T_{p_{\text{MSE}}}$ relates to real time units (seconds) once the sampling rate is known.

By using the perturbation formula (4.282) to substitute for $\xi_{\min}$ in (4.314), the misadjustment for the DSD algorithm is rewritten as

$$M = \frac{N\Delta_s\lambda_{\text{av}}}{4KP} \tag{4.317}$$

The time constant defined by (4.316) is rewritten using (4.315) as

$$T_{p_{\text{MSE}}} = \frac{NK}{2\Delta_s\lambda_p} \tag{4.318}$$

from which one can conclude that

$$\lambda_p = \frac{NK}{2\Delta_s}\left(\frac{1}{T_{p_{\text{MSE}}}}\right) \tag{4.319}$$

so that

$$\lambda_{\text{av}} = \frac{NK}{2\Delta_s}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} \tag{4.320}$$

Combining (4.26) and (4.320) then yields the misadjustment as

$$M = \frac{N^2}{8P}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} \tag{4.321}$$

Equation (4.321) shows that for the DSD algorithm, misadjustment is proportional to the square of the number of weights and inversely proportional to the perturbation. In addition, the misadjustment is also inversely proportional to the speed of adaptation (fast adaptation results in high misadjustment). Since the DSD algorithm is based on steepest descent, it suffers from the disparate eigenvalue problem discussed in Section 4.2.2.

It is appropriate here to compare the misadjustment for the DSD algorithm given by (4.321) with the misadjustment for the LMS algorithm given by (4.83). With a specified level of misadjustment for the LMS algorithm, the adaptive time constants increase linearly with the number of weights rather than with the square of the number of weights as is the case with the DSD algorithm. Furthermore, with the LMS algorithm there is no perturbation. As a result, in typical circumstances much faster adaptation is possible with the LMS algorithm than with the DSD algorithm.

M is defined as a normalized performance penalty that results from noise in the weight vector. In an actual adaptive system employing the DSD algorithm, the weight vector is not only stochastically perturbed due to the presence of noise but in addition is deterministically perturbed so the gradient can be measured. As a consequence of the deterministic perturbation, another performance penalty accrues as measured by the perturbation P, which is also a normalized ratio of excess MSE. The total excess MSE is therefore the sum of the "stochastic" and "deterministic" perturbation components. The total misadjustment can be expressed as

$$M_{\text{tot}} = M + P \tag{4.322}$$

Adding the previous two components then yields

$$M_{\text{tot}} = \frac{N^2}{8P}\left(\frac{1}{T_{\text{MSE}}}\right)_{\text{av}} + P \tag{4.323}$$

Since P is a design parameter given by (4.282), it can be selected by choosing the deterministic perturbation size δ. It is desirable to minimize the total misadjustment M_{tot} by appropriately selecting P. The result of such optimization is to make the two right-hand terms of (4.323) equal so that

$$P_{\text{opt}} = \frac{1}{2} M_{\text{tot}} \tag{4.324}$$

The minimum total misadjustment then becomes

$$(M_{\text{tot}})_{\min} = \frac{N^2}{4P_{\text{opt}}} \left(\frac{1}{T_{\text{MSE}}} \right)_{\text{av}} = \left[\frac{N^2}{2} \left(\frac{1}{T_{\text{MSE}}} \right)_{\text{av}} \right]^{1/2} \tag{4.325}$$

Unlike the LMS algorithm, the DSD algorithm is sensitive to any correlation that exists between successive samples of the error signal $e(k)$, since such correlation has the effect of making the effective statistical sample size less than the actual number of error samples in computing the estimated gradient vector. Because of such reduced effective sample size, the actual misadjustment experienced is greater than that predicted by (4.325), which was derived using the assumption of statistical independence between successive error samples.

4.7 | THE ACCELERATED GRADIENT APPROACH (AG)

Algorithms based on the steepest descent method exhibit an undesirable degree of sensitivity of the convergence speed to the eigenvalue spread in the input signal covariance matrix. The conjugate gradient method [33–39] is faster than the steepest descent approach. This reluctance to apply accelerated gradient methods to adaptive array problems is due principally to the following reasons:

1. There is increased hardware complexity associated with implementing the algorithm.
2. The enhanced convergence speed realized is not as fast as what can be obtained with other methods (to be discussed in later chapters).
3. The various accelerated gradient methods are all susceptible (although with different degrees) to signals that are noise corrupted.
4. There is a required increase in computation and memory space over steepest descent methods.

Despite these objections, it is worthwhile to apply one of the accelerated gradient methods to determine what improvement in convergence speed reasonably might be expected by recourse to such techniques. Powell's method [35] is tolerant of noise-induced errors, although other methods theoretically may yield faster convergence speeds. Powell's method assumes that if a performance measure $\Re(\mathbf{w})$ is quadratic in the independent variables, then any line that passes through the minimum point of the quadratic performance surface intersects the family of constant performance contours at equal angles. This property is illustrated in Figure 4-25 for two dimensions where it is seen that the line AC connecting point A with the minimum point C intersects the constant performance contours at equal angles. As a consequence of the equal angle property, the line

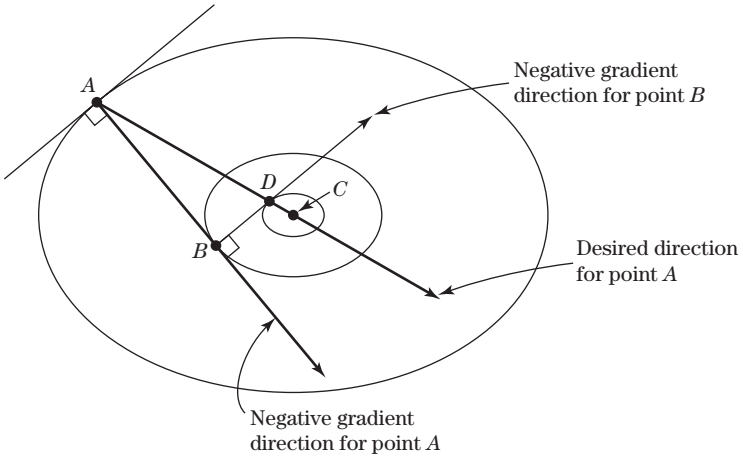


FIGURE 4-25 ■ Two-dimensional diagram showing directional relationships for the Powell descent method.

joining the point A with the point D in Figure 4-25 passes through the point C where the derivative of the performance measure $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line AD is zero.

Given an initial estimate $\mathbf{w}_0$ at point A, first find the gradient direction that is normal to the tangent of the constant performance measure contour. Proceed along the line defined by the negative gradient direction to the point B where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line is zero. The point B may in fact be any arbitrary point on the line that is a finite distance from A; however, by choosing it in the manner described the convergence of the method is assured.

Having found point B, the negative gradient direction that is parallel to the original tangent at $\mathfrak{P}(\mathbf{w}_0)$. Traveling in this new normal direction, we find a point where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line is zero (point D in Figure 4-25). The line passing through the points A and D also passes through the point C. The desired point C is the point where the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to distance along the line AD is zero.

The generalization of the foregoing procedure to an N-dimensional space can be obtained by recognizing that the directional relationships (which depend on the equal angle property) given in Figure 4-25 are valid only in a two-dimensional plane. The first step (moving from point A to point B) is accomplished by moving in the negative gradient direction in the N-dimensional space. Having found point B, we can construct $(N-1)$ planes between the original negative gradient direction and $(N-1)$ additional mutually orthogonal vectors, thereby defining points C, D, E, . . . , until $(N-1)$ additional points have been defined. The last three points in the N-dimensional space obtained in the foregoing manner may now be treated in the same fashion as points A, B, and D of Figure 4-25 by drawing a connecting line between the last point obtained and the point defined two steps earlier. Traveling along the connecting line one may then define a new point C of Figure 4-25. This new point may then be considered as point D in Figure 4-25, and a new connecting line may be drawn between the new point and the point obtained three steps earlier.

The steps corresponding to one complete Powell descent cycle for five dimensions are illustrated in Figure 4-26. The first step from A to B merely involves traveling in the negative gradient direction $\mathbf{v}_1$ with a step size α_1 chosen to satisfy the condition

$$\frac{d}{d\alpha_1}\{\mathfrak{P}[\mathbf{w}(0)+\alpha_1\mathbf{v}_1]\}=0 \tag{4.326}$$

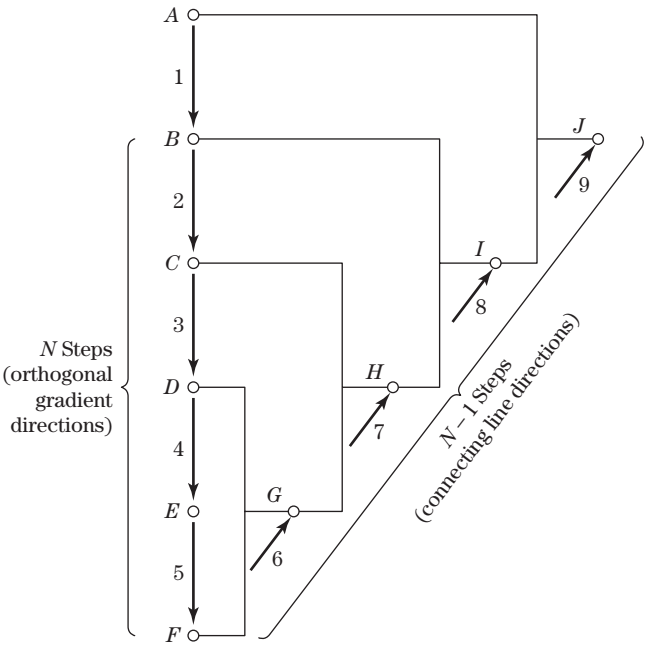


FIGURE 4-26 ■ Illustration of Powell descent method steps required in five dimensions for one complete cycle.

so that

$$\mathbf{w}(1) = \mathbf{w}(0) + \alpha_1 \mathbf{v}_1 \tag{4.327}$$

Having determined point B, we determine point C by traveling in the negative gradient direction $\mathbf{v}_2$ (a direction that is also orthogonal to $\mathbf{v}_1$) from point B with step size α_2 selected to satisfy

$$\frac{d}{d\alpha_2}\{\Re[\mathbf{w}(1) + \alpha_2 \mathbf{v}_2]\} = 0 \tag{4.328}$$

Point D is determined from point C by using the aforementioned procedure, and this process continues until a total of five points (B through F in Figure 4-26) are defined. A descent direction $\mathbf{v}_6$ is defined by drawing a connecting line between points D and F (analogous to points A and D of Figure 4-25) and traveling along this line with step size α_6 selected to satisfy

$$\frac{d}{d\alpha_6}\{\Re[\mathbf{w}(5) + \alpha_6 \mathbf{v}_6]\} = 0 \tag{4.329}$$

thereby arriving at point G. A new descent direction $\mathbf{v}_7$ is defined by drawing a connecting line between points C and G and traveling along this line with step size α_7 selected to satisfy

$$\frac{d}{d\alpha_7}\{\Re[\mathbf{w}(6) + \alpha_7 \mathbf{v}_7]\} = 0 \tag{4.330}$$

thereby arriving at point H. This process continues until the solution point J for the cycle is found from the descent direction $\mathbf{v}_9$ (defined by the connecting line between points A and I) and the step size α_9. In general, one Powell descent cycle in N-dimensional space

therefore requires $N + N - 1 = 2N - 1$ steps. On completion of one descent cycle, the entire cycle are repeated using the last weight vector obtained as the initial weight vector for the new cycle.

4.7.1 Algorithm Based on the Powell Accelerated Gradient Cycle

Each step in a Powell descent cycle involves traveling from a weight vector $\mathbf{w}$ along a direction $\mathbf{v}$ with step size α such that

$$\frac{d}{d\alpha}[\Re(\mathbf{w} + \alpha\mathbf{v})] = 0 \tag{4.331}$$

For complex weights the MSE performance measure is given by

$$\xi(\mathbf{w}) = E\{d^2\} + \mathbf{w}^\dagger \mathbf{r}_{xd} + \mathbf{r}_{xd}^\dagger \mathbf{w} + \mathbf{w}^\dagger \mathbf{R}_{xx} \mathbf{w} \tag{4.332}$$

The gradient of $\xi(\mathbf{w} + \alpha\mathbf{v})$ with respect to α is then given by

$$_\alpha[\xi(\mathbf{w} + \alpha\mathbf{v})] = \mathbf{v}^\dagger \mathbf{r}_{xd} + \mathbf{r}_{xd}^\dagger \mathbf{v} + \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{w} + \mathbf{w}^\dagger \mathbf{R}_{xx} \mathbf{v} + 2\alpha \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v} \tag{4.333}$$

and this gradient is equal to zero when the step size is

$$\alpha = -\frac{\mathbf{v}^\dagger \mathbf{r}_{xd} + \mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{w}}{\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}} \tag{4.334}$$

Since $\mathbf{r}_{xd}$ and $\mathbf{R}_{xx}$ are unknown, some estimate of the numerator is employed to obtain an appropriate step size estimate. Noting that $\mathbf{r}_{xd} + \mathbf{R}_{xx}\mathbf{w}$ is one-half the gradient of $\xi(\mathbf{w})$, it follows that the numerator of (4.334) are approximated by $\mathbf{v}^\dagger \text{Av}\{e(k)\mathbf{x}(k)\}$. Note that the quantity $\mathbf{v}^\dagger \mathbf{x}$ is regarded as the output of a processor whose weights correspond to $\mathbf{v}$ and that $\text{Av}\{(\mathbf{v}^\dagger \mathbf{x})(\mathbf{x}^\dagger \mathbf{v})\}$ is an approximation of the quantity $\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}$, where the average Av{ } is taken over K data samples. The simultaneous generation of the estimates $\hat{}_w$ and $\text{Av}\{\mathbf{v}^\dagger \mathbf{x}\mathbf{x}^\dagger \mathbf{v}\}$ requires parallel processors: one processor with weight values equal to $\mathbf{w}(k)$ and another processor with weight values equal to $\mathbf{v}(k)$. Having described the procedure for determining the appropriate step size along a direction $\mathbf{v}$, we may now consider the steps required to implement an entire Powell descent cycle.

The steps required to generate one complete Powell descent cycle are as follows.

Step 1 Starting with the initial weight setting $\mathbf{w}(0)$, estimate the negative gradient direction $\mathbf{v}(0)$ using K data samples then travel in this direction with the appropriate step size to obtain $\mathbf{w}(1)$. The step size determination requires an additional K data samples to obtain by way of (4.334).

Steps 2 $\rightarrow N$ Estimate the negative gradient direction at $\mathbf{w}(k)$ using K data samples. If the gradient estimates and the preceding step size were error free, the current gradient is automatically orthogonal to the previous gradient directions. Since the gradient estimate is not error free, determine the new direction of travel $\mathbf{v}(k)$ by requiring it to be orthogonal to all previous directions $\mathbf{v}(0), \mathbf{v}(1), \ldots, \mathbf{v}(k-1)$ by employing the Schmidt orthogonalization process so that

$$\mathbf{v}(k) = \hat{}(k) - \sum_{i=0}^{k-1} \frac{[\mathbf{v}^\dagger(i) \hat{}(k)]}{[\mathbf{v}^\dagger(i)\mathbf{v}(i)]} \cdot \mathbf{v}(i) \tag{4.335}$$

Travel in the direction $-\mathbf{v}(k)$ using the appropriate step size (which requires an additional K data samples to obtain) to arrive at $\mathbf{w}(k+1)$.

Steps $N+1 \rightarrow 2N$ **1** Determine the new direction of travel at $\mathbf{w}(k)$ by forming

$$\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}[2(N-1)-k] \tag{4.336}$$

Travel in the direction $-\mathbf{v}(k)$ from $w[2(N-1)-k]$ using the appropriate step size to arrive at $\mathbf{w}(k+1)$. These steps require only K data samples since now the direction of travel does not require that a gradient estimate be obtained.

4.8 GRADIENT ALGORITHM WITH CONSTRAINTS

The early applications of sidelobe cancellation to radar antennas neglected the effects of signals in the main beam on the adapted response. Such neglect was amply justified because adaptive processors would not respond to low level reflected target signals, and the small number of degrees of freedom then available to the adaptive processor limited the response to large targets or extended clutter. Current adaptive arrays with large numbers of degrees of freedom are explicitly constrained to avoid degradation of the main beam.

Adaptive arrays having large numbers of degrees of freedom and fast response times operating with high-energy, long-duration waveforms may have reflected signal returns that are large enough to elicit undesirable responses from the adaptive processor. Such undesirable responses may produce signal cancellation and signal waveform distortion. Furthermore, jammer power level affects the array response in the main beam direction, thereby allowing blinking jammers to modulate the signal response and consequently degrade the performance of any subsequent coherent processing. The Frost [40] algorithm imposes constraints on the adaptive weights such that certain main beam properties are preserved. It turns out that the resulting constrained optimization system has two parts: (1) a preprocessing part called a "spatial correction filter," which compensates the signals for the misalignment between the plane wave front and the sensor array geometry; and (2) a signal processor that includes the adaptive weights and accounts for the primary function of the adaptive array.

The constrained LMS algorithm requires that the direction of arrival and a frequency band of interest be specified a priori for the appropriate constraint conditions to be imposed. Because the look direction frequency response relates to the adaptive weights, the algorithm maintains a selected frequency response in the look direction while simultaneously minimizing output noise power. If the look direction is perpendicular to the line of sensors, then identical signal components appear at the first taps [so $x_1(t) = x_2(t) = \cdots = x_N(t)$ in Figure 4-27] and propagate down the tapped delay lines following each sensor [so $x_{N+1}(t) = x_{N+2}(t) = \cdots = x_{2N}(t)$, and $x_{(J-1)N+1}(t) = x_{(J-1)N+2}(t) = \cdots = x_{NJ}(t)$]. Noise component waveforms arriving at the sensors from any direction other than the look direction will not usually produce equal voltage components at any vertical column of taps. Consequently, as far as the signal is concerned, the adaptive processor appears as an equivalent single tapped delay line in which each adaptive weight equals the sum of the weights in the corresponding vertical column of the original processor. These J summation weights in the equivalent tapped delay line are assigned a value to give the desired frequency response characteristic in the look direction, thereby giving rise to J constraint conditions. In the event that the look direction is other than that perpendicular

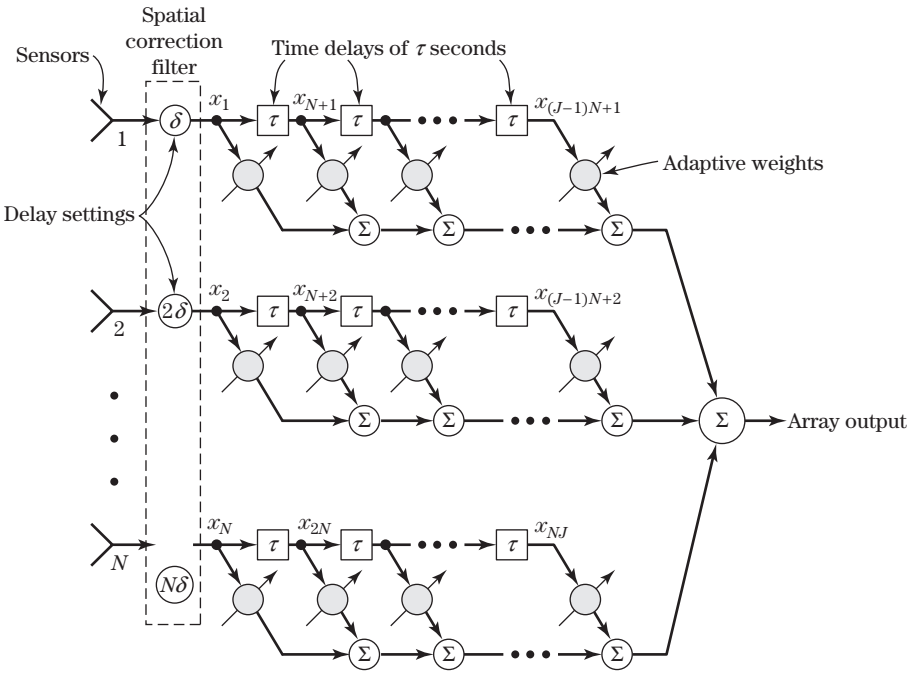


FIGURE 4-27 ■ Broadband adaptive array having N sensors and J adjustable weights per sensor.

to the line of sensors (as the previous discussion has assumed), then the time delays in the spatial correction filter are adjusted so the signal components of each channel at the output of the preprocessor are in phase.

The adaptive signal processor of Figure 4-27 has N sensors and J taps per sensor for a total of NJ adjustable weights. Using J constraints to determine the look direction frequency response leaves $NJ - J$ degrees of freedom to minimize the total array output power. Since the J constraints fix the look direction frequency response, minimizing the total output power is equivalent to minimizing the nonlook direction noise power (provided the signal voltages at the taps are uncorrelated with the corresponding noise voltages at these taps). If signal-correlated noise in the array is present, then part or all of the signal component of the array output may be cancelled. Although signal-correlated noise may not occur frequently, sources of such noise include multiple signal-propagation paths, and coherent radar or sonar "clutter."

It is desirable for proper noise cancellation that the noise voltages appearing at the adaptive processor taps be correlated among themselves (although uncorrelated with the signal voltages). Such noise sources may be generated by lightning, "jammers," noise from nearby vehicles, spatially localized incoherent clutter, and self-noise from the structure carrying the array. Noise voltages that are uncorrelated between taps (e.g., amplifier thermal noise) are partially rejected by the adaptive array either as the result of incoherent noise voltage addition at the array output or by reducing the weighting applied to any taps that may have a disproportionately large uncorrelated noise power.

4.8.1 Optimum Constrained Weight Vector Solution

The voltages appearing at each array tap in Figure 4-27 are sampled every Δ seconds (where Δ is a multiple of the delay τ between taps). The vector of tap voltages at the kth

sample is defined by

$$\mathbf{x}^T(k) \triangleq [x_1(k\Delta), x_2(k\Delta), \ldots, x_{NJ}(k\Delta)] \tag{4.337}$$

At any tap the voltages that appear may be regarded as the sums of voltages due to look direction signals s and nonlook direction noises n, so that

$$\mathbf{x}(k) = \mathbf{s}(k) + \mathbf{n}(k) \tag{4.338}$$

where the NJ-dimensional vector of look direction signals is defined by

$$\mathbf{s}(k) \triangleq \begin{bmatrix} s(k\Delta) \\ \vdots \\ s(k\Delta) \\ s(k\Delta - \tau) \\ \vdots \\ s(k\Delta - \tau) \\ \vdots \\ s[k\Delta - (J-1)\tau] \\ \vdots \\ s[k\Delta - (J-1)\tau] \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N \text{ taps} \end{matrix} \tag{4.339}$$

and the vector of nonlook direction noises is defined by

$$\mathbf{n}^T(k) \triangleq [n_1(k\Delta), n_2(k\Delta), \ldots, n_{NJ}(k\Delta)] \tag{4.340}$$

The weight vector appearing at each tap is denoted by $\mathbf{w}$, where

$$\mathbf{w}^T \triangleq [w_1, w_2, \ldots, w_{NJ}] \tag{4.341}$$

We assume that the signals and noises are zero-mean random processes with unknown second-order statistics. The covariance matrices of $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{n}$ are given by

$$E\{\mathbf{x}(k)\mathbf{x}^T(k)\} = \mathbf{R}_{xx} \tag{4.342}$$

$$E\{\mathbf{s}(k)\mathbf{s}^T(k)\} = \mathbf{R}_{ss} \tag{4.343}$$

$$E\{\mathbf{n}(k)\mathbf{n}^T(k)\} = \mathbf{R}_{nn} \tag{4.344}$$

Since the vector of look direction signals is assumed uncorrelated with the vector of nonlook direction noises

$$E\{\mathbf{n}(k)\mathbf{s}^T(k)\} = \mathbf{0} \tag{4.345}$$

Assume that the noise environment is such that $\mathbf{R}_{xx}$ and $\mathbf{R}_{nn}$ are positive definite and symmetric.

The adaptive array output (which forms the signal estimate) at the kth sample is given by

$$y(k) = \mathbf{w}^T\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{w} \tag{4.346}$$

From (4.346), it follows that the expected array output power is

$$E\{y^2(k)\} = E\{\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}\} = \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{4.347}$$

Suppose that the weights in the jth vertical column of taps sums to a selected number f_j. This constraint may be expressed by the relation

$$\mathfrak{c}_j^T\mathbf{w} = f_j, \quad j = 1, 2, \ldots, J \tag{4.348}$$

where the NJ-dimensional vector $\mathfrak{c}_j$ is given by

$$\mathfrak{c}_j = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \vdots \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \left.\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}\right\} j\text{th column of } N \text{ elements} \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \\ \vdots \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} N \end{bmatrix} \tag{4.349}$$

Now consider the requirement of constraining the entire weight vector to satisfy all J equations given by (4.348). Define a $J \times NJ$ constraint matrix $\mathbf{C}$ having $\mathfrak{c}_j$ as elements.

$$\mathbf{C} \triangleq [\mathfrak{c}_1 \cdots \mathfrak{c}_j \cdots \mathfrak{c}_J] \tag{4.350}$$

Furthermore, define $\mathbf{f}$ as the J-dimensional vector of summed weight values for each of the j vertical columns that yield the desired frequency response characteristic in the look direction as

$$\mathbf{f} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_J \end{bmatrix} \tag{4.351}$$

It immediately follows by inspection that the full set of constraints (4.348) can be written in matrix form as

$$\mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{4.352}$$

Now that the look direction frequency response is fixed by the constraint equation (4.352), minimizing the nonlook direction noise power is equivalent to minimizing the total output power given by (4.347). The constrained optimization problem reduces to

$$\underset{\mathbf{w}}{\text{Minimize}}\ \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} \tag{4.353}$$

$$\text{subject to } \mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{4.354}$$

Lagrange multipliers are used to find $\mathbf{w}_{\text{opt}}$ that satisfy (4.353) and (4.354) [41]. Adjoining the constraint equation (4.354) to the cost function (4.353) by a J-dimensional vector $\boldsymbol{\lambda}$, whose elements are undetermined Lagrange multipliers (and including a factor of $\frac{1}{2}$ to simplify the arithmetic), then yields

$$\underset{\mathbf{w}}{\text{Minimize}}\ \Re(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{xx}}\mathbf{w} + \lambda^{\mathrm{T}}[\mathbf{C}^{\mathrm{T}}\mathbf{w} - \mathbf{f}] \tag{4.355}$$

The gradient of (4.355) with respect to $\mathbf{w}$ is given by

$$_{\mathbf{w}}\Re(\mathbf{w}) = \mathbf{R}_{xx}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} \tag{4.356}$$

A necessary condition for (4.355) to be minimized is that the gradient be equal to zero so that

$$\mathbf{R}_{xx}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} = \mathbf{0} \tag{4.357}$$

Therefore, the optimal weight vector is given by

$$\mathbf{w}_{\text{opt}} = -\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda} \tag{4.358}$$

where the vector $\boldsymbol{\lambda}$ remains to be determined. The vector of Lagrange multipliers may now be evaluated from the constraint equation

$$\mathbf{C}^T\mathbf{w}_{\text{opt}} = \mathbf{f} = \mathbf{C}^T\left[-\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda}\right] \tag{4.359}$$

It then follows that the vector $\boldsymbol{\lambda}$ is given by

$$\boldsymbol{\lambda} = -\left[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right]^{-1}\mathbf{f} \tag{4.360}$$

where the existence of $[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}]^{-1}$ is guaranteed by the fact that $\mathbf{R}_{xx}$ is positive definite and $\mathbf{C}$ has full rank. Combining (4.358) and (4.360) then yields the optimum constrained weight vector

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{C}\left[\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right]^{-1}\mathbf{f} \tag{4.361}$$

If we substitute $\mathbf{w}_{\text{opt}}$ into (4.346), it follows that the constrained least squares estimate of the look direction signal provided by the array is

$$y_{\text{opt}}(k) = \mathbf{w}_{\text{opt}}^T\mathbf{x}(k) \tag{4.362}$$

If the vector of summed weight values $\mathbf{f}$ is selected so the frequency response characteristic in the look direction is all-pass and linear phase (distortionless), then the output of the constrained LMS signal processor is the maximum likelihood (ML) estimate of a stationary process in Gaussian noise (provided the angle of arrival is known) [42]. A variety of other optimal processors can also be obtained by a suitable choice of the vector $\mathbf{f}$ [43]. It is also worth noting that the solution (4.361) is sensitive to deviations of the actual signal direction from that specified by $\mathbf{C}$ and to various random errors in the array parameters [44].

4.8.2 The Adaptive Algorithm

Assume the adaptive processor must determine the correlation matrix $\mathbf{R}_{xx}$. It follows that in stationary environments during learning and in time-varying environments an estimate of the optimum adaptive processor weights must be periodically recomputed. The initial guess of an appropriate weight vector satisfies (4.354), so a good starting point is

$$\mathbf{w}(0) = \mathbf{C}[\mathbf{C}^T\mathbf{C}]^{-1}\mathbf{f} \tag{4.363}$$

where the quantity $\mathbf{C}[\mathbf{C}^T\mathbf{C}]^{-1}$ represents the pseudo-inverse of the singular matrix $\mathbf{C}^T$ [45]. For a gradient type algorithm, after the kth iteration the next weight vector is given by

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \Delta_s \quad {}_w\Re[\mathbf{w}(k)] \\ &= \mathbf{w}(k) - \Delta_s[\mathbf{R}_{xx}\mathbf{w}(k) + \mathbf{C}\lambda(k)]\end{aligned} \tag{4.364}$$

where Δ_s is the step size constant, and $\Re$ denotes the performance measure. Requiring $\mathbf{w}(k+1)$ to satisfy (4.352) then yields

$$\mathbf{f} = \mathbf{C}^T\mathbf{w}(k+1) = \mathbf{C}^T\{\mathbf{w}(k) - \Delta_s[\mathbf{R}_{xx}\mathbf{w}(k) + \mathbf{C}\lambda(k)]\} \tag{4.365}$$

Consequently, the Lagrange multipliers are given by

$$\begin{aligned}\lambda(k) = &-[\mathbf{C}^T\mathbf{C}]^{-1}\mathbf{C}^T\mathbf{R}_{xx}\mathbf{w}(k) - \frac{1}{\Delta_s}[\mathbf{C}^T\mathbf{C}]^{-1} \\ &\cdot[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)]\end{aligned} \tag{4.366}$$

Substituting (4.366) into (4.364) then gives the iterative relation

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) &- \Delta_s[I - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T]\mathbf{R}_{xx}\mathbf{w}(k) \\ &+ \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)]\end{aligned} \tag{4.367}$$

It is convenient to define the NJ-dimensional vector

$$\dagger \triangleq \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} \tag{4.368}$$

and the $NJ \times NJ$ matrix

$$\mathbf{P} \triangleq \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{4.369}$$

Then the iterative relation (4.369) may be rewritten as

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s\mathbf{R}_{xx}\mathbf{w}(k)] + \dagger \tag{4.370}$$

In the actual system the input correlation matrix is not known, and it is necessary to adopt some estimate of this matrix to insert in place of $\mathbf{R}_{xx}$ in the iterative weight adjustment equation. An approximation for $\mathbf{R}_{xx}$ at the kth iteration is merely the outer product of the tap voltage vector with itself: $\mathbf{x}(k)\mathbf{x}^T(k)$. Substituting this estimate of $\mathbf{R}_{xx}$ into (4.370) and recognizing that $y(k) = \mathbf{x}^T(k)\mathbf{w}(k)$ then yields the constrained LMS algorithm

$$\left.\begin{aligned} \mathbf{w}(0) &= \tilde{\mathbf{f}} \\ \mathbf{w}(k+1) &= \mathbf{P}[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)] + \tilde{\mathbf{f}} \end{aligned}\right\} \tag{4.371}$$

If it is merely desired to ensure that the complex response of the adaptive array system to a normalized signal input from the look direction is unity, then the spatial correction filter is dispensed with and the compensation for phase misalignment incorporated directly into the variable weight selection as suggested by Takao et al. [46]. Denote the complex response (amplitude and phase) of the array system by $Y(\theta)$, where θ is the angle measured from the normal direction to the array face. The appropriate conditions to impose on the adaptive weights are found by requiring that $\Re e\{Y(\theta)\} = 1$ and $Im\{Y(\theta)\} = 0$ when $\theta = \theta_c$, the look direction.

4.8.3 Conditions Ensuring Convergence to the Optimum Solution

The weight vector $\mathbf{w}(k)$ obtained by employing (4.371) is a random vector. Convergence of the mean value of the weight vector to the optimum is shown by considering the length of the difference vector between the mean of the actual weight vector and the optimum weight vector: convergence is assured if the length of the difference vector asymptotically approaches zero.

If we start with the weight adjustment equation

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s \mathbf{x}(k)y(k)] + \tilde{\mathbf{f}} \tag{4.372}$$

and recognize that $y(\mathbf{k}) = \mathbf{x}^T(k)\mathbf{w}(k)$, then taking the expected value of both sides of (4.372) yields

$$E[\mathbf{w}(k+1)] = \mathbf{P}\{E[\mathbf{w}(k)] - \Delta_s \mathbf{R}_{xx} E[\mathbf{w}(k)]\} + \tilde{\mathbf{f}} \tag{4.373}$$

Define the difference vector $\mathbf{v}(k+1)$ by

$$\mathbf{v}(k+1) \triangleq E\left[\mathbf{w}(k+1)\right] - \mathbf{w}_{\text{opt}} \tag{4.374}$$

Substitute (4.373) into (4.374) and use $\tilde{\mathbf{f}} = (\mathbf{I} - \mathbf{P})\mathbf{w}_{\text{opt}}$ and $\mathbf{P}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} = 0$ [which may be verified by direct substitution of (4.361) and (4.369)], then the difference vector satisfies

$$\mathbf{v}(k+1) = \mathbf{P}\mathbf{v}(k) - \Delta_s \mathbf{P}\mathbf{R}_{xx}\mathbf{v}(k) \tag{4.375}$$

Note from (4.369) that $\mathbf{P}$ is idempotent (i.e., $\mathbf{P}^2 = \mathbf{P}$), then premultiplying (4.375) by $\mathbf{P}$ reveals that $\mathbf{P}\mathbf{v}(k+1) = \mathbf{v}(k+1)$ for all k, so (4.375) can be rewritten as

$$\begin{aligned} \mathbf{v}(k+1) &= [\mathbf{I} - {}_s\mathbf{P}\mathbf{R}_{xx}\mathbf{P}]\mathbf{v}(k) \\ &= [\mathbf{I} - {}_s\mathbf{P}\mathbf{R}_{xx}\mathbf{P}]^{(k+1)}\mathbf{v}(0) \end{aligned} \tag{4.376}$$

From (4.376) it follows that the matrix $\mathbf{P}\mathbf{R}_{xx}\mathbf{P}$ determines both the rate of convergence of the mean weight vector to the optimum solution and the steady-state variance of the weight

vector about the optimum. The matrix $\mathbf{PR}_{xx}\mathbf{P}$ has J zero eigenvalues (corresponding to the column vectors of the constraint matrix $\mathbf{C}$) and $NJ - J$ nonzero eigenvalues σ_i, $i = 1, 2, \ldots, NJ - J$ [48]. The values of the $NJ - J$ nonzero eigenvalues are bounded by the relation

$$\lambda_{\min} \leq \sigma_{\min} \leq \sigma_i \leq \sigma_{\max} \leq \lambda_{\max} \tag{4.377}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of R_{xx}, respectively, and $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest nonzero eigenvalues of $\mathbf{PR}_{xx}\mathbf{P}$, respectively.

The initial difference vector $\mathbf{v}(0) = \tilde{\mathbf{f}} - \mathbf{w}_{\text{opt}}$ can be expressed as a linear combination of the eigenvectors of $\mathbf{PR}_{xx}\mathbf{P}$ corresponding to the nonzero eigenvalues [47]. Consequently, if $\mathbf{v}(0)$ is equals an eigenvector $\mathbf{e}_i$ of $\mathbf{PR}_{xx}\mathbf{P}$ corresponding to the nonzero eigenvalue σ_i, then

$$\begin{aligned} \mathbf{v}(k+1) &= [\mathbf{I} - \Delta_s \mathbf{PR}_{xx}\mathbf{P}]^{(k+1)}\mathbf{e}_i \\ &= [1 - \Delta_s \sigma_i]^{(k+1)}\mathbf{e}_i \end{aligned} \tag{4.378}$$

From (4.378) it follows that along any eigenvector of $\mathbf{PR}_{xx}\mathbf{P}$ the mean weight vector converges to the optimum weight vector geometrically with the geometric ratio $(1 - \Delta_s \sigma_i)$. Consequently, the time required for the difference vector length to decay to $1/e$ of its initial value is given by the time constant

$$\begin{aligned} \tau_i &= \frac{\Delta t}{\ln(1 - \Delta_s \sigma_i)} \\ &\cong \frac{\Delta t}{\Delta_s \sigma_i} \quad \text{if } \Delta_s \sigma_i \ll 1 \end{aligned} \tag{4.379}$$

where Δt denotes the time interval corresponding to one iteration.

If the step size constant Δ_s is selected so that

$$0 < \Delta_s < \frac{1}{\sigma_{\max}} \tag{4.380}$$

then the length (given by the norm) of any difference vector is bounded by

$$\begin{aligned} (1 - \Delta_s \sigma_{\max})^{(k+1)} \|\mathbf{v}(0)\| &\leq \|\mathbf{v}(k+1)\| \\ &\leq (1 - \Delta_s \sigma_{\min})^{(k+1)} \|\mathbf{v}(0)\| \end{aligned} \tag{4.381}$$

It immediately follows that if the initial difference vector length is finite, then the mean weight vector converges to the optimum so that

$$\lim_{k \to \infty} \|E\{\mathbf{w}(k)\} - \mathbf{w}_{\text{opt}}\| = 0 \tag{4.382}$$

where the convergence occurs with the time constants given by (4.379).

The LMS algorithm is designed to cope with nonstationary noise environments by continually adapting the weights in the signal processor. In stationary environments, however, this adaptation results in the weight vector exhibiting an undesirable variance about the optimum solution thereby producing an additional (above the optimum) component of noise to appear at the adaptive array output.

The optimum (minimum) output power level is given by

$$E\{y_{\text{opt}}^2(k)\} = \mathbf{w}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} = \mathbf{f}^T\left(\mathbf{C}^T\mathbf{R}_{xx}^{-1}\mathbf{C}\right)^{-1}\mathbf{f} \tag{4.383}$$

The additional noise caused by adaptively adjusting the weights can be compared with (4.383) to determine the penalty incurred by the adaptive algorithm. A direct measure of this penalty is the "misadjustment" M defined by (4.55). For a step size constant satisfying

$$0 < \Delta_s < \frac{1}{\sigma_{\max} + \frac{1}{2}\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})} \tag{4.384}$$

The steady-state misadjustment has been shown to be bounded by [48]

$$\frac{\Delta_s}{2} \cdot \frac{\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})}{1 - (\Delta_s/2)[\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P}) + 2\sigma_{\min}]} \leq M \leq \frac{\Delta_s}{2} \cdot \frac{\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P})}{1 - (\Delta_s/2)[\text{tr}(\mathbf{P}\mathbf{R}_{xx}\mathbf{P}) + 2\sigma_{\max}]} \tag{4.385}$$

If Δ_s is chosen to satisfy

$$0 < \Delta_s < \frac{2}{3\text{tr}(\mathbf{R}_{xx})} \tag{4.386}$$

then it will automatically also satisfy (4.384). It is also worth noting that the upper bound in (4.383) can be easily calculated directly from observations since $\text{tr}(\mathbf{R}_{xx}) = E\{\mathbf{x}^T(k)\mathbf{x}(k)\}$, the sum of the powers of the tap voltages.

4.8.4 A Useful Geometrical Interpretation

The constrained LMS algorithm (4.372) has a simple geometrical interpretation [40] that is useful for visualizing the error correcting property that prevents the weight vector from deviating from the constraint condition. Even unavoidable computational errors due to roundoff, truncation, or quantization are prevented from accumulating by the error correcting property, which continuously corrects for any errors that may occur, whatever their source may be.

In an error-free algorithm, the successive values of the NJ-dimensional weight vector $\mathbf{w}$ all exactly satisfy the constraint equation (4.354) and therefore all lie on a constraint plane $\mathbf{\Lambda}$ defined by

$$\mathbf{\Lambda} = \{\mathbf{w} : \mathbf{C}^T\mathbf{w} = \mathbf{f}\} \tag{4.387}$$

This constraint plane [which is $(NJ - J)$-dimensional] may be indicated diagramatically as shown in Figure 4-28.

Any vectors that point in a direction normal to the constraint plane are linear combinations of the constraint matrix column vectors and therefore all have the form $\mathbf{Ca}$, where $\mathbf{a}$ is a constant vector whose components determine the linear combination. Consequently, the initial weight vector in the algorithm (4.371), $\mathbf{f} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$, points in a direction normal to the constraint plane. In addition, the initial weight vector terminates exactly on the constraint plane since $\mathbf{C}^T\mathbf{f} = \mathbf{f}$. As a result, $\mathbf{f}$ is the shortest vector that can terminate on the constraint plane, as illustrated in Figure 4-28.

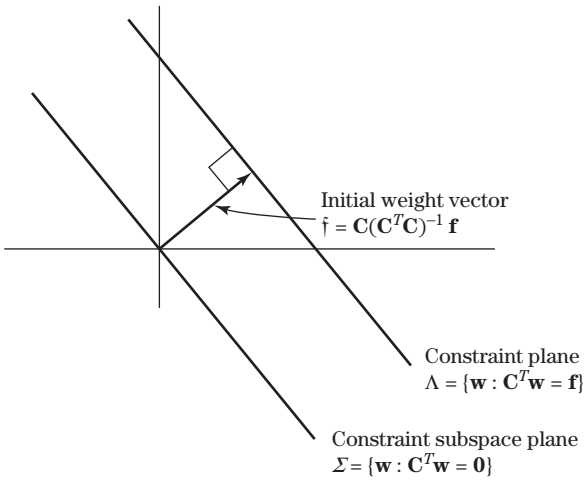


FIGURE 4-28 ■ Representation of the constraint plane, the constraint subspace plane, and initial weight vector in the **w**-space.

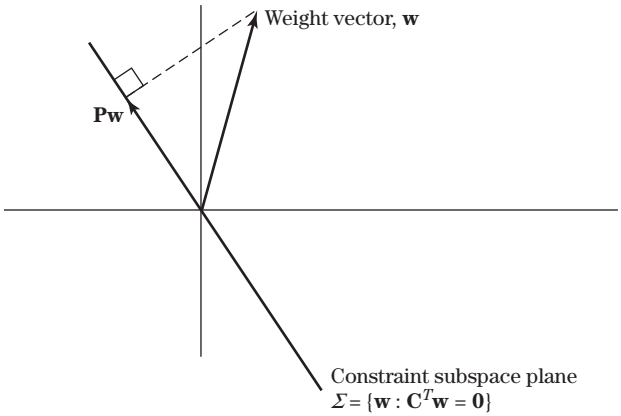


FIGURE 4-29 ■ Matrix **P** projects vectors onto the constraint subspace plane.

By setting the constraint weight vector **f** equal to zero, the homogeneous form of the constraint equation

$$\mathbf{C}^T\mathbf{w} = \mathbf{0} \tag{4.388}$$

defines a second plane [that is also $(NJ-J)$-dimensional] that passes through the coordinate space origin. This constraint subspace is depicted in Figure 4-28.

The constrained LMS algorithm (4.371) premultiplies a certain vector in the **W**-space by the matrix **P**, a projection operator. Premultiplication of any weight vector by the matrix **P** results in the elimination of any vector components perpendicular to the plane $\sum$, thereby projecting the original weight vector onto the constraint subspace plane as illustrated in Figure 4-29.

The only factor in (4.371) remaining to be discussed is the vector $y(k)\mathbf{x}(k)$, which is an estimate of the unconstrained gradient of the performance measure. Recall from (4.355) that the unconstrained performance measure is $\frac{1}{2}\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}$ and from (4.356) that the unconstrained gradient is given by $\mathbf{R}_{xx}\mathbf{w}$. Since the covariance matrix $\mathbf{R}_{xx}$ is unknown a priori, the estimate provided by $y(k)\mathbf{x}(k)$ is used in the algorithm.

The constrained optimization problem posed by (4.353) and (4.354) is illustrated diagramatically in the **w**-space as shown in Figure 4-30. The algorithm must succeed

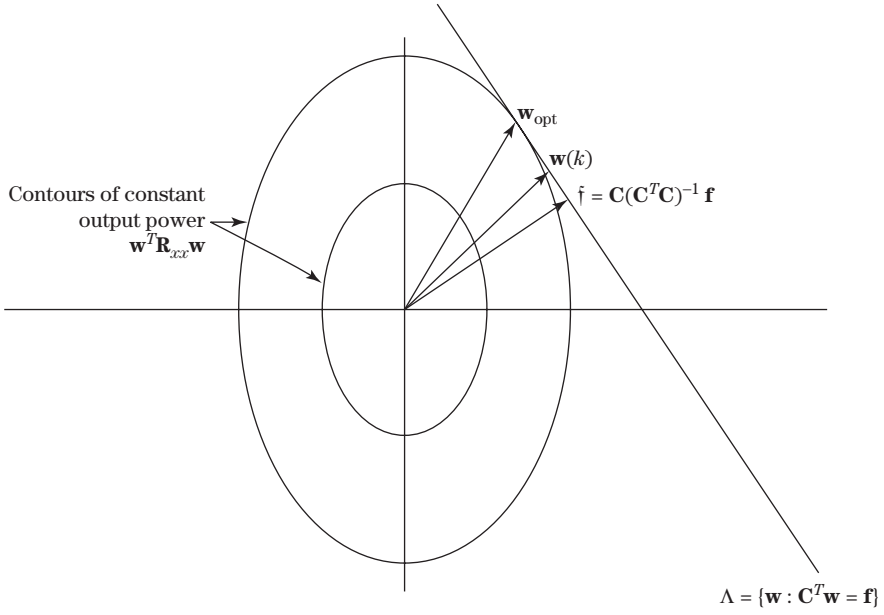


FIGURE 4-30 ■ Diagrammatic representation of the constrained optimization problem showing contours of constant output power, the constraint plane $\mathbf{\Lambda}$, the initial weight vector $\hat{\mathbf{f}}$, and the optimum constrained weight vector $\mathbf{w}_{\text{opt}}$ that minimizes the output power.

in moving from the initial weight vector $\hat{\mathbf{f}}$ to the optimum weight vector $\mathbf{w}_{\text{opt}}$ along the constraint plane $\mathbf{\Lambda}$. The operation of the constrained LMS algorithm (4.371) in solving the previously given constrained optimization problem is considered.

In Figure 4-31 the current value of the weight vector, $\mathbf{w}(k)$, is to be modified by taking the unconstrained negative gradient estimate $-y(k)\mathbf{x}(k)$, scaling it by Δ_s, and adding the result to $\mathbf{w}(k)$. In general, the resulting vector lies somewhere off the constraint plane. Premultiplying the vector $[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)]$ by the matrix $\mathbf{P}$, the projection onto the constraint subspace plane is obtained. Finally, adding $\hat{\mathbf{f}}$ to constraint subspace plane projection produces a new weight vector that lies on the constraint plane. This new

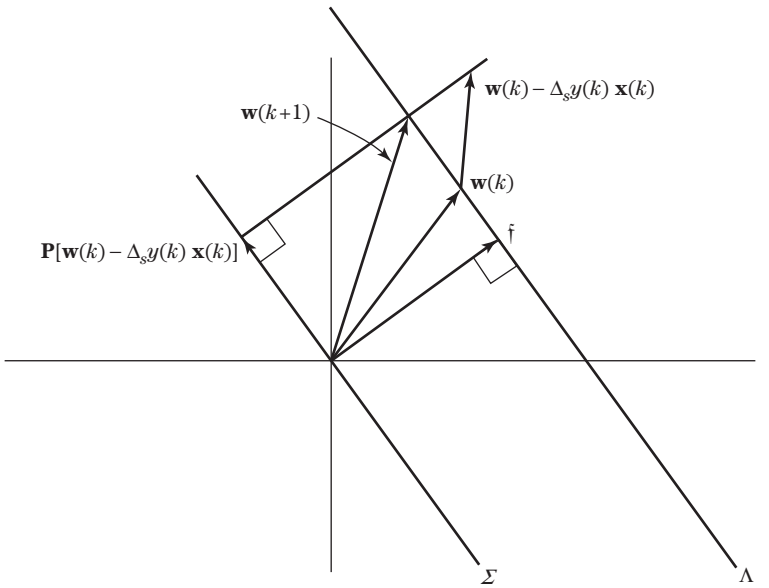


FIGURE 4-31 ■ Operation of the constrained LMS algorithm: $\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - \Delta_s y(k)\mathbf{x}(k)] + \hat{\mathbf{f}}$.

weight vector $\mathbf{w}(k+1)$ satisfies the constraint to within the numerical accuracy of the computations. This error-correcting feature of the constrained LMS algorithm prevents any computational errors from accumulating.

The convergence properties of the constrained LMS algorithm are closely related to those for the unconstrained LMS algorithm and have been previously discussed. Likewise, the same procedures that increased convergence speed for the LMS algorithm also work for the constrained LMS algorithm.

4.9 SIMULATION RESULTS

The fundamental misadjustment versus speed of adaptation trade-off is less favorable for the DSD algorithm than for the LMS algorithm [13]. Thus, it remains to determine the improvement in this fundamental trade-off that can be realized using the Powell accelerated gradient (PAG) algorithm compared with the LMS algorithm where eigenvalue spread in the input signal covariance matrix is present.

Figure 4-32 depicts a four-element Y array having $d = 0.787\lambda$ element spacing with the desired signal located at 0° and three distinct narrowband Gaussian jamming signals located at 15°, 90°, and 165°. The received signal covariance matrix is therefore given by

$$\frac{1}{n}\mathbf{R}_{xx} = \frac{s}{n}(\mathbf{u}\mathbf{u}^\dagger) + \sum_{i=1}^{3}\frac{J_i}{n}\left(\mathbf{v}_i\mathbf{v}_i^\dagger\right) + \mathbf{I} \tag{4.389}$$

where n denotes the thermal noise power (taken to be unity), s/n denotes the signal-to-thermal noise ratio, and J_i/n denotes the jammer-to-thermal noise ratios for each of the three jammers ($i = 1, 2, 3$). The elements of the signal steering vector $\mathbf{u}$ and the jammer steering vectors $\mathbf{v}_i$ are easily defined from the array geometry and the signal arrival angles. The desired signal is a biphase modulated signal having a phase angle of either 0° or 180° with equal probability at each sample.

Two signal conditions were simulated corresponding to two values of eigenvalue spread in the received signal covariance matrix. The first condition represents a respectable

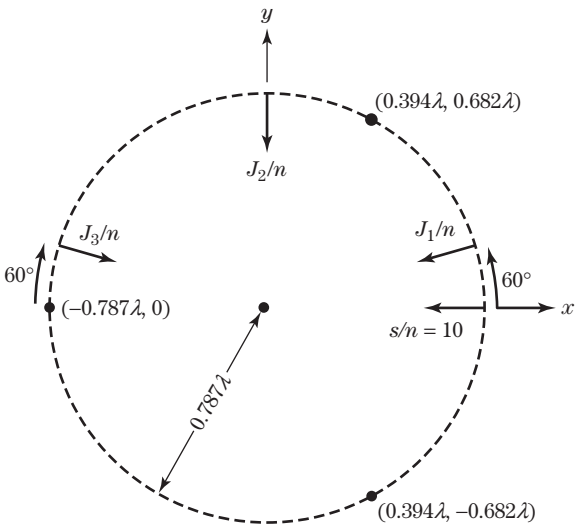


FIGURE 4-32 ■ Four-element Y-array geometry with signal and jammer locations for selected example.

eigenvalue spread of $\lambda_{max}/\lambda_{min} = 2440$, whereas the second condition represents a more extreme eigenvalue spread of $\lambda_{max}/\lambda_{min} = 16,700$. Choosing the jammer-to-thermal noise ratios to be $J_1/n = 500$, $J_2/n = 40$, and $J_3/n = 200$ together with $s/n = 10$ yields the corresponding eigenvalues $\lambda_1 = 2.44 \times 10^3$, $\lambda_2 = 4.94 \times 10^2$, $\lambda_3 = 25.62$, and $\lambda_4 = 1.0$ for which the optimum output SNR is $SNR_{opt} = 15.0$ (11.7 dB). Likewise, choosing the jammer-to-thermal noise ratios to be $J_1/n = 4000$, $J_2/n = 40$, and $J_3/n = 400$ along with $s/n = 10$ yields the eigenvalues $\lambda_1 = 1.67 \times 10^4$, $\lambda_2 = 10^3$, $\lambda_3 = 29$, and $\lambda_4 = 1.0$ for which the optimum output SNR is also $SNR_{opt} = 15.0$. In all cases the initial weight vector setting was taken to be $\mathbf{w}^T(0) = [0.1, 0, 0, 0]$. Figures 4-33 and 4-34 show the convergence results for the LMS and PAG algorithms, respectively, plotted as output SNR in decibels versus number of iterations for an eigenvalue spread of 2,440 (here output SNR means output signal-to-jammer plus thermal noise ratio). The expected value of the gradient and $\mathbf{v}^\dagger \mathbf{R}_{xx} \mathbf{v}$ required by the PAG algorithm was taken over $K = 9$ data samples, and one iteration of the PAG algorithm occurred every nine data samples,

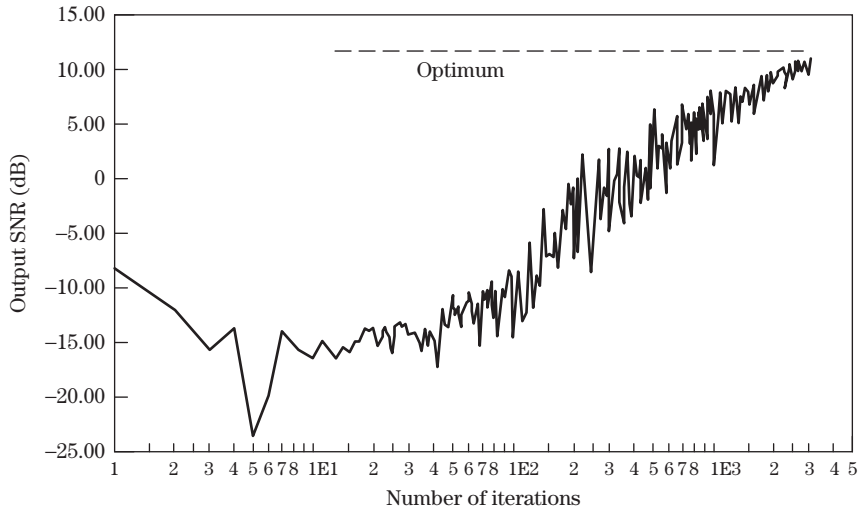


FIGURE 4-33 ■ Output SNR versus number of iterations for LMS algorithm with eigenvalue spread = 2440 and $\alpha_L = 0.1$.

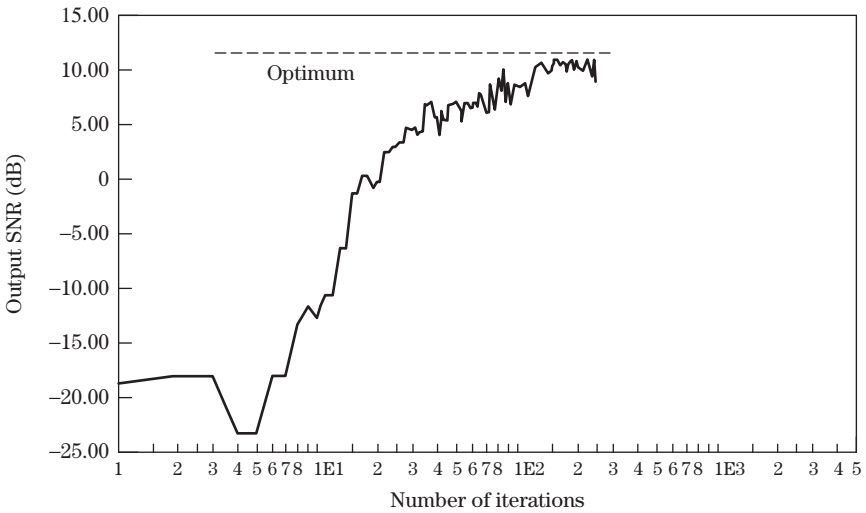


FIGURE 4-34 ■ Output SNR versus number of iterations for PAG algorithm with eigenvalue spread = 2,440 and $K = 9$.

even though a weight update does not occur on some iterations. The loop gain of the LMS loop was selected in accordance with (4.49), which requires that $\Delta_s \text{tr}(\mathbf{R}_{xx}) < 1$ for stability. Letting $\Delta_s \text{tr}(\mathbf{R}_{xx}) = \alpha_L$ and choosing $\alpha_L = 0.1$ therefore ensures stability while giving reasonably fast convergence with an acceptable degree of misadjustment error. As a consequence of the manner in which an iteration was defined for the PAG algorithm, the time scale for Figure 4-34 is nine times greater than the time scale for Figure 4-33. In Figure 4-34 the PAG algorithm is within 3 dB of the optimum after approximately 80 iterations (720 data samples), whereas in Figure 4-33 the LMS algorithm requires approximately 1500 data samples to reach the same point. Furthermore, it may be seen that the steady-state misadjustment for the two algorithms in these examples is very comparable so the PAG algorithm converges twice as fast as the LMS algorithm for a given level of misadjustment in this example.

Figures 4-35 and 4-36 show the convergence of the LMS and PAG algorithms for the same algorithm parameters as in Figures 4-33 and 4-34 but with the eigenvalue spread = 16,700. In Figure 4-36 the PAG algorithm is within 3 dB of the optimum after approx-

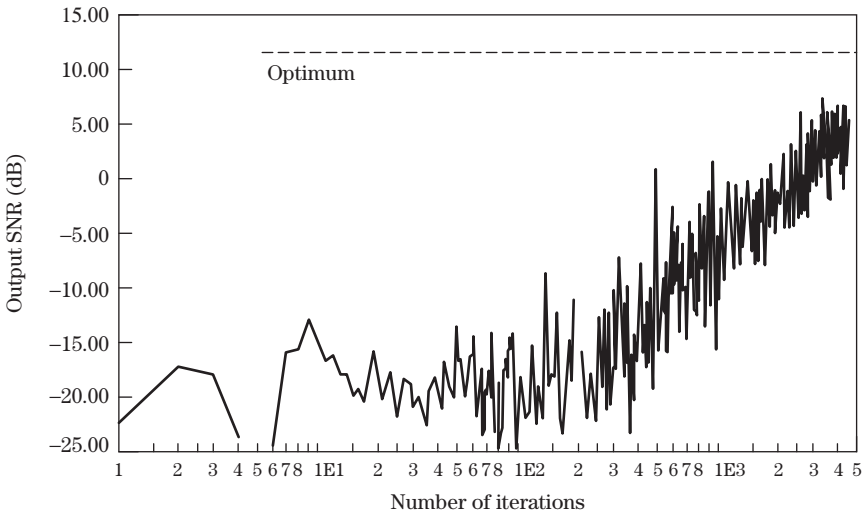


FIGURE 4-35 ■ Output SNR versus number of iterations for LMS algorithm with eigenvalue spread = 16,700 and $\alpha_L = 0.1$.

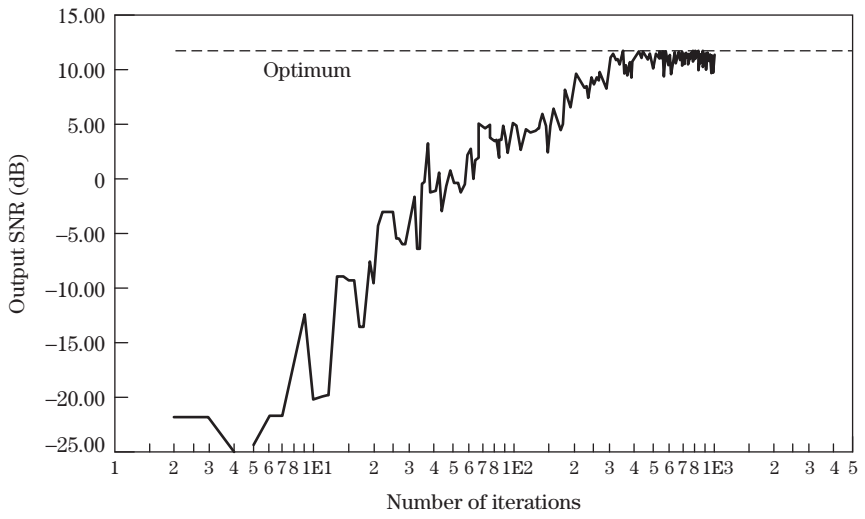


FIGURE 4-36 ■ Output SNR versus number of iterations for PAG algorithm with eigenvalue spread = 16,700 and $K = 9$.

imately 200 iterations (1,800 data samples), whereas the LMS algorithm in Figure 4-35 does not reach the same point even after 4,500 data samples. The degree of convergence speed improvement that is attainable therefore increases as the degree of eigenvalue spread increases.

A word of caution is needed concerning the expected convergence when using the PAG algorithm. The simulation results given here were compiled for an array having only four elements; as the number of array elements increases, the number of consecutive steps in orthogonal gradient directions also increases, thereby yielding significant direction errors in the later steps (since estimation errors accumulate over the consecutive step directions). Accordingly, for a given level of misadjustment the learning curve time constant does not increase linearly with N (as with LMS adaptation), but rather increases more rapidly. In fact, when $N > 10$, the PAG algorithm actually converges more slowly than the LMS algorithm.

4.10 | PHASE-ONLY ADAPTIVE NULLING USING STEEPEST DESCENT

A phased array may or may not have variable amplitude weights but always has phase shifters for beam steering and calibration. Since the phase shifters are already in place for beam steering purposes, they can also serve as adaptive elements to adjust the antenna pattern. The authors in [49] presented a beam-space algorithm for low sidelobe arrays that relies on small phase shifts. When the direction of arrival for all the interfering sources is known, then cancellation beams in the directions of the sources are subtracted from the original pattern. Adaptation consists of matching the peak of the cancellation beam with the culprit sidelobe and subtracting [50].

The steepest descent algorithm can also find phase settings that minimize the output power [51]. Multiplying the change in output power due to a change in phase at each element by a step size, μ, produces a gradient vector for the phase weights having components given by

$$\delta_n(\kappa + 1) = \delta_n(\kappa) + \mu \frac{P(\kappa) - P(\kappa - 1)}{\Delta(\kappa)} \tag{4.390}$$

where

$P(\kappa) =$ array output power at time step κ
$\delta_n(\kappa) =$ phase shift at element n
$\Delta(\kappa) =$ small phase increment

$$\mu = \frac{\Delta^2}{\sqrt{\sum_{n=1}^{N} [P(\kappa) - P(-1)]^2}}$$

This algorithm worked well for phase-only simultaneous nulling of the sum and difference patterns of an 80-element linear array of H plane sectoral horns [50]. A diagram of the array appears in Figure 2-25 of Chapter 2. The sum channel has a 30 dB low sidelobe Taylor taper and the difference channel has a 30 dB low sidelobe Bayliss taper. These channels share eight-bit beam steering phase shifters. Experiments used a CW signal incident on a sidelobe but no signal incident on the main beam. The cost function takes

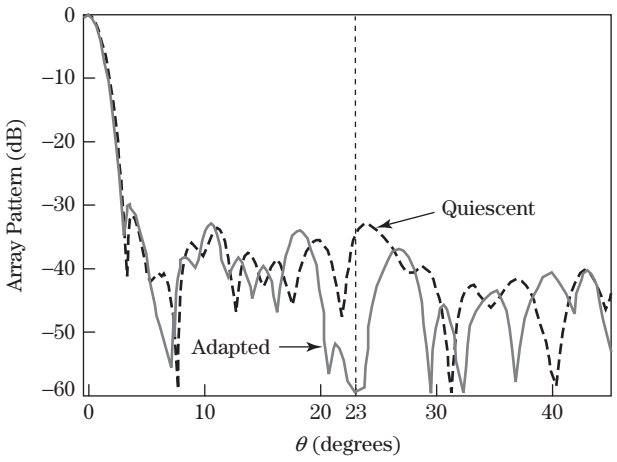


FIGURE 4-37 ■ Adapted sum pattern for simultaneous phase-only nulling in the sum and difference channels.

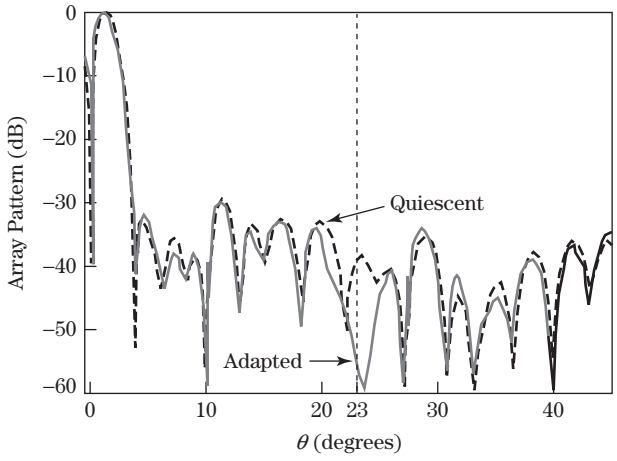


FIGURE 4-38 ■ Adapted difference pattern for simultaneous phase-only nulling in the sum and difference channels.

into consideration both the sum and difference channel output powers; otherwise, a null will not appear in both patterns. Minimizing the output power of both channels when an interfering signal appears at $\theta = 23^\circ$ results in the patterns shown in Figure 4-37 and Figure 4-38. The desired nulls are place with relatively small deviations from the quiescent patterns.

4.11 SUMMARY AND CONCLUSIONS

The LMS algorithm applies the method of steepest descent to the MSE performance measure to obtain a simple implementation that is particularly well suited to continuous signal communication systems. The LMS algorithm requires a reference signal that is compared with the array output to form an error signal. This technique is useful for adaptive arrays that are expected to distinguish between desired and undesired signals on the basis of differences in modulation characteristics. The heart of an LMS loop is the correlator (multiplier), which forms the product $e(t)x_i(t)$ that is required to obtain the estimated gradient. For an N-element array, N correlators are therefore required to implement the LMS algorithm to control each array element.

The Howells–Applebaum adaptive processor is generally employed in situations where the desired signal is usually absent (in contrast to the LMS algorithm, which requires that the desired signal be present) and makes use of a beam steering vector instead of a reference signal. We find that the Howells–Applebaum processor behavior is characterized by a quiescent mode (when the desired signal is absent) and an adapted mode (when the desired signal is present). The transient behavior of the algorithm is most easily described in terms of eigenvector beams, which can be analyzed by introduction of a transformation to a normal coordinate system that diagonalizes the signal covariance matrix. The processor exhibits the characteristic of sensitivity to eigenvalue spread, so that strong interference sources are canceled rapidly while weak interference sources are suppressed slowly. The dynamic range requirements of the circuitry used to implement the Howells–Applebaum processor can be reduced by introduction into the control loop of a hard limiter that modifies the effective signal covariance matrix but does not affect the eigenvalue spread.

Different methods for constraining the maximum SNR algorithm to maintain a favorable desired signal response are discussed:

1. The use of pilot signals
2. Preadaptation spatial filters
3. Control loop spatial filters
4. Discrimination constraint for known desired signal power level

The close relationship that exists between the Howells–Applebaum maximum SNR processor and the LMS algorithm makes the similar transient behavior characteristics of these two algorithms hardly surprising. The susceptibility of the algorithm performance to eigenvalue spread in the signal covariance matrix leads to a consideration of ways this susceptibility can be reduced. One way of reducing this susceptibility and maintaining fast transient response for all eigenvalue conditions is to employ a direct matrix inversion (DMI) algorithm, which is introduced in the next chapter.

For some practical applications it may be undesirable to require N correlators as the LMS algorithm does. In such cases, the alternative presented by the DSD algorithm, which requires only direct performance index measurements (error power measurements in the case of the MSE criterion), may be attractive. The DSD algorithm does not have as favorable a convergence speed versus misadjustment trade-off as the LMS algorithm, and both the DSD and LMS algorithms exhibit the same degree of convergence speed sensitivity to eigenvalue spread in the input signal covariance matrix.

One way of reducing the convergence speed sensitivity to eigenvalue spread is to employ an algorithm based on an accelerated gradient approach, provided the number of degrees of freedom of the array processor is not too high. An algorithm based on the Powell descent cycle was presented illustrating the improvement in the speed of convergence that can be realized. Accelerated gradient approaches have certain implementation drawbacks, however, and other methods (discussed in later chapters) may be preferred to obtain the desired reduction in convergence speed sensitivity to eigenvalue spread. In applications involving high energy, long-duration waveforms, it is often desirable to constrain the main beam of the array so that undesirable signal waveform distortion will not occur.

A nice summary of the LMS and Howells–Applebaum algorithms is given by [52]

$$\mathbf{w}_{n+1} = \begin{cases} (\mathbf{I} - \gamma \mathbf{R}_n)\mathbf{w}_n + \gamma \mu \mathbf{b}^* & \text{Howells–Applebaum} \\ (\mathbf{I} - \gamma \mathbf{R}_n)\mathbf{w}_n + \gamma \mathbf{r}_{xd} & \text{LMS} \end{cases} \tag{4.391}$$

4.12 PROBLEMS

1. ***Misadjustment-Speed of Adaptation Trade-off for the LMS and DSD Algorithms [13]*** For the LMS algorithm the total misadjustment in the steady state is given by (4.83), whereas the total (minimum) misadjustment for the DSD algorithm is given by (4.317).

(a) Assuming all eigenvalues are equal so that $(T_{p_{\text{MSE}}})_{\text{av}} = T_{\text{MSE}}$ and that $M = 10\%$ for the LMS algorithm, plot T_{MSE} versus N for $N = 2, 4, 8, \ldots, 512$.

(b) Assuming all eigenvalues are equal so that $(T_{p_{\text{MSE}}})_{\text{av}} = T_{\text{MSE}}$ and that $(M_{\text{tot}})_{\min} = 10\%$ for the DSD algorithm, plot T_{MSE} versus N for $N = 2, 4, 8, \ldots, 512$ and compare this result with the previous diagram obtained in part (a).

2. ***Reference Signal Generation for LMS Adaptation Using Polarization as a Desired Signal Discriminant [53]*** LMS adaptation requires a reference signal to be generated having properties sufficiently correlated either to the desired signal or the undesired signal to permit the adaptive system to preserve the desired signal in its output. Usually, the desired signal waveform properties (e.g., frequency, duration, type of modulation, signal format) are used to generate the reference signal, but if the signal and the interference can be distinguished by polarization, then polarization may be employed as a useful discriminant for reference signal generation.

Let s denote a linearly polarized desired signal having the known polarization angle θ, and let n denote a linearly polarized interference signal having the polarization angle α (where it is known only that $\alpha \neq \theta$). Assume that the desired signal and interference impinge on two linearly polarized antennas (A and B) as shown in Figure 4-39 where the antennas differ in orientation by the angle β. The two signals v_a and v_b may then be expressed as

$$v_a = s\cos\theta + n\cos\alpha$$
$$v_b = s\cos(\beta - \theta) + n\cos(\beta - \alpha)$$

(a) Show that by introducing the weight w_1 as illustrated in Figure 4-39, then the signal $v_b' = v_b - w_1 v_a$ can be made to be signal free (have zero desired signal content) by setting

$$w_1 = \frac{\cos(\beta - \theta)}{\cos\theta}$$

so that

$$v_b' = n\frac{\sin\beta}{\cos\theta}\sin(\alpha - \theta) = nf(\alpha, \beta, \theta)$$

(b) From the results of part (a), show that

$$v_0 = v_a - w_2 v_b' = s\cos\theta + n[\cos\alpha - w_2 f(\alpha, \beta, \theta)]$$

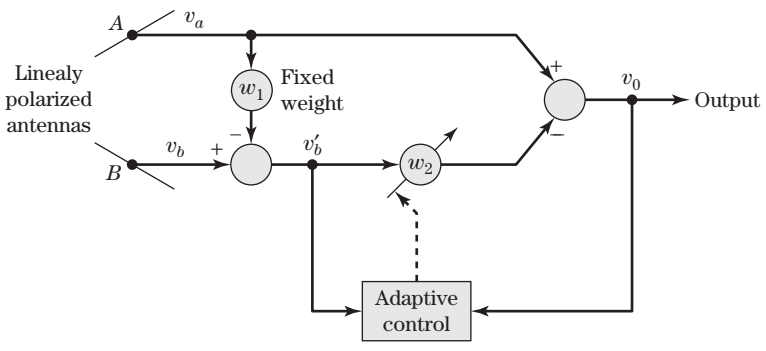


FIGURE 4-39 ■ Adaptive array configuration for interference rejection on the basis of polarization using LMS adaptation.

Since the output signal v_0 contains both desired signal and interference components, correlating it with the signal-free voltage v_b' yields a measure of the interference remaining in the output signal, and the adaptive weight w_2 can then be adjusted to reduce the interference content in the output.

(c) The error in the output signal v_0 is the interference signal component that is still present after $w_2 b_b'$ is subtracted from v_a. Assume that the interference and the desired signal are uncorrelated; then

$$E\left\{v_0^2\right\} = E\{s^2\}\cos^2\theta + E\{n^2\}[\cos\alpha - w_2 f(\alpha, \beta, \theta)]^2$$

If the rate of change of w_2 is proportional to $\partial E\left\{v_0^2\right\}/\partial w_2$, show that the final value of the weight occurs when $\partial E\left\{v_0^2\right\}/\partial w_2 = 0$ so that

$$w_2 = \frac{\cos\alpha}{f(\alpha, \beta, \theta)}$$

(d) With w_2 set to the final value determined in part (c), show that the steady-state system output is given by

$$v_0 = s\cos\theta$$

thereby showing that the system output is free of interference under steady-state conditions. The previous result assumes that (1) knowledge of θ and the setting of w_1 are error free; (2) the circuitry is noiseless; And (3) the number of input signals equals the number of antennas available. These ideal conditions are not met in practice, and [53] analyzes the system behavior under nonideal operating conditions.

3. ***Relative Sensitivity of the Constrained Look-Direction Response Processor to Perturbed Wavefronts [44]*** The solution to the problem of minimizing the expected output power of an array $\eta = E\{\mathbf{w}^\dagger \mathbf{x}\mathbf{x}^\dagger \mathbf{w}\}$ subject to $\mathbf{x}_0^\dagger \mathbf{w} = f$ (or equivalently, $\eta = f^2$) is given by (4.366). Since the look direction response is constrained by $\mathbf{x}_0^\dagger \mathbf{w} = f$ where $\mathbf{x}_0$ denotes a plane wave signal arriving from the angle θ_0, the rationale behind this constraint is to regard the processor as a filter that will pass plane waves from the angle θ_0 but attenuate plane waves from all other directions.

Let a perturbed plane wave be represented by $\mathbf{x}$, having components

$$\mathbf{x}_k = A_{k_0}(1 + \alpha_k)\exp[j(\phi_{k_0} + \xi_k)]$$

where α_k represents amplitude deviations, and ξ_k represents phase deviations from the nominal plane wave signal $\mathbf{x}_0$. Assume that α_k, ξ_k are all uncorrelated zero-mean Gaussian random variables with variances $\sigma_\alpha^2, \sigma_\xi^2$ at each sensor of the array.

(a) Using $\eta = \mathbf{w}^\dagger E\{\mathbf{x}\mathbf{x}^\dagger\}\mathbf{w}$ and the fact that

$$E\{x_i x_j^*\} = x_{i_0} x_{j_0}^* \exp\left(-\sigma_\xi^2\right) \quad \text{for } i \neq j$$

and

$$E\left\{x_i x_j^*\right\} = |x_{i_0}|^2\left(1 + \sigma_\alpha^2\right) \quad \text{for } i = j$$

show that

$$\eta = \exp\left(-\sigma_\xi^2\right)\mathbf{w}^\dagger \mathbf{x}_0 \mathbf{x}_0^\dagger \mathbf{w} + \left[1 - \exp\left(-\sigma_\xi^2\right) + \sigma_\alpha^2\right]\mathbf{w}^\dagger \mathbf{w}$$

or $\eta \cong f^2 + \left(\sigma_\xi^2 + \sigma_\alpha^2\right)\mathbf{w}^\dagger\mathbf{w}$ for small values of $\sigma_\xi^2, \sigma_\alpha^2$ assuming that $|x_{i_0}|^2 = 1$ (which is the case for a planar wave).

(b) The result in part (a) can be rewritten as

$$\eta = f^2\left[1 + \mathfrak{s}\left(\sigma_\xi^2 + \sigma_\alpha^2\right)\right]$$

where

$$\mathfrak{s} \triangleq \frac{\mathbf{w}^\dagger\mathbf{w}}{f^2}$$

Consequently, the ratio $\mathfrak{s}$ can be regarded as the relative sensitivity of the processor to the perturbations whose variances are $\sigma_\xi^2, \sigma_\alpha^2$. Using the weights given by (4.169), show that

$$\mathfrak{s} = \frac{\mathbf{x}_0^\dagger\mathbf{R}_{xx}^{-2}\mathbf{x}_0}{\left(\mathbf{x}_0^\dagger R_{xx}^{-1}\mathbf{x}_0\right)^2}$$

The previous relative sensitivity can become large if the eigenvalues of $\mathbf{R}_{xx}$ have a large spread, but if the eigenvalues of $\mathbf{R}_{xx}$ have a small spread then $\mathfrak{s}$ cannot become large.

4. ***MSLC Relationships [29]*** Show that (4.219) results from (4.215) by the following:

(a) Substitute (4.217) and (4.218) into (4.219).

(b) Let $\mathbf{K} = \mathbf{I} + g\mathbf{R}_{nn}$.

(c) Apply the matrix inversion lemma [(D.10) of Appendix D] to the resulting expression.

5. ***MSLC Relationships [29]*** Show that (4.234) follows from the steady-state relationship given by (4.232).

6. ***Control Loop Spatial Filter Relationships [29]*** Apply the matrix inversion identity

$$[\mathbf{Q} + \mathbf{e}\mathbf{f}^T]^{-1}\mathbf{e} = \frac{\mathbf{Q}^{-1}\mathbf{e}}{1 + \mathbf{f}^T\mathbf{Q}^{-1}\mathbf{e}}$$

where $\mathbf{Q}$ is a nonsingular $N \times N$ matrix and $\mathbf{e}$ and $\mathbf{f}$ are $\mathbf{N} \times 1$ vectors to (4.247), and show that (4.248) results.

7. ***Control Loop Spatial Filter Relationships [29]*** By substituting the relationships expressed by (4.248) and (4.249) into (4.247), show that the steady-state weight vector relationship given by (4.250) results.

8. ***Control Loop Spatial Filter Relationships [29]*** To show that (4.267) can be developed from (4.265), define the ratio

$$SN' \triangleq \frac{\mathbf{w}'^\dagger\mathbf{s}'\mathbf{s}'^\dagger\mathbf{w}'}{\mathbf{w}'^\dagger\mathbf{R}'_{xx}\mathbf{w}'}$$

where

$$\mathbf{w}' = \begin{bmatrix} -\mathbf{w} \\ \hline 1 \end{bmatrix}, \quad \mathbf{s}' = \begin{bmatrix} \mathbf{s} \\ \hline s_0 \end{bmatrix}$$

and

$$\mathbf{R}'_{xx} = \left[\begin{array}{c|c} \mathbf{R}_{xx} & \mathbf{r}_{xx_0} \\ \hline \mathbf{r}_{xx_0}^\dagger & P_0 \end{array}\right]$$

(a) Show that $|s_0 - \mathbf{w}^\dagger\mathbf{s}|^2 = \mathbf{w}'^\dagger\mathbf{s}'\mathbf{s}'^\dagger\mathbf{w}'$.

(b) Show that $P_e \triangleq \mathbf{w}'^{\dagger}\mathbf{R}'_{xx}\mathbf{w}' = \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} - \mathbf{w}^{\dagger}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}w + P_0$.

(c) Since $\mathbf{w} = [\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{r}_{xx_0}$ from (4.263) show that

$$\mathbf{w} = \mathbf{w}_{\text{opt}} + \Delta\mathbf{w}$$

where $\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0}$ (the Wiener solution), and $\Delta\mathbf{w} = -a\mathbf{R}^{-1}_{xx}\mathbf{w}$.

(d) Substitute $\mathbf{w} = \mathbf{w}_{\text{opt}} + \Delta\mathbf{w}$ into P_e from part (b) and show that

$$P_e = P_{e_0} + \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\Delta\mathbf{w}$$

where

$$\begin{aligned} P_{e_0} &= P_0 - \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}\mathbf{w}_{\text{opt}} + \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} \\ &= P_0 - \mathbf{r}^{\dagger}_{xx_0}\mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0} \\ &= P_0 - \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} \end{aligned}$$

Hint: Note that

$$\begin{aligned} &\Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} + \mathbf{w}^{\dagger}_{\text{opt}}\mathbf{R}_{xx}\Delta\mathbf{w} \\ &\quad - \Delta\mathbf{w}^{\dagger}\mathbf{r}_{xx_0} - \mathbf{r}^{\dagger}_{xx_0}\Delta\mathbf{w} = 0 \end{aligned}$$

because

$$\begin{aligned} \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} &= \Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{R}^{-1}_{xx}\mathbf{r}_{xx_0} \\ &= \Delta\mathbf{w}^{\dagger}\mathbf{r}_{xx_0} \end{aligned}$$

(e) Show that

$$\Delta\mathbf{w}^{\dagger}\mathbf{R}_{xx}\Delta\mathbf{w} = a^2\sum_{i=1}^{N}\frac{|(\mathbf{Q}\mathbf{r}_{xx_0})_i|^2}{\lambda_i(\lambda_i + a)^2}$$

by using $\Delta\mathbf{w} = -a\mathbf{R}^{-1}_{xx}\mathbf{w}$.

Hint: Note that

$$\mathbf{r}^{\dagger}_{xx_0}\mathbf{Q}^{-1}\mathbf{Q}[\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{R}^{-1}_{xx}\mathbf{Q}\mathbf{Q}^{-1}\cdot[\mathbf{R}_{xx} + a\mathbf{I}]^{-1}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{r}_{xx_0}$$

is composed entirely of diagonalized matrices since

$$\mathbf{Q}\mathbf{R}^{-1}_{xx}\mathbf{Q}^{-1} = \mathbf{\Lambda} \text{ and } \mathbf{Q}\mathbf{Q}^{-1} = \mathbf{I}$$

9. ***Performance Degradation Due to Errors in the Assumed Direction of Signal Incidence [54]***
The received signal vector can be represented by

$$\mathbf{x}(t) = \mathbf{s}(t) + \sum_{i=2}^{m}\mathbf{g}_i(t) + \mathbf{n}(t)$$

where

$$\begin{aligned} \mathbf{s}(t) &= \text{desired signal vector} = s(t)\mathbf{v}_1 \\ \mathbf{g}_i(t) &= \text{directional noise sources} = g_i(t)\mathbf{v}_i \end{aligned}$$

and

$$\mathbf{n}(t) = \text{thermal noise vector comprised of narrowband Gaussian noise components independent from one sensor element to the next.}$$

The vectors $\mathbf{v}_i, i = 1, \ldots, m$ can be regarded as steering vectors where

$$\mathbf{v}_i^T = [\exp(-j\omega_c\tau_{i1}), \exp(-j\omega_c\tau_{i2}), \ldots, \exp(-j\omega_c\tau_{iN})]$$

and τ_{ik} represents the delay of the ith directional signal at the kth sensor relative to the geometric center of the array; ω_c is the carrier signal frequency.

The optimum weight vector should satisfy

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}$$

where $\mathbf{R}_{xx}$ is the received signal covariance matrix, and $\mathbf{r}_{xd}$ is the cross-correlation vector between the desired signal s and the received signal vector $\mathbf{x}$. Direction of arrival information is contained in $\mathbf{r}_{xd}$, and if the direction of incidence is assumed known, then $\mathbf{r}_{xd}$ can be specified and only $\mathbf{R}_{xx}^{-1}$ need be determined to find $\mathbf{w}_{\text{opt}}$. If the assumed direction of incidence is in error, however, then $\mathbf{w} = \mathbf{R}_{xx}^{-1}\tilde{\mathbf{r}}_{xd}$ where $\tilde{\mathbf{r}}_{xd}$ represents the cross-correlation vector computed using the errored signal steering vector $\tilde{\mathbf{v}}_1$.

(a) For the foregoing signal model, the optimum weight vector can be written as $\mathbf{w}_{\text{opt}} = [S\mathbf{v}_1\mathbf{v}_1^\dagger + \mathbf{R}_{nn}]^{-1} \cdot (S\mathbf{v}_1)$, where $\mathbf{R}_{nn}$ denotes the noise covariance matrix, and S denotes the desired signal power per sensor. If $\mathbf{v}_1$ is in error, then $\tilde{\mathbf{r}}_{xd} = (S\tilde{\mathbf{v}}_1)$. Show that the resulting weight vector computed using $\tilde{\mathbf{r}}_{xd}$ is given by

$$\mathbf{w} = \frac{S}{1 + S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1}\left[\left(1 + S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1\right)\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1 - S\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1\mathbf{R}_{nn}^{-1}\mathbf{v}_1\right]$$

(b) Using the result obtained in part (a), show that the output signal-to-noise power ratio (when only the desired signal and thermal noise are present) from the array is given by

$$\left(\frac{S}{N}\right)_{\text{out}} = \frac{\mathbf{w}^\dagger E\{\mathbf{s}\mathbf{s}^\dagger\}\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}} = \frac{S\mathbf{w}^\dagger\left(\mathbf{v}_1\mathbf{v}_1^\dagger\right)\mathbf{w}}{\mathbf{w}^\dagger\mathbf{R}_{nn}\mathbf{w}}$$

$$= \frac{S|\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2}{\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1 - 2S|\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2 + \mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1[S^2\{(\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\mathbf{v}_1)^* \times (\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1) - |\mathbf{v}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1|^2\} + 2S\tilde{\mathbf{v}}_1^\dagger\mathbf{R}_{nn}^{-1}\tilde{\mathbf{v}}_1]}$$

(c) Use the fact that $\mathbf{R}_{nn} = \sigma^2\mathbf{I}$ and the result of part (b) to show that

$$\left(\frac{S}{N}\right)_{\text{out}} = \frac{S\left(\frac{N}{\sigma^2}\right)\frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}}{\left(1 + \frac{NS}{\sigma^2}\right)^2\left[1 - \frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}\right] + 2\left(\frac{NS}{\sigma^2}\right)\left[1 - \frac{\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right|^2}{N^2}\right]}$$

(d) Show for a uniform linear array that

$$\left|\mathbf{v}_1^\dagger\tilde{\mathbf{v}}_1\right| = \frac{|\sin[(N\pi d/\lambda_c)\sin\tilde{\theta}]|^2}{|\sin[(\pi d/\lambda_c)\sin\tilde{\theta}]|^2}$$

where d represents the separation between sensors, and $\tilde{\theta}$ represents the angular uncertainty from boresight.

4.13 REFERENCES

[1] B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, edited by R. E. Kalman and N. DeClaris, New York, Holt, Rinehart and Winston, 1970, pp. 563–587.

[2] F. R. Ragazzini and G. F. Franklin, *Sampled-Data Control Systems*, New York, McGraw-Hill, 1958.

[3] E. I. Jury, *Sampled-Data Control Systems*, New York, Wiley, 1958.

[4] J. T. Tou, *Digital and Sampled-Data Control Systems*, New York, McGraw-Hill, 1959.

[5] B. C. Kuo, *Discrete-Data Control Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1970.

[6] B. Widrow, "Adaptive Filters I: Fundamentals," Stanford Electronics Laboratories, Stanford, CA, Rept. SEL-66-126 (Tech. Rept. 6764-6), December 1966.

[7] J. S. Koford and G. F. Groner, "The Use of an Adaptive Threshold Element to Design a Linear Optimal Pattern Classifier," *IEEE Trans. Inf. Theory*, Vol. IT-12, January 1966, pp. 42– 50.

[8] K. Steinbuch and B. Widrow, "A Critical Comparison of Two Kinds of Adaptive Classification Networks," *IEEE Trans. Electron. Comput. (Short Notes)*, Vol. EC-14, October 1965, pp. 737–740.

[9] F. W. Smith, "Design of Quasi-Optimal Minimum-Time Controllers," *IEEE Trans. Autom. Control*, Vol. AC-11, January 1966, pp. 71–77.

[10] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, No. 12, December 1967, pp. 2143–2159.

[11] L. J. Griffiths, "Signal Extraction Using Real-Time Adaptation of a Linear Multichannel Filter," Ph.D. Disseration, Stanford University, December 1967.

[12] K. D. Senne and L. L. Horowitz, "New Results on Convergence of the Discrete-Time LMS Algorithm Applied to Narrowband Adaptive Arrays," *Proceedings of the 17th IEEE Conference on Decision and Control,* San Diego, CA, January 10–12, 1979, pp. 1166–1167.

[13] B. Widrow and J. M. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 615–638.

[14] R. T. Compton, Jr., "An Adaptive Array in a Spread-Spectrum Communication System," *Proc. IEEE*, Vol. 66, No. 3, March 1978, pp. 289–298.

[15] D. M. DiCarlo and R. T. Compton, Jr., "Reference Loop Phase Shift in Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-14, No. 4, July 1978, pp. 599–607.

[16] P. W. Howells, "Intermediate Frequency Side-Lobe Canceller," U.S. Patent 3202990, August 24, 1965.

[17] S. P. Applebaum, "Adaptive Arrays," Syracuse University Research Corp., Report SPL TR 66-1, August 1966.

[18] L. E. Brennan and I. S. Reed, "Theory of Adaptive Radar," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-9, No. 2, March 1973, pp. 237–252.

[19] L. E. Brennan and I. S. Reed, "Adaptive Space-Time Processing in Airborne Radars," Technology Service Corporation, Report TSC-PD-061-2, Santa Monica, CA, February 24, 1971.

[20] A. L. McGuffin, "Adaptive Antenna Compatibility with Radar Signal Processing," in *Proceedings of the Array Antenna Conference,* February 1972, Naval Electronics Laboratory, San Diego, CA.

[21] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 2, March 1971, pp. 254–262.

[22] W. F. Gabriel, "Adaptive Arrays—An Introduction," *Proc. IEEE*, Vol. 64, No. 2, February 1976, pp. 239–272.

[23] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control Loop Noise in Adaptive Array Antennas," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 2, March 1971, pp. 254–262.

[24] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. of IEEE,* Vol. 57, No. 10, October 1969, pp. 1696–1704.

[25] R. T. Compton, Jr., *Adaptive Antennnas Concepts and Performance,* Englewood Cliffs, NJ, Prentice-Hall, Inc., 1988.

[26] M. W. Ganz, "Rapid Convergence by Cascading Applebaum Adaptive Arrays," *IEEE Trans. AES,* Vol. 30, No. 2, April 1994, pp. 298–306.

[27] A. J. Berni, "Weight Jitter Phenomena in Adaptive Control Loops," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-14, No. 4, July 1977, pp. 355–361.

[28] L. E. Brennan and I. S. Reed, "Effect of Envelope Limiting in Adaptive Array Control Loops," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-7, No. 4, July 1971, pp. 698–700.

[29] S. P. Applebaum and D. J. Chapman, "Adaptive Arrays with Main Beam Constraints," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 650– 662.

[30] R. T. Compton, Jr., "Adaptive Arrays: On Power Equalization with Proportional Control," Ohio State University, Columbus, Quarterly Report 3234-1, Contract N0019-71-C-0219, December 1971.

[31] C. L. Zahm, "Application of Adaptive Arrays to Suppress Strong Jammers in the Presence of Weak Signals," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-9, No. 2, March 1973, pp. 260–271.

[32] B. W. Lindgren, *Statistical Theory*, New York, Macmillan, 1960, Ch. 5.

[33] M. R. Hestenes and E. Stiefel, "Method of Conjugate Gradients for Solving Linear Systems," *J. Res. Natl. Bur. Stand.*, Vol. 29, 1952, p. 409.

[34] J. D. Powell, "An Iterative Method for Finding Stationary Values of a Function of Several Variables," *Comput. J. (Br.)*, Vol. 5, No. 2, July 1962, pp. 147–151.

[35] R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," *Comput. J. (Br.)*, Vol. 6, No. 2, July 1963, pp. 163–168.

[36] R. Fletcher and C. M. Reeves, "Functional Minimization by Conjugate Gradients," *Comput. J. (Br.)*, Vol. 7, No. 2, July 1964, pp. 149–154.

[37] D. G. Luenberger, *Optimization by Vector Space Methods*, New York, Wiley, 1969, Ch. 10.

[38] L. Hasdorff, *Gradient Optimization and Nonlinear Control*, New York, Wiley, 1976.

[39] L. S. Lasdon, S. K. Mitter, and A. D. Waren, "The Method of Conjugate Gradients for Optimal Control Problems," *IEEE Trans. Autom. Control*, Vol. AC-12, No. 2, April 1967, pp. 132–138.

[40] O. L. Frost, III, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proc. IEEE*, Vol. 60, No. 8, August 1972, pp. 926–935.

[41] A. E. Bryson, Jr. and Y. C. Ho, *Applied Optimal Control*, Waltham, MA, Blaisdell, 1969, Ch. 1.

[42] E. J. Kelly and M. J. Levin, "Signal Parameter Estimation for Seismometer Arrays," Massachusetts Institute of Technology, Lincoln Laboratories Technical Report 339, January 1964.

[43] A. H. Nuttall and D. W. Hyde, "A Unified Approach to Optimum and Suboptimum Processing for Arrays," U.S. Navy Underwater Sound Laboratory, New London, CT, USL Report 992, April 1969.

[44] R. N. McDonald, "Degrading Performance of Nonlinear Array Processors in the Presence of Data Modeling Errors," *J. Acoust. Soc. Am.*, Vol. 51, No. 4, April 1972, pp. 1186–1193.

[45] D. Q. Mayne, "On the Calculation of Pseudoinverses," *IEEE Trans. Autom. Control*, Vol. AC-14, No. 2, April 1969, pp. 204–205.

[46] K. Takao, M. Fujita, and T. Nishi, "An Adaptive Antenna Array Under Directional Constraint," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 662–669.

[47] O. L. Frost, III, "Adaptive Least Squares Optimization Subject to Linear Equality Constraints," Stanford Electronics Laboratories, Stanford, CA, DOC. SEL-70-053, Technical Report TR6796-2, August 1970.

[48] J. L. Moschner, "Adaptive Filtering with Clipped Input Data," Stanford Electronics Laboratories, Stanford, CA, Dec. SEL-70-053, Technical Report TR6796-1, June 1970.

[49] C. Baird, and G. Rassweiler, "Adaptive Sidelobe Nulling Using Digitally Controlled Phase-Shifters," *IEEE Transactions on Antennas and Propagation,* Vol. 24, No. 5, 1976, pp. 638–649.

[50] H. Steyskal, "Simple Method for Pattern Nulling by Phase Perturbation," *IEEE Transactions on Antennas and Propagation,* Vol. 31, No. 1, 1983, pp. 163–166.

[51] R. L. Haupt, "Adaptive Nulling in Monopulse Antennas," *IEEE Transactions on Antennas and Propagation,* Vol. 36, No. 2, 1988, pp. 202–208.

[52] D. S. De Lorenzo, J. Gautier, J. Rife, P. Enge, and D. Akos, "Adaptive Array Processing for GPS Interference Rejection," *Proc. ION GNSS 2005*, pp. 618–627.

[53] H. S. Lu, "Polarization Separation by an Adaptive Filter," *IEEE Trans. Aerosp. Electron. Sys.*, Vol. AES-9, No. 6, November 1973, pp. 954–956.

[54] C. L. Zahm, "Effects of Errors in the Direction of Incidence on the Performance of an Adaptive Array," *Proc. IEEE*, Vol. 60, No. 8, August 1972, pp. 1008–1009.

CHAPTER 5

Direct Inversion of the Sample Covariance Matrix

Chapter Outline

The usefulness of an adaptive array often depends on its convergence rate. For example, when adaptive radars simultaneously reject jamming and clutter while providing automatic platform motion compensation, then rapid convergence to steady-state solutions is essential. Convergence of adaptive sensor arrays using the popular maximum signal-to-noise ratio (SNR) or least mean squares (LMS) algorithms depend on the eigenvalues of the noise covariance matrix. When the covariance matrix eigenvalues differ by orders of magnitude, then convergence is exceedingly long and highly example dependent. One way to speed convergence and circumvent the convergence rate dependence on eigenvalue distribution is to directly compute the adaptive weights using the sample covariance matrix of the signal environment [1–3].

5.1 THE DIRECT MATRIX INVERSION (DMI) APPROACH

The signals impinging on the receiving elements of an N-element adaptive array are represented by the N-dimensional signal vector $\mathbf{x}$, whose associated covariance matrix is given by

$$\mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^{\dagger}\} \tag{5.1}$$

When the desired signal is absent, then only noise and interference are present and

$$\mathbf{R}_{xx} = \mathbf{R}_{nn} \tag{5.2}$$

When the desired signal is present, then from Chapter 3 the optimal weight vector solution is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{5.3}$$

where $\mathbf{r}_{xd}$ is the cross-correlation between the random vector $\mathbf{x}(t)$ and the reference signal $d(t)$. When the desired signal is absent, then the optimal weight vector solution is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{nn}^{-1}\mathbf{r}_{xd} = \mathbf{R}_{nn}^{-1}\mathbf{b}^{*} \tag{5.4}$$

where $\mathbf{b}^{*}$ is the vector of beam steering signals matched to the target Doppler frequency and angle of incidence. Note that specifying $\mathbf{r}_{xd}$ is equivalent to specifying $\mathbf{b}^{*}$.

If the signal, clutter, and interference situation are known a priori, then the covariance matrix is evaluated and the optimal solution for the adaptive weights is computed directly using either (5.3) or (5.4). In practice the signal, clutter, and interference situation are not known a priori, and furthermore the interference environment frequently changes due to the presence of moving near-field scatterers, antenna motion, interference, and jamming. Consequently, the adaptive processor continually updates the weight vector to respond to the changing environment. In the absence of detailed a priori information, the weight vector is updated using estimates of $\mathbf{R}_{xx}$ or $\mathbf{R}_{nn}$, and $\mathbf{r}_{xd}$ from a finite observation interval and substituting into (5.3) or (5.4). This method for implementing the adaptive processor is referred to as the DMI or sample matrix inversion (SMI) technique. The estimates of $\mathbf{R}_{xx}$, $\mathbf{R}_{nn}$, and $\mathbf{r}_{xd}$ are based on the maximum likelihood (ML) principle, which yields unbiased estimates having minimum variance [4]. Although an algorithm based on DMI theoretically converges faster than the LMS or maximum SNR algorithms, the covariance matrix could be ill conditioned, so the degree of eigenvalue spread also affects the practicality of this approach.

It is worth noting that when the covariance matrix to be inverted has the form of a Toeplitz matrix (a situation that arises when using tapped-delay line channel processing), then the matrix inversion algorithm of W. F. Trench [5] can be exploited to facilitate the computation. The convergence results discussed in this chapter assume that all computations are done with sufficient accuracy to overcome the effects of any ill-conditioning and therefore represent an upper limit on how well any DMI approach can be expected to perform.

5.1.1 Use of the Sample Covariance Matrix

Suppose that the cross-correlation vector $\mathbf{r}_{xd}$ (or equivalently that the beam steering vector $\mathbf{b}^{*}$) is known. The optimal weight vector estimate $\hat{\mathbf{w}}$ is formed by using

$$\hat{\mathbf{w}}_1 = \hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd} \tag{5.5}$$

and assuming $\mathbf{x}(t)$ contains the desired signal, where $\hat{\mathbf{R}}_{xx}$ is the sample covariance estimate of $\mathbf{R}_{xx}$, or by using

$$\hat{\mathbf{w}}_2 = \hat{\mathbf{R}}_{nn}^{-1}\mathbf{r}_{xd} = \hat{\mathbf{R}}_{nn}^{-1}\mathbf{b}^{*} \tag{5.6}$$

if we assume $\mathbf{x}(t)$ does not contain the desired signal where $\hat{\mathbf{R}}_{nn}$ is the sample covariance estimate of $\mathbf{R}_{nn}$.

The array output SNR using $\hat{\mathbf{w}}_1$ or $\hat{\mathbf{w}}_2$ is written as

$$\left(\frac{s}{n}\right)_1 = \frac{\hat{\mathbf{w}}_1^\dagger \mathbf{s}\mathbf{s}^\dagger \hat{\mathbf{w}}_1}{\hat{\mathbf{w}}_1^\dagger \mathbf{R}_{nn} \hat{\mathbf{w}}_1} = \frac{\mathbf{r}_{xd}^\dagger \hat{\mathbf{R}}_{xx}^{-1} \mathbf{s}\mathbf{s}^\dagger \hat{\mathbf{R}}_{xx}^{-1} \mathbf{r}_{xd}}{\mathbf{r}_{xd}^\dagger \hat{\mathbf{R}}_{xx}^{-1} \mathbf{R}_{nn} \hat{\mathbf{R}}_{xx}^{-1} \mathbf{r}_{xd}} \tag{5.7}$$

$$\left(\frac{s}{n}\right)_2 = \frac{\mathbf{r}_{xd}^\dagger \hat{\mathbf{R}}_{nn}^{-1} \mathbf{s}\mathbf{s}^\dagger \hat{\mathbf{R}}_{nn}^{-1} \mathbf{r}_{xd}}{\mathbf{r}_{xd}^\dagger \hat{\mathbf{R}}_{nn}^{-1} \mathbf{R}_{nn} \hat{\mathbf{R}}_{nn}^{-1} \mathbf{r}_{xd}} \tag{5.8}$$

where $\mathbf{s}$ denotes the desired signal vector component of $\mathbf{x}$ [it will be recalled from Chapter 3 that $\mathbf{s}(t) = s(t)\mathbf{v}$]. The SNR $(s/n)_2$ has meaning only during time intervals when a desired signal is present; the weight adjustment in this case takes place when the desired signal is absent. The "rate of convergence" of the two algorithms (5.5) and (5.6) depends on the output SNR normalized to the optimum output SNR, SN_o, compared with the number of independent signal samples K used to obtain the required sample covariance matrices.

Assuming that all signals present at the array input are modeled as sample functions from zero-mean Gaussian processes, then an ML estimate of $\mathbf{R}_{xx}$ (or $\mathbf{R}_{nn}$ when the desired signal is not present) is formed using the sample covariance matrix given by

$$\hat{\mathbf{R}}_{xx} = \frac{1}{K} \sum_{j=1}^{K} \mathbf{x}(j)\mathbf{x}^\dagger(j) \tag{5.9}$$

where $\mathbf{x}(j)$ denotes the jth time sample of the signal vector $\mathbf{x}(t)$. Note that the assumption of independent zero-mean samples implies $E[\mathbf{x}(i)\mathbf{x}^\dagger(j)] = 0$ for $i \neq j$.

Since each element of the matrix $\hat{\mathbf{R}}_{xx}$ is a random variable, the output SNR is also a random variable. It is instructive to compare the actual SNR obtained using $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ of (5.5) and (5.6) with the optimum SNR obtained using (5.3) and (5.4) ($SN_o = \mathbf{s}^\dagger \mathbf{R}_{nn}^{-1} \mathbf{s}$), by forming the normalized SNR as follows:

$$\rho_1 = \frac{(s/n)_1}{SN_o} \tag{5.10}$$

$$\rho_2 = \frac{(s/n)_2}{SN_o} \tag{5.11}$$

It can be shown [2] that the probability distribution of ρ_2 is described by the incomplete beta distribution given by

$$Pr(\rho_2 \leq y) = \frac{K!}{(N-2)!(K+1-N)!} \int_0^y (1-u)^{N-2} u^{K+1-N} du \tag{5.12}$$

where

K = total number of independent time samples used in obtaining $\hat{\mathbf{R}}_{nn}$
N = number of adaptive degrees of freedom

The probability distribution function of (5.12) contains important information concerning the convergence of the DMI algorithm that is easily seen by considering the mean and the variance of ρ_2. From (5.12) it follows that the average value of ρ_2 is given by

$$E\{\rho_2\} = \overline{\rho}_2 = \frac{K+2-N}{K+1} \tag{5.13}$$

The variance of ρ_2 is given by

$$\text{var}(\rho_2) = \frac{(K+2-N)(N-1)}{(K+1)^2(K+2)} \tag{5.14}$$

For fixed K and N, (5.14) suggests that var(ρ_2) is independent of the amount of noise the system must contend with and the eigenvalue spread of the noise covariance matrix. Recalling that ρ is a normalized SNR, however, we see that both the actual SNR (s/n) and the optimum SNR SN_o are affected in the same way by any noise power increase, so the normalized ratio remains the same, and the variance of the normalized ratio likewise remains unchanged. Eigenvalue spread has no effect on (5.14), since this expression assumes that the sample matrix inversion is computed exactly. As a result, (5.14) contains only the sample covariance matrix estimation errors. The effect of eigenvalue spread on the matrix inversion computation is addressed in a later section.

A plot of (5.13) in Figure 5-1 (we assume that N is significantly larger than 2) shows that so long as $K \geq 2N$, the loss in $\overline{\rho}_2$ due to nonoptimum weights is less than 3 dB. This result leads to the convenient rule of thumb that the number of time samples required to obtain a useful sample covariance matrix (when the desired signal is absent) is twice the number of adaptive degrees of freedom.

Next, consider the convergence behavior when the signal is present while estimating $\mathbf{w}$ with (5.5). Rather than attempt to derive the probability distribution function of ρ_1 directly, it is more convenient to exploit the results obtained for ρ_2 by defining the random variable

$$\rho_1' = \frac{\mathbf{r}_{xd}^{\dagger}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{s}\mathbf{s}^{\dagger}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}}{\mathbf{r}_{xd}^{\dagger}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{R}_{xx}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}\mathbf{s}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{s}} \tag{5.15}$$

which has the same probability distribution function as that of ρ_2. Knowing the statistical properties of ρ_1' and the relationship between ρ_1' and ρ_1 then enables the desired information about ρ_1 to be easily obtained. It can be shown that the relationship between ρ_1' and ρ_1 is

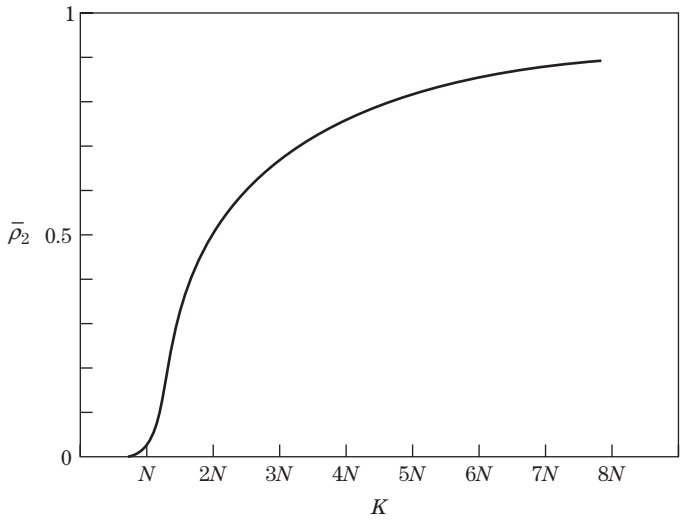


FIGURE 5-1 ■ Plot of $\overline{\rho}_2$ versus K of (5.13). We assume $N \gg 2$.

given by [3]

$$\rho_1 = \frac{\rho_1'}{SN_o(1-\rho_1')+1} \tag{5.16}$$

Since ρ_1' has the same probability distribution function as ρ_2, it immediately follows that

$$E\{\rho_1\} < E\{\rho_2\} \tag{5.17}$$

and

$$\lim_{SN_o \to 0} E\{\rho_1\} = E\{\rho_2\} \tag{5.18}$$

The inequality expressed by (5.17) implies that, on the average, the output SNR achieved by using $\hat{\mathbf{w}}_1 = \hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$ is less than the output SNR achieved using $\hat{\mathbf{w}}_2 = \hat{\mathbf{R}}_{nn}^{-1}\mathbf{r}_{xd}$ (except in the limit as $K \to \infty$, in which case both estimates are equally accurate). This behavior derives from the fact that the presence of the desired signal increases the time required (or number of samples required) to obtain accurate estimates of $\mathbf{R}_{xx}$ from the sample covariance matrix compared with the time required to obtain accurate estimates of $\mathbf{R}_{nn}$ when the desired signal is absent. The limit expressed by (5.18) indicates that for $SN_o < 1$, the difference in SNR performance obtained using $\hat{\mathbf{w}}_1$ or $\hat{\mathbf{w}}_2$ is negligible.

The mean of ρ_1 in (5.16) can be expressed as the following infinite series [3]:

$$E\{\rho_1\} = \frac{a}{a+b}\left\{1 + \sum_{i=1}^{\infty}(-SN_o)^i \left(\frac{b}{a+b+1}\right)\left(\frac{b+1}{a+b+2}\right)\cdots\left(\frac{i+b-1}{a+b+i}\right)\right\} \tag{5.19}$$

where $a = K - N + 2$ and $b = N - 1$. The manner in which $E\{\rho_1\}$ depends on SN_o is illustrated in Figure 5-2 for $N = 4$ with K as an independent parameter.

Since $\hat{\mathbf{w}}_2$ significantly outperforms $\hat{\mathbf{w}}_1$ when $SN_o \gg 1$, it is advantageous to remove the signal components from $\mathbf{x}(t)$ before forming $\hat{\mathbf{R}}_{xx}$. When the desired signal vector component $\mathbf{s}(t)$ is completely known, then subtracting $\mathbf{s}(t)$ from the received signal vector

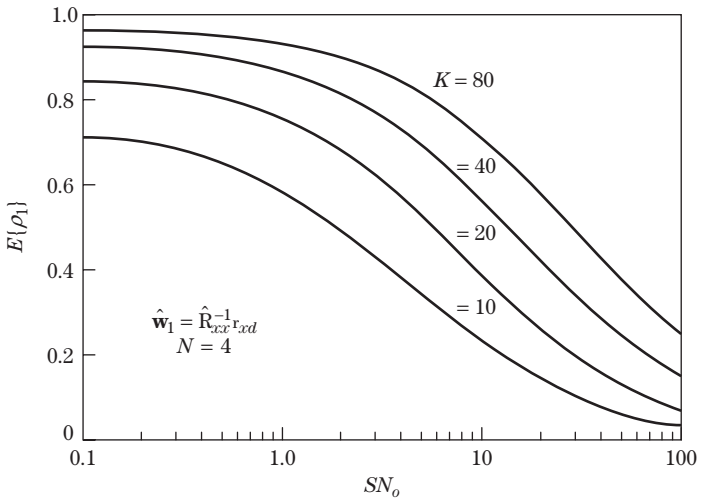


FIGURE 5-2 ■ The average normalized output SNR versus optimum output SNR for several sample sizes.

$\mathbf{x}(t)$ enables a desired signal-free noise vector to be formed that can be used to generate the sample noise covariance matrix $\hat{\mathbf{R}}_{nn}$. An improper procedure that is occasionally suggested for eliminating the desired signal component is to form the sample covariance matrix $\hat{\mathbf{D}}$, where

$$\hat{\mathbf{D}} = \hat{\mathbf{R}}_{xx} - \mathbf{s}\mathbf{s}^{\dagger} \tag{5.20}$$

so that

$$E\{\hat{\mathbf{D}}\} = \mathbf{R}_{nn} \tag{5.21}$$

The procedure suggested by (5.20) is unsatisfactory for obtaining the fast convergence associated with $\hat{\mathbf{w}}_2$ because even though $E\{\hat{\mathbf{D}}\} = \mathbf{R}_{nn}$, the weight vector estimate obtained using $\hat{\mathbf{w}}_3 = \hat{\mathbf{D}}^{-1}\mathbf{r}_{xd}$ results only in an estimate that is a scalar multiple of $\hat{\mathbf{w}}_1 = \hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$ and therefore has the associated convergence properties of $\hat{\mathbf{w}}_1$. This fact may easily be seen by forming

$$\begin{aligned}\hat{\mathbf{w}}_3 &= \hat{\mathbf{D}}^{-1}\mathbf{r}_{xd} \\ &= [\hat{\mathbf{R}}_{xx} - \mathbf{s}\mathbf{s}^{\dagger}]^{-1}\mathbf{r}_{xd} \\ &= \frac{1}{1 + \mathbf{s}^{\dagger}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{s}}\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}\end{aligned} \tag{5.22}$$

The coefficient of $\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$ in (5.22) is a scalar, and therefore the output SNR resulting from the use of $\hat{\mathbf{w}}_3$ is identical to that obtained using $\hat{\mathbf{w}}_1$, so no transient performance improvement results. If any transient response improvement were to be possible, it would be necessary for $\hat{\mathbf{w}}_3$ of (5.22) to be a scalar multiple of $\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$. Since $\hat{\mathbf{w}}_3 = \alpha\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$, however, the transient response of $\hat{\mathbf{w}}_3$ is the same as that of $\mathbf{w}_1$, and a priori knowledge of $\mathbf{s}$ as employed in (5.20) does not improve the DMI algorithm response.

5.1.2 Use of the Sample Covariance Matrix and the Sample Cross-Correlation Vector

In many practical radar and communications systems the cross-correlation vector $\mathbf{r}_{xd}$ (or the beam-steering vector $\mathbf{b}^*$) is not known a priori. An alternative approach determines the optimal weight vector using

$$\hat{\mathbf{w}}_4 = \hat{\mathbf{R}}_{xx}^{-1}\hat{\mathbf{r}}_{xd} \tag{5.23}$$

where $\hat{\mathbf{r}}_{xd}$ is the sample cross-correlation vector given by

$$\hat{\mathbf{r}}_{xd} = \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}(j)d^*(j) \tag{5.24}$$

The transient behavior of the DMI algorithm represented by $\hat{\mathbf{w}}_4$ of (5.23) is different from that found for $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ of (5.5) and (5.6), respectively.

As an example, consider a four-element uniform array with $\lambda/2$ spacing. The desired signal is incident at 0° and one interference signal is incident at 45° with $\sigma_{noise} = 0.01$. The amplitude and phase of the weights as a function of sample are shown in Figure 5-3 and Figure 5-4. The adapted pattern after 500 samples is shown in Figure 5-5. The weights do not vary much after 50 samples. A null appears where a sidelobe peak used to be.

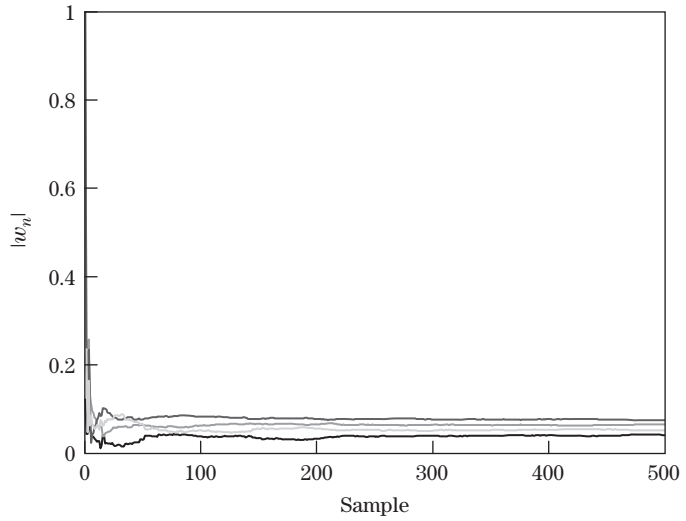


FIGURE 5-3 ■ Magnitude of the DMI weights versus sample.

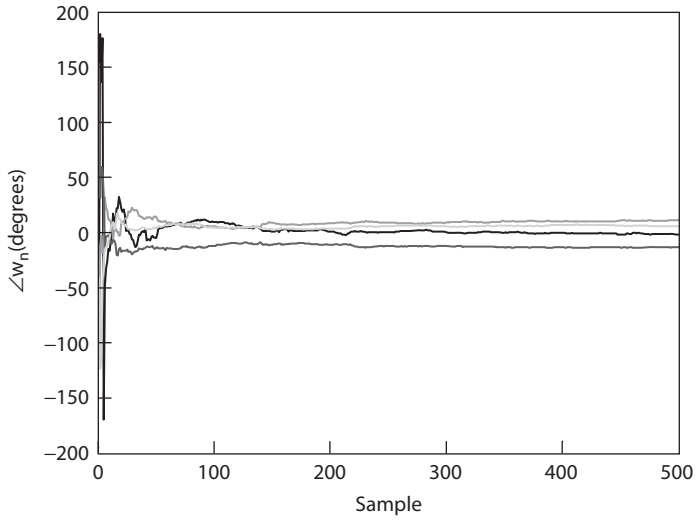


FIGURE 5-4 ■ Phase of the DMI weights versus sample.

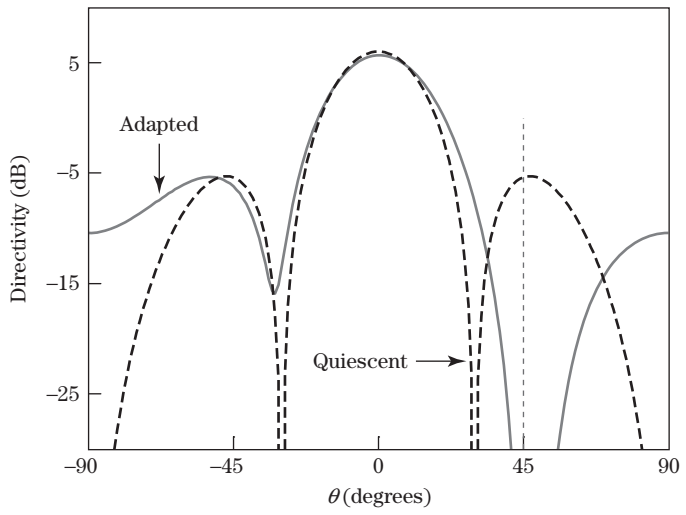


FIGURE 5-5 ■ The DMI adapted pattern for the four-element array after 500 samples.

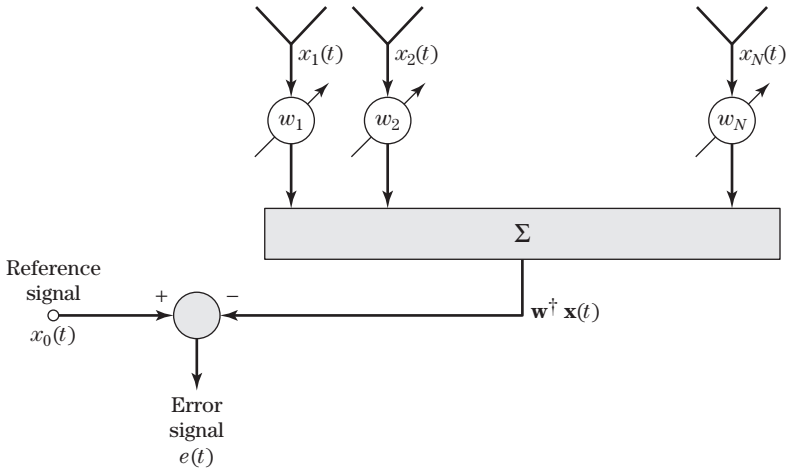


FIGURE 5-6 ■ Adaptive array configuration for considering transient behavior of $\hat{\mathbf{w}}_4$.

The transient response characteristics of $\hat{\mathbf{w}}_4$ are determined by considering the least squares estimate of $\mathbf{w}_{\text{opt}}$ based on K independent samples of the input vector $\mathbf{x}$. It is convenient in posing this problem to assume the adaptive array configuration shown in Figure 5-6. This configuration represents two important adaptive array structures in the following manner. If $x_0(t) = d(t)$, the reference signal representation of the desired signal, then the minimum mean square error (MMSE) and the maximum output SNR are both given by the Wiener solution $\mathbf{w}_{\text{opt}}$. Likewise, if $x_0(t)$ represents the output of a reference antenna (usually a high-gain antenna pointed in the direction of the desired signal source), then the configuration represents a coherent sidelobe canceller (CSLC) system, for which obtaining the MMSE weight vector solution minimizes the output error (or residue) power and hence minimizes the interference power component of the array output.

The transient response of $\hat{\mathbf{w}}_4$ given by (5.23) is characterized in terms of the output SNR versus the number of data samples K used to form the estimates $\hat{\mathbf{R}}_{xx}$ and $\hat{\mathbf{r}}_{xd}$. An alternate way of characterizing performance that is appropriate for CSLC applications considers the output residue power versus the number of data samples. System performance in terms of the output residue power is considered. The output residue power (or MSE) is given by

$$\xi(\hat{\mathbf{w}}) = E|e(t)|^2 = \sigma_0^2 - \hat{\mathbf{w}}^\dagger \mathbf{r}_{xd} - \mathbf{r}_{xd}^\dagger \hat{\mathbf{w}} + \hat{\mathbf{w}}^\dagger \mathbf{R}_{xx} \hat{\mathbf{w}}^\dagger \tag{5.25}$$

and the estimated MSE (or sample MSE) may be expressed as

$$\hat{\xi}(\hat{\mathbf{w}}) = \frac{1}{K} \sum_{i=1}^{k} |\, e(i)\,|^2 = \hat{\sigma}_0^2 - \hat{\mathbf{w}}^\dagger \hat{\mathbf{r}}_{xd} - \hat{\mathbf{r}}_{xd}^\dagger \hat{\mathbf{w}} + \hat{\mathbf{w}}^\dagger \hat{\mathbf{R}}_{xx} \hat{\mathbf{w}} \tag{5.26}$$

where σ_0^2 is the signal power in the reference antenna, and

$$\hat{\sigma}_0^2 = \frac{1}{K} \sum_{i=1}^{K} x_0(i) x_0^*(i) \tag{5.27}$$

The transient behavior of the system is characterized by evaluating the statistical properties of $\xi(\hat{\mathbf{w}})$ for a given number of data samples K. Assume that $\mathbf{x}(i)$ and $x_0(i)$ are sample

functions from an $(N+1)$ variate, zero-mean, Gaussian random vector $\mathbf{x}'$ having the density function

$$p(\mathbf{x}') = \pi^{-(N+1)}|\mathbf{R}'_{xx}|^{-1}\exp\{-\mathbf{x}'^{\dagger}\mathbf{R}'^{-1}_{xx}\mathbf{x}'\} \tag{5.28}$$

where $\mathbf{R}'_{xx}$ is the $(N+1)\times(N+1)$ covariance matrix

$$\begin{aligned}\mathbf{R}'_{xx} &= E\left\{\begin{bmatrix}\mathbf{x}(t)\\ \hline x_0(t)\end{bmatrix}\left[\mathbf{x}^{\dagger}(t)|x_0^*(t)\right]\right\}\\ &= \left[\begin{array}{c|c}\mathbf{R}_{xx} & \mathbf{r}_{xx_0}\\ \hline \mathbf{r}^{\dagger}_{xx_0} & \sigma_0^2\end{array}\right]\end{aligned} \tag{5.29}$$

where $\mathbf{r}_{xx_0}$ plays the role of $\mathbf{r}_{xd}$ so that $\hat{\mathbf{R}}'_{xx}$ is given by the estimates $\hat{\mathbf{R}}_{xx}$, $\hat{\mathbf{r}}_{xd}$, and $\hat{\sigma}_0^2$. The sample covariance matrix $\hat{\mathbf{R}}'_{xx}$ has the following important properties:

1. The elements of $\hat{\mathbf{R}}'_{xx}$ are jointly distributed according to the complex Wishart probability density function [6]

$$p(\mathbf{A}) = \frac{|\mathbf{A}|^{K-N-1}\exp[-\mathrm{tr}(\mathbf{R}'^{-1}_{xx}\mathbf{A})]}{\pi^{1/2(N+1)N}\Gamma(K)\Gamma(K-1)\cdots\Gamma(K-N)|\mathbf{R}'_{xx}|^K} \tag{5.30}$$

where

$$\begin{aligned}\mathbf{A} &= K\hat{\mathbf{R}}'_{xx}\\ \Gamma(k) &= (k-1)!\end{aligned}$$

2. $\hat{\mathbf{R}}'_{xx}$ is the ML estimate of $\mathbf{R}'_{xx}$ [6]. Therefore $\hat{\mathbf{R}}_{xx}$, $\hat{\mathbf{r}}_{xd}$, and $\hat{\sigma}_0^2$ are the ML estimates of $\mathbf{R}_{xx}$, $\mathbf{r}_{xd}$, and σ_0^2, respectively.

By using a series of transformations on the partitioned matrix $\hat{\mathbf{R}}'_{xx}$, the following important results are obtained [7]:

1. The mean and variance of the sample MSE $\hat{\xi}$, realized using $\hat{\mathbf{w}}_4$ of (5.23) is given by

$$E\{\hat{\xi}\} = \left(1-\frac{N}{K}\right)\xi_{\min} \tag{5.31}$$

$$\mathrm{var}\{\hat{\xi}\} = \frac{1}{K}\left(1-\frac{N}{K}\right)\xi^2_{\min} \tag{5.32}$$

where $\xi_{\min} = \sigma_0^2 - \mathbf{r}^{\dagger}_{xd}\mathbf{R}^{-1}_{xx}\mathbf{r}_{xd}$.

2. The difference between the output residue power $\xi(\hat{\mathbf{w}}_4)$ and the minimum output residue power $\xi_{\min}$ is a direct measure of the quality of the array performance relative to the optimum. The normalized performance quality parameter (or "misadjustment") $M = r^2$ is defined as

$$r^2 \triangleq \frac{\xi(\hat{\mathbf{w}}_4) - \xi_{\min}}{\xi_{\min}} \tag{5.33}$$

The parameter r is a random variable having the density function

$$p(r) = 2\frac{K!}{(K-N)!(N-1)!}\cdot\frac{r^{2N-1}}{(1+r^2)^{K+1}}, \qquad 0 < r < \infty \tag{5.34}$$

The statistical moments of the misadjustment r^2 are easily obtained by recognizing that the variable y defined by

$$y = \frac{1}{1+r^2} \tag{5.35}$$

is governed by an incomplete beta function distribution with parameters $a = K-N+1$ and $b = N$. Consequently, the mean and variance of r^2 are given by

$$E\{r^2\} = \frac{N}{K-N} \tag{5.36}$$

and

$$\text{var}\{r^2\} = \frac{NK}{(K-N)^2(K-N-1)} \tag{5.37}$$

3. For the case when $x_0(t) = d(t)$, define a normalized SNR as in (5.10), (5.11) for $\hat{\mathbf{w}}_4$ according to

$$\rho_3 = \frac{(s/n)_3}{SN_o} = \frac{\hat{\mathbf{w}}_4^\dagger \mathbf{s}\mathbf{s}^\dagger \hat{\mathbf{w}}_4}{\hat{\mathbf{w}}_4^\dagger \mathbf{R}_{nn} \hat{\mathbf{w}}_4 \mathbf{s}^\dagger \mathbf{R}_{nn}^{-1} \mathbf{s}} \tag{5.38}$$

The probability density function of ρ_3 is difficult to evaluate in closed form, but the mean and variance of ρ_3 can be determined numerically using the relations given as follows:

$$\rho_3 = \frac{1}{(1+SN_o)\left[\frac{C+\sin^2\phi_1}{C+\sin^2\phi_1\cos^2\phi_2}\right] - SN_o} \tag{5.39}$$

where $C = (1/r)\sqrt{(s/n)_3}$ and where the joint density function of r, ϕ_1, and ϕ_2 is given by

$$p(r,\phi_1,\phi_2) = \frac{2}{\pi}\frac{K!}{(K-N)!(N-2)!}\frac{r^{2N-1}}{(1+r^2)^{K+1}} \cdot (\sin\phi_1)^{2N-2}(\sin\phi_2)^{2N-3} \tag{5.40}$$

for $0 < r < \infty$, $0 \le \phi_1 < \pi$, $0 \le \phi_2 < \pi$. Note that r, ϕ_1, and ϕ_2 are statistically independent and that r has the same density as (5.34). With the foregoing expressions the numerical evaluation of $E\{\rho_3\}$ can be obtained from

$$E\{\rho_3\} = \int_R P(r) \int_{\Phi_1} P(\phi_1) \int_{\Phi_2} \rho_3 P(\phi_2) d\phi_2 d\phi_1 dr \tag{5.41}$$

$E\{\rho_3^2\}$ can likewise be obtained from (5.41) with ρ_3^2 replacing ρ_3. The variance of ρ_3 is then given by $\text{var}\{\rho_3\} = E\{\rho_3^2\} - E^2\{\rho_3\}$.

4. The normalized MSE performance measure defined by

$$\hat{\xi}_N \triangleq \frac{2K\hat{\xi}}{\xi_{\min}} \tag{5.42}$$

is statistically independent of both $\hat{\mathbf{w}}$ and $\hat{\mathbf{R}}_{xx}$.

The results just summarized place limits on the transient performance of the DMI algorithm. Let us first consider the results expressed by (5.36) and (5.37). From (5.36) it is seen that the output residue power is within 3 dB of the optimum value after only $2N$ distinct time samples, or within 1 dB after $5N$ samples, thereby indicating rapid convergence independent of the signal environment or the array configuration. We see this rapid convergence property, however, applies directly to the interference suppression of a sidelobe canceller (SLC) system *assuming no desired signal is present* when forming $\hat{\mathbf{R}}_{xx}$.

In communications systems the desired signal is usually present, and the SNR performance measure is the primary quantity of interest rather than the MSE, ξ. Furthermore, in radar systems the SLC is often followed by a signal processor that rejects clutter returns, so only that portion of the output residue power due to radiofrequency (RF) interference (rather than clutter) must be suppressed by the SLC system. Let us now show that the presence of either the desired signal or clutter returns in the main beam of an SLC system acts as a disturbance that tends to slow the rate of convergence of a DMI algorithm.

Consider the radar SLC configuration shown in Figure 5-7. The system consists of an SLC designed to cancel only interference followed by a signal processor to remove clutter. To simplify the discussion, assume the clutter power ξ_c^2 received in the main antenna is much larger than clutter power entering the low-gain auxiliary antennas so that clutter in the auxiliary channels is neglected and $\xi_{\min} = \sigma_c^2 + \xi_{N_0}$ where ξ_{N_0} represents the minimum output interference plus thermal noise power. Furthermore, assume that the clutter returns are represented as a sample function from a stationary, zero-mean Gaussian process. It then follows from (5.25), (5.33), and (5.36) that

$$\frac{E[\xi(\hat{\mathbf{w}}) - \xi_{\min}]}{\xi_{N_0}} = \frac{N}{K - N}\left[1 + \frac{\sigma_c^2}{\xi_{N_0}}\right] \tag{5.43}$$

From (5.43) it is evident that the presence of main beam clutter prolongs convergence of the DMI algorithm by an amount that is approximately proportional to the main antenna clutter power divided by the minimum output interference power. This slower convergence is due to the presence of clutter-jammer cross-terms in $\hat{\mathbf{r}}_{xd}$, which results in noisier estimates of the weights. For rapid convergence, adaptation should be confined to time intervals that are relatively clutter free, or a means for minimizing clutter terms present in the estimate of $\hat{\mathbf{r}}_{xd}$ must be found.

The foregoing result is also applicable to communications systems where the main antenna is pointed toward an active desired signal source. Let σ_s^2 represent the desired signal power received in the main channel, and assume the desired signal entering the auxiliary channels can be neglected. Then, the SLC output signal to noise ratio SNR can

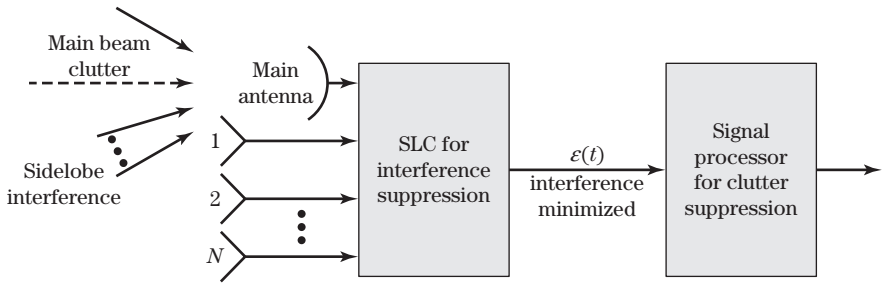


FIGURE 5-7 ■ SLC for interference suppression in a radar system.

easily be shown to be given by

$$\text{SNR} = \frac{\sigma_s^2}{\sigma_s^2 r^2 + \xi_{N_0}(1 + r^2)} \tag{5.44}$$

The maximum value of SNR is denoted by SN_o where $SN_o = \sigma_s^2/\xi_{N_0}$ occurs when $r^2 = 0$ or $\hat{\mathbf{w}} = \mathbf{w}_{\text{opt}}$. Normalizing (5.44) to SN_o yields the expression for the normalized output signal-to-interference plus noise ratio:

$$\rho_4 = \frac{\text{SNR}}{SN_o} = \frac{1}{1 + [1 + (SN_o)]r^2} \tag{5.45}$$

Note that $0 \leq \rho_4 \leq 1$ so the optimum value of ρ_4 is unity. When the output signal to interference plus noise ratio is small so that $SN_o \ll 1$, then the probability distribution of ρ_4 is approximated by the following beta probability density function:

$$P(\rho_4) = \frac{K!}{(N-1)!(K-N)!}(1 - \rho_4)^{N-1}\rho_4^{K-N} \tag{5.46}$$

On comparing (5.46) with the probability density function contained in (5.12), it immediately follows that the two density functions are identical provided that N in (5.12) is replaced by $N + 1$. It then follows from (5.13) for the CSLC system of Figure 5-6 that

$$E\{\rho_4\} = \frac{K - N + 1}{K + 1}; \qquad SN_o \ll 1 \tag{5.47}$$

Hence, for small SN_o only $K = 2N - 1$ independent samples are needed to converge within 3 dB of SN_o. For large SNR ($\gg K/N$), the expected SNR (unnormalized) is approximated by

$$\overline{(\text{SNR})} \cong \frac{K}{N} - 1; \qquad SN_o \gg \frac{K}{N}; \qquad K > N \tag{5.48}$$

The presence of a strong desired signal in the main channel therefore slows convergence to the optimum SNR but does not affect the average output SNR after K samples under the conditions of (5.48).

Finally, consider the results given in (5.39) and (5.40) for the case $x_0(t) = d(t)$ (continuous reference signal present). For large SN_o, the distribution of ρ_3 in (5.39) is approximated by the density function of (5.46) so that

$$E\{\rho_3\} \cong \frac{K - N + 1}{K + 1}; \qquad SN_o \gg 1 \tag{5.49}$$

However, the rate of convergence decreases as SN_o decreases below zero decibels. This effect is illustrated by the plot of $E\{\rho_3\}$ versus K in Figure 5-8.

The behavior described for a reference signal configuration is just the converse of that obtained for the SLC configuration with the desired signal present. The presence of the desired signal in the main channel of the SLC introduced a disturbance that reduced the accuracy of the weight estimate and slowed the convergence. For the reference signal configuration, however, the estimates $\hat{\mathbf{r}}_{xd}$ and $\hat{\mathbf{R}}_{xx}$ are highly correlated under strong desired signal conditions, and the errors in each estimate tend to compensate each other, thereby yielding an improved weight estimate and faster convergence.

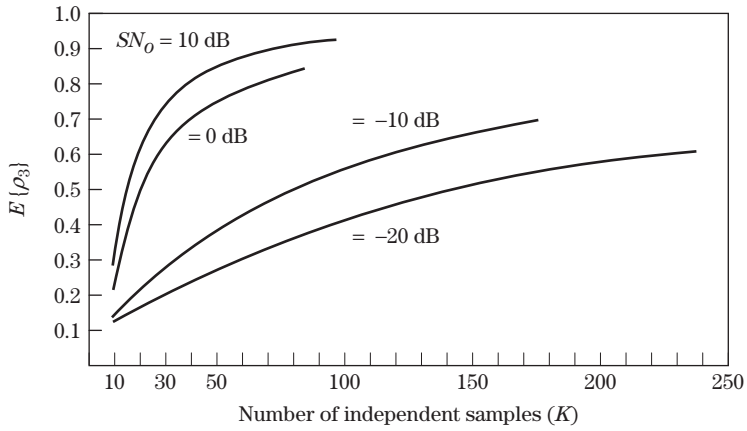


FIGURE 5-8 ■ $E\{\rho_3\}$ of (5.38) versus number of independent samples for $N = 8$ and selected values of SN_O.

5.2 | DIAGONAL LOADING OF THE SAMPLE COVARIANCE MATRIX

Diagonal loading augments the diagonal elements of the signal covariance matrix with either a positive or a negative term and in effect alters the background noise level in which the array is operating [8,9]. Positive loading speeds the weight convergence (although to a suboptimal solution) and avoids very high pattern sidelobes that result from inadequate estimation of the ambient noise; the price paid for these advantages is the reduced ability to suppress weak interference signals. Negative loading creates deeper nulls in the direction of interfering signals (which is useful in television reception from geosynchronous satellites where weak interference from adjacent satellites can produce "ghosts"); negative loading produces a suboptimal SINR value yet also produces a higher signal-to-interference ratio (SIR) value than that of the unloaded result because the resulting weights will produce deeper nulls on weak interfering signals [10].

Consider the example of a four-element linear array with half-wavelength spacing where there are two weak signals incident on the array: a desired signal having a 10 dB SNR incident from the broadside direction ($\theta_d = 0^\circ$) and an interfering signal having a 0 dB interference-to-noise ratio (INR) arriving from $\theta_i = 35^\circ$. Figure 5-9 shows the results of

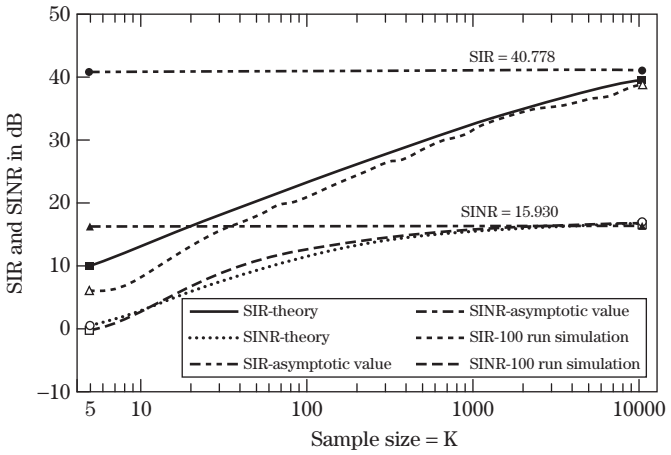


FIGURE 5-9 ■ SINR and SIR vs. K, the number of samples. Input SNR = 10 dB, Input IRN = 0 dB, No Diagonal 3 dB Loading. From Ganz et al., IEEE Trans. Ant. & Prop., March 1990.

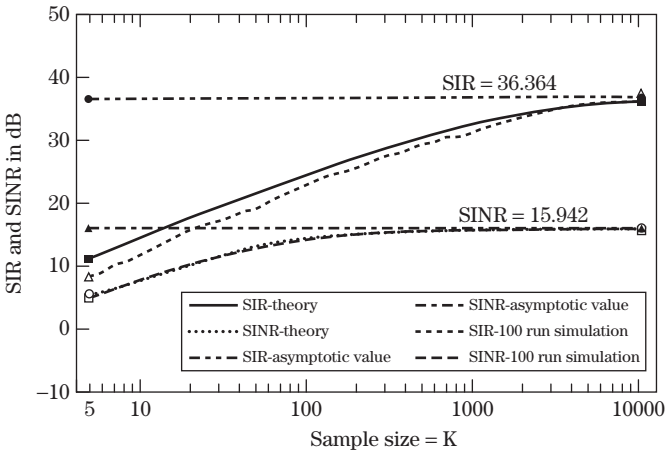


FIGURE 5-10 ■ SINR and SIR vs. K, the number of samples. Input SNR = 10 dB, Input IRN = 0 dB, Positive Diagonal 3 dB Loading. From Ganz et al., IEEE Trans. Ant. & Prop., March 1990.

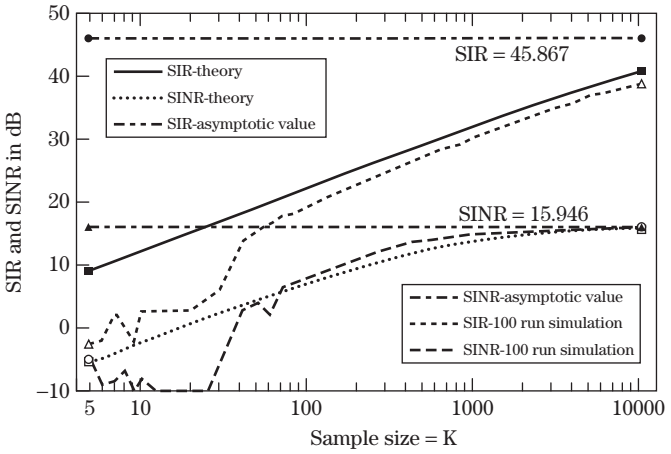


FIGURE 5-11 ■ SINR and SIR vs. K, the number of samples. Input SNR = 10 dB, Input IRN = 0 dB, Negative Diagonal 3 dB Loading. From Ganz et al., IEEE Trans. Ant. & Prop., March 1990.

the SIR and SINR versus number of samples for the case when there is no diagonal loading. Figure 5-10 shows the same results when 3 dB of positive loading is added to the covariance matrix (i.e., σ^2 is added to each diagonal element). The SIR asymptote is approximately 5.8 dB lower than it was without loading, but the SINR asymptote is essentially unchanged. The positive loading has decreased the number of samples required for convergence of the SINR curves to their asymptotic value. Figure 5-11 shows the results of the SIR and SINR versus number of samples for the case when $0.5\sigma^2$ is subtracted (negative loading) from each diagonal element of the covariance matrix. The SINR asymptote remains essentially unchanged, but the SIR asymptote is 5.9 dB higher than it was without loading. Negative loading achieves a better SIR but a slower convergence to the asymptotic value compared with the unloaded case. Notice, however, that with negative loading both the output SIR and SINR curves are quite erratic when a small number of samples are taken; this behavior is due to the variance of the output powers is large for small sample sizes.

5.3 FACTORIZATION METHODS

For the system of equations, $\mathbf{Rw} = \mathbf{v}$, where $\mathbf{R}$ is N × N, and $\mathbf{w}$ and $\mathbf{v}$ are N × 1 vectors. The vector $\mathbf{w}$ is computed by factoring the matrix $\mathbf{R}$ rather than directly computing $\mathbf{R}^{-1}$ since the direct inversion process can be unstable. Three $\mathbf{R}$ factorization approaches for obtaining $\mathbf{w}$ are presented: triangularization, Cholesky factorization, and spectral (or **U-D**, **L-D**) factorization.

5.3.1 Triangularization of R

Suppose there is a transformation matrix, $\mathbf{Q}$, such that $\mathbf{QR} = \mathbf{U}$ where $\mathbf{U}$ is an upper triangular matrix. In this case it is possible to solve the system $\mathbf{Uw} = \mathbf{Qv} = \mathbf{y}$ for $\mathbf{w}$ directly without finding a matrix inverse by means of a simple back-substitution algorithm [11]. The following back-substitution algorithm computes the elements of $\mathbf{w}$ directly.

For j = N, N − 1, . . . , 1, compute

$$\mathrm{w(j)} = \left(\mathrm{y}(j) - \sum_{k=j+1}^{N} \mathrm{u}(j,k)\mathrm{w}(k) \right) \Big/ \mathrm{u}(j,j) \tag{5.50}$$

The algorithm begins with w(N) = y(N)/U(N,N) and progresses backward until computing element w(1). Suppose instead that $\mathbf{QR} = \mathbf{L}$ where $\mathbf{L}$ is a lower triangular matrix; then a simple forward-substitution algorithm (which is an analog of the forward-substitution algorithm) yields the desired solution for the vector $\mathbf{w}$.

Sometimes there is a need to explicitly have the inverse $\mathbf{Z} = \mathbf{U}^{-1}$. A simple algorithm for obtaining the elements of $\mathbf{Z}$ (which is also upper triangular) is given by

$$\mathrm{z}(1,1) = 1/\mathrm{u}(1,1) \tag{5.51}$$

Then for j = 2, . . . , N, evaluate the following two equations:

$$\mathrm{z}(j,j) = 1/\mathrm{u}(j,j) \tag{5.52}$$

$$\mathrm{z}(k,j) = -\left(\sum_{m=k}^{j-1} \mathrm{z}(k,m)\mathrm{u}(m,j) \right) \mathrm{u}(j,j), \qquad k = 1, \ldots, j-1 \tag{5.53}$$

We now need to determine a transformation matrix, $\mathbf{Q}$, which succeeds in triangularizing the original matrix, **R.** There are two methods for obtaining such a transformation matrix: (1) Givens rotations; and (2) Householder transformations.

5.3.1.1 Givens Rotations [12]

The original $\mathbf{R}$ matrix is N × N, and it is desired to find a transformation $\mathbf{Q}$ such that the product $\mathbf{QR}$ is an upper triangular matrix, $\mathbf{U}$. One way of accomplishing this result is to annihilate the elements of $\mathbf{R}$ one element at a time; first progress up the first column of $\mathbf{R}$ starting with $\mathrm{r_{N1}}$ and ending with $\mathrm{r_{21}}$ and continue across N − 1 columns. Therefore, there are a total of $(\mathrm{N}-1) + (\mathrm{N}-2) + \cdots + 2 + 1 = \sum_{k=1}^{N-1} k = \begin{cases} \gamma\mathrm{N}, & \mathrm{N\ odd} \\ \gamma(\mathrm{N}-1), & \mathrm{N\ even} \end{cases}$,

where $\gamma = \mathrm{Int}\left[\frac{N}{2}\right]$ annihilations and Int[ ] denotes the integer remaining after rounding the number in the bracket to the next lower integer.

Givens rotations annihilate one element at a time using the planar rotation transformation

$$\mathbf{G} = \begin{bmatrix} c^* & s \\ -s^* & c \end{bmatrix} \tag{5.54}$$

With the complex vector $\mathbf{v} = \begin{bmatrix} \mathrm{v}_1 \\ \mathrm{v}_2 \end{bmatrix}$, we find that

$$\mathbf{Gv} = \begin{bmatrix} c^*\mathrm{v}_1 + s\mathrm{v}_2 \\ -s^*\mathrm{v}_1 + c\mathrm{v}_2 \end{bmatrix} \tag{5.55}$$

and require that $s^*\mathrm{v}_1 = \mathrm{cv}_2$ with the unitary condition $|c|^2 + |s|^2 = 1$. It follows that

$$\mathrm{c} = \frac{\mathrm{v}_1}{\sqrt{|\mathrm{v}_1|^2 + |\mathrm{v}_2|^2}} \tag{5.56}$$

$$\mathrm{s} = \frac{\mathrm{v}_2^*}{\sqrt{|\mathrm{v}_1|^2 + |\mathrm{v}_2|^2}} \tag{5.57}$$

$$\mathbf{Gv} = \mathbf{v}' \text{ with } \mathrm{v}_1{}' = \begin{bmatrix} \mathrm{v}_1{}' \\ 0 \end{bmatrix} \tag{5.58}$$

$$\text{where } v_1{}' = \sqrt{|v_1|^2 + |v_2|^2} \tag{5.59}$$

Consider the complex column vector $\mathbf{v} = [\mathrm{v}_1 \ldots \mathrm{v}_\mathrm{m}, \ldots \mathrm{v}_\mathrm{n}, \ldots, \mathrm{v}_\mathrm{K}]^\mathrm{T}$. For a K × K matrix, **A**, to eliminate the a(m,n) element requires altering the elements in the *m*th row and the *n*th column using the following Givens transformation:

$$\mathbf{G}(m,n) = \begin{bmatrix} 1 & 0 & & & & 0 \\ 0 & 1 & & & & 0 \\ 0 & & c_{mm}^* & & s_{mn} & 0 \\ 0 & & & 1 & & 0 \\ & & -s_{nm}^* & & c_{nn} & \\ 0 & & & & & 1 \end{bmatrix}, \tag{5.60}$$

$$\text{where} \quad c_{mm}^* = c_{nn} = \frac{\mathrm{v}_m}{\sqrt{|\mathrm{v}_m|^2 + |\mathrm{v}_n|^2}} \tag{5.61}$$

$$\text{and} \quad s_{nm} = s_{mn}^* = \frac{\mathrm{v}_n^*}{\sqrt{|\mathrm{v}_m|^2 + |\mathrm{v}_n|^2}}. \tag{5.62}$$

$$\text{Then} \quad \mathbf{G}(m,n)\mathbf{v} = [\mathrm{v}_1 \ldots \sqrt{|\mathrm{v}_m|^2 + |\mathrm{v}_n|^2} \ldots 0, \ldots \mathrm{v}_\mathrm{K}]^\mathrm{T} \tag{5.63}$$

As an example, let $\mathbf{A} = \begin{bmatrix} \mathrm{a}_{11} & \mathrm{a}_{12} & \mathrm{a}_{13} \\ \mathrm{a}_{21} & \mathrm{a}_{22} & \mathrm{a}_{23} \\ \mathrm{a}_{31} & \mathrm{a}_{32} & \mathrm{a}_{33} \end{bmatrix}$. To annihilate element a_{31} requires affecting elements in the first and third rows (rotate a_{31} into a_{11}) so that

$$\mathbf{G}_1 = \begin{bmatrix} c_1^* & 0 & s_1 \\ 0 & 1 & 0 \\ -s_1^* & 0 & c_1 \end{bmatrix} \quad \text{and} \quad \mathbf{G}_1\mathbf{A} = \begin{bmatrix} r_1 & \mathrm{a}_{12}' & \mathrm{a}_{13}' \\ \mathrm{a}_{21} & \mathrm{a}_{22} & \mathrm{a}_{23} \\ 0 & \mathrm{a}_{32}' & \mathrm{a}_{33}' \end{bmatrix} \tag{5.64}$$

where the 3,1 product element becomes zero since $c_1 a_{31} = s_1^* a_{11}$, $c_1 = \dfrac{a_{11}}{r_1}$, $s_1 = \dfrac{a_{31}^*}{r_1}$ and $r_1 = \sqrt{|a_{11}|^2 + |a_{31}|^2}$. The next step is to annihilate element a_{21}, which is done by rotating element a_{21} into element r_1 using the Givens rotation

$$\mathbf{G}_2 = \begin{bmatrix} c_2^* & s_2 & 0 \\ -s_s^* & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{so that} \quad \mathbf{G}_2\mathbf{A}' = \begin{bmatrix} c_2^* a_{11}' + s_2 a_{21} & c_2^* a_{12}' + s_2 a_{23} & c_2^* a_{13}' + s_2 a_{23} \\ -s_2^* a_{11}' + c_2 a_{21} & -s_2^* a_{12}' + c_2 a_{22} & -s_2^* a_{13}' + c_2 a_{23} \\ 0 & a_{32}' & a_{33}' \end{bmatrix} \tag{5.65}$$

which requires that

$$c_2 a_{21} = s_2^* a_{11}', \text{ where } c_2 = \frac{r_1}{r_{11}}, \quad s_2 = \frac{a_{21}^*}{r_{11}} \text{ and } r_{11} = \sqrt{|r_1|^2 + |a_{12}'|^2}. \tag{5.66}$$

The last step is then to annihilate element a_{32}' with Givens rotation $\mathbf{G}_3$ by rotating element a_{32}' into element a_{22}'. The complete triangularization $\mathbf{QR} = \mathbf{U}$ is then accomplished with the matrix $\mathbf{Q}$, where $\mathbf{Q}$ is the product of all the successive Givens rotation matrices, i.e., $\mathbf{Q} = \mathbf{G}_p \mathbf{G}_{p-1}, \ldots, \mathbf{G}_2 \mathbf{G}_1$ where $\mathbf{G}_1$ corresponds to the matrix required to annihilate element $r_{N,1}$, and $\mathbf{G}_p$ corresponds to the matrix required to annihilate element $r_{N,N-1}$. Obviously, annihilating the upper right off-diagonal entries in $\mathbf{R}$ will result in the lower triangular analog, $\mathbf{L}$, of the upper triangular result.

5.3.1.2 Householder Transformations [12]

Householder transformations annihilate a complete column of entries to upper triangularize $\mathbf{R}$. The basic idea is shown as follows:

$$\mathbf{T}_1 \mathbf{R}_1 = \begin{bmatrix} s_1 & \mathbf{r}_1^T \\ 0 & \mathbf{R}_2 \end{bmatrix} \tag{5.67}$$

where $\mathbf{R}_1$ is N × N, s_1 is a scalar, $\mathbf{R}_2$ is (N − 1) × (N − 1), "0" is a (N − 1) × 1 column vector, and $\mathbf{r}_1^T$ is a 1 × (N − 1) row vector. The matrix defined by $\begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{R}_2 \end{bmatrix}$ is denoted by $\tilde{\mathbf{R}}_1$. The transformation $\mathbf{T}_1$ is not explicitly computed, although it can be obtained if desired. Having annihilated all the elements in the first column (except for "s_1"), the next step applies a second transformation to annihilate the elements in the second column as follows:

$$\mathbf{T}_2 \mathbf{R}_2 = \begin{bmatrix} s_2 & \mathbf{r}_2^T \\ 0 & \mathbf{R}_3 \end{bmatrix} \tag{5.68}$$

This process continues until the triangularization of the original matrix is complete. The overall Householder transformation can then be written as

$$\mathbf{T}_h = \begin{bmatrix} \mathbf{I}_{N-2} & 0 \\ 0 & \mathbf{T}_{N-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N-3} & 0 \\ 0 & \mathbf{T}_{N-2} \end{bmatrix} \cdots \begin{bmatrix} \mathbf{I}_1 & 0 \\ 0 & \mathbf{T}_2 \end{bmatrix} \mathbf{T}_1 \tag{5.69}$$

where $\mathbf{I}_k$ denotes the k × k identity matrix. Clearly, the $\mathbf{T}_{N-1}$ transformation is just a 2 × 2 Givens rotation that annihilates the N, N − 1 element of the $\mathbf{R}$ matrix. The various $\mathbf{T}_k$ transformations are then orthogonal (both symmetric and orthogonal so that $\mathbf{T}_k^{-1} = \mathbf{T}_k^T$). The operation for computing the triangularization of a matrix, $\mathbf{R}$, with elements r(i,j)

involves a process described by the matrix operation

$$\mathrm{T_u}\mathbf{R} = \begin{bmatrix} s & \\ 0 & \tilde{\mathbf{R}} \end{bmatrix} \tag{5.70}$$

The algorithms for computing s and $\tilde{\mathbf{R}}$ are then given by [12]

$$s = -\mathrm{sgn}(\mathrm{r}(1,1)) \left(\sum_{i=1}^{N} [\mathrm{r}(i,1)]^2 \right) \tag{5.71}$$

$$\mathrm{u}(1) = \mathrm{r}(1,1) - \mathrm{s} \tag{5.72}$$

$$\mathrm{u(i)} = \mathrm{r(i,1)}, \text{ for i} = 2 \ldots \mathrm{N} \tag{5.73}$$

$$\beta = 1/(\mathrm{su}(1)) \tag{5.74}$$

For $\mathrm{j} = 2, \ldots, \mathrm{N}$, evaluate (5.75) and (5.76) (apply $\mathbf{T}$ to the successive columns of $\mathbf{R}$)

$$\gamma = \beta \cdot \sum_{i=1}^{N} \mathrm{u}(i)\mathrm{R}(i,j) \tag{5.75}$$

$$\tilde{\mathrm{R}}(\mathrm{i}, \mathrm{j}-1) = \mathrm{R(i,j)} + \gamma \mathrm{u(i)}, \mathrm{i} = 1, \ldots, \mathrm{N} \tag{5.76}$$

where $\tilde{\mathrm{R}}$(i,j) may be replaced by R(i,j). This completes the reduction of an N × N matrix to a triangular form.

5.3.2 Cholesky Factorization

To compute the lower triangular Cholesky factorization $\mathbf{R} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, the original problem $\mathbf{R}\mathbf{w} = \mathbf{v}$ is written as $\mathbf{L}\mathbf{L}^{\mathrm{T}}\mathbf{w} = \mathbf{v}$. Letting $\mathbf{L}^{\mathrm{T}}\mathbf{w} = \mathbf{u}$ yields a pair of rectangular matrix equations to solve:

$\mathbf{L}\mathbf{u} = \mathbf{v}$ is solved for $\mathbf{u}$ by a forward substitution algorithm

$\mathbf{L}^{\mathrm{T}}\mathbf{w} = \mathbf{u}$ is solved for $\mathbf{w}$ by a backward substitution algorithm

The Cholesky factorization of $\mathbf{R}$ with positive diagonal elements is given by the following algorithm [11,13]:

$$\text{For } \mathrm{j} = 1, 2, \ldots, \mathrm{N}-1$$

$$\mathrm{L}(j,j) = \mathrm{R}(j,j)^{1/2} \tag{5.77}$$

$$\text{For } \mathrm{k} = \mathrm{j}+1, \ldots, \mathrm{N}$$

$$\mathrm{L(k,j)} = \mathrm{R(k,j)/L(j,j)} \tag{5.78}$$

$$\text{For } \mathrm{i} = \mathrm{k}, \ldots, \mathrm{N}$$

$$\mathrm{R(i,k)} = \mathrm{R(i,j)} - \mathrm{L(i,j)L(k,j)} \tag{5.79}$$

$$\text{end for}$$

$$\mathrm{L(N,N)} = \mathrm{P}(N,N)^{1/2} \tag{5.80}$$

An analogous algorithm starting with U(N, N) obviously applies for the upper triangular Cholesky factorization.

5.3.3 Spectral (U-D or L-D) Factorization [11,14]

Consider the spectral factorization $\mathbf{R} = \mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{\mathrm{T}}$ (or $\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$, $\mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$), where $\boldsymbol{\Lambda}$ is diagonal (consisting of the eigenvalues of the matrix $\mathbf{R}$ arranged in descending order), the columns of $\mathbf{M}$ are the normalized eigenvectors corresponding to each of the eigenvalues, and $\mathbf{U}$ is upper triangular with ones along the principal diagonal where the elements of $\mathbf{D}$ and $\mathbf{U}$ are given by the following algorithm:

The diagonal elements of $\mathbf{D}$ are denoted by d(j)

$$
\begin{aligned}
&\text{for } \mathrm{j} = \mathrm{N}, \mathrm{N}-1, \ldots, 2 \\
&\quad \mathrm{d(j)} = \mathrm{R(j,j)} && (5.81)\\
&\quad \mathrm{U(j,j)} = 1 \\
&\quad \text{for } \mathrm{k} = 1, 2, \ldots, \mathrm{j}-1 \\
&\qquad \mathrm{U(k,j)} = \mathrm{R(k,j)/d(j)} && (5.82)\\
&\qquad \text{for } \mathrm{i} = 1, 2, \ldots, \mathrm{k} \\
&\qquad\quad \mathrm{R(i,k)} = \mathrm{R(i,k)} - \mathrm{U(i,j)U(k,j)d(j)} && (5.83)\\
&\quad \text{end for} \\
&\quad \mathrm{U(1,1)} = 1 \quad \text{and} \quad \mathrm{d(1)} = \mathrm{R(1,1)} && (5.84)
\end{aligned}
$$

The **U-D** factorization given by (5.81)–(5.84) as well as the **L-D** factorization is different from the spectral factorization involving eigenvalues and normalized eigenvectors. Spectral factorization renders finding the desired inverses particularly simple, thereby making the additional computation required by the accompanying eigenvalue and eigenvector computations worthwhile. Note that the **U-D** and **L-D** factorizations are also different from each other.

The matrix $\mathbf{R}^{-1}$ is formed by inspection from the spectral factorization as follows. Since $\mathbf{R}^{-1} = [\mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{\mathrm{T}}]^{-1} = (\mathbf{M}^{\mathrm{T}})^{-1}\boldsymbol{\Lambda}^{-1}\mathbf{M}^{-1}$ where the elements of $\boldsymbol{\Lambda}^{-1}$ are merely $1/\lambda(\mathrm{i})$ for $\mathrm{i} = 1, \ldots, N$, $\mathbf{M}^{\mathrm{H}} = \mathbf{M}^{-1}$ and $(\mathbf{M}^{\mathrm{T}})^{-1} = \mathbf{M}$ since $\mathbf{M}$ is unitary. The extra computation involved in finding the eigenvalues and associated eigenvectors is rewarded by the ease in finding the desired inverses.

The inverses required by the **U-D** factorization are $\mathbf{U}^{-1}$, $\mathbf{D}^{-1}$, where the elements of $\mathbf{D}^{-1}$ are merely the inverse of the elements of $\mathbf{D}$, and the inverse of $\mathbf{U}$ is given by equations (5.51)–(5.53).

5.4 TRANSIENT RESPONSE COMPARISONS

The transient response characteristics of the DMI algorithms corresponding to $\hat{\mathbf{w}}_1$ of (5.5) and $\hat{\mathbf{w}}_4$ of (5.23) may be obtained by examining

$$\overline{\rho} = \frac{E\{\text{output SNR}\}}{SN_o}$$

as a function of K (the number of independent samples used in obtaining $\hat{\mathbf{R}}_{xx}$ and $\hat{\mathbf{r}}_{xd}$) for selected signal conditions. Assuming an input interference-to-signal ratio of 14 dB and an input signal-to-thermal noise ratio of 0 dB, $\overline{\rho}$ was obtained for $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ by averaging 50 independent responses for a four-element linear array for which $d/\lambda = \frac{1}{2}$. A single

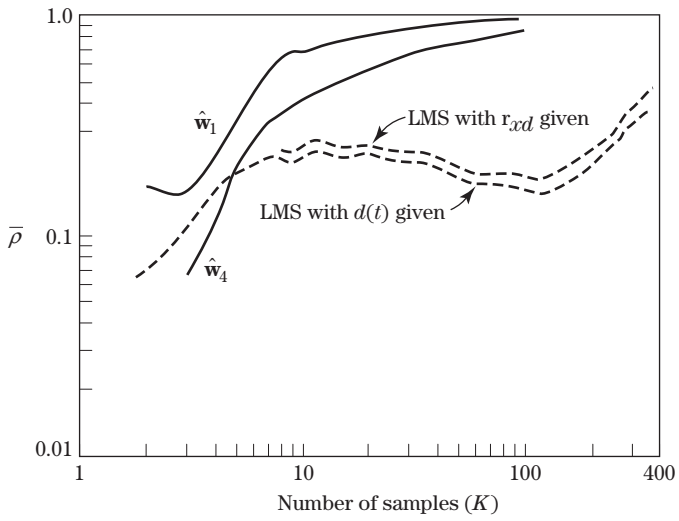


FIGURE 5-12 ■ Transient response of $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ for $SN_o = 3.8$ and one interference signal located 75° away from the desired signal.

directional interference source was assumed that was separated from the desired signal location by 75°. The resulting transient response is given in Figure 5-12.

The transient response obtained using $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ can be compared with the transient response that would result using the LMS algorithm having an iteration period equal to the intersample collection period of the DMI algorithm by considering the behavior of $\overline{\rho}(K)$. For the DMI approach, K represents the number of independent signal samples used to compute the sample covariance matrix, whereas for the LMS algorithm K represents the number of iterations completed. This comparison of the DMI and LMS algorithms is not satisfactory for the following reasons:

1. The transient response of the LMS algorithm depends on the selection of the step size, which can be made arbitrarily small (thereby resulting in an arbitrarily long transient response time constant).
2. The transient response of the LMS algorithm depends on the starting point at which the initial weight vector guess is set. A good initial guess may result in excellent transient response.
3. Whereas the variance of $\rho(K)$ decreases as K increases with the DMI approach, the variance of $\rho(K)$ remains constant (in the steady state) as K increases for the LMS algorithm since the steady-state variance is determined by the step size selection.

Nevertheless, a comparison of $\overline{\rho}(K)$ behavior does yield an indication of the output SNR response speed and is therefore of some value.

The LMS algorithm convergence condition (4.49) is violated if the step size Δ_s exceeds $1/P_{\text{IN}}$, where P_{IN} = total array input power from all sources. Consequently, by selecting $\Delta_s = 0.4/P_{\text{IN}}$, the step size is only 4 dB below the convergence condition limit, the LMS control loop is quite noisy, and the resulting speed of the transient response is close to the theoretical maximum. Since the maximum time constant for the LMS algorithm transient response is associated with the minimum covariance matrix eigenvalue

$$\tau_{\max} \cong \frac{1}{k_s \Delta t \lambda_{\min}} \tag{5.85}$$

and $\lambda_{\min} \geq \sigma^2$, the thermal noise power variance,[1] it follows that

$$\tau_{\max} \cong \frac{P_{\text{IN}}}{0.4\sigma^2} \tag{5.86}$$

With a four-element array, $P_{\text{IN}} = 4(P_D + \sigma^2 + P_I)$,
where

P_D = desired signal power
σ^2 = thermal noise power
P_I = interference signal power

Therefore, if $P_I/P_D = 14$ dB and $P_D/\sigma^2 = 0$ dB, then $P_{\text{IN}}/\sigma^2 = 108$, and $\tau_{\max} = 270$ samples.

The transient response time constant of the desired signal power is proportional to the eigenvalue λ_1 associated with the desired signal power. Since in the current example $\lambda_1 \cong 4P_D$, the time constant associated with the desired signal power is approximately

$$\tau_1 \cong \frac{1}{\Delta_s \lambda_1} = \frac{P_{\text{IN}}}{(0.4)4P_D} = 67 \text{ samples} \tag{5.87}$$

A convenient starting point for the weight vector with the LMS algorithm is $\mathbf{w}_0^T = [1, 0, 0, 0]$, since this corresponds to an omnidirectional array pattern. For the foregoing conditions, the behavior of $\overline{\rho}(K)$ resulting from the use of $\hat{\mathbf{w}}_3$, $\hat{\mathbf{w}}_4$ and the two versions of the LMS algorithm given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + k_s \Delta t\, [\mathbf{r}_{xd} - \mathbf{x}(k)\mathbf{x}^{\dagger}(k)\mathbf{w}(k)], \qquad \mathbf{r}_{xd} \text{ given} \tag{5.88}$$

and

$$\mathbf{w}(k+1) = \mathbf{w}(k) + k_s \Delta t\, [\mathbf{x}(k)\, d^*(k) - \mathbf{x}(k)\mathbf{x}^{\dagger}(k)\mathbf{w}(k)], \qquad d(k) \text{ given} \tag{5.89}$$

was determined by simulation. The results are illustrated in Figure 5-12 where for the specified conditions $SN_o = 3.8$.

The results of Figure 5-12 indicate that the response resulting from the use of the DMI derived weights $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ is superior to that obtained from the LMS derived weights. Whereas the initial response of the LMS derived weights indicated improved output SNR with increasing K, this trend reverses when the LMS algorithm begins to respond along the desired signal eigenvector, since any decrease in the desired signal response without a corresponding decrease in the thermal noise response causes the output signal to thermal noise ratio to decrease. Once the array begins to respond along the thermal noise eigenvectors, then $\overline{\rho}$ again begins to increase.

The undesirable transient behavior of the two LMS algorithms in Figure 5-12 can be avoided by selecting an initial starting weight vector different from that chosen for the foregoing comparison. For example, by selecting $\mathbf{w}(0) = \mathbf{r}_{xd}$ the initial LMS algorithm response is greatly improved since this initial starting condition biases the array pattern toward the desired signal direction thereby providing an initially high output SNR. Furthermore, by selecting $\mathbf{w}(0) = \alpha \mathbf{r}_{xd}$ where α is a scalar constant the initial starting condition can also result in a monotonically increasing $\overline{\rho}(K)$ as K increases for the two

[1] It is assumed that each weighted channel contains thermal noise (with noise power σ^2) that is uncorrelated between channels.

LMS algorithms since by appropriately weighting $\mathbf{r}_{xd}$, the magnitude of the initial array response to the desired signal can be made small enough so that as adaptation proceeds, the resulting $\overline{\rho}(K)$ always increases.

Improvement of the transient response through judicious selection of the initial starting condition can also be introduced into the DMI derived weight vectors as well. For example, by selecting

$$\hat{\mathbf{w}}_1 = \left[\frac{1}{K}\left(\sum_{j=1}^{K}\mathbf{x}(j)\mathbf{x}^{\dagger}(j) + \alpha\mathbf{I}\right)\right]^{-1}\mathbf{r}_{xd} \tag{5.90}$$

then even before forming an estimate of $\mathbf{R}_{xx}$ the weight vector is biased toward the desired signal direction, and the transient responses corresponding to $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ in Figure 5-12 is greatly improved. Nevertheless, even the improved transient response for an LMS algorithm obtained by appropriately biasing the initial weight vector is slower than the convergence speed of DMI.

It is also interesting to obtain the transient response of $\overline{\rho}(K)$ with two interference signals present. Assume one interference-to-signal ratio of 30 dB and a second interference-to-signal ratio of 10 dB, where the stronger interference signal is located 30° away from the desired signal, the weaker interference signal is located 60° away from the desired signal, with all other conditions the same as for Figure 5-12. The resulting transient response for the two DMI derived weight vectors and for the two LMS algorithms [with $d(t)$ given and with $\mathbf{r}_{xd}$ given] with initial starting weight vector = [1, 0, 0, 0] is illustrated in Figure 5-13. The presence of two directional interference sources with widely different power levels results in a wide eigenvalue spread and a consequent slow convergence rate for the LMS algorithms. Since P_{IN}/λ_1 is now 40 times larger than for the conditions of Figure 5-12, the time constant τ_1 is now 40 times greater than before. The DMI derived weight transient response, however, is virtually unaffected by this eigenvalue spread.

The principal convergence results for DMI algorithms under various array configurations and signal conditions can be conveniently summarized as shown in Table 5-1. The derivation of these results may be found in [2,3].

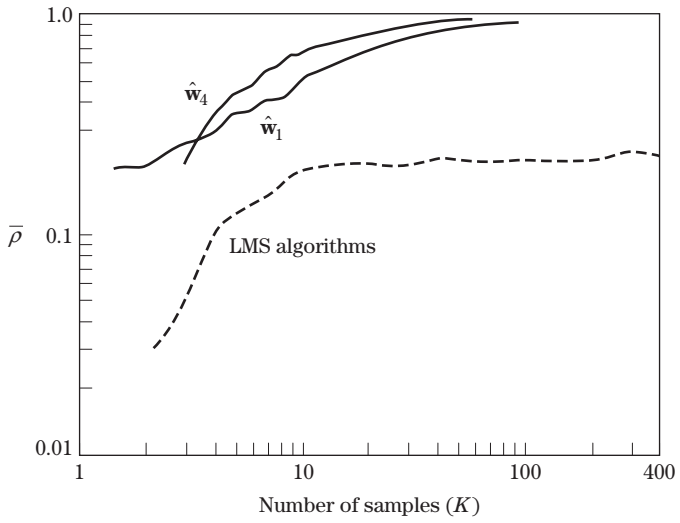


FIGURE 5-13 ■ Transient response of $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_4$ for $SN_o = 0.4158$ and two interference signals located 30° and 60° away from the desired signal.

TABLE 5-1 ■ Comparison of DMI Algorithm Convergence Rates for Selected Array Configurations and Signal Conditions

Array Configuration	$\hat{\mathbf{w}}_{\text{opt}}$	Signal Conditions	Performance Measure	Approximate Number of Independent Samples Required for Convergence (to within 3 dB of Optimum)
Sidelobe canceller	$\hat{\mathbf{R}}_{xx}^{-1}\hat{\mathbf{r}}_{xx_0}$	Desired signal absent	Minimum MSE	$2N$, N = number of array elements
		Desired signal present in main beam only	Maximum SNR, SN_o	$2N\left[1+\dfrac{SN_o}{2}\right]$, $SN_o \gg 1$ $2N$, $SN_o \ll 1$
		Clutter returns in main beam only	Minimum output (interference + noise) power, ξ_{N_0}	$2N\left[1+\dfrac{\sigma_c^2}{2\xi_{N_0}}\right]$, σ_c^2 = main channel clutter power
Fully adaptive array	$\hat{\mathbf{R}}_{xx}^{-1}\hat{\mathbf{r}}_{xd}$	Known desired signal present $d(t) = s(t)$	$2N$, Maximum SNR, SN_o	$SN_o \gg 1$ $2N\left[1+\dfrac{1}{2SN_o}\right]$, $SN_o \ll 1$
	$\hat{\mathbf{R}}_{xx}^{-1}\mathbf{r}_{xd}$	Desired signal direction of arrival known but desired signal absent	Maximum SNR, SN_o	$2N-3$
		Desired signal direction of arrival known and desired signal present	Maximum SNR, SN_o	$2N$, $SN_o \ll 1$ $2N\left[1+\dfrac{SN_o}{2}\right]$, $SN_o \gg 1$

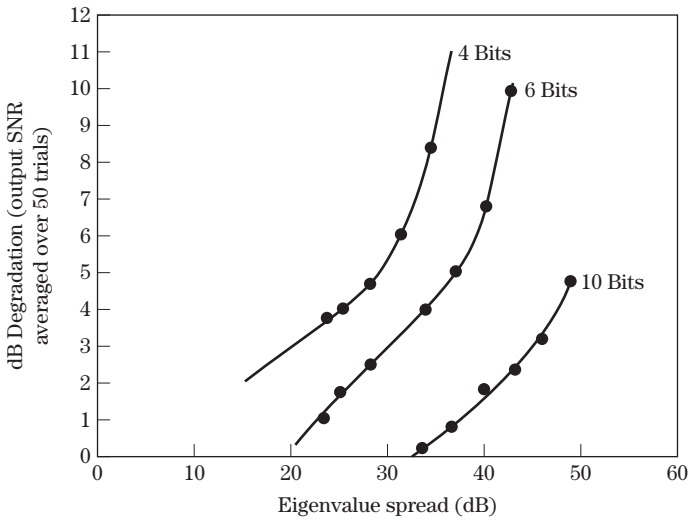


FIGURE 5-14 ■ Output SNR degradation versus eigenvalue spread for a specified number of available bits.

5.5 SENSITIVITY TO EIGENVALUE SPREAD

Despite the fact that the convergence speed of a DMI algorithm is insensitive to eigenvalue spread in $\mathbf{R}_{xx}$, the accuracy of the steady-state solution will exhibit sensitivity to eigenvalue spread once that spread exceeds a certain critical amount. This steady-state accuracy sensitivity arises in the following manner.

If a sufficient number of independent time samples have been collected to ensure that the sample covariance matrix estimate is arbitrarily close to $\mathbf{R}_{xx}$ (or $\mathbf{R}_{nn}$), then the exact solution for the adaptive weight vector results unless the sample matrix inversion has insufficient accuracy due to matrix ill conditioning (as measured by the eigenvalue spread). The eigenvalue spread can always reach a point where the computer cannot accurately invert the matrix and output SNR degradation results.

Consequently, there is a trade-off between the accuracy of the inversion algorithm and the allowable eigenvalue spread in the input signal covariance matrix for any DMI algorithm. This trade-off was examined by simulating the desired matrix inversion for a prescribed number of available bits, and Figure 5-14 illustrates the nature of this trade-off for a four-element linear array in a two-jammer interference signal environment where the eigenvalue spread is defined by $\lambda_{\max}/\lambda_{\min}$, the ratio of maximum to minimum eigenvalues in the input signal covariance matrix. It is seen that so long as the sample covariance matrix has an eigenvalue spread less than a critical value (that depends on the number of bits available in the computer to accomplish the required matrix inversion), then a DMI algorithm is insensitive to eigenvalue spread. Once this critical value of eigenvalue spread is exceeded, however, very rapid degradation in the output SNR results with any additional increase in eigenvalue spread.

5.6 SUMMARY AND CONCLUSIONS

DMI algorithms yield convergence speeds that surpass the LMS algorithm. The resulting weight vector depends only on the signal covariance matrix (through the Wiener solution) and knowledge of the signal covariance matrix comes from forming the sample covariance

matrix. When the desired signal is absent from the signal environment, then use of the beam steering vector $\mathbf{r}_{xd}$ yields the most desirable transient response characteristics. When the desired signal is strong, however, formation of the estimate $\hat{r}_{xd}$ yields transient response characteristics that are superior to the use of apriori information represented by $\mathbf{r}_{xd}$ alone.

DMI algorithms are insensitive to eigenvalue spread until a certain critical level is exceeded that depends on the number of bits available in the computer to perform the matrix inversion computation. The practicality of the DMI approach is restricted by the number of degrees of freedom of the adaptive processor. When the feasibility of the DMI approach is not precluded, however, the additional complexity introduced by directly obtaining the sample covariance matrix is rewarded by the rapid convergence and insensitivity to the previously noted eigenvalue spread.

The difficult problem of the direct computation of a matrix inverse can be circumvented by recourse to factorization methods that boast superior accuracy and numerical stability properties. The three methods that have been presented here (triangularization of the covariance matrix and solution of the triangular system by forward and backward substitution, Cholesky factorization, and spectral factorization [**U**-**D** or **L**-**D**]) offer attractive alternatives to direct matrix inversion computations. Furthermore, algorithms based on these methods are extremely fast.

5.7 PROBLEMS

1. ***Development of a Recursion Formula.*** The DMI algorithm requires a matrix inversion each time a new weight is to be calculated (as, e.g., when the signal environment changes). By applying the matrix inversion lemma

$$[A + \mathbf{u}^\dagger \mathbf{R}\mathbf{u}]^{-1} = A^{-1} - A^{-1}\mathbf{u}^\dagger[\mathbf{R}^{-1} + \mathbf{u}A^{-1}\mathbf{u}^\dagger]^{-1}\mathbf{u}A^{-1}$$

show that the weights can be updated at each sample using the recursive formulas

$$W(k) = W(k-1) + \frac{\mathbf{P}(k-1)\mathbf{x}(k)\,\varepsilon^*(k)}{1 + \mathbf{x}^\dagger(k)\mathbf{P}(k-1)\mathbf{x}(k)}$$

where

$$\varepsilon^*(k) = -\mathbf{x}^\dagger(k)\mathbf{P}(k-1)\,\mathbf{b}^*$$

and

$$\mathbf{P}(k) = \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{x}(k)\mathbf{x}^\dagger(k)\mathbf{P}(k-1)}{1 + \mathbf{x}^\dagger(k)\mathbf{P}(k-1)\mathbf{x}(k)}$$

Hint: Apply the lemma to the inversion of

$$[\hat{\mathbf{R}}_{xx}(k-1) + \mathbf{x}(k)\mathbf{x}^\dagger(k)]$$

2. ***Similarity between Array Element Outputs and Tapped-Delay Line Outputs.*** Consider an adaptive filter consisting of a tapped delay line with a complex weight at each tap.

(a) Show that the DMI algorithm for minimizing the MSE between the filter output and the reference signal is of the same form as for an adaptive array if the tap outputs are taken as analogous to the array inputs.

(b) What restrictions on sampling are imposed if we desire independent samples of the inputs?

(c) Apply the analytical expressions derived for the adaptive array to determine the transient behavior of the DMI algorithm applied to adaptive filters. Assume independent samples.

3. ***Statistical Relations [7].*** Starting with the probability density for $r^2 = [\xi(\hat{\mathbf{w}}) - \xi_{\min}]/\xi_{\min}$, determine the density of the weight vector $\Delta\mathbf{w}$ where $\Delta\mathbf{w} = \hat{\mathbf{w}} - \mathbf{w}_{\text{opt}}$. Show that the covariance of $\Delta\mathbf{w}$ depends on the covariance matrix $\mathbf{R}_{xx}$.

4. ***Statistical Relations [7].*** Starting with the following expression for ρ (for the reference signal case)

$$\rho = \frac{\hat{\mathbf{w}}^\dagger \mathbf{s}\mathbf{s}^\dagger \hat{\mathbf{w}}}{\hat{\mathbf{w}}^\dagger (\mathbf{R}_{xx} - \mathbf{s}\mathbf{s}^\dagger)\hat{\mathbf{w}}} \frac{1}{SN_o}$$

show that

$$\rho = \frac{1}{1 + (1 + SN_o)\sum_{i=2}^{N} \frac{|\Delta q_i|^2}{|\sqrt{SN_o} + \Delta q_1|^2}}$$

where

$$\Delta\mathbf{q} = \sqrt{SN_o}\hat{G} - \begin{bmatrix} \sqrt{SN_o} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\hat{G} = \sqrt{\frac{SN_o}{1 + SN_o}} \mathbf{P}^{-1} \mathbf{R}_{xx}^{1/2} \hat{\mathbf{w}}$$

and where $\mathbf{P}$ is the unitary transformation that gives

$$\sqrt{\frac{SN_o}{1 + SN_o}} \mathbf{R}_{xx}^{1/2} \mathbf{w}_0 = \mathbf{P} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The variable $\Delta\mathbf{q}$ has the probability density function

$$p(\Delta\mathbf{q}) = \frac{k!}{\pi^N (k - N)!} \frac{1}{(1 + \Delta\mathbf{q}^\dagger \Delta\mathbf{q})^{k+1}}$$

5. ***Statistical Relations [7].*** Using the density for $\Delta\mathbf{q}$ in Problem 4 and the following transformation:

$$\begin{aligned} \text{Re}\{\Delta q_1\} &= r \cos\phi_1 \\ \text{Im}\{\Delta q_1\} &= r \sin\phi_1 \cos\phi_2 \\ \text{Re}\{\Delta q_2\} &= r \sin\phi_1 \sin\phi_2 \cos\phi_3 \\ \text{Im}\{\Delta q_2\} &= r \sin\phi_1 \sin\phi_2 \sin\phi_3 \cos\phi_4 \\ &\vdots \end{aligned}$$

$$\text{Re}\{\Delta q_m\} = r \sin\phi_1 \cdots \sin\phi_{2N-2} \cos\phi_{2N-1}$$
$$\text{Im}\{\Delta q_m\} = r \sin\phi_1 \cdots \sin\phi_{2N-2} \sin\phi_{2N-1}$$

where

$$0 \le r < \infty; \qquad i + 1, 2, \cdots, 2N - 1$$
$$0 \le \phi_i < \pi$$

derive the density $p(r, \phi_1, \phi_2)$ of (5.40) and the expression for $\rho(r, \phi_1, \phi_2)$.

Hint: The Jacobian of the transformation is

$$J = r^{2N-1}(\sin\phi_1)^{2N-2}(\sin\phi_2)^{2N-3} \cdots \sin\phi_{2N-2}$$

and

$$\int_0^\pi (\sin\phi)^n d\phi = \frac{\Gamma[(n+1)/2]}{\Gamma[(n+2)/2]}\sqrt{\pi}$$

6. ***Statistical Relations [7].*** Derive the result in (5.44) using (5.33) and the fact that for a CSLC $\xi(\hat{\mathbf{w}}) = \sigma_s^2 + \sigma_N^2$ and $\xi_{\min} = \xi_{N_0} + \sigma_s^2$, and σ_N^2 is the output noise plus jammer power.

7. ***Statistical Relations [7].*** Define the following transformation of $\mathbf{R}'_{xx}$ in (5.29):

$$x = K\left[\hat{\sigma}_0^2 - \hat{\mathbf{r}}_{xx_0}^\dagger \hat{\mathbf{R}}_{xx}^{-1} \hat{\mathbf{r}}_{xx_0}\right]$$
$$\mathbf{Y} = K\hat{\mathbf{R}}_{xx}$$
$$\hat{\mathbf{w}} = \hat{\mathbf{R}}_{xx}^{-1}\hat{\mathbf{r}}_{xd}$$

Then the joint density of x, $\mathbf{Y}$, and $\hat{\mathbf{w}}$ is given by

$$p(x, \mathbf{Y}, \hat{\mathbf{w}}) = P(x)\, P(\hat{\mathbf{w}}, \mathbf{Y})$$

where

$$p(\hat{w}, \mathbf{Y}) = \frac{|Y|^{k-N+1} \exp\{-\text{tr}[I + (1/\xi_0)\Delta\mathbf{w}\,\Delta\mathbf{w}^\dagger \mathbf{R}_{xx}]\,\mathbf{R}_{xx}^{-1}\mathbf{Y}\}}{\pi^N \pi^{1/2N(N-1)}\Gamma(k)\cdots\Gamma(k-N+1)|\mathbf{R}_{xx}|^k |\xi_0|^N}$$

and where $\Delta\mathbf{w} = \hat{\mathbf{w}} - \mathbf{w}_0$. Derive the density of $r^2 = (1/\xi_0)\Delta\mathbf{w}^\dagger \mathbf{R}_{xx}\Delta\mathbf{w}$ [see (5.34)] from the foregoing density function.

8. ***Statistical Relations [7].*** The random variable x defined in Problem 7 has the probability density function

$$p(x) = \frac{1}{(\xi_0)^{K-N}} \frac{1}{\Gamma(K-N)} |x|^{K-N-1} \exp\left[-\frac{x}{\xi_0}\right]$$

(a) Show that $2x/\xi_0$ is χ^2 distributed with $2(K - N)$ degrees of freedom.

(b) Show that $\xi(\hat{\mathbf{w}}) = (1/K)x$ has mean and variance [see (5.31), (5.32)].

$$E\{\xi(\hat{\mathbf{w}})\} = \left(1 - \frac{N}{k}\right)\xi_0$$
$$\text{var}\,[\xi(\hat{\mathbf{w}})] = \frac{K-N}{K^2}\xi_0^2$$

9. ***Inversion of Complex Matrices Using Real Arithmetic [14,15]***
Techniques for inverting complex matrices using only real matrix operations are of considerable value since although complex inversion routines are generally superior (in terms of accuracy and computing time) to real matrix approaches; nevertheless, mathematical packages for small computers generally include matrix inversion routines that apply only to real matrices.

(a) To invert the complex $n \times n$ matrix $M = A + jB$ to obtain $M^{-1} = E + jF$ is equivalent to solving the two simultaneous equations

$$AE - BF = I$$
$$AF + BE = 0$$

for the unknown matrices E and F. Premultiply the two previous equations by B and A, respectively, and by subtracting show that

$$[AB - BA]\, E + [A^2 + B^2]\, F = -B$$

Similarly, premultiply the original equation pair by A and B, respectively, and by adding show that

$$[BA - AB]\, F + [A^2 + B^2]\, E = A$$

If A and B commute (so that $AB = BA$), show that $M^{-1} = [A^2 + B^2]^{-1}[A - jB]$. The foregoing result involves the inverse of a real $n \times n$ matrix to obtain M^{-1} but is restricted by the requirement that A and B commute.

(b) Let $C = A + B$ and $D = A - B$. Show that the original equation pair in part (a) reduce to

$$CE + DF = I$$
$$-DE + CF = -I$$

From the results expressed in the equation pair immediately preceding, show that either

$$M^{-1} = [C + DC^{-1}D]^{-1}[(DC^{-1} + I) + j(DC^{-1} - I)]$$

or

$$M^{-1} = [D + CD^{-1}C]^{-1}[(CD^{-1} + I) + j(-CD^{-1} + I)]$$

provided the indicated inverses exist. The foregoing equation pair represents alternate ways of obtaining M^{-1} by inverting real $n \times n$ matrices without the restriction that A and B commute.

(c) An isomorphism exists between the field of complex numbers and a special set of 2×2 matrixes, that is,

$$a + jb \sim \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

Consider the $2n \times 2n$ real matrixes defined by

$$\mathbf{G} \triangleq \begin{bmatrix} A & B \\ -B & A \end{bmatrix} \quad \text{and} \quad \mathbf{H} = \begin{bmatrix} E & F \\ -F & E \end{bmatrix}$$

Show that H is the inverse of G only if the original equation pair in part (a) is satisfied. Therefore, one way of obtaining M^{-1} is to compute G^{-1} and identify the $n \times n$ submatrices E and F appearing in G^{-1}. Then $M^{-1} = E + jF$. This approach does not involve the restrictions that beset the approaches of (a) and (b) but suffers from the fact that it requires the inversion of a $2n \times 2n$ matrix that drastically increases computer storage requirements, and therefore should be used only as a last resort.

10. Development of the Cholesky Decomposition Algorithm [11]
Consider the quadratic form $\mathrm{x^T Px}$ for the case $n = 3$, where P is positive definite and symmetric.

(a) Express the quadratic form as a difference of squares plus a remainder where the two squares involve only P(1, 1), P(1, 2), and P(1, 3) and the remainder involves P(2, 2), P(2, 3) and P(3, 3).

(b) Now set $\mathrm{L}(1,1) = \mathrm{P}(1,1)^{1/2}, \mathrm{L}(2,1) = \mathrm{P}(2,1)/\mathrm{L}(1,1)$, and $\mathrm{L}(3,1) = \mathrm{P}(3,1)/\mathrm{L}(1,1)$. Furthermore let $y_1 = \sum_{i=1}^{3} L(i,1)x_i$. With this notation rewrite $\mathbf{x^T Px}$ as

$$\begin{aligned} &\mathrm{y}_1^2 + [\mathrm{P}(2,2) - \mathrm{L}(2,1)L(2,1)]\mathrm{x}_2^2 + 2[\mathrm{P}(2,3) - \mathrm{L}(2,1)\mathrm{L}(3,1)]\mathrm{x}_2\mathrm{x}_3 \\ &+ [\mathrm{P}(3,3) - \mathrm{L}(3,1)\mathrm{L}(3,1)]\mathrm{x}_3^2 = \mathrm{y}_1^2 + \overline{P}(2,2)\mathrm{x}_2^2 + 2\overline{P}(2,3)\mathrm{x}_2\mathrm{x}_3 + \overline{P}(3,3)\mathrm{x}_3^2 \end{aligned}$$

where $\overline{P}(\mathrm{i},\mathrm{j}) = \mathrm{P}(\mathrm{i},\mathrm{j}) - \mathrm{L}(\mathrm{I},1)\mathrm{L}(\mathrm{j},1)$ for $\mathrm{i} = 2, 3$ and $\mathrm{j} = \mathrm{i}, 3$.

(c) Now complete the square on the x_2 variables, that is,

$$\begin{aligned} \overline{P}(2,2)\mathrm{x}_2^2 + 2\overline{P}(2,3)\mathrm{x}_2\mathrm{x}_3 + \overline{P}(3,3)\mathrm{x}_3^2 = [\overline{P}(2,2)^{1/2}\mathrm{x}_2 + (\overline{P}(2,3)/\overline{P}(2,2)^{1/2})\mathrm{x}_3]^2 \\ -[(\overline{P}(2,3)/\overline{P}(2,2)^{1/2})\mathrm{x}_3]^2 + \overline{P}(3,3)\mathrm{x}_3^2 \end{aligned}$$

(d) Consistent with part (b), set $\mathrm{L}(2,2) = \overline{P}(2,2)^{1/2}$, $\mathrm{L}(3,2) = \overline{P}(3,2)/\mathrm{L}(2,2)$, and $\mathrm{y}_2 = \mathrm{L}(2,2)\mathrm{x}_2 + \mathrm{L}(3,2)\mathrm{x}_3$.
Now by setting $\mathrm{L}(3,3) = [\overline{P}(3,3) - \mathrm{L}(3,2)\mathrm{L}(3,2)]^{1/2}$ and $\mathrm{y}_3 = \mathrm{L}(3,3)\mathrm{x}_3$, we now have $\mathbf{x^T Px} = \mathbf{y^T y} = \mathbf{x^T L L^T x}$.

11. Development of the Inversion of a Triangular Matrix [11]
The following identity for triangular matrices is easily verified:

$$\begin{bmatrix} R_j & y \\ 0 & \sigma_{j+1} \end{bmatrix}^{-1} = \begin{bmatrix} R_j^{-1} & -R_j^{-1} y \sigma_{j+1}^{-1} \\ 0 & \sigma_{j+1}^{-1} \end{bmatrix} = R_{j+1}^{-1}$$

This relation enables one to compute recursively the inverse of an ever larger matrix, that is, if $\mathrm{R}_j^{-1} = \mathrm{U_j}$, where $\mathrm{R_j}$ is the upper left $\mathrm{j} \times \mathrm{j}$ partition of R, then show that

$$\mathrm{U_{j+1}} = \begin{bmatrix} U_j & -U_j\,(R(1,j+1), \ldots, R(j,j+1))^T\,\sigma_{j+1} \\ 0 & \sigma_{j+1} \end{bmatrix}$$

where $\sigma_{j+1} = 1/R(j+1, j+1)$. This result is readily transformed into the desired algorithmic form given by (5.57), (5.58), and (5.59).

12. Structure of the Householder Transformation [16]
The Householder transformation can be written as

$$\mathbf{H} = \mathbf{I} - \frac{2uu^H}{\|u\|^2} = \mathbf{I} - 2\mathrm{u}[\mathrm{u}^{\mathrm{H}}\,\mathrm{u}]^{-1}\mathrm{u}^{\mathrm{H}} = \mathbf{I} - 2\mathbf{P}_{\mathrm{u}}$$

where $\mathbf{P}_{\mathrm{u}} = \mathrm{u}[\mathrm{u}^{\mathrm{H}}\,\mathrm{u}]^{-1}\mathrm{u}^{\mathrm{H}}$

(a) Using the fact that $\mathbf{H}$ is Hermitian unitary ($\mathbf{H}^{-1} = \mathbf{H}^{\mathrm{H}} = \mathbf{H}$), show that

$$\|\mathbf{Hv}\| = \|\mathbf{v}\| \text{ (length of } \mathbf{v} \text{ is preserved where } \mathbf{v} = [\mathrm{v}_1\mathrm{v}_2 \ldots \mathrm{v}_{\mathrm{N}}]^{\mathrm{T}})$$

(b) Define $\mathrm{v}_{\mathrm{in}} = \frac{\mathrm{v}_i}{|\mathrm{v}_i|}$ where v_i denotes the ith element of v.
We can now proceed to show that $\mathbf{H}$ can be used to eliminate all the elements of v except the ith element.
Let $\mathrm{u} = \mathrm{v} + \mathrm{v}_{\mathrm{in}}\|\mathbf{v}\|e_i$ where $e_i = [0, \ldots, 0, 1, 0, \ldots 0]^{\mathrm{T}}$, and 1 is the ith element of e_i.
Show that $\mathrm{u}^{\mathrm{H}}\mathrm{v} = (\mathrm{v}^{\mathrm{H}} + (\mathbf{v}^H + \mathrm{v}_{in}^*\|\mathbf{v}\|e_i^T)\mathrm{v} = \|\mathbf{v}\|^2 + |\mathrm{v}_i|\|\mathbf{v}\|$

(c) Using the result of part (b), show that

$$\mathbf{Hv} = \left(\mathbf{I} - \frac{2\mathbf{u}\mathbf{u}^H}{\|\mathbf{u}\|^2}\right)\mathbf{v} = \mathbf{v} - \mathbf{u} = -\mathrm{v}_{\mathrm{in}}\|\mathbf{v}\|e_i, \text{ which is the desired result.}$$

13. Equivalence of Diagonal Loading and Omnidirectional Jamming [17]
Consider a narrowband, uniform linear array consisting of n elements with spacing d where each element has a cosine element pattern. Let the external interference environment consist of a uniform continuum of jammers from all azimuth angles, θ, as measured from broadside to the array.

(a) Show that the correlation element of the covariance matrix for the pair of elements with indexes n and m is given by

$$R_{n,m} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\theta) \exp\left(j\frac{2\pi}{\lambda}d(n-m)\sin(\theta)\right) d\theta$$

(b) Making the change of variable $u = \sin(\theta)$, show that $R_{n,m}$ of part (a) becomes

$$R_{n,m} = \int_{-1}^{1} \exp\left(j\frac{2\pi}{\lambda}d(n-m)u\right) du$$

(c) Part (b) becomes

$$\begin{aligned} R_{n,m} &= \frac{1}{j\pi(n-m)}\left[\exp(j\pi(n-m)) - \exp(-j\pi(n-m))\right] \\ &= 2\frac{\sin(\pi(n-m))}{\pi(n-m)} \\ &= 2 \text{ for } n-m = 0 \\ &= 0 \text{ for } n-m \neq 0 \end{aligned}$$

14. ***Computer Simulation Problem*** An eight-element uniform array with $\lambda/2$ spacing has the desired signal incident at 0° and two interference signals incident at -21° and 61°. Use the DMI algorithm to place nulls in the antenna pattern. Assume $\sigma_{noise} = 0.01$. In MATLAB, represent the signal by $\cos(2\pi\ (1:K)/K)\exp(j\ \text{rand})$ and the interference by sign(randn(1, K)).

15. ***Development of the Inverse of a Toeplitz Matrix [5]***

(a) A Toeplitz matrix has the form

$$\begin{bmatrix} \tau_0 & \tau_{-1} & \tau_{-2} & \tau_{-3} \\ \tau_1 & \tau_0 & \tau_{-1} & \tau_{-2} \\ \tau_2 & \tau_1 & \tau_0 & \tau_{-1} \\ \tau_3 & \tau_2 & \tau_1 & \tau_0 \end{bmatrix}$$

As a result, a normalized Toeplitz matrix can be written as

$$L_{n+1} = \lambda_n \begin{bmatrix} 1 & a^t \\ r & L_n \end{bmatrix}$$

where a^t is a $1xn$ row vector
r is an $nx1$ column vector
L_n is an nxn Toeplitz matrix

Postulate that the inverse of L_{n+1}, denoted as B_{n+1}, can be written as

$$B_{n+1} = \frac{1}{\lambda_n} \begin{bmatrix} 1 & e^t \\ g & M_n \end{bmatrix}$$

By taking the product $B_{n+1}L_{n+1,}$ show that $\lambda_n = 1 + e^t r$ and $g = -M_n r$

(b) There now remain two terms in the product $B_{n+1}L_{n+1}$ to be examined. Multiply each of these terms by $B_n = L_n^{-1}$ to show that $e^t = -a^t B_n$ and $M_n = \lambda_n B_n + g e^t$ Therefore $\lambda_n B_{n+1} = \begin{bmatrix} 1 & e^t \\ g & M_n \end{bmatrix} = \begin{bmatrix} 1 & e^t \\ g & \lambda_n B_n + g e^t \end{bmatrix}$, which shows that all elements of B_{n+1} can be expressed as a function of B_n.

(c) Define a persymmetric Exchange Matrix as $E = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \ldots & 0 \end{bmatrix}$ where persymmetric means symmetry about the cross-diagonal.
Show that $EP^t E = P$ where P is persymmetric.
and $(P^t)^{-1}P^t = \underline{I}$
Likewise show that $EE = \underline{I}$ and $E(P^t)^{-1}E = P^{-1}$

(d) Use the fact that

$$\lambda_n B_{n+1} = E_{n+1} \begin{bmatrix} 1 & g^t \\ e & \lambda_n B_n + e g^t \end{bmatrix} E_{n+1}$$

to show that

$$\lambda_n B_{n+1} = \begin{bmatrix} \lambda_n B_n + \hat{e}\hat{g}^t & \hat{e} \\ \hat{g}^t & 1 \end{bmatrix}$$

where $Eg = \hat{g}$ and $Ee = \hat{e}$. This results shows that given an element of $B_{n+1,}$ all the remaining elements along the same diagonal are given if we know λ_n, g_n, and e_n (the elements of the 1^{st} row and column of B_{n+1}.

(e) Since B_{i+1} is persymmetric, we can write

$$e_{i+1} = -(E_{i+1}B_{i+1}E_{i+1})a_{i+1} = -E_{i+1}B_{i+1}\hat{a}_{i+1}$$

Writing this result out explicitly then yields

$$e_{i+1} = -\frac{1}{\lambda_i}E_{i+1}\begin{bmatrix} 1 & e_i^t \\ g_i & \lambda_i B_i + g_i e_i^t \end{bmatrix}\begin{bmatrix} \rho_{-(i+1)} \\ \hat{a}_i \end{bmatrix}, \quad 1 \le i < n$$

where $\hat{a}_{i+1}$ has been expressed as $\hat{a}_i$ augmented by the term $\rho_{-(i+1)}$. The above result allows one to write a recursion relationship for e_{i+1}:

$$e_{i+1} = \begin{bmatrix} e_i \\ 0 \end{bmatrix} - \frac{\rho_{-(i+1)} + e_i^t \hat{a}_i}{\lambda_i}\begin{bmatrix} \hat{g}_i \\ 1 \end{bmatrix}, \quad 1 \le i < n$$

Symmetry relations then allow a corresponding recursion relationship for g_{i+1} to be written.

5.8 REFERENCES

[1] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Sample Matrix Inversion Technique," *Proceedings of the 1974 Adaptive Antenna Systems Workshop,* March 11–13, Vol. 1, NRL Report 7803, Naval Research Laboratory, Washington, DC, pp. 219–222.

[2] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *FIEEE Trans. Aerosp. Electron. Sys.,* Vol. AES-10, No. 6, November 1974, pp. 853–863.

[3] T. W. Miller, "The Transient Response of Adaptive Arrays in TDMA Systems," Ph.D. Dissertation, Department of Electrical Engineering, The Ohio State University, 1976.

[4] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part 1,* New York, Wiley, 1968, Ch. 1.

[5] S. Zohar, "Toeplitz matrix inversion: the algorithm of W. F. Trench," J. Assoc. Comput. Mach, Vol 16, Oct 1969, pp. 592–601.

[6] N. R. Goodman, "Statistical Analysis Based on a Certain Multivariate Gaussian Distribution," *Ann. Math. Stat.,* Vol. 34, March 1963, pp. 152–177.

[7] A. B. Baggeroer, "Confidence Intervals for Regression (MEM) Spectral Estimates," *IEEE Trans. Info. Theory,* Vol. IT-22, No. 5, September 1976, pp. 534–545.

[8] B. D. Carlson, "Covariance Matrix Estimation Errors and Diagonal Loading in Adaptive Arrays," *IEEE Trans. on Aerosp. & Electron. Sys.,* Vol. AES-24, No. 4, July 1988, pp. 397–401.

[9] M. W Ganz, R. L. Moses, and S. L. Wilson, "Convergence of the SMI and the Diagonally Loaded SMI Algorithms with Weak Interference," *IEEE Trans. Ant. & Prop.,* Vol. AP-38, No. 3, March 1990, pp. 394–399.

[10] I. J. Gupta, "SMI Adaptive Antenna Arrays for Weak Interfering Signals," *IEEE Trans. Ant.& Prop.,* Vol. AP-34, No. 10, October 1986, pp. 1237–1242.

[11] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation,* New York, Academic Press, 1977.

[12] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for Modifying Matrix Factorizations," Mathematics of Computation, Vol. 28, No. 126, April 1974, pp. 505–535.

[13] J. E. Gentle, "Cholesky Factorization," Section 3.2.2 in *Numerical Linear Algebra for Applications in Statistics,* Berlin: Springer-Verlag, 1998, pp. 93–95.

[14] H. L. Van Trees, *Optimum Array Processing, Part IV of Detection, Estimation and Modulation Theory,* New York, Wiley-Interscience, 2002.

[15] W. W. Smith, Jr. and S. Erdman, "A Note on Inversion of Complex Matrices," *IEEE Trans. Automatic Control,* Vol. AC-19, No. 1, February 1974, p. 64.

[16] M. E. El-Hawary, "Further Comments on 'A Note on Inversion of Complex Matrices,' " *IEEE Trans. Automatic Control,* Vol. AC-20, No. 2, April 1975, pp. 279–280.

[17] B. D. Carlson, "Equivalence of Adaptive Array Diagonal Loading and Omnidirectional Jamming," *IEEE Trans. Ant. & Prop.,* Vol. Ap-43, No. 5, May 1995, pp. 540–541.

CHAPTER 6

Recursive Methods for Adaptive Array Processing

Chapter Outline

The least mean squares (LMS) and maximum signal-to-noise ratio (SNR) algorithms avoid the computational problems associated with the direct calculation of a set of adaptive weights. Chapter 8 shows that random search algorithms also circumvent computational problems. These algorithms have the advantage that the required calculations are usually much simpler than the corresponding direct calculation, are less susceptible to hardware inaccuracy, and are continually updated to compensate for a time-varying signal environment.

Recursively inverting the matrix circumvents many computational problems [1–4]. The recursive algorithms exhibit a steady-state sensitivity to eigenvalue spread in the signal covariance matrix as found for direct matrix inversion (DMI) algorithms. Furthermore, since the principal difference between the recursive methods and the DMI algorithms lies in the manner in which the matrix inversion is computed, their rates of convergence are comparable. The recursive algorithms are based on least square estimation techniques and are closely related to Kalman filtering methods [5]. For stationary environments, these recursive procedures compute the best possible selection of weights (based on a least squares fit to the data received) at each sampling instant, whereas in contrast the LMS, maximum SNR, and random search methods are only asymptotically optimal.

6.1 | THE WEIGHTED LEAST SQUARES ERROR PROCESSOR

Consider the conventional N-element array of Figure 6-1 having a sequence of (real or complex) weights multiplying the received signals to form the adaptive processor.

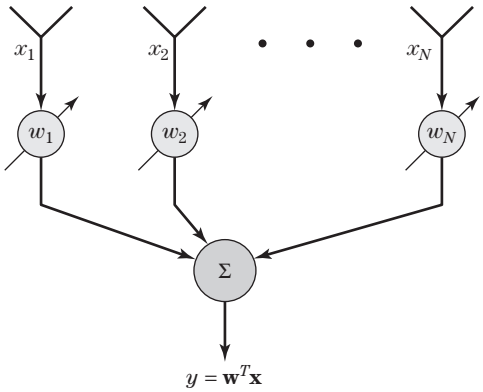


FIGURE 6-1 ■ Conventional N-element adaptive array processor.

Assume that the received signals $x_i(t)$ contain a directional desired signal component $s_i(t)$ and a purely random component $n_i(t)$, due to both directional and thermal noise so that $x_i(t) = s_i(t) + n_i(t)$. Collecting the received signals $x_i(t)$ and the multiplicative weights $w_i(t)$ as components in the N-dimensional vectors $\mathbf{x}(t)$ and $\mathbf{w}(t)$, we write the adaptive processor output signal $y(t)$ as

$$y(t) = \mathbf{w}^T(t)\mathbf{x}(t) \tag{6.1}$$

The narrowband processor model of Figure 6-1 has been chosen instead of the more general tapped delay line wideband processor in each element channel because the mathematical manipulations are simplified. The weighted least squares error processor extends to the more general tapped delay line form.

Consider the weighted least squares performance measure based on k data samples following Baird [6]

$$\Re(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{k}\alpha_i[\mathbf{w}^T\mathbf{x}(i) - d(i)]^2 = \frac{1}{2}[\mathbf{X}(k)\mathbf{w} - \mathbf{d}(k)]^T\mathbf{A}_k^{-1}[\mathbf{X}(k)\mathbf{w} - \mathbf{d}(k)] \tag{6.2}$$

where the elements of $\mathbf{X}(k)$ are received signal vector samples, and the elements of $\mathbf{d}(k)$ are desired (or reference) signal samples as follows:

$$\mathbf{X}(k) \triangleq \begin{bmatrix} \mathbf{x}^T(1) \\ \mathbf{x}^T(2) \\ \vdots \\ \mathbf{x}^T(k) \end{bmatrix} \tag{6.3}$$

$$\mathbf{d}(k) \triangleq \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(k) \end{bmatrix} \tag{6.4}$$

Both (6.4) and (6.2) presume that the desired array output signal, $d(t)$, is known and sampled at $d(1), d(2), \ldots, d(k)$. In the performance measure of (6.2), $\mathbf{A}_k$ is a diagonal

weighting matrix that deemphasizes old data points and is of the form

$$\mathbf{A}_k = \begin{bmatrix} \alpha^{k-1} & 0 & \cdots & . & 0 \\ 0 & \alpha^{k-2} & \cdots & . & . \\ & & & & \vdots \\ 0 & \cdots & & \alpha & 0 \\ 0 & 0 & & & 1 \end{bmatrix} \tag{6.5}$$

where $0 < \alpha \leq 1$, so that older data have increasingly less importance. If the signal environment is stationary so that all data samples are equally important, then $\mathbf{A}_k = \mathbf{I}$, the identity matrix. The performance measure given by (6.2) is minimized by selecting the weight vector to yield the "best" (weighted least squares) estimate of the desired signal vector $\mathbf{d}(k)$.

To minimize the weighted least squares performance measure (6.2), set the derivative of $\mathfrak{P}(\mathbf{w})$ with respect to $\mathbf{w}$ equal to zero, thereby yielding the optimum weight setting as

$$\mathbf{w}_{ls}(k) = \left[\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}(k)\right]^{-1}\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{d}(k) \tag{6.6}$$

When an additional data sample is taken, the foregoing weight vector solution is updated in the most efficient manner. The updated signals $\mathbf{X}(k+1)$ and $\mathbf{d}(k+1)$ as well as the updated matrix $\mathbf{A}_{k+1}$ can each be partitioned as follows:

$$\mathbf{X}(k+1) = \left[\frac{\mathbf{X}(k)}{\mathbf{x}^T(k+1)}\right] \tag{6.7}$$

$$\mathbf{d}(k+1) = \left[\frac{\mathbf{d}(k)}{d(k+1)}\right] \tag{6.8}$$

and

$$\mathbf{A}_{k+1} = \left[\begin{array}{ccc:c} & & & 0 \\ & & & \vdots \\ & \alpha\mathbf{A}_k & & \\ & & & 0 \\ \hdashline 0 \cdots & 0 & \vdots & 1 \end{array}\right] \tag{6.9}$$

With this partitioning, the updated weight vector can be written as

$$\begin{aligned} \mathbf{w}_{ls}(k+1) &= \left[\mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{X}(k+1)\right]^{-1}\mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{d}(k+1) \\ &= \left[\mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{X}(k+1)\right]^{-1} \\ &\quad \cdot \left[\alpha\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{d}(k) + \mathbf{x}(k+1)\,d(k+1)\right] \end{aligned} \tag{6.10}$$

From (6.10) it is seen that the updated weight vector solution requires the inversion of the matrix $[\mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{X}(k+1)]$, which can also be expanded by the partitioning previously given as

$$\begin{aligned} \left[\mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{X}(k+1)\right]^{-1} &= \left\{ [\mathbf{X}^T(k)\,|\,\mathbf{x}(k+1)] \left[\begin{array}{ccc:c} & & & 0 \\ & & & \vdots \\ & \alpha\mathbf{A}_k^{-1} & & \\ & & & 0 \\ \hdashline 0 \cdots & 0 & \vdots & 1 \end{array}\right] \left[\frac{\mathbf{X}(k)}{\mathbf{x}^T(k+1)}\right] \right\}^{-1} \\ &= \left\{\alpha\left[\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}^T(k)\right] + \mathbf{x}(k+1)\mathbf{x}^T(k+1)\right\}^{-1} \end{aligned} \tag{6.11}$$

Now define

$$\mathbf{P}^{-1}(k) \triangleq \mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}(k) \tag{6.12}$$

Likewise, define

$$\begin{aligned}\mathbf{P}^{-1}(k+1) &\triangleq \mathbf{X}^T(k+1)\mathbf{A}_{k+1}^{-1}\mathbf{X}(k+1)\\ &= \alpha\left[\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}(k)\right] + \mathbf{x}(k+1)\mathbf{x}^T(k+1)\\ &= \alpha\left[\mathbf{P}^{-1}(k) + \frac{1}{\alpha}\mathbf{x}(k+1)\mathbf{x}^T(k+1)\right]\end{aligned} \tag{6.13}$$

Inverting both sides of (6.13) and applying the matrix inversion lemma [(D.10) of Appendix D] then yields

$$\mathbf{P}(k+1) = \frac{1}{\alpha}\left\{\mathbf{P}(k) - \frac{\mathbf{P}(k)\mathbf{x}(k+1)\mathbf{x}^T(k+1)\mathbf{P}(k)}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}\right\} \tag{6.14}$$

By our use of (6.14) in (6.10) and recognition that $\mathbf{w}_{ls}(k) = \mathbf{P}(k)\mathbf{X}(k)\mathbf{A}_k^{-1}\mathbf{d}(k)$, the updated solution for the weight vector now becomes

$$\begin{aligned}\mathbf{w}_{ls}(k+1) = \mathbf{w}_{ls}(k) &+ \frac{\mathbf{P}(k)\mathbf{x}(k+1)}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}\\ &\cdot\left[d(k+1) - \mathbf{w}_{ls}^T(k)\mathbf{x}(k+1)\right]\end{aligned} \tag{6.15}$$

Equations (6.14) and (6.15) are the iterative relations of the recursive least squares algorithm. Equations (6.14) and (6.15) are started by adopting an initial guess for the weight vector $\mathbf{w}(0)$ and the initial Hermitian matrix $\mathbf{P}(0)$. It is common practice to select as an initial weight vector $\mathbf{w}(0) = [1\angle 0^\circ, 0, 0, \ldots, 0]$, thereby obtaining an omnidirectional array pattern (provide the sensor elements each have omnidirectional patterns) and to select $\mathbf{P}(0)$ as the identity matrix.

Equations (6.14) and (6.15) yield the updated weight vector in a computationally efficient manner that avoids calculating the matrix inverses present in (6.6) and (6.10). It is instructive to consider (6.6) in more detail for the additional insight to be gained into the mechanics of the processor. Since the trace of $\mathbf{A}_k$, tr $[\mathbf{A}_k]$, is a scalar, (6.6) can be rewritten as

$$\mathbf{w}_{ls}(k) = \left\{\frac{\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}(k)}{\text{tr}\,[\mathbf{A}_k]}\right\}^{-1}\left\{\frac{\mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{d}(k)}{\text{tr}\,[\mathbf{A}_k]}\right\} \tag{6.16}$$

The first bracketed term on the right-hand side of (6.16) is an estimate of the autocorrelation matrix $\hat{\mathbf{R}}_{xx}$ based on k data samples, that is,

$$\left[\hat{\mathbf{R}}_{xx}\right]_{i,j}(k) = \sum_{n=1}^{k}\alpha^{k-n}x_i(k)x_j(k) \tag{6.17}$$

Equation (6.17) is an expression for forming exponentially deweighted estimates of the matrix $\mathbf{R}_{xx}$ also used by Mantey and Griffiths [2]. Similarly, the second bracketed term on the right-hand side of (6.16) is an estimate for the cross-correlation vector $\hat{\mathbf{r}}_{xd}(k)$.

Consequently, the **P** matrix defined by (6.12) is directly related to the autocorrelation matrix since

$$\hat{\mathbf{R}}_{xx}^{-1}(k) = \text{tr}\,[\mathbf{A}_k]\mathbf{P}(k) = (1 + \alpha + \alpha^2 + \cdots + \alpha^{k-1})\mathbf{P}(k)$$
$$= \left[\frac{1-\alpha^k}{1-\alpha}\right]\mathbf{P}(k) \tag{6.18}$$

The form of the algorithm given by (6.14) and (6.15) requires that the desired signal be known at each sample point, which is an unrealistic assumption. An estimate of the desired signal is used in practice, Consequently, replacing $d(k+1)$ by $\hat{d}(k+1)$ in (6.15) results in a practical weighted least square error recursive algorithm.

Other useful forms of (6.15) arise from replacing certain instantaneous quantities by their known average values [3,7]. To obtain these equivalent forms, rewrite (6.15) as

$$\mathbf{w}(k+1) = \text{w}(k) + \frac{P(k)}{\alpha + x^T(k+1)P(k)x(k+1)}$$
$$\cdot \left[\mathbf{x}(k+1)d(k+1) - \mathbf{x}(k+1)y(k+1)\right] \tag{6.19}$$

where $y(k+1)$ is the array output. The product $\mathbf{x}(k+1) \cdot d(k+1)$ is replaced by its average value, which is an estimate of the cross-correlation vector $\hat{\mathbf{r}}_{xd}$. Since the estimate $\hat{\mathbf{r}}_{xd}$ does not follow instantaneous fluctuations of $\mathbf{x}(t)$ and $d(t)$, it may be expected that the convergence time would be greater using $\hat{\mathbf{r}}_{xd}$ than when using $\mathbf{x}(k)\,d(k)$ as shown in Chapter 5.

In the event that only the direction of arrival of the desired signal is known and the desired signal is absent, then the cross-correlation vector $\mathbf{r}_{xd}$ (which conveys direction of arrival information) is known, and the algorithm for updating the weight vector becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mathbf{P}(k)}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}\left[\mathbf{r}_{xd} - \mathbf{x}(k+1)y(k+1)\right] \tag{6.20}$$

Equation (6.20) may of course also be used when the desired signal is present, but the rate of convergence is then slower than for (6.19) with the same desired signal present conditions.

For stationary signal environments $\alpha = 1$, but this choice is not practical. As long as $0 < \alpha < 1$, (6.14) and (6.15) lead to stable numerical procedures. When $\alpha = 1$, however, after many iterations the components of $\mathbf{P}(k)$ become so small that round-off errors have a significant impact. To avoid this numerical sensitivity problem, both sides of (6.14) are multiplied by the factor $(k+1)$ to yield numerically stable equations, and (6.18) then becomes

$$\hat{\mathbf{R}}_{xx}^{-1}(k) = k\mathbf{P}(k) \tag{6.21}$$

6.2 | UPDATED COVARIANCE MATRIX INVERSE

The weighted least squares error processor of the previous section was based on weighting the current received signal vector data compared with past data according to

$$\mathbf{P}^{-1}(k+1) = \alpha\mathbf{P}^{-1}(k) + \mathbf{x}^*(k+1)\mathbf{x}^T(k+1) \tag{6.22}$$

where $0 \leq \alpha \leq 1$ and $\mathbf{P}^{-1}(k) \triangleq \mathbf{X}^T(k)\mathbf{A}_k^{-1}\mathbf{X}(k)$. A closely related alternative data weighting scheme uses the sample covariance matrix, $\hat{\mathbf{R}}_{xx}$, for summarizing the effect of old data, so that

$$\hat{\mathbf{R}}_{xx}(k+1) = \alpha\hat{\mathbf{R}}_{xx}(k) + \mathbf{x}^*(k+1)\mathbf{x}^T(k+1) \tag{6.23}$$

where $0 \leq \alpha \leq 1$. Inverting both sides of (6.24) yields

$$\hat{\mathbf{R}}_{xx}^{-1}(k+1) = \frac{1}{\alpha}\left\{\hat{\mathbf{R}}_{xx}(k) + \frac{1}{\alpha}\mathbf{x}^*(k+1)\mathbf{x}^T(k+1)\right\}^{-1} \tag{6.24}$$

Applying the matrix inversion lemma to (6.24) then results in

$$\hat{\mathbf{R}}_{xx}^{-1}(k+1) = \frac{1}{\alpha}\left\{\hat{\mathbf{R}}_{xx}^{-1}(k) - \frac{\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)\mathbf{x}^T(k+1)}{\alpha + \mathbf{x}^T(k+1)\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)}\hat{\mathbf{R}}_{xx}^{-1}(k)\right\} \tag{6.25}$$

In the absence of a desired signal

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{b}^* \tag{6.25}$$

(or $\mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}$) so that each side of (6.25) can be postmultiplied by $\mathbf{b}^*$ to give

$$\hat{\mathbf{w}}(k+1) = \frac{1}{\alpha}\left\{\hat{\mathbf{w}}(k) - \frac{\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)}{\alpha + \mathbf{x}^T(k+1)\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)}\mathbf{x}^T(k+1)\hat{\mathbf{w}}(k)\right\} \tag{6.26}$$

This is known as the recursive least squares (RLS) algorithm, because it recursively updates the correlation matrix such that more recent time samples receive a higher weighting than past samples [19].

As an example, consider a four-element uniform linear array with $\lambda/2$ spacing. The desired signal is incident at 0°, and one interference signal is incident at 45° with $\sigma_{noise} = 0.01$. After $K = 25$ iterations and $\alpha = 0.9$, the antenna pattern appears in Figure 6-2 with a directivity of 5.6 dB, which is less than the quiescent pattern directivity of 6 dB. Figure 6-3 and Figure 6-4 are the RLS weights as a function of time. They converge in about 20 iterations.

The data weighting represented by (6.26) implies that past data [represented by $\hat{\mathbf{R}}_{xx}(k)$] is never more important than current data [represented by $\mathbf{x}^*(k+1)\mathbf{x}^T(k+1)$]. An

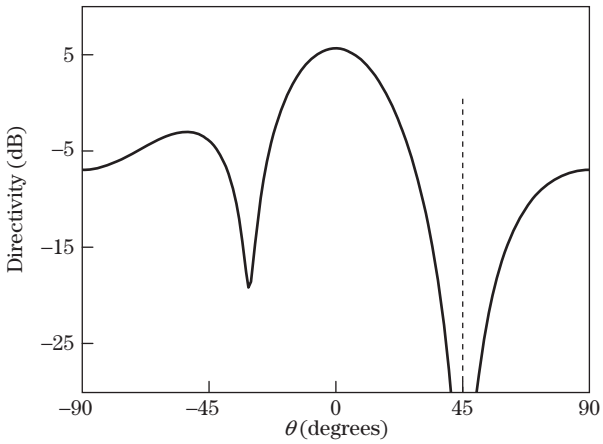


FIGURE 6-2 ■ RLS adapted pattern.

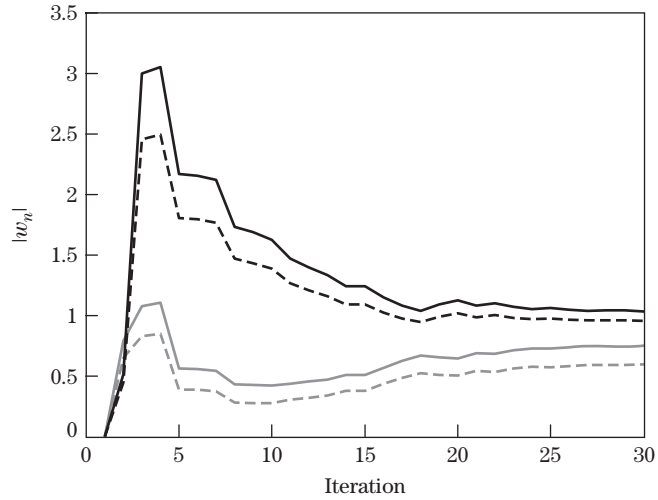


FIGURE 6-3 ■ Magnitude of the RLS weights as a function of iteration.

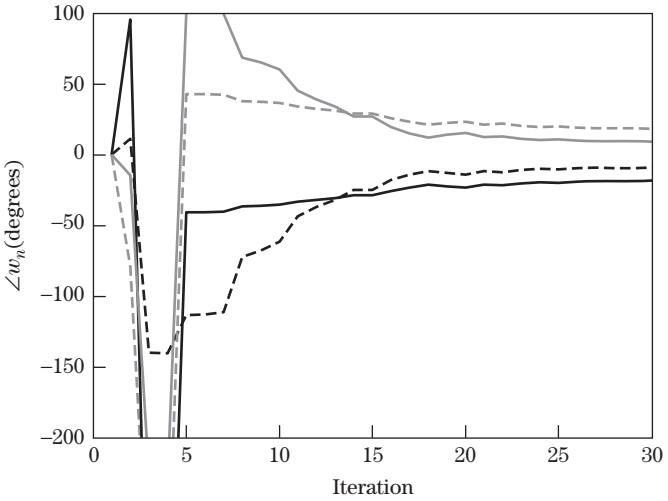


FIGURE 6-4 ■ Phase of the RLS weights as a function of iteration.

alternative data weighting scheme that permits past data to be regarded either as less important or more important than current data is to use

$$\hat{\mathbf{R}}_{xx}(k+1) = (1-\beta)\hat{\mathbf{R}}_{xx}(k) + \beta\mathbf{x}^*(k+1)\mathbf{x}^T(k+1), \quad 0 \le \beta \le 1 \tag{6.27}$$

The data weighting scheme represented by (6.27) has been successfully employed [using $\beta = 1/(k+1)$ so that each sample is then equally weighted] to reject clutter, to compensate for platform motion, and to compensate for near-field scattering effects in an airborne moving target indication (AMTI) radar system [8]. Inverting both sides of (6.27) and applying the matrix inversion lemma results in [9]

$$\hat{\mathbf{R}}_{xx}^{-1}(k+1) = \frac{1}{(1-\beta)}\hat{\mathbf{R}}_{xx}^{-1}(k) - \frac{\beta}{(1-\beta)} \cdot \frac{\left[\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)\right]\left[\mathbf{x}^T(k+1)\hat{\mathbf{R}}_{xx}^{-1}(k)\right]}{(1-\beta)+\beta\left[\mathbf{x}^T(k+1)\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)\right]} \tag{6.28}$$

To obtain the updated weight vector $\hat{\mathbf{R}}_{xx}^{-1}(k+1)$ must be postmultiplied by the beam steering vector $\mathbf{b}^*$. The beam steering vector for AMTI radar systems is matched to a moving target by including the expected relative Doppler and polarization phase factors, thereby minimizing the effects of main beam clutter due to (for example) stationary targets [8,10]. Carrying out the postmultiplication of (6.28) by $\mathbf{b}^*$ then yields

$$\hat{\mathbf{w}}(k+1) = \frac{1}{(1-\beta)} \left\{ \hat{\mathbf{w}}(k) - \beta \frac{\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)}{(1-\beta) + \beta\left[\mathbf{x}^T(k+1)\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)\right]} \mathbf{x}^T(k+1)\hat{\mathbf{w}}(k) \right\} \tag{6.29}$$

The updating computation represented by (6.28) requires N^2 complex multiplications to form $\hat{\mathbf{R}}_{xx}^{-1}(k)\mathbf{x}^*(k+1)$, where N is the number of degrees of freedom present in the adaptive processor. Furthermore, (6.28) also requires an additional $0.75N^2$ and $2.25N$ multiplications to complete the computation of $\hat{\mathbf{R}}_{xx}^{-1}(k+1)$. On comparing (6.29) and (6.26) with (6.20), it may appear that the direction-of-arrival information contained in $\mathbf{r}_{xd}$ (or $\mathbf{b}^*$) is missing in (6.26) and (6.29). We realize that the initial starting weight selected in either (6.26) or (6.29) reflects any direction-of-arrival information, however, so that (6.20), (6.26), and (6.29) merely represent different data weighting versions of the same basic weight update equation.

6.2.1 Recursive Algorithms Based on Factorization Methods

The recursive update equation (6.29) is not the most accurate and numerically stable equation possible using the factorization methods introduced in Chapter 5. In [11,12], recursive update equations are described that possess highly desirable accuracy and stability properties superior to those of (6.29). We will consider the solution to a linearly constrained optimization problem like that of Chapter 4. It will then be easy to see how the various algorithms based on factorization methods relate to the development given here.

An array output is described by

$$y(n) = \mathbf{w}^\dagger(n)\mathbf{x}(n), \tag{6.30}$$

where $\mathbf{x}(n)$ is an $N \times 1$ data vector describing the outputs of each array element at time nT and $\dagger$ denotes complex conjugate transpose. A linear constraint for (6.30) is given by

$$\mathbf{C}^\dagger\mathbf{w}(n) = \mathbf{f} \tag{6.31}$$

where $\mathbf{C}$ is an $N \times K$ constraint matrix, and $\mathbf{f}$ is a $K \times 1$ response column vector.

$$\text{Let } \mathbf{\Lambda}^{1/2} = \text{diag}\left[\sqrt{\mu^{n-1}} \quad \sqrt{\mu^{n-2}} \quad \ldots \quad \sqrt{\mu} \quad 1\right] \tag{6.32}$$

where $\mathbf{\Lambda}^{1/2}$ is an $n \times n$ matrix that describes the exponential weighting to be applied to the data ($0 < \mu \leq 1$). By defining the $n \times N$ data matrix

$$\mathbf{X}(n) = \left[\mathbf{x}(1)\mathbf{x}(2)\ldots\mathbf{x}(n)\right]^\dagger \tag{6.33}$$

where $\mathbf{x}(i)$ is a data vector of length N at time iT, then the weighted output power is given by

$$\xi = \left\|\mathbf{\Lambda}^{1/2}\mathbf{X}(n)\mathbf{w}(n)\right\| \tag{6.34}$$

where $\| \|$ denotes the Euclidian norm, and $\mathbf{X}(n)$ is the data matrix of (6.33). The term "Q-R decomposition" is usually employed to describe the problem of obtaining an upper triangular transformation of an arbitrary matrix, $\mathbf{A}$, by applying an orthogonal matrix, $\mathbf{Q}$, obtained by a series of Givens rotations as described in Section 6.4.1.1. When this procedure is applied to the $n \times N$ matrix $\mathbf{\Lambda}^{1/2}\mathbf{X}(n)$, the result can be described by

$$\mathbf{Q}(n)\mathbf{\Lambda}^{1/2}\mathbf{X}(n) = \begin{bmatrix} \mathbf{R}(n) \\ 0 \end{bmatrix} \tag{6.35}$$

where $\mathbf{R}(n)$ is an $N \times N$ upper triangular matrix, and 0 is an $(n - N) \times N$ null matrix. In keeping with notation introduced earlier, we retain $\mathbf{U}(n)$ to denote the upper triangular matrix in (6.35), even though $\mathbf{R}(n)$—called the Cholesky factor—is what the terminology "Q-R decomposition" refers to. The weight vector that minimizes ξ of (6.34) is then given by

$$\mathbf{w}(n) = [\mathbf{U}^{\dagger}(n)\mathbf{U}(n)]^{-1}\mathbf{C}\{\mathbf{C}^{\dagger}[\mathbf{U}^{\dagger}(n)\mathbf{U}(n)]^{-1}\mathbf{C}\}^{-1}\mathbf{f} \tag{6.36}$$

The similarity of equation (6.36) to equation (4.169) is duly noted. The Q-R decomposition of the input data matrix using Givens rotations enables the weight vector to be obtained by using back substitution and can be implemented in parallel and systolic structures. Back substitution is a costly operation to perform in an algorithm and impedes a parallel implementation, so the inverse Q-R decomposition, which uses the inverse Cholesky factor, $\mathbf{U}^{-1}(n)$, is used instead.

To develop a recursive implementation of (6.36), a recursive update of the upper triangular matrix $\mathbf{U}(n)$ is implemented [11]

$$\begin{bmatrix} \mathbf{U}(n) \\ 0^T \end{bmatrix} = \mathbf{T}(n) \begin{bmatrix} \sqrt{\mu}\,\mathbf{U}(n-1) \\ \mathbf{x}^H(n) \end{bmatrix} \tag{6.37}$$

where $\mathbf{T}(n)$ is an $(N + 1) \times (N + 1)$ orthogonal matrix that annihilates the row vector $\mathbf{x}^{\dagger}(n)$ by rotating it into $\sqrt{\mu}\,\mathbf{U}(n - 1)$. By premultiplying both sides of (6.37) by their respective Hermitian forms and recognizing that $\mathbf{T}^{\dagger}(n)\mathbf{T}(n) = \mathbf{I}$, it follows that

$$\mathbf{U}^{\dagger}(n)\mathbf{U}(n) = \mu\mathbf{U}^{\dagger}(n-1)\mathbf{U}(n-1) + \mathbf{x}(n)\mathbf{x}^{\dagger}(n) \tag{6.38}$$

Using the matrix inversion lemma on (6.38) then yields

$$\mathbf{U}^{-1}(n)\mathbf{U}^{-\dagger}(n) = \frac{1}{\mu}\mathbf{U}^{-1}(n-1)\mathbf{U}^{-\dagger}(n-1) - \frac{\mathbf{U}^{-1}(n-1)\mathbf{z}(n)\mathbf{z}^H(n)\mathbf{U}^{-H}(n-1)}{\mu\, t^2(n)} \tag{6.39}$$

where

$$\mathbf{z}(n) = \frac{\mathbf{U}^{-1}(n-1)\mathbf{x}(n)}{\sqrt{\mu}} \tag{6.40}$$

and the scalar is defined by

$$t(n) = \sqrt{1 + \mathbf{z}^H(n)\mathbf{z}(n)} \tag{6.41}$$

Equation (6.39) can now be shown to be equivalent to

$$\mathbf{U}^{-1}(n)\mathbf{U}^{-\dagger}(n) = \frac{1}{\mu}\mathbf{U}^{-1}(n-1)\mathbf{U}^{-\dagger}(n-1) - \mathbf{g}(n)\mathbf{g}^{\dagger}(n) \tag{6.42}$$

where

$$\mathbf{g}(n) = \frac{\mathbf{U}^{-1}(n-1)\mathbf{z}(n)}{\sqrt{\mu}\, t(n)} \tag{6.43}$$

On comparing (6.42) and (6.38), it is apparent that there is a relationship corresponding to (6.37) using an orthogonal matrix $\mathbf{P}(n)$, which can be written as

$$\begin{bmatrix} \mathbf{U}^{-H}(n) \\ \mathbf{g}^{H}(n) \end{bmatrix} = \mathbf{P}(n) \begin{bmatrix} \mu^{-1/2}\mathbf{U}^{-H}(n-1) \\ 0^{T} \end{bmatrix} \tag{6.44}$$

Equation (6.44) may easily be verified by forming the product of each side of (6.44) with its respective complex conjugate transpose and verifying that (6.43) results. There is a major difference, however, between the orthogonal matrix $\mathbf{P}(n)$ in (6.44) and the orthogonal matrix $\mathbf{T}(n)$ in (6.37). The derivation of the orthogonal matrix $\mathbf{P}(n)$ is given in [13], and the result is

$$\mathbf{P}(n) \begin{bmatrix} \mathbf{z}(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ t(n) \end{bmatrix} \tag{6.45}$$

In other words, $\mathbf{P}(n)$ is a rotation matrix that successively annihilates the elements of the vector $\mathbf{z}(n)$, starting from the top, by rotating them into the last element at the bottom.

To relate these results to equation (6.36), it will be convenient to define a new $N \times N$ matrix $\mathbf{S}(n)$, given by

$$\mathbf{S}(n) = \mathbf{U}^{-1}(n)\mathbf{U}^{-\dagger}(n) \quad \text{or} \quad \mathbf{S}^{-1}(n) = \mathbf{U}^{\dagger}(n)\mathbf{U}(n) \tag{6.46}$$

Since it can easily be shown that $\mathbf{S}^{-1}(n) = \mathbf{X}^{\dagger}(n)\mathbf{\Lambda}(n)\mathbf{X}(n)$, $\mathbf{S}^{-1}(n)$ is referred to as a "correlation matrix" of the exponential weighted sensor outputs averaged over n samples. It is convenient to define $\mathbf{\Gamma}(n) = \mathbf{S}(n)\mathbf{C}$ and $\mathbf{\Phi}(n) = \mathbf{C}^{\dagger}\mathbf{\Gamma}(n) = \mathbf{C}^{\dagger}\mathbf{S}(n)\mathbf{C}$. Then we may rewrite (6.36) as

$$\mathbf{w}(n) = \mathbf{\Gamma}(n)\mathbf{\Phi}^{-1}(n)\,\mathbf{f} \tag{6.47}$$

Substituting $\mathbf{S}(n)$ of (6.46) into (6.39) and using the matrix inversion lemma, it follows that

$$\mathbf{S}(n) = \frac{1}{\mu}\mathbf{S}(n-1) - \frac{1}{\mu}\mathbf{k}(n)\mathbf{x}^{\dagger}(n)\mathbf{S}(n-1) \tag{6.48}$$

where

$$\mathbf{k}(n) = \frac{\frac{1}{\mu}\mathbf{S}(n-1)\mathbf{x}(n)}{1 + \frac{1}{\mu}\mathbf{x}^{H}(n)\mathbf{S}(n-1)\mathbf{x}(n)} = \mathbf{S}(n)\mathbf{x}(n). \tag{6.49}$$

Using the previously given definition for $\mathbf{\Gamma}(n)$ and right multiplying both sides of (6.48) by $\mathbf{C}$ yields

$$\mathbf{\Gamma}(n) = \frac{1}{\mu}\mathbf{\Gamma}(n-1) - \frac{1}{\mu}\mathbf{k}(n)\mathbf{x}^{\dagger}(n)\mathbf{\Gamma}(n-1) \tag{6.50}$$

Likewise, premultiplying both sides of $\mathbf{\Gamma}(n) = \mathbf{S}(n)\mathbf{C}$ by $\mathbf{C}^\dagger$ and using the previously given definition for $\mathbf{\Phi}(n)$ gives the result

$$\mathbf{\Phi}(n) = \frac{1}{\mu}\mathbf{\Phi}(n-1) - \frac{1}{\mu}\mathbf{k}(n)\mathbf{x}^\dagger(n)\mathbf{\Phi}(n-1) \tag{6.51}$$

Postmultiplying both sides of (6.43) by $\mathbf{C}$ then yields

$$\mathbf{\Gamma}(n) = \mu^{-1}\mathbf{\Gamma}(n-1) - \mathbf{g}(n)\boldsymbol{\alpha}(n) \tag{6.52}$$

where

$$\boldsymbol{\alpha}(n) = \mathbf{g}^\dagger(n)\mathbf{C} \tag{6.53}$$

A similar development can be given for $\mathbf{\Phi}(n)$ starting with

$$\begin{aligned}\mathbf{\Phi}(n) &= \mathbf{C}^\dagger\mathbf{U}^{-1}(n)\mathbf{U}^{-\dagger}(n)\mathbf{C} = \mathbf{C}^\dagger[\mu^{-1}\mathbf{U}^{-1}(n-1)\mathbf{U}^{-\dagger}(n-1) - \mathbf{g}(n)\mathbf{g}^\dagger(n)]\mathbf{C} \\ &= \mu^{-1}\mathbf{\Phi}(n-1) - \boldsymbol{\alpha}^\dagger(n)\boldsymbol{\alpha}(n)\end{aligned} \tag{6.54}$$

Setting $\gamma(n) = \sqrt{\mu}\,\boldsymbol{\alpha}^\dagger(n)$, (6.54) can be rewritten as

$$\mathbf{\Phi}(n) = \mu^{-1}[\mathbf{\Phi}(n-1) - \gamma(n)\gamma^\dagger(n)] \tag{6.55}$$

Applying the matrix inversion lemma to (6.55) then yields the recursive relation

$$\mathbf{\Phi}^{-1}(n) = \mu[\mathbf{I} + \sqrt{\mu}\,\mathbf{q}(n)\boldsymbol{\alpha}(n)]\mathbf{\Phi}^{-1}(n-1) \tag{6.56}$$

where

$$\mathbf{q}(n) = \frac{\sqrt{\mu}\mathbf{\Phi}^{-1}(n-1)\boldsymbol{\alpha}^\dagger(n)}{1 - \mu\boldsymbol{\alpha}(n)\mathbf{\Phi}^{-1}(n-1)\boldsymbol{\alpha}^\dagger(n)} = \mu^{-1/2}[\mathbf{\Phi}^{-1}(n)\boldsymbol{\alpha}^\dagger(n)] \tag{6.57}$$

Finally, to apply the previous recursive relationships to the weight vector of (6.47), we have

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu[\mathbf{g}(n) - \sqrt{\mu}\mathbf{\Gamma}(n)\mathbf{q}(n)]\boldsymbol{\alpha}(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{f} \tag{6.58}$$

The recursive relationship of (6.58) can be further simplified by applying the definitions of (6.40) and (6.42) to obtain

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \rho(n)\xi(n, n-1) \tag{6.59}$$

where

$$\rho(n) = \mathbf{k}(n) - \frac{\sqrt{\mu}}{t(n)}\mathbf{\Gamma}(n)\mathbf{q}(n) \tag{6.60}$$

$$\xi(n, n-1) = \mathbf{x}^\dagger(n)\mathbf{w}(n-1) \tag{6.61}$$

A summary of the previous development for a recursive beamforming algorithm is given in Table 6-1.

TABLE 6-1 ■ Recursive Algorithm Based on Inverse QR Updating

- Initialization: $\mathbf{U}^{-1}(0) = \delta^{-1}\mathbf{I}$, δ = small positive constant

$$\mathbf{\Gamma}(0) = \mathbf{U}^{-1}(0)\mathbf{U}^{-\dagger}(0)\mathbf{C}$$
$$\mathbf{w}(0) = \mathbf{\Gamma}(0)[\mathbf{C}^{\dagger}\mathbf{\Gamma}(0)]^{-1}\,\mathbf{f}$$

- For $n = 1, 2, \ldots,$ do the following:
  - Compute the intermediate vector $\mathbf{z}(n)$

$$\mathbf{z}(n) = \frac{\mathbf{U}^{-\dagger}(n-1)\mathbf{x}(n)}{\sqrt{\mu}}$$

  - Evaluate the rotations that define $\mathbf{P}(n)$, which annihilate $\mathbf{z}(n)$ and compute the scalar $t(n)$ from

$$\mathbf{P}(n)\begin{bmatrix}\mathbf{z}(n)\\ 1\end{bmatrix} = \begin{bmatrix}0\\ t(n)\end{bmatrix}$$

  - Update the lower triangular matrix $\mathbf{U}^{-\dagger}(n)$, and compute the vector $\mathbf{g}(n)$ and $\alpha(n) = \mathbf{g}^{\dagger}(n)\,\mathbf{C}$ from

$$\mathbf{P}(n)\begin{bmatrix}\mu^{-1/2}\mathbf{U}^{-\dagger}(n-1)\\ 0^{T}\end{bmatrix} = \begin{bmatrix}\mathbf{U}^{-\dagger}(n)\\ \mathbf{g}^{\dagger}(n)\end{bmatrix}$$

  - Update $\mathbf{\Gamma}(n)$, $\mathbf{q}(n)$, and $\mathbf{\Phi}^{-1}(n)$ using

$$\mathbf{\Gamma}(n) = \mu^{-1}\mathbf{\Gamma}(n-1) - \mathbf{g}(n)\boldsymbol{\alpha}(n)$$
$$\mathbf{q}(n) = \frac{\sqrt{\mu}\mathbf{\Phi}^{-1}(n-1)\boldsymbol{\alpha}^{\dagger}(n)}{1 - \mu\boldsymbol{\alpha}(n)\mathbf{\Phi}^{-1}(n-1)\boldsymbol{\alpha}^{\dagger}(n)}$$
$$\mathbf{\Phi}^{-1}(n) = \mu[\mathbf{I} + \sqrt{\mu}\,\mathbf{q}(n)\boldsymbol{\alpha}(n)]\mathbf{\Phi}^{-1}(n-1)$$

  - Update the weight vector

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \rho(n)\xi(n, n-1)$$

where

$$\rho(n) = \mathbf{k}(n) - \frac{\sqrt{\mu}}{t(n)}\mathbf{\Gamma}(n)\mathbf{q}(n)$$

and $\xi(n, n-1) = \mathbf{x}^{\dagger}(n)\mathbf{w}(n-1)$

6.3 | KALMAN FILTER METHODS FOR ADAPTIVE ARRAY PROCESSING

The adaptive algorithms for small communications and data collection arrays are often based on relatively simple gradient and random search methods. More complex problems, such as command and control of remote vehicles or rapid angular tracking in radar systems, require sophisticated processing, such as Kalman filtering [5,14,15]. These methods include more a priori signal environment data than other methods and open the possibility of constructing processors that integrate adaptive array control with other system functions such as position location and navigation.

6.3.1 Development of a Kalman-Type Array Processor

Consider the simple N-element linear narrowband array model of Figure 6-1 for which the (sampled) signal vector is $\mathbf{x}(k)$, the adaptive processor weight vector (at corresponding sample times) is $\mathbf{w}(k)$, and the array output is $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$. To accommodate wider bandwidth signals, each channel of the array has a tapped delay line filter, but the tap weights and delayed signals are represented as vectors $\mathbf{w}(k)$ and $\mathbf{x}(k)$ so the mathematical development for this case remains the same as for the narrowband case.

Let the dynamic behavior of the optimal array weights be represented by

$$\mathbf{w}_{\text{opt}}(k+1) = \mathbf{\Phi}(k+1, k)\mathbf{w}_{\text{opt}}(k), \mathbf{w}_{\text{opt}}(0) = \mathbf{w}_0 \tag{6.62}$$

where $\mathbf{\Phi}(k+1, k)$ is a transition matrix. If the array signal environment is stationary, then the optimal weights are fixed and $\mathbf{\Phi}(k+1, k)$ is the identity matrix. For time-varying environments, a more complex model for $\mathbf{\Phi}(k+1, k)$ must be developed that reflects how the optimal array weights change in response to the varying environment. Let the system measurements be represented by a noise corrupted version of the optimal (not the actual) array output:

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{\text{opt}}(k) + v(k) \tag{6.63}$$

where $v(k)$ is a member of a white, Gaussian noise sequence having zero mean and variance given by

$$E\{v(k)v(j)\} = \sigma^2(k)\delta_{kj} \tag{6.64}$$

The selection of a value for $\sigma^2(k)$ is discussed later.

For the dynamical model (6.62) having state vector $\mathbf{w}_{\text{opt}}(k)$ and measurement from (6.63), Kalman filter theory leads to a minimum mean square error (MMSE) estimator for the array weights (rather than for the desired signal). It immediately follows that the optimum filtered estimate of the optimal array weight vector $\hat{\mathbf{w}}_{\text{opt}}(k/k)$ is given by the relationship [12]

$$\hat{\mathbf{w}}_{\text{opt}}(k/k) = \hat{\mathbf{w}}_{\text{opt}}(k/k-1) + \mathbf{K}(k)\left[d(k) - \mathbf{x}^T(k)\hat{\mathbf{w}}_{\text{opt}}(k/k-1)\right] \tag{6.65}$$

where (k/k) denotes a filtered quantity at sample time k based on measurements through (and including) k, $(k/k-1)$ denotes a predicted quantity at sample time k based on measurements through $k-1$, and $\mathbf{K}(k)$ is the Kalman gain matrix. For complex quantities (6.65) is rewritten as

$$\begin{aligned}\hat{\mathbf{w}}_{\text{opt}}(k/k) = {} & \hat{\mathbf{w}}_{\text{opt}}(k/k-1) \\ & + \mathbf{K}(k)\left[d^*(k) - \mathbf{x}^\dagger(k)\hat{\mathbf{w}}_{\text{opt}}(k/k-1)\right]\end{aligned} \tag{6.66}$$

Now

$$\hat{\mathbf{w}}_{\text{opt}}(k/k-1) = \mathbf{\Phi}(k, k-1)\hat{\mathbf{w}}_{\text{opt}}(k-1/k-1) \tag{6.67}$$

and the quantity in brackets of (6.66) is the difference between the optimal array output (or desired reference signal) and the actual array output. The Kalman-type processor based on the foregoing equations is shown in Figure 6-5 where the Kalman gain vector is given by

$$\mathbf{K}(k) = \frac{\mathbf{P}(k/k-1)\mathbf{x}(k)}{[\mathbf{x}^T(k)\mathbf{P}(k/k-1)\mathbf{x}(k) + \sigma^2(k)]} \tag{6.68}$$

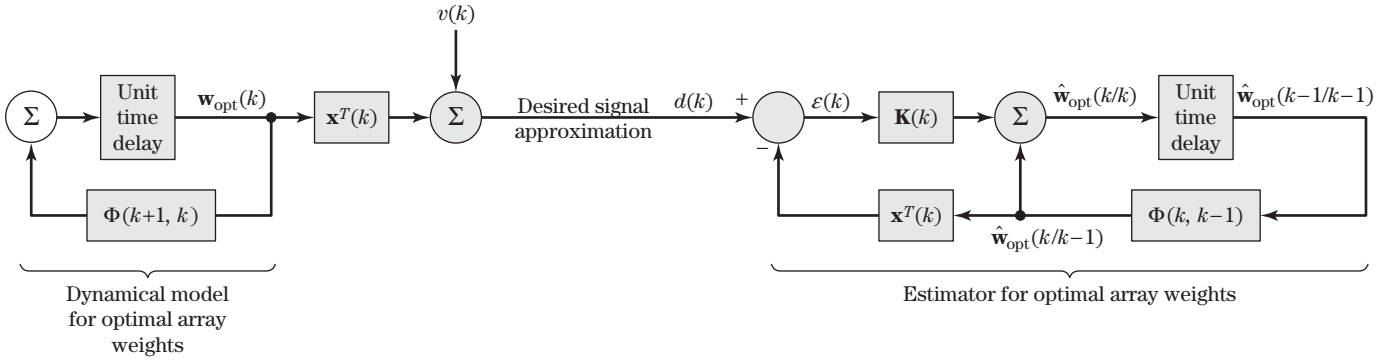


FIGURE 6-5 ■ Kalman-type processor for optimal array weights.

or for complex quantities

$$\mathbf{K}(k) = \frac{\mathbf{P}(k/k-1)\mathbf{x}^*(k)}{[\mathbf{x}^T(k)\mathbf{P}(k/k-1)\mathbf{x}^*(k) + \sigma^2(k)]} \tag{6.69}$$

The predicted error covariance matrix is given by

$$\mathbf{P}(k/k-1) = \mathbf{\Phi}(k, k-1)\mathbf{P}(k-1/k-1)\mathbf{\Phi}^T(k, k-1) \tag{6.70}$$

The filtered error covariance matrix is defined by

$$\mathbf{P}(k/k) \triangleq E\left\{\left[\mathbf{w}_{\text{opt}}(k) - \hat{\mathbf{w}}_{\text{opt}}(k/k)\right]\left[\mathbf{w}_{\text{opt}}(k) - \hat{\mathbf{w}}_{\text{opt}}(k/k)\right]^T\right\} \tag{6.71}$$

Equation (6.71) can be expressed in the equivalent form

$$\mathbf{P}(k/k) = \mathbf{P}(k/k-1) - \mathbf{K}(k)\mathbf{x}^T(k)\mathbf{P}(k/k-1) \tag{6.72}$$

On substituting (6.68) into (6.72) there results

$$\mathbf{P}(k/k) = \mathbf{P}(k/k-1) - \frac{\mathbf{P}(k/k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{P}(k/k-1)}{[\sigma^2(k) + \mathbf{x}^T(k)\mathbf{P}(k/k-1)\mathbf{x}(k)]} \tag{6.73}$$

Comparing (6.73) with (6.14) reveals significant similarity between the recursive equations for $\mathbf{P}(k+1)$ of (6.14) and $\mathbf{P}(k/k)$ of (6.73). Equation (6.73) is rewritten by application of the matrix inversion lemma [(D.14) of Appendix D] as

$$\mathbf{P}^{-1}(k/k) = \mathbf{P}^{-1}(k/k-1) + \frac{1}{\sigma^2(k)}\mathbf{x}(k)\mathbf{x}^T(k) \tag{6.74}$$

Comparing (6.74) with (6.13) shows that the Kalman error covariance matrix with $\sigma^2(k) = 1$ corresponds to $\mathbf{P}(k)$ for the weighted least square error processor with $\alpha = 1$. In this case, (6.21) leads to the error covariance matrix $\mathbf{P}(k/k)$ being related to the finite-time average estimate for the received signal autocorrelation matrix $\hat{\mathbf{R}}_{xx}(k)$ by the relationship

$$\hat{\mathbf{R}}_{xx}^{-1}(k) = k\mathbf{P}(k/k) \tag{6.75}$$

To begin the recursive equations (6.65)–(6.73), the initial values for $\hat{\mathbf{w}}(0/0)$ and $\mathbf{P}(0/0)$ must be specified. It is desirable to select $\hat{\mathbf{w}}_{\text{opt}}(0/0) = E\{\mathbf{w}_{\text{opt}}\}$ and $\mathbf{P}(0/0) = E\{\Delta\mathbf{w}(0)\Delta\mathbf{w}^T(0)\}$ [5] where $\Delta\mathbf{w}(0) = \hat{\mathbf{w}}(0/0) - \mathbf{w}_{\text{opt}}$. From Chapter 3, $\mathbf{w}_{\text{opt}}$ is given by the Wiener solution.

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}(k)\mathbf{r}_{xd}(k) \tag{6.76}$$

In general, the signal statistics represented by the solution (6.76) are unknown, so a different procedure (discussed subsequently) is employed to initialize the recursive equations. In the event that a priori environment data are available, then such information is used to form a refined initial estimate of $\mathbf{w}_{\text{opt}}$ using (6.76) as well as to construct a dynamical system model by way of (6.62).

In situations where a priori information concerning the signal environment is not available, then $\hat{\mathbf{w}}_{\text{opt}}(0)$ is generally chosen to yield an omnidirectional array beam pattern, and $\mathbf{P}(0/0)$ can merely be set equal to the identity matrix. Furthermore, some means of selecting a value for the noise statistic $\sigma^2(k)$ must be given.

As can be seen from (6.65), the bracketed quantity represents the difference between the actual array output (using the estimated weights) and a noise corrupted version of the optimal array output derived from an identical array employing the optimal weight vector on the received signal set. This optimal output signal $d(t)$ is interpreted as a reference signal approximation for the actual desired signal since the optimal array weights are designed to provide an MMSE estimate of the desired signal. Such an interpretation of the optimal array output signal in turn suggests a procedure for selecting a value for the noise statistic $\sigma^2(k)$. Since $d(t)$ is an approximation for the actual desired signal, $s(t)$, one may write [5]

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{\text{opt}}(k) + v(k) = s(k) + \eta(k) \tag{6.77}$$

where $\eta(k)$ indicates the error in this approximation for the desired signal $s(k)$. Consequently,

$$v(k) = s(k) + \eta(k) - \mathbf{x}^T(k)\mathbf{w}_{\text{opt}}(k) \tag{6.78}$$

If $s(k)$, $\eta(k)$, and $\mathbf{x}(k)$ are all zero mean processes, then $E\{v(k)\} = 0$. Furthermore, if the noise sequence $\eta(k)$ is not correlated with $s(k)$ and the noise components of $\mathbf{x}(k)$, then

$$\begin{aligned}\sigma^2(k) &= E\{v(k)v(k)\} = \mathbf{w}_{\text{opt}}^T\mathbf{R}_{xx}\mathbf{w}_{\text{opt}} - \mathbf{2}\mathbf{w}_{\text{opt}}^T\mathbf{r}_{xs} + E\{s^2\} + E\{\eta^2\} \\ &= \text{MMSE} + E\{\eta^2\}\end{aligned} \tag{6.79}$$

It follows that if the "measurement" $d(k)$ is generated by using a realistic approximation for the desired signal (possibly obtained from a modem connected to the array output), then the value of $\sigma^2(k)$ is chosen based on the MMSE that can be achieved by an optimal array and the quality (as measured by the error variance) of the desired signal approximation.

If we rewrite (6.65) as

$$\hat{\mathbf{w}}(k/k) = \hat{\mathbf{w}}(k/k-1) + \frac{\mathbf{P}(k/k-1)}{[\mathbf{x}^T(k)\mathbf{P}(k/k-1)\mathbf{x}(k) + \sigma^2(k)]}[\mathbf{x}(k)d(k) - \mathbf{x}(k)y(k)] \tag{6.80}$$

where $y(k) = \hat{\mathbf{w}}^T(k/k-1)\mathbf{x}(k)$, it is apparent that the quantity $\mathbf{x}(k)d(k)$ can be replaced by its average value, in exactly the same manner as with the weighted least squares recursive algorithm of the preceding section. Consequently, the Kalman filter weight update equation is also modified to accommodate two additional forms involving the use of either $\hat{\mathbf{r}}_{xd}(k)$ (when either the desired signal or a reference signal model is available) or $\mathbf{r}_{xd}(k)$ (when only direction of arrival information is available).

Yet another alternative for handling a nonstationary environment with a Kalman-type processor exists when the time-varying nature of the signal statistics are unknown. Rather than attempt to construct an accurate dynamical model representation by way of (6.62), simply use

$$\mathbf{w}_{\text{opt}}(k+1) = \mathbf{w}_{\text{opt}}(k) + \boldsymbol{\xi}(k) \tag{6.81}$$

where $\boldsymbol{\xi}(k)$ is a zero-mean white noise process with

$$\text{cov}[\boldsymbol{\xi}(k)] = \mathbf{Q}(k) \tag{6.82}$$

The elements of $\mathbf{Q}$ represent the degree of uncertainty associated with adopting the stationary environment assumption represented by using the identity state transition matrix

in (6.62). Equation (6.70) then becomes

$$\mathbf{P}(k/k-1) = \mathbf{\Phi}(k, k-1)\mathbf{P}(k-1/k-1)\mathbf{\Phi}^T(k, k-1) + \mathbf{Q} \tag{6.83}$$

The practical effect of the previous modification is to prevent the Kalman gains in $\mathbf{K}(k)$ from decaying to values that are too small, so when variations in the environment occur sufficient importance is attached to the most recent measurements. The estimate $\hat{\mathbf{w}}_{\text{opt}}$ then "follows" variations in the actual value of $\mathbf{w}_{\text{opt}}$, although the resulting optimal weight vector estimates are more "noisy" than when the $\mathbf{Q}$ matrix was absent.

6.3.2 Speed of Convergence

The fact that the value of $\sigma^2(k)$ is related to the MMSE and to the quality of the desired signal approximation makes it possible to obtain an expression from which the rate of convergence for the algorithm can be deduced under certain conditions [16]. Equation (6.68) can be rewritten as

$$\mathbf{K}(k)\mathbf{x}^T(k)\mathbf{P}(k/k-1)\mathbf{x}(k) + \mathbf{K}(k)\sigma^2(k) = \mathbf{P}(k/k-1)\mathbf{x}(k) \tag{6.84}$$

or

$$\mathbf{K}(k)\sigma^2(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{x}^T(k)]\mathbf{P}(k/k-1)\mathbf{x}(k) \tag{6.85}$$

Equation (6.85) may now be substituted into (6.72) to obtain

$$\mathbf{P}(k/k)\mathbf{x}(k) = \mathbf{K}(k)\sigma^2(k) \tag{6.86}$$

so that

$$\mathbf{K}(k) = \frac{\mathbf{P}(k/k)\mathbf{x}(k)}{\sigma^2(k)} \tag{6.87}$$

When we substitute the result expressed by (6.87) into (6.72) there results

$$\mathbf{P}(k/k) = \mathbf{P}(k/k-1) - \frac{\mathbf{P}(k/k)\mathbf{x}(k)}{\sigma^2(k)}\mathbf{x}^T(k)\mathbf{P}(k/k-1) \tag{6.88}$$

or

$$\mathbf{P}(k/k) = \mathbf{P}(k/k-1) - \frac{[\mathbf{P}(k/k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{P}(k/k-1)]}{\sigma^2(k)} \tag{6.89}$$

Premultiplying both sides of (6.89) by $\mathbf{P}^{-1}(k/k)$ and postmultiplying both sides of (6.90) by $\mathbf{P}^{-1}(k/k-1)$, we see it follows that [also see (6.74)]

$$\mathbf{P}^{-1}(k/k) = \mathbf{P}^{-1}(k/k-1) + \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\sigma^2(k)} \tag{6.90}$$

Equation (6.90) can be rewritten as

$$\mathbf{P}^{-1}(k/k) = \frac{1}{\sigma^2(k)}[\sigma^2(k)\mathbf{P}^{-1}(k/k-1) + \mathbf{x}(k)\mathbf{x}^T(k)] \tag{6.91}$$

so that

$$\mathbf{P}(k/k) = \sigma^2(k)[\sigma^2(k)\mathbf{P}^{-1}(k/k-1) + \mathbf{x}(k)\mathbf{x}^T(k)]^{-1} \tag{6.92}$$

The recursive relationship expressed by (6.92) can be repeatedly applied beginning with $\mathbf{P}^{-1}(0/-1)$ to obtain

$$\mathbf{P}(k/k) = \sigma^2(k)\left[\sigma^2(k)\mathbf{P}^{-1}(0/-1) + \sum_{i=1}^{k}\mathbf{x}(i)\mathbf{x}^T(i)\right]^{-1} \tag{6.93}$$

For cases where the desired signal approximation is quite good $\hat{\sigma}^2(k) \approx$ MMSE, and the diagonal matrix $\hat{\sigma}^2(k)\mathbf{P}^{-1}(0/-1)$ can be neglected in comparison with $\sum_{i=1}^{k}\mathbf{x}(i)\mathbf{x}^T(i)$ so that

$$\mathbf{P}(k/k) \cong \hat{\sigma}^2(k)\left[\sum_{i=1}^{k}\mathbf{x}(i)\mathbf{x}^T(i)\right]^{-1} \tag{6.94}$$

When we use the result (6.87), it follows immediately that

$$\mathbf{K}(k) \cong \left[\sum_{i=1}^{k}\mathbf{x}(i)\mathbf{x}^T(i)\right]^{-1}\mathbf{x}(k) \tag{6.95}$$

which is independent of $\sigma^2(k)$. The arithmetic average is given by

$$\frac{1}{k}\sum_{i=1}^{k}\mathbf{x}(i)\mathbf{x}^T(i) \to \mathbf{R}_{xx}(k) \quad \text{as} \quad k \to \infty \tag{6.96}$$

The MSE at the kth sampling instant, $\xi^2(k)$, can be written as [16]

$$\xi^2(k) = \text{trace}[\mathbf{P}(k/k)\mathbf{R}_{xx}(k)] + \text{MMSE} \tag{6.97}$$

From (6.94) and (6.96) it follows that

$$\text{trace}[\mathbf{P}(k/k)\mathbf{R}_{xx}(k)] \cong \sigma^2(k)Nk^{-1} \tag{6.98}$$

where N is the dimension of $\mathbf{x}(k)$ so the MSE at the kth sampling instant becomes

$$\xi^2(k) \cong \text{MMSE}[1 + Nk^{-1}] \tag{6.99}$$

The result in (6.99) means that convergence for this Kalman-type algorithm is theoretically obtained within $2N$ iterations, which is similar to the convergence results for the DMI algorithm.

6.4 THE MINIMUM VARIANCE PROCESSOR

It is useful at this point to consider the application of the concepts used in obtaining a recursive algorithm to a broadband signal aligned array processor to illustrate the slight modifications involved in handling a multichannel tapped delay line processor. If the direction of arrival of the desired signal is known, this information may be used to construct a signal-aligned array as shown in Figure 6-6. The signal-aligned array processor uses spatial correction filters (SCF) and knowledge of the desired signal's direction of arrival to align the desired signal components in each channel of the array by properly selecting the time delays τ_i, that is [17]

$$\mathbf{x}(t) = \begin{bmatrix} z_1(t-\tau_1) \\ z_2(t-\tau_2) \\ \vdots \\ z_N(t-\tau_N) \end{bmatrix} = d(t)\mathbf{1} + \mathbf{n}(t) \tag{6.100}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^T$, $d(t)$ is the desired reference signal, and $\mathbf{n}(t)$ is the vector of the interference terms after the time delays. Collecting the steered received signal vector $\mathbf{x}(t)$ and its delayed components along the tapped delay line into a single $(M+1)N \times 1$

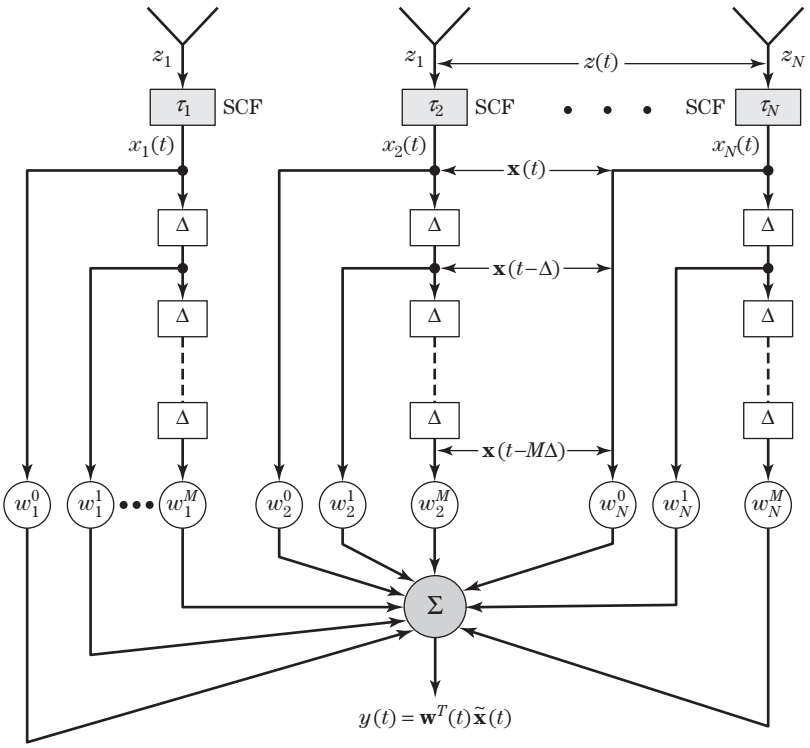


FIGURE 6-6 ■ Broadband signal aligned array processor.

dimensional vector gives

$$\mathbf{x}'(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t-\Delta) \\ \vdots \\ \mathbf{x}(t-M\Delta) \end{bmatrix} = \begin{bmatrix} d(t)\mathbf{1} \\ d(t-\Delta)\mathbf{1} \\ \vdots \\ d(t-M\Delta)\mathbf{1} \end{bmatrix} + \mathbf{n}'(t) \tag{6.101}$$

In what follows it will be convenient to drop the prime notation and simply remember that all signals under consideration are the collection of terms in the tapped delay line filter. Furthermore, collecting the N-dimensional weight vectors $\mathbf{w}^0(t), w^1(t), \ldots, \mathbf{w}^M(t)$, as

$$\mathbf{w}(t) = \begin{bmatrix} \mathbf{w}^0(t) \\ \mathbf{w}^1(t) \\ \vdots \\ \mathbf{w}^M(t) \end{bmatrix} \tag{6.102}$$

the array output can be written in the form

$$\begin{aligned} y(t) = \mathbf{w}^T\mathbf{x}(t) &= d(t)\sum_{i=1}^{N} w_i^0 \qquad (6.103) \\ &= d(t)\sum_{i=1}^{N} w_i^0 + d(t-\Delta)\sum_{i=1}^{N} w_i^1 + L\cdots + d(t-M\Delta)\sum_{i=1}^{N} w_i^M + \mathbf{w}^T\mathbf{n}(t) \end{aligned}$$

since $\mathbf{w}^{l^T}\mathbf{1} = \sum_{i=1}^{N} w_i^l$. If the adaptive weights are constrained according to

$$\sum_{i=1}^{N} w_i^l = \begin{cases} 1, & l = 0 \\ 0, & l = 1, 2, \cdots, k \end{cases} \tag{6.104}$$

then the output signal can be written as

$$y(t) = d(t) + \mathbf{w}^T\mathbf{n}(t) \tag{6.105}$$

The output signal of (6.105) is unbiased since

$$E\{y(t)\} = E\{d(t) + \mathbf{w}^T\mathbf{n}(t)\} = d(t) \tag{6.106}$$

if we assume the noise vector has zero mean. The variance of the output signal is then given by

$$\text{var}\,[y(t)] = E\{\mathbf{w}^T\mathbf{n}(t)\mathbf{n}^T(t)\mathbf{w}\} = \mathbf{w}^T\mathbf{R}_{nn}\mathbf{w} \tag{6.107}$$

Define the $N(M+1) \times (M+1)$ matrix,

$$\mathbf{I}_1 \triangleq \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix} \tag{6.108}$$

where $\mathbf{0}$ is an $N \times 1$ vector with all zero components. If it is desired to minimize the output noise variance, then the noise variance performance measure can be defined by

$$\mathfrak{P}_{\text{mv}} = \mathbf{w}^T \mathbf{R}_{nn} \mathbf{w} \tag{6.109}$$

It is now desired to minimize (6.109) subject to the constraint (6.104), which can be rewritten as

$$\mathbf{I}_1^T \mathbf{w} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{c} \tag{6.110}$$

The weight vector $\mathbf{w}$ that minimizes (6.109) subject to (6.110) can be chosen by using a vector Lagrange multiplier to form the modified performance measure

$$\mathfrak{P}_{\text{mvm}} = \frac{1}{2} \mathbf{w}^T \mathbf{R}_{nn} \mathbf{w} + \boldsymbol{\lambda} \left[\mathbf{c} - \mathbf{I}_1^T \mathbf{w} \right] \tag{6.111}$$

Setting the derivative of $\mathfrak{P}_{\text{mvm}}$ with respect to $\mathbf{w}$ equal to zero to obtain $\mathbf{w}_{\text{mv}}$, requiring $\mathbf{w}_{\text{mv}}$ to satisfy (6.110) to evaluate $\boldsymbol{\lambda}$, and substituting the resulting value of $\boldsymbol{\lambda}$ into $\mathbf{w}_{\text{mv}}$ gives the minimum variance weight vector solution

$$\mathbf{w}_{\text{mv}} = \mathbf{R}_{nn}^{-1} \mathbf{I}_1 \left[\mathbf{I}_1^T \mathbf{R}_{nn}^{-1} \mathbf{I}_1 \right]^{-1} \mathbf{c} \tag{6.112}$$

For a signal-aligned array like that of Figure 6-6, it can also be established (as was done in Chapter 3 for a narrowband processor) that the minimum variance estimator resulting from the use of (6.112) is identical to the maximum likelihood estimator [18].

The weight vector computation defined by (6.112) requires the measurement and inversion of the noise autocorrelation matrix $\mathbf{R}_{nn}$. The noise autocorrelation matrix can be obtained from the data that also include desired signal terms, and use of a recursive algorithm will circumvent the necessity of directly inverting $\mathbf{R}_{nn}$.

The difficulty of measuring the noise autocorrelation matrix when desired signal terms are present can be avoided by reformulating the optimization problem posed by (6.109) and (6.110). The minimization of (6.109) subject to the constraint (6.110) is completely equivalent to the following problem:

$$\text{Minimize} \mathfrak{P} = \mathbf{w}^T \mathbf{R}_{xx} \mathbf{w} \tag{6.113}$$

subject to the constraint

$$\mathbf{I}_1 \mathbf{w} = \mathbf{c} \tag{6.114}$$

The solution of (6.113), (6.114) may be found by once again using Lagrange multipliers with the result that

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1} \mathbf{I}_1 [\mathbf{I}_1^T \mathbf{R}_{xx}^{-1} \mathbf{I}_1]^{-1} \mathbf{c} \tag{6.115}$$

To show the equivalence between the problem (6.113) and (6.114) and the original problem (6.109) and (6.110), expand the matrix $\mathbf{R}_{xx}$ as follows:

$$\mathbf{R}_{xx} = E\{\mathbf{x}(t)\mathbf{x}^T(t)\} = \mathbf{R}_{dd} + \mathbf{R}_{nn} \tag{6.116}$$

where

$$\mathbf{R}_{dd} = E\left\{ \begin{bmatrix} d(t)\mathbf{1} \\ d(t-\Delta)\mathbf{1} \\ \vdots \\ d(t-M\Delta)\mathbf{1} \end{bmatrix} [d(t)\mathbf{1}d(t-\Delta)\mathbf{1}, \ldots, d(t-M\Delta)\mathbf{1}] \right\} \tag{6.117}$$

Substituting (6.116) into (6.113), the problem now becomes

$$\text{Minimize } \mathfrak{P} = \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} = \mathbf{w}^T\mathbf{R}_{nn}\mathbf{w} + \mathbf{w}^T\mathbf{R}_{dd}\mathbf{w} \tag{6.118}$$

With the constraint (6.104), however, it follows that

$$\mathfrak{P} = \mathbf{w}^T\mathbf{R}_{nn}\mathbf{w} + E\{d^2(t)\} \tag{6.119}$$

Now the minimization of (6.119) subject to (6.114) must give exactly the same solution as the optimization problem of (6.109) and (6.110), since $E\{d^2(t)\}$ is not a function of $\mathbf{w}$, and the two problems are therefore completely equivalent.

The received signal correlation matrix at the kth sample time $\mathbf{R}_{xx}(k)$ can be measured using the exponentially deweighted finite time average

$$\hat{\mathbf{R}}_{xx}(k) = \frac{1}{\left(\sum_{n=1}^{k} \alpha^{k-n}\right)} \sum_{n=1}^{k} \alpha^{k-n}\mathbf{x}(n)\mathbf{x}^T(n) \tag{6.120}$$

where $0 < \alpha \leq 1$. By defining the matrices

$$\mathbf{X}(k) \triangleq \begin{bmatrix} \mathbf{x}^T(1) \\ \mathbf{x}^T(2) \\ \vdots \\ \mathbf{x}^T(k) \end{bmatrix} \tag{6.121}$$

and

$$\mathbf{A}(k) \triangleq \begin{bmatrix} \alpha^{k-1} & 0 & \cdots & 0 \\ 0 & \alpha^{k-2} & & \vdots \\ \vdots & & \alpha & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{6.122}$$

then $\hat{\mathbf{R}}_{xx}(k)$ can be rewritten as

$$\hat{\mathbf{R}}_{xx}(k) = \frac{1}{\text{tr}[\mathbf{A}(k)]}\mathbf{X}^T(k)\mathbf{A}(k)\mathbf{X}(k) = \frac{1}{\text{tr}[\mathbf{A}(k)]}\mathbf{P}^{-1}(k) \tag{6.123}$$

where $\mathbf{P}^{-1}(k) \triangleq \mathbf{X}^T(k)\mathbf{A}(k)\mathbf{X}(k)$. Partitioning $\mathbf{x}(k+1)$ and $\mathbf{A}(k+1)$ as in (6.7) and (6.9) and applying the matrix inversion lemma immediately leads to (6.14). The desired updated weights may then be written in terms of $\mathbf{P}(k+1)$ from (6.115) as

$$\mathbf{w}_{\text{opt}}(k+1) = \mathbf{P}(k+1)\mathbf{I}_1\left[\mathbf{I}_1^T\mathbf{P}(k+1)\mathbf{I}_1\right]^{-1}\mathbf{c} \tag{6.124}$$

The inverse of $\mathbf{I}_1^T\mathbf{P}(k+1)\mathbf{I}_1$ required in (6.124) can be efficiently computed by application of the matrix inversion lemma to yield

$$\mathbf{w}_{\text{opt}}(k+1) = \left\{\mathbf{I} - \left[\frac{\mathbf{P}(k)}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)} - \mathbf{P}(k+1)\right] \cdot \mathbf{x}(k+1)\mathbf{x}^T(k+1)\right\}\mathbf{I}_1^{-1}\mathbf{c} \tag{6.125}$$

The minimum variance recursive processor for the narrowband case takes a particularly simple form. It was found in Chapter 3 for this case that

$$\mathbf{w}_{\text{mv}} = \frac{\mathbf{R}_{nn}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{nn}^{-1}\mathbf{1}} \tag{6.126}$$

The use of (6.126) presents an additional difficulty since the received signal vector $\mathbf{x}(t)$ generally contains signal as well as noise components. This difficulty can be circumvented, though.

The input signal covariance matrix is given by

$$\mathbf{R}_{xx} \triangleq E\{\mathbf{x}^*(t)\mathbf{x}^T(t)\} = E\{d^2(t)\}\mathbf{1}\mathbf{1}^T + \mathbf{R}_{nn} \tag{6.127}$$

where $E\{d^2(t)\} = \beta$, a scalar quantity. Inverting both sides of (6.127) and applying the matrix inversion lemma yields

$$\mathbf{R}_{xx}^{-1} = [\beta\mathbf{1}\mathbf{1}^T + \mathbf{R}_{nn}]^{-1} = \mathbf{R}_{nn}^{-1} - \frac{\beta\mathbf{R}_{nn}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{R}_{nn}^{-1}}{1 + \beta\mathbf{1}^T\mathbf{R}_{nn}^{-1}\mathbf{1}} \tag{6.128}$$

Substituting (6.128) into the ratio

$$\frac{\mathbf{R}_{xx}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{xx}^{-1}\mathbf{1}}$$

the following matrix identity results:

$$\frac{\mathbf{R}_{xx}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{xx}^{-1}\mathbf{1}} \equiv \frac{\mathbf{R}_{nn}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{nn}^{-1}\mathbf{1}} \tag{6.129}$$

Exploit (6.21) for the case when $\alpha = 1$; it then follows from (6.129) and (6.126) that

$$\mathbf{w}_{\text{mv}}(k+1) = \frac{(k+1)\mathbf{P}(k+1)\mathbf{1}}{(k+1)\mathbf{1}^T\mathbf{P}(k+1)\mathbf{1}} = \frac{\mathbf{P}(k+1)\mathbf{1}}{\mathbf{1}^T\mathbf{P}(k+1)\mathbf{1}} \tag{6.130}$$

where $\mathbf{P}(k+1)$ is given by (6.14). Note that when there is no desired signal, $\mathbf{w}_{\text{mv}}$ of (6.130) converges faster than when the desired signal is present, a result already found in Chapter 5.

6.5 SIMULATION RESULTS

The recursive processor defined by (6.66), (6.70), and (6.73) represents both the Kalman and the weighted least squares error processor, since the Kalman error covariance matrix with $\sigma^2(k) = 1$ corresponds to $\mathbf{P}(k)$ for the weighted least square error processor with $\alpha = 1$. The parameter $\sigma^2(k)$ for the Kalman processor is selected to be equal to the MMSE in accordance with (6.79). To simulate this recursive algorithm, it is necessary to define

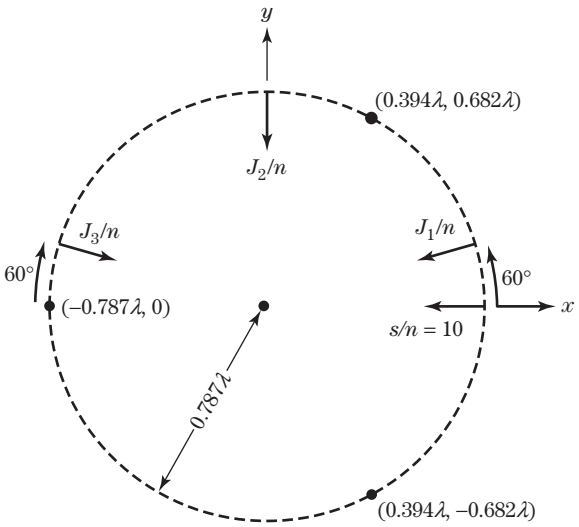


FIGURE 6-7 ■ Array geometry with signal and jammer locations for selected example.

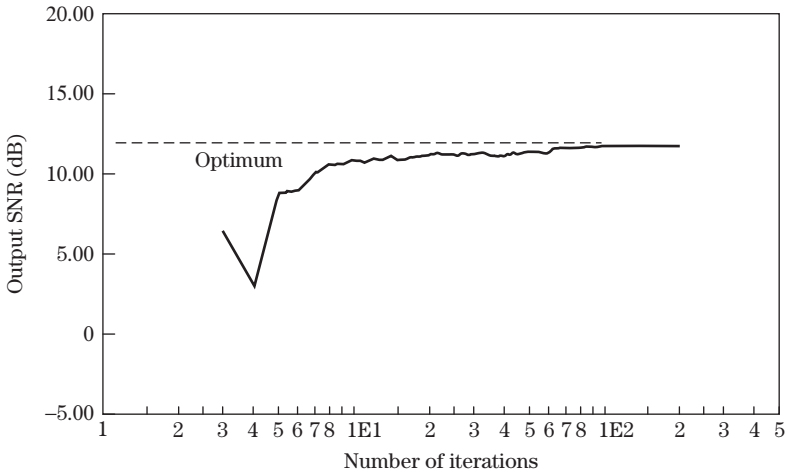


FIGURE 6-8 ■ Output SNR versus number of iterations for Kalman processor with eigenvalue spread of 16,700. Input $s/n = 10$ for which the output $SNR_{opt} = 15$ (11.76 dB) with algorithm parameters $\sigma^2 = \text{MMSE}$, $\mathbf{w}(0) = \mathbf{0}$, and $\mathbf{P}(0) = \mathbf{I}$.

an array geometry and signal environment as in Figure 6-7 (which duplicates Figure 4-30 and is repeated here for convenience). This figure depicts a four-element Y array having $d = 0.787\lambda$ element spacing with one desired signal at 0° and three distinct narrowband Gaussian jamming signals located at 15°, 90°, and 165°.

The desired signal in each case was taken to be a biphase modulated signal having a phase angle of either 0° or 180° with equal probability at each sample. Two signal environments are considered corresponding to eigenvalue spreads of 16,700 and 2,440. Figures 6-8 to 6-11 give convergence results for an eigenvalue spread of 16,700, where the jammer-to-thermal noise ratios are $J_1/n = 4000$, $J_2/n = 400$, and $J_3/n = 40$, for which the corresponding noise covariance matrix eigenvalues are given by $\lambda_1 = 1.67 \times 10^4$, $\lambda_2 = 1 \times 10^3$, $\lambda_3 = 29.0$, and $\lambda_4 = 1.0$. The input signal-to-thermal noise ratio is $s/n = 10$ for Figures 6-8 and 6-9, $s/n = 0.1$ for Figure 6-10, and $s/n = 0.025$ for Figure 6-11. The performance of the algorithm in each case is recorded in terms of the output SNR versus number of iterations, where one weight iteration occurs with each new independent data sample.

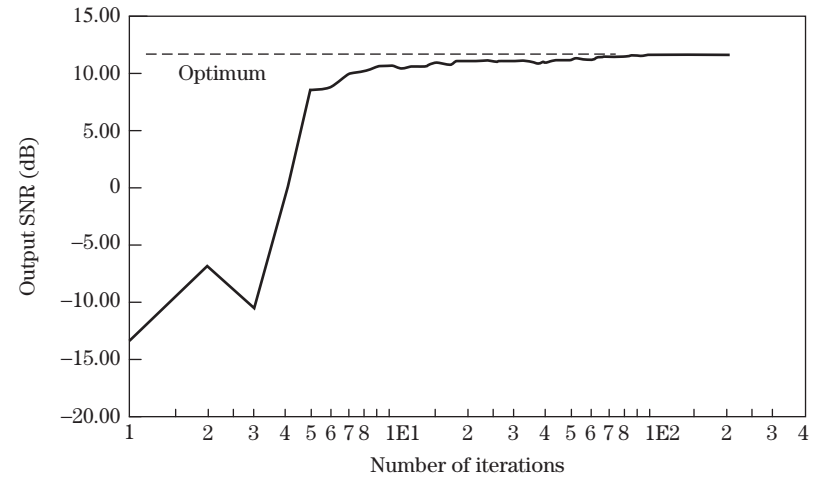


FIGURE 6-9 ■ Output SNR versus number of iterations for weighted least squares error processor with eigenvalue spread of 16,700. Input $s/n = 10$ for which output $\text{SNR}_{\text{opt}} = 15$ with algorithm parameters $\alpha = 1$, $\mathbf{w}^T(0) = [1, 0, 0, 0]$, and $\mathbf{P}(0) = \mathbf{I}$.

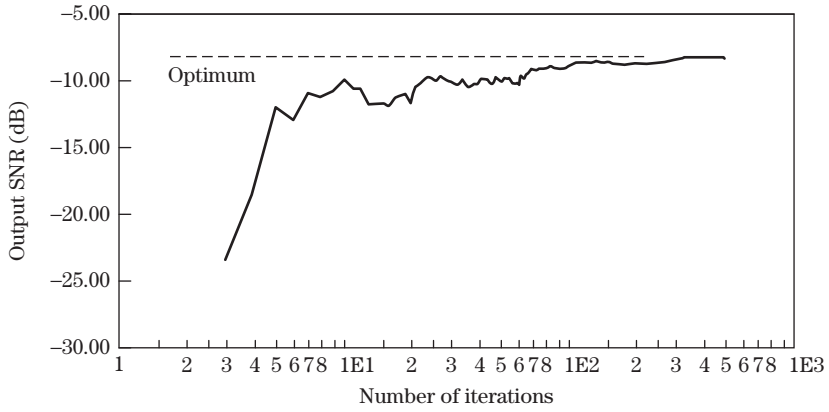


FIGURE 6-10 ■ Output SNR versus number of iterations for Kalman processor with eigenvalue spread of 16,700. Input $s/n = 0.1$ for which output $\text{SNR}_{\text{opt}} = 0.15\ (-8.24\ \text{dB})$ with algorithm parameters $\sigma^2 = \text{MMSE}$, $\mathbf{w}(0) = \mathbf{0}$, and $\mathbf{P}(0) = \mathbf{I}$.

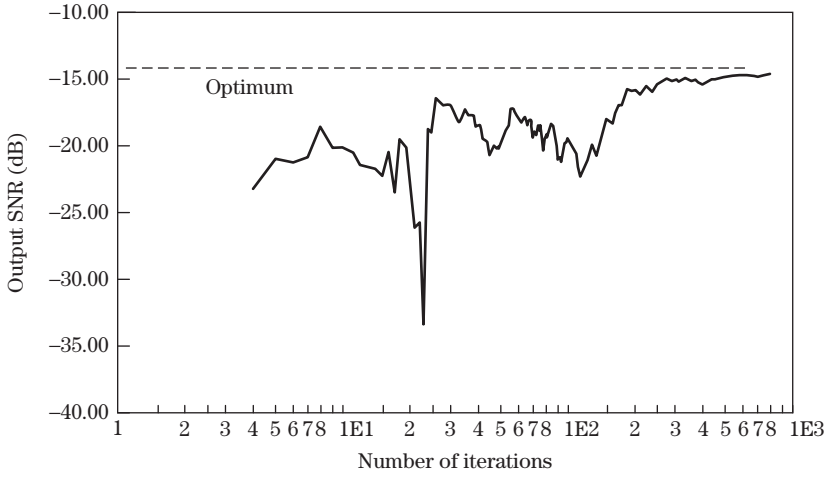


FIGURE 6-11 ■ Output SNR versus number of iterations for Kalman processor with eigenvalue spread of 16,700. Input $s/n = 0.025$ for which output $\text{SNR}_{\text{opt}} = 0.038\ (-14.2\ \text{dB})$ with algorithm parameters $\sigma^2 = \text{MMSE}$, $\mathbf{w}(0) = \mathbf{0}$, and $\mathbf{P}(0) = \mathbf{I}$.

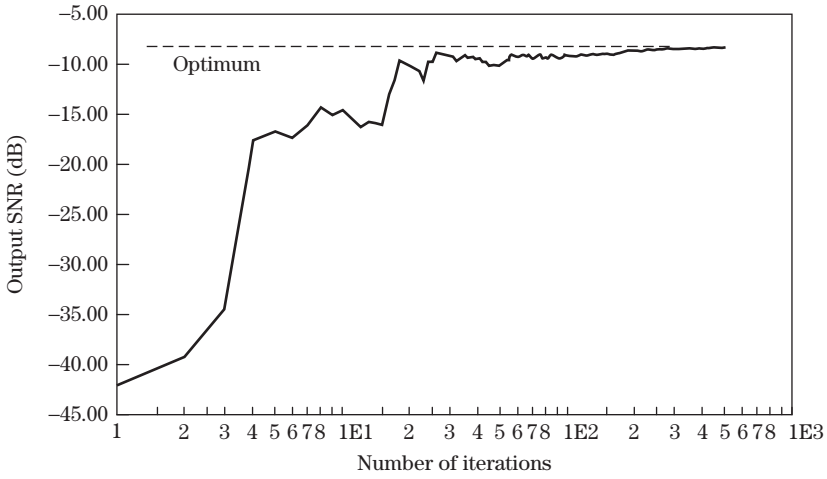


FIGURE 6-12 ■ Output SNR versus number of iterations for Kalman processor with eigenvalue spread of 2440. Input $s/n = 0.1$ for which output $\text{SNR}_{\text{opt}} = 0.15(-8.24\ \text{dB})$ with algorithm parameters $\sigma^2 = \text{MMSE}$, $\mathbf{w}(0) = \mathbf{0}$, and $\mathbf{P}(0) = \mathbf{I}$.

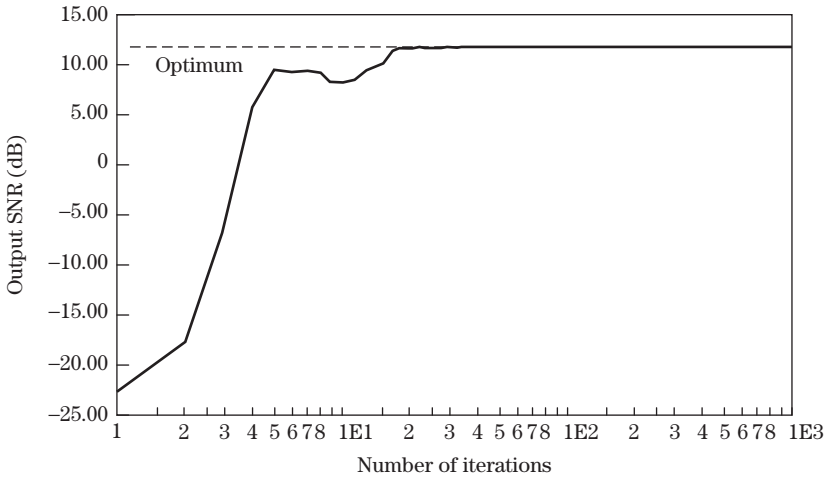


FIGURE 6-13 ■ Output SNR versus number of iterations for Kalman processor with eigenvalue spread of 2440. Input $s/n = 10$ for which output $\text{SNR}_{\text{opt}} = 15\ (11.76\ \text{dB})$ with algorithm parameters $\sigma^2 = \text{MMSE}$, $\mathbf{w}(0) = \mathbf{0}$, and $\mathbf{P}(0) = \mathbf{I}$.

Figures 6-12 and 6-13 give convergence results for an eigenvalue spread of 2,440, where the jammer-to-thermal noise ratios are $J_1/n = 500$, $J_2/n = 200$, and $J_3/n = 40$, for which the corresponding noise covariance matrix eigenvalues are given by $\lambda_1 = 2.44 \times 10^3$, $\lambda_2 = 4.94 \times 10^2$, $\lambda_3 = 25.62$, and $\lambda_4 = 1.0$. The input signal-to-thermal noise ratio is $s/n = 0.1$ for Figure 6-12 and $s/n = 10.0$ for Figure 6-13.

The simulation results shown in Figures 6-8 to 6-13 illustrate the following important properties of the recursive algorithms:

1. Recursive algorithms exhibit fast convergence comparable to that of DMI algorithms, especially when the output SNR_{opt} is large (approximately five or six iterations when $\text{SNR}_{\text{opt}} = 15.0$ for the examples shown). On comparing Figure 6-8 (for an eigenvalue spread of 1.67×10^4) with Figure 6-13 (for an eigenvalue spread of 2.44×10^3), it is seen that the algorithm convergence speed is insensitive to eigenvalue spread, which also reflects the similar property exhibited by DMI algorithms.
2. Algorithm convergence is relatively insensitive to the value selected for the parameter $\sigma^2(k)$ in (6.69). On comparing the results obtained in Figure 6-8 where $\sigma^2 = \text{MMSE}$

with the results obtained in Figure 6-9 where $\sigma^2 = 1$, it is seen that virtually the same number of iterations are required to arrive within 3 dB of SNR_{opt} even with different initial weight vectors.

3. The convergence speed of the recursive algorithm for the examples simulated here is slower for small values of SNR_{opt} and faster for large values of SNR_{opt}, as seen in Figures 6-8, 6-10, and 6-11. This behavior is also exhibited by the DMI algorithm (and to some degree by all algorithms that do not assume direction-of-arrival information). When no direction-of-arrival information is assumed, such information can be "learned" from a strong desired signal component.

6.6 SUMMARY AND CONCLUSIONS

The specific form selected for a recursive processor should reflect the data weight scheme that is appropriate for the desired application. The various recursive algorithms may be developed by applying the matrix inversion lemma to the same basic weight update equation.

Since the recursive algorithms are different from a DMI algorithm primarily because the required matrix inversion is accomplished in a recursive manner, it is hardly surprising that many of the desirable properties found to apply to DMI algorithms also hold for recursive algorithms. Rapid convergence rates and insensitivity to eigenvalue spread are characteristics that make recursive processors attractive algorithm candidates provided sufficient computational power and accuracy are available to carry out the required calculations.

6.7 PROBLEMS

1. ***The Minimum Variance Weight Vector Solution***

   (a) Show that setting the derivative of $\Re_{\text{mvm}}$ of (6.111) equal to zero yields

   $$\mathbf{w}_{\text{mv}} = \mathbf{R}_{nn}^{-1}\mathbf{I}_1\lambda$$

   (b) Show that requiring $\mathbf{w}_{\text{mv}}$ obtained in part (a) to satisfy (6.110) results in

   $$\lambda = \left[\mathbf{I}_1^T\mathbf{R}_{nn}^{-1}\mathbf{I}_1\right]^{-1}\mathbf{c}$$

   (c) Show that substituting the result obtained in part (b) into $\mathbf{w}_{\text{mv}}$ of part (a) results in (6.112).

2. ***Equivalence of the Maximum Likelihood and Minimum Variance Estimates [15].*** In some signal reception applications, the desired signal waveform is completely unknown and cannot be treated as a known waveform or even as a known function of some unknown parameters. Hence, no a priori assumptions regarding the signal waveform are made, and the waveform is regarded as an unknown time function that is to be estimated. One way of obtaining an undistorted estimate of an unknown time function with a signal-aligned array like that of Figure 6-6 is to employ a maximum likelihood estimator that assumes that the noise components of the received signal have a multidimensional Gaussian distribution. The likelihood function of the received

signal at the SCF output can then be written as

$$\Re = \frac{1}{(2\pi)^{N/2(2k+1)}|\boldsymbol{\rho}|^{1/2}} \exp\left[-\frac{1}{2}\sum_{m,n=-k}^{k}(\mathbf{x}_m - \mathbf{s}_m))^T \boldsymbol{\rho}^{-1}(\mathbf{x}_n - \mathbf{s}_n)\right]$$

where n, m denote distinct sample times, and ρ is the noise covariance matrix that is a matrix of $N \times N$ submatrices corresponding to the various tap points along the tapped delay line. Differentiate the logarithm of the likelihood function with respect to $\mathbf{s}_n$ and equate the result to zero to obtain $\hat{\mathbf{s}}_m$, the maximum likelihood estimator for $\mathbf{s}_m$. Show that this result corresponds to the signal estimate obtained from the processor defined by the result in Problem 1.

3. ***Derivation of Optimum Weights via Lagrange Multipliers [18].*** Show that the solution to the optimization problem posed by (6.113) and (6.114) is given by (6.115).

4. ***Development using the M.I.L [18].*** Show that (6.124) leads to (6.125) by means of the following steps:

   (a) Pre- and postmultiply (6.14) by $\mathbf{I}_1^T$ and $\mathbf{I}_1$, respectively, to obtain

   $$\mathbf{I}_1^T\mathbf{P}(k+1)\mathbf{I}_1 = \frac{1}{\alpha}\left[\mathbf{I}_1^T\mathbf{P}(k)\mathbf{I}_1 - \frac{\mathbf{I}_1^T\mathbf{P}(k)\mathbf{x}(k+1)\mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{I}_1}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}\right]$$

   (b) Apply the matrix identity (D.4) of Appendix D to the result obtained in part (a) to show that

   $$\left[\mathbf{I}_1^T\mathbf{P}(k+1)\mathbf{I}_1\right]^{-1} = \alpha\left[\mathbf{I}_1^T\mathbf{P}(k)\mathbf{I}_1\right]^{-1} + \mathbf{I}_1^{-1}\mathbf{x}(k+1)\mathbf{x}^T(k+1)\mathbf{I}_1^{-T}$$

   (c) Using the result of part (b) in (6.124), show that

   $$\mathbf{w}_{\text{opt}}(k+1) = \mathbf{P}(k+1)[\alpha\mathbf{P}^{-1}(k) + \mathbf{x}(k+1)\mathbf{x}^T(k+1)]\mathbf{I}_1^{-T}\mathbf{c}$$

   (d) Noting that

   $$\mathbf{P}(k+1)\mathbf{P}^{-1}(k) = \frac{1}{\alpha}\left[\mathbf{I} - \frac{\mathbf{P}(k)\mathbf{x}(k+1)\mathbf{x}^T(k+1)}{\alpha + \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}\right]$$

   show that the result obtained in part (c) leads to (6.125).

5. ***Derivation using the M.I.L. [18].*** Show that the matrix identity (6.129) results from the substitution of (6.128) into the ratio

   $$\frac{\mathbf{R}_{xx}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{R}_{xx}^{-1}\mathbf{1}}$$

6. ***Derivation of the Orthogonal Matrix Relation of Equation (6.77) [5]***

   (a) Equation (6.76) suggests that $\mathbf{P}(n)$ may be partitioned in the following manner:

   $$\mathbf{P}(n) = \begin{bmatrix}\mathbf{E} & \psi \\ \eta^H & h\end{bmatrix}$$

   where $\mathbf{E}$ is $N \times N$, $\boldsymbol{\psi}$ and $\boldsymbol{\eta}$ are both $N \times 1$, and h is a scalar.
   Show that substituting the aforementioned $\mathbf{P}(n)$ of part (a) into (6.76) gives the following relations:

   $$\frac{\mathbf{E}\mathbf{U}^{-H}(n-1)}{\sqrt{\mu}} = \mathbf{U}^{-H}(n) \quad \text{and} \quad \frac{\eta^H\mathbf{U}^{-H}(n-1)}{\sqrt{\mu}} = \mathbf{g}^H(n)$$

(b) Recognizing that $\mathbf{P}(n)$ is orthogonal ($\mathbf{P}(n)\mathbf{P}^H(n) = \mathbf{I}$), show that substituting $\mathbf{P}(n)$ of part (a) into the orthogonal condition yields the following relationships:

$$\begin{aligned}\mathbf{E}\mathbf{E}^H + \boldsymbol{\Psi}\boldsymbol{\Psi}^H &= \mathbf{I}\\ \mathbf{E}\boldsymbol{\eta} + \boldsymbol{\Psi}h &= \mathbf{0}\\ h^2 + \boldsymbol{\eta}^H\boldsymbol{\eta} &= \mathbf{1}\end{aligned}$$

(c) Show that substituting (6.75) into the corresponding result of part (a) and defining

$$\mathbf{a}(n) = \frac{\mathbf{U}^{-H}(n-1)\mathbf{x}(n)}{\sqrt{\mu}}$$

results in $\boldsymbol{\eta} = \frac{\mathbf{a}(n)}{t(n)}$

(d) Show that substituting $\boldsymbol{\eta}$ from part (c) into the third relationship of part (b) gives

$$h = \frac{1}{t(n)}$$

(e) The unknown vector $\boldsymbol{\psi}$ may be eliminated by using h of part (d) and $\boldsymbol{\eta}$ of part (c) in the second relationship of part (b) to give

$$\boldsymbol{\psi} = -\mathbf{E}\mathbf{a}(n)$$

(f) Form the product of $\mathbf{P}(n)$ with the augmented vector $[\mathbf{z}^H(n)1]^H$ and use the partitioning of part (a) for $\mathbf{P}(n)$ to produce

$$\mathbf{P}(n)\begin{bmatrix}\mathbf{z}(n)\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{E} & \boldsymbol{\Psi}\\ \eta & h\end{bmatrix}\begin{bmatrix}\mathbf{z}(n)\\ 1\end{bmatrix}$$

(g) Finally, show that substituting the results from parts (c) and (d) into the result of part (f) yields the desired result

$$\mathbf{P}(n)\begin{bmatrix}\mathbf{z}(n)\\ 1\end{bmatrix} = \begin{bmatrix}0\\ t(n)\end{bmatrix}$$

7. ***RLS algorithm.*** An eight-element uniform array with $\lambda/2$ spacing has the desired signal incident at 0° and two interference signals incident at -21° and 61°. Use the RLS algorithm to place nulls in the antenna pattern. Assume that $\sigma_n = 0.01$.

8. ***RLS algorithm.*** Plot the received signal as a function of iteration for the RLS and LMS algorithms when a 0 dB desired signal is incident on the eight-element array at 0° while 12 dB interference signals are incident at -21° and 61°.

6.8 REFERENCES

[1] C. A. Baird, "Recursive Algorithms for Adaptive Arrays," Final Report, Contract No. F30602-72-C-0499, Rome Air Development Center, September 1973.

[2] P. E. Mantey and L. J. Griffiths, "Iterative Least-Squares Algorithms for Signal Extraction," *Second Hawaii International Conference on System Sciences*, January 1969, pp. 767–770.

[3] C. A. Baird, "Recursive Processing for Adaptive Arrays," *Proceedings of the Adaptive Antenna Systems Workshop*, March 11–13, 1974, Vol. I, NRL Report 7803, Naval Research Laboratory, Washington, DC, pp. 163–182.

[4] R. D. Gitlin and F. R. Magee, Jr., "Self-Orthogonalizing Adaptive Equalization Algorithms," *IEEE Trans. Commun*., Vol. COM-25, No. 7, July 1977, pp. 666–672.

[5] C. A. Baird, "Kalman-Type Processing for Adaptive Antenna Arrays," *IEEE International Conference on Communications*, June 1974, Minneapolis, Minnesota, pp. 10G-1–10G-4.

[6] C. A. Baird, Jr. and J. T. Rickard, "Recursive Estimation in Array Processing," *Fifth Asilomar Conference on Circuits and Systems*, Pacific Grove, CA, November 1971, pp. 509–513.

[7] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. IEEE*, Vol. 57, No. 10, October 1969, pp. 1695–1704.

[8] L. E. Brennan, J. D. Mallet, and I. S. Reed, "Adaptive Arrays in Airborne MTI Radar," *IEEE Trans. Antennas Propag*., Vol. AP-24, No. 5, September 1976, pp. 607–615.

[9] J. M. Shapard, D. Edelblute, and G. Kinnison, "Adaptive Matrix Inversion," Naval Undersea Research and Development Center, NUC-TN-528, May 1971.

[10] L. E. Brennan and I. S. Reed, "Theory of Adaptive Radar," *IEEE Trans. Aerosp. Electron. Syst*., Vol. AES-9, No. 2, March 1973, pp. 237–252.

[11] S. J. Chern and C.Y. Chang, "Adaptive Linearly Constrained Inverse QRD-RLS Beamforming Algorithm for Moving Jammers Suppression," *IEEE Trans. Ant. & Prop.*, Vol. AP-50, No. 8, August 2002, pp. 1138–1150.

[12] R. Schreiber, "Implementation of Adaptive Array Algorithms," *IEEE Trans. Acoustics., Speech, & Signal Processing*, Vol. ASSP-34, No. 5, October 1986, pp. 205–212.

[13] S. T. Alexander and A. L. Ghirnikar, "A Method for Recursive Least Squares Filtering Based Upon an Inverse QR Decomposition," *IEEE Trans. Signal Processing*, Vol. 41, No. 1, January 1993, pp. 20–30.

[14] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Eng*., Series D, Vol. 82, 1960, pp. 35–45.

[15] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Trans. ASME, J. Basic Eng*. Series D, Vol. 83, March 1961, pp. 95–108.

[16] D. Godard, "Channel Equalization Using a Kalman Filter for Fast Data Transmission," *IBM J. Res. Dev*., May 1974, pp. 267–273.

[17] C. A. Baird, Jr., "Recursive Minimum Variance Estimation for Adaptive Sensor Arrays," *Proceedings of the IEEE 1972 International Conference on Cybernetics and Society*, October 9– 12, Washington, DC, pp. 412–414.

[18] J. Capon, R. J. Greenfield, and R. J. Kolker, "Multidimensional Maximum-Likelihood Processing of a Large Aperture Seismic Array," *Proc. IEEE*, Vol. 55, No. 2, February 1967, pp. 192–211.

[19] F. B. Gross, Smart antennas for wireless communications: with MATLAB, New York: McGraw-Hill, 2005.

CHAPTER 7

Cascade Preprocessors

Chapter Outline

The least mean squares (LMS) and maximum signal-to-noise ratio (SNR) algorithms converge slowly whenever there is a wide spread in the eigenvalues of the input signal correlation matrix. A wide eigenvalue spread occurs if the signal environment includes a very strong source of interference together with other weaker but nevertheless potent interference sources. This condition also happens when two or more very strong interference sources arrive at the array from closely spaced but not identical directions.

It was shown in Chapter 4 that by appropriately selecting the step size and moving in suitably chosen directions an accelerated gradient procedure offers marked improvement in the convergence rate over that obtained with an algorithm that moves in directions determined by the gradient alone. Another approach for obtaining rapid convergence rescales the space in which the minimization is taking place by appropriately transforming the input signal coordinates so that the constant cost contours of the performance surface (represented by ellipses in Chapter 4) are approximately circular and no eigenvalue spread is present in the rescaled space. If such a rescaling is done, then in principle it would be possible to correct all the error components in a single step by choosing an appropriate step size.

With a method called scaled conjugate gradient descent (SCGD) [1], this philosophy is followed with a procedure that employs a CGD cycle of N iterations and uses the information gained from this cycle to construct a scaling matrix that yields very rapid convergence on the next CGD cycle.

The philosophy behind the development of cascade preprocessors is similar to that of the SCGD method. The cascade preprocessor introduced by White [2,3] overcomes the problem of (sometimes) slow convergence by reducing the eigenvalue spread of the input signal correlation matrix by introducing an appropriate transformation (represented by the preprocessing network). Used in this manner, the cascade network resolves the input

signals into their eigenvector components. Equalizing the resolved signals with automatic gain control (AGC) amplifiers reduces the eigenvalue spread, thereby simplifying the task of any gradient algorithm.

By modifying the performance measure governing the control of the adaptive elements in a cascade preprocessor, a cascade network performs the complete task of array pattern null steering without any need for a gradient type processor [4]. Using a cascade preprocessor in this manner reduces the complexity and cost of the overall processor and represents an attractive alternative to conventional gradient approaches.

Finally, we introduce the use of a cascade preprocessor developed by Brennan et al. [5–7] to achieve adaptive null steering based on the Gram–Schmidt orthogonalization procedure [8–10]. A Gram–Schmidt cascade preprocessor is simpler than the other cascade networks discussed and possesses very fast convergence properties that make it a most appealing practical alternative [11]. The discussion of eigenvector component preprocessing networks gives perspective to the development and use of cascade preprocessors.

7.1 NOLEN NETWORK PREPROCESSOR

Suppose that it were possible to construct a lossless transformation network to insert between the sensor elements and the adaptive weights that resolves the input signals into their eigenvector components, so the correlation matrix of the transformed signals was diagonal with the eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ at the elements. The insertion of such a transformation network ahead of the adaptive weights to resolve the signals into orthogonal normalized eigenvector beams was suggested by Gabriel [12]. If a second transformation equalizes the various eigenvalues, then in principle it should be possible to select a step size that convergences in a single step. If the necessary a priori eigenvalue information were available to the designer, then a five-element array would have the transformation matrix and equalizing network shown in Figure 7-1.

Since the eigenvalues of the input signal correlation matrix are unknown, an approximation of the eigenvector network must be constructed. Otherwise, when the correlation matrix is ill conditioned (i.e., the eigenvalues are widely diverse), then small errors made in estimating certain matrix components become magnified enough that the signals cannot be resolved into eigenvector components. Fortunately, the method presented here works quite well with only a rough approximation. A preprocessor excels under the conditions that are the most difficult for gradient-type algorithms.

7.1.1 Single-Stage Nolen Network

Consider the lossless, passive, reflectionless network shown in Figure 7-2. This network has N input ports (one for each array sensor output) and N output ports. The network elements consist of $N - 1$ variable phase shifters and $N - 1$ variable directional couplers that are connected in a series feed configuration. This network was first described by Nolen in an unpublished Bendix memorandum [13] concerned with the problem of synthesizing multiple beam antennas [14] and is therefore termed a Nolen transformation network. Since the network is passive, lossless, and reflectionless, the total output power is the same as the total input power, and the overall transformation matrix is unitary.

The signals in the transformation network are denoted by v_n^k, where k indicates the level in the processor, and n indicates the element channel to which the signal corresponds.

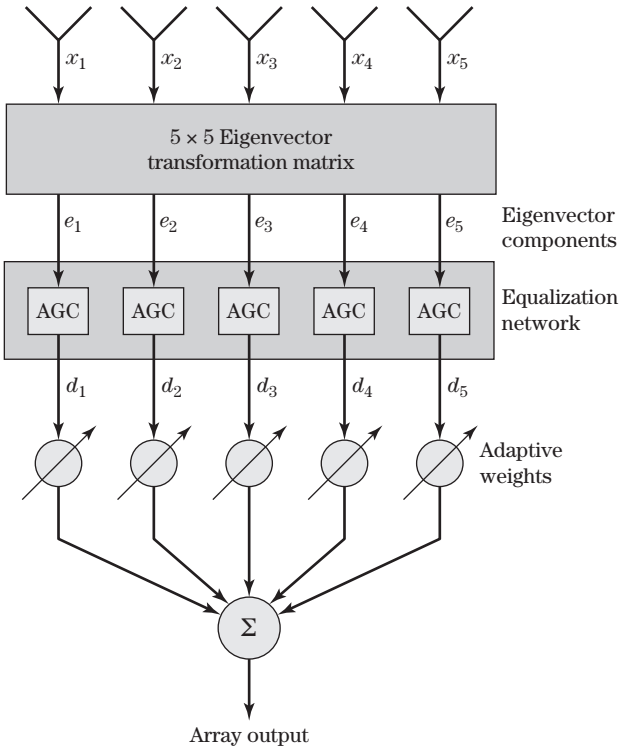


FIGURE 7-1 ■ Five-element adaptive array with eigenvector component transformation and eigenvalue equalization network.

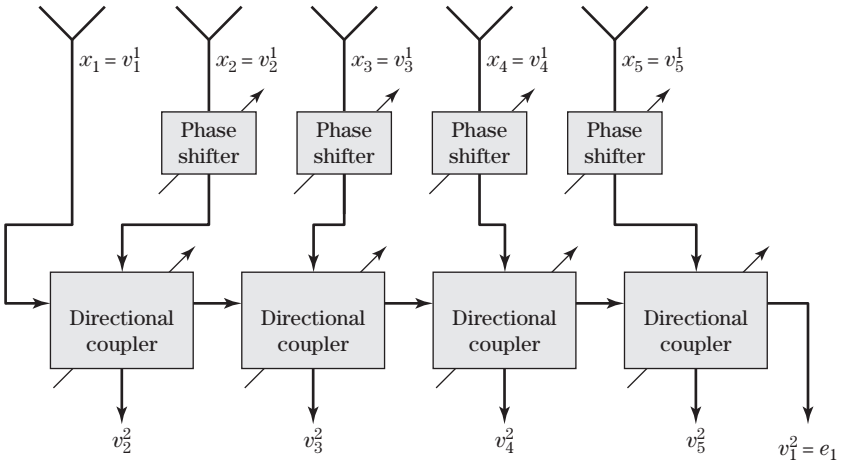


FIGURE 7-2 ■ Single-stage lossless, passive, reflectionless Nolen transformation network for $N = 5$.

The output signal in Figure 7-2 is expressed as

$$v_1^2 = \sum_{n=1}^{N} a_n v_n^1 \tag{7.1}$$

where the weighting factors are constrained by the unitary condition $\sum_n |a_n|^2 = 1$ and the condition that a_1 must be real.

Now suppose we want to maximize the output power resulting from the single-level Nolen network. The output power is expressed as

$$\begin{aligned} P_1 = E\{|v_1^2|^2\} = E\{v_1^2 v_1^{2*}\} &= \sum_{n=1}^{N}\sum_{l=1}^{N} a_n E\{v_n^1 v_l^{1*}\} a_l^* \\ &= \sum_{n=1}^{N}\sum_{l=1}^{N} a_n m_{nl}^1 a_l^* \end{aligned} \tag{7.2}$$

where

$$m_{nl}^k = E\{v_n^k v_l^{k*}\} \tag{7.3}$$

is an element of the correlation matrix of the input signals. To introduce the unitary constraint $\sum_n |a_n|^2 = 1$ while maximizing P_1, employ the method of Lagrange multipliers by maximizing the quantity

$$Q = P_1 + \lambda \left[1 - \sum_{n=1}^{N} |a_n|^2\right] \tag{7.4}$$

where λ denotes a Lagrange multiplier. Setting $a_n = u_n + jv_n$, the partial derivatives of Q with respect to u_L and v_L is found to be

$$\begin{aligned} \frac{\partial Q}{\partial u_L} &= \sum_n m_{Ln}^1 (u_n - jv_n) + \sum_l (u_l + iv_l) m_{lL}^1 - 2\lambda u_L \\ &= \sum_n a_n m_{nL}^1 + \sum_l m_{Ll}^1 a_l^* - \lambda(a_L + a_L^*) \end{aligned} \tag{7.5}$$

$$\begin{aligned} \frac{\partial Q}{\partial v_L} &= j\sum_n m_{Ln}^1 (u_n - jv_n) - j\sum_l (u_l + jv_l) m_{lL}^1 - 2\lambda v_L \\ &= -j\sum_n a_n m_{nL}^1 + j\sum_l m_{Ll}^1 a_l^* + j\lambda(a_L - a_L^*) \end{aligned} \tag{7.6}$$

Setting both $\partial Q/\partial u_L$ and $\partial Q/\partial v_L = 0$, the solution for a_L must satisfy the following conditions:

$$\left.\begin{aligned} \lambda(a_L + a_L^*) &= \sum_n a_n m_{nL}^1 + \sum_l m_{Ll}^1 a_l^* \\ \lambda(a_L - a_L^*) &= \sum_n a_n m_{nL}^1 - \sum_l m_{Ll}^1 a_l^* \end{aligned}\right\} \tag{7.7}$$

The conditions represented by (7.7) immediately simplify to

$$\lambda a_L = \sum_{n=1}^{N} a_n m_{nL}^1 \tag{7.8}$$

which is a classical eigenvector equation. Nontrivial solutions of (7.8) exist only when λ has values corresponding to the eigenvalues of the matrix $\mathbf{M}^1$ (of which m_{nL}^1 is the nLth element).

Substituting (7.8) into (7.2) yields

$$P_1 = \lambda \sum_{n=1}^{N} a_n a_n^* \tag{7.9}$$

In view of the unitary constraint $\sum |a_n|^2 = 1$, (7.9) becomes

$$P_1 = \lambda \tag{7.10}$$

Consequently, the largest value of P_1 (which is the maximum power available at the right-hand output port) is just the largest eigenvalue of the matrix $\mathbf{M}^1$.

When the phase shifters and directional couplers are adjusted to maximize P_1, then the signal v_1^2 in Figure 7-2 is orthogonal to all the other signals v_k^2. That is,

$$E\{v_1^2 v_k^{2*}\} = 0 \quad \text{for } k = 2, 3, \ldots, N \tag{7.11}$$

This orthogonality condition reflects the fact that no other unitary combination of v_1^2 and any of the other v_k^2 signals delivers more power than is obtained with v_1^2 alone—such a result would contradict the fact that P_1 has been maximized.

The result of (7.11) shows that all the off-diagonal elements in the first row and the first column of the covariance matrix of the signals v_k^2 are set equal to zero. Consequently, the transformation introduced by the single-stage network of Figure 7-2 is step one in diagonalizing the covariance matrix of the input signals. To complete the diagonalization, additional networks are introduced as described in the next section.

7.1.2 Cascaded Networks

Now let the single-stage transformation network of Figure 7-2 be followed by a cascade of similar networks as shown in Figure 7-3. The second transformation network operates only on the signals v_2^2 through v_5^2, leaving $v_1^2 = \varepsilon_1$ undisturbed. Because of the reflectionless character of the transformation networks, the parameters of the second network are adjusted without affecting the prior maximization of P_1. Adjust the parameters of the second transformation network to maximize P_2—the output power from the right-hand port of Level 2. The maximum power now available equals the largest eigenvalue of the covariance matrix of the input set $\{v_2^2, v_3^2, \ldots, v_N^2\}$. If P_1 was truly maximized in the first level, then the largest eigenvalue of the submatrix equals the second largest eigenvalue of the complete covariance matrix.

The signals emerging from network 2 form the signal set $\{v_2^2, v_3^2, \ldots, v_N^2\}$ and adjusting the parameters of network 2 to maximize P_2 results in setting all the off-diagonal elements in the first row and first column of the submatrix for the signal set $\{v_2^2, v_3^2, \ldots, v_N^2\}$ to zero. As far as the complete covariance matrix is concerned, the first two transformation networks have diagonalized the first two rows and the first two columns. Each succeeding transformation network likewise diagonalizes one row and one column at a time. Once the output of the last transformation in the cascaded network system is reached, the entire matrix diagonalization is complete. Since the adjustment of parameters in each network leaves previous networks in the cascade undisturbed, there is no need to iterate the process unless the input signal statistics change.

Figure 7-1 shows that an equalization network that consists of AGC amplifiers through which the signals are now passed follows the eigenvector transformation matrix. As a result, the signal powers on the various output leads $\{d_1, d_2, \ldots, d_N\}$ are equalized. Since the covariance matrix corresponding to the eigenvector component signals $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$ is diagonalized, the eigenvalues are consequently equalized. Therefore, the covariance matrix of the output signals from the equalization network is a scalar constant times the identity matrix.

The transformation resulting from the entire cascade of Figure 7-3 is unitary, since each of the networks in the cascade is unitary. Consequently, not only is the total output

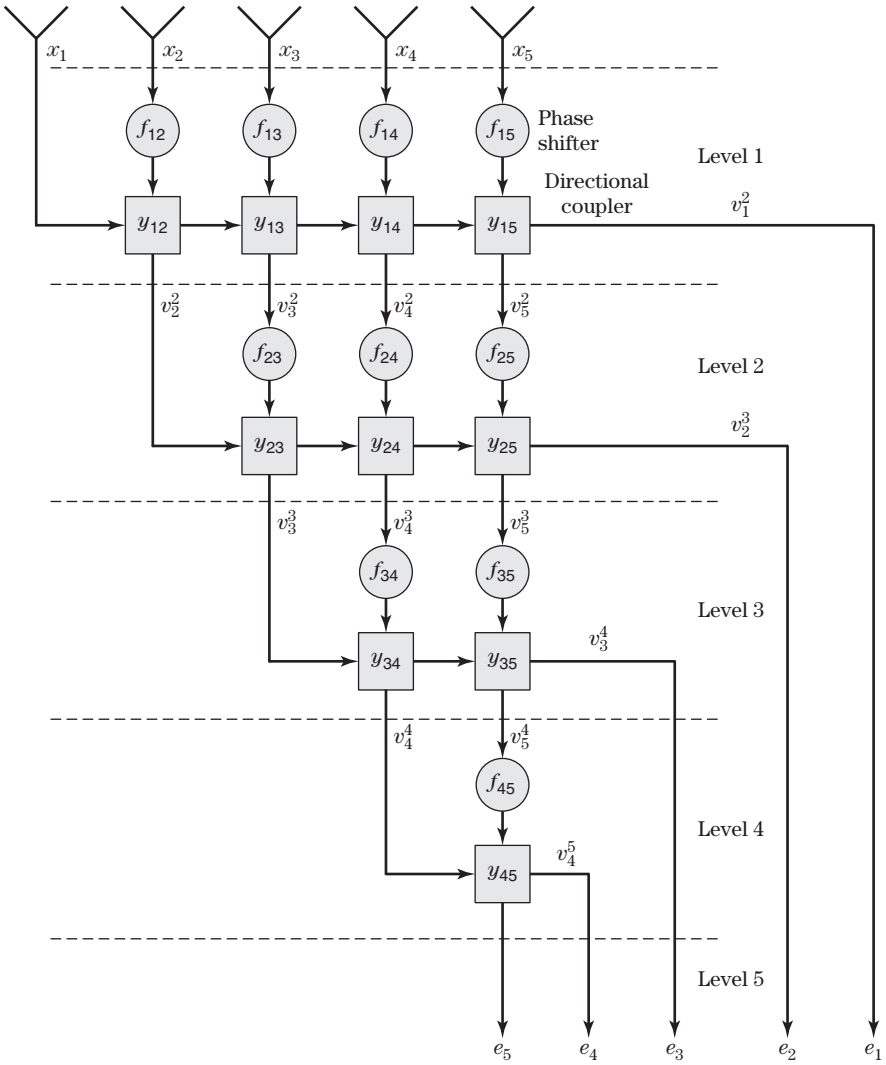


FIGURE 7-3 ■ Nolen cascade network for five-element array.

power strictly equal to the total input power, but also the eigenvalues of the covariance matrix are unchanged. If the element $\mathbf{G}_J$ represents the overall transfer matrix of the first J stages, then $\mathbf{G}_J$ is a product of factors in which each factor represents a single stage, that is,

$$\mathbf{G}_J = \mathbf{F}_J \cdot \mathbf{F}_{J-1} \cdots \mathbf{F}_2 \cdot \mathbf{F}_1 \tag{7.12}$$

where

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_J \\ v_{J+1}^{J+1} \\ \vdots \\ v_N^{J+1} \end{bmatrix} = \mathbf{G}_J \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \tag{7.13}$$

Furthermore, the second and lower stages of the network have no effect on the signal ε_1. It follows that the first row of $\mathbf{G}_J$ is the same as the first row of

$$\mathbf{G}_1 = \mathbf{F}_1 \tag{7.14}$$

Likewise, since the third and lower stages of the network have no effect on F_2, the second row of $\mathbf{G}_J$ is identical to the second row of

$$\mathbf{G}_2 = \mathbf{F}_2 \cdot \mathbf{F}_1 \tag{7.15}$$

Continuing in the foregoing manner, the entire $\mathbf{G}_N$ matrix is implemented in the Nolen form using a step-by-step process. Since the first row of $\mathbf{G}_J$ is identical to the first row of $\mathbf{G}_1$, the second row of $\mathbf{G}_J$ is identical to the second row of $\mathbf{G}_2, \ldots$, and the $(J-1)$st row of $\mathbf{G}_J$ is identical to the $(J-1)$st row of $\mathbf{G}_{J-1}$, it follows that the first $J-1$ rows and $J-1$ columns of $\mathbf{F}_J$ are the same as those of an identity matrix. Furthermore, since there are no phase shifters between the leftmost input port at any stage and the right-hand output port, the Jth diagonal element of $\mathbf{F}_J$ is real. These constraints in addition to the unitary constraint on the transfer matrix define the bounds of the element values (phase shift and directional coupling) of the Jth row.

If the output power maximization at each stage of the cascade transformation network is only approximate, the off-diagonal elements of the covariance matrix are not completely nulled. However, with only a rough approximation, the off-diagonal elements are at least reduced in amplitude, and, although the equalization network will no longer exactly equalize the eigenvalues, it reduces the eigenvalue spread.

7.1.3 Control of the Eigenvector Transformation Networks

Consider the problem of adjusting the parameters in each stage of an eigenvector transformation network to realize the desired covariance matrix diagonalization. It turns out that the signal environment conditions that prove the most difficult for the LMS algorithm are the same conditions that prove to be easy for the preprocessor [3].

For the single-stage network of Figure 7-2, the power at the right-hand output port is maximized by appropriately adjusting each phase shifter and directional coupler in the network. Denote the signal flowing downward into the phase shifter directional coupler combination located in the lth row and the kth column of the cascade of Figure 7-3 by v_k^l. Likewise, denote the signal flowing into this same coupler from the left by y_k^l. The signal coming out of the right-hand port (which is equivalent to y_{N+1}^l) is then ε_l.

Regard the directional coupler as equivalent to a goniometer having shaft angle ψ_{lk}; the signals into and out of one phase shifter directional coupler combination are written as

$$y_{k+1}^l = y_k^l \cos\psi_{l,k} + y_k^l e^{j\phi_{l,k}} \sin\psi_{l,k} \tag{7.16}$$

$$v_k^{l+1} = y_k^l \sin\psi_{l,k} + v_k^l e^{j\phi_{l,k}} \cos\psi_{l,k} \tag{7.17}$$

Now $P_l = E\{|\varepsilon_l|^2\}$ is the power out of the lth stage where

$$\varepsilon_l = \sum_{k=l}^{N} a_k^l v_k^l \tag{7.18}$$

$$P_l = \sum_{i=l}^{N} \sum_{k=l}^{N} a_i^l m_{ik}^l a_k^{l*} \tag{7.19}$$

where m_{ik}^l is given by (7.3) and

$$a_i^l = \begin{cases} \prod_{k=l+1}^{N} \cos \psi_{lk} & \text{for } i = 1 \\ e^{j\phi_{li}} \sin \psi_{li} \prod_{k=i+1}^{N} \cos \psi_{lk} & \text{for } i > 1 \end{cases} \tag{7.20}$$

Substituting the expression $\sum_{k=l}^{N-1} a_k^l v_k^l = \varepsilon_l - a_N v_N^l$ and (7.20) into (7.19) and taking partial derivatives with respect to ϕ_{lk} and ψ_{lk} results in the following:

$$\frac{\partial P_l}{\partial \phi_{lk}} = c_1 \operatorname{Im} [E\{\varepsilon_l v_k^{l*}\} e^{-j\phi_{lk}}] \tag{7.21}$$

$$\frac{\partial P_l}{\partial \psi_{lk}} = c_2 \operatorname{Re} [E\{\varepsilon_l v_k^{l*}\} e^{j\phi_{lk}} \cos \psi_{lk} - E\{\varepsilon_l y_k^{l*}\} \sin \psi_{lk}] \tag{7.22}$$

where c_1 does not depend on ϕ_{lk}, and c_2 does not depend on ψ_{lk}. Setting both of the aforementioned partial derivatives equal to zero then yields

$$\phi_{lk} = \arg [E\{\varepsilon_l v_k^{l*}\}] \tag{7.23}$$

$$\psi_{lk} = \tan^{-1} \left[\frac{\operatorname{Re}(E\{\varepsilon_l v_k^{l*}\} e^{-j\phi_{lk}})}{\operatorname{Re}(E\{\varepsilon_l y_k^{l*}\})} \right] \tag{7.24}$$

where the arctangent function is a multiple valued function. It turns out that by taking the value lying in the range $0 \to \pi$ radians, the output power is maximized. From (7.23) and (7.24) it is seen that measuring the correlations $E\{\varepsilon_l v_l^{k*}\}$ and $E\{\varepsilon_l y_k^{l*}\}$ yields an indication of the correct settings for ϕ_{lk} and ψ_{lk}. Starting with the first phase shifter coupler pair at the extreme left of any stage and proceeding with each successive pair until the last phase shift coupler pair is reached, ϕ_{lk} and ψ_{lk} are adjusted in accordance with (7.23) and (7.24). However, each time a new setting of these parameters is made while proceeding toward the output port, the value of ε_l changes; therefore, the values of the correlations change, so new settings are required and the entire process is applied recursively. There is presently no proof of convergence for this recursive adjustment process, but experience indicates that convergence in five or six complete row adjustments is commonplace [3]. When the eigenvalues of the input signal covariance matrix are widely separated, then convergence is even faster.

The eigenvector transformation network parameter adjustment time is reduced, because an exact eigenvector decomposition is not required to realize substantial eigenvalue spread reduction. This may be done by resorting to a piecemeal maximization technique in which each phase shift coupler combination is set to produce maximum power at its horizontal output port without regard to power at the entire stage output port and considering only the signals present at the two inputs of each individual combination. With this piecemeal procedure, the phase shift and directional coupler are now set according to

$$\phi_{lk} = \arg(E\{y_k^l v_k^{l*}\}) \tag{7.25}$$

$$\psi_{lk} = \frac{1}{2} \tan^{-1} \left[\frac{2|E\{y_k^l v_k^{l*}\}|}{|E\{[y_k^l]^2\}| - |E\{[v_k^l]^2\}|} \right] \tag{7.26}$$

where $0 \le \tan^{-1}(\cdot) \le \pi$.

Each phase shift coupler combination is set from left to right along each row. At the end of a row, drop down to the next stage and repeat the process until the bottom

of the cascade is reached. The eigenvector beams produced by the piecemeal adjustment procedure are shown by White [3] to be surprisingly close to those obtained by the complete recursive adjustment procedure. If the adjustment of each phase shifter directional coupler combination is one iteration, then a total of $N(N-1)/2$ iterations completes the piecemeal adjustment procedure.

7.2 | INTERFERENCE CANCELLATION WITH A NOLEN NETWORK PREPROCESSOR

Instead of using a Nolen eigenvector component preprocessor in cascade with a gradient-type processor to alleviate the eigenvalue spread problem and speed convergence, the entire task of interference cancellation and desired signal preservation are accomplished using a Nolen cascade network alone by selecting a different performance measure from that adopted in the preceding section [4]. Each stage in the Nolen network of Figure 7-3 introduces one null in the radiation pattern corresponding to the bottom output port. Errors in parameter settings in the lower stages do not disturb the nulls set by the upper stages, but the inverse statement is not true, so a nonadaptive Nolen network is vulnerable to errors in the early stages. When adaptive control is applied to the parameter adjustment, however, a portion of the lower stage adjustment capability partially compensates for upper stage parameter setting errors. By virtue of the properties of the Nolen network, an appropriate set of parameter adjustments for interference suppression are determined by means of a step-by-step process in which each step involves only a single stage.

7.2.1 Problem Formulation

Consider the representation of a Nolen beamforming network shown in Figure 7-4 in which a general (passive, lossless, and matched) network having N input ports and N output ports follows an array having N elements. Under these conditions, the total output power equals the total input power, and the transfer function of the network is a unitary matrix.

Let $\mathbf{x}$ be a complex vector representing the input signal envelopes at each sensor element, and let $\boldsymbol{\varepsilon}$ be a complex vector representing the output signal envelopes, that is,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix} \tag{7.27}$$

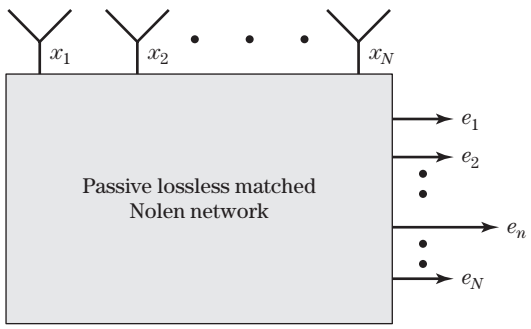


FIGURE 7-4 ■ Nolen beamforming network.

The vectors $\mathbf{x}$ and $\boldsymbol{\varepsilon}$ are related by

$$\boldsymbol{\varepsilon} = \mathbf{G}\mathbf{x} \tag{7.28}$$

where $\mathbf{G}$ is the transfer matrix of the Nolen network. Since $\mathbf{G}$ is unitary it follows that

$$\boldsymbol{\varepsilon}^{\dagger}\boldsymbol{\varepsilon} = \mathbf{x}^{\dagger}\mathbf{x} \tag{7.29}$$

$$\mathbf{G}^{\dagger}\mathbf{G} = \mathbf{I} \tag{7.30}$$

Equation (7.30) is referred to as the "unitary constraint" that $\mathbf{G}$ must satisfy. Define the covariance matrices of the input and the output signals by

$$\mathbf{R}_{xx} = E\{\mathbf{x}^{*}\mathbf{x}^{T}\} \tag{7.31}$$

$$\mathbf{R}_{\varepsilon\varepsilon} = E\{\boldsymbol{\varepsilon}^{*}\boldsymbol{\varepsilon}^{T}\} = \mathbf{G}^{*}\mathbf{R}_{xx}\mathbf{G}^{T} \tag{7.32}$$

The total output power appearing at the nth output port is the nth diagonal element of $\mathbf{R}_{\varepsilon\varepsilon}$. Each output signal consists of a desired signal component and an interference signal component (*interference* here denotes the sum of external jamming and internal thermal noise). Assume that the desired signal is not correlated with any of the interference signal sources; then $\mathbf{R}_{\varepsilon\varepsilon}$ are divided into signal and interference components

$$\mathbf{R}_{\varepsilon\varepsilon} = \mathbf{G}^{*}[\mathbf{R}_{ss} + \mathbf{R}_{II}]\mathbf{G}^{T} \tag{7.33}$$

where

$$\mathbf{R}_{ss} = E\{\mathbf{x}_{s}^{*}\mathbf{x}_{s}^{T}\} \tag{7.34}$$

$$\mathbf{R}_{II} = E\{\mathbf{x}_{I}^{*}\mathbf{x}_{I}^{T}\} \tag{7.35}$$

The vectors $\mathbf{x}_s$ and $\mathbf{x}_I$ represent the signal and interference components, respectively, of the input signal envelopes. At the nth output port denote the desired signal power by $\mathfrak{S}_n$ and the interference power by $\mathfrak{I}_n$ so that

$$\mathfrak{S}_n = [\mathbf{R}_{ss}]_{nn} \tag{7.36}$$

$$\mathfrak{I}_n = [\mathbf{R}_{II}]_{nn} \tag{7.37}$$

That is, the output power of interest is the nth diagonal element of the corresponding covariance matrix.

The goal is to maximize the signal power $\mathfrak{S}_n$ and to minimize the interference power $\mathfrak{I}_n$ at the output port n. Since these two objectives conflict, a trade-off between them is necessary. One approach to this trade-off selects the unitary transfer matrix, $\mathbf{G}$, that maximizes $\Re_n = \mathfrak{S}_n/\mathfrak{I}_n$. A more convenient way of attacking the problem is to adopt as the performance measure

$$\Omega_n = \mathfrak{S}_n - \Gamma\mathfrak{I}_n \tag{7.38}$$

where Γ is a fixed scalar constant that reflects the relative importance on minimizing interference compared with maximizing the signal. If $\Gamma = 0$, then the desired signal is maximized, while if $\Gamma \to \infty$, then only the interference is minimized.

There is a value $\Gamma = \Re_{\text{opt}}$ for which maximizing Ω_n produces exactly the same result as maximizing the ratio $\Re_n$. The value $\Re_{\text{opt}}$ is a function of the environment and is not ordinarily known in advance. By setting $\Gamma = \Re_{\min}$ where $\Re_{\min}$ is the minimum acceptable signal-to-interference ratio that provides acceptable performance, then maximizing Ω_n ensures that $\Re_n$ is maximized under the conditions when it is most needed. If the signal

environment improves, then $\Re_n$ also improves, although not quite as much as if $\Re_n$ were maximized directly.

It is useful to define the matrix

$$\mathbf{Z} \triangleq \mathbf{G}^* \mathbf{R}_{ss} \mathbf{G}^T - \Gamma \mathbf{G}^* \mathbf{R}_{II} \mathbf{G}^T \tag{7.39}$$

The performance measure Ω_n is then the nth diagonal element of $\mathbf{Z}$. Furthermore

$$\mathbf{Z} = \mathbf{G}^* \mathbf{M} \mathbf{G}^T \tag{7.40}$$

where

$$\mathbf{M} = \mathbf{R}_{ss} - \mathbf{R}_{II} \tag{7.41}$$

A consequence of (7.41) is that the matrix $\mathbf{M}$ (and hence $\mathbf{Z}$) have both positive and negative eigenvalues. Selecting the transfer matrix, $\mathbf{G}$, that maximizes the nth diagonal element of $\mathbf{Z}$ subject to (7.30) maximizes the element Ω_n. Using the method of Lagrangian multipliers to maximize the element Z_{nn} with the unitary constraint on $\mathbf{G}$ and setting the resulting gradient equal to zero yield the relation

$$\sum_L M_{nL} G_{Lm} = \lambda G_{mn} \tag{7.42}$$

Equation (7.42) is precisely the form of an eigenvector equation. Consequently, the nth column of $\mathbf{G}^T$ is the eigenvector of the matrix $\mathbf{M}$ corresponding to the largest eigenvalue. When the nth column of $\mathbf{G}^T$ is so constructed, then the element Z_{nn} equals this largest eigenvalue, and all other elements of the nth row and the nth column of $\mathbf{Z}$ vanish.

It is desired to obtain a unitary transfer matrix $\mathbf{G}$ for which the elements of the nth row are the elements of the eigenvector of the matrix $\mathbf{M}$ corresponding to the maximum eigenvalue (resulting in maximizing Ω_n). If the first stage of the cascade Nolen network is adjusted to minimize Ω_1, if there is no conflict with maximizing Ω_n, and if the first diagonal element of $\mathbf{Z}$ is set equal to the most negative eigenvalue of $\mathbf{M}$, then the off-diagonal elements of the first row and first column of $\mathbf{Z}$ will disappear, and the first row of $\mathbf{G}$ will correspond to the appropriate eigenvector. Likewise, proceed to adjust the second stage to minimize Ω_2. This second adjustment results in the diagonalization of the second row and the second column of $\mathbf{Z}$, and the second row of $\mathbf{G}$ corresponds to the second eigenvector. Continue in this manner adjusting in turn to minimize the corresponding Ω until reaching stage n. At this point (the nth stage) it is desired to maximize Ω_n so the adjustment criterion must be reversed.

The physical significance of the foregoing adjustment procedure is that the upper stages of the Nolen network are adjusted to maximize the interference and minimize the desired signal observed at the output of each stage. This process is the same as maximizing the desired signal and minimizing the interference that proceeds downward to the lower stages. When the nth stage is reached where the useful output is desired, then the desired signal should be maximized and the interference minimized so the adjustment criterion is reversed.

7.2.2 Recognition of the Desired Signal

The performance criterion must have the ability to distinguish between the desired signal and interference. If the desired signal comes from an unknown direction (as would usually be the case), it is necessary to recognize some distinguishing characteristics of the signals.

Since communication systems designed to operate in a jamming environment commonly employ some form of spread spectrum modulation, then the signal recognition scheme is based on the characteristics associated with spread spectrum signals.

Spread spectrum modulation applies some form of pseudo-random coding at the transmitter to widen the transmitted signal spectrum. When the receiver applies appropriate decoding to the spread spectrum signal, the original (unspread) desired signal is recovered with a narrow spectrum, whereas any interference still emerges from the decoder having a wide spectrum. The demodulation process for spread spectrum signals uses narrowband filters to at least partially separate the desired signal from the surrounding interference.

A first step toward determining whether the parameter adjustments in each stage of the Nolen network are succeeding in the diagonalization of the matrix $\mathbf{Z}$ is to define a generalized correlation product [4]

$$A^* \otimes B = E\{A_s^* B_s\} - \Gamma E\{A_I^* B_I\} \tag{7.43}$$

where A is the complex envelope of one waveform having desired signal and interference components A_s and A_I. Likewise, B represents the complex envelope of a second waveform having desired signal and interference components B_s and B_I. If we assume that the desired signal is modulated with pseudo-random phase reversals and that a synchronized key generator is available for demodulation at the receiver, Figure 7-5 shows the block diagram of a generalized correlator that forms an estimate of $A^* \otimes B$. After passing the received signal through a synchronized demodulator to obtain the original unspread signal, narrow passband filters extract the desired signal while band reject filters extract the interference. The narrow passband outputs are applied to one correlator (the "signal correlator") that then forms estimates of $E\{A_s^* B_s\}$. The band reject outputs likewise are applied to a second correlator (the "interference correlator") that then forms estimates of $E\{A_I^* B_I\}$. A weighted combination of the outputs (signal component weighted by K_s and interference component weighted by K_I) then forms an estimate of the complex quantity $A^* \otimes B$. The ratio of K_s to K_I determines the effective value of Γ.

If the correlation product operator $\otimes$ is taken to include operations on vector quantities, then we may write

$$\mathbf{M} = \mathbf{x}^* \otimes \mathbf{x}^T \tag{7.44}$$

$$\mathbf{Z} = \mathbf{h}^* \otimes \mathbf{h}^T \tag{7.45}$$

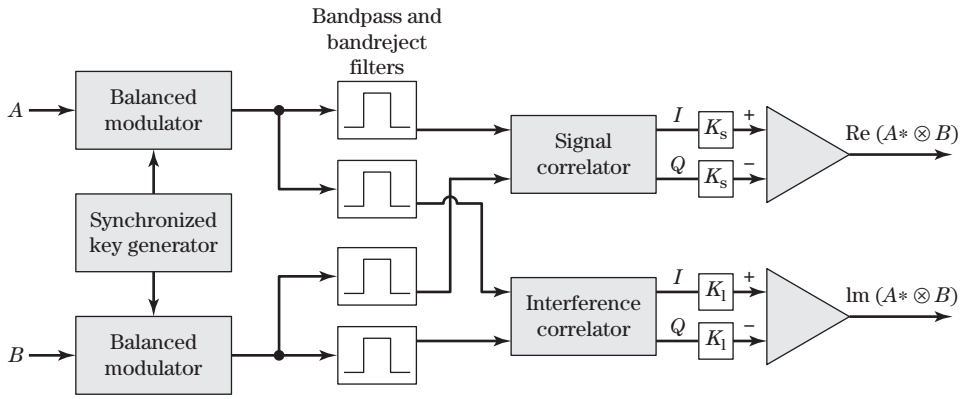


FIGURE 7-5 ■ Generalized correlator.

The generalized correlator network of Figure 7-5 provides a basis for estimating the elements of the matrix **Z** and determining whether the diagonalization of this matrix is complete.

7.2.3 Piecemeal Adjustment of the Nolen Network Stages

The piecemeal adjustment procedure for Nolen network parameters described by (7.25) and (7.26) can no longer employ the correlation products of $\overline{y_k^l v_k^{l*}}$, $\overline{(y_k^l)^2}$ and $\overline{(v_k^l)^2}$ if the modified performance measure Ω_l is to be maximized (or minimized) for the lth stage, since it is now the generalized correlation product that is related to the elements of the matrix **Z** [4]. Therefore, (7.25) and (7.26) should now be modified to yield

$$\phi_{lk} = \arg\{y_k^l \otimes v_k^{l*}\} \tag{7.46}$$

$$\psi_{lk} = \frac{1}{2}\tan^{-1}\left\{\frac{2|y_k^l \otimes v_k^{l*}|}{|y_k^l \otimes y_k^{l*}| - |v_k^l \otimes v_k^{l*}|}\right\} \tag{7.47}$$

The value of the arctangent function lying between $-\pi$ and zero radians minimizes Ω_l, whereas the value lying between zero and π radians maximizes Ω_l. The complete piecemeal adjustment of a full Nolen cascade network for an N-element array using (7.46) and (7.47) requires $N(N-1)/2$ iterations, making the practical use of a Nolen eigenvector component cascade processor less attractive when compared with the Gram–Schmidt cascade described in the next section.

7.3 | GRAM–SCHMIDT ORTHOGONALIZATION PREPROCESSOR

Section 7.1 shows that using a unitary transformation to obtain an orthogonal signal set accelerates convergence of a gradient-based algorithm by circumventing the eigenvalue spread problem. The Gram–Schmidt orthogonalization procedure [15] is another way to obtain an orthogonal signal set. While a normalized Gram–Schmidt preprocessor may be followed by a Howells–Applebaum adaptive processor to realize accelerated convergence as illustrated in Figure 7-6, canceling the interference is also possible by using the preprocessor alone in a coherent sidelobe canceller (CSLC) configuration.

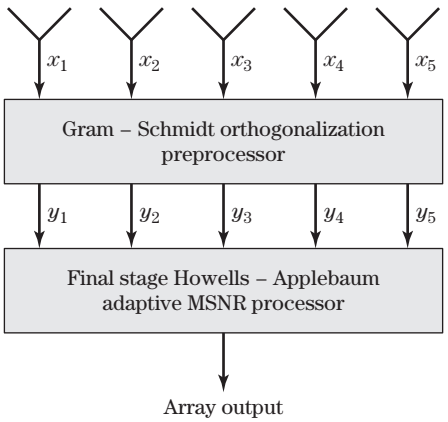


FIGURE 7-6 ■ Gram–Schmidt orthogonalization network with Howells–Applebaum adaptive processor for accelerated convergence.

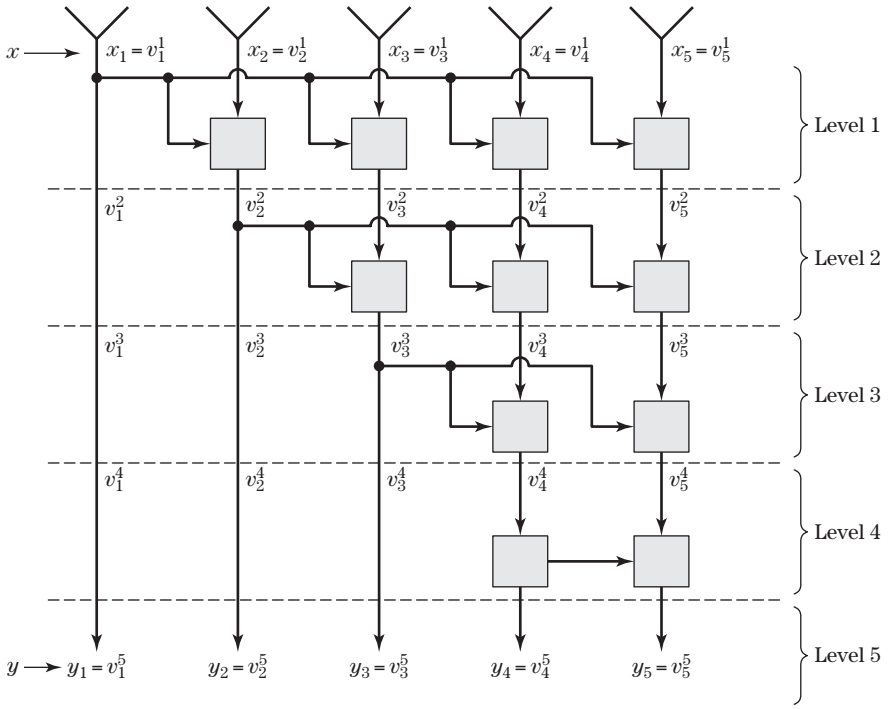


FIGURE 7-7 ■ Gram–Schmidt transformation to obtain independent variables.

The Gram–Schmidt orthogonalization cascade preprocessor is very easily implemented, has excellent transient response characteristics, and therefore presents an attractive alternative to the eigenvector component preprocessor of Section 7.1.

To understand the coordinate transformation based on the Gram–Schmidt orthogonalization procedure, consider the five-element array of Figure 7-7 in which a transformation (indicated by a square) is introduced at each node in the cascade network to achieve independence (orthogonality) between the transformed output signal and a reference input signal [5].

Each transformation in the network has two input signals $v_k^k = y_k$ (which is the reference signal for level k) and v_n^k where $n \geq k + 1$. Every transformation achieves independence between v_k^k and v_n^k. At the last level the output signals $y_1, y_2, \ldots, y_k$ are mutually independent so that $E\{y_m y_n^*\} = 0$ for all $m \neq n$. Having achieved mutual independence among the output signals, a single Howells–Applebaum loop is applied to each signal where the loop gain is high for small signals and low for large signals, thereby equalizing the time response for the various signals in the interference environment. The first level of the network provides the necessary transformation of $v_n^1, n \geq 2$, to achieve independence from $v_1^1 = y_1$ for the signal set $v_n^2, n \geq 2$. The signal set $v_n^2, n \geq 3$ is then transformed in the second level of the network to achieve independence from the signal $v_2^2 = y_2$. This process continues until a complete transformation $\mathbf{G}$ is obtained, where

$$\mathbf{y} = \mathbf{G}\mathbf{x} \tag{7.48}$$

The Gram–Schmidt transformation, $\mathbf{G}$, is found by ensuring that the appropriate transformations take place at each node indicated in Figure 7-7. Recalling from Chapter 4 that a single Howells–Applebaum adaptive loop yields an output signal that in the steady state is uncorrelated with a selected reference signal, we see that each transformation in Figure 7-7 may be realized with a single Howells–Applebaum adaptive control loop as

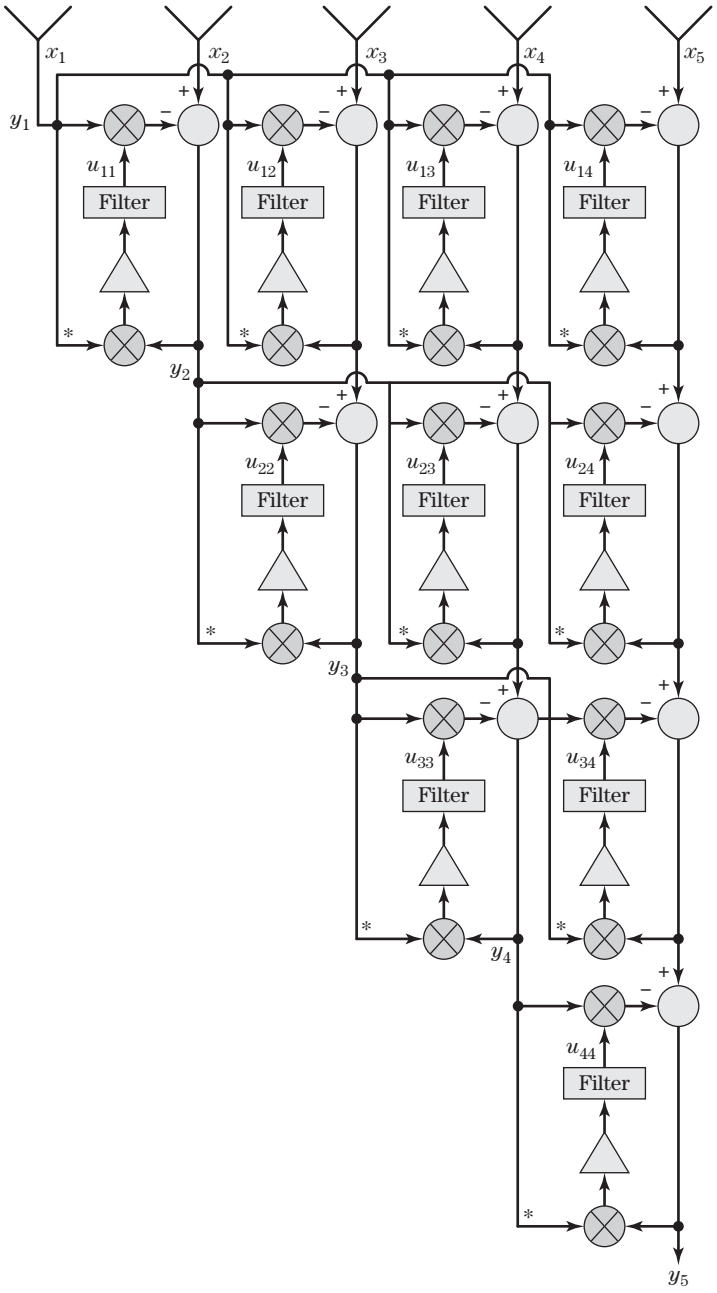


FIGURE 7-8 ■ Gram–Schmidt orthogonalization for five-element array realized with Howells–Applebaum adaptive loops.

shown in Figure 7-8. The transformation occurring at each node in Figure 7-7 (using the weight indexing of Figure 7-8) are expressed as

$$v_n^{k+1} = v_n^k - u_{k(n-1)} v_k^k, \quad k+1 \le n \le N \tag{7.49}$$

where $N =$ number of elements in the array. In the steady state, the adaptive weights have values given by

$$u_{k(n-1)} = \frac{\overline{\left(v_k^{k*} v_n^k\right)}}{\left(v_k^{k*} v_k^k\right)} \tag{7.50}$$

where the overbars denote expected values. The transformation represented by (7.48) and (7.49) is a close analog of the familiar Gram–Schmidt orthogonalization equations. For an N-element array, $N(N-1)/2$ adaptive weights produce the Gram–Schmidt orthogonalization. Since an adaptive Howells–Applebaum maximum SNR processor requires N adaptive weights, the configuration of Figure 7-6 requires a total of $N(N+1)/2$ adaptive weights.

The transformation of the input signal vector $\mathbf{x}$ into a set of independent output signals $\mathbf{y}$ is not unique. Unlike the eigenvector transformation, which yields output signals in a normal coordinate system in which the maximum power signal is the first output component, the Gram–Schmidt transformation adopts any component of $\mathbf{x}$ as the first output component and any of the remaining components of $\mathbf{x}$ as the signal component v_n^k to be transformed by way of (7.49).

The fact that the transformation network of Figure 7-7 yields a set of uncorrelated output signals suggests that this network functions in the manner of a CSLC system whose output signal (in the steady state) is uncorrelated with each of the auxiliary channel input signals. Recall from the discussion of the SNR performance measure in Chapter 3 that an $N-1$ element CSLC is equivalent to an N-element adaptive array with a generalized signal vector given by

$$\mathbf{t} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.51}$$

It is shown in what follows that by selecting x_5 of Figure 7-7 as the main beam channel signal b (so that $\mathbf{t}^T = [0, 0, \ldots, 0, 1]$) and $z = y_5$ as the output signal, then the cascade preprocessor yields an output that converges to $z = b - \mathbf{w}^T\mathbf{x}$. Here $\mathbf{x}$ is the auxiliary channel signal vector, and $\mathbf{w}$ is the column vector of auxiliary channel weights for the equivalent CSLC system of Figure 7-9 that minimizes the output noise power. The CSLC system of Figure 7-9 and the cascade preprocessor of Figure 7-7 are equivalent in the

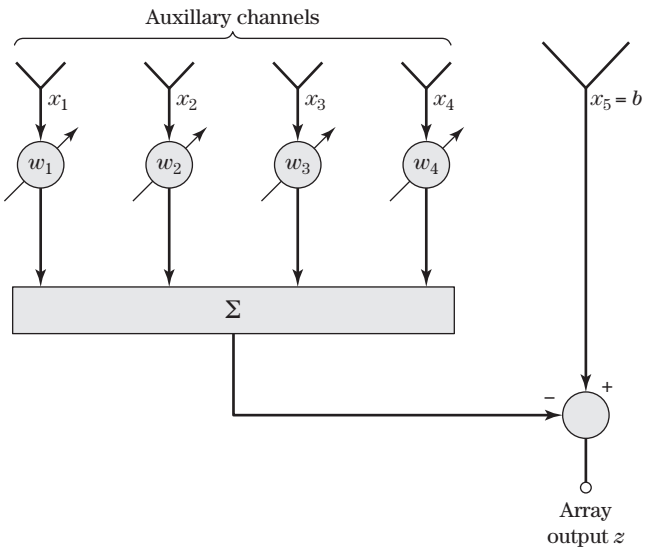


FIGURE 7-9 ■ Equivalent CSLC system for the cascade preprocessor system of Figure 7-7.

sense that the cascade network output is the same as the CSLC system output once the optimum solutions are reached. Recall from Chapter 3 that the optimum weight vector for the CSLC system is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\overline{(\mathbf{x}^* b)} \tag{7.52}$$

If the main beam channel signal is replaced by a locally generated pilot signal $p(t)$, then

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\overline{(\mathbf{x}^* p)} \tag{7.53}$$

and the array output is given by

$$\mathrm{z(t)} = \mathrm{p(t)} - \mathbf{w}^{\dagger}\mathbf{x(t)} \tag{7.54}$$

7.3.1 Convergence of the Gram–Schmidt Cascade Preprocessor

Let us first demonstrate that the cascade network of Figure 7-7 converges to the solution provided by a conventional CSLC system. Complete steady-state equivalence is shown for the three-element CSLC system of Figure 7-10 (in which two Howells–Applebaum SLC control loops yield weights w_1 and w_2) and the three-element cascade system of Figure 7-11 (in which three Howells–Applebaum control loops yield weights u_{11}, u_{12}, and u_{22}) [5]. The analysis for this case may then be easily extended to any arbitrary number of elements.

For the conventional three-element CSLC system of Figure 7-10, the steady-state solution for the optimum weight vector (whose components are w_1 and w_2) is given by (7.52), where $\mathbf{R}_{xx} = \mathbf{R}_{nn}$ in the absence of a desired signal. From (7.52) it follows that the optimum steady-state weights of Figure 7-10 are given by

$$\begin{aligned}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} &= \begin{bmatrix} \overline{x_1^* x_1} & \overline{x_1^* x_2} \\ \overline{x_2^* x_1} & \overline{x_2^* x_2} \end{bmatrix}^{-1} \begin{bmatrix} \overline{x_1^* b} \\ \overline{x_2^* b} \end{bmatrix} \\ &= \frac{\begin{bmatrix} \overline{x_2^* x_2} & \overline{-x_1^* x_2} \\ \overline{-x_2^* x_1} & \overline{x_1^* x_1} \end{bmatrix} \begin{bmatrix} \overline{x_1^* b} \\ \overline{x_2^* b} \end{bmatrix}}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})}\end{aligned} \tag{7.55}$$

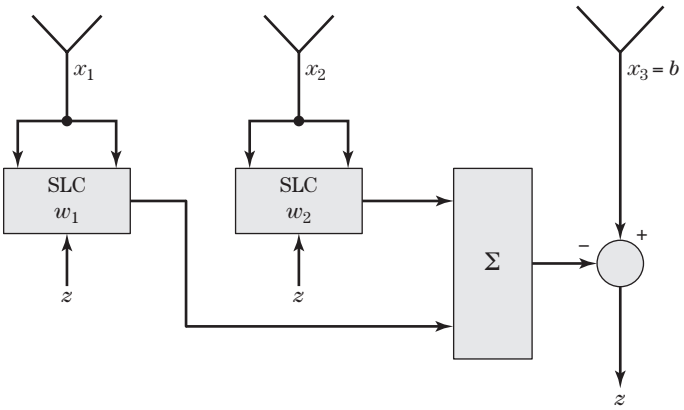


FIGURE 7-10 ■ Conventional three-element CSLC system having two Howells–Applebaum SLC control loops.

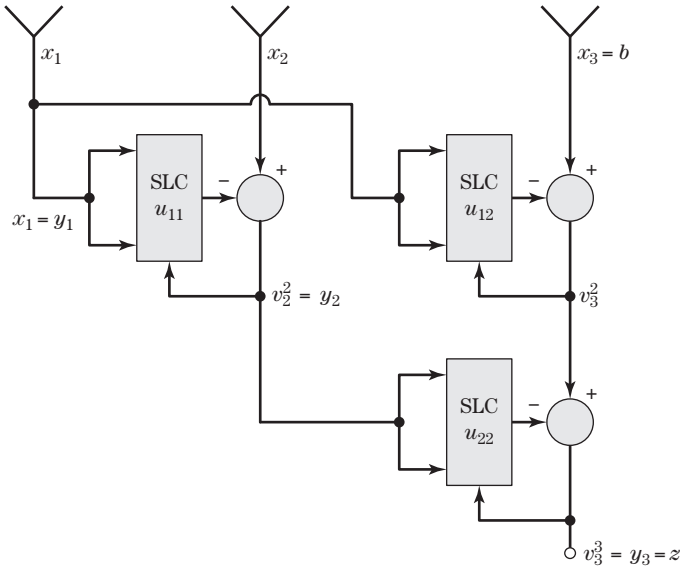


FIGURE 7-11 ■ Cascade Gram–Schmidt three-element CSLC system having three Howells–Applebaum SLC control loops.

so that

$$w_1 = \frac{(\overline{x_2^* x_2})(\overline{x_1^* b}) - (\overline{x_1^* x_2})(\overline{x_2^* b})}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})} \tag{7.56}$$

$$w_2 = \frac{(\overline{x_1^* x_1})(\overline{x_2^* b}) - (\overline{x_2^* x_1})(\overline{x_1^* b})}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})} \tag{7.57}$$

For the cascade system of Figure 7-11, it follows from (7.49) and (7.50) that

$$y_2 = x_2 - u_{11} x_1 \tag{7.58}$$

$$v_3^2 = b - u_{12} x_1 \tag{7.59}$$

$$z = v_3^2 - u_{22} y_2 \tag{7.60}$$

The weight elements have steady-state values given by

$$u_{11} = \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} \tag{7.61}$$

$$u_{12} = \frac{(\overline{x_1^* b})}{(\overline{x_1^* x_1})} \tag{7.62}$$

$$u_{22} = \frac{(\overline{y_2^* v_3^2})}{(\overline{y_2^* y_2})} \tag{7.63}$$

From (7.58) to (7.60) it follows that

$$z = b - [(u_{12} - u_{11} u_{22}) x_1 + u_{22} x_2] \tag{7.64}$$

Comparing (7.64) with Figure 7-10, we see that w_2 and w_1 fill the roles taken by u_{22} and $u_{12} - u_{11} u_{22}$, respectively, in the cascade system.

If we substitute (7.58) and (7.59) into (7.63) it then follows that

$$u_{22} = \frac{(\overline{x_1^* x_1})(\overline{x_2^* b}) - (\overline{x_2^* x_1})(\overline{x_1^* b})}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})} \tag{7.65}$$

which is identical with the solution (7.57) for w_2. Likewise, if we substitute (7.61), (7.62), and (7.63) into $u_{12} - u_{11}u_{22}$ it follows that

$$u_{12} - u_{11}u_{22} = \frac{(\overline{x_1^* b})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* b})}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})} \tag{7.66}$$

which is identical with the solution (7.56) for w_1. Therefore, the steady-state solutions reached by the two CSLC of Figures 7-10 and 7-11 are identical with weight element equivalences given by

$$\left.\begin{aligned} w_2 &= u_{22} \\ w_1 &= u_{12} - u_{11}u_{22} \end{aligned}\right\} \tag{7.67}$$

Additional weight element equivalences may easily be obtained for an arbitrary number of sensor elements by considering the transformation (7.48). By induction, the transformation $\mathbf{G}$ is a lower triangular matrix whose elements g_{ij} are given by the following iterative relationships:

$$g_{ij} = 0 \quad \text{if } i < j \tag{7.68}$$

$$g_{ij} = 1 \quad \text{if } i = j \tag{7.69}$$

$$g_{ij} = -\sum_{k=j}^{i-1} u_{jk}^* \cdot g_{i(k+1)} \quad \text{for } i > j \tag{7.70}$$

The jth row of the transformation matrix $\mathbf{G}$ contains the desired weight equivalences for a j-element CSLC system where the auxiliary channel weights are given by

$$\left.\begin{aligned} w_{j-1} &= -g_{j(j-1)} \\ w_{j-2} &= -g_{j(j-2)} \\ &\vdots \\ w_1 &= -g_{j1} \end{aligned}\right\} \tag{7.71}$$

Having found the steady-state weight element equivalence relationships, it is now appropriate to consider the transient response of the networks of Figures 7-10 and 7-11. The transient responses of the two CSLC systems under consideration are investigated by examining the system response to discrete signal samples of the main beam and the auxiliary channel outputs. The following analysis assumes that the adaptive weights reach their steady-state expected values on each iteration and therefore ignores errors that would be present due to loop noise. Let x_{kn} denote the nth signal sample for the kth element channel. The main beam channel samples are denoted by b_n. For a system having N auxiliary channels and one main beam channel, after N independent samples have been collected, a set of auxiliary channel weights are computed using

$$\sum_{k=1}^{N} w_k x_{kn} = b_n \quad \text{for } n = 1, 2, \ldots, N \tag{7.72}$$

The foregoing system of equations yields a unique solution since the matrix defined by the signal samples x_{kn} is nonsingular provided that either thermal receiver noise or N directional interference sources are present. Equation (7.72) is closely related to (7.52), since multiplying both sides of (7.72) by x_{mn}^* and summing over the index n yields

$$\sum_{n=1}^{N}\sum_{k=1}^{N} w_k x_{mn}^* x_{kn} = \sum_{n=1}^{N} x_{mn}^* b_n \tag{7.73}$$

which are written in matrix form as

$$\hat{\mathbf{R}}_{xx}\mathbf{w} = \overset{\wedge}{(\mathbf{x}^*\mathbf{b})} \tag{7.74}$$

Comparing (7.74) with (7.52) reveals that these two equations yield similar solutions for the weight vector $\mathbf{w}$ in a stationary signal environment.

The iterative weight correction procedure for a single Howells–Applebaum SLC loop is modeled as shown in Figure 7-12. On receipt of the ith signal sample, the resulting change in the weight $u_{k(n-1)}$ is computed by applying (7.50) and assuming that the signal samples are approximately equal to their expected values to yield

$$\Delta u_{k(n-1)}(i) = \frac{v_k^{k*}[v_n^k(i) - u_{k(n-1)}(i-1)\cdot v_k^k(i)]}{(v_k^{k*}v_k^k)} \tag{7.75}$$

The element weight value is then updated in accordance with

$$u_{k(n-1)}(i) = u_{k(n-1)}(i-1) + \Delta u_{k(n-1)}(i) \tag{7.76}$$

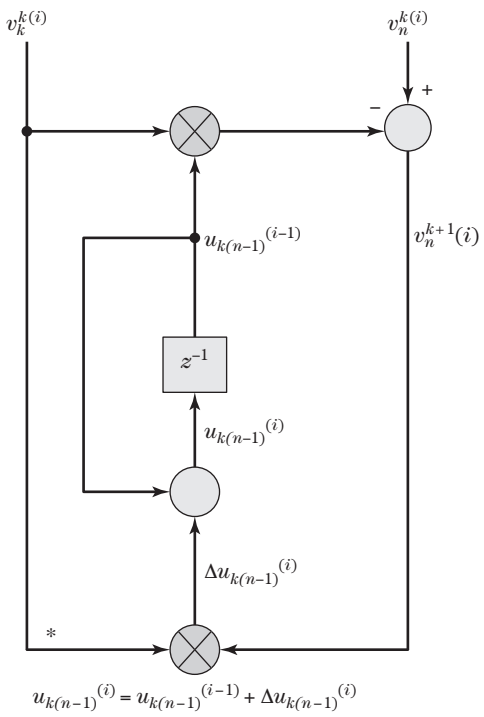


FIGURE 7-12 ■ Iterative weight correction model for Howells–Applebaum SLC loops.

With all weights in the cascade Gram–Schmidt network initially set to zero, it follows that on receipt of the first signal sample $v_2^2 = x_{21}$, and $v_3^2 = v_3^3 = b_1$ so the change in weight settings after the first iteration results in

$$u_{11}(1) = \frac{x_{11}^* v_2^1}{|x_{11}|^2} = \frac{x_{11}^* x_{21}}{|x_{11}|^2} \tag{7.77}$$

$$u_{12}(1) = \frac{x_{11}^* v_3^1}{|x_{11}|^2} = \frac{x_{11}^* b_1}{|x_{11}|^2} \tag{7.78}$$

and

$$u_{22}(1) = \frac{v_2^{2*} v_3^2}{|v_2^2|^2} = \frac{x_{21}^* b_1}{|x_{21}|^2} \tag{7.79}$$

On receipt of the second signal sample $v_2^2 = x_{22} - u_{11}(1)x_{12}$, $v_3^2 = b_2 - u_{12}(1)x_{12}$, and $v_3^3 = v_3^2 - u_{22}(1)v_2^2$ so that

$$\Delta u_{11}(2) = \frac{x_{12}^*[x_{22} - u_{11}(1)x_{12}]}{|x_{12}|^2} \tag{7.80}$$

$$\Delta u_{12}(2) = \frac{x_{12}^*[b_2 - u_{12}(1)x_{12}]}{|x_{12}|^2} \tag{7.81}$$

$$\Delta u_{22}(2) = \frac{(x_{22} - u_{11}(1)x_{12}) * [b_2 - u_{12}(1)x_{12} - u_{22}(1)(x_{22} - u_{11}(1)x_{12})]}{|x_{22} - u_{11}(1)x_{12}|^2} \tag{7.82}$$

Substitute (7.77) to (7.79) into (7.80) to (7.82), it then follows that

$$u_{11}(2) = u_{11}(1) + \Delta u_{11}(2) = \frac{x_{12}^* x_{22}}{|x_{12}|^2} \tag{7.83}$$

$$u_{12}(2) = u_{12}(1) + \Delta u_{12}(2) = \frac{x_{12}^* b_2}{|x_{12}|^2} \tag{7.84}$$

$$u_{22}(2) = u_{22}(1) + \Delta u_{22}(2) = \frac{x_{11} b_2 - x_{12} b_1}{x_{11} x_{22} - x_{12} x_{21}} \tag{7.85}$$

By use of the weight equivalence relationships of (7.67) it follows that $w_2(2) = u_{22}(2)$ and

$$w_1(2) = u_{12}(2) - u_{11}(2)u_{22}(2) = \frac{x_{22} b_1 - x_{21} b_2}{x_{11} x_{22} - x_{12} x_{21}} \tag{7.86}$$

Note, however, that $w_1(2)$ and $w_2(2)$ of (7.85) and (7.86) are the weights that satisfy (7.72) when $N = 2$. Therefore, for a two-auxiliary channel CSLC, the cascade Gram–Schmidt network converges to a set of near-optimum weights in only two iterations. The foregoing analysis can also be carried out for the case of an N-auxiliary channel CSLC and leads to the conclusion that convergence to a set of near-optimum weights in the noise-free case occurs after N iterations. Averaging the input signals mitigates the effects of noise-induced errors.

Note that using (7.75) to update the weight settings results in values for u_{11} and u_{12} (the weights in the first level) that depend only on the current signal sample. The update setting for u_{22}, however, depends on the last two signal samples. Therefore, an N-level cascade Gram–Schmidt network always updates the weight settings based on the N preceding signal samples.

The conventional two-weight element CSLC does not converge to a near-optimum weight solution with only two signal samples. To see this result note that for the configuration of Figure 7-10 after receipt of the first signal sample that

$$w_1(1) = \frac{x_{11}^* b_1}{|x_{11}|^2} \tag{7.87}$$

$$w_2(1) = \frac{x_{21}^* b_1}{|x_{21}|^2} \tag{7.88}$$

Following the second signal sample

$$w_1(2) = w_1(1) + \frac{x_{12}^*(b_2 - w_1(1)x_{12} - w_2(1)x_{22})}{|x_{12}|^2} \tag{7.89}$$

$$w_2(2) = w_2(1) + \frac{x_{22}^*(b_2 - w_1(1)x_{12} - w_2(1)x_{22})}{|x_{12}|^2} \tag{7.90}$$

The weights given by (7.89) and (7.90) are not the same as the weights given by (7.85) and (7.86) and consequently do not satisfy (7.72). Simulation results show that the configuration of Figure 7-10 yields a transient response that is considerably slower (even with a higher level of loop noise) than the transient response of a Gram–Schmidt cascade preprocessor.

7.4 SIMULATION RESULTS

A four-element network configuration like that of Figure 7-6 was simulated with a digital LMS adaptive loop used in place of each Howells–Applebaum loop in the final maximum SNR stage. The four loop gains of the LMS adaptive loops in the final maximum SNR stage of the selected configuration were set inversely proportional to an estimate of the input signal power to each loop $\hat{p}_i$ obtained by averaging the squared loop input signal over K samples so that $\Delta_{s_i} = \alpha_L/\hat{p}_i$. Likewise, the loop gain of the digital version of each Howells–Applebaum loop in the cascade preprocessor was selected as $G_i = \alpha/\hat{p}_i$. Selecting the adaptive loop gains in this manner equalizes the transient response time to the signals $y_1, \ldots, y_4$. Furthermore, the correlations (products) performed in the analog processor of Figure 7-8 are represented in the digital version as products of individual sample values averaged over K samples. The averaging interval also permits signal power estimates to be obtained by averaging the instantaneous signal powers appearing at the loop inputs of the final processing stage. The array configuration and signal environment correspond to that given in Figure 7-4, and the desired signal was again selected to be a biphase modulated signal having a 0° or 180° reference phase equally likely at each sample.

The algorithm performance in each case is given in terms of the resulting output SNR versus the number of iterations. The signal environment conditions are specified to represent two values eigenvalue spread: $\lambda_{\max}/\lambda_{\min} = 16{,}700$ and 2,440. The convergence results given in Figures 7-13 to 7-17 are for an eigenvalue spread of 16,700 and five values of α_L ranging over 0.1, 0.25, 0.3, 0.5, and 0.65. Likewise the convergence results displayed in Figures 7-18 to 7-20 are for An eigenvalue spread of 2,440 and various conditions for the gain, K, and the initial weight vector setting. All other conditions are specified by the

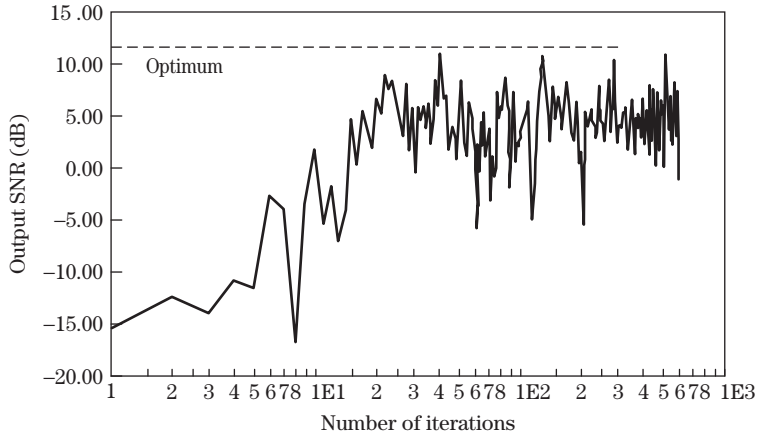


FIGURE 7-13 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 16,700. Algorithm parameters are $K = 3, \alpha = \alpha_L = 0.5$, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$ (11.76 dB).

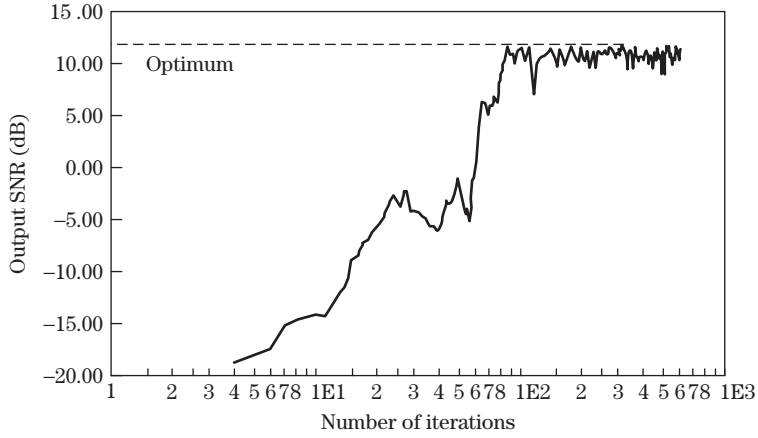


FIGURE 7-14 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 16,700. Algorithm parameters are $K = 3, \alpha = \alpha_L = 0.1$, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

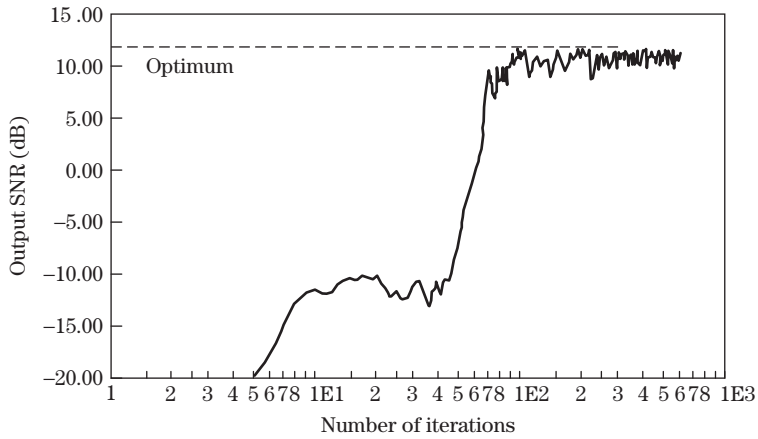


FIGURE 7-15 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 16,700. Algorithm parameters are $K = 9, \alpha = 0.3, \alpha_L = 0.25\alpha$, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

number of measurement samples averaged per iteration K, the Howells–Applebaum loop gain α, the LMS loop gain α_L, the desired signal-to-thermal noise ratio s/n, and the initial weight vector selection $\mathbf{w}(0)$. For $s/n = 10$, the optimum output SNR is $\text{SNR}_{\text{opt}} = 15(11.76 \text{ dB})$, whereas for $s/n = 0.1$ the optimum output SNR is $\text{SNR}_{\text{opt}} = 0.15(-8.2 \text{ dB})$.

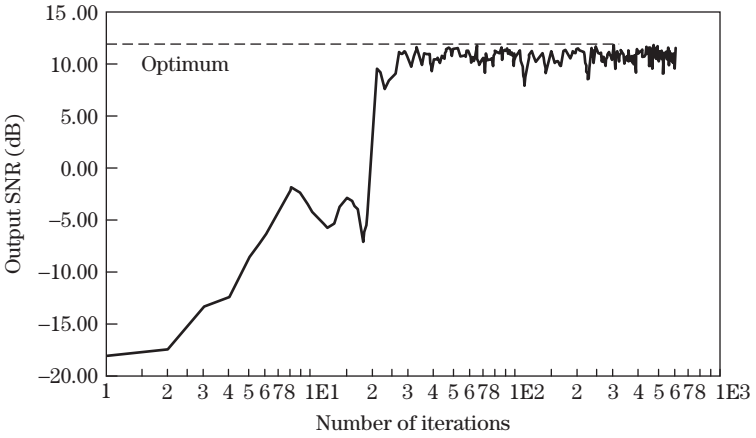


FIGURE 7-16 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 16,700. Algorithm parameters are $K = 9, \alpha = \alpha_L = 0.3$, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

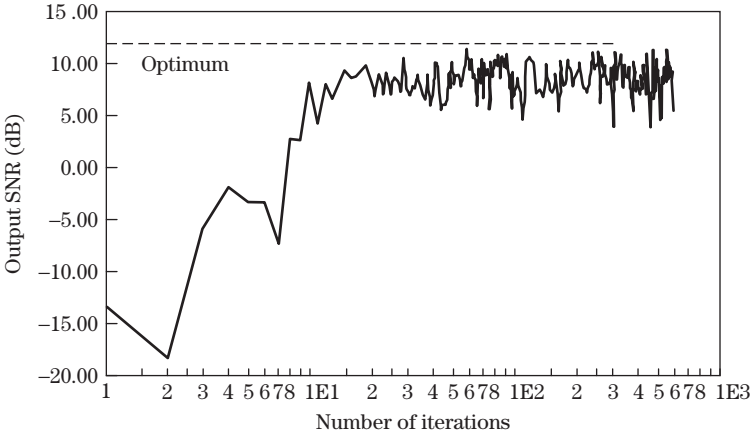


FIGURE 7-17 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 16,700. Algorithm parameters are $K = 9, \alpha = \alpha_L = 0.65$, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

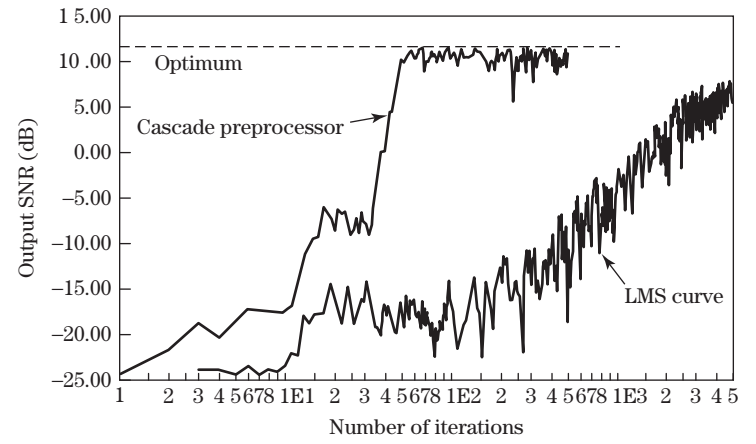


FIGURE 7-18 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor and LMS processor with eigenvalue spread = 2,440. Algorithm parameters are $K = 3, \alpha = \alpha_L = 0.1$, and $\mathbf{w}^T(0) = [1, 0, 0, 0]$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

Observations from Figures 7-13 to 7-20 include the following:

1. A large spread in covariance matrix eigenvalues has little effect on the speed of the algorithm response (compare Figure 7-14 with Figure 7-20).
2. Comparable steady-state output variations (reflecting the same level of loop noise) are obtained for $K = 3, \alpha = 0.3/p$ and $K = 9, \alpha = 0.1/p$, thereby indicating that the product $K \cdot \alpha$ determines the loop noise level (compare Figure 7-14 with Figure 7-15).

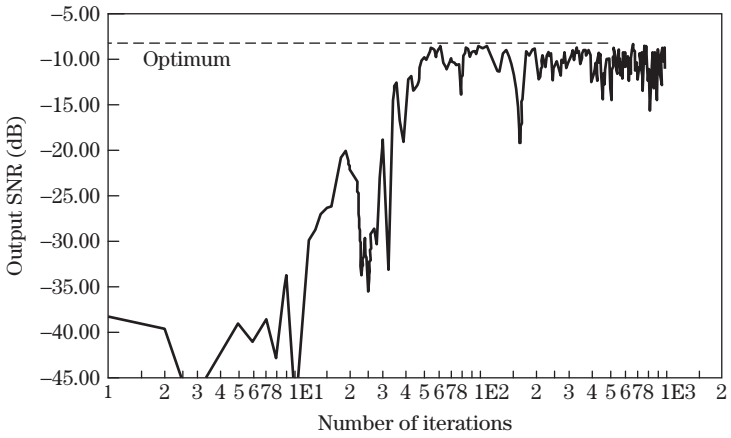


FIGURE 7-19 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 2,440. Algorithm parameters are $K = 3, \alpha = \alpha_L =$ 0.1, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 0.1$ for which $\text{SNR}_{\text{opt}} =$ 0.15(−8.2 dB).

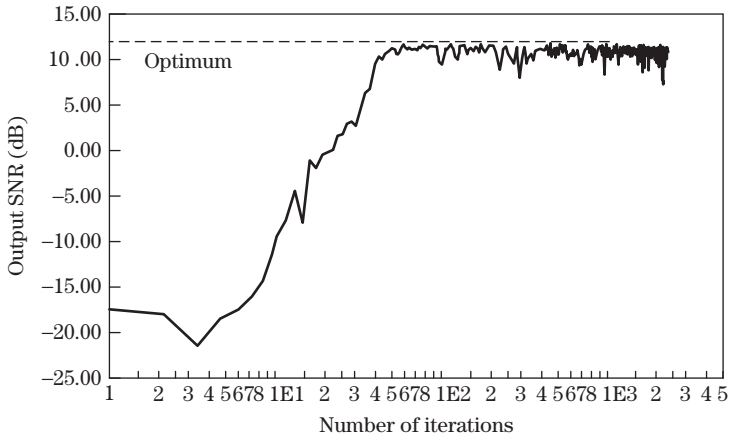


FIGURE 7-20 ■ Output SNR versus number of iterations for Gram–Schmidt cascade preprocessor with eigenvalue spread = 2,440. Algorithm parameters are $K = 3, \alpha = \alpha_L =$ 0.1, and $\mathbf{w}(0) = \mathbf{0}$ with $s/n = 10$ for which $\text{SNR}_{\text{opt}} = 15$.

3. Only a slight increase in the loop gain α causes the weights to become excessively noisy (compare Figure 7-16 with Figure 7-13 and Figure 7-17).
4. The appropriate value to which the product $K \cdot \alpha$ should be set for an acceptable level of loop noise depends on the value of SNR_{opt} (compare Figures 7-19 and 7-20). Smaller values of SNR_{opt} require smaller values of the product $K \cdot \alpha$ to maintain the same loop noise level. It may also be seen that the adaptation time required with these parameter values for the Gram–Schmidt orthogonalization preprocessor is greater than that required for either a recursive or direct matrix inversion (DMI) algorithm. The question of relative adaptation times for the various algorithms is pursued further in Chapter 10.
5. The degree of transient response improvement that are obtained with a cascade preprocessor is shown in Figure 7-18 where the Gram–Schmidt cascade preprocessor response is compared with the LMS processor response. The LMS curve in this figure was obtained simply by removing the cascade preprocessing stage in Figure 7-6, thereby leaving the final maximum SNR stage consisting of four LMS adaptive loops. The greater the eigenvalue spread of the $\mathbf{R}_{xx}$ matrix, then the greater is the degree of improvement that are realized with a cascade preprocessor compared with the LMS processor.

7.5 SUMMARY AND CONCLUSIONS

The sensitivity of the convergence rate of the LMS and Howells–Applebaum adaptive processors to eigenvalue spread are greatly reduced by using a cascade preprocessor that resolves the input signal vector into orthogonal components. A Nolen cascade network resolves the input signal vector into orthogonal eigenvector beam components but requires an excessive number of iterations to adjust the network parameters before rapid convergence is achieved. The Gram–Schmidt cascade preprocessor resolves the input signal vector into orthogonal components (although these components are not eigenvector beam components) and succeeds in greatly reducing the sensitivity of the convergence rate to eigenvalue spread while requiring a relatively small number of iterations to adjust the preprocessor parameters to the desired steady-state values. The Gram–Schmidt preprocessor therefore is a highly attractive candidate for achieving fast algorithm convergence rates while retaining the implementation simplicity associated with the LMS and Howells–Applebaum adaptive processors.

7.6 PROBLEMS

Problems 1 through 12 all concern Cascade SLC Control Loops

1. Consider the cascade control loop configuration of Figure 7-21.

(a) Show that the weight equivalences between the configuration of Figure 7-21 and the standard CSLC configuration are given by

$$w_1 = u_1 + u_3 \quad w_2 = -u_2 u_3$$

(b) Show that the steady-state weight values of Figure 7-21 are given by

$$u_1 = \frac{\overline{x_1^* b}}{\overline{x_1^* x_1}} \quad u_2 = \frac{\overline{x_2^* x_1}}{\overline{x_2^* x_2}}$$

$$u_3 = \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} \left[\frac{(\overline{x_1^* x_1})(\overline{x_2^* b}) - (\overline{x_2^* x_1})(\overline{x_1^* b})}{(\overline{x_2^* x_2})(\overline{x_1^* x_1}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})} \right]$$

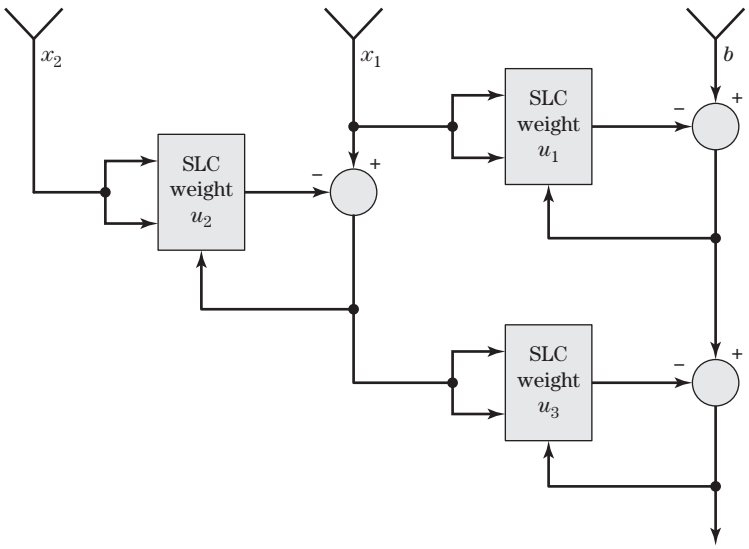


FIGURE 7-21 ■ Cascade arrangement of three Howells–Applebaum SLC control loops yielding convergence to incorrect solution.

Note that these steady-state weights do not correspond to the correct solution for the standard CSLC configuration.

Note: Problems 2 through 12 concern derivations that may be found in [7].

2. Consider the three-element array cascade configuration of Figure 7-11.

(a) Show that the steady-state weight values for this configuration are given by (7.61), (7.62), and (7.65).

(b) Using (7.64) for the array output, show that

$$z = x_3 - u_{22}x_2 - \frac{(\overline{x_1^* x_3})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_3})}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_2^* x_1})(\overline{x_1^* x_2})} x_1$$

(c) The average noise output power is defined by $N_0 = |z|^2 = \overline{zz^*}$. For notational simplicity let $u_1 = u_{11}$, $u_2 = u_{12}$, $u_3 = u_{22}$, and show that

$$\begin{aligned} N_0 = {} & (\overline{x_3^* x_3}) + |\overline{u}_3|^2(\overline{x_2^* x_2}) + |\overline{u}_2 - \overline{u}_1\overline{u}_3|^2(\overline{x_1^* x_1}) \\ & - \overline{u}_3(\overline{x_3^* x_2}) - \overline{u_3^*}(\overline{x_2^* x_3}) \\ & - (\overline{u}_2 - \overline{u}_1\overline{u}_3)(\overline{x_3^* x_1}) - (\overline{u_2^*} - \overline{u_1^* u_3^*})(\overline{x_1^* x_3}) \\ & + \overline{u}_3(\overline{u_2^*} - \overline{u_1^* u_3^*})(\overline{x_1^* x_2}) + \overline{u_3^*}(\overline{u}_2 - \overline{u}_1\overline{u}_3)(\overline{x_2^* x_1}) \end{aligned}$$

3. Using the weight notation of Problem 2(c) for the configuration of Figure 7-11, let

$$u_n = \overline{u}_n + \delta_n$$

where δ_n represents the fluctuation component of the weight. The total output noise power is then given by

$$\begin{aligned} N_{0_{\text{TOT}}} &= N_0 + N_u \\ &= |\overline{x_3 - (\overline{u}_3 + \delta_3)x_2} - \overline{[(\overline{u}_2 + \delta_2) - (\overline{u}_1 + \delta_1)(\overline{u}_2 + \delta_3)]x_1}|^2 \end{aligned}$$

where N_u represents the excess weight noise due to the fluctuation components. Note that $\overline{\delta}_n = 0$ and neglect third- and fourth-order terms in δ_n to show that

$$N_u = |\overline{\delta_3}|^2[(\overline{x_2^* x_2}) - (\overline{x_2^* x_1})\overline{u}_1] + (\overline{x_1^* x_1})|\overline{\delta_2 - \overline{u}_3\delta_1}|^2$$

4. The weight $u_1 = u_{11}$ of Figure 7-11 satisfies the differential equation

$$\tau_1\left(\frac{\dot{u}_1}{G_1}\right) + \left(\frac{u_1}{G_1}\right) = x_1^*(x_2 - u_1 x_1)$$

where G_1 denotes amplifier gain, and τ_1 denotes time constant of the integrating filter in the Howells–Applebaum control loop for u_1. The foregoing equation is rewritten as

$$\frac{1}{\alpha_1}\dot{u}_1 + \left(x_1^* x_1 + \frac{1}{G_1}\right)u_1 = x_1^* x_2 \quad \text{where } \alpha_1 = \frac{G_1}{\tau_1}$$

so that

$$\frac{1}{\alpha_1}\dot{\overline{u}}_1 + \left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)\overline{u}_1 = \overline{x_1^* x_2}$$

Subtracting the mean value differential equation from the instantaneous value differential equation and recalling that $u_1 = \overline{u}_1 + \delta_1$ then yields

$$\frac{1}{\alpha_1}\dot{\delta}_1 + \left(\frac{1}{G_1} + \overline{x_1^* x_1}\right)\delta_1 + (x_1^* x_1 - \overline{x_1^* x_1})u_1 = (x_1^* x_2 - \overline{x_1^* x_2})$$

provided that the second-order term $\delta_1(x_1^* x_1 - \overline{x_1^* x_1})$ is ignored. In the steady state where

$$\overline{u}_1 = \frac{\overline{x_1^* x_2}}{\overline{x_1^* x_1}}$$

show that

$$\frac{1}{\alpha_1}\dot{\delta}_1 + \left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)\delta_1 = \left(x_1^* x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})}x_1^* x_1\right) = \frac{1}{\alpha_1}f_1(t)$$

where $f_1(t)$ may be regarded as a random variable.

5. The solution of the differential equation obtained in Problem 4 is given by

$$\delta_1(t) = \int_0^t f_1(\tau)\exp\left[-\alpha_1\left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)(t-\tau)\right]d\tau$$

so that

$$\dot{\delta}_1(t) = f_1(t) - \alpha_1\left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)\delta_1(t)$$

Substitute $\dot{\delta}_1(t)$ into the differential equation for $\dot{\delta}_1(t)$ obtained in Problem 4, and show that this reduces the resulting expression to an identity, thereby proving that $\delta_1(t)$ is indeed a solution to the original differential equation.

6. The second moment of the fluctuation δ_1 is given by $\overline{\delta_1^* \delta_1}$.

(a) Use the expression for $\delta_1(t)$ developed in Problem 5 to show that

$$\overline{\delta_1^* \delta_1} = \exp\left[-2\alpha_1\left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)\right]\int_0^t\int_0^t \overline{f_1^*(\tau) f_1(u)}\exp\left[\alpha_1\left(\overline{x_1^* x_1} + \frac{1}{G_1}\right)(\tau + u)\right]d\tau d\mathrm{u}$$

(b) Assume that the random variable

$$f_1(t) = \alpha_1\left[x_1^* x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})}x_1^* x_1\right]$$

has a correlation interval denoted by ε, such that values of $f_1(t)$ separated by more than ε are independent. Let

$$\int_0^t \overline{f_1^*(\tau) f_1(u)}\,du = \overline{f_1^* f_1 \varepsilon} \quad \text{for } t > \tau + \varepsilon.$$

Show that $\overline{\delta_1^* \delta_1}$ then reduces to

$$\overline{\delta_1^* \delta_1} = \frac{\overline{\varepsilon f_1^* f_1}}{2\alpha_1[(\overline{x_1^* x_1}) + 1/G_1]} \quad \text{for } t > \tau + \varepsilon$$

7. Show that the second moment of f_1 defined in Problem 6 (b) is given by

$$\begin{aligned} f_1^* f_1 &= \alpha_1^2 \overline{\left[x_1^* x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} x_1^* x_1 \right]} \\ &\quad \times \overline{\left[x_1 x_2^* - \frac{(\overline{x_2^* x_1})}{(\overline{x_1^* x_1})} x_1^* x_1 \right]} \\ &= \alpha_1^2 [(\overline{x_1^* x_1})(\overline{x_1^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})] \end{aligned}$$

8. Substitute $\overline{f_1^* f_1}$ obtained in Problem 7 into $\overline{\delta_1^* \delta_1}$ obtained in Problem 6 (b) to show that

$$\overline{\delta_1^* \delta_1} = \frac{\varepsilon \alpha_1 [(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_2^* x_1})(\overline{x_1^* x_2})]}{2[(\overline{x_1^* x_1}) + 1/G_1]}$$

Note that to obtain $\overline{\delta_2^* \delta_2}$, merely replace x_2 with x_3, α_1 by α_2, and G_1 by G_2 in the previous expression.

9. From Figure 7-11 it may be seen that the inputs to $u_3 (u_{22}$ of the figure) are y_1 and y_2. Consequently, replace x_1 by y_2, x_2 by y_1, and G_1 by G_3 in the expression obtained in Problem 8. This replacement requires the second moments $\overline{y_1^* y_1}$ and $\overline{y_2^* y_2}$ and $\overline{y_2^* y_1}$ to be computed. Show that

$$\overline{y_1^* y_1} = |\overline{x_3 - \overline{u}_3 x_1}|^2 = (\overline{x_3^* x_3}) - \frac{(\overline{x_1^* x_3})(\overline{x_3^* x_1})}{(\overline{x_1^* x_1})}$$

and

$$\begin{aligned} y_2^* y_2 = |\overline{x_2 - \overline{u}_1 x_1}|^2 &= \left| \overline{x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} x_1} \right|^2 \\ &= (\overline{x_2^* x_2}) - \frac{(\overline{x_1^* x_2})(\overline{x_2^* x_1})}{(\overline{x_1^* x_1})} \end{aligned}$$

Furthermore

$$\begin{aligned} \overline{y_2^* y_1} &= \overline{\left[x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} x_1 \right]^* \left[x_3 - \frac{(\overline{x_1^* x_3})}{(\overline{x_1^* x_1})} x_1 \right]} \\ &= (\overline{x_2^* x_3}) - \frac{(\overline{x_2^* x_1})(\overline{x_1^* x_3})}{(\overline{x_1^* x_1})} \end{aligned}$$

10. The computation of N_u found in Problem 3 requires the second moment $\overline{\delta_2^* \delta_1}$. Corresponding to the expression for $\delta_1(t)$ in Problem 5 we may write

$$\delta_2(t) = \int_0^t f_2(\tau) \exp -\alpha_2 \left(\overline{x_1^* x_1} + \frac{1}{G_2} \right) (t - \tau) d\tau$$

where

$$f_2(t) = \alpha_2 \left[x_1^* x_3 - \frac{(\overline{x_1^* x_3})}{(\overline{x_1^* x_1})} x_1^* x_1 \right]$$

With the previous results, show that

$$\begin{aligned} \overline{\delta_2^* \delta_1} = \int_0^t \int_0^t & \overline{f_2^*(\tau) f_1(u)} \\ & \cdot \exp \left[-\alpha_2 \left(\overline{x_1^* x_1} + \frac{1}{G_2} \right) (t - \tau) - \alpha_1 \left(\overline{x_1^* x_1} + \frac{1}{G_1} \right) (t - u) d\tau du \right. \end{aligned}$$

Now assuming that

$$\int_0^t \overline{f_2^*(\tau) f_1(u)}\, du = \overline{f_2^* f_1}$$

show that

$$\overline{\delta_2^* \delta_1} = \frac{\varepsilon \overline{f_2^* f_1}}{[\alpha_2(\overline{x_1^* x_1}) + 1/G_2] + \alpha_1[(\overline{x_1^* x_1}) + 1/G_1]}$$

11. Evaluate $\overline{f_2^* f_1}$ by retaining only terms of the form $\overline{x_m^* x_n}$ in the expansion

$$\overline{f_2^* f_1} = \alpha_2\alpha_1 \left[x_1 x_3^* - \frac{(\overline{x_3^* x_1})}{(\overline{x_1^* x_1})} x_1^* x_1\right]\left[x_1^* x_2 - \frac{(\overline{x_1^* x_2})}{(\overline{x_1^* x_1})} x_1^* x_1\right]$$

show that the resulting expression for $\overline{f_2^* f_1}$ is given by

$$\overline{f_2^* f_1} = \alpha_2\alpha_1[(\overline{x_1^* x_1})(\overline{x_3^* x_2}) - (\overline{x_3^* x_1})(\overline{x_1^* x_2})]$$

so that

$$\overline{\delta_2^* \delta_1} = \frac{\varepsilon\alpha_1\alpha_2[(\overline{x_1^* x_1})(\overline{x_3^* x_2}) - (\overline{x_3^* x_1})(\overline{x_1^* x_2})]}{\alpha_2[(\overline{x_1^* x_1}) + 1/G_2] + \alpha_1[(\overline{x_1^* x_1}) + 1/G_1]}$$

12. Substitute the results obtained in Problem 11 into the expression for N_u, assuming that $\alpha_1 = \alpha_2$ and neglecting all $1/G$ terms, show that

$$N_u = \{[(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_1^* x_2})(\overline{x_2^* x_1})][(\overline{x_1^* x_1})(\overline{x_3^* x_3}) - (\overline{x_3^* x_1})(\overline{x_1^* x_2})] - |(\overline{x_1^* x_1})(\overline{x_2^* x_3}) - (\overline{x_2^* x_1})(\overline{x_1^* x_2})|^2\} \left[\frac{\alpha_3\varepsilon}{2(\overline{x_1^* x_1})^2} + \frac{\alpha_1\varepsilon}{2(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_2^* x_1})(\overline{x_1^* x_2})}\right]$$

Note that the portion of N_u proportional to α_3 is the component due to fluctuations in u_3. The remaining part of N_u that is proportional to α_1 is the contribution due to fluctuations in both u_1 and u_2. If τ_1, τ_2, and τ_3 are adjusted so that all three circuits have the same effective time constant,

$$\alpha_1 = \alpha_2 = \alpha_3 \frac{(\overline{x_1^* x_1})^2}{(\overline{x_1^* x_1})(\overline{x_2^* x_2}) - (\overline{x_2^* x_1})(\overline{x_1^* x_2})}$$

then the previous expression for N_u shows that the control loop noise contribution due to fluctuations in u_3 is just equal to the sum of the contributions from fluctuations in u_1 and u_2.

7.7 REFERENCES

[1] L. Hasdorff, *Gradient Optimization and Nonlinear Control*, New York, Wiley, 1976, Ch. 3.

[2] W. D. White, "Accelerated Convergence Techniques," *Proceedings of the 1974 Adaptive Antenna Systems Workshop*, March 11–13, Vol. 1, Naval Research Laboratory, Washington, DC, pp. 171–215.

[3] W. D. White, "Cascade Preprocessors for Adaptive Antennas," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 670–684.

[4] W. D. White, "Adaptive Cascade Networks for Deep Nulling," *IEEE Trans. Antennas Propag.*, Vol. AP-26, No. 3, May 1978, pp. 396–402.

[5] L. E. Brennan, J. D. Mallett, and I. S. Reed, "Convergence Rate in Adaptive Arrays," Technology Service Corporation Report No. TSC-PD-A177-2, July 15, 1977.

[6] R. C. Davis, "Convergence Rate in Adaptive Arrays," Technology Service Corporation Report No. TSC-PD-A177-3, October 18, 1977.

[7] L. E. Brennan and I. S. Reed, "Convergence Rate in Adaptive Arrays," Technology Service Corporation Report No. TSC-PD-177-4, January 13, 1978.

[8] C. C. Ko, "On the Performance of Adaptive Gram–Schmidt Algorithm for Interference Cancelling Arrays," *IEEE Transactions on Antennas and Propagation*, Vol. 39, No. 4, 1991, pp. 505–511.

[9] H. Liu, A. Ghafoor, and P. H. Stockmann, "Application of Gram–Schmidt Algorithm to Fully Adaptive Arrays," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 28, No. 2, 1992, pp. 324–334.

[10] R. W. Jenkins and K. W. Moreland, "A Comparison of the Eigenvector Weighting and Gram–Schmidt Adaptive Antenna Techniques," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 2, 1993, pp. 568–575.

[11] C. C. Ko, "Simplified Gram–Schmidt Preprocessor for Broadband Tapped Delay-Line Adaptive Array," *IEE Proceedings G Circuits, Devices and Systems*, Vol. 136, No. 3, 1989, pp. 141–149.

[12] W. F. Gabriel, "Adaptive Array Constraint Optimization," Program and Digest 1972 G-AP International Symposium, December 11–14, 1972, pp. 4–7.

[13] J. C. Nolen, "Synthesis of Multiple Beam Networks for Arbitrary Illuminations," Bendix Corporation, Radio Division, Baltimore, MD, April 21, 1960.

[14] N.J.G. Fonseca, "Printed S-Band 4 4 Nolen Matrix for Multiple Beam Antenna Applications," *IEEE Transactions on Antennas and Propagation*, Vol. 57, No. 6, June 2009, pp. 1673–1678.

[15] P. R. Halmos, *Finite Dimensional Vector Spaces*, Princeton, NJ, Princeton University Press, 1948, p. 98.

CHAPTER 8

Random Search Algorithms

Chapter Outline

Gradient-based algorithms assume the performance measures are either quadratic or unimodal. For some classes of problems [1–3], the mathematical relation of the variable parameters to the performance measure is either unknown or is too complex to be useful. In yet other problems, constraints are placed on the variable parameters of the adaptive controller with the result that the performance surface is no longer unimodal. When the performance surface of interest is multimodal and contains saddlepoints, then any gradient-based algorithms finds only a local minimum. A random search algorithm has the ability to jump out of one valley with a local minimum into another valley with a potentially lower local minimum. Random algorithms have global search capabilities that work for any computable performance measure [4–14]. Random search algorithms tend to have slow convergence, especially in unimodal applications. They do, however, have the advantages of being simple to implement in logical form, of requiring little computation, of being insensitive to discontinuities, and of exhibiting a high degree of efficiency where little is known about the performance surface.

Systematic searches exhaustively survey the parameter space within specified bounds, making them capable of finding the global extremum of a multimodal performance measure. As a practical matter, however, this type of search is very time-consuming and incurs a high search loss, since most of the search period occurs in regions of poor performance.

Random searches are classified as either guided or unguided, depending on whether information is retained whenever the outcome of a trial step is learned. Furthermore, both the guided and unguided varieties of random search are given accelerated convergence by increasing the adopted step size in a successful search direction. Four representative examples of random search algorithms used for adaptive array applications are considered

in this chapter: linear random search (LRS), accelerated random search (ARS), guided accelerated random search (GARS), and genetic algorithm (GA).

8.1 LINEAR RANDOM SEARCH

The LRS algorithm discussed by Widrow and McCool [15] attempts to "learn" when a trial change in the adaptive weight vector is rejected. After adding a random change, $\Delta\mathbf{w}_k$, to the weight vector at the beginning of each iteration, the corresponding change in the performance measured is observed. A permanent weight vector change is proportional to the estimated performance measure change times the tentative weight vector change, that is,

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_s\{\hat{\mathfrak{P}}[\mathbf{w}_k] - \hat{\mathfrak{P}}[\mathbf{w}_k + \Delta\mathbf{w}_k]\}\Delta\mathbf{w}_k \tag{8.1}$$

where $\mathfrak{P}[\cdot]$ denotes the selected array performance measure, and μ_s is a step size constant. The random vector $\Delta\mathbf{w}_k$ has components generated from a normal probability density function with zero mean and variance σ^2. The constants μ_s and σ^2 are selected to ensure a fast, stable algorithm convergence. The LRS algorithm is "linear" because the weight change is proportional to the change in the performance measure.

The true change in the performance measure resulting from adding $\Delta\mathbf{w}_k$ to $\mathbf{w}_k$ is

$$(\Delta\mathfrak{P})_k \triangleq \mathfrak{P}[\mathbf{w}_k + \Delta\mathbf{w}_k] - \mathfrak{P}[\mathbf{w}_k] \tag{8.2}$$

When the performance measure value is estimated, then the corresponding estimated change in the performance measure is given by

$$(\Delta\hat{\mathfrak{P}})_k \triangleq \hat{\mathfrak{P}}[\mathbf{w}_k + \Delta\mathbf{w}_k] - \hat{\mathfrak{P}}[\mathbf{w}_k] \tag{8.3}$$

The error in the estimated change of the performance measure is then

$$\gamma_k \triangleq (\Delta\mathfrak{P})_k - (\Delta\hat{\mathfrak{P}})_k \tag{8.4}$$

and the variance in this error is given by

$$\begin{aligned}\text{var}[\gamma_k] &= \text{var}[(\Delta\hat{\mathfrak{P}})_k] \\ &= \text{var}\{\hat{\mathfrak{P}}[\mathbf{w}_k + \Delta\mathbf{w}_k]\} + \text{var}\{\hat{\mathfrak{P}}[\mathbf{w}_k]\}\end{aligned} \tag{8.5}$$

To determine the variance of the estimate $\hat{\mathfrak{P}}[\mathbf{w}_k]$, it is necessary to consider a specific performance measure and the estimate of that measure to be employed.

The mean square error (MSE) is the performance measure of interest so that $\mathfrak{P}[\mathbf{w}] = \xi[\mathbf{w}]$. The estimate of the MSE comes from averaging K independent samples as given by (4.98) of Section 4.3.2. With this choice of performance measure and its corresponding estimate, it follows that

$$\text{var}[\gamma_k] = \frac{2}{K}\{\xi^2[\mathbf{w}_k + \Delta\mathbf{w}_k] + \xi^2[\mathbf{w}_k]\} \tag{8.6}$$

where K = the number of independent samples on which the estimate $\hat{\xi}[\mathbf{w}]$ is based. In the steady state when the weight adjustment process operates near the minimum point of

the performance surface, then (8.6) is very nearly

$$\text{var}[\gamma_k] \approx \frac{4}{K}\xi_{\min}^2 \tag{8.7}$$

The tentative random changes in the weight vector produced by the LRS algorithm result in MSE perturbations. K data samples each iteration result in $\hat{\xi}[\mathbf{w}_k + \Delta\mathbf{w}_k]$. Parallel processors (with one processor using $\mathbf{w}_k$ and the other processor using $\mathbf{w}_k + \Delta\mathbf{w}_k$) decrease the convergence time, because the samples required to obtain $\hat{\xi}[\mathbf{w}_k]$ and $\hat{\xi}[\mathbf{w}_k + \Delta\mathbf{w}_k]$ are collected simultaneously. The value of the adaptive weight vector at the beginning of the next iteration is selected after the two $\hat{\xi}$ estimates are obtained. For any given iteration the average excess MSE resulting from the perturbation in the weight vector is given by

$$E\left\{\xi[\mathbf{w}_k] - \frac{\xi[\mathbf{w}_k] + \xi[\mathbf{w}_k + \Delta\mathbf{w}_k]}{2}\right\} = \frac{1}{2}E\{\xi[\mathbf{w}_k] - \xi[\mathbf{w}_k + \Delta\mathbf{w}_k]\} \tag{8.8}$$

The random weight perturbation vector $\Delta\mathbf{w}_k$ has zero mean and is uncorrelated with $\mathbf{w}_k$. Let $\Delta\mathbf{w}'_k$ denote the random weight perturbation vector in normal coordinates (in which the covariance matrix is diagonal so $\mathbf{R}'_{xx} = \mathbf{\Lambda}$), then $\text{cov}[\Delta\mathbf{w}_k] = \text{cov}[\Delta\mathbf{w}'_k] = \sigma^2\mathbf{I}$. Consequently, the average excess MSE can also be expressed as

$$\frac{1}{2}E\left\{\Delta\mathbf{w}_k^T\mathbf{R}_{xx}\Delta\mathbf{w}_k\right\} = \frac{1}{2}E\left\{\Delta\mathbf{w}_k'^T\mathbf{\Lambda}\Delta\mathbf{w}'_k\right\} = \frac{1}{2}\sigma^2\text{tr}(\mathbf{R}_{xx}) \tag{8.9}$$

Define the perturbation P as the ratio of the average excess MSE (resulting from the random perturbations in the weight vector) to the minimum MSE, then

$$P = \frac{\sigma^2\text{tr}(\mathbf{R}_{xx})}{2\xi_{\min}} \tag{8.10}$$

8.1.1 LRS Algorithm Stability

The weight adjustment equation (8.1) are rewritten in terms of the definitions given by (8.2), (8.3), and (8.4) for the MSE performance measure as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_s\{-(\Delta\xi)_k + \gamma_k\}\Delta\mathbf{w}_k \tag{8.11}$$

Recalling that $\mathbf{v}_k \stackrel{\Delta}{=} \mathbf{w}_k - \mathbf{w}_{\text{opt}}$, then we can rewrite (8.11) as

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mu_s\{-(\Delta\xi)_k + \gamma_k\}\Delta\mathbf{w}_k \tag{8.12}$$

Specify that σ^2 be small so that $\Delta\mathbf{w}_k$ is always small, and then

$$(\Delta\xi)_k = \Delta\mathbf{w}_k^T\nabla_k \tag{8.13}$$

where ∇_k is the gradient of the performance surface evaluated at $\mathbf{w}_k$. Since $\nabla_k = 2\mathbf{R}_{xx}\mathbf{v}_k$, (8.13) is written as

$$(\Delta\xi)_k = 2\Delta\mathbf{w}_k^T\mathbf{R}_{xx}\mathbf{v}_k \tag{8.14}$$

Consequently, (8.12) is rewritten as

$$\begin{aligned}\mathbf{v}_{k+1} &= \mathbf{v}_k + \mu_s\Delta\mathbf{w}_k\left[-2\Delta\mathbf{w}_k^T\mathbf{R}_{xx}\mathbf{v}_k + \gamma_k\right] \\ &= \left(\mathbf{I} - 2\mu_s\Delta\mathbf{w}_k\Delta\mathbf{w}_k^T\mathbf{R}_{xx}\right)\mathbf{v}_k + \mu_s\gamma_k\Delta\mathbf{w}_k\end{aligned} \tag{8.15}$$

The alternate (but equivalent) form of (8.1) represented by (8.15) is more useful for analysis even though the algorithm is implemented in the form suggested by (8.1). Equation (8.15) emphasizes that the adaptive weight vector is regarded as the solution of a first-order linear vector difference equation having a randomly varying coefficient $\mathbf{I} - 2\mu_s \Delta\mathbf{w}_k \Delta\mathbf{w}_k^T \mathbf{R}_{xx}$ and a random driving function $\mu_s \gamma_k \Delta\mathbf{w}_k$.

Premultiplying both sides of (8.15) by the transformation matrix $\mathbf{Q}$ of Section 4.1.3 converts the foregoing linear vector difference equation into normal coordinates

$$\mathbf{v}'_{k+1} = \left(\mathbf{I} - 2\mu_s \Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k \mathbf{\Lambda}\right)\mathbf{v}'_k + \mu_s \gamma_k \Delta\mathbf{w}'_k \tag{8.16}$$

Although (8.16) is somewhat simpler than (8.15), the matrix coefficient of $\mathbf{v}'_k$ still contains cross-coupling and randomness, thereby rendering (8.16) a difficult equation to solve. Stability conditions for the LRS algorithm are obtained without an explicit solution to (8.16) by considering the behavior of the adaptive weight vector mean.

Taking the expected value of both sides of (8.16) and recognizing that $\Delta\mathbf{w}'_k$ is a random vector that is uncorrelated with γ_k and $\mathbf{v}'_k$, we find that

$$\begin{aligned} E\{\mathbf{v}'_{k+1}\} &= E\left\{\left(\mathbf{I} - 2\mu_s \Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k \mathbf{\Lambda}\right)\mathbf{v}'_k\right\} + \mu_s E\{\gamma_k \Delta\mathbf{w}'_k\} \\ &= \left(\mathbf{I} - 2\mu_s E\left\{\Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k\right\}\mathbf{\Lambda}\right) E\{\mathbf{v}'_k\} + 0 \\ &= (\mathbf{I} - 2\mu_s \sigma^2 \mathbf{\Lambda}) E\{\mathbf{v}'_k\} \end{aligned} \tag{8.17}$$

The solution to (8.17) is given by [15]

$$E\{\mathbf{v}'_k\} = (\mathbf{I} - 2\mu_s \sigma^2 \mathbf{\Lambda})^k \mathbf{v}'_0 \tag{8.18}$$

For the initial conditions $\mathbf{v}'_0$, (8.18) gives the expected value of the weight vector's transient response. If (8.18) is stable, then the mean of $\mathbf{v}'_k$ must converge. The stability condition for (8.18) is

$$\frac{1}{\lambda_{\max}} > \mu_s \sigma^2 > 0 \tag{8.19}$$

If we choose $\mu_s \sigma^2$ to satisfy (8.19), it then follows that

$$\lim_{k \to \infty} E\{\mathbf{v}'_k\} = 0 \tag{8.20}$$

Since the foregoing transient behavior is analogous to that of the method of steepest descent discussed in Section 4.1.2, it is argued by analogy that the time constant of the pth mode of the expected value of the weight vector is given by

$$\tau_p = \frac{1}{2\mu_s \sigma^2 \lambda_p} \tag{8.21}$$

Furthermore, the time constant of the pth mode of the MSE learning curve is one-half the aforementioned value so that

$$\tau_{p_{\text{mse}}} = \frac{1}{4\mu_s \sigma^2 \lambda_p} \tag{8.22}$$

Satisfying the stability condition (8.19) implies only that the mean of the adaptive weight vector will converge according to (8.20); variations in the weight vector about the mean value may be quite severe, however. It is therefore of interest to obtain an

indication of the severity of variations in the weight vector when using the LRS algorithm by deriving an expression for the covariance of the weight vector. In obtaining such an expression, it will simply be assumed that the weight vector covariance is bounded and that the weight vector behaves as a stationary stochastic process after initial transients have died out. Assuming that a bounded steady-state covariance matrix exists, we may calculate an expression for such a covariance by multiplying both sides of (8.16) by their respective transposes to obtain

$$\begin{aligned}\mathbf{v}'_{k+1}\mathbf{v}'^{T}_{k+1} = &\left(\mathbf{I} - 2\mu_s \Delta\mathbf{w}'_k \Delta\mathbf{w}'^{T}_k \mathbf{\Lambda}\right)\mathbf{v}'_k\mathbf{v}'^{T}_k\left(\mathbf{I} - 2\mu_s \mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\\ &+\mu_s^2\gamma_k^2\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\\ &+\left(I - 2\mu_s\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\right)\mathbf{v}'_k\mu_s\gamma_k\Delta\mathbf{w}'^{T}_k\\ &+\mu_s\gamma_k\Delta\mathbf{w}'_k\mathbf{v}'^{T}_k\left(\mathbf{I} - 2\mu_s\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\end{aligned} \tag{8.23}$$

Now we take expected values of both sides of (8.23) recalling that γ_k and $\Delta\mathbf{w}'_k$ are zero-mean uncorrelated stationary processes so that

$$\begin{aligned}E\{\mathbf{v}'_{k+1}\mathbf{v}'^{T}_{k+1}\} = &E\{\left(\mathbf{I} - 2\mu_s\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\right)\mathbf{v}'_k\mathbf{v}'^{T}_k\left(\mathbf{I} - 2\mu_s\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\}\\ &+\mu_s^2E\{\gamma_k^2\}E\{\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\} + 0\end{aligned} \tag{8.24}$$

Since $\text{var}[\gamma_k] \cong (4/K)\xi_{\min}^2$ and $\text{cov}[\Delta\mathbf{w}'_k] = \sigma^2\mathbf{I}$, it follows that (8.24) is expressed as

$$\begin{aligned}E\{\mathbf{v}'_{k+1}\mathbf{v}'^{T}_{k+1}\} = &E\{\left(\mathbf{I} - 2\mu_s\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\right)\\ &\cdot\mathbf{v}'_k\mathbf{v}'^{T}_k\left(\mathbf{I} - 2\mu_s\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\} + \mu_s^2\frac{4}{K}\xi_{\min}^2\sigma^2\mathbf{I}\end{aligned} \tag{8.25}$$

In the steady state $\mathbf{v}'_k$ is also a zero-mean stationary random process that is uncorrelated with $\Delta\mathbf{w}'_k$ so (8.25) is written as

$$\begin{aligned}E\{\mathbf{v}'_{k+1}\mathbf{v}'^{T}_{k+1}\} = &E\{\left(\mathbf{I} - 2\mu_s\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\right)E\left[\mathbf{v}'_k\mathbf{v}'^{T}_k\right]\left(\mathbf{I} - 2\mu_s\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\}\\ &+\mu_s^2\frac{4}{K}\xi_{\min}^2\sigma^2\mathbf{I}\end{aligned} \tag{8.26}$$

Consequently, the steady-state covariance of the adaptive weight vector is

$$\begin{aligned}\text{cov}[\mathbf{v}'_k] &= E\{\left(\mathbf{I} - 2\mu_s\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\right)\text{cov}[\mathbf{v}'_k]\left(\mathbf{I} - 2\mu_s\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\right)\} + \mu_s^2\frac{4}{N}\xi_{\min}^2\sigma^2\mathbf{I}\\ &= \text{cov}[\mathbf{v}'_k] - 2\mu_sE\{\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\}\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k]\\ &\quad - 2\mu_s\text{cov}\,[\mathbf{v}'_k]\mathbf{\Lambda}E\{\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\}\\ &\quad + 4\mu_s^2E\{\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k]\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\} + \mu_s^2\frac{4}{N}\xi_{\min}^2\sigma^2\mathbf{I}\\ &= \text{cov}[\mathbf{v}'_k] - 2\mu_s\sigma^2\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k] - 2\mu_s\sigma^2\text{cov}[\mathbf{v}'_k]\mathbf{\Lambda}\\ &\quad + 4\mu_s^2E\{\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k]\mathbf{\Lambda}\Delta\mathbf{w}'_k\Delta\mathbf{w}'^{T}_k\} + \mu_s^2\frac{4}{K}\xi_{\min}^2\sigma^2\mathbf{I}\end{aligned} \tag{8.27}$$

Equation (8.27) is not easily solved for the covariance of $\mathbf{v}'_k$, because the matrices appearing in the equation cannot be factored. It is likely (although not proven) that the steady-state covariance matrix of $\mathbf{v}'_k$ is diagonal. The results obtained with such a simplifying assumption do, however, indicate that there is some merit in the plausibility argument.

The random driving function appearing in (8.16) consists of components that are uncorrelated with each other and uncorrelated over time. Furthermore, the random coefficient $\mathbf{I} - 2\mu_s \Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k \mathbf{\Lambda}$ is diagonal on the average (though generally not for every value of k) and uncorrelated both with $\mathbf{v}'_k$ and with itself over time. Thus, it is plausible that the covariance of $\mathbf{v}'_k$ is a diagonal matrix.

Assuming that $\text{cov}[\mathbf{v}'_k]$ is in fact diagonal, then (8.27) can immediately be rewritten by merely rearranging terms as

$$4\mu_s\sigma^2\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k] - 4\mu_s^2 E\{\Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k \mathbf{\Lambda}\text{cov}[\mathbf{v}'_k]\mathbf{\Lambda}\Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k\} = \mu_s^2 \frac{4}{K}\xi_{\min}^2\sigma^2\mathbf{I} \quad (8.28)$$

Of greatest interest is the case for which adaptation is slow, in which

$$\mu_s\sigma^2\mathbf{\Lambda} \ll \mathbf{I} \quad (8.29)$$

Furthermore, it may be noted that

$$\mu_s^2 E\{\Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k \mathbf{\Lambda}\text{cov}[\mathbf{v}'_k]\mathbf{\Lambda}\Delta\mathbf{w}'_k \Delta\mathbf{w}'^T_k\} \cong (\mu_s\sigma^2\mathbf{\Lambda})^2\text{cov}[\mathbf{v}'_k] \quad (8.30)$$

and from (8.29) it follows that

$$(\mu_s\sigma^2\mathbf{\Lambda})^2\text{cov}[\mathbf{v}'_k] \ll \mu_s\sigma^2\mathbf{\Lambda}\text{cov}[\mathbf{v}'_k] \quad (8.31)$$

With the result of (8.31) it follows that the term $-4\mu_s^2 E\{\ \}$ appearing in (8.28) is neglected. Consequently, (8.28) is rewritten as

$$\text{cov}[\mathbf{v}'_k] = \frac{\mu_s}{K}\xi_{\min}^2\mathbf{\Lambda}^{-1} \quad (8.32)$$

The steady-state covariance matrix of $\mathbf{v}'_k$ given by (8.32) is based on a plausible assumption, but experience indicates that the predicted misadjustment obtained using this quantity generally yields accurate results [15].

The misadjustment experienced using the LRS algorithm is obtained by considering the average excess MSE due to noise in the weight vector, which is given by

$$E\{\mathbf{v}'^T_k \mathbf{\Lambda}\mathbf{v}'_k\} = \sum_{p=1}^{N} \lambda_p E\{(v'_{p_k})^2\} \quad (8.33)$$

where N is the number of eigenvalues of $\mathbf{\Lambda}$. If we use (8.32), it follows that for the LRS algorithm

$$E\left[\mathbf{v}'^T_k \mathbf{\Lambda}\mathbf{v}'_k\right] = \sum_{p=1}^{N} \lambda_p \left(\frac{\mu_s}{K}\xi_{\min}^2 \frac{1}{\lambda_p}\right) = \frac{N\mu_s}{K}\xi_{\min}^2 \quad (8.34)$$

Since the misadjustment M is defined to be the average excess MSE divided by the minimum MSE

$$M \triangleq \frac{E\{\mathbf{v}'^T_k \mathbf{\Lambda}\mathbf{v}'_k\}}{\xi_{\min}} \quad (8.35)$$

It follows that for the LRS algorithm

$$M = \frac{N\mu_s}{K}\xi_{\min} \quad (8.36)$$

The result given by (8.36) can be expressed in terms of the perturbation of the LRS process as

$$M = \frac{N\mu_s\sigma^2\text{tr}(\mathbf{R}_{xx})}{2KP} = \frac{N^2\mu_s\sigma^2\lambda_{\text{av}}}{2KP} \tag{8.37}$$

Now recall that the time constant of the pth mode of the learning curve for the LRS algorithm (in terms of the number of iterations required) is given by (8.21). Since one iteration of the weight vector requires two estimates of $\hat{\xi}$, $2K$ samples of data are used per iteration, and the learning curve time constant expressed in terms of the number of data samples is

$$T_{p_{\text{mse}}} \triangleq 2K\tau_{p_{\text{mse}}} = \frac{K}{2\mu_s\sigma^2\lambda_p} \tag{8.38}$$

From (8.38) it follows immediately that

$$\lambda_p = \frac{K}{2\mu_s\sigma^2}\left(\frac{1}{T_{p_{\text{mse}}}}\right) \tag{8.39}$$

and

$$\lambda_{\text{av}} = \frac{K}{2\mu_s\sigma^2}\left(\frac{1}{T_{p_{\text{mse}}}}\right)_{\text{av}} \tag{8.40}$$

Substituting (8.40) into (8.37) then yields

$$M = \frac{N^2}{4P}\left(\frac{1}{T_{p_{\text{mse}}}}\right)_{\text{av}} \tag{8.41}$$

Since the total misadjustment consists of a stochastic component M and a deterministic component P from (8.41), we may write

$$M_{\text{tot}} = \frac{N^2}{4P}\left(\frac{1}{T_{p_{\text{mse}}}}\right)_{\text{av}} + P \tag{8.42}$$

If the deterministic component of the total misadjustment is optimally chosen, then both M and P are equal and P is one-half the total misadjustment so that

$$(M_{\text{tot}})_{\min} = \frac{N^2}{2P_{\text{opt}}}\left(\frac{1}{T_{p_{\text{mse}}}}\right)_{\text{av}} = N\left[\left(\frac{1}{T_{p_{\text{mse}}}}\right)_{\text{av}}\right]^{1/2} \tag{8.43}$$

It is informative to compare this result with the corresponding result (4.83) for the LMS algorithm.

8.2 ACCELERATED RANDOM SEARCH

Suppose we want to minimize the performance measure, $\Re(\mathbf{w})$, over some range of values for the complex weight vector $\mathbf{w}$. Let the adaptive processor change the complex weights using a simplified version of accelerated random search reported by Baird and Rassweiler [16] as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_s(k)[\Delta\mathbf{w}(k)] \tag{8.44}$$

where $\mu_s(k)$ is the step size initially set at $\mu_s(0) = \mu_0$, and $\Delta\mathbf{w}(k)$ is a random vector whose components are given by

$$\Delta w_i(k) = \cos\theta_i + j\sin\theta_i, \quad i = 1, 2, \ldots, m \tag{8.45}$$

where θ_i is a uniformly distributed random angle on the interval $\{0, 2\pi\}$ so that $|\Delta w_i(k)| = 1$ and $\Delta\mathbf{w}(k)$ controls the direction of the weight vector change while μ_s controls the magnitude.

Initially the weight vector, $\mathbf{w}(0)$, and the corresponding performance measure, $\mathfrak{P}[\mathbf{w}(0)]$, (or an estimate thereof $\hat{\mathfrak{P}}[\mathbf{w}(0)]$) is evaluated. The weight vector changes in accordance with (8.44) using $\mu_s(0) = \mu_0$. The performance index $\mathfrak{P}[\mathbf{w}(1)]$ is evaluated and compared with $\mathfrak{P}[\mathbf{w}(0)]$. If this comparison indicates improved performance, then the weight direction change vector $\Delta\mathbf{w}$ is retained, and the step size μ_s is doubled (resulting in "accelerated" convergence). If, however, the resulting performance is not improved, then the previous value of $\mathbf{w}$ is retained as the starting point, a new value of $\Delta\mathbf{w}$ is selected, and μ_s is reset to μ_0. As a consequence of always returning to the previous value of $\mathbf{w}$ as the starting point for a new weight perturbation in the event the performance measure is not improved, the ARS approach is inherently stable, and stability considerations do not play a role in step size selection. A block diagram of this simplified version of accelerated random search is given in Figure 8.1.

Consider a single component of the complex weight vector for which $v_i = w_i - w_{\text{opt}_i}$. If $v_i(k)$ lies within $\mu_0/2$ of w_{opt}, then any further perturbation in that component of the weight vector of step size μ_0 will result in $v_i(k+1) \geq v_i(k)$ as shown in Figure 8.2. Consequently, if all components of the weight vector lie within or on the best performance surface contour contained within the circle of radius $\mu_0/2$ about w_{opt}, then no further improvement in the performance measure can possibly occur using step size μ_0. The condition where all weight vector components lie within this best performance surface contour therefore represents a lower limit on the possible improvement that is achieved with the ARS procedure, and this is the ultimate condition to which the weight vector is driven in the steady state.

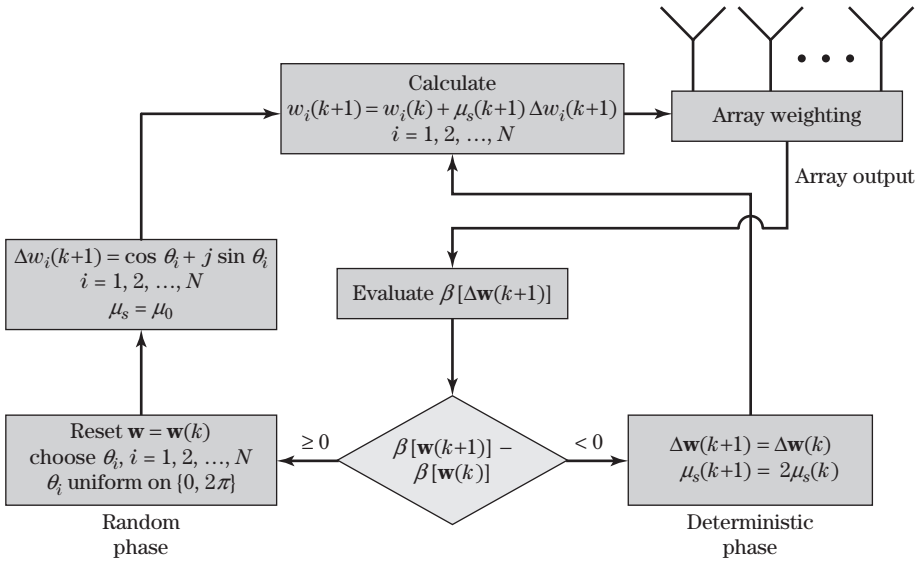


FIGURE 8-1 ■ Block diagram for ARS algorithm.

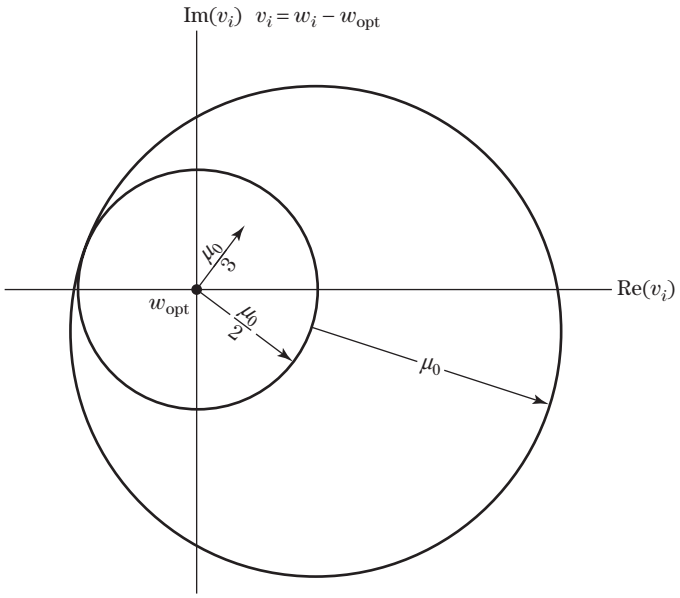


FIGURE 8-2 ■ Complex perturbation of step size μ_0 for a single weight vector component.

To simplify the development, assume it is equally likely that the weight vector component $w_i(k)$ lies anywhere within or on the circle for which $v_i = \mu_0/2$, it then follows that the steady-state expected value of v_i is $v_{ss}(k) = 0$, and the average excess MSE in this steady-state condition is (adopting $\mathfrak{P} = \xi$ and noting that $E\{|v_i|^2\} = \mu_0^2/8$)

$$E\{\mathbf{v}'^T(k)\mathbf{\Lambda}\mathbf{v}'(k)\} = \frac{\mu_0^2}{8}\xi_{\min}\,\text{tr}(\mathbf{R}_{xx}) \tag{8.46}$$

The average misadjustment for this steady-state condition is therefore

$$M_{\text{av}} = \frac{\mu_0^2}{8}\text{tr}(\mathbf{R}_{xx}) \tag{8.47}$$

On each successive iteration the weight vector components are perturbed by μ_0 from their steady-state values. Assume that the perturbation is taken from $v_{ss} = \rho$ as shown in Figure 8.3, then $E\{v_p\} = \mu_0$, $E\{|v_p|^2\} = 9\mu_0^2/8$, and the average total misadjustment for the random search perturbation is therefore

$$M_{\text{tot}} = \frac{9\mu_0^2}{8}\text{tr}(\mathbf{R}_{xx}) \tag{8.48}$$

From the foregoing discussion, it follows that in the steady state, as long as a correct decision is made concerning $\xi[\mathbf{w}(k+1)] - \xi[\mathbf{w}(k)]$, the average total misadjustment is given by (8.48).

In practice, the ARS algorithm examines the statistic $\hat{\mathfrak{P}}[\mathbf{w}(k+1)] - \hat{\mathfrak{P}}[\mathbf{w}(k)]$ instead of $\mathfrak{P}[\mathbf{w}(k+1)] - \mathfrak{P}[\mathbf{w}(k)]$, and the measured statistic contains noise that may yield a misleading indication of the performance measure difference. The performance measure difference due to the weight vector perturbation must be significantly larger than the standard deviation of the error in the estimated change in the performance measure: this is done by selecting $\Delta\mathfrak{P} = \mathfrak{P}[\mathbf{w}(k+1)] - \mathfrak{P}[\mathbf{w}(k)] > \sigma_\gamma$ where σ_γ^2 is given by (8.7). Selecting K and μ_s so that $\Delta\mathfrak{P} > \sigma_\gamma$ results in the average steady-state misadjustment approximated by (8.48). Furthermore, the performance measure difference due to the

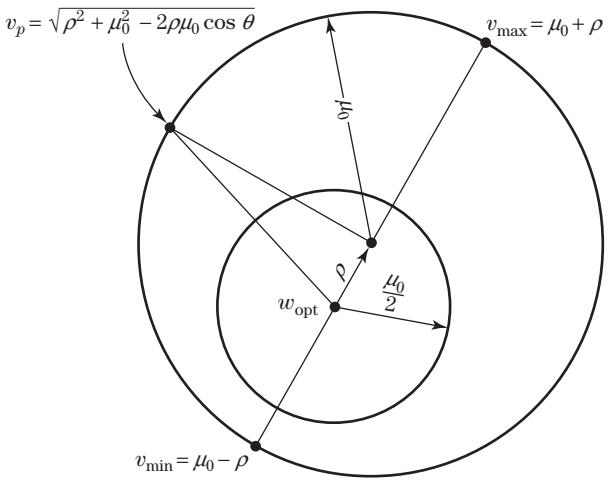


FIGURE 8-3 ■ Weight vector component perturbation resulting from step size μ_0 starting from $v_{ss} = \mu_0/3$.

weight vector perturbation should be less than $E\{\mathfrak{P}[\mathbf{w}(k)]\}$. Therefore, for $\mathfrak{P}$ selected to be ξ, in the steady state the constants K and μ_s should be selected to satisfy

$$\xi_{\min} > \frac{9}{8}\mu_0^2 \text{tr}(\mathbf{R}_{xx}) > \frac{2\xi_{\min}}{\sqrt{K}} \tag{8.49}$$

Even with K and μ_s selected to satisfy $\Delta\mathfrak{P} > \sigma_\gamma$, it is still possible for noise present in the measurements of system performance to produce deceptively good results for any one experiment. Such spurious results will on the average be corrected on successive trials.

8.3 GUIDED ACCELERATED RANDOM SEARCH

The GARS introduced by Barron [17–19] consists of two phases: a random phase and a deterministic phase. Control of the parameter space search passes back and forth between these two phases as the search finds parameter space regions having better or worse performance. In the initial random phase (which is an information-gathering phase), the adaptive weights are randomly perturbed according to a multivariate probability density function (PDF). Once a direction is found in which performance improvement occurs, the deterministic phase is entered, and the information acquired from the initial random phase is exploited as larger step sizes are taken in the direction of improved performance. Whenever an accelerated step in the deterministic phase produces an unsuccessful outcome, the random phase is reentered, but now the PDF governing the random search assigns smaller excursions to the parameter values than occurred in the initial random phase.

A block diagram representation of a simplified version of GARS is shown in Figure 8.4. Starting with an initial weight vector $\mathbf{w}_0$ a corresponding performance measure $\mathfrak{P}[\mathbf{w}_0]$ is evaluated. At all times the minimum value of the performance measure attained is stored and denoted by $\mathfrak{P}^*$, so initially $\mathfrak{P}^* = \mathfrak{P}[\mathbf{w}_0]$. The GARS algorithm begins in its random phase by generating a random weight vector perturbation $\Delta\mathbf{w}$, each of whose elements are drawn from a normal probability density function having zero mean and a variance σ^2. The variance is selected according to

$$\sigma^2 = K_1 + K_2\mathfrak{P}^* \tag{8.50}$$

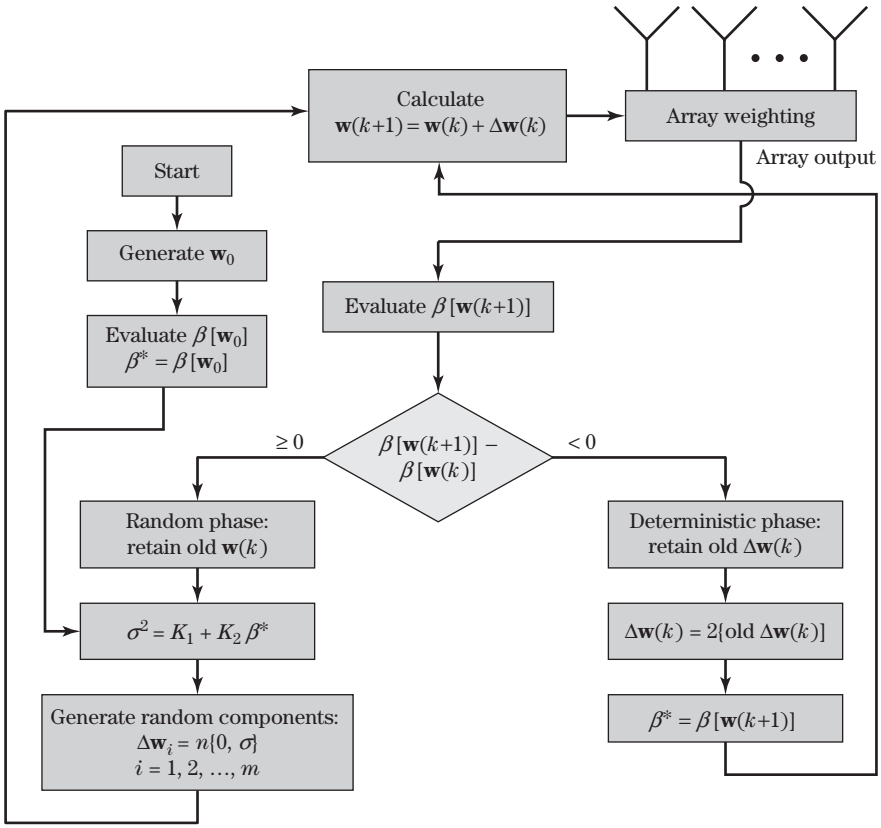


FIGURE 8-4 ■ Block diagram for GARS algorithm.

where K_1 and K_2 are design constants for the GARS algorithm selected so the step size is small enough when the optimum performance is realized yet big enough to gain useful performance surface information when the trial weight vector is far from optimum.

The next trial adaptive weight vector is then computed using

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) \tag{8.51}$$

and the corresponding performance measure, $\Re[\mathbf{w}(k+1)]$, is evaluated. If no improvement in the performance measure is realized, the algorithm remains in the random phase for the next trial weight vector, returning to the previous value of $\mathbf{w}$ as the starting point for the next weight perturbation. Once a direction in which to move for improved performance is determined, the deterministic phase of the algorithm is entered, and convergence is accelerated by continuing to travel in the direction of improved performance with twice the previous step size. The weight vector step $\Delta\mathbf{w}$ is continually doubled as long as performance measure improvements are realized. Once the performance measure begins to degrade, the search is returned to the random phase where the adaptive weight vector perturbations $\Delta\mathbf{w}$ are considerably smaller than before due to the smaller value of σ used in generating new search directions.

From the foregoing description of the simplified version of GARS, it is seen that the principal difference between GARS and ARS lies in how the random phase of the search is conducted. Not only is the search direction random (as it was before), but the step size is also random and governed by the parameter σ whose assigned value depends on the minimum value that the selected performance measure has attained. As a result, the search

step reduces as performance measure improvements are realized. The observations made in the previous section for the ARS algorithm with minimum step size μ_0 now apply in a statistical sense to the GARS algorithm. When $\mathfrak{P}$ is selected to be ξ the condition expressed by (8.49) should also be satisfied where now the expected change in performance measure due to weight perturbations is given by

$$E\{\Delta\xi\} = \sigma^2 \text{tr}(\mathbf{R}_{xx}) \tag{8.52}$$

Noise in the measurements of the performance measure can produce deceptively good results for any one experiment, thereby creating the risk of spurious measurements locking the search to a false solution. Local minima are avoided in the random phase of the algorithm by periodically reexamining the performance measure of the best weight vector found so far. Furthermore, the algorithm handles nonstationary operating conditions by periodically using large step sizes and conducting the exploration perturbations uniformly throughout the parameter space.

The general GARS algorithm incorporates features that are not simulated here. The most important such feature is the provision for a long-term memory due to employing a nonuniform multivariate probability distribution function [20] to generate search directions and thereby to guide the search to increase the probability that future trials will yield better performance scores than past trials. This multivariate PDF is shaped according to the results of a series of initial trials conducted during the opening stage of the search when no preference is given to any search direction. During the middle stage of the search, the multivariate PDF formed during the opening stage guides the search by generating new search directions. In the final search stage, the dimensionality of the parameter space search is reduced by converting from a simultaneous search involving all the parameters to a nearly sequential search involving only a small fraction of the parameters at any step. This selected fraction of the parameters to search is chosen randomly for each new iteration.

8.4 GENETIC ALGORITHM

A GA uses genetic and natural selection rules to maximize the fitness (i.e., minimize the cost) of a fitness (cost) function. The GA has proven very useful in complex antenna design and adaptive nulling. Some of the advantages of a GA for adaptive nulling use include the following:

- It optimizes with continuous or discrete variables. Thus, the GA interfaces with digitally controlled hardware like phase shifters and attenuators without having to quantize variables.
- It has constraints inherent in the formulation for limiting adaptive weights to a reasonable range.
- It does not require derivative information.
- It deals with a large number of variables, so it works with large arrays.
- It is well suited for parallel computers.
- It is able to avoid getting stuck in the valley of a local minimum.

A diagram of a GA is shown in Figure 8.5, and its operation is explained in the following paragraphs. Details for a practical implementation of a genetic algorithm are found in [21,22].

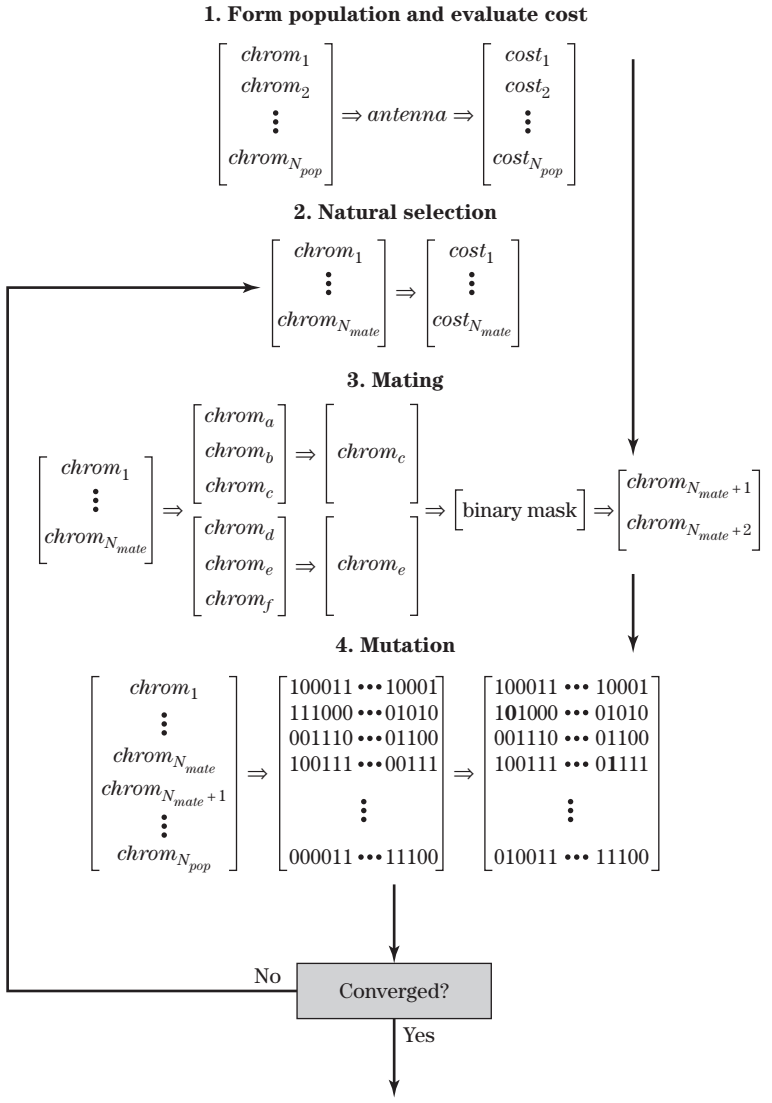


FIGURE 8-5 ■ GA flowchart.

1. Form population and evaluate cost. A GA starts with a random population matrix with N_{pop} rows . Each row is a chromosome and contains the adaptive weights for all the array elements. Since the adapted weights are normally digital, the population matrix is binary. If the adaptive weights have N_b bits and there are N_a adaptive elements, then each chromosome contains $N_b \times N_a$ bits. Thus, the population matrix is given by

$$\mathbf{P} = \begin{bmatrix} \underbrace{\begin{matrix} b_{1,1,1} & b_{1,1,2} & \cdots & b_{1,1,N_{bits}} \\ b_{2,1,1} & b_{2,1,2} & & \\ \vdots & & \ddots & \vdots \\ b_{N_{pop},1,1} & b_{N_{pop},1,2} & \cdots & b_{N_{pop},N_{bits}} \end{matrix}}_{\text{Element 1}} & \cdots & \underbrace{\begin{matrix} b_{1,N_a,1} & b_{1,N_a,2} & \cdots & b_{1,N_a,N_{bits}} \\ b_{2,N_a,1} & b_{2,N_a,2} & & \\ \vdots & & \ddots & \vdots \\ b_{N_{pop},N_a,1} & b_{N_{pop},N_a,2} & \cdots & b_{N_{pop},N_a,N_{bits}} \end{matrix}}_{\text{Element } N_a} \end{bmatrix} = \begin{bmatrix} chrom_1 \\ chrom_2 \\ \vdots \\ chrom_{N_{pop}} \end{bmatrix} \tag{8.53}$$

A cost function generates the cost or output from the chromosomes in the population matrix and places them in a corresponding cost vector. Each row is sent to the cost function for evaluation, so that cost, c_n, of $chrom_n$ is given by

$$c_n = f(chrom_n) \tag{8.54}$$

Then, all the costs are placed in a vector.

$$\mathbf{C} = [\, c_1 \; c_2 \; \cdots \; c_{N_{pop}} \,]^T \tag{8.55}$$

In the case of an adaptive array, the cost is the array performance measure.

2. Natural selection. Once all of the costs are determined, then natural selection occurs each generation (iteration) and discards unacceptable chromosomes that have a high cost. Of the N_{pop} chromosomes in a generation, the N_{mate} chromosomes with the lowest cost survive and form the mating pool, whereas the bottom $N_{pop} - N_{mate}$ are discarded to make room for the new offspring.

3. Mating. Mating is the process of selecting two chromosomes from the mating pool of N_{mate} chromosomes to produce two new offspring. Chromosomes from the mating population are selected and paired to create $N_{pop} - N_{mate}$ offspring that replace the discarded chromosomes. Tournament selection is a popular way to select chromosomes for mating. It randomly selects a subset of chromosomes from the mating pool, and the chromosome with the lowest cost in this subset becomes a parent. The tournament repeats $N_{pop} - N_{mate}$ times.

Mating combines the characteristics of two chromosomes to form two new chromosomes that replace two chromosomes discarded in the natural selection step. Uniform crossover is a general procedure that selects variables from each parent chromosome based on a mask and then places them in a new offspring chromosome. First, a random binary mask is created. A 1/0 in the mask column means the offspring receives the variable value from $chrom_{m/n}$. If it has a 0/1, then the offspring receives the variable value in $chrom_{n/m}$.

$$\begin{array}{rcccccccc}
chrom_m = & b_{m,1} & b_{m,2} & b_{m,3} & b_{m,4} & b_{m,5} & b_{m,6} & b_{m,7} & b_{m,8} \\
chrom_n = & b_{n,1} & b_{n,2} & b_{n,3} & b_{n,4} & b_{n,5} & b_{n,6} & b_{n,7} & b_{n,8} \\
\text{mask} = & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\text{offspring} = & b_{n,1} & b_{m,2} & b_{m,3} & b_{n,4} & b_{n,5} & b_{m,6} & b_{n,7} & b_{m,8} \\
\text{offspring} = & b_{m,1} & b_{n,2} & b_{n,3} & b_{m,4} & b_{m,5} & b_{n,6} & b_{m,7} & b_{n,8}
\end{array} \tag{8.56}$$

4. Mutation. The final step induces random mutations to alter a certain percentage of the variables in the list of chromosomes. A mutation changes a "1" to a "0" or vice versa in the $N_{pop} \times N_{bits}$ total number of bits in the population matrix. The best chromosome is usually not mutated. Mutation is specified as a percentage of the bits in the population matrix.

Particle swarm optimization (PSO) is another possible adaptive algorithm that minimizes the total output power [23]. It starts with the same population matrix as the GA. The rows in the matrix are called particles instead of chromosomes. The particles move on the cost surface with a velocity. The particles update their velocities and positions based on the local and global best solutions.

$$v_{m,n}^{new} = v_{m,n}^{old} + \ell_1 \times r_1 \times \left(p_{m,n}^{local\,best} - p_{m,n}^{old} \right) + \ell_2 \times r_2 \times \left(p_{m,n}^{global\,best} - p_{m,n}^{old} \right) \tag{8.57}$$

$$p_{m,n}^{new} = p_{m,n}^{old} + v_{m,n}^{new} \tag{8.58}$$

where

$$\begin{aligned} v_{m,n} &= \text{ particle velocity} \\ p_{m,n} &= \text{ particle variables} \\ r_1, r_2 &= \text{ independent uniform random numbers} \\ \ell_1 &= \text{cognitive parameter} \\ \ell_2 &= \text{social parameter} \\ p_{\text{m,n}}^{\text{local best}} &= \text{ best local solution} \\ p_{\text{m,n}}^{\text{global best}} &= \text{ best global solution} \end{aligned}$$

PSO updates the velocity vector for each particle and then adds that velocity to the particle position. Velocity updates depend on the estimate of the global solution found thus far and the best local solution in the present population. If the best local solution has a cost less than the cost of the estimate of the global solution, then the best local solution replaces the global solution estimate.

As an example, consider a 40-element array along the x-axis with elements spaced $d = \lambda/2$ apart and a 30 dB Chebyshev amplitude taper (a_n) [24]. Elements have a $\sin\phi$ element pattern and six-bit phase shifters. Assume there are two interference sources at $\phi = 43.9^\circ$ and 51.7° that are 60 dB stronger than the desired signal power in the main beam. The cost function assumes the phase shifts are antisymmetric about the center of the array.

$$\text{cost} = 20\log_{10}\left\{\sum_{i=1}^{2} s_i \sin\phi_i \left|\sum_{n=1}^{20} a_n \cos\left[(n-1)\frac{2\pi}{\lambda}du_i + \delta_n\right]\right|\right\} \tag{8.59}$$

where s_i is the signal strength of the two interference signals, $u_i = \cos\phi_i$, and δ_n are the adapted quantized phases.

Only two least significant bits are needed to perform the nulling (11.25° and 5.625°). The adapted phase settings are shown in Figure 8.6. Figure 8.7 shows the adapted pattern

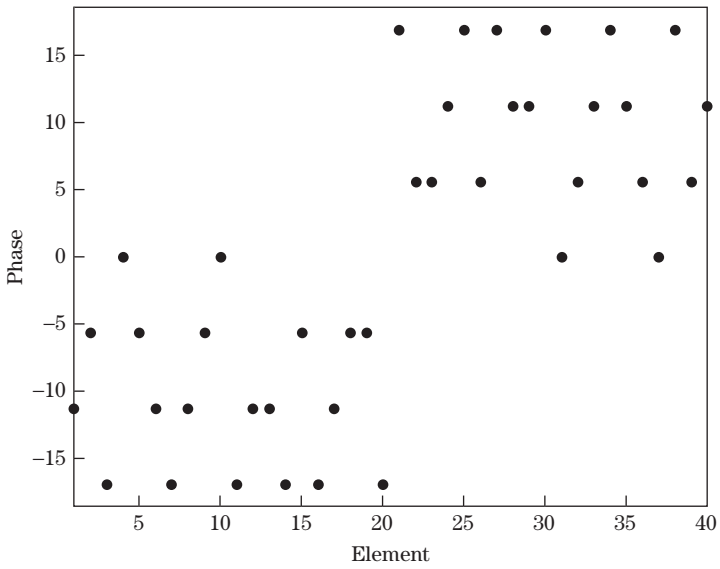


FIGURE 8-6 ■ Adapted phase weights that place nulls in the far-field pattern at $u = 0.62$ and $u = 0.72$. From R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation,* Vol. 45, No. 6, 1997, pp. 1009–1015.

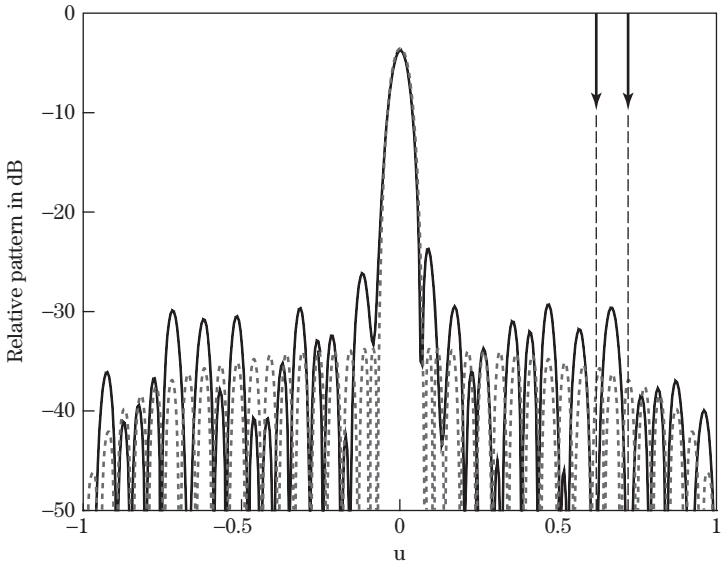


FIGURE 8-7 ■ Adapted (solid line) and quiescent (dotted line) array patterns for two 60 dB interference sources at $u = 0.62$ and $u = 0.72$. From R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation,* Vol. 45, No. 6, 1997, pp. 1009–1015.

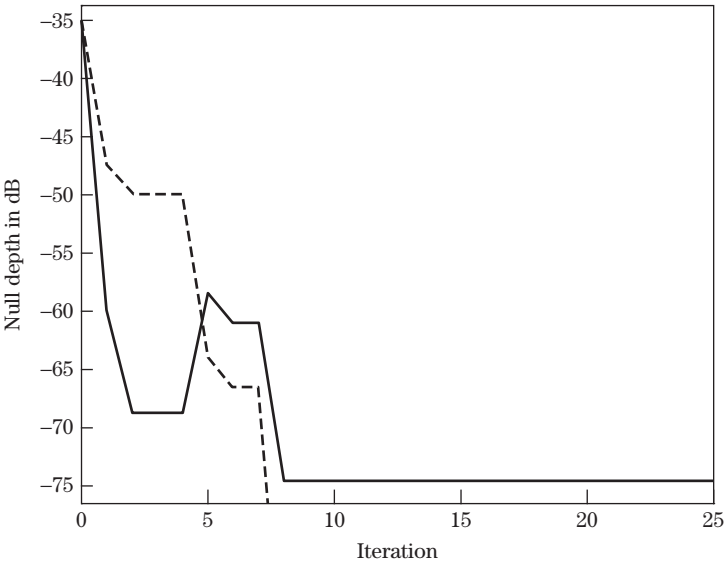


FIGURE 8-8 ■ Null depths at each null $u = 0.62$ (solid line) and $u = 0.72$ (dashed line). From R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 45, No. 6, 1997, pp. 1009–1015.

superimposed on the quiescent pattern. Convergence occurred in eight iterations as shown by the plots in Figure 8.8, Figure 8.9, and Figure 8.10. The solid line represents the best chromosome of the population, and the dashed line is the average of the population. After eight iterations, the algorithm reaches a set of adaptive weights that have effectively suppressed the two interferers. At this point, the algorithm required 68 power measurements.

Amplitude and phase adaptive nulling with a genetic algorithm has been successfully verified on an experimental antenna (Figure 8.11) that has 128 vertical columns with 16 dipoles per column equally spaced around a cylinder 104 cm in diameter [25]. Figure 8.12 is a cross sectional view of the antenna. The signals from the 16 dipoles are added together

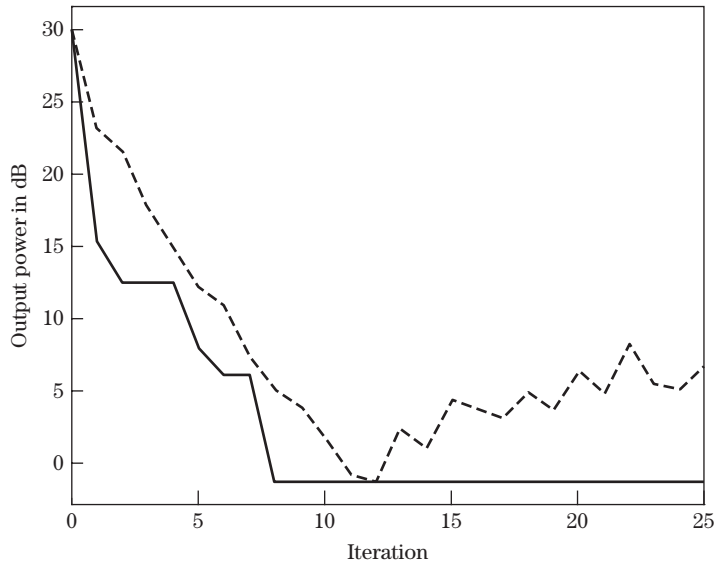


FIGURE 8-9 ■ Output power of the best chromosome (solid line) and average chromosome of the population (dashed line). From R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 45, No. 6, 1997, pp. 1009–1015.

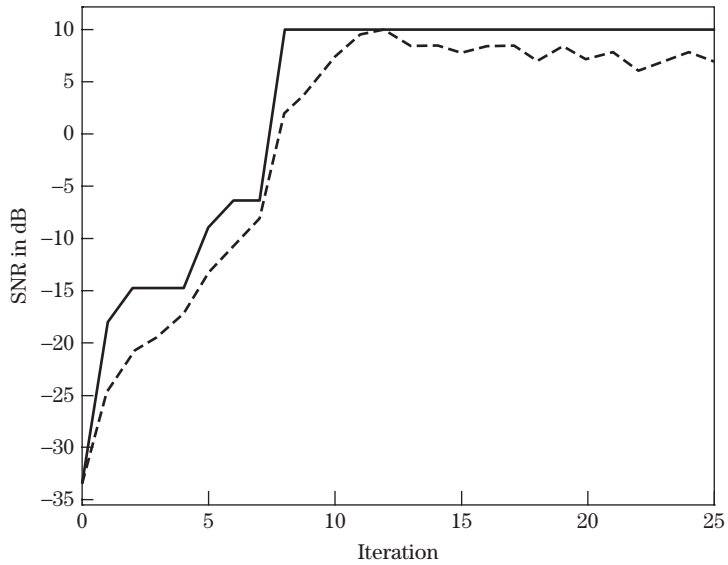


FIGURE 8-10 ■ Signal-to-noise ratio (SNR) of the array as a function of the number of iterations of the genetic algorithm. The solid line is the SNR of the best chromosome, and the dashed line is the SNR of the average chromosome of the population. From R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation,* Vol. 45, No. 6, 1997, pp. 1009–1015.

to form a fixed elevation main beam pointing 3° above horizontal. Eight consecutive elements are active at a time and with the elements spaced 0.42λ apart at 5 GHz. Each element has an eight-bit phase shifter (least significant bit equal to 0.0078125π radians) and eight-bit attenuator (least significant bit equal to .3125 dB). The antenna has a 25 dB $\overline{n} = 3$ Taylor amplitude taper.

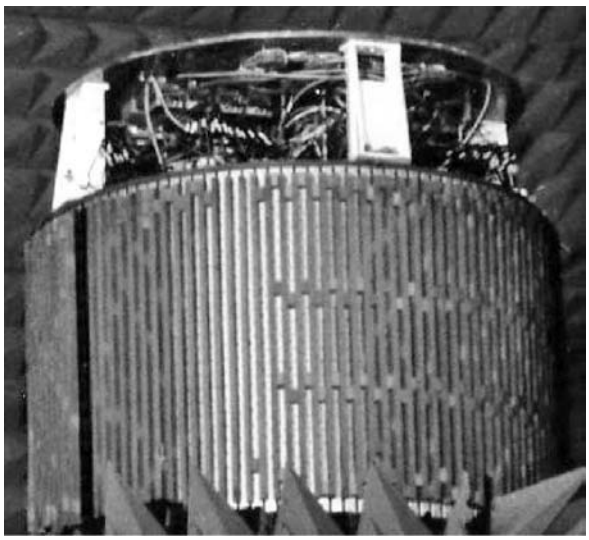


FIGURE 8-11 ■ The cylindrical array has 128 elements with 8 active at a time. From R. L. Haupt, "Adaptive nulling with a cylindrical array," AFRL-SN-RS-TR-1999-36, March 1999.

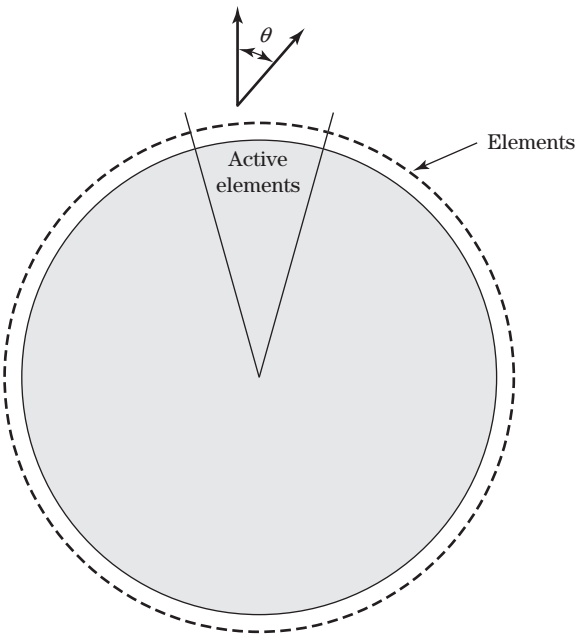


FIGURE 8-12 ■ Cross section of experimental cylindrical array.

A 5 GHz continuous source served as the interference. Only the four least significant bits of the phase shifters and attenuators were used to perform the adaptive nulling, so minimal distortion occurs to the main beam. The genetic algorithm had a population size of 16 chromosomes, and only one bit in the population was mutated every generation (mutation rate of 0.1%). The algorithm placed a deep null in less than 30 power measurements as shown in Figure 8.13 when the interference was at 45°. Figure 8.14 is the convergence plot for placing the null in the antenna pattern in Figure 8.13.

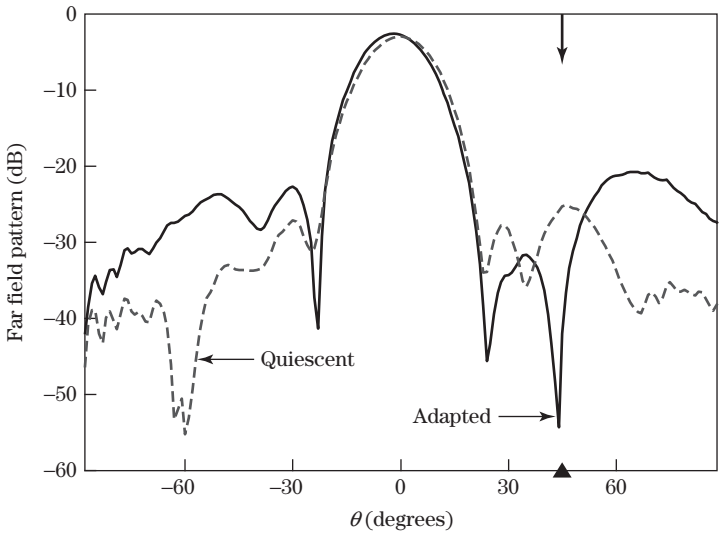


FIGURE 8-13 ■ Measured adapted and quiescent far-field patterns for an interference source at 45°.

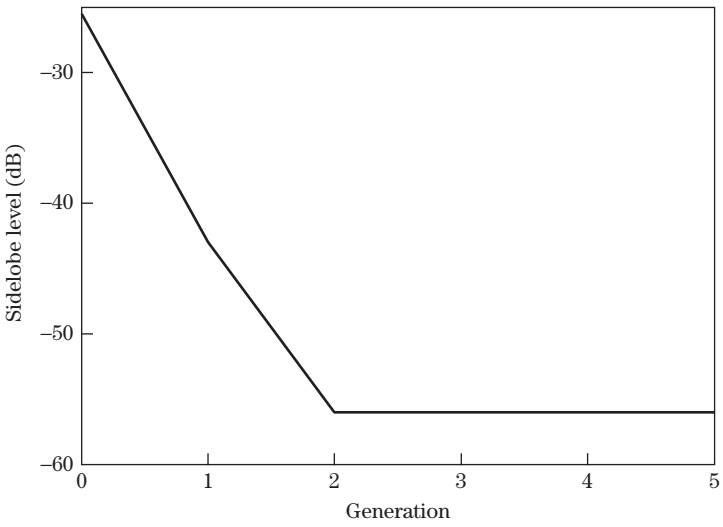


FIGURE 8-14 ■ GA convergence for placing a null at 45°.

8.5 | COMPARISON OF RANDOM SEARCH ALGORITHMS

A four-element array was simulated having the array configuration and signal environment represented in Figure 4.30. The random search algorithms have convergence speeds that are much slower than for the LMS algorithm when the MSE performance measure is used, and the selected three-jammer scenario was simulated under two different jamming conditions: one condition having a moderate eigenvalue spread of $\lambda_{max}/\lambda_{min} = 153.1$; and another condition with a more severe eigenvalue spread of $\lambda_{max}/\lambda_{min} = 2440$. When we select jammer-to-thermal noise ratios of $J_1/n = 25$, $J_2/n = 4$, and $J_3/n = 20$ and a signal-to-thermal noise ratio of $s/n = 10$, the corresponding eigenvalues are $\lambda_1 = 153.1$, $\lambda_2 = 42.6$, $\lambda_3 = 3.34$, and $\lambda_4 = 1$ for which $SNR_{opt} = 15.9$ (12 dB). Figures 8.15–8.18

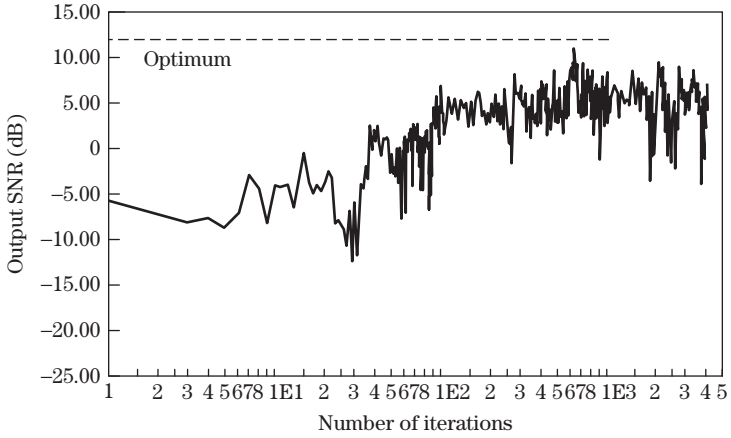


FIGURE 8-15 ■ Output SNR versus number of iterations for the LRS algorithm with eigenvalue spread = 153.1.

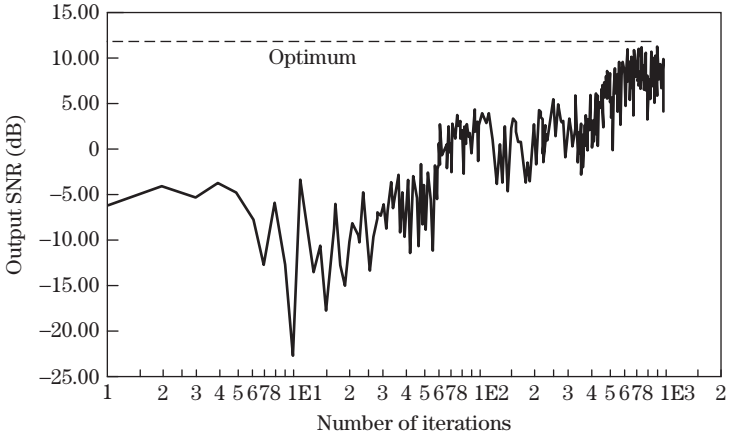


FIGURE 8-16 ■ Output SNR versus number of iterations for the ARS algorithm with eigenvalue spread = 153.1.

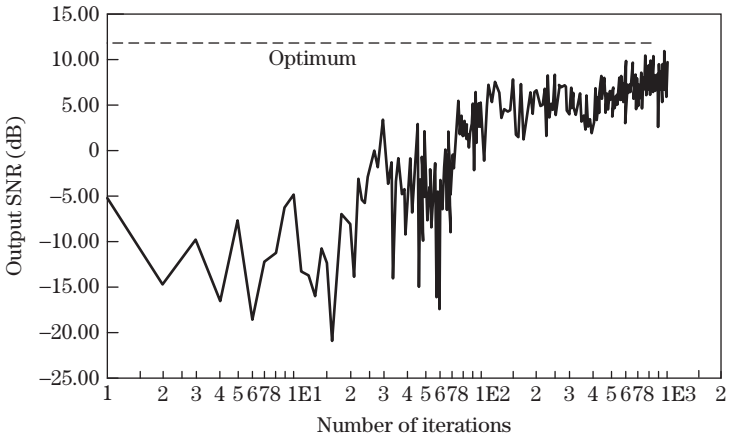


FIGURE 8-17 ■ Output SNR versus number of iterations for the GARS algorithm with eigenvalue spread = 153.1.

give typical convergence results for the case where $\lambda_{max}/\lambda_{min} = 153.1$. Likewise, when we select jammer-to-thermal noise ratios of $J_1/n = 500$, $J_2/n = 40$, and $J_3/n = 200$ and a signal-to-thermal noise ratio of $s/n = 10$, the corresponding eigenvalues are then $\lambda_1 = 2440$, $\lambda_2 = 494$, $\lambda_3 = 25.6$, and $\lambda_4 = 1$ for which $SNR_{opt} = 15.08$ (11.8 dB). Figures 8.19–8.24 then give typical convergence results for the case where $\lambda_{max}/\lambda_{min} = 2,440$.

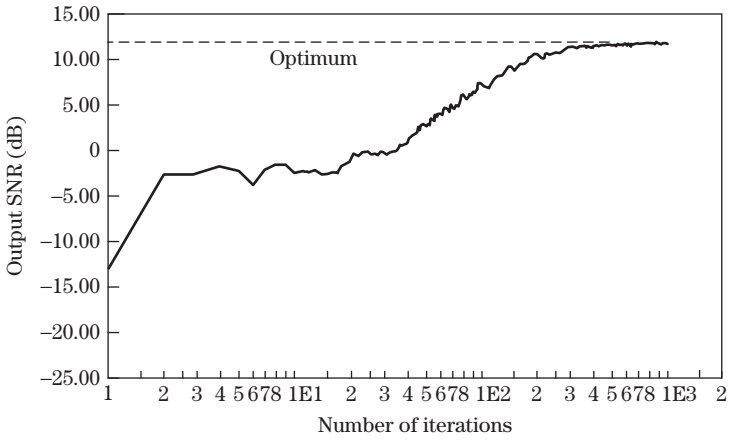


FIGURE 8-18 ■ Output SNR versus number of iterations for the LMS algorithm with eigenvalue spread = 153.1.

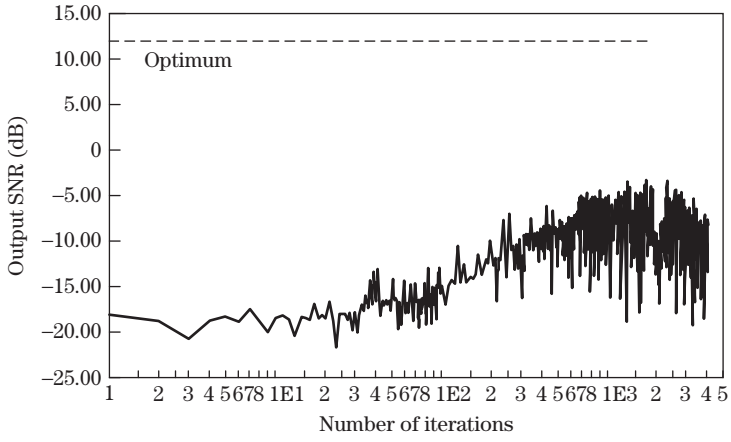


FIGURE 8-19 ■ Output SNR versus number of iterations for the LRS algorithm with eigenvalue spread = 2,440.

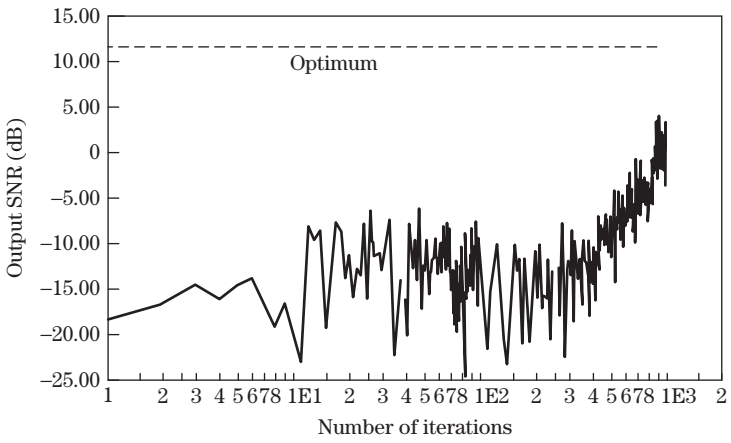


FIGURE 8-20 ■ Output SNR versus number of iterations for the ARS algorithm with eigenvalue spread = 2,440.

The ARS weight adjustment scheme in Figure 8.1 needs modification, because the farther $\mathbf{w}(k+1)$ is from $\mathbf{w}_{\text{opt}}$ the greater is the variance in the estimate $\hat{\xi}[\mathbf{w}(k+1)]$. Consequently, if the step size μ_0 is selected to obtain an acceptable steady-state error in the neighborhood of $\mathbf{w}_{\text{opt}}$, it may well be that the changes in $\xi[\mathbf{w}(k+1)]$ occurring as a consequence of the perturbation $\Delta\mathbf{w}(k+1)$ are overwhelmed by the random fluctuations

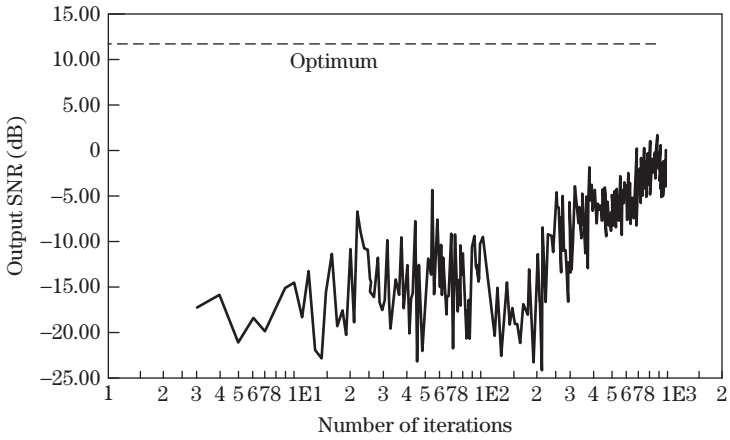


FIGURE 8-21 ■
Output SNR versus number of iterations for the GARS algorithm with eigenvalue spread = 2,440.

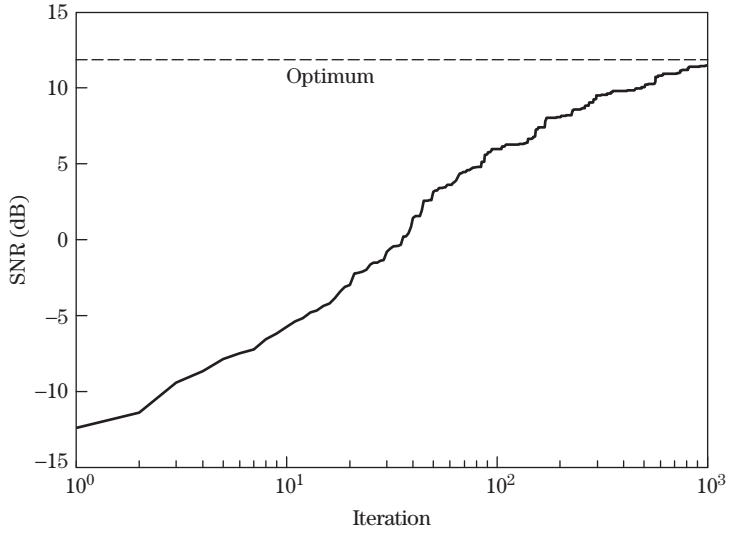


FIGURE 8-22 ■
Output SNR versus number of iterations for the GA with eigenvalue spread = 2,440.

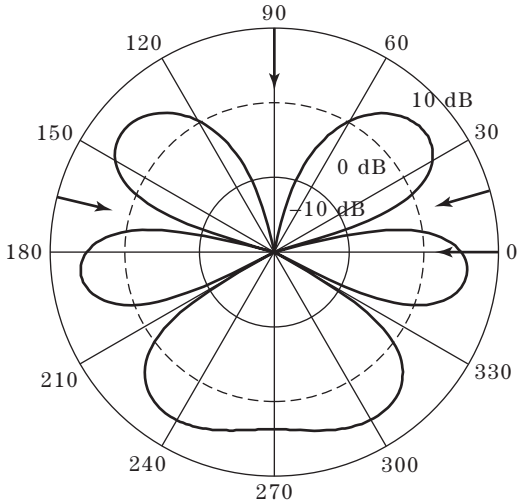


FIGURE 8-23 ■
Adapted pattern after 1,000 iterations of the GA.

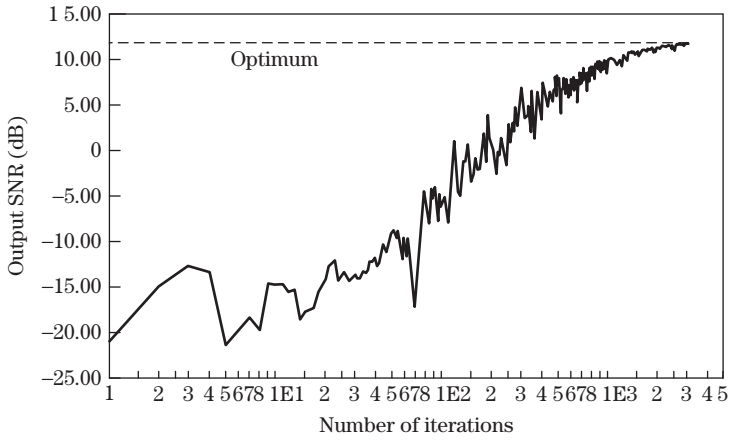


FIGURE 8-24 ■ Output SNR versus number of iterations for the LMS algorithm with eigenvalue spread = 2,440.

experienced in $\hat{\xi}[\mathbf{w}(k+1)]$ when $\mathbf{w}(k+1)$ is far from $\mathbf{w}_{\text{opt}}$. When this situation occurs, the adjustment algorithm yields a succession of weights that slowly meander aimlessly with step size μ_0. As a result, the step size μ_0 should reflect the changes in the variance of $\hat{\xi}[\mathbf{w}(k+1)]$ that occur when $\mathbf{w}(k+1)$ is far from $\mathbf{w}_{\text{opt}}$. This correction involves incorporating a step size $\mu_s = \sqrt{K_1 + K_2 \mathfrak{P}^*}$ into the ARS algorithm in accordance with the philosophy expressed by the GARS algorithm in Figure 8.4. Of course, it would be preferable to use $\mu_s = \sqrt{K_1 + K_2(\mathfrak{P}^* - \mathfrak{P}_{\min})}$, but in general $\mathfrak{P}_{\min}$ is unknown.

The LRS, ARS, and GARS algorithms were all simulated using $K = 90$ to obtain the estimate of MSE, which was the performance measure used in all cases. To satisfy the condition imposed by (8.49), the GARS algorithm was simulated using

$$\sigma^2 \text{tr}(\mathbf{R}_{xx}) = \mathfrak{f}_1 + \mathfrak{f}_2 \xi^* \tag{8.60}$$

where $\mathfrak{f}_1 = \frac{1}{160}$ and $\mathfrak{f}_2 = 0.1$. Likewise, the ARS algorithm was simulated using

$$\mu_s^2 \text{tr}(\mathbf{R}_{xx}) = \mathfrak{f}_1 + \mathfrak{f}_2 \xi^* \tag{8.61}$$

with $\mathfrak{f}_1$ and $\mathfrak{f}_2$ assigned the same values as for the GARS algorithm. The LRS algorithm was simulated using the constants $\mu_s = 1.6$ and $\sigma^2 \text{tr}(\mathbf{R}_{xx}) = 0.05$, thereby yielding a greater misadjustment error than either the ARS or GARS algorithms. The LMS algorithm was also simulated for purposes of comparison with step size corresponding to $\mu_s \text{tr}(\mathbf{R}_{xx}) = 0.1$ and using an estimated gradient derived from the average value of three samples of $e(k)\mathbf{x}(k)$ so $K = 3$ instead of the more common $K = 1$. In all cases the initial weight vector was taken to be $\mathbf{w}^T(0) = [0.1, 0, 0, 0]$.

The results of Figures 8.15–8.17 show that both the ARS and GARS algorithms are within 3 dB of the optimum SNR after about 800 iterations, whereas the LRS algorithm does not reach this point after 4,000 iterations, even though the misadjustment is more severe than for the ARS and GARS algorithms. This result indicates that the misadjustment versus speed of adaptation trade-off favors the ARS and GARS algorithms more than the LRS algorithm. The LMS algorithm by contrast is within 3 dB of the optimum output SNR after only 150 iterations with only a small degree of misadjustment. The extreme disparity in speed of convergence between the LMS algorithm and the three random search algorithms is actually more pronounced than the comparison of number of iterations indicates because each iteration in the random search algorithms represents 90 samples,

whereas each iteration in the LMS algorithm represents only three samples. Consequently, the time scale on Figures 8.15–8.17 is 30 times greater than the corresponding time scale on Figure 8.18.

The results given in Figures 8.18–8.24 for the case where eigenvalue spread = 2,440 confirm the previous results obtained with an eigenvalue spread of only 153.1. These results also show that both the random search algorithms and the LMS algorithm are sensitive to eigenvalue spread in the $\mathbf{R}_{xx}$ matrix.

Figure 8.22 is a plot of the signal to noise ratio versus iteration when a GA controls the complex element weights of the Y-array. The GA has a population size of 8 and a mutation rate of 15%. The best result for each iteration appears in the plot. The number of power measurements per iteration is less than 7. The GA optimized the SNR without knowledge of the signal and jammer directions and powers. Figure 8.23 shows the adapted pattern after 1,000 iterations of the GA. Phase-only adaptive nulling is not a good alternative in this case, because there are not enough degrees of freedom to null all the jammers. It is remarkable that the convergence results of the GA in Figure 8-22 are very close to the results obtained for the LMS algorithm shown in Figure 8-24. The GA algorithm is the only random search algorithm that is actually competitive with the LMS algorithm in this extreme eigenvalue spread condition.

8.6 SUMMARY AND CONCLUSIONS

Random search algorithms search irregular and multimodal performance surfaces and require only a direct evaluation of the selected performance measure to implement. The weight adjustment computation for the LRS, ARS, GARS, and GA algorithms is extremely simple, requiring only modest computational power. More elaborate and complicated random search algorithms have been applied to adaptive control and pattern recognition systems [15], but the introduction of more sophisticated measures into random searches removes their simplicity (which is a primary virtue for adaptive array applications).

The price to be paid for simple computation and implementation requirements is longer convergence time to reach the optimal weight vector solution, although both the ARS and GARS algorithms have more favorable misadjustment versus speed of convergence trade-offs than the LRS algorithm. Nevertheless, the convergence speed realized with the LMS algorithm on unimodal performance surfaces is orders of magnitude faster than that realized by a random search algorithm. Furthermore, these three random search algorithms exhibited the same degree of convergence speed sensitivity to eigenvalue spread as the LMS algorithm. Finally, these random searches exhibit relatively slow convergence in high-dimensional spaces, a characteristic that reflects the fact that as the number of possible directions to be searched increases, then the convergence time also increases. This performance characteristic leads to the suggestion that reducing the dimensionality of the parameter space search over the final stages of convergence is helpful. Note that the brief discussion given here is not a complete survey of search algorithms that employ only measurements of the output power. In particular, consideration should be given to the "directed search" techniques like that proposed by Hooke and Jeeves [26] which may well exhibit superior convergence properties under certain conditions.

The genetic algorithm and other evolutionary and nature-based algorithms have proven useful in overcoming the deficiencies of the other random search algorithms. They easily escape from local minima, handle large numbers of parameters, and have been experimentally demonstrated.

8.7 PROBLEMS

1. ***Misadjustment versus Speed of Adaptation Trade-off for the LRS Algorithm [15]*** Assuming all eigenvalues are equal so that $(T_{p_{\text{mse}}})_{\text{av}} = T_{\text{mse}}$, plot T_{mse} versus N for the LRS algorithm assuming $(M_{\text{tot}})_{\min} = 10\%$ in (8.43), and compare this result with the corresponding plots obtained for the LMS and DSD algorithms in Problem 1 of Chapter 4.

2. ***Search Loss for a Simple Random Search with Reversing Step [27]*** Consider the simple random search algorithm described by

$$\mathbf{x}(i+1) = \mathbf{x}(i) + \Delta\mathbf{x}(i+1)$$

where $\Delta\mathbf{x}$ is a random displacement vector satisfying $|\Delta\mathbf{x}|^2 = 1$ and

$$\Delta\mathbf{x}(i+1) = \begin{cases} \xi \text{ if performance improvement is observed} \\ -\Delta\mathbf{x}(i) \text{ if no performance improvement is observed} \end{cases}$$

where ξ is the random direction of the displacement vector.

(a) Consider the parameter space of Figure 8.25 in which a performance improvement is realized for any angle in the range $-\pi/2 < \phi < \pi/2$ and no performance improvement is realized for any angle in the range $\pi/2 \leq \phi \leq 3\pi/2$. The mean displacement in the direction of a successful random step is given by

$$U(n) = \int_0^{\pi/2} \cos\phi p(\phi) d\phi$$

where n is the number of degrees of freedom and $p(\phi)$ is the probability density function of the angle ϕ for a uniform distribution of directions of the random step in the n-dimensional space. Show that

$$p(\phi) = \frac{\sin^{n-2}\phi}{2\int_0^{\pi/2} \sin^{n-2}\phi d\phi} = \frac{\Gamma(n-1)}{2^{n-2}\left[\Gamma\left(\frac{n-1}{2}\right)\right]^2} \sin^{n-2}\phi$$

where $\Gamma(\cdot)$ is the gamma function.

Hint: Note that the area of a ring-shaped zone on the surface of an n-dimensional sphere corresponding to the angle $d\phi$ is $A_{n-2} \times \sin^{n-2}\phi d\phi$. Consequently, the area

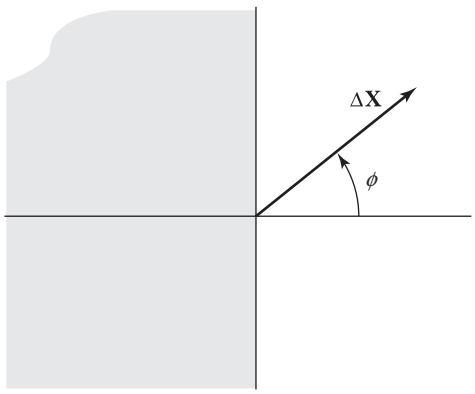


FIGURE 8-25 ■ Parameter space section showing displacement vector $\Delta\mathbf{X}$ and direction ϕ.

of the surface of the hypersphere included in the hypercone with vertical angle 2ϕ is given by

$$S(\phi) = A_{n-2} \int_0^{\phi} \sin^{n-2} \phi d\phi$$

The probability that a random vector lies in this cone for a uniform probability choice of the random direction is equal to the ratio of the "areas" $S(\phi)$ and $S(\pi)$. The desired probability density is then the derivative of this ratio with respect to the angle ϕ.

(b) Show that $U(n)$ defined in part (a) is given by

$$U(n) = \frac{\int_0^{\pi/2} \cos\phi \sin^{n-2} \phi d\phi}{\int_0^{\pi/2} \sin^{n-2} \phi d\phi} = \frac{\Gamma(n-1)}{2^{n-3}(n-1)\left[\Gamma\left(\frac{n-1}{2}\right)\right]^2}$$

Since the probability of successful and unsuccessful steps is the same for the parameter space of Figure 8.25, then on the average for one successful step there is one unsuccessful step and a corresponding reverse step—three steps in all. Therefore, the mean displacement for one successful step is reduced by two-thirds and is only $\frac{1}{3}U(n)$.

Defining search loss to be the number of steps required by the search such that the vector sum of these steps has the same length as one operating step in the successful direction, it follows that the mean search loss for the aforementioned algorithm is just $3/U(n)$.

3. ***Relative Search Efficiency Using the Search Loss Function [27]***
Consider a fixed step size gradient search defined by

$$\mathbf{x}(i+1) = \mathbf{x}(i) - a(i)\mu_s \frac{\Delta(i)}{|\Delta(i)|}$$

where

$$\begin{aligned} \mu_s &= \text{step size} \\ \Delta(i) &= \left[\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_n}\right]_{\mathbf{x}(i)} \\ a(i) &= \begin{cases} 1 & \text{if } F[\mathbf{x}(i+1)] < F[\mathbf{x}(i)] \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

and where $F[\cdot]$ denotes a known performance measure. Likewise, consider a fixed step size random search defined by

$$\mathbf{x}(i+1) = \mathbf{x}(i) - b(i)\Delta\mathbf{x}(i) + \Delta\mathbf{x}(i+1)$$

where $\Delta\mathbf{x}(i)$ is a random vector having a uniform distribution on a hypersphere of radius μ_s and centered at the origin, and

$$b(i) = \begin{cases} 1 & \text{if } F[\mathbf{x}(i)] \geq F[\mathbf{x}(i-1)] \\ 0 & \text{otherwise} \end{cases}$$

Define the search loss function to be the performance measure $F[x]$ divided by the ratio of performance improvement per performance evaluation, that is,

$$SL(\mathbf{x}) \triangleq \frac{F(\mathbf{x})}{\left(\frac{F(\mathbf{x})-F(\mathbf{x}+\Delta\mathbf{x})}{N}\right)}$$

where N is the total number of performance measure evaluations required at each search point.

(a) For the quadratic performance measure $F(\mathbf{x}) = \rho^2$, show that for the gradient search algorithm $F(\mathbf{x} + \Delta\mathbf{x}) = \rho^2 - 2\rho\mu_s + \mu_s^2$.

(b) To empirically evaluate the gradient Δ at some point $\mathbf{x}$ requires the evaluation of $F[\cdot]$ at $\mathbf{x}$ and at points along each coordinate direction separated from $\mathbf{x}$ by some distance ε. Therefore, a total of $n + 1$ function evaluations are required (where $n =$ number of degrees of freedom). Show for the previously defined gradient search that

$$SL(\mathbf{x}) = \frac{\rho^2(n+1)}{(2\rho\mu_s - \mu_s^2)}$$

(c) For a given base point $\mathbf{x}(i)$, the successor trial state $\mathbf{x}(i + 1)$ for the random search defines an angle ϕ with respect to a line connecting $\mathbf{x}(i)$ with the extremum point of the performance measure for which the probability density function $p(\phi)$ was obtained in Problem 2. Show that the expected value of performance measure improvement using the previously defined random search is

$$E\{-\Delta F\} = E\{\Delta\rho^2\} = \int_0^{\phi_0} \Delta\rho^2 p(\phi) d\phi$$

where

$$\phi_0 = \cos^{-1}\left\{\frac{\mu_s}{2\rho}\right\} \quad \text{and} \quad \Delta\rho^2 = 2\rho\mu_s\cos\phi - \mu_s^2$$

(d) Show for the previous random search algorithm that

$$SL(\mathbf{x}) = \frac{2\rho^2 \int_0^{\pi/2} \sin^{n-2}\phi d\phi}{\int_0^{\phi_0} 2\mu_s\rho\cos\phi\sin^{n-2}\phi d\phi - \int_0^{\phi_0} \mu_s^2 \sin^{n-2}\phi d\phi}$$

The search loss function of parts (b) and (d) is compared for specific values of ρ, μ_s, and n to determine whether the gradient search or the random search is more efficient.

4. ***Search Loss Function Improvement Using Step Reversal [28]***

The relative efficiency of the fixed step size random algorithm introduced in Problem 3 is significantly improved merely by adding a "reversal" feature to the random search. The fixed step size random search algorithm with reversal is described by

$$\mathbf{x}(i+1) = \mathbf{x}(i) + c(i)\Delta\mathbf{x}(i+1) + 2[c(i) - 1]\Delta\mathbf{x}(i)$$

where

$$c(i) = \begin{cases} 1 & \text{if } F[\mathbf{x}(i)] \leq F[\mathbf{x}(i-1)] \\ 0 & \text{otherwise} \end{cases}$$

The previous modification of the random search in effect searches at $\mathbf{y} - \Delta\mathbf{y}$ if the initial search at $\mathbf{y} + \Delta\mathbf{y}$ failed to produce an improvement in the performance measure.

(a) Show that

$$E\{-\Delta F\} = E\{\Delta\rho^2\} = \frac{3\int_0^{\phi_0} \Delta\rho^2 \sin^{n-2}\phi d\phi}{2\int_0^{\pi} \sin^{n-2}\phi d\phi}$$

where

$$\phi_0 = \cos^{-1}\left\{\frac{\mu_s}{2}\right\} \quad \text{and } \Delta\rho^2 = 2\rho\mu_s\cos\phi - \mu_s^2$$

(b) Show that

$$SL(\mathbf{x}) = \frac{4\mu_s^2\int_0^{\pi/2}\sin^{n-2}\phi d\phi}{\int_0^{\phi_0} 6\mu_s\rho\cos\phi\sin^{n-2}\phi d\phi - \int_0^{\phi_0} 3\mu_s^2\sin^{n-2}\phi d\phi}$$

This search loss is compared with that obtained in Problem 3(d) for specified ρ, μ_s, and n to determine exactly the improvement that is realized by addition of the reversal feature.

5. ***Genetic Algorithm.*** A 20-element array of point sources spaced 0.5λ apart has six-bit amplitude and phase weights and a 20 dB, $\overline{n} = 3$ low sidelobe Taylor amplitude taper. The desired signal is incident on the peak of the main beam and is normalized to 1 or 0 dB. Two 30 dB jammers enter the sidelobes at 111° and 117°. The genetic algorithm has a population size of 8 and a mutation rate of 10%. Plot the power levels received by the array versus generation, the ratio of the signal power to the jammer versus generation, and the adapted antenna pattern superimposed on the quiescent pattern.

8.8 | REFERENCES

[1] M. K. Leavitt, "A Phase Adaptation Algorithm," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 754–756.

[2] P. A. Thompson, "Adaptation by Direct Phase-Shift Adjustment in Narrow-Band Adaptive Antenna Systems," *IEEE Trans. Antennas Propag.*, Vol. AP-24, No. 5, September 1976, pp. 756–760.

[3] G. J. McMurty, "Search Strategies in Optimization," Proceedings of the 1972 International Conference on Cybernetics and Society, October, Washington, DC, pp. 436–439.

[4] C. Karnopp, "Random Search Techniques for Optimization Problems," *Automatica*, Vol. 1, August 1963, pp. 111–121.

[5] G. J. McMurty and K. S. Fu, "A Variable Structure Automation Used as a Multimodal Searching Technique," *IEEE Trans. Autom. Control*, Vol. AC-11, July 1966, pp. 379–387.

[6] R. L. Barton, "Self-Organizing Control: The Elementary SOC—Part I," *Control Eng.*, February 1968.

[7] R. L. Barton, "Self-Organizing Control: The General Purpose SOC— Part II," *Control Eng*., March 1968.

[8] M. A. Schumer and K. Steiglitz, "Adaptive Step Size Random Search," *IEEE Trans. Autom. Control*, Vol. AC-13, June 1968, pp. 270–276.

[9] R. A. Jarvis, "Adaptive Global Search in a Time-Variant Environment Using a Probabilistic Automaton with Pattern Recognition Supervision," *IEEE Trans. Syst. Sci. Cybern*., Vol. SSC-6, July 1970, pp. 209–217.

[10] A. N. Mucciardi, "Self-Organizing Probability State Variable Parameter Search Algorithms for the Systems that Must Avoid High-Penalty Operating Regions," *IEEE Trans. Syst., Man, Cybern*., Vol. SMC-4, July 1974, pp. 350–362.

[11] R. A. Jarvis, "Adaptive Global Search by the Process of Competitive Evolution," *IEEE Trans. Syst., Man, Cybern*., Vol. SMC-5, May 1975, pp. 297–311.

[12] R. L. Haupt, "Beamforming with Genetic Algorithms," *Adaptive Beamforming,* S. Chandran, ed., pp. 78–93, Berlin, Springer-Verlag, 2004.

[13] R. L. Haupt, "Genetic Algorithms for Antennas," *Modern Antenna Handbook*, C. Balanis, New York, Wiley, 2008.

[14] R. L. Haupt, *Antenna Arrays: A Computational Approach*, New York, Wiley, 2010.

[15] B. Widrow and J. M. McCool, "A Comparison of Adaptive Algorithms based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Antennas Propag*., Vol. AP-24, No. 5, September 1976, pp. 615–637.

[16] C. A. Baird and G. G. Rassweiler, "Search Algorithms for Sonobuoy Communication," *Proceedings of the Adaptive Antenna Systems Workshop,* March 11–13, 1974, NRL Report 7803, Vol. 1, September 27, 1974, pp. 285–303.

[17] R. L. Barron, "Inference of Vehicle and Atmosphere Parameters from Free-Flight Motions," *AIAA J. Spacecr. Rockets*, Vol. 6, No. 6, June 1969, pp. 641–648.

[18] R. L. Barron, "Guided Accelerated Random Search as Applied to Adaptive Array AMTI Radar," *Proceedings of the Adaptive Antenna Systems Workshop,* March 11–13, 1974, Vol. 1, NRL Report 7803, September 27, 1974, pp. 101–112.

[19] A. E. Zeger and L. R. Burgess, "Adaptive Array AMTI Radar," *Proceedings of the Adaptive Antenna Systems Workshop,* March 11–13, 1974, NRL Report 7803, Vol. 1, September 27, 1974, pp. 81–100.

[20] A. N. Mucciardi, "A New Class of Search Algorithms for Adaptive Computation," *Proc. 1973 IEEE Conference on Decision and Control,* December, San Diego, Paper No. WA5-3, pp. 94–100.

[21] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2d ed., New York, John Wiley & Sons, 2004.

[22] R. L. Haupt and D. Werner, *Genetic Algorithms in Electromagnetics,* New York, Wiley, 2007.

[23] J. Kennedy and R. C. Eberhart, *Swarm Intelligence,* San Francisco, Morgan Kaufmann Publishers, 2001.

[24] R. L. Haupt, "Phase-Only Adaptive Nulling with a Genetic Algorithm," *IEEE Transactions on Antennas and Propagation,* Vol. 45, No. 6, 1997, pp. 1009–1015.

[25] R. L. Haupt and H. L. Southall, "Experimental Adaptive Nulling with a Genetic Algorithm," *Microwave Journal,* Vol. 42, No. 1, January 1999, pp. 78–89.

[26] R. Hooke and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems," *J. of the Assoc. for Comput. Mach.,* Vol. 8, April 1961, pp. 212–229.

[27] L. A. Rastrigin, "The Convergence of the Random Search Method in the External Control of a Many-Parameter System," *Autom. Remote Control*, Vol. 24, No. 11, April 1964, pp. 1337–1342.

[28] J. P. Lawrence, III and F. P. Emad, "An Analytic Comparison of Random Searching and Gradient Searching for the Extremum of a Known Objective Function," *IEEE Trans. Autom. Control*, Vol. AC-18, No. 6, December 1973, pp. 669–671.

CHAPTER

9

Adaptive Algorithm Performance Summary

Chapters 4 through 9 considered the transient response characteristics and implementation considerations associated with different classes of algorithms that are widely used for adaptive array applications. This chapter summarizes the principal characteristics of each algorithm class before considering some practical problems associated with adaptive array system design.

In each chapter of Part 2 the convergence speed of an algorithm representing a distinct adaptation philosophy was compared with the convergence speed of the least mean squares (LMS) algorithm. The convergence speeds of the various algorithms are compared for a selected example in this chapter. Since the misadjustment versus rate of adaptation trade-offs for the random search algorithms—linear random search (LRS), accelerated random search (ARS), and guided accelerated random search (GARS)—and for the differential steepest descent (DSD) algorithm of Chapter 4 are unfavorable compared with the LMS algorithm, recourse to these methods would be taken only if the meager instrumentation required was regarded as a cardinal advantage or nonunimodal performance surfaces were of concern. Furthermore, the Howells–Applebaum maximum signal-to-noise ratio (SNR) algorithm has a misadjustment versus convergence speed trade-off that is nearly identical with the LMS algorithm. Attention for the direct comparison consequently is focused on the following adaptive algorithms:

1. LMS error algorithm (Section 4.2 of Chapter 4)
2. Powell's accelerated gradient (PAG) algorithm (Section 4.4.1 of Chapter 4)
3. Direct matrix inversion (DMI) [Version (6.24) from Section 5.1.2 of Chapter 5]
4. Recursive (R) algorithm (Section 6.5 of Chapter 6)
5. Gram–Schmidt cascade preprocessor (GSCP) (Section 7.4 of Chapter 7)

Each algorithm starts with the same initial weight vector $\mathbf{w}^T(0) = [0.1, 0, 0, 0]$, for the array geometry and signal configuration of Figure 4.19 with signal conditions corresponding to an eigenvalue spread of 2440. The convergence results averaged over 10 random trials for each algorithm appear in Figures 9.1–9.5.

The results of Figures 9.1–9.5 show that the LMS algorithm requires about 1,750 data samples to converge within 3 dB of the optimum output SNR, the PAG algorithm requires 110 iterations (990 data samples), the DMI and recursive algorithms both require 8 data samples, and the GSCP algorithm requires 40 iterations (120 data samples). Algorithm parameters were selected so the degree of steady-state misadjustment for the LMS, PAG, and GSCP algorithms is comparable. Misadjustment of the DMI and

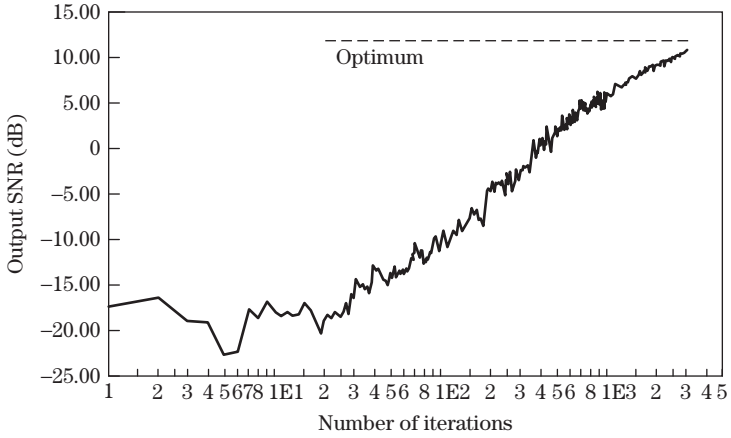


FIGURE 9-1 ■
Output SNR versus number of iterations for LMS algorithm with $\alpha_L = 0.1$.

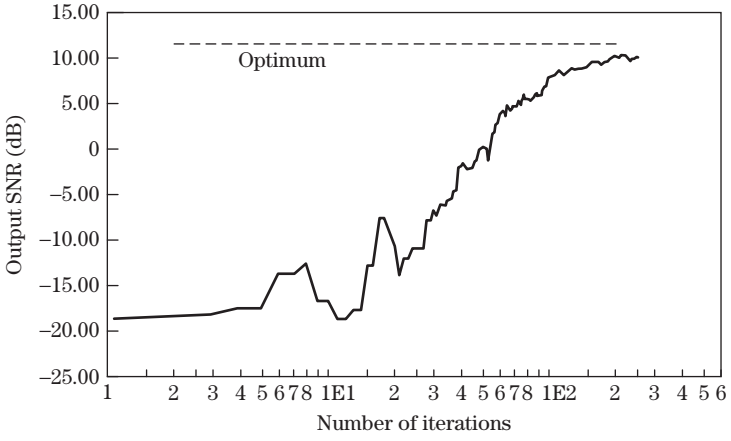


FIGURE 9-2 ■
Output SNR versus number of iterations for PAG algorithm with $K = 9$ samples per iteration.

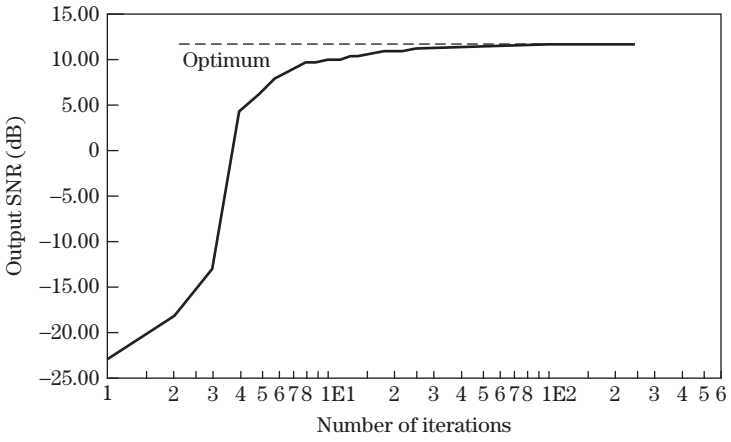


FIGURE 9-3 ■
Output SNR versus number of iterations for DMI algorithm.

recursive algorithms decreases as the number of iterations increases and cannot be modified by altering algorithm parameters. These results indicate that the DMI and recursive algorithms offer by far the best misadjustment versus speed of convergence trade-off, followed (in order) by the GSCP algorithm, the PAG algorithm, and the LMS algorithm for this moderate eigenvalue spread condition of $\lambda_{max}/\lambda_{min} = 2{,}440$.

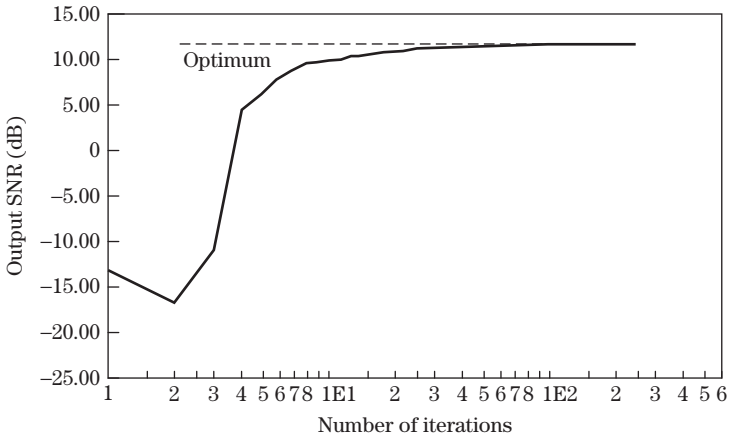


FIGURE 9-4 ■ Output SNR versus number of iterations for recursive algorithm with $\alpha = 1$ and $\mathbf{P}(0) = \mathbf{I}$.

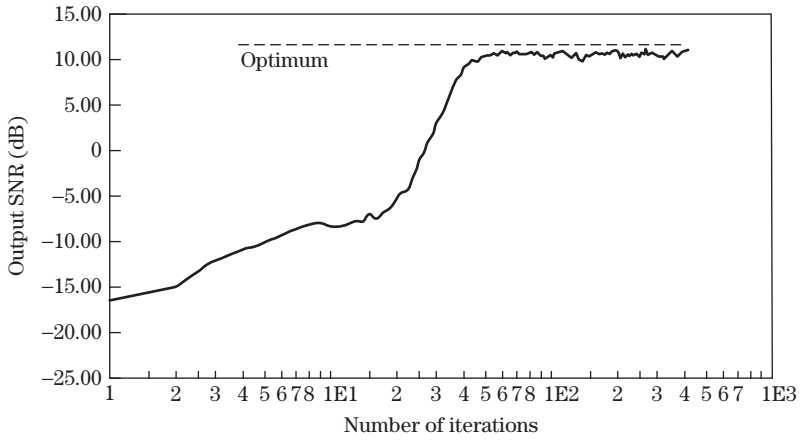


FIGURE 9-5 ■ Output SNR versus number of iterations for GSCP with $\alpha = \alpha_L = 0.1$ and $K = 3$ samples per iteration.

Table 9.1 summarizes the principal operational characteristics associated with the adaptive algorithms considered throughout Part 2. The algorithms that achieve the simplest possible instrumentation by requiring only direct measurement of the selected performance measure pay a severe penalty in terms of the increased convergence time required to reach the steady-state solution for a given degree of misadjustment. Accepting the instrumentation necessary to incorporate one correlator for each controlled array element enables the misadjustment versus convergence speed trade-off for the LMS and Howells–Applebaum interference suppression loops to be achieved. Further improvement in the misadjustment versus convergence speed trade-off is obtained where sensitivity to eigenvalue spread is a concern by paying the price of additional instrumentation or additional computational power. As the shift to digital processing continues, algorithms requiring more sophisticated computation become not only practicable but also preferable in many cases. Not only can high performance be achieved that was impractical before, but also the low cost of the increased computational power may in some cases render a sophisticated algorithm more economical.

The genetic algorithm (GA) is a "global" minimum seeker, so it is less likely to get stuck in a local minimum like the steepest descent algorithm. It minimizes the total output power, if the number of adaptive elements or the adaptive weight range are limited.

TABLE 9-1 ■ Operational Characteristics Summary of Selected Adaptive Algorithms

	LMS	MSNR	PAG	DSD	DMI	R	GSCP	RS	GA
Algorithm philosophy	Steepest descent	Similar to LMS	Conjugate gradient descent	Perturbation technique	Direct estimate of covariance; open loop	Data weighting similar to Kalman filter; closed loop	Orthogonalize input signals	Trial and error	Minimize total power or maximize SNR
Transient response characteristic	Misadjustment vs. convergence speed trade-off is acceptable for numerous applications	Nearly the same as LMS	More favorable misadjustment vs. convergence speed trade-off than LMS	Unfavorable misadjustment vs. convergence speed trade-off compared to LMS	Achieves the fastest convergence with most favorable misadjustment vs. convergence speed trade-off	Same as DMI	More favorable misadjustment vs. convergence speed trade-off than PAG. Convergence speed approaches DMI	Unfavorable misadjustment vs. convergence speed trade-off compared to LMS; accelerated steps improve speed	Limited by the hardware
Algorithm strengths	Easy to implement, requiring N correlators and integrators; tolerant of hardware errors	Same as LMS	Convergence speed less sensitive to eigenvalue spread than LMS; fast convergence for small number of degrees of freedom	Easy to implement; requires only instrumentation to directly measure the performance index	Very fast convergence speed independent of eigenvalue spread	Same as DMI; different data weighting schemes are easily incorporated	Convergence speed enjoys reduced sensitivity to eigenvalue spread compared with LMS; tolerant of hardware errors	Can be applied to any directly measurable performance index; easy to implement, with meager instrumentation and computation requirements	Minimal hardware, fast, does not get stuck in local minimum, independent of eigenvalue spread
Algorithm weaknesses	Convergence speed sensitive to eigenvalue spread	Same as LMS	Relatively difficult to implement and requires parallel processors; convergence speed sensitive to number of degrees of freedom	Rate of convergence sensitive to eigenvalue spread, with speed comparable to that of RS with accelerated step	Requires $N(N+1)/2$ correlators to implement; matrix inversion requires adequate precision and $N^3/2 + N^2$ complex multiplies	Requires $N(N+1)/2$ correlators and heavy computational load	Requires a respectable amount of hardware—$N(N+1)/2$ adaptive loops to implement	Convergence speed sensitive to eigenvalue spread and the slowest of all algorithms considered	Some measurements are bad each iteration
Chapter reference	4	4	4	4	5	6	7	8	8

Since it works with standard phased array hardware, it can be implemented on existing arrays and is much cheaper than requiring a receiver at each element. The convergence is relatively independent of the interference and signal power levels. Each iteration, the GA must evaluate the entire population, which can be small. As a result, there are bad measurements in addition to the improved power and SNR measurements. The transient response is limited by the hardware switching speed and settling time. It may be necessary to average a few power measurements to get the desired results.

PART III

Advanced Topics

CHAPTER

10

Compensation of Adaptive Arrays

Chapter Outline

Narrowband adaptive arrays need only one complex adaptive weight in each element channel. Broadband adaptive arrays, however, require tapped delay lines (transversal filters) in each element channel to make frequency-dependent amplitude and phase adjustments. The analysis presented so far assumes that each element channel has identical electronics and no reflected signals. Unfortunately, the electrical characteristics of each channel are slightly different and lead to "channel mismatching" in which significant differences in frequency-response characteristics from channel to channel may severely degrade an array's performance without some form of compensation. This chapter starts with an analysis of array errors and then addresses array calibration and frequency-dependent mismatch compensation using tapped delay line processing, which is important for practical broadband adaptive array designs.

The number of taps used in a tapped delay line processor depends on whether the tapped delay line compensates for broadband channel mismatch effects or for the effects of multipath and finite array propagation delay. Minimizing the number of taps required for a specified set of conditions is an important practical design consideration, since each additional tap (and associated weighs) increases the cost and complexity of the adaptive array system.

10.1 ARRAY ERRORS

Array errors result from the manufacturing tolerances defined by the materials, processes, and construction of the components in an array. These small errors are random, because the manufacturing techniques employed have very tight tolerances. The random differences between any components distort the signal path by adding phase and amplitude errors as well as noise to each signal. These types of errors are static, because once measured they remain relatively unchanged over the life of the component. Higher frequencies have tighter tolerances for phase distortion than lower frequencies, because the errors are a function of wavelength. Not only are the accuracy of the dimensions of the components important, but the accuracy of the values of the constitutive parameters of the components are also important. For instance, the dielectric constant determines the wavelength and hence the phase of the signal passing through it, so an error in the dielectric constant produces a phase error.

Dynamic errors change with time and are primarily due to changes in temperature. Online calibration corrects for these dynamic errors also takes care of any drift in the static errors. The dynamic errors are also frequency dependent. The effects of temperature are smallest at the center frequency and increase as the frequency migrates away from the center frequency.

10.1.1 Error Analysis

Random errors that affect arrays fall into four categories:

1. Random amplitude error, δ_n^a
2. Random phase error, δ_n^p
3. Random position error, δ_n^s
4. Random element failure, $P_n^e = \begin{cases} 1 & \text{element functioning properly} \\ 0 & \text{element failure} \end{cases}$

The first three types of random errors fit into the array factor as perturbations to the array weights and element locations

$$AF_{err} = \sum_{n=1}^{N} \left(a_n + \delta_n^a\right) e^{j\left(p_n + \delta_n^p\right)} e^{jk\left(s_n + \delta_n^s\right)u} \tag{10.1}$$

Element failures result when an element no longer transmits or receives. The probability that an element has failed, $1 - P_e$, is the same as a root mean square (rms) amplitude error, $\overline{\delta_n^{a^2}}$. Position errors are not usually a problem, so a reasonable formula to calculate the rms sidelobe level of the array factor for amplitude and phase errors with element failures is [1]

$$sll_{rms} = \frac{(1 - P_e) + \overline{\delta_n^{a^2}} + P_e \overline{\delta_n^{p^2}}}{P_e \left(1 - \overline{\delta_n^{p^2}}\right) \eta_t N} \tag{10.2}$$

Figure 10-1 is an example of a typical corporate-fed array. A random error that occurs at one element is statistically uncorrelated with a random error that occurs in another element in the array as long as that error occurs after the last T junction and before an

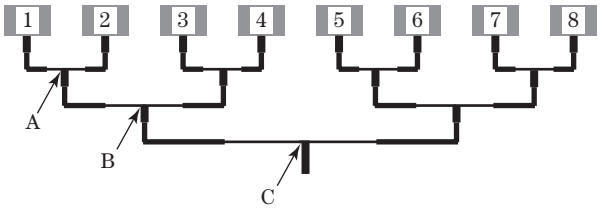


FIGURE 10-1 ■ Corporate-fed array with random errors.

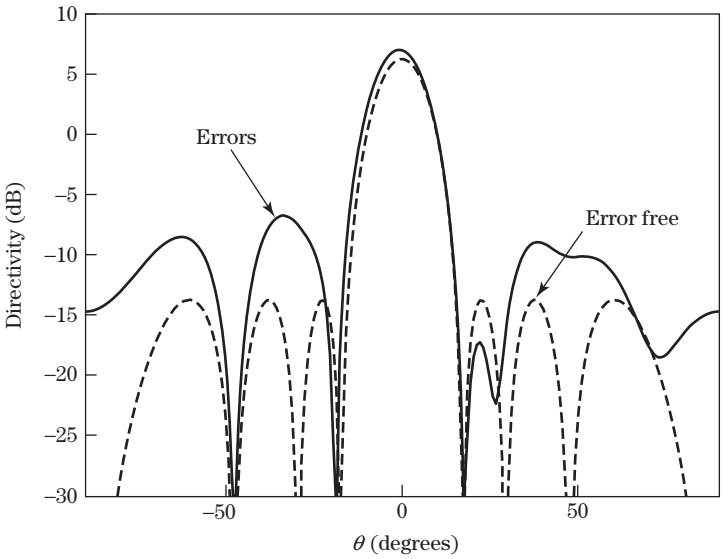


FIGURE 10-2 ■ Array factor with random, uncorrelated errors superimposed on the error-free array factor.

element. If a random error occurs prior to A, for instance, then the random error becomes correlated between the elements that share the error. For instance, a random error between A and B results in a random correlated error shared by elements 1 and 2. Likewise, a random error between B and C results in a random correlated error shared by elements 1, 2, 3, and 4.

As an example, consider an eight-element, 20 dB Chebyshev array that has elements spaced $\lambda/2$ apart. If the random errors are represented by $\delta_n^a = 0.15$ and $\delta_n^p = 0.15$, then an example of the array factor with errors is shown in Figure 10-2. Note that the random errors lower the main beam directivity, induce a slight beam-pointing error, increase the sidelobe levels, and fill in some of the nulls.

10.1.2 Quantization Errors

Phase shifters and attenuators have N_{bp} control bits with the least significant bits given by

$$\Delta_a = 2^{-N_{ba}} \tag{10.3}$$

$$\Delta_p = 2\pi \times 2^{-N_{bp}} \tag{10.4}$$

If the difference between the desired and quantized amplitude weights is a uniformly distributed random number with the bounds being the maximum amplitude error of $\pm\Delta_a/2$, then the rms amplitude error is $\delta_n^a = \Delta_a/\sqrt{12}$. The quantization error is random only when no two adjacent elements receive the same quantized phase shift. The difference

between the desired and quantized phase shifts is treated as uniform random variables between $\pm\Delta_p/2$. As with the amplitude error, the random phase error formula in this case is $\delta_n^p = \Delta_p/\sqrt{12}$. Substituting this error into (b) yields the rms sidelobe level.

The phase quantization errors become correlated when the beam steering phase shift is small enough that groups of adjacent elements have their beam steering phase quantized to the same level. This means that N/N_Q subarrays of N_Q elements receive the same phase shift. The grating lobes due to these subarrays occur at [2]

$$\sin\theta_m = \sin\theta_s \pm \frac{m\lambda}{N_Q d_e} = \sin\theta_s\left[1 \pm \frac{m\,(N-1)\,2^{N_{bp}}}{N}\right] \simeq \sin\theta_s\left(1 \pm m2^{N_{bp}}\right) \quad (10.5)$$

The approximation in (10.5) assumes that the array has many elements. For large scan angles, quantization lobes do not form, because the element-to-element phase difference appears random. The relative peaks of the quantization lobes are given by [1]

$$AF_N^{QL} = \frac{1}{2^{N_p}}\sqrt{\frac{\sqrt{1-\sin\theta^2}}{\sqrt{1-\sin\theta_s^2}}} \quad (10.6)$$

Figure 10-3 shows an array factor with a 20 dB $\overline{n} = 3$ Taylor amplitude taper for a 20-element, $d = 0.5\lambda$ array with its beam steered to $\theta = 3^\circ$ when the phase shifters have three bits. Four quantization lobes appear. The quantization lobes decrease when higher-precision phase shifters are used and when the beam is steered to higher angles.

Significant distortion also results from mutual coupling, variation in group delay between filters, differences in amplifier gain, tolerance in attenuator accuracy, and aperture jitter in a digital beamforming array. Aperture jitter is the timing error between samples in an analog-to-digital (A/D) converter. Without calibration, beamforming or estimation of the direction of arrival (DOA) of the signal is difficult, as the internal distortion is uncorrelated with the signal. As a result, the uncorrelated distortion changes the weights at each element and therefore distorts the array pattern.

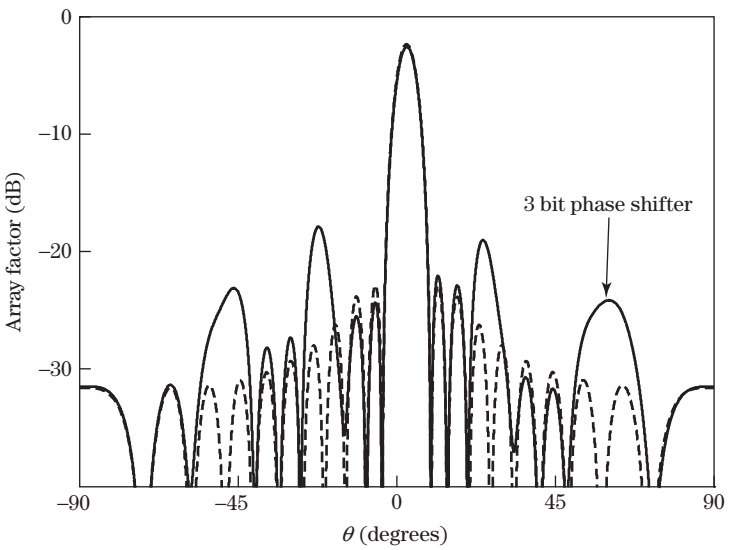


FIGURE 10-3 ■ Array factor steered to 3 degrees with three-bit phase shifters compared with phase shifters with infinite precision.

10.2 ARRAY CALIBRATION

A phased array needs calibrated before it can generate an optimum coherent beam. Calibration involves tuning, for example, the phase shifters, attenuators, or receivers to maximize the gain and to create the desired sidelobe response. Offline calibration takes care of the static errors and is done at the factory or on deployment. Narrowband calibration is applied at the center frequency of operation. Broadband calibration is applied over the whole operating bandwidth of the array. The calibrated phase settings are stored for all beam steering angles. Temperature causes drift in the component characteristics over time, so the array requires periodic recalibration. The gain of the radiofrequency (RF) channels must be accurately controlled to avoid nonlinearities arising from saturation of components, because these nonlinearities cannot be removed.

The top vector in Figure 10-4 shows the resulting uncalibrated array output when the individual five-element vectors have random amplitude and phase errors. When the array is calibrated (bottom vector in Figure 10-4), then the individual element vectors are the same length and align. As a result, the calibrated array output vector magnitude is maximized, and its phase is zero. Methods for performing array calibration use a calibrated source, signal injection, or near-field scanning. These approaches are discussed in the following sections.

10.2.1 Calibrated Source

A known calibration source radiates a calibration signal to all elements in the array [3]. Figure 10-5 shows a calibration source in the far field of an array. At regular intervals, the main beam is steered to receive the calibration source signal. Alternatively, a multibeam antenna can devote one beam to calibration. Calibration with near-field sources requires that distance and angular differences be taken into account. If the calibration source is in the far field, then the phase shifters are set to steer the beam in the direction of the source. In either case, each element toggles through all of its phase settings until the output signal is maximized. The difference between the steering phase and the phase that yields the maximum signal is the calibration phase.

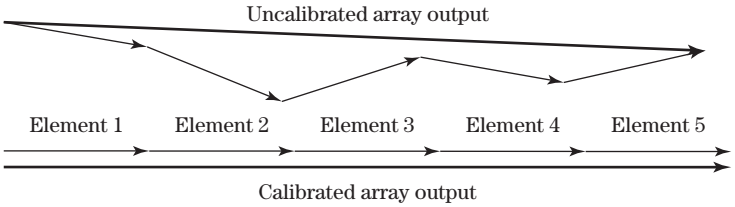


FIGURE 10-4 ■ The uncalibrated array output is less than the calibrated array output, because errors in the uncalibrated array do not allow the signal vectors from the elements to align.

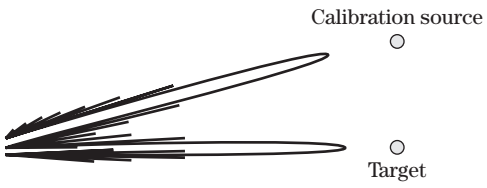


FIGURE 10-5 ■ Far-field calibration.

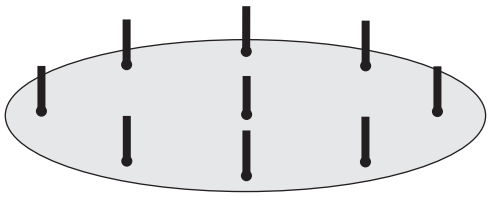


FIGURE 10-6 ■
Layout of the smart antenna test bed.

Making power measurements for every phase setting at every element in an array is extremely time-consuming. Calibration techniques that measure both amplitude and phase of the calibrated signal tend to be much faster. Accurately measuring the signal phase is reasonable in an anechoic chamber but difficult in the operational environment. Measurements at four orthogonal phase settings yield sufficient information to obtain a maximum likelihood estimate of the calibration phase [4]. The element phase error is calculated from power measurements at the four phase states, and the procedure is repeated for each element in the array. Additional measurements improve signal-to-noise ratio, and the procedure can be repeated to achieve desired accuracy within resolution of the phase shifters, since the algorithm is intrinsically convergent.

Another approach uses amplitude-only measurements from multiple elements to find the complex field at an element [5]. The first step measures the power output from the array when the phases of multiple elements are successively shifted with the different phase intervals. Next, the measured power variation is expanded into a Fourier series to derive the complex electric field of the corresponding elements. The measurement time reduction comes at the expense of increased measurement error.

Transmit/receive module calibration is an iterative process that starts with adjusting the attenuators for uniform gain at the elements [6]. The phase shifters are then adjusted to compensate for the insertion phase differences at each element. Ideally, when calibrating the array, the phase shifter's gain remains constant as the phase settings are varied, but the attenuator's insertion phase can vary as a function of the phase setting. This calibration should be done across the bandwidth, range of operating temperatures, and phase settings. If the phase shifter's gain varies as a function of setting, then the attenuators need to be compensated as well. After iterating over this process, all the calibration settings are saved and applied at the appropriate times.

Figure 10-6 shows an eight-element uniform circular array (UCA) in which a center element radiates a calibration signal to the other elements in the array [7]. Since the calibration source is in the center of the array, the signal path from the calibration source to each element is identical. As previously noted, random errors are highly dependent on temperature [8]. An experimental model of the UCA in Figure 10-6 was placed inside a temperature-controlled room and calibrated at 20°C. The measured amplitude and phase errors at three temperatures are shown in Figure 10-7 and Figure 10-8, respectively. Increasing the temperature of the room to 25°C then to 30°C without recalibration increases the errors shown in Figure 10-7 and Figure 10-8. This experiment demonstrates the need of dynamic calibration in a smart antenna array.

10.2.2 Signal Injection

Calibrating with a radiating source is difficult, because the calibration signal transmission/reception depends on the environment. One technique commonly used in digital

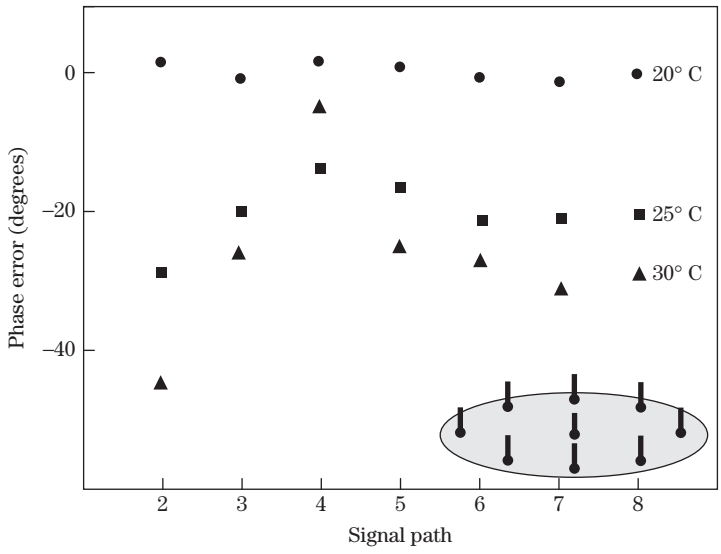


FIGURE 10-7 ■ Amplitude error for the UCA antenna as the system temperature changes from 20°C with calibration to 25°C without recalibration and to 30°C without recalibration.

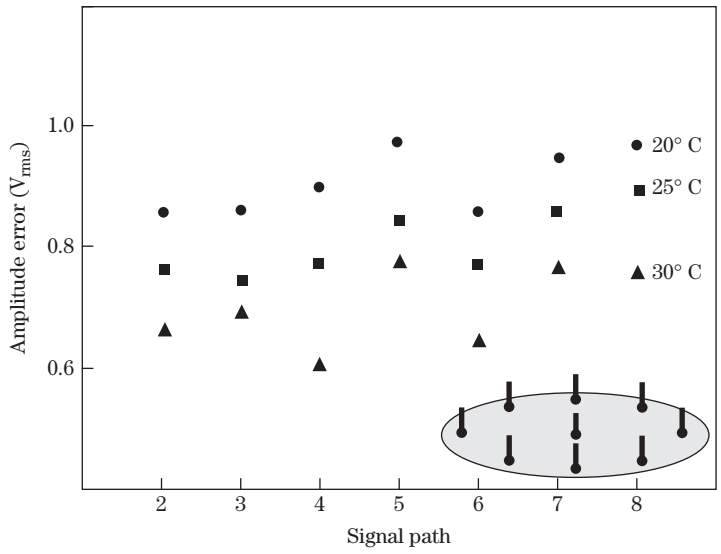


FIGURE 10-8 ■ Phase error for the UCA antenna as the system temperature changes from 20°C with calibration to 25°C without recalibration and to 30°C without recalibration.

beamforming arrays is injecting a calibration signal into the signal path of each element in the array behind each element as shown in Figure 10-9 [9]. This technique provides a high-quality calibration signal for the circuitry behind the element. Unfortunately, it does not calibrate for the element patterns that have significant variations due to mutual coupling, edge effects, and multipath.

10.2.3 Near-Field Scan

A planar near-field scanner positioned very close to the array moves a probe directly in front of each element to measure the amplitude and phase of all the elements [10]. The measured field is transformed back to the aperture to recreate the field radiated at each

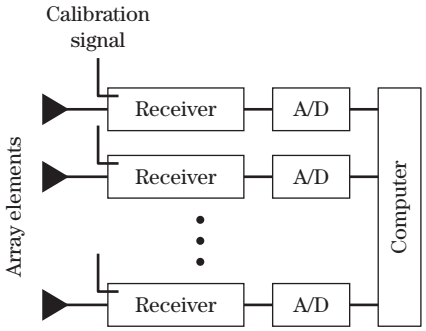


FIGURE 10-9 ■ Inserting a calibration signal into the signal paths in a digital beamformer.

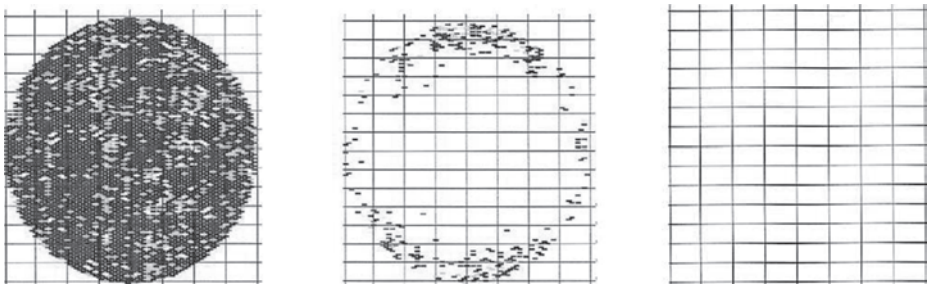


FIGURE 10-10 ■ Alignment results (measured phase deviation from desired value). a: Unaligned. b: After single alignment with uncorrected measurements. c: After alignment with fully corrected measurements. From W. T. Patton and L. H. Yorinks, "Near-field alignment of phased-array antennas," *IEEE Transactions on Antennas and Propagation*, Vol. 47, No. 3, March 1999, pp. 584–591.

element. The calibration algorithm iterates between the measured phase and the array weights until the phase at all the elements is the same. Figure 10-10 shows the progression of the phase correction algorithm from left to right. The picture on the left is uncalibrated, the center picture is after one iteration, and the picture on the right is after calibration is completed. This techniques is exceptionally good at correcting static errors prior to deploying an antenna is not practical for dynamic errors.

10.3 | BROADBAND SIGNAL PROCESSING CONSIDERATIONS

Broadband arrays use tapped delay lines that have frequency-dependent transfer functions. Array performance is a function of the number of taps, the tap spacing, and the total delay in each channel. The minimum number of taps required to obtain satisfactory performance for a given bandwidth may be determined as discussed in Section 2.5. The discussion of broadband signal processing considerations given here follows the treatment of this subject given by Rodgers and Compton [11–13]. The ideal (distortionless) channel transfer functions are derived; adaptive array performance using quadrature hybrid processing and two-, three-, and five-tap delay line processing are considered; and results and conclusions for broadband signal processing are then discussed.

10.3.1 Distortionless Channel Transfer Functions

The element channels of the two-element array in Figure 10-11 are represented by the transfer functions $H_1(\omega)$ and $H_2(\omega)$. Let the desired signal arrive at θ_s, measured relative to the array face normal. The array carrier frequency is ω_0, and the point sources spacing is $d = \lambda_0/2 = \pi b/\omega_0$, where $\mathfrak{v}$ is the wavefront propagation velocity.

From the point of view of the desired signal, the overall transfer function encountered in passing through the array of Figure 10-11 is

$$H_d(\omega) = H_1(\omega) + H_2(\omega) \exp\left(-j\frac{\omega d}{\mathfrak{v}} \sin\theta_s\right) \tag{10.7}$$

and the overall transfer function seen by the interference signal is

$$H_I(\omega) = H_1(\omega) + H_2(\omega) \exp\left(-j\frac{\omega d}{\mathfrak{v}} \sin\theta_i\right) \tag{10.8}$$

Now require that

$$H_d(\omega) = \exp(-j\omega T_1) \tag{10.9}$$

and

$$H_I(\omega) = 0 \tag{10.10}$$

By choosing $H_d(\omega)$ according to (10.9), the desired signal is permitted to experience a time delay T_1 in passing through the array but otherwise remains undistorted. Choosing $H_I(\omega) = 0$ results in complete suppression of the interference signal from the array output.

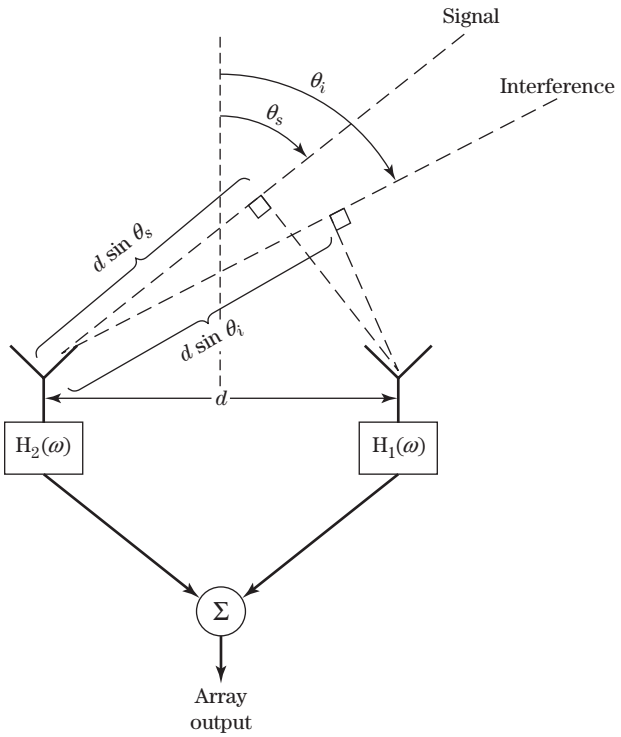


FIGURE 10-11 ■ Two-element array.

To determine whether it is possible to select $H_1(\omega)$ and $H_2(\omega)$ to satisfy (10.9) and (10.10), solve (10.9) and (10.10) for $H_1(\omega)$ and $H_2(\omega)$. Setting $H_1(\omega) = |H_1(\omega)| \exp[j\alpha_1(\omega)]$ and $H_2(\omega) = |H_2(\omega)| \exp[j\alpha_2(\omega)]$ results in

$$|H_1(\omega)| \exp[j\alpha_1(\omega)] + |H_2(\omega)| \exp\left\{j\left[\alpha_2(\omega) - \frac{\pi\omega}{\omega_0}\sin\theta_s\right]\right\} = \exp(-j\omega T_1) \quad (10.11)$$

$$|H_1(\omega)| \exp[j\alpha_1(\omega)] + |H_2(\omega)| \exp\left\{j\left[\alpha_2(\omega) - \frac{\pi\omega}{\omega_0}\sin\theta_i\right]\right\} = 0 \quad (10.12)$$

To satisfy (10.9) and (10.10), it follows from (10.11) and (10.12) (as shown by the development outlined in the Problems section) that

$$H_1(\omega) = H_2(\omega) = \frac{1}{\sqrt{2\left(1 - \cos\left[\frac{\pi\omega}{\omega_0}(\sin\theta_i - \sin\theta_s)\right]\right)}} \quad (10.13)$$

$$\alpha_2(\omega) = \frac{\pi}{2}\left(\frac{\omega}{\omega_0}\right)[\sin\theta_s + \sin\theta_i] \mp n\frac{\pi}{2} - \omega T_1 \quad (10.14)$$

$$\alpha_1(\omega) = \frac{\pi}{2}\left(\frac{\omega}{\omega_0}\right)[\sin\theta_s - \sin\theta_i] \pm n\frac{\pi}{2} - \omega T_1 \quad (10.15)$$

where n is any odd integer. This result means that the amplitude of the ideal transfer functions are equal and frequency dependent. Equations (10.14) and (10.15) furthermore show that the phase of each filter is a linear function of frequency with the slope dependent on the spatial arrival angles of the signals as well as on the time delay T_1 of the desired signal.

Plots of the amplitude function in (10.13) are shown in Figure 10-12 for two choices of arrival angles ($\theta_s = 0^\circ$ and $\theta_s = 80^\circ$), where it is seen that the amplitude of the distortionless transfer function is nearly flat over a 40% bandwidth when the desired signal is at broadside ($\theta_s = 0^\circ$) and the interference signal is 90° from broadside ($\theta_i = 90^\circ$). Examination of (10.13) shows that whenever $(\sin\theta_I - \sin\theta_s)$ is in the neighborhood of ± 1,

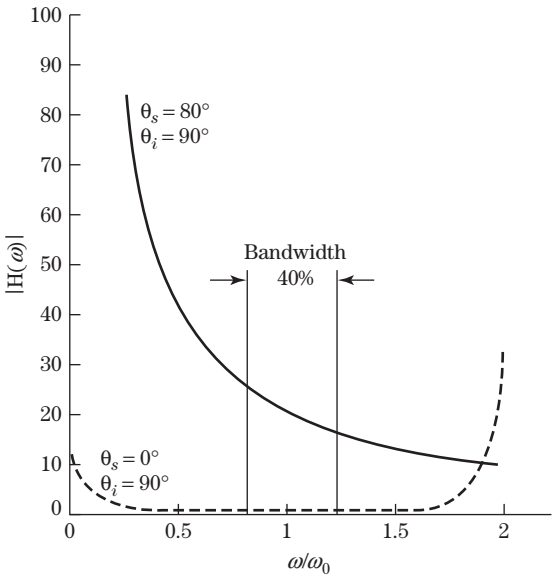


FIGURE 10-12 ■ Distortionless transfer function amplitude versus normalized frequency for $d = \lambda_0/2$. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

then the resulting amplitude function will be nearly flat over the 40% bandwidth region. If, however, both the desired and interference signals are far from broadside (as when $\theta_d = 80^\circ$ and $\theta_i = 90^\circ$), then the amplitude function is no longer flat.

The degree of "flatness" of the distortionless filter amplitude function is interpreted in terms of the signal geometry with respect to the array sensitivity pattern. In general, when the phases of $H_1(\omega)$ and $H_2(\omega)$ are adjusted to yield the maximum undistorted response to the desired signal, the corresponding array sensitivity pattern will have certain nulls. The distortionless filter amplitude function is then the most flat when the interference signal falls into one of these pattern nulls.

Equation (10.13) furthermore shows that singularities occur in the distortionless channel transfer functions whenever $(\omega/\omega_0)\pi(\sin\theta_i - \sin\theta_s) = n2\pi$ where $n = 0, 1, 2, \ldots$. The case when $n = 0$ occurs when the desired and interference signals arrive from exactly the same direction, so it is hardly surprising that the array would experience difficulty trying to receive one signal while nulling the other in this case. The other cases when $n = 1, 2, \ldots$, occur when the signals arrive from different directions, but the phase shifts between elements differ by a multiple of 2π radians at some frequency ω in the signal band.

The phase functions $\alpha_1(\omega)$ and $\alpha_2(\omega)$ of (10.14) and (10.15) are linear functions of frequency. When $T_1 = 0$, the phase slope of $H_1(\omega)$ is proportional to $\sin\theta_s - \sin\theta_i$, whereas that of $H_2(\omega)$ is proportional to $\sin\theta_i + \sin\theta_s$. Consequently, when the desired signal is broadside, $\alpha_1(\omega) = -\alpha_2(\omega)$. Furthermore, the phase difference between $\alpha_1(\omega)$ and $\alpha_2(\omega)$ is also a linear function of frequency, a result that would be expected since this allows the interelement phase shift (which is also a linear function of frequency) to be canceled.

10.3.2 Quadrature Hybrid and Tapped Delay Line Processing for a Least Mean Squares Array

Consider a two-element adaptive array using the least mean squares (LMS) algorithm. If $\mathbf{w}$ is the column vector of array weights, $\mathbf{R}_{xx}$ is the correlation matrix of input signals to each adaptive weight, and $\mathbf{r}_{xd}$ is the cross-correlation vector between the received signal vector $\mathbf{x}(t)$ and the reference signal $d(t)$, then as shown in Chapter 3 the optimum array weight vector that minimizes $E\{\varepsilon^2(t)\}$ (where $\varepsilon(t) = d(t)$—array output) is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{10.16}$$

If the signal appearing at the output of each sensor element consists of a desired signal, an interference signal, and a thermal noise component (where each component is statistically independent of the others and has zero mean), then the elements of $\mathbf{R}_{xx}$ can readily be evaluated in terms of these component signals.

Consider the tapped delay line employing real (instead of complex) weights shown in Figure 10-13. Since each signal $x_i(t)$ is just a time-delayed version of $x_1(t)$, it follows that

$$\left.\begin{aligned} x_2(t) &= x_1(t - \Delta) \\ x_2(t) &= x_1(t - 2\Delta) \\ &\vdots \\ x_L(t) &= x_1[t - (L-1)\Delta] \end{aligned}\right\} \tag{10.17}$$

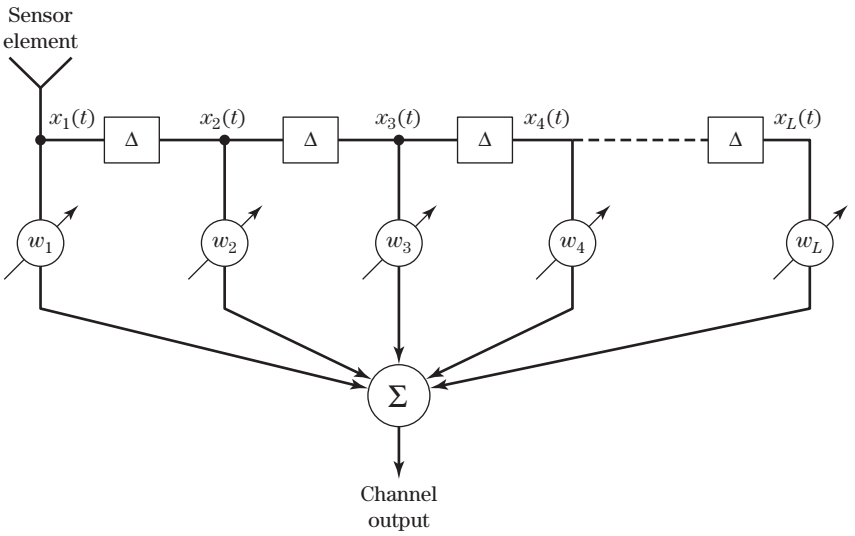


FIGURE 10-13 ■ Tapped delay line processor for a single-element channel having real adaptive weights.

Now since the elements of $\mathbf{R}_{xx}$ are given by

$$r_{x_i x_j} \triangleq E\{x_i(t)x_j(t)\} \tag{10.18}$$

it follows from (10.17) that

$$r_{x_i x_j} = r_{x_1 x_1}(\tau_{ij}) \tag{10.19}$$

where $r_{x_1 x_1}(\tau_{ij})$ is the autocorrelation function of $x_1(t)$, and τ_{ij} is the time delay between $x_i(t)$ and $x_j(t)$. Furthermore, $r_{x_i x_i}(\tau_{ij})$ is the sum of three autocorrelation functions—those of the desired signal, the interference, and the thermal noise so that

$$r_{x_1 x_1}(\tau_{ij}) = r_{dd}(\tau_{ij}) + r_{II}(\tau_{ij}) + r_{nn}(\tau_{ij}) \tag{10.20}$$

For the elements of $\mathbf{R}_{xx}$ corresponding to $x_i(t)$ and $x_j(t)$ from different element channels, $r_{x_i x_j}$ consists only of the sum of the autocorrelation functions of the desired signal and the interference signal (with appropriate delays) but not the thermal noise since the element noise from channel to channel is uncorrelated. Thus, for signals in different element channels

$$r_{x_i x_j}(\tau_{ij}) = r_{dd}(\tau_{d_{ij}}) + r_{II}(\tau_{I_{ij}}) \tag{10.21}$$

where $\tau_{d_{ij}}$ denotes the time delay between $x_i(t)$ and $x_j(t)$ for the desired signal, and $\tau_{I_{ij}}$ denotes the time delay between $x_i(t)$ and $x_j(t)$ for the interference signal (these two time delays will in general be different due to the different angles of arrival of the two signals). Only when $x_i(t)$ and $x_j(t)$ are from the same array element channel will $\tau_{d_{ij}} = \tau_{I_{ij}}$ (which may then be denoted by τ_{i_j}).

Next, consider the quadrature hybrid array processor depicted in Figure 10-14. Let $x_1(t)$ and $x_3(t)$ denote the in-phase signal components and $x_2(t)$ and $x_4(t)$ denote the quadrature-phase signal components of each of the elements output signals. Then the in-phase and quadrature components are related by

$$\left.\begin{aligned} x_2(t) &= \check{x}_1(t) \\ x_4(t) &= \check{x}_3(t) \end{aligned}\right\} \tag{10.22}$$

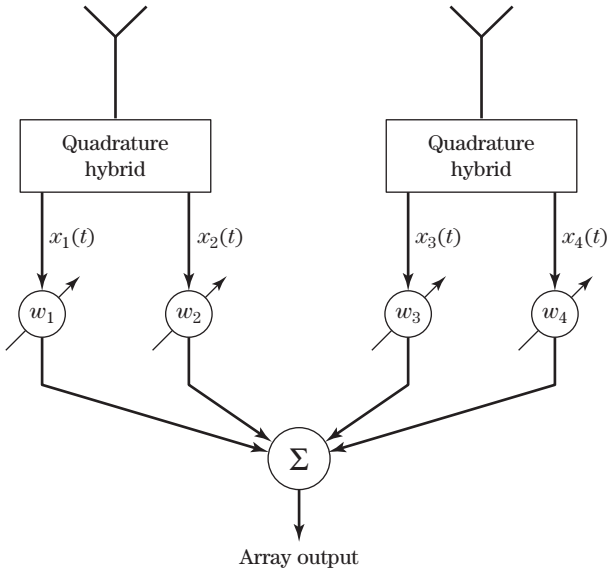


FIGURE 10-14 ■ Quadrature hybrid processing for a two-element array.

The symbol ˇ denotes the Hilbert transform

$$\check{x}(t) \triangleq \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau \tag{10.23}$$

where the previous integral is regarded as a Cauchy principal value integral. The various elements of the correlation matrix

$$r_{x_i x_j} = E\{x_i(t)x_j(t)\} \tag{10.24}$$

can then be found by making use of certain Hilbert transform relations as follows [14,15]:

$$E\{\check{x}(t)\check{y}(s)\} = E\{x(t)y(s)\} \tag{10.25}$$
$$E\{\check{x}(t)y(s)\} = -E\{x(t)\check{y}(s)\} \tag{10.26}$$

so that

$$E\{\check{x}(t)x(t)\} = 0 \tag{10.27}$$
$$E\{x(t)\check{y}(s)\} = \check{E}\{x(t)y(s)\} \tag{10.28}$$

where $\check{E}\{x(t)y(s)\}$ denotes the Hilbert transform of $r_{xy}(\tau)$ where $\tau = s - t$. With the previous relations and from (10.22) it then follows that

$$r_{x_1x_1} = E\{x_1(t)x_1(t)\} = r_{x_1x_1}(0) \tag{10.29}$$
$$r_{x_1x_2} = E\{x_1(t)x_2(t)\} = E\{x_1(t)\check{x}_1(t)\} = 0 \tag{10.30}$$
$$r_{x_2x_2} = E\{x_2(t)x_2(t)\} = E\{\check{x}_1(t)\check{x}_1(x)\} \tag{10.31}$$
$$= E\{x_1(t)x_1(t)\} = r_{x_1x_1}(0)$$

where $r_{x_1x_1}(\tau)$ is the autocorrelation function of $x_1(t)$ given by (10.20).

When two different sensor element channels are involved [as with $x_1(t)$ and $x_3(t)$, for example], then

$$E\{x_1(t)\, x_3(t)\} = r_{dd}(\tau_{d_{13}}) + r_{II}(\tau_{I_{13}}) \tag{10.32}$$

where $\tau_{d_{13}}$ and $\tau_{I_{13}}$ represent the spatial time delays between the sensor elements of Figure 10-14 for the desired and interference signals, respectively. Similarly

$$\begin{aligned} E\{x_1(t)x_4(t)\} = E\{x_1(t)\check{x}_3(t)\} &= \check{E}\{x_1(t)x_3(t)\} \\ &= \check{r}_{dd}(\tau_{d_{13}}) + \check{r}_{II}(\tau_{I_{13}}) \qquad (10.33) \\ E\{x_2(t)x_3(t)\} = E\{\check{x}_1(t)x_3(t)\} &= -E\{\check{x}_1(t)x_3(t)\} \\ &= -\check{E}\{x_1(t)x_3(t)\} = -\check{r}_{dd}(\tau_{d_{13}}) - \check{r}_{II}(\tau_{I_{13}}) \qquad (10.34) \\ E\{x_2(t)x_4(t)\} = E\{\check{x}_1(t)\check{x}_3(t)\} &= x\{x_1(t)x_3(t)\} \\ &= r_{dd}(\tau_{d_{13}}) + r_{II}(\tau_{I_{13}}) \qquad (10.35) \end{aligned}$$

Now consider the cross-correlation vector $\mathbf{r}_{xd}$ defined by

$$\mathbf{r}_{xd} \triangleq E\begin{bmatrix} x_1(t)d(t) \\ x_2(t)d(t) \\ \vdots \\ x_{2N}(t)d(t) \end{bmatrix} \tag{10.36}$$

where N is the number of sensor elements. Each element of $\mathbf{r}_{xd}$, denoted by r_{x_id}, is just the cross-correlation between the reference signal $d(t)$ and signal $x_i(t)$. Since the reference signal is just a replica of the desired signal and is statistically independent of the interference and thermal noise signals, the elements of $\mathbf{r}_{xd}$ consist only of the autocorrelation function of the desired signal so that

$$r_{x_id} = E\{x_i(t)d(t)\} = r_{dd}(\tau_{d_i}) \tag{10.37}$$

where τ_{d_i} represents the time delay between the reference signal and the desired signal component of $x_i(t)$. For an array with tapped delay line processing, each element of $\mathbf{r}_{xd}$ is the autocorrelation function of the desired signal evaluated at a time-delay value that reflects both the spatial delay between sensor elements and the delay line delay to the tap of interest. For an array with quadrature hybrid processing, the elements of $\mathbf{r}_{xd}$ corresponding to an in-phase channel yield the autocorrelation function of the desired signal evaluated at the spatial delay appropriate for that element as follows:

$$r_{x_id}(\text{in-phase channel}) = E\{x_i(t)d(t)\} = r_{dd}(\tau_{d_i}) \tag{10.38}$$

The elements of $\mathbf{r}_{xd}$ corresponding to quadrature-phase channels can be evaluated using (10.27) and (10.28) as follows:

$$\begin{aligned} r_{x_{i+1}d}(\text{quadrature-phase channel}) &= E\{x_{i+1}(t)d(t)\} \\ &= E\{\check{x}_i(t)d(t)\} = -E\{x_i(t)\check{d}(t)\} \qquad (10.39) \\ &= -\check{E}\{x_i(t)d(t)\} = -\check{r}_{x_id}(\tau_{d_i}) \end{aligned}$$

Once $\mathbf{R}_{xx}$ and $\mathbf{r}_{xd}$ have been evaluated for a given signal environment, the optimal LMS weights can be computed from (10.16), and the steady-state response of the entire array can then be evaluated.

The tapped delay line in the element channel of Figure 10-13 has a channel transfer function given by

$$H_1(\omega) = w_1 + w_2 e^{-j\omega\Delta} + w_3 e^{-j2\omega\Delta} + \ldots + w_L e^{-j(L-1)\omega\Delta} \tag{10.40}$$

Likewise, the quadrature hybrid processor of Figure 10-14 has a channel transfer function

$$H_1(\omega) = w_1 - jw_2 \tag{10.41}$$

The array transfer function for the desired signal and the interference accounts for the effects of spatial delays between array elements. A two-element array transfer function for the desired signal is

$$H_d(\omega) = H_1(\omega) + H_2(\omega)e^{-j\omega\tau_d} \tag{10.42}$$

whereas the transfer function for the interference is

$$H_I(\omega) = H_1(\omega) + H_2(\omega)\, e^{-j\omega\tau_I} \tag{10.43}$$

The spatial time delays associated with the desired and interference signals are represented by τ_d and τ_I, respectively, between element 1 [with channel transfer function $H_1(\omega)$] and element 2 [with channel transfer function $H_2(\omega)$]. With two sensor elements spaced apart by a distance d as in Figure 10-11, the two spatial time delays are given by

$$\tau_d = \frac{d}{\mathfrak{v}} \sin\theta_s \tag{10.44}$$

$$\tau_I = \frac{d}{\mathfrak{v}} \sin\theta_I \tag{10.45}$$

The output signal-to-total-noise ratio is defined as

$$\text{SNR} \triangleq \frac{P_d}{P_I + P_n} \tag{10.46}$$

where P_d, P_I, and P_n represent the output desired signal power, interference signal power, and thermal noise power, respectively. The array output power for each of the foregoing three signals may now be evaluated. Let $\phi_{dd}(\omega)$ and $\phi_{II}(\omega)$ represent the power spectral densities of the desired signal and the interference signal, respectively; then the desired signal output power is given by

$$P_d = \int_{-\infty}^{\infty} \phi_{dd}(\omega)|H_d(\omega)|^2 d\omega \tag{10.47}$$

where $H_d(\omega)$ is the overall transfer function seen by the desired signal, and the interference signal output power is

$$P_I = \int_{-\infty}^{\infty} \phi_{II}(\omega)|H_1(\omega)|^2 d\omega \tag{10.48}$$

where $H_I(\omega)$ is the overall transfer function seen by the interference signal. The thermal noise present in each element output is statistically independent from one element to the next. Let $\phi_{nn}(\omega)$ denote the thermal noise power spectral density; then the noise power

contributed to the array output by element 1 is

$$P_{n_1} = \int_{-\infty}^{\infty} \phi_{nn}(\omega)|H_1(\omega)|^2 d\omega \tag{10.49}$$

whereas that contributed by element 2 is

$$P_{n_2} = \int_{-\infty}^{\infty} \phi_{nn}(\omega)|H_2(\omega)|^2 d\omega \tag{10.50}$$

Consequently, the total thermal noise output power from a two-element array is

$$P_n = \int_{-\infty}^{\infty} \phi_{nn}(\omega)[|H_1(\omega)|^2 + |H_2(\omega)|^2]\, d\omega \tag{10.51}$$

The foregoing expressions may now be used in (10.46) to obtain the output signal-to-total-noise ratio.

10.3.3 Performance Comparison of Four Array Processors

In this subsection, four adaptive arrays—one with quadrature hybrid processing and three with tapped delay line processing (using real weights)—are compared for signal bandwidths of 4, 10, 20, and 40%. Tapped delay lines use real weights to preserve as much simplicity as possible in the hardware implementation, although this sacrifices the available degrees of freedom with a consequent degradation in tapped delay line performance relative to combined amplitude and phase weighting. The results obtained will nevertheless serve as an indication of the relative effectiveness of tapped delay line processing compared with quadrature hybrid processing for broadband signals.

The four array processors to be compared are shown in Figure 10-15, where each array has two sensor elements and the elements are spaced one-half wavelength apart at the center frequency of the desired signal bandwidth. Figure 10-15a shows an array having quadrature hybrid processing, whereas Figure 10-15b–10-15d exhibit tapped delay line processing. The processor of Figure 10-15b has one delay element corresponding to one-quarter wavelength at the center frequency and two associated taps. The processor of Figure 10-15c has two delay elements, each corresponding to one-quarter wavelength at the center frequency, and three associated taps. The processor of Figure 10-15d has four delay elements, each corresponding to one-eighth wavelength at the center frequency, and five associated taps. Note that the total delay present in the tapped delay line of Figure 10-15d is the same as that of Figure 10-15c, so the processor in Figure 10-15d may be regarded as a more finely subdivided version of the processor in Figure 10-15c.

Assume that the desired signal is biphase modulated of the form

$$s_d(t) = A\cos[\omega_0 t + \phi(t) + \theta] \tag{10.52}$$

where $\phi(t)$ denotes a phase angle that is either zero or π over each bit interval, and θ is an arbitrary constant phase angle (within the range $[0, 2\pi]$) for the duration of any signal pulse. The nth bit interval is defined over $T_0 + (n-1)T \le t \le T_0 + nT$, where n is any integer, T is the bit duration, and T_0 is a constant that determines where the bit transitions occur, as shown in Figure 10-16.

Assume that $\phi(t)$ is statistically independent over different bit intervals and is zero or π with equal probability and that T_0 is uniformly distributed over one bit interval;

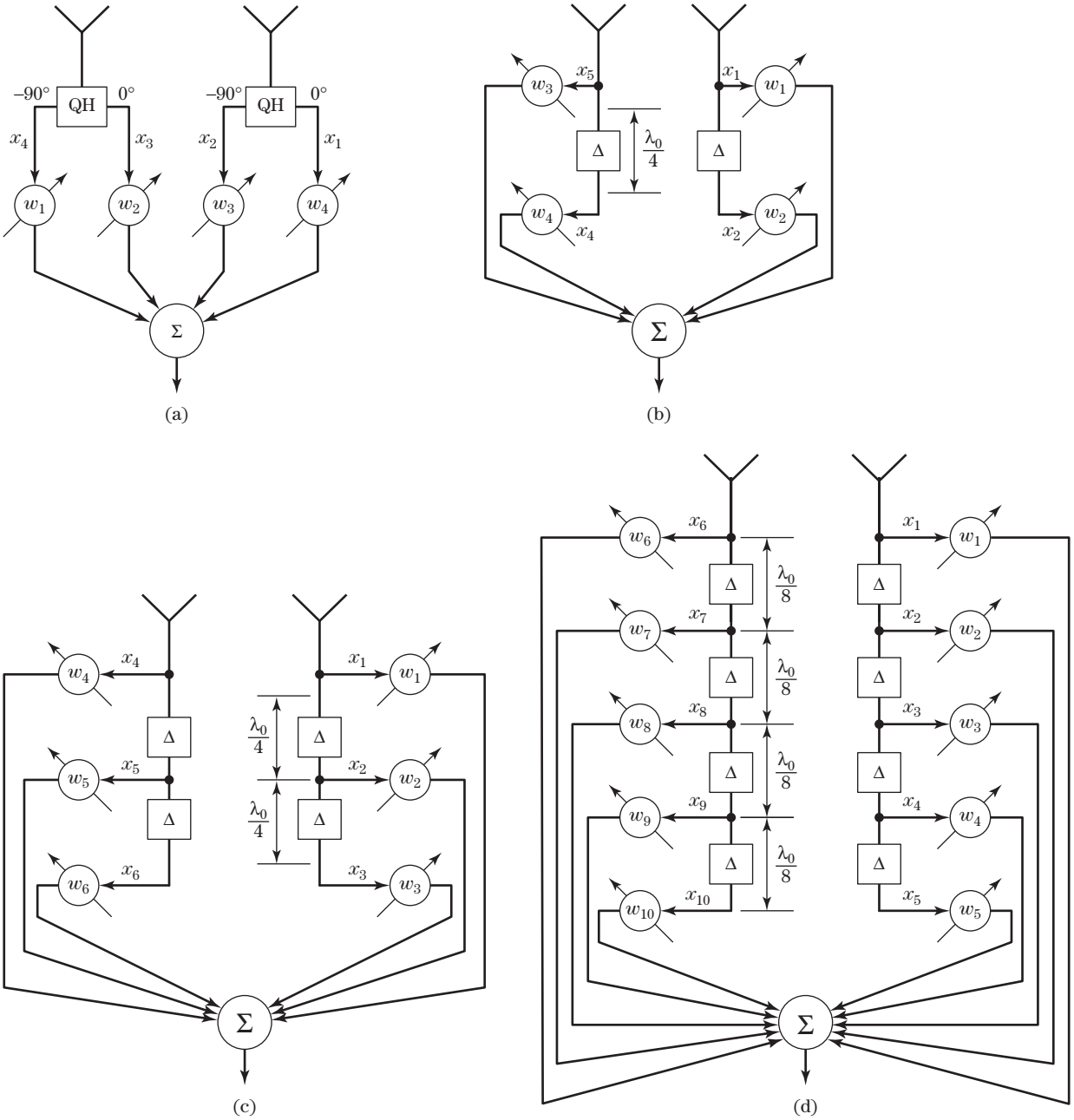


FIGURE 10-15 ■ Four adaptive array processors for broadband signal processing comparison. a: Quadrature hybrid. b: Two-tap delay line. c: Three-tap delay line. d: Five-tap delay line. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst*., January 1979 [13].

then, $s_d(t)$ is a stationary random process with power spectral density given by

$$\phi_{dd}(\omega) = \frac{A^2 T}{2}\left[\frac{\sin(T/2)(\omega-\omega_0)}{(T/2)\,(\omega-\omega_0)}\right]^2 \tag{10.53}$$

This power spectral density is shown in Figure 10-17.

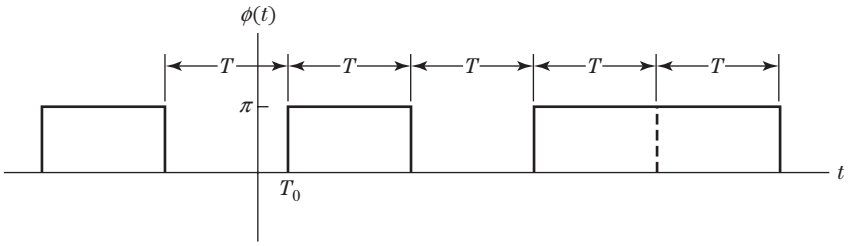


FIGURE 10-16 ■ Bit transitions for biphase modulated signal.

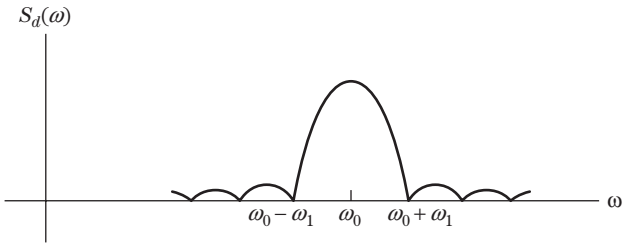


FIGURE 10-17 ■ Desired signal power spectral density.

The reference signal equals the desired signal component of $x_1(t)$ and is time aligned with the desired component of $x_2(t)$. The desired signal "bandwidth" will be taken to be the frequency range defined by the first nulls of the spectrum given by (10.53). With this definition, the fractional bandwidth then becomes

$$\text{desired signal bandwidth} = \frac{2\omega_1}{\omega_0} \tag{10.54}$$

where ω_1 is the frequency separation between the center frequency ω_0 and the first null

$$\omega_1 = \frac{2\pi}{T} \tag{10.55}$$

Assume that the interference signal is a Gaussian random process with a flat, bandlimited power spectral density over the range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$; then the interference signal spectrum appears in Figure 10-18. Finally, the thermal noise signals present at each element are statistically independent between elements, having a flat, bandlimited, Gaussian spectral density over the range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$ (identical with the interference spectrum of Figure 10-18).

With the foregoing definitions of signal spectra, the integrals of (10.48) and (10.51) yielding interference and thermal noise power are taken only over the frequency range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$. The desired signal power also is considered only over the frequency range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$ to obtain a consistent definition of signal-to-noise ratio (SNR). Therefore, the integral of (10.47) is carried out only over $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$.

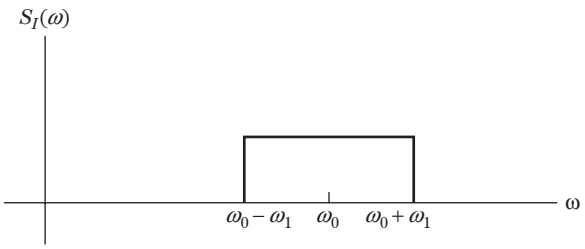


FIGURE 10-18 ■ Interference signal power spectral density.

To compare the four adaptive array processors of Figure 10-15, the output SNR performance is evaluated for the aforementioned signal conditions. Assume the element thermal noise power p_n is 10 dB below the element desired signal power p_s so that $p_s/p_n = 10$ dB. Furthermore, suppose that the element interference signal power p_i is 20 dB stronger than the element desired signal power so that $p_s/p_i = -20$ dB. Now assume that the desired signal is incident on the array from broadside. The output SNR given by (10.46) can be evaluated from (10.47), (10.48), and (10.49) by assuming the processor weights satisfy (10.16) for each of the four processor configurations. The resulting output signal-to-total noise ratio that results using each processor is plotted in Figures 10-19–10-22 as a function of the interference angle of arrival for 4, 10, 20, and 40% bandwidth signals, respectively.

In all cases, regardless of the signal bandwidth, when the interference approaches broadside (near the desired signal) the SNR degrades rapidly, and the performance of

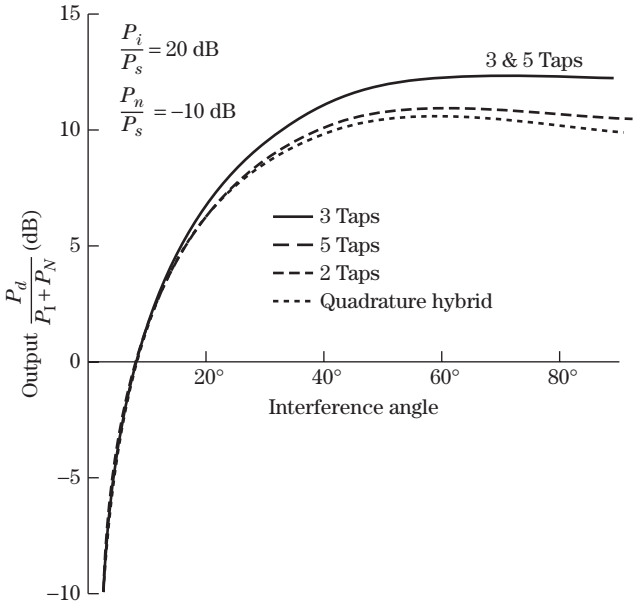


FIGURE 10-19 ■ Output signal-to-interference plus noise ratio interference angle for four adaptive processors with 4% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

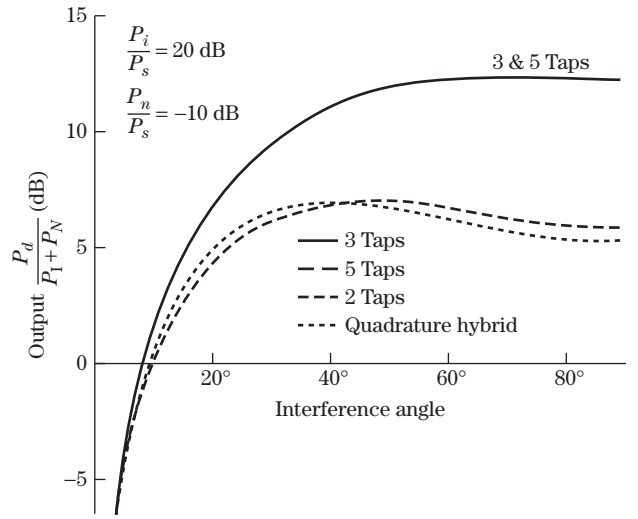


FIGURE 10-20 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 10% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

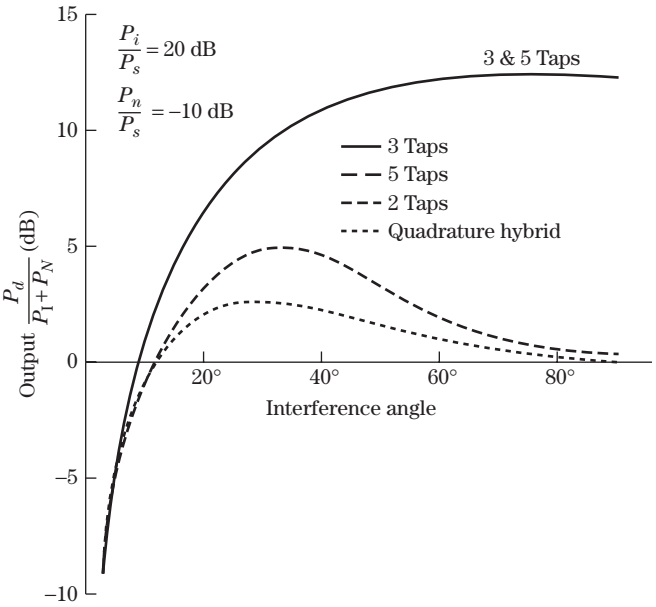


FIGURE 10-21 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 20% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

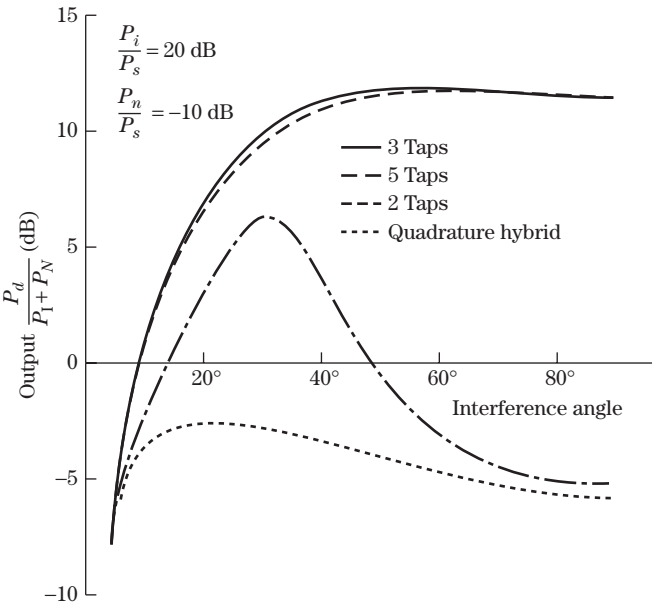


FIGURE 10-22 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 40% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

all four processors becomes identical. This SNR degradation is expected since, when the interference approaches the desired signal, the desired signal falls into the null provided to cancel the interference, and the output SNR consequently falls. Furthermore, as the interference approaches broadside, the interelement phase shift for this signal approaches zero. Consequently, the need to provide a frequency-dependent phase shift behind each array element to deal with the interference signal is less, and the performance of all four processors becomes identical.

When the interference signal is widely separated from the desired signal, then the output SNR is different for the four processors being considered, and this difference becomes more pronounced as the bandwidth increases. For 20 and 40% bandwidth signals,

for example, neither the quadrature hybrid processor nor the two-tap delay line processor provides good performance as the interference signal approaches endfire. The performance of both the three- and five-tap delay line processors remains quite good in the endfire region, however. If 20% or more bandwidth signals are accommodated, then tapped delay line processing becomes a necessity. Figure 10-22 shows that there is no significant performance advantage provided by the five-tap processor compared with the three-tap processor, so a three-tap processor is adequate for up to 40% bandwidth signals in the case of a two-element array.

Figures 10-21 and 10-22 show that the output SNR performance of the two-tap delay line processor peaks when the interference signal is 30° off broadside, because the interelement delay time is $\lambda/4$ (since the elements are spaced apart by $\lambda/2$). Consequently, the single-delay element value of $\lambda/4$ provides just the right amount of time delay to compensate exactly for the interelement time delay and to produce an improvement in the output SNR.

The three-tap and five-tap delay line processors both produce a maximum SNR of about 12.5 dB at wide interference angles of 70° or greater. For ideal channel processing, the interference signal is eliminated, the desired signal in each channel is added coherently to produce $P_d = 4p_s$, and the thermal noise is added noncoherently to yield $P_N = 2p_n$. Thus, the best possible theoretical output SNR for a two-element array with thermal noise 10 dB below the desired signal and no interference is 13 dB. Therefore, the three-tap and five-tap delay line processors are successfully rejecting nearly all the interference signal power at wide off-boresight angles.

10.3.4 Processor Transfer Functions

Ideally, the array transfer function for the desired signal should be constant across the desired signal bandwidth, thereby preventing desired signal distortion. The interference transfer function should be a low array response over the interference bandwidth.

The transfer functions for the four processors and the two-element array are evaluated using (10.40)–(10.45). Using the same conditions adopted in computing the SNR performance, Figures 10-23–10-26 show $|H_d(\omega)|$ and $|H_I(\omega)|$ for the four processors of

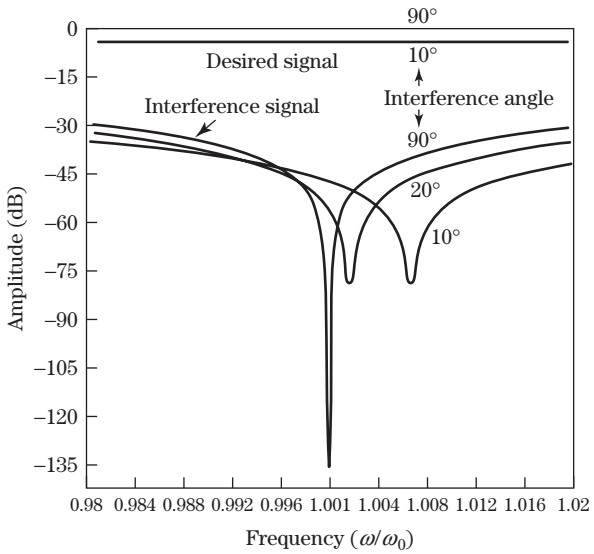


FIGURE 10-23 ■ Quadrature hybrid transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

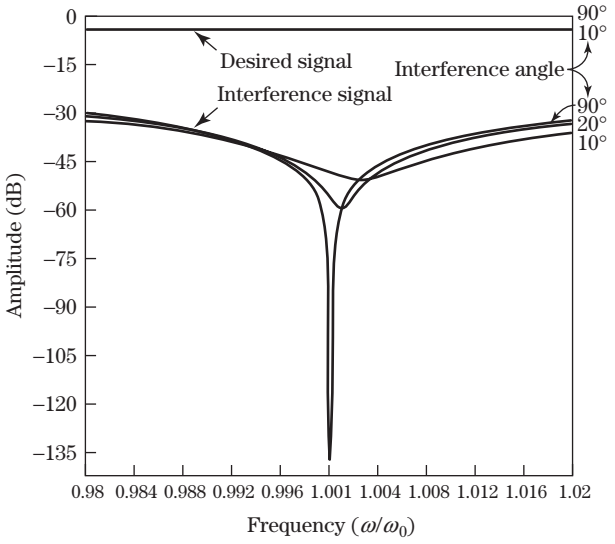


FIGURE 10-24 ■ Two-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

Figure 10-15 with a 4% signal bandwidth and various interference signal angles. The results shown in these figures indicate that for all four processors and for all interference angles the desired signal response is quite flat over the signal bandwidth. As the interference approaches the desired signal angle at broadside, however, the (constant) response level of the array to the desired signal drops because of the desired signal partially falling within the array pattern interference null.

The results in Figure 10-23 for quadrature hybrid processing show that the array response to the interference signal has a deep notch at the center frequency when the interference signal is well separated ($\theta_i > 20°$) from the desired signal. As the interference signal approaches the desired signal ($\theta_i < 20°$), the notch migrates away from the center frequency, because the processor weights must compromise between rejection of the interference signal and enhancement of the desired signal when the two signals are close. Migration of the notch improves the desired signal response (since the desired signal power spectral density peaks at the center frequency) while affecting interference rejection only slightly (since the interference signal power spectral density is constant over the signal band).

The array response for the two-tap processor is shown in Figure 10-24. The response to both the desired and interference signals is very similar to that obtained for quadrature hybrid processing. The most notable change is the slightly different shape of the transfer function notch presented to the interference signal by the two-tap delay line processor compared with the quadrature hybrid processor.

Figure 10-25 shows the three-tap processor array response. The interference signal response is considerably reduced, with a minimum rejection of the interference signal of about 45 dB. When the interference signal is close to the desired signal, the array response has a single mild dip. As the separation angle between the interference signal and the desired signal increases, the single dip becomes more pronounced and finally develops into a double dip at very wide angles. It is difficult to attribute much significance to the double-dip behavior since it occurs at such a low response level (of more than 75 dB attenuation). The five-tap processor response of Figure 10-26 is very similar to the three-tap processor response except slightly more interference signal rejection is achieved.

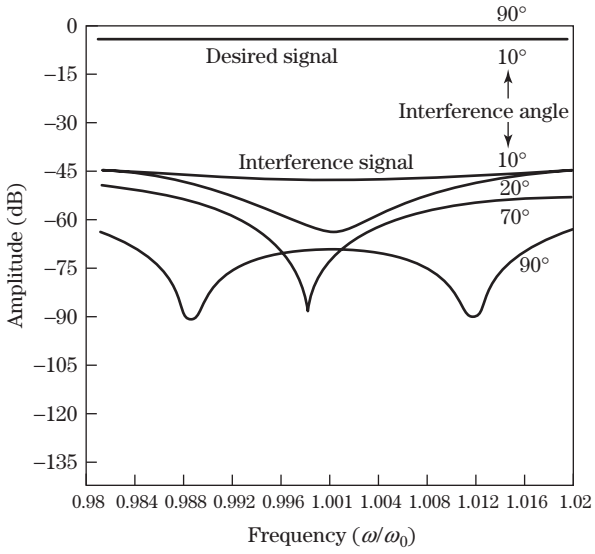


FIGURE 10-25 ■ Three-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

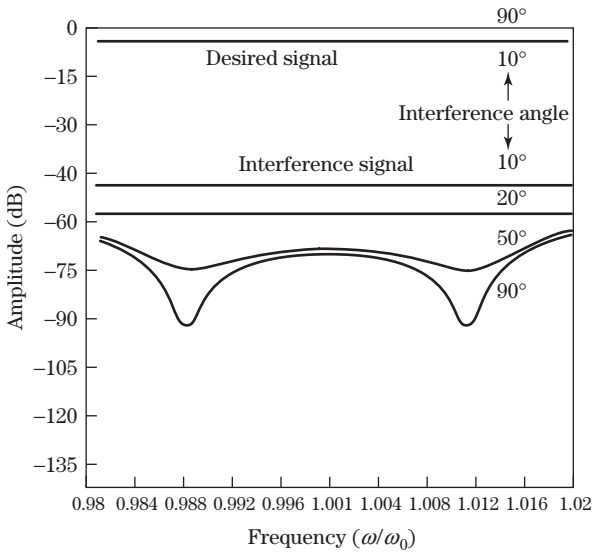


FIGURE 10-26 ■ Five-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

As the signal bandwidth increases, the processor response curves remain essentially the same as in Figures 10-23–10-26 except the following:

1. As the interference signal bandwidth increases, it becomes more difficult to reject the interference signal over the entire bandwidth, so the minimum rejection level increases.
2. The desired signal response decreases because the array feedback reduces all weights to compensate for the presence of a greater interference signal component at the array output, thereby resulting in greater desired signal attenuation.

The net result is that as the signal bandwidth increases, the output SNR performance degrades, as confirmed by the results of Figures 10-19–10-22.

10.4 COMPENSATION FOR MUTUAL COUPLING

In many applications, the limited space available for mounting an antenna motivates the use of a small array. As the array size decreases, the array element spacing becomes less than a half-wavelength, and mutual coupling effects become more of a factor in degrading the array performance. When an array consists of single-mode elements (meaning that the element aperture currents may change in amplitude but not in shape as a function of the signal angle of arrival), then it is possible to modify the element weights to compensate for the pattern distortion caused by the mutual coupling at a particular angle [16]. These weight adjustments may work for more than one angle.

Let the vector $\underline{v}$ denote the coupling perturbed measured voltages appearing at the output of the array elements, and let $\underline{v}^{d}$ represent the coupling unperturbed voltages that would appear at the array element outputs if no mutual coupling were present. The effect of mutual coupling on single-mode elements is written as

$$\mathbf{v}(u) = \mathbf{C}\,\mathbf{v}^{d}(u) \tag{10.56}$$

where $u = \sin\theta$, θ is the angle of arrival, and the matrix $\mathbf{C}$ describes the effects of mutual coupling and is independent of the signal scan angle. If the array is composed of multimode elements, then the matrix $\mathbf{C}$ would be scan angle dependent.

It follows that the unperturbed signal vector, $\mathbf{v}^{d}$ can be recovered from the perturbed signal vector by introducing compensation for the mutual coupling

$$\mathbf{v}^{d} = \mathbf{C}^{-1}\mathbf{v} \tag{10.57}$$

Introducing the compensation network $\mathbf{C}^{-1}$ as shown in Figure 10-27 then allows all subsequent beamforming operations to be performed with ideal (unperturbed) element signals, as are customarily assumed in pattern synthesis.

This mutual coupling compensation is applied to an eight-element linear array having element spacing $d = 0.517\ \lambda$ consisting of identical elements. Figure 10-28(a) shows the effects of mutual coupling by displaying the difference in element pattern shape between a central and an edge element in the array.

Figure 10-28 displays a synthesized 30 dB Chebyshev pattern both without (a) and with (b) mutual coupling compensation. It is apparent from this result that the compensation network gives about a 10 dB improvement in the sidelobe level.

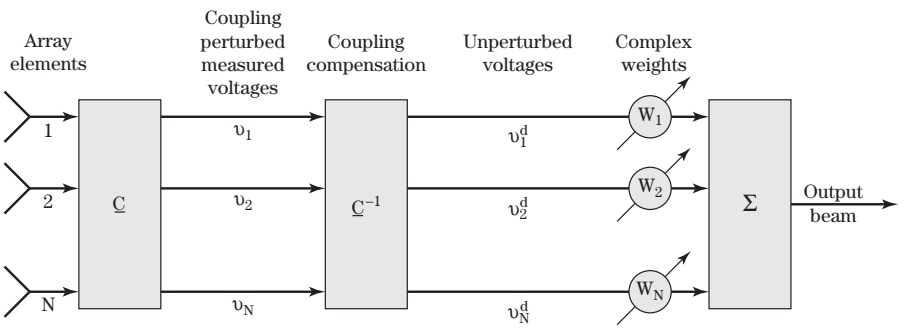


FIGURE 10-27 ■ Coupling Compensation and Beamforming in an Array Antenna. From Steyskal & Herd, IEEE Trans. Ant & Prop., Dec. 1995.

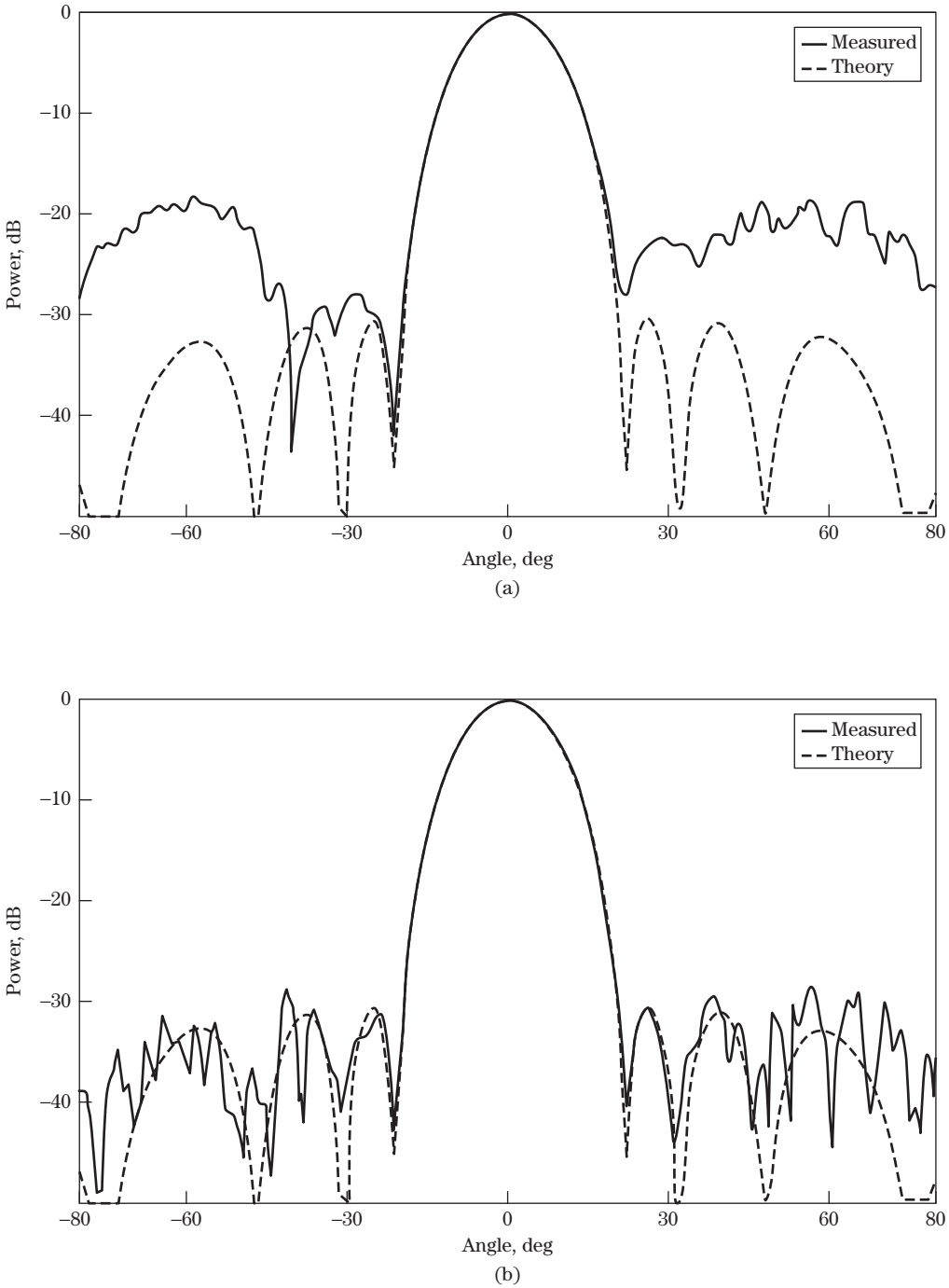


FIGURE 10-28 ■ 30 dB Chebyshev pattern (a) without and (b) with Coupling Compensation with a Scan Angle of 0°. From Steyskal & Herd, IEEE Trans. Ant. & Prop. Dec. 1995.

10.5 MULTIPATH COMPENSATION

In many operating environments, multipath rays impinge on the array shortly after the direct path signal arrives at the sensors. Multipath distorts any interference signal that may appear in the various element channels, thereby severely limiting the interference cancellation. A tapped delay line processor combines delayed and weighted replicas of the input signal to form the filtered output signal and thereby has the potential to compensate for multipath effects, since multipath rays also consist of delayed and weighted replicas of the direct path ray.

10.5.1 Two-Channel Interference Cancellation Model

Consider an ideal two-element adaptive array with one channel's (called the "auxiliary" channel) response adjusted so that any jamming signal entering the other channel through the sidelobes (termed the "main" channel) is canceled at the array output. A system designed to suppress sidelobe jamming in this manner is called a coherent sidelobe canceller (CSLC), and Figure 10-29 depicts a two-channel CSLC system in which the auxiliary channel employs tapped delay line compensation involving L weights and $L-1$ delay elements of value Δ seconds each. A delay element of value $D=(L-1)\Delta/2$ is included in the main channel so the center tap of the auxiliary channel corresponds to the output of the delay D in the main channel, thereby permitting compensation for both positive and negative values of the off-broadside angle θ. This ideal two-element CSLC system model exhibits all the salient characteristics that a more complex system involving several auxiliary channels would have, so the two-element system serves as a convenient model for performance evaluation of multipath cancellation [17].

The system performance measure is the ability of the CSLC to cancel an undesired interference signal through proper design of the tapped delay line. In actual practice, an adaptive algorithm adjusts the weight settings. To eliminate the effect of algorithm selection from consideration, only the steady-state performance is evaluated. Since the steady-state solution can be found analytically, it is necessary to determine only the resulting solution for the output residue power. This residue power is then a direct measure of the interference cancellation ability of the two-element CSLC model.

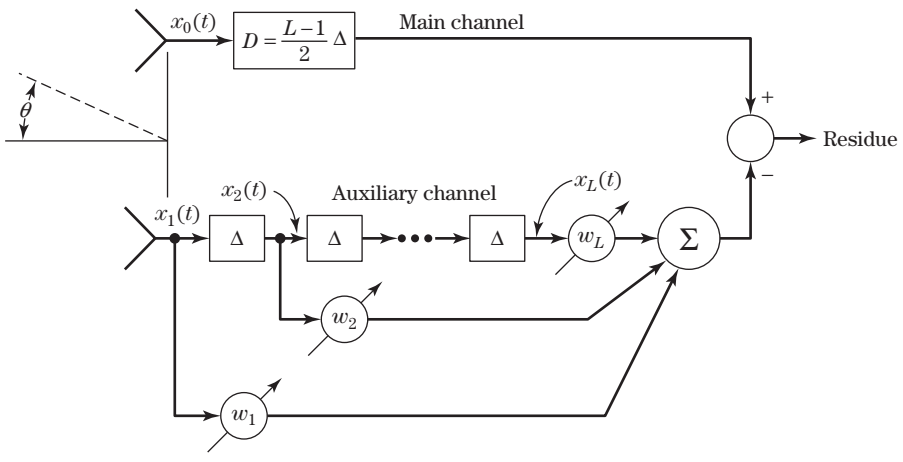


FIGURE 10-29 ■ Ideal two-element CSLC model with auxiliary channel compensation involving L weights and $L-1$ delay elements.

Let $x_0(t)$, $x_1(t)$, and $e(t)$ represent the complex envelope signals of the main channel input signal, the auxiliary channel input signal, and the output residue signal, respectively. Define the complex signal vector

$$\mathbf{x}^T \triangleq [x_1(t), x_2(t), \ldots, x_L(t)] \tag{10.58}$$

where

$$\begin{aligned} x_2(t) &\triangleq x_1(t - \Delta) \\ &\vdots \\ x_L(t) &\triangleq x_1\,[t - (L\ -\ 1)\Delta] \end{aligned}$$

Also, define the complex weight vector

$$\mathbf{w}^T \triangleq [w_1, w_2, \ldots, w_L] \tag{10.59}$$

The output of the tapped delay line may then be expressed as

$$\text{filter output} = \sum_{i=1}^{L} x_1[t - (i-1)\Delta]w_i^* = \mathbf{w}^\dagger \mathbf{x}(t) \tag{10.60}$$

The residue (complex envelope) signal is given by

$$e(t) = x_0(t - D) + \mathbf{w}^\dagger \mathbf{x}(t) \tag{10.61}$$

The weight vector $\mathbf{w}$ minimizes the residue signal in a mean square error (MSE) sense. For stationary random processes, this is equivalent to minimizing the expression

$$R_{ee}(0) = E\ \{e(t)e^*(t)\} \tag{10.62}$$

From (10.61) and the fact that

$$E\{x_0(t - D)x_0^*(t - D)\} = r_{x_0x_0}(0) \tag{10.63}$$

$$E\{\mathbf{x}(t)x_0^*(t - D)\} = r_{xx_0}(-D) \tag{10.64}$$

$$E\{\mathbf{x}(t)\mathbf{x}^\dagger(t)\} = \mathbf{R}_{xx}(0) \tag{10.65}$$

it follows that

$$\begin{aligned} R_{ee}(0) = r_{x_0x_0}(0) &- \mathbf{r}_{xx_0}^\dagger(-D)\mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \\ &+ [\mathbf{r}_{xx_0}^\dagger(-D) + \mathbf{w}^\dagger\mathbf{R}_{xx}(0)] \bullet \mathbf{R}_{xx}^{-1}(0)\,[\mathbf{r}_{xx_0}(-D) + \mathbf{R}_{xx}(0)\mathbf{w}] \end{aligned} \tag{10.66}$$

Minimize (10.66) by appropriately selecting the complex weight vector $\mathbf{w}$. Assume the matrix $\mathbf{R}_{xx}(0)$ is nonsingular: the value of $\mathbf{w}$ for which this minimum occurs is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \tag{10.67}$$

The corresponding minimum residue signal power then becomes

$$R_{ee}(0)_{\min} = r_{x_0x_0}(0) - \mathbf{r}_{xx_0}^\dagger(-D)\mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \tag{10.68}$$

Interference cancellation performance of the CSLC model of Figure 10-27 is determined by evaluating (10.66) using selected signal environment assumptions.

10.5.2 Signal Environment Assumptions

Let $s_1(t, \theta_1)$ represent the interference signal arriving from direction θ_1, and let $s_m(t, \rho_m, D_m, \theta_{m+1})$ for $m = 2, \ldots, M$ represent the multipath structure associated with the interference signal that consists of a collection of $M - 1$ correlated plane wave signals of the same frequency arriving from different directions so that $\theta_{m+k} \neq \theta_1$ and $\theta_{m+k} \neq \theta_{m+l}$ for $k \neq l$. The multipath rays each have an associated reflection coefficient ρ_m and a time delay with respect to the direct ray D_m. The structure of the covariance matrix for this multipath model can then be expressed as [18]

$$\mathbf{R}_{ss} = \mathbf{V}_s \mathbf{A} \mathbf{V}_s^{\dagger} \tag{10.69}$$

where $\mathbf{V}_s$ is the $N \times M$ signal matrix given by

$$\mathbf{V}_s = \left[\mathbf{v}_{s_1} \,\middle|\, \mathbf{v}_{s_2} \,\middle|\, \cdots \,\middle|\, \mathbf{v}_{s_M} \right] \tag{10.70}$$

whose components are given by the $N \times 1$ vectors

$$\mathbf{v}_{s_m} = \sqrt{P_{s_m}} \begin{bmatrix} 1 \\ \exp\left[j2\pi (d/\lambda_0) \sin\theta_m \right] \\ \exp\left[j2\pi (d/\lambda_0) 2 \sin\theta_m \right] \\ \vdots \\ \exp\left[j2\pi (d/\lambda_0) (N-1) \sin\theta_m \right] \end{bmatrix} \tag{10.71}$$

where $P_{s_m} = \rho_m^2$ denotes the power associated with the signal s_m, and $\mathbf{A}$ is the multipath correlation matrix. When $\mathbf{A} = \mathbf{I}$, the various signal components are uncorrelated whereas for $\mathbf{A} = \mathbf{U}$ (the $M \times M$ matrix of unity elements) the various components are perfectly correlated. For purposes of numerical evaluation the correlation matrix model may be selected as [18]

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{M-1} \\ \alpha & 1 & \alpha & \cdots & \alpha^{M-1} \\ \vdots & & & & \\ \alpha^{M-1} & . & . & \cdots & 1 \end{bmatrix} \quad 0 \leq \alpha \leq 1 \tag{10.72}$$

Note that channel-to-channel variations in θ_m, D_m, and ρ_m cannot be accommodated by this simplified model. Consequently, a more general model must be developed to handle such variations, which tend to occur where near-field scattering effects are significant. The input signal covariance matrix may be written as

$$\mathbf{R}_{xx} = \mathbf{R}_{nn} + \mathbf{V}_s \mathbf{A} \mathbf{V}_s^{\dagger} \tag{10.73}$$

where $\mathbf{R}_{nn}$ denotes the noise covariance matrix.

If only a single multipath ray is present, then $s(t, \theta_1)$ denotes the direct interference signal, and $s_m(t, \rho_m, D_m, \theta_2)$ represents the multipath ray associated with the direct interference signal. The received signal at the main channel element is then given by

$$x_0(t) = s(t, \theta_1) + s_m(t, \rho_m, D_m, \theta_2) \tag{10.74}$$

Denote $s(t, \theta_1)$ by $s(t)$; then $s_m(t, \rho_m, D_m, \theta_2)$ can be written as $\rho_m s(t - D_m) \times \exp(-j\omega_0 D_m)$ so that

$$x_0(t) = s(t) + \rho_m s(t - D_m) \exp(-j\omega_0 D_m) \tag{10.75}$$

where ω_0 is the center frequency of the interference signal. It then follows that

$$\begin{aligned} x_1(t) &= s(t-\tau_{12})\,\exp(-j\omega_0\tau_{12}) \\ &\quad + \rho_m s(t-D_m-\tau_{22})\,\exp\left[-j\omega_0(D_m+\tau_{22})\right] \end{aligned} \tag{10.76}$$

where τ_{12} and τ_{22} represent the propagation delay between the main channel element and the auxiliary channel element for the wavefronts of $s(t, \theta_1)$ and $s_m(t, \rho_m, d_m, \theta_2)$, respectively.

Assuming the signals $s(t, \theta_1)$ and $s_m(t, \rho_m, D_m, \theta_2)$ possess flat spectral density functions over the bandwidth B, as shown in Figure 10-30*a*, then the corresponding auto- and cross-correlation functions of $x_0(t)$ and $x_1(t)$ can be evaluated by recognizing that

$$\mathbf{R}_{xx}(\tau) = \mathfrak{F}^{-1}\{\mathbf{\Phi}_{xx}(\omega)\} \tag{10.77}$$

where $\mathfrak{F}^{-1}\{\cdot\}$ is the "inverse Fourier transform," and $\mathbf{\Phi}_{xx}(\omega)$ denotes the cross-spectral density matrix of $\mathbf{x}(t)$.

From (10.74), (10.76), and (10.77) it immediately follows that

$$r_{x_0x_0}(0) = 1 + |\rho_m|^2 + \frac{\sin\pi BD_m}{\pi BD_m}\left(\rho_m e^{-j\omega_0 D_m} + \rho_m^* e^{j\omega_0 D_m}\right) \tag{10.78}$$

Likewise, defining $f[\psi, \text{sgn1}, \text{sgn2}] \triangleq \dfrac{\sin\pi B\,[\psi + \text{sgn1}\cdot(i-1)\Delta + \text{sgn2}\cdot D]}{\pi B\,[\psi + \text{sgn1}\cdot(i-1)\Delta + \text{sgn2}\cdot D]}$

and $\quad g[\psi, \text{sgn}] \triangleq \dfrac{\sin\pi B\,[\psi + \text{sgn}\cdot(i-k)\Delta]}{\pi B[\psi + \text{sgn}\cdot(i-k)\,\Delta]}$, $\quad$ then

$$\begin{aligned} r_{x_ix_0}(-D) &= f[\tau_{12}, +, -]\exp\{-j\omega_0[\tau_{12} + (i-1)\Delta]\} \\ &\quad + f[D_m+\tau_{22}, +, -]\rho_m \exp\{-j\omega_0[\tau_{22} + (i-1)\Delta + D_m]\} \\ &\quad + f[D_m-\tau_{12}, -, +]\rho_m^* \exp\{-j\omega_0[\tau_{12} + (i-1)\Delta - D_m]\} \\ &\quad + f[\tau_{22}, +, -]|\rho_m|^2 \exp\{-j\omega_0[\tau_{22} + (i-1)\Delta]\} \end{aligned} \tag{10.79}$$

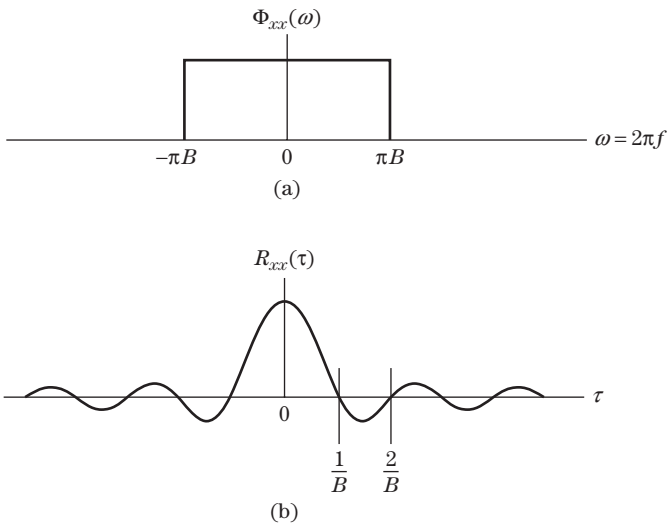


FIGURE 10-30 ■ Flat spectral density function and corresponding autocorrelation function for interference signal. a: Spectral density function. b: Autocorrelation function.

$$
\begin{aligned}
r_{x_i x_k}(0) = {} & g[0,+][1+|\rho_m|^2]\exp[-j\omega_0(i-k)\Delta] \\
& + g[\tau_{12}-\tau_{22}-D_m,-]\rho_m \exp\{j\omega_0[\tau_{12}-\tau_{22}-D_m-(i-k)\Delta]\} \\
& + g[\tau_{12}-\tau_{22}-D_m,+]\rho_m^* \exp\{-j\omega_0[\tau_{12}-\tau_{22}-D_m+(i-k)\Delta]\}
\end{aligned} \tag{10.80}
$$

The vector $\mathbf{r}_{xx_0}(-D)$ is then given by

$$
\mathbf{r}_{xx_0}(-D) = \begin{bmatrix} r_{x_1x_0}(-D) \\ r_{x_2x_0}(-D) \\ \vdots \\ r_{x_Nx_0}(-D) \end{bmatrix} \tag{10.81}
$$

and the matrix $\mathbf{R}_{xx}(0)$ is given by

$$
\mathbf{R}_{xx}(0) = \begin{bmatrix} r_{x_1x_1}(0) & r_{x_1x_2}(0) & \cdots & r_{x_1x_N}(0) \\ \vdots & r_{x_2x_2}(0) & & \\ & & \ddots & \\ r_{x_1x_N}(0) & \cdots & & r_{x_Nx_N}(0) \end{bmatrix} \tag{10.82}
$$

To evaluate (10.68) for the minimum possible value of output residue power (10.78), (10.79), and (10.80), show that it is necessary to specify the following parameters:

N = number of taps in the transversal filter
ρ_m = multipath reflection coefficient
ω_0 = (radian) center frequency of interference signal
D_m = multipath delay time with respect to direct ray
τ_{12} = propagation delay between the main antenna element and the auxiliary antenna element for the direct ray
τ_{22} = propagation delay between the main antenna element and the auxiliary antenna element for the multipath ray
Δ = transversal filter intertap delay
B = interference signal bandwidth
D = main channel receiver time delay

The quantities τ_{12} and τ_{22} are related to the CSLC array geometry by

$$
\left.\begin{aligned} \tau_{12} &= \frac{d}{\mathfrak{v}}\sin\theta_1 \\ \tau_{22} &= \frac{d}{\mathfrak{v}}\sin\theta_2 \end{aligned}\right\} \tag{10.83}
$$

where

d = interelement array spacing
$\mathfrak{v}$ = wavefront propagation speed
θ_1 = angle of incidence of direct ray
θ_2 = angle of incidence of multipath ray

10.5.3 Example: Results for Compensation of Multipath Effects

An interference signal has a direct ray angle of arrival is $\theta_1 = 30^\circ$, the multipath ray angle of arrival is $\theta_2 = -30^\circ$, and the interelement spacing is $d = 2.25\lambda_0$. Some additional

signal and multipath characteristics are

$$\begin{aligned} \text{center frequency} \quad & f_0 = 237 \text{ MHz} \\ \text{signal bandwidth} \quad & B = 3 \text{ MHz} \\ \text{multipath reflection coefficient} \quad & \rho_m = 0.5 \end{aligned} \tag{10.84}$$

Referring to (10.76), (10.79), and (10.80), we see that the parameters ω_0, τ_{12}, τ_{22}, D_m, and Δ enter the evaluation of the output residue power in the form of the products $\omega_0\tau_{12}$, $\omega_0\tau_{22}$, $\omega_0 D_m$, and $\omega_0\Delta$. These products represent the phase shift experienced at the center frequency ω_0 as a consequence of the four corresponding time delays. Likewise, the parameters B, D, D_m, τ_{12}, τ_{22}, and Δ enter the evaluation of the output residue power in the form of the products BD, BD_m, $B\tau_{12}$, $B\tau_{22}$, and $B\Delta$; these time–bandwidth products are phase shifts experienced by the highest frequency component of the complex envelope interference signal as a consequence of the five corresponding time delays. Both the intertap delay Δ and the multipath delay D_m are important parameters that affect the CSLC system performance through their corresponding time–bandwidth products; thus, the results are given here with the time–bandwidth products taken as the fundamental quantity of interest.

Since for this example $\theta_1 = -\theta_2$, the product $\omega_0\tau_{12}$ is specified as

$$\left.\begin{aligned} & \omega_0\tau_{12} = \frac{\pi}{4} \\ \text{then the product} \quad & \\ & \omega_0\tau_{22} = -\frac{\pi}{4} \end{aligned}\right\} \tag{10.85}$$

Furthermore, let the products $\omega_0 D_m$ and $\omega_0\Delta$ be given by

$$\left.\begin{aligned} \omega_0 D_m &= 0 \pm 2k\pi, \qquad k \text{ any integer} \\ \omega_0\Delta &= 0 \pm 2l\pi, \qquad l \text{ any integer} \end{aligned}\right\} \tag{10.86}$$

For the element spacing $d = 2.25\lambda_0$ and $\theta_1 = 30^\circ$, then specify

$$B\tau_{12} = -B\tau_{22} = \frac{1}{P}, \qquad P = 72 \tag{10.87}$$

Finally, specifying the multipath delay time to correspond to 46 meters yields

$$BD_m = 0.45 \tag{10.88}$$

Since

$$D = \frac{N-1}{2}\Delta \tag{10.89}$$

Only N and $B\Delta$ need to be specified to evaluate the output residue power by way of (10.68).

To evaluate the output residue power by way of (10.68) resulting from the array geometry and multipath conditions specified by (10.84)–(10.89) requires that the cross-correlation vector $\mathbf{r}_{xx_0}(-D)$, the $N \times N$ autocorrelation matrix $\mathbf{R}_{xx}(0)$, and the autocorrelation function $r_{x_0x_0}(0)$ be evaluated by way of (10.78)–(10.80). A computer program to evaluate (10.68) for the multipath conditions specified was written in complex, double-precision arithmetic.

Figure 10-31 shows a plot of the output residue power where the resulting minimum possible value of canceled power output in dB is plotted as a function of $B\Delta$ for various

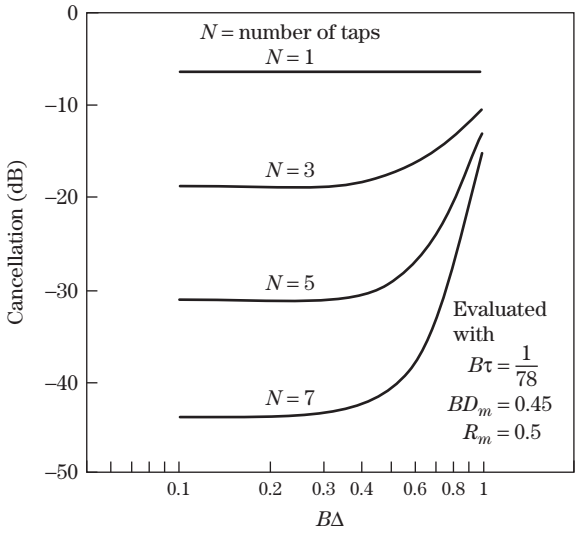


FIGURE 10-31 ■
Decibel cancellation versus $B\Delta$ for multipath.

specified values of N. It will be noted in Figure 10-31 that for $N = 1$ the cancellation performance is independent of $B\Delta$ since no intertap delays are present with only a single tap. As explained in Appendix B, the transfer function of the tapped delay line transversal filter has a periodic structure with (radian) frequency period $2\pi B_f$, which is centered at the frequency f_0. It should be noted that the transversal filter frequency bandwidth B_f is not necessarily the same as the signal-frequency bandwidth B. The transfer function of a transversal filter within the primary frequency band ($|f - f_0| < B_f/2$) may be expressed as

$$F(f) = \sum_{k=1}^{N} [A_k e^{j\phi_k}] \exp[-j2\pi(k-1)\delta f \Delta] \tag{10.90}$$

where $A_k e^{j\phi_k}$ represents the kth complex weight, $\delta f = f - f_0$, $f_0 =$ center frequency, and the transversal filter frequency bandwidth is

$$B_f = \frac{1}{\Delta} \tag{10.91}$$

Since the transversal filter should be capable of adjusting the complex weights to achieve appropriate amplitude and phase values over the entire signal bandwidth B, it follows that B_f should satisfy

$$B_f \geq B \tag{10.92}$$

Consequently, the maximum intertap delay spacing is given by

$$\Delta_{\max} = \frac{1}{B} \tag{10.93}$$

It follows that values of $B\Delta$ that are greater than unity should not be considered for practical compensation designs; however, values of $B_f > B$ (resulting in $0 < B\Delta < 1$) are sometimes desirable.

Figure 10-31 shows that, as $B\Delta$ decreases from 1, for values of $N > 1$ the cancellation performance rapidly improves (the minimum canceled residue power decreases) until $B\Delta = BD_m$ (0.45 for this example), after which very little significant improvement occurs. As $B\Delta$ becomes very much smaller than BD_m (approaching zero), the cancellation performance degrades since the intertap delay is effectively removed. The simulation could not compute this result since as $B\Delta$ approaches zero the matrix $\mathbf{R}_{xx}(0)$ becomes singular and matrix inversion becomes impossible. Cancellation performance of -30 dB is virtually assured if the transversal filter has at least five taps and Δ is selected so that $\Delta = D_m$.

Suppose for example that the transversal filter is designed with $B\Delta = 0.45$. Using the same set of selected constants as for the previous example, we find it useful to consider what results would be obtained when the actual multipath delay is different from the anticipated value corresponding to $BD_m = 0.45$. From the results already obtained in Figure 10-31, it may be anticipated that, if $BD_m > B\Delta$, then the cancellation performance would degrade. If, however, $BD_m \ll B\Delta$, then the cancellation performance would improve since in the limit as $D_m \to 0$ the system performance with no multipath present would result.

10.5.4 Results for Compensation of Array Propagation Delay

In the absence of a multipath ray, the analysis presented in the preceding section includes all the features necessary to account for array propagation delay effects. When we set $\rho_m = 0$ and let $\tau_{12} = \tau$ represent the element-to-element array propagation delay, (10.78)–(10.80) permit (10.68) to be used to investigate the effects of array propagation delay on cancellation performance. On the basis of the behavior already found for multipath compensation, it would be reasonable to anticipate that with $B\Delta = B\tau$ then maximum cancellation performance would obtain, whereas if $B\Delta > B\tau$ then the cancellation performance would degrade. Figure 10-32 gives the resulting cancellation performance as a function of $B\Delta$ for fixed $B\tau$. The number of taps N is an independent parameter, and all other system constants are the same as those in the example of Section 10.4.3. It is seen that the results confirm the anticipated performance noted already.

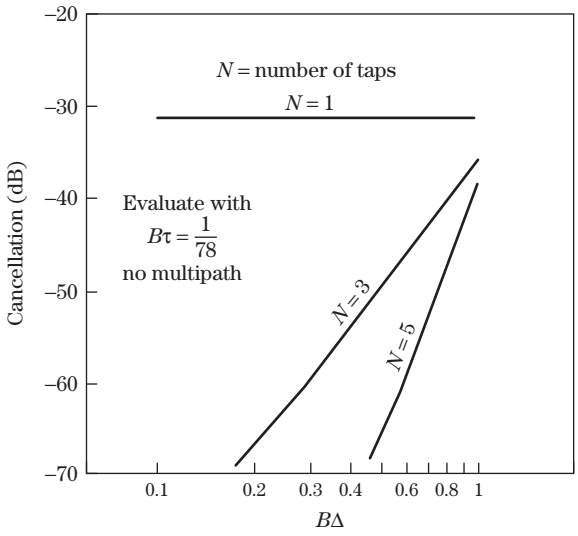


FIGURE 10-32 ■ Decibel cancellation versus $B\Delta$ for array propagation delay.

10.6 | ANALYSIS OF INTERCHANNEL MISMATCH EFFECTS

Any adaptive array processor is susceptible to unavoidable frequency-dependent variations in gain and phase between the various element channels. Additional degrees of freedom provided by a tapped delay line compensate for such frequency-dependent "channel mismatch" effects. Since a simple two-element CSLC system exhibits all the salient characteristics of channel mismatching present in more complex systems, the two-element model is again adopted as the example for performance evaluation of channel mismatch compensation.

Figure 10-33 is a simplified representation of a single auxiliary channel CSLC system in which the single complex weight is a function of frequency. The transfer function $T_0(\omega, \theta)$ reflects all amplitude and phase variations in the main beam sidelobes as a function of frequency as well as any tracking errors in amplitude and phase between the main and auxiliary channel electronics. Likewise, the equivalent transfer function for the auxiliary channel (including any auxiliary antenna variations) is denoted by $T_1(\omega, \theta)$. The spectral power density of a wideband jammer is given by $\phi_{JJ}(\omega)$. The signal from the auxiliary channel is "multiplied" by the complex weight $w_1 = \alpha e^{j\phi}$, and the "cancelled" output of residue power spectral density is represented by $\phi_{rr}(\omega, \theta)$.

The objective of the CSLC is to minimize the residue power, appropriately weighted, over the bandwidth. Since the integral of the power spectral density over the signal frequency spectrum yields the signal power, the requirement to minimize the residue power is expressed as

$$\underset{w_1}{\text{Min}} \int_{-\infty}^{\infty} \phi_{rr}(\omega, \theta) d\omega \tag{10.94}$$

where

$$\phi_{rr}(\omega, \theta) = |T_0(\omega, \theta) - w_1 T_1(\omega, \theta)|^2 \phi_{JJ}(\omega) \tag{10.95}$$

Now replace the complex weight w_1 in Figure 10-33 by a tapped delay line having $2N + 1$ adaptively controlled complex weights each separated by a time delay Δ as in Figure 10-34. A delay element of value $N\Delta$ is included in the main channel (just as in the preceding section) so that compensation for both positive and negative angles of arrival is provided. The main and auxiliary channel transfer functions are written in terms of the output of the main channel, so no delay terms occur in the resulting main channel transfer

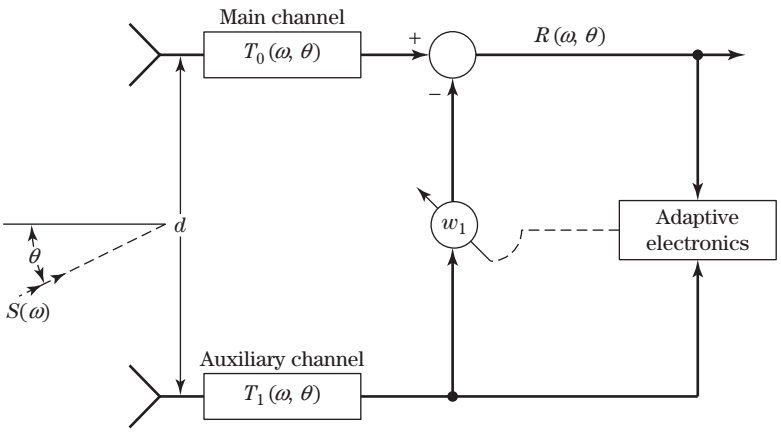


FIGURE 10-33 ■ Simplified model of single-channel CSLC.

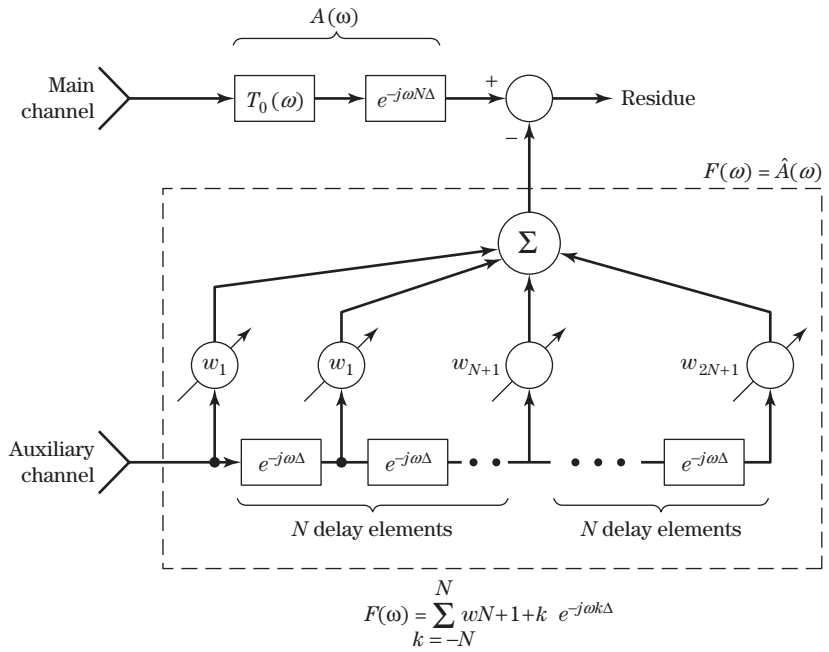


FIGURE 10-34 ■ Single-channel CSLC having main channel distortion and tapped delay line auxiliary channel compensation.

function, $A(\omega)$. Assume for analysis purposes that all channel distortion is confined to the main channel and that $T_1(\omega, \theta) = 1$. The transversal filter transfer function, $F(\omega)$, can be expressed as

$$F(\omega) = \sum_{k=-N}^{N} w_{N+1+k} e^{-j\omega k\Delta} \tag{10.96}$$

where the w_{N+1+k}'s are nonfrequency-dependent complex weights.

We want to minimize the output residue power over the signal bandwidth by appropriately selecting the weight vector **w**. Assuming the jammer power spectral density is constant over the frequency region of interest, then minimizing the output residue power is equivalent to selecting the $F(\omega)$ that provides the "best" estimate (denoted by $\hat{A}(\omega)$) of the main channel transfer function over that frequency range. If the estimate $\hat{A}(\omega)$ is to be optimal in the MSE sense, then the error in this estimate $e(\omega) = A(\omega) - F(\omega)$ must be orthogonal to $\hat{A}\,(\omega) = F(\omega)$, that is,

$$E\{[A(\omega) - F(\omega)]F^*(\omega)\} = 0 \tag{10.97}$$

where the expectation $E\{\cdot\}$ is taken over frequency and is therefore equivalent to

$$E\{\cdot\} = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\{\ \}\,d\omega \tag{10.98}$$

where all frequency-dependent elements in the integrand of (10.98) are reduced to baseband. Letting $A(\omega) = A_0(\omega)e^{-j\phi_0(\omega)}$, substituting (10.96) into (10.97), and requiring the error to be orthogonal to all tap outputs to obtain the minimum MSE estimate $\hat{A}(\omega)$ then yields the condition

$$E\{[A_0(\omega)\exp[-j\phi_0(\omega)] - F(\omega)]\exp(j\omega k\,\Delta)\} = 0 \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.99}$$

Equation (10.99) can be rewritten as

$$E\{A_0(\omega)\exp[j(\omega k\,\Delta-\phi_0(\omega)]\}-E\left\{\left[\sum_{l=-N}^{N}W_{N+1+l}\exp(-j\omega l\Delta)\right]\cdot\exp(j\omega k\Delta)\right\}=0\qquad\text{for } k=-N,\ldots,0,\ldots,N \tag{10.100}$$

Note that

$$E\{\exp[-j\omega(l-k)\Delta]\}=\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.101}$$

it follows that

$$E\left\{\left[\sum_{l=-N}^{N}W_{N+1+l}\exp(-j\omega l)\right]\exp(j\omega l\Delta)\right\}=\sum_{l=-N}^{N}W_{N+1+l}\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.102}$$

so that (10.100) can be rewritten in matrix form as

$$\mathbf{v}=\mathbf{C}\mathbf{w} \tag{10.103}$$

where

$$v_k=E\{A_0(\omega)\exp[j(\omega k\Delta-\phi_0(\omega))]\} \tag{10.104}$$

$$C_{k,l}=\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.105}$$

Consequently, the complex weight vector must satisfy the relation

$$\mathbf{w}=\mathbf{C}^{-1}\mathbf{v} \tag{10.106}$$

Using (10.106) to solve for the optimum complex weight vector, we can find the output residue signal power by using

$$R_{ee}(0)=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}|A(\omega)-F(\omega)|^2\phi_{JJ}(\omega)d\omega \tag{10.107}$$

where $\phi_{JJ}(\omega)$ is the constant interference signal power spectral density. Assume the interference power spectral density is unity across the bandwidth of concern; then the output residue power due only to main channel amplitude variations is given by

$$R_{ee_A}=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}|A_0(\omega)-F(\omega)|^2d\omega \tag{10.108}$$

Since $A(\omega)-F(\omega)$ is orthogonal to $F(\omega)$, it follows that [15]

$$E\{|A(\omega)-F(\omega)|^2\}=E\{|A(\omega)|^2\}-E\{|F(\omega)|^2\} \tag{10.109}$$

and hence

$$R_{ee_A}=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[A_0^2(\omega)-|F(\omega)|^2]\,d\omega \tag{10.110}$$

It likewise follows from (10.107) that the output residue power contributed by main channel phase variations is given by

$$R_{ee_p} = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} |e^{-j\phi_0(\omega)} - F(\omega)|^2 \phi_{JJ}(\omega)\, d\omega \tag{10.111}$$

where $\phi_0(\omega)$ represents the main channel phase variation. Once again assuming that the input signal spectral density is unity across the signal bandwidth and noting that $[e^{-j\phi_0(\omega)} - F(\omega)]$ must be orthogonal to $F(\omega)$, it immediately follows that

$$\begin{aligned} R_{ee_p} &= \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} [1 - |F(\omega)|^2]\, d\omega \\ &= 1 - \sum_{j=-N}^{N}\sum_{k=-N}^{N} w_k w_j^* \frac{\sin[\pi B\Delta(k-j)]}{\pi B\Delta(k-j)} \end{aligned} \tag{10.112}$$

where the complex weight vector elements must satisfy (10.103)–(10.106).

If it is desired to evaluate the effects of both amplitude and phase mismatching simultaneously, then the appropriate expression for the output residue power is given by (10.107), which (because of orthogonality) may be rewritten as

$$R_{ee}(0) = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} \{|A(\omega)|^2 - |F(\omega)|^2\}\phi_{JJ}(\omega)\, d\omega \tag{10.113}$$

where the complex weights used to obtain $F(\omega)$ must again satisfy (10.102)–(10.106), which now involve both a magnitude and a phase component and it is assumed that $\phi_{JJ}(\omega)$ is a constant.

10.6.1 Example: Effects of Amplitude Mismatching

To evaluate (10.110) it is necessary to adopt a channel amplitude model corresponding to $A(\omega)$. One possible channel amplitude model is given in Figure 10-35 for which

$$A(\omega) = \begin{cases} 1 + R\cos\omega T_0 & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases} \tag{10.114}$$

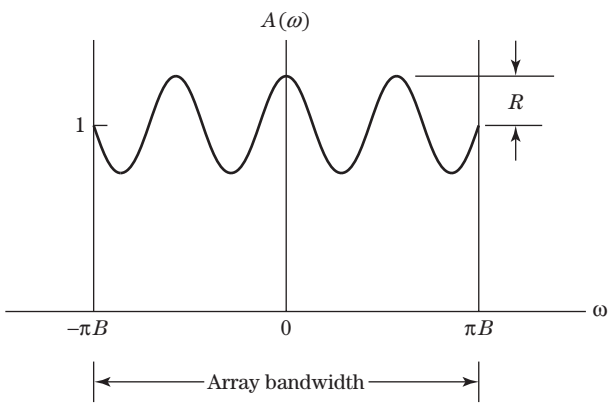


FIGURE 10-35 ■ Channel amplitude model having $3\frac{1}{2}$ cycles of ripple for evaluation of amplitude mismatch effects.

where

$$T_0 = \frac{2n+1}{2B} \qquad \text{for } n = 0, 1, 2, \ldots$$

and the integer n corresponds to $(2n+1)/2$ cycles of amplitude mismatching across the bandwidth B. Letting the phase error $\phi_0(\omega) = 0$, it follows from (10.104) that

$$v_k = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]e^{j\omega k\Delta}\,d\omega \tag{10.115}$$

or

$$v_k = \frac{\sin(\pi Bk\Delta)}{\pi Bk\,\Delta} + \frac{R}{2}\left[\frac{\sin(\pi B[T_0 + k\Delta])}{\pi B\,[T_0 + k\Delta]} + \frac{\sin(\pi B[T_0 - k\Delta])}{B[T_0 - k\Delta]}\right] \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.116}$$

Evaluation of (10.116) permits the complex weight vector to be found, which in turn may be used to determine the residue power by way of (10.110).

Now

$$|F(\omega)|^2 = F(\omega)\,F^*(\omega) = \mathbf{w}^\dagger\boldsymbol{\beta}\boldsymbol{\beta}^\dagger\mathbf{w} \tag{10.117}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} e^{j\omega N\Delta} \\ e^{j\omega(N-1)\,\Delta} \\ \vdots \\ e^{-j\omega N\Delta} \end{bmatrix} \tag{10.118}$$

Carrying out the vector multiplications indicated by (10.117) then yields

$$|F(\omega)|^2 = \sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta} \tag{10.119}$$

The output residue power is therefore given by [see equation (10.110)]

$$R_{ee_A} = \int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]^2\,d\omega - \int_{-\pi B}^{\pi B}\sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta}\,d\omega \tag{10.120}$$

Equation (10.120) may be evaluated using the following expressions:

$$\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]^2\,d\omega = \left(1 + \frac{R^2}{2}\right) + 2R\frac{\sin\pi[(2n+1)/2]}{\pi[(2n+1)/2]} + \frac{R^2}{2}\frac{\sin\pi(2n+1)}{\pi(2n+1)} \tag{10.121}$$

$$\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta}\,d\omega = \sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^*\frac{\sin\pi(k-i)B\Delta}{\pi(k-i)B\Delta} \tag{10.122}$$

10.6.2 Results for Compensation of Selected Amplitude Mismatch Model

The evaluation of (10.120) requires knowing the ripple amplitude R, the number of cycles of amplitude mismatching across the bandwidth, and the product of $B\Delta$ (where B is the cancellation bandwidth and Δ is the intertap delay spacing). The results of a computer evaluation of the output residue power are summarized in Figures 10-36–10-39 for $B\Delta = 0.25$, 0.5, 0.75, and 1, and $R = 0.09$. Each of the figures presents a plot of the decibel cancellation (of the undesired interference signal) achieved as a function of the number of taps in the transversal filter and the number of cycles of ripple present across the cancellation bandwidth. No improvement (over the cancellation that can be achieved with only one tap) is realized until a sufficient number of taps is present in the transversal filter to achieve the resolution required by the amplitude versus frequency variations in

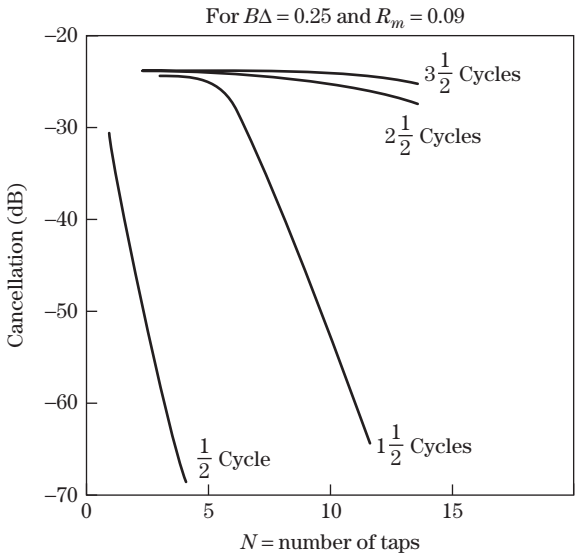


FIGURE 10-36 ■ Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.25$.

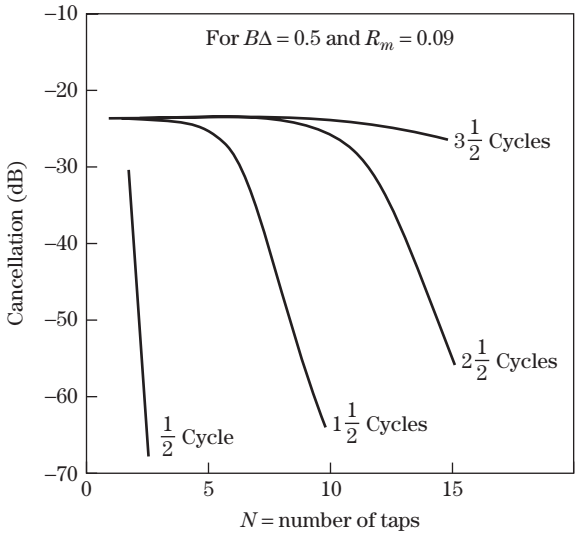


FIGURE 10-37 ■ Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.5$.

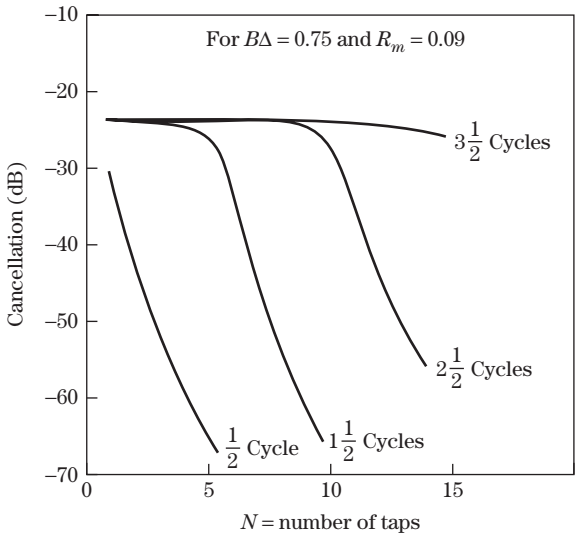


FIGURE 10-38 ■
Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.75$.

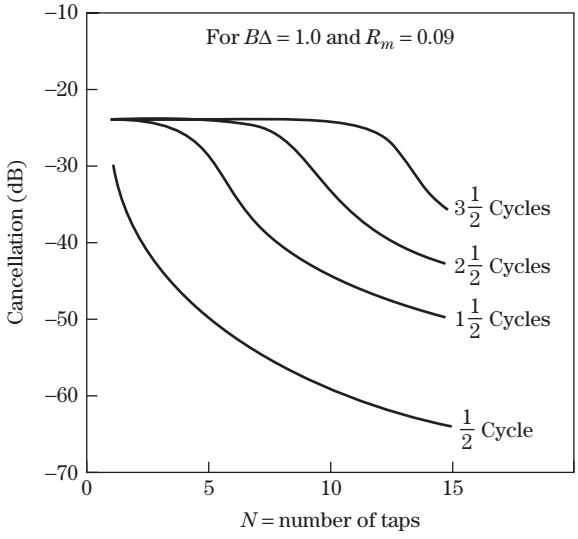


FIGURE 10-39 ■
Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 1.0$.

the amplitude mismatch model. The sufficient number of taps for the selected amplitude mismatch model was found empirically to be given by

$$N_{\text{sufficient}} \approx \left(\frac{N_r - 1}{2}\right)[7 - 4(B\Delta)] + 1 \tag{10.123}$$

where N_r is the number of half-cycles of ripple appearing in the mismatch model.

If there are a sufficient number of taps in the transversal filter, the cancellation performance improves when more taps are added depending on how well the resulting transfer function of the transversal filter matches the gain and phase variations of the channel mismatch model. Since the transversal filter transfer function resolution depends in part on the product $B\Delta$, a judicious selection of this parameter ensures that providing additional taps provides a better match (and hence a significant improvement in cancellation performance), whereas a poor choice results in very poor transfer function matching even with the addition of more taps.

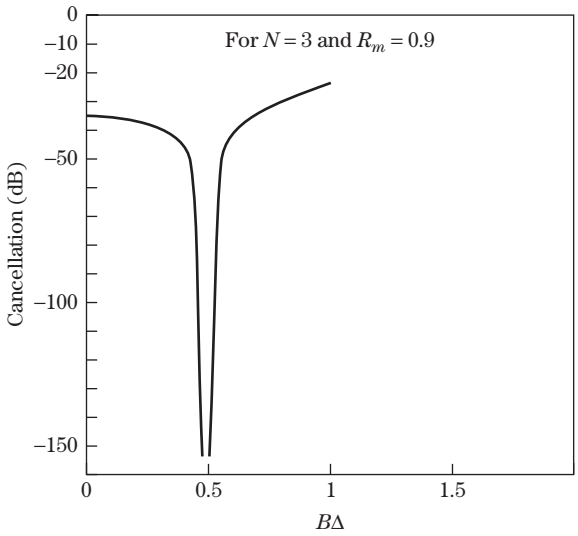


FIGURE 10-40 ■ Decibel cancellation versus $B\Delta$ for one-half-cycle amplitude mismatch model.

Taking the inverse Fourier transform of (10.114) $\mathfrak{F}^{-1}\{A(\omega)\}$ yields a time function corresponding to an autocorrelation function $f(t)$ that can be expressed as

$$f(t) = s(t) + Ks(t \pm T_0) \tag{10.124}$$

The results of Section 10.5.3 and equation (10.124) imply that $\Delta = T_0$ (or equivalently, $B\Delta =$ number cycles of ripple mismatch) if the product $B\Delta$ is to "match" the amplitude mismatch model. This result is illustrated in Figure 10-40 where decibel cancellation is plotted versus $B\Delta$ for a one-half-cycle ripple mismatch model. A pronounced minimum occurs at $B\Delta = \frac{1}{2}$ for $N = 3$ and $R_m = 0.9$.

When the number of cycles of mismatch ripple exceeds unity, the foregoing rule of thumb leads to the spurious conclusion that $B\Delta$ should exceed unity. Suppose, for example, there were two cycles of mismatch ripple for which it was desired to compensate. By setting $B\Delta = 2$ (corresponding to $B_f = \frac{1}{2}B$), two complete cycles for the transversal filter transfer function are found to occur across the cancellation bandwidth. By matching only one cycle of the channel mismatch, quite good matching of the entire mismatch characteristic occurs but at the price of sacrificing the ability to independently adjust the complex weights across the entire cancellation bandwidth, thereby reducing the ability to appropriately process broadband signals. Consequently, if the number of cycles of mismatch ripple exceeds unity, it is usually best to set $B\Delta = 1$ and to accept whatever improvement in cancellation performance can be obtained with that value, or increase the number of taps.

10.6.3 Example: Effects of Phase Mismatching

Let $\phi(\omega)$ corresponding to the phase error be characterized by

$$\phi(\omega) = \begin{cases} A\cos\omega T_0 & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases} \tag{10.125}$$

where A represents the peak number of degrees associated with the phase error ripples. This model corresponds to the error ripple model of (10.112) (with zero average value present).

Since

$$v_k = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} \exp(j\{A\cos\omega T_0 + \omega k\Delta\})\,d\omega \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.126}$$

it can easily be shown by defining

$$f(K, \text{sgn}) \triangleq \frac{\sin\pi[K + \text{sgn}\cdot(i - (N+1))B\Delta]}{\pi[K + \text{sgn}\cdot(i - (N+1))B\Delta]}$$

and $g(K) \triangleq f(K, +) + f(K, -)$ that

$$v_i = J_0(A)f(0, +) + jJ_1(A)g\left[\frac{2n+1}{2}\right] + \sum_{k=1}^{\infty}(-1)^k\left\{J_{2k}(A)g[k(2n+1)] + jJ_{2k+1}(A)g\left[(2k+1)\left(\frac{2n+1}{2}\right)\right]\right\} \tag{10.127}$$

where $J_n(\cdot)$ denotes a Bessel function of the nth order for $i = 1, 2, \ldots, 2N+1$.

10.6.4 Results for Compensation of Selected Phase Mismatch Model

The computer evaluation of the output residue power resulted in the performance summarized in Figures 10-41–10-43 for $B\Delta = 0.2$, 0.45, and 1.0 and $A = 5^\circ$. These figures present the decibel cancellation achieved as a function of the number of taps in the transversal filter and the number of cycles of phase ripple present across the cancellation bandwidth. The general nature of the curves appearing in Figures 10-41–10-43 is the same as that of Figures 10-36–10-39 for amplitude mismatching. Furthermore, just as in the amplitude mismatch case, a better channel transfer function fit can be obtained with the transversal filter when the mismatch characteristic has a fewer number of ripples.

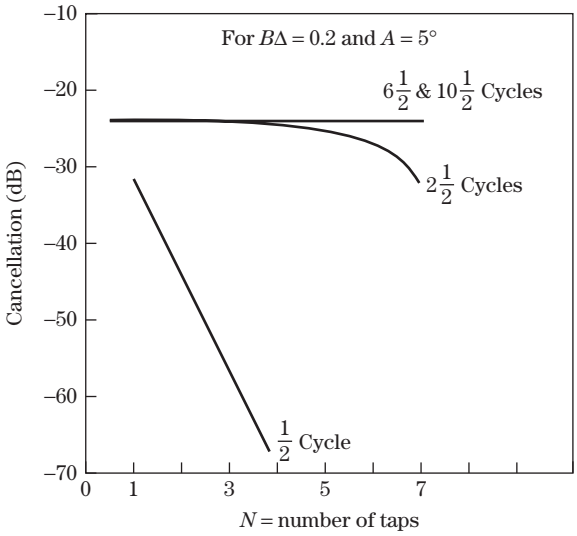


FIGURE 10-41 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 0.2$.

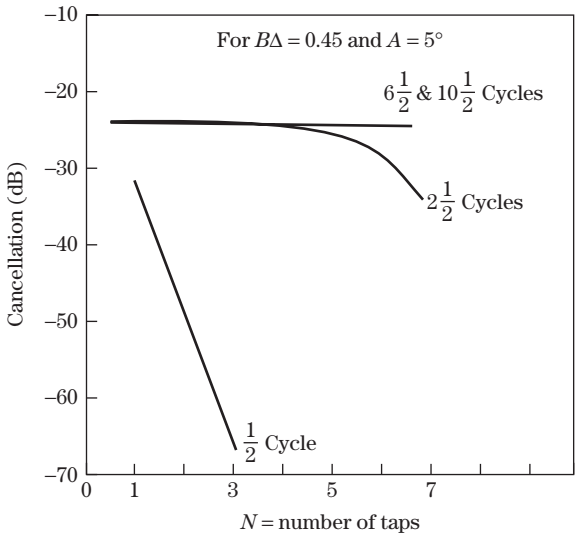


FIGURE 10-42 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 0.45$.

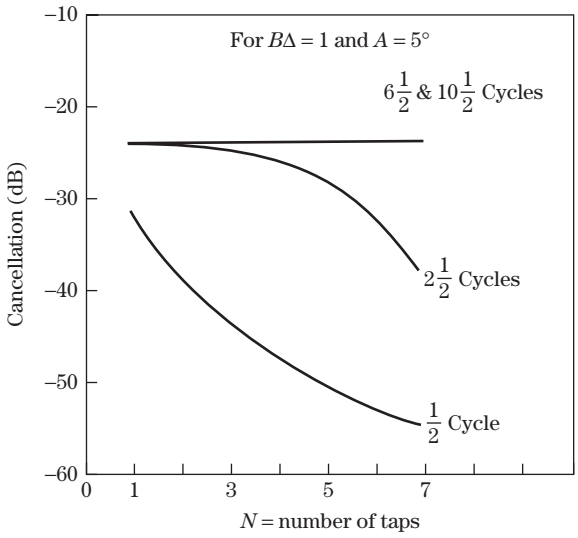


FIGURE 10-43 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 1.0$.

10.7 SUMMARY AND CONCLUSIONS

Array errors due to manufacturing tolerances distort the array pattern. To minimize these errors, the array must be calibrated at the factory and at regular intervals once deployed.

The transversal filter consisting of a sequence of weighted taps with intertap delay spacing offers a practical means for achieving the variable amplitude and phase weighting as a function of frequency that is required if an adaptive array system is to perform well against wideband interference signal sources. The distortionless channel transfer functions for a two-element array were derived. It was found that to ensure distortion-free response to a broadband signal the channel phase is a linear function of frequency, whereas the channel amplitude function is nearly flat over a 40% bandwidth. Quadrature hybrid processing provides adequate broadband signal response for signals having as much as 20% bandwidth. Tapped delay line processing is a practical necessity for 20% or more

bandwidth signals. A transversal filter provides an attractive means of compensating the system auxiliary channels for the undesirable effects of the following:

1. Multipath interference
2. Interchannel mismatch
3. Propagation delay across the array

For multipath interference, the value of the intertap delay is in the neighborhood of the delay time associated with the multipath ray. If the intertap delay time exceeds the multipath delay time by more than about 30% and the multipath delay time is appreciable, a severe loss of compensation capability is incurred. If the intertap delay is too small, then an excessive number of taps will be required for effective cancellation to occur. Since multipath delay having "small" values of associated time delay do not severely degrade the array performance, it is reasonable to determine the most likely values of multipath delay that will occur for the desired application and base the multipath compensation design on those delay times (assuming $B\Delta \leq 1$). For reflection coefficients of 0.5 and $BD_m = 0.45$, the use of five taps will ensure a -30 dB cancellation capability.

The results shown in Figures 10-31 and 10-32 indicate that array propagation delay effects are usually much easier to compensate than are multipath effects. This result occurs because multipath in effect introduces two (or more) signals in each channel (that are essentially uncorrelated if $BD_m \gg 1$), which require more degrees of freedom to adequately compensate.

The problem presented by interchannel mismatch is to obtain a transfer function with the transversal filter that succeeds in matching the amplitude and phase error characteristics exhibited among the various sensor channels. As might be expected, the more severe the mismatching between channels, the more difficult it is to achieve an acceptable degree of compensation. In particular, it is highly undesirable for more than $2\frac{1}{2}$ cycles of mismatch ripple to occur over the cancellation bandwidth; even this degree of mismatch requires seven taps on the transversal filter before a truly effective degree of compensation can be achieved. It may very well result that the best choice of intertap delay spacing for the interchannel mismatch characteristic of concern is far different from the optimum choice of intertap delay selected for multipath compensation; should this actually occur, it is necessary to adopt a compromise value for the intertap delay spacing. Such a compromise value for the intertap delay spacing hopefully results in an acceptable degree of compensation for both multipath and interchannel mismatch effects.

10.8 | PROBLEMS

Distortionless Transfer Functions

1. From (10.11) and (10.12) it immediately follows that $|H_1(\omega) = |H_2(\omega)|$, thereby yielding the pair of equations

$$f_1\{|H_1|, \alpha_1, \alpha_2, \theta_s\} = \exp(-j\omega T_1)$$

and

$$f_2\{|H_1|, \alpha_1, \alpha_2, \theta_i\} = 0$$

(a) Show from the previous pair of equations that $\alpha_1(\omega)$ and $\alpha_2(\omega)$ must satisfy

$$\alpha_2(\omega) - \alpha_1(\omega) = \frac{\pi\omega}{\omega_0} \sin\theta_i \pm n\pi$$

where n is any odd integer.

(b) Since the magnitude of $\exp(-j\omega T_1)$ must be unity, show using $f_1\{\ \} = \exp(-j\omega T_1)$ that (10.13) results.

(c) Show that the angle condition associated with $f_1\{\ \} = \exp(-j\omega T_1)$ yields (10.14).

(d) Show that substituting (10.14) into the results from part (a) yields (10.15).

2. For a three-element linear array, the overall transfer function encountered by the desired signal in passing through the array is

$$H_d(\omega) = H_1(\omega) + H_2(\omega)\exp\left(-j\frac{\omega d}{c}\sin\theta_s\right) + H_3(\omega)e\left(-j\frac{\omega 2d}{c}\sin\theta_s\right)$$

and the overall transfer function seen by the interference signal is

$$H_I(\omega) = H_1(\omega) + H_2(\omega)\exp\left(-j\frac{\omega d}{c}\sin\theta_i\right) + H_3(\omega)e\left(-j\frac{\omega 2d}{c}\sin\theta_i\right)$$

What does imposing the requirements (10.9) and (10.10) now imply for the three-channel transfer functions?

Hilbert Transform Relations

3. Prove the Hilbert transform relations given by (10.25)–(10.28).

4. Using (10.61), (10.62), and the results of (10.63)–(10.65), show that $\mathbf{R}_{ee}$ is given by (10.66).

5. Derive the correlation functions given by (10.78)–(10.80) for the signal environment assumptions (10.75) and (10.76)

6. Show that as the time–bandwidth product $B\Delta$ approaches zero, then the matrix $\mathbf{R}_{xx}(0)$ [whose elements are given by (10.80)] becomes singular so that matrix inversion cannot be accomplished.

Compensation for Channel Phase Errors

7. For the phase error $\phi(\omega)$ given by (10.125), show that v_k given by (10.127) follows from the application of (10.126).

8. Let $\phi(\omega)$ correspond to the phase error model be given by

$$\phi(\omega) = \begin{cases} A\left[1 - \cos\frac{2\omega}{B}\right] & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases}$$

Show that v_k of (10.126) is given by

$$v_k = \int_{-\pi B}^{\pi B}\left[\cos\left\{A\left(1-\cos\omega\frac{2}{B}\right)\right\} + j\sin\left\{A\left(1-\cos\omega\frac{2}{B}\right)\right\}\right]\exp(j\omega k\Delta_{d\omega})$$

Use the trigonometric identities

$$\cos\left[A - A\cos\omega\frac{2}{B}\right] = \cos A\cos\left[A\cos\omega\frac{2}{B}\right] + \sin A\sin\left[A\cos\omega\frac{2}{B}\right]$$

$$\sin\left[A - A\cos\omega\frac{2}{B}\right] = \sin A\cos\left[A\cos\omega\frac{2}{B}\right] - \cos A\sin\left[A\cos\omega\frac{2}{B}\right]$$

and the fact that

$$\cos(A\cos\omega T_0) = J_0(A) + 2\sum_{k=1}^{\infty}(-1)^k \cdot J_{2k}(A)\cos[(2k)\omega T_0]$$

$$\sin(A\cos\omega T_0) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(A)\cdot\cos[(2k+1)\omega T_0]$$

where $J_n(\cdot)$ denotes a Bessel function of the nth order and define

$$f(n, \text{sgn}) \triangleq \frac{\sin\pi[n + \text{sgn}\cdot(i-(N+1))B\Delta]}{\pi[n+\text{sgn}\cdot(i-(N+1))B\Delta]}$$

$$g(n) \triangleq f(n,+) + f(n,-)$$

to show that

$$\begin{aligned} v_i = {} & J_0(A)\cdot f(0,+)[\cos A + j\sin A] + J_1(A)\cdot g(2)[\sin A - j\cos A] \\ & + \sum_{k=1}^{\infty}(-1)^k\{J_{2k}(A)\cdot g(4k)[\cos A + j\sin A] \\ & + J_{2k+1}(A)\cdot g[(2k+1)2][\sin A - j\cos A]\} \\ \text{for } i = {} & 1, 2, \ldots, 2N+1 \end{aligned}$$

9. Let $\phi(\omega)$ corresponding to the phase error model be given by

$$\phi(\omega) = \begin{cases} b\omega^2(\pi B - |\omega|) & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases}$$

As before, it follows that

$$v_i = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\exp\{j[b\omega^2(\pi B - |\omega|) + \omega i\Delta]\}\,d\omega$$

Letting $u = \omega/\pi B$, applying Euler's formula, and ignoring all odd components of the resulting expression, show that

$$v_i = \int_0^1 \exp\left\{j\left[\frac{27A}{4}u^2(1-u)\right]\right\}\cos\pi[u(i-(N+1))B\Delta]du$$

where $A = 4b(\pi B/3)^3$ for $i = 1, 2, \ldots, 2N+1$. The foregoing equation for v_i can be evaluated numerically to determine the output residue power contribution due to the previous phase error model.

Computer Simulation Problems

10. A 30-element linear array ($d = 0.5\lambda$) has a 20 dB, $\overline{n} = 2$ Taylor taper applied at the elements. Plot the array factor when $\delta_n^a = 0.1$ and $\delta_n^a = 0.1$.

11. A 30-element linear array ($d = 0.5\lambda$) has a 30 dB, $\overline{n} = 7$ low sidelobe taper. Plot the array factors for a single element failure at (1) the edge and (2) the center of the array.

12. Find the location and heights of the quantization lobes for a 20-element array with $d = 0.5\lambda$ and the beam steered to $\theta = 3^\circ$ when the phase shifters have three, four, and five bits.

10.9 REFERENCES

[1] E. Brookner, *Antenna Array Fundamentals—Part 2, Practical Phased Array Antenna Systems,* E. Brookner, ed., Norwood, MA, Artech House, 1991

[2] R.J. Mailloux, "Array Grating Lobes Due to Periodic Phase, Amplitude, and Time Delay Quantization," *IEEE AP-S Trans.*, Vol. 32, December 1984, pp. 1364–1368.

[3] A. J. Boonstra, and A. J. van der Veen, "Gain Calibration Methods for Radio Telescope Arrays," *IEEE Signal Processing Trans.,* Vol. 51, No. 1, 2003, pp. 25–38.

[4] R. Sorace, "Phased Array Calibration," *IEEE AP-S Trans.,* Vol. 49, No. 4, 2001, pp. 517–525.

[5] T. Takahashi, Y. Konishi, S. Makino, et al., "Fast Measurement Technique for Phased Array Calibration," *IEEE AP-S Trans.,* Vol. 56, No. 7, 2008, pp. 1888–1899.

[6] M. Borkowski, "Solid-State Transmitters," *Radar Handbook*, M. I. Skolnick, ed., pp. 11.1–11.36, New York, McGraw Hill, 2008.

[7] N. Tyler, B. Allen, and H. Aghvami, "Adaptive Antennas: The Calibration Problem," *IEEE Communications Magazine,* Vol. 42, No. 12, 2004, pp. 114–122.

[8] N. Tyler, B. Allen, and A. H. Aghvami, "Calibration of Smart Antenna Systems: Measurements and Results," *IET Microwaves, Antennas & Propagation,* Vol. 1, No. 3, 2007, pp. 629–638.

[9] R. L. Haupt, *Antenna Arrays: A Computational Approach*, New York, Wiley, 2010.

[10] W. T. Patton and L. H. Yorinks, "Near-Field Alignment of Phased-Array Antennas," *IEEE AP-S Trans.,* Vol. 47, No. 3, 1999, pp. 584–591.

[11] W. E. Rodgers and R. T. Compton, Jr., "Tapped Delay-Line Processing in Adaptive Arrays," Report 3576-3, April 1974, prepared by The Ohio State University Electro Science Laboratory, Department of Electrical Engineering under Contract N00019-73-C-0195 for Naval Air Systems Command.

[12] W. E. Rodgers and R. T. Compton, Jr., "Adaptive Array Bandwidth with Tapped Delay-Line Processing," Report 3832-3, May 1975, prepared by The Ohio State University Electro Science Laboratory, Department of Electrical Engineering under Contract N00019-74-C-0141 for Naval Air Systems Command.

[13] W. E. Rodgers and R. T. Compton, Jr., "Adaptive Array Bandwidth with Tapped Delay-Line Processing," *IEEE Trans. Aerosp. Electron. Syst*., Vol. AES-15, No. 1, January 1979, pp. 21–28.

[14] T. G. Kincaid, "The Complex Representation of Signals," General Electric Report No. R67EMH5, October 1966, HMED Publications, Box 1122 (Le Moyne Ave.), Syracuse, NY, 13201.

[15] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York, McGraw-Hill, 1965, Ch. 7.

[16] H. Steyskal & J. Herd, "Mutual Coupling Compensation in Small Array Antennas," *IEEE Trans. Ant. & Prop.,* Vol. AP-38, No. 12, December 1995, pp. 603–606.

[17] R. A. Monzingo, "Transversal Filter Implementation of Wideband Weight Compensation for CSLC Applications," unpublished Hughes Aircraft Interdepartmental Correspondence Ref. No. 78-1450.10/07, March 1978.

[18] A. M. Vural, "Effects of Perturbations on the Performance of Optimum/Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst*., Vol. AES-15, No. 1, January 1979, pp. 76–87.

PART III

Advanced Topics

CHAPTER

10

Compensation of Adaptive Arrays

Chapter Outline

Narrowband adaptive arrays need only one complex adaptive weight in each element channel. Broadband adaptive arrays, however, require tapped delay lines (transversal filters) in each element channel to make frequency-dependent amplitude and phase adjustments. The analysis presented so far assumes that each element channel has identical electronics and no reflected signals. Unfortunately, the electrical characteristics of each channel are slightly different and lead to "channel mismatching" in which significant differences in frequency-response characteristics from channel to channel may severely degrade an array's performance without some form of compensation. This chapter starts with an analysis of array errors and then addresses array calibration and frequency-dependent mismatch compensation using tapped delay line processing, which is important for practical broadband adaptive array designs.

The number of taps used in a tapped delay line processor depends on whether the tapped delay line compensates for broadband channel mismatch effects or for the effects of multipath and finite array propagation delay. Minimizing the number of taps required for a specified set of conditions is an important practical design consideration, since each additional tap (and associated weighs) increases the cost and complexity of the adaptive array system.

10.1 ARRAY ERRORS

Array errors result from the manufacturing tolerances defined by the materials, processes, and construction of the components in an array. These small errors are random, because the manufacturing techniques employed have very tight tolerances. The random differences between any components distort the signal path by adding phase and amplitude errors as well as noise to each signal. These types of errors are static, because once measured they remain relatively unchanged over the life of the component. Higher frequencies have tighter tolerances for phase distortion than lower frequencies, because the errors are a function of wavelength. Not only are the accuracy of the dimensions of the components important, but the accuracy of the values of the constitutive parameters of the components are also important. For instance, the dielectric constant determines the wavelength and hence the phase of the signal passing through it, so an error in the dielectric constant produces a phase error.

Dynamic errors change with time and are primarily due to changes in temperature. Online calibration corrects for these dynamic errors also takes care of any drift in the static errors. The dynamic errors are also frequency dependent. The effects of temperature are smallest at the center frequency and increase as the frequency migrates away from the center frequency.

10.1.1 Error Analysis

Random errors that affect arrays fall into four categories:

1. Random amplitude error, δ_n^a
2. Random phase error, δ_n^p
3. Random position error, δ_n^s
4. Random element failure, $P_n^e = \begin{cases} 1 & \text{element functioning properly} \\ 0 & \text{element failure} \end{cases}$

The first three types of random errors fit into the array factor as perturbations to the array weights and element locations

$$AF_{err} = \sum_{n=1}^{N} \left(a_n + \delta_n^a\right) e^{j\left(p_n + \delta_n^p\right)} e^{jk\left(s_n + \delta_n^s\right)u} \tag{10.1}$$

Element failures result when an element no longer transmits or receives. The probability that an element has failed, $1 - P_e$, is the same as a root mean square (rms) amplitude error, $\overline{\delta_n^{a^2}}$. Position errors are not usually a problem, so a reasonable formula to calculate the rms sidelobe level of the array factor for amplitude and phase errors with element failures is [1]

$$sll_{rms} = \frac{(1 - P_e) + \overline{\delta_n^{a^2}} + P_e \overline{\delta_n^{p^2}}}{P_e \left(1 - \overline{\delta_n^{p^2}}\right) \eta_t N} \tag{10.2}$$

Figure 10-1 is an example of a typical corporate-fed array. A random error that occurs at one element is statistically uncorrelated with a random error that occurs in another element in the array as long as that error occurs after the last T junction and before an

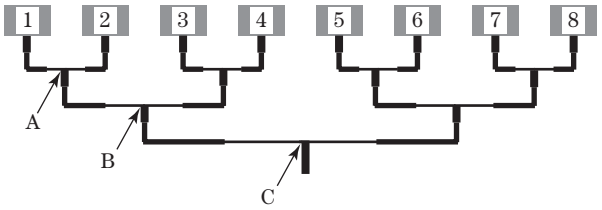


FIGURE 10-1 ■ Corporate-fed array with random errors.

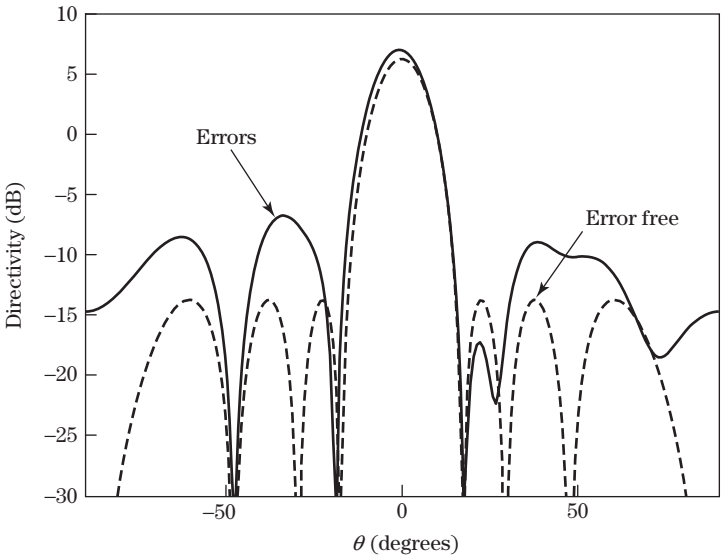


FIGURE 10-2 ■ Array factor with random, uncorrelated errors superimposed on the error-free array factor.

element. If a random error occurs prior to A, for instance, then the random error becomes correlated between the elements that share the error. For instance, a random error between A and B results in a random correlated error shared by elements 1 and 2. Likewise, a random error between B and C results in a random correlated error shared by elements 1, 2, 3, and 4.

As an example, consider an eight-element, 20 dB Chebyshev array that has elements spaced $\lambda/2$ apart. If the random errors are represented by $\delta_n^a = 0.15$ and $\delta_n^p = 0.15$, then an example of the array factor with errors is shown in Figure 10-2. Note that the random errors lower the main beam directivity, induce a slight beam-pointing error, increase the sidelobe levels, and fill in some of the nulls.

10.1.2 Quantization Errors

Phase shifters and attenuators have N_{bp} control bits with the least significant bits given by

$$\Delta_a = 2^{-N_{ba}} \tag{10.3}$$

$$\Delta_p = 2\pi \times 2^{-N_{bp}} \tag{10.4}$$

If the difference between the desired and quantized amplitude weights is a uniformly distributed random number with the bounds being the maximum amplitude error of $\pm\Delta_a/2$, then the rms amplitude error is $\delta_n^a = \Delta_a/\sqrt{12}$. The quantization error is random only when no two adjacent elements receive the same quantized phase shift. The difference

between the desired and quantized phase shifts is treated as uniform random variables between $\pm\Delta_p/2$. As with the amplitude error, the random phase error formula in this case is $\delta_n^p = \Delta_p/\sqrt{12}$. Substituting this error into (b) yields the rms sidelobe level.

The phase quantization errors become correlated when the beam steering phase shift is small enough that groups of adjacent elements have their beam steering phase quantized to the same level. This means that N/N_Q subarrays of N_Q elements receive the same phase shift. The grating lobes due to these subarrays occur at [2]

$$\sin\theta_m = \sin\theta_s \pm \frac{m\lambda}{N_Q d_e} = \sin\theta_s\left[1 \pm \frac{m\,(N-1)\,2^{N_{bp}}}{N}\right] \simeq \sin\theta_s\left(1 \pm m2^{N_{bp}}\right) \quad (10.5)$$

The approximation in (10.5) assumes that the array has many elements. For large scan angles, quantization lobes do not form, because the element-to-element phase difference appears random. The relative peaks of the quantization lobes are given by [1]

$$AF_N^{QL} = \frac{1}{2^{N_p}}\sqrt{\frac{\sqrt{1-\sin\theta^2}}{\sqrt{1-\sin\theta_s^2}}} \quad (10.6)$$

Figure 10-3 shows an array factor with a 20 dB $\overline{n} = 3$ Taylor amplitude taper for a 20-element, $d = 0.5\lambda$ array with its beam steered to $\theta = 3^\circ$ when the phase shifters have three bits. Four quantization lobes appear. The quantization lobes decrease when higher-precision phase shifters are used and when the beam is steered to higher angles.

Significant distortion also results from mutual coupling, variation in group delay between filters, differences in amplifier gain, tolerance in attenuator accuracy, and aperture jitter in a digital beamforming array. Aperture jitter is the timing error between samples in an analog-to-digital (A/D) converter. Without calibration, beamforming or estimation of the direction of arrival (DOA) of the signal is difficult, as the internal distortion is uncorrelated with the signal. As a result, the uncorrelated distortion changes the weights at each element and therefore distorts the array pattern.

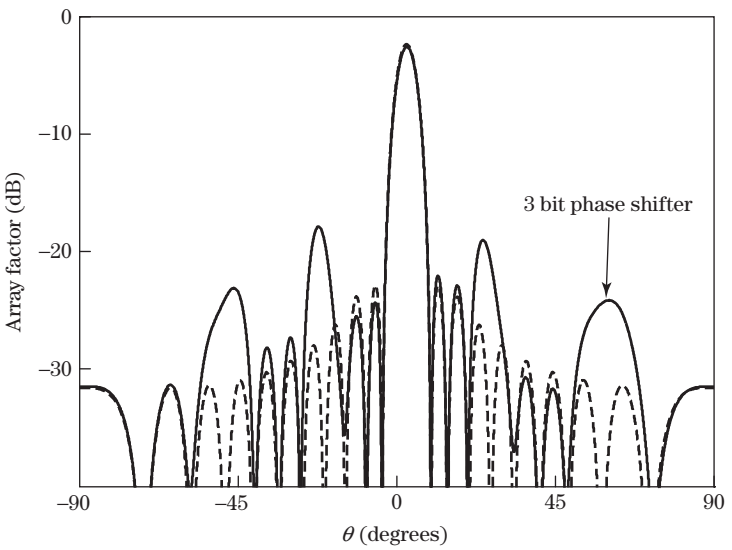


FIGURE 10-3 ■ Array factor steered to 3 degrees with three-bit phase shifters compared with phase shifters with infinite precision.

10.2 ARRAY CALIBRATION

A phased array needs calibrated before it can generate an optimum coherent beam. Calibration involves tuning, for example, the phase shifters, attenuators, or receivers to maximize the gain and to create the desired sidelobe response. Offline calibration takes care of the static errors and is done at the factory or on deployment. Narrowband calibration is applied at the center frequency of operation. Broadband calibration is applied over the whole operating bandwidth of the array. The calibrated phase settings are stored for all beam steering angles. Temperature causes drift in the component characteristics over time, so the array requires periodic recalibration. The gain of the radiofrequency (RF) channels must be accurately controlled to avoid nonlinearities arising from saturation of components, because these nonlinearities cannot be removed.

The top vector in Figure 10-4 shows the resulting uncalibrated array output when the individual five-element vectors have random amplitude and phase errors. When the array is calibrated (bottom vector in Figure 10-4), then the individual element vectors are the same length and align. As a result, the calibrated array output vector magnitude is maximized, and its phase is zero. Methods for performing array calibration use a calibrated source, signal injection, or near-field scanning. These approaches are discussed in the following sections.

10.2.1 Calibrated Source

A known calibration source radiates a calibration signal to all elements in the array [3]. Figure 10-5 shows a calibration source in the far field of an array. At regular intervals, the main beam is steered to receive the calibration source signal. Alternatively, a multibeam antenna can devote one beam to calibration. Calibration with near-field sources requires that distance and angular differences be taken into account. If the calibration source is in the far field, then the phase shifters are set to steer the beam in the direction of the source. In either case, each element toggles through all of its phase settings until the output signal is maximized. The difference between the steering phase and the phase that yields the maximum signal is the calibration phase.

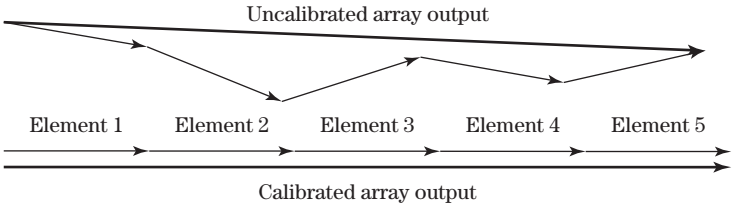


FIGURE 10-4 ■ The uncalibrated array output is less than the calibrated array output, because errors in the uncalibrated array do not allow the signal vectors from the elements to align.

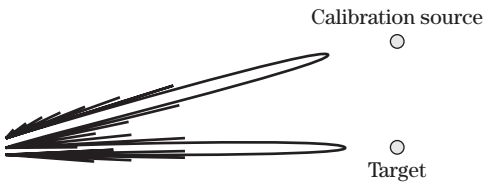


FIGURE 10-5 ■ Far-field calibration.

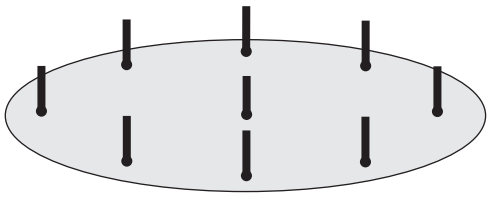


FIGURE 10-6 ■
Layout of the smart antenna test bed.

Making power measurements for every phase setting at every element in an array is extremely time-consuming. Calibration techniques that measure both amplitude and phase of the calibrated signal tend to be much faster. Accurately measuring the signal phase is reasonable in an anechoic chamber but difficult in the operational environment. Measurements at four orthogonal phase settings yield sufficient information to obtain a maximum likelihood estimate of the calibration phase [4]. The element phase error is calculated from power measurements at the four phase states, and the procedure is repeated for each element in the array. Additional measurements improve signal-to-noise ratio, and the procedure can be repeated to achieve desired accuracy within resolution of the phase shifters, since the algorithm is intrinsically convergent.

Another approach uses amplitude-only measurements from multiple elements to find the complex field at an element [5]. The first step measures the power output from the array when the phases of multiple elements are successively shifted with the different phase intervals. Next, the measured power variation is expanded into a Fourier series to derive the complex electric field of the corresponding elements. The measurement time reduction comes at the expense of increased measurement error.

Transmit/receive module calibration is an iterative process that starts with adjusting the attenuators for uniform gain at the elements [6]. The phase shifters are then adjusted to compensate for the insertion phase differences at each element. Ideally, when calibrating the array, the phase shifter's gain remains constant as the phase settings are varied, but the attenuator's insertion phase can vary as a function of the phase setting. This calibration should be done across the bandwidth, range of operating temperatures, and phase settings. If the phase shifter's gain varies as a function of setting, then the attenuators need to be compensated as well. After iterating over this process, all the calibration settings are saved and applied at the appropriate times.

Figure 10-6 shows an eight-element uniform circular array (UCA) in which a center element radiates a calibration signal to the other elements in the array [7]. Since the calibration source is in the center of the array, the signal path from the calibration source to each element is identical. As previously noted, random errors are highly dependent on temperature [8]. An experimental model of the UCA in Figure 10-6 was placed inside a temperature-controlled room and calibrated at 20°C. The measured amplitude and phase errors at three temperatures are shown in Figure 10-7 and Figure 10-8, respectively. Increasing the temperature of the room to 25°C then to 30°C without recalibration increases the errors shown in Figure 10-7 and Figure 10-8. This experiment demonstrates the need of dynamic calibration in a smart antenna array.

10.2.2 Signal Injection

Calibrating with a radiating source is difficult, because the calibration signal transmission/reception depends on the environment. One technique commonly used in digital

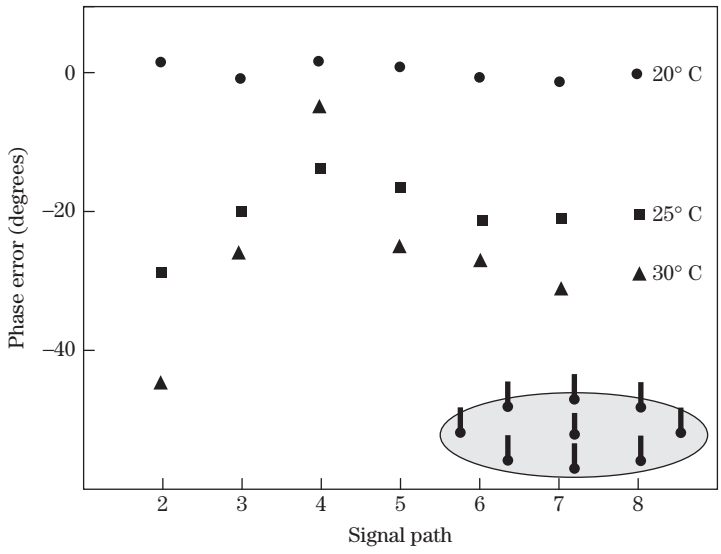


FIGURE 10-7 ■ Amplitude error for the UCA antenna as the system temperature changes from 20°C with calibration to 25°C without recalibration and to 30°C without recalibration.

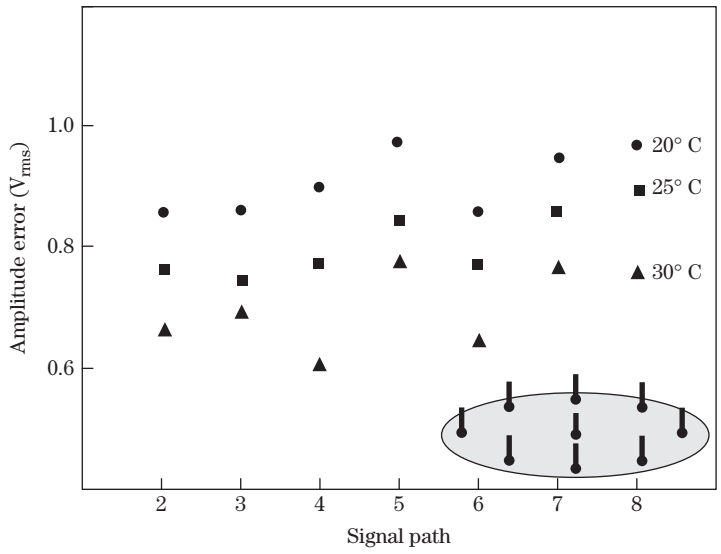


FIGURE 10-8 ■ Phase error for the UCA antenna as the system temperature changes from 20°C with calibration to 25°C without recalibration and to 30°C without recalibration.

beamforming arrays is injecting a calibration signal into the signal path of each element in the array behind each element as shown in Figure 10-9 [9]. This technique provides a high-quality calibration signal for the circuitry behind the element. Unfortunately, it does not calibrate for the element patterns that have significant variations due to mutual coupling, edge effects, and multipath.

10.2.3 Near-Field Scan

A planar near-field scanner positioned very close to the array moves a probe directly in front of each element to measure the amplitude and phase of all the elements [10]. The measured field is transformed back to the aperture to recreate the field radiated at each

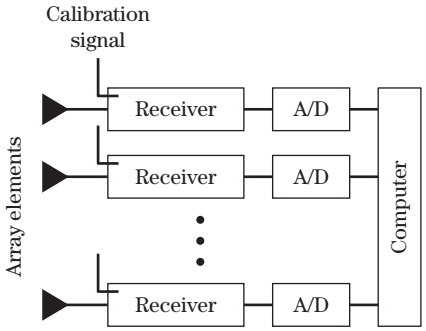


FIGURE 10-9 ■ Inserting a calibration signal into the signal paths in a digital beamformer.

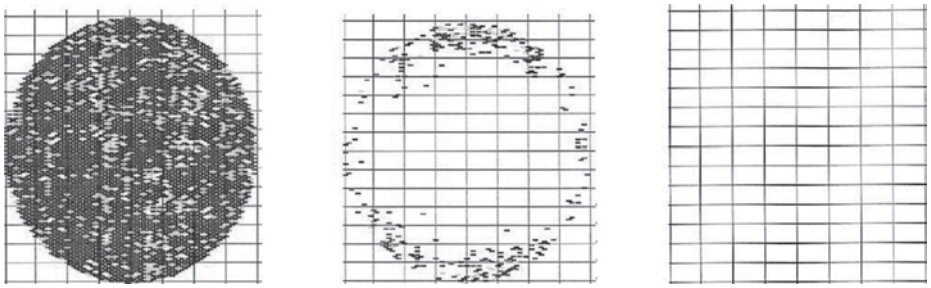


FIGURE 10-10 ■ Alignment results (measured phase deviation from desired value). a: Unaligned. b: After single alignment with uncorrected measurements. c: After alignment with fully corrected measurements. From W. T. Patton and L. H. Yorinks, "Near-field alignment of phased-array antennas," *IEEE Transactions on Antennas and Propagation*, Vol. 47, No. 3, March 1999, pp. 584–591.

element. The calibration algorithm iterates between the measured phase and the array weights until the phase at all the elements is the same. Figure 10-10 shows the progression of the phase correction algorithm from left to right. The picture on the left is uncalibrated, the center picture is after one iteration, and the picture on the right is after calibration is completed. This techniques is exceptionally good at correcting static errors prior to deploying an antenna is not practical for dynamic errors.

10.3 | BROADBAND SIGNAL PROCESSING CONSIDERATIONS

Broadband arrays use tapped delay lines that have frequency-dependent transfer functions. Array performance is a function of the number of taps, the tap spacing, and the total delay in each channel. The minimum number of taps required to obtain satisfactory performance for a given bandwidth may be determined as discussed in Section 2.5. The discussion of broadband signal processing considerations given here follows the treatment of this subject given by Rodgers and Compton [11–13]. The ideal (distortionless) channel transfer functions are derived; adaptive array performance using quadrature hybrid processing and two-, three-, and five-tap delay line processing are considered; and results and conclusions for broadband signal processing are then discussed.

10.3.1 Distortionless Channel Transfer Functions

The element channels of the two-element array in Figure 10-11 are represented by the transfer functions $H_1(\omega)$ and $H_2(\omega)$. Let the desired signal arrive at θ_s, measured relative to the array face normal. The array carrier frequency is ω_0, and the point sources spacing is $d = \lambda_0/2 = \pi b/\omega_0$, where $\mathfrak{v}$ is the wavefront propagation velocity.

From the point of view of the desired signal, the overall transfer function encountered in passing through the array of Figure 10-11 is

$$H_d(\omega) = H_1(\omega) + H_2(\omega) \exp\left(-j\frac{\omega d}{\mathfrak{v}} \sin\theta_s\right) \tag{10.7}$$

and the overall transfer function seen by the interference signal is

$$H_I(\omega) = H_1(\omega) + H_2(\omega) \exp\left(-j\frac{\omega d}{\mathfrak{v}} \sin\theta_i\right) \tag{10.8}$$

Now require that

$$H_d(\omega) = \exp(-j\omega T_1) \tag{10.9}$$

and

$$H_I(\omega) = 0 \tag{10.10}$$

By choosing $H_d(\omega)$ according to (10.9), the desired signal is permitted to experience a time delay T_1 in passing through the array but otherwise remains undistorted. Choosing $H_I(\omega) = 0$ results in complete suppression of the interference signal from the array output.

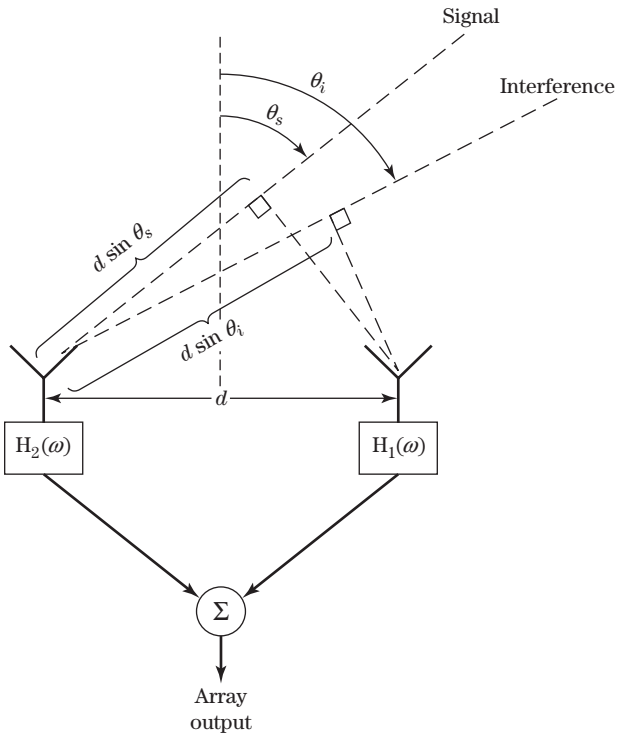


FIGURE 10-11 ■ Two-element array.

To determine whether it is possible to select $H_1(\omega)$ and $H_2(\omega)$ to satisfy (10.9) and (10.10), solve (10.9) and (10.10) for $H_1(\omega)$ and $H_2(\omega)$. Setting $H_1(\omega) = |H_1(\omega)| \exp[j\alpha_1(\omega)]$ and $H_2(\omega) = |H_2(\omega)| \exp[j\alpha_2(\omega)]$ results in

$$|H_1(\omega)| \exp[j\alpha_1(\omega)] + |H_2(\omega)| \exp\left\{j\left[\alpha_2(\omega) - \frac{\pi\omega}{\omega_0}\sin\theta_s\right]\right\} = \exp(-j\omega T_1) \quad (10.11)$$

$$|H_1(\omega)| \exp[j\alpha_1(\omega)] + |H_2(\omega)| \exp\left\{j\left[\alpha_2(\omega) - \frac{\pi\omega}{\omega_0}\sin\theta_i\right]\right\} = 0 \quad (10.12)$$

To satisfy (10.9) and (10.10), it follows from (10.11) and (10.12) (as shown by the development outlined in the Problems section) that

$$H_1(\omega) = H_2(\omega) = \frac{1}{\sqrt{2\left(1 - \cos\left[\frac{\pi\omega}{\omega_0}(\sin\theta_i - \sin\theta_s)\right]\right)}} \quad (10.13)$$

$$\alpha_2(\omega) = \frac{\pi}{2}\left(\frac{\omega}{\omega_0}\right)[\sin\theta_s + \sin\theta_i] \mp n\frac{\pi}{2} - \omega T_1 \quad (10.14)$$

$$\alpha_1(\omega) = \frac{\pi}{2}\left(\frac{\omega}{\omega_0}\right)[\sin\theta_s - \sin\theta_i] \pm n\frac{\pi}{2} - \omega T_1 \quad (10.15)$$

where n is any odd integer. This result means that the amplitude of the ideal transfer functions are equal and frequency dependent. Equations (10.14) and (10.15) furthermore show that the phase of each filter is a linear function of frequency with the slope dependent on the spatial arrival angles of the signals as well as on the time delay T_1 of the desired signal.

Plots of the amplitude function in (10.13) are shown in Figure 10-12 for two choices of arrival angles ($\theta_s = 0^\circ$ and $\theta_s = 80^\circ$), where it is seen that the amplitude of the distortionless transfer function is nearly flat over a 40% bandwidth when the desired signal is at broadside ($\theta_s = 0^\circ$) and the interference signal is 90° from broadside ($\theta_i = 90^\circ$). Examination of (10.13) shows that whenever $(\sin\theta_I - \sin\theta_s)$ is in the neighborhood of ± 1,

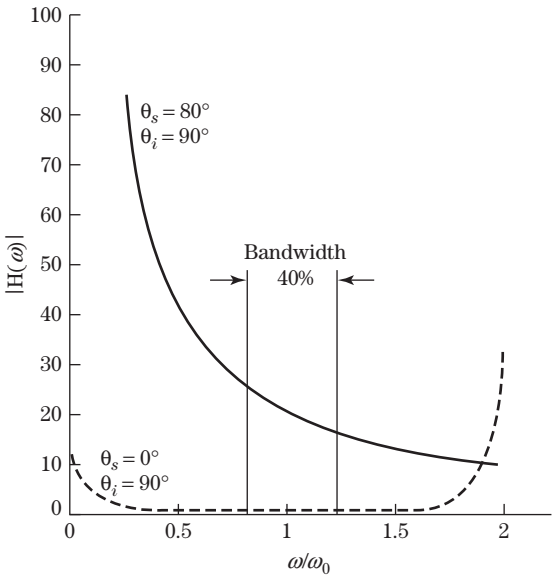


FIGURE 10-12 ■ Distortionless transfer function amplitude versus normalized frequency for $d = \lambda_0/2$. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

then the resulting amplitude function will be nearly flat over the 40% bandwidth region. If, however, both the desired and interference signals are far from broadside (as when $\theta_d = 80^\circ$ and $\theta_i = 90^\circ$), then the amplitude function is no longer flat.

The degree of "flatness" of the distortionless filter amplitude function is interpreted in terms of the signal geometry with respect to the array sensitivity pattern. In general, when the phases of $H_1(\omega)$ and $H_2(\omega)$ are adjusted to yield the maximum undistorted response to the desired signal, the corresponding array sensitivity pattern will have certain nulls. The distortionless filter amplitude function is then the most flat when the interference signal falls into one of these pattern nulls.

Equation (10.13) furthermore shows that singularities occur in the distortionless channel transfer functions whenever $(\omega/\omega_0)\pi(\sin\theta_i - \sin\theta_s) = n2\pi$ where $n = 0, 1, 2, \ldots$. The case when $n = 0$ occurs when the desired and interference signals arrive from exactly the same direction, so it is hardly surprising that the array would experience difficulty trying to receive one signal while nulling the other in this case. The other cases when $n = 1, 2, \ldots$, occur when the signals arrive from different directions, but the phase shifts between elements differ by a multiple of 2π radians at some frequency ω in the signal band.

The phase functions $\alpha_1(\omega)$ and $\alpha_2(\omega)$ of (10.14) and (10.15) are linear functions of frequency. When $T_1 = 0$, the phase slope of $H_1(\omega)$ is proportional to $\sin\theta_s - \sin\theta_i$, whereas that of $H_2(\omega)$ is proportional to $\sin\theta_i + \sin\theta_s$. Consequently, when the desired signal is broadside, $\alpha_1(\omega) = -\alpha_2(\omega)$. Furthermore, the phase difference between $\alpha_1(\omega)$ and $\alpha_2(\omega)$ is also a linear function of frequency, a result that would be expected since this allows the interelement phase shift (which is also a linear function of frequency) to be canceled.

10.3.2 Quadrature Hybrid and Tapped Delay Line Processing for a Least Mean Squares Array

Consider a two-element adaptive array using the least mean squares (LMS) algorithm. If $\mathbf{w}$ is the column vector of array weights, $\mathbf{R}_{xx}$ is the correlation matrix of input signals to each adaptive weight, and $\mathbf{r}_{xd}$ is the cross-correlation vector between the received signal vector $\mathbf{x}(t)$ and the reference signal $d(t)$, then as shown in Chapter 3 the optimum array weight vector that minimizes $E\{\varepsilon^2(t)\}$ (where $\varepsilon(t) = d(t)$—array output) is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd} \tag{10.16}$$

If the signal appearing at the output of each sensor element consists of a desired signal, an interference signal, and a thermal noise component (where each component is statistically independent of the others and has zero mean), then the elements of $\mathbf{R}_{xx}$ can readily be evaluated in terms of these component signals.

Consider the tapped delay line employing real (instead of complex) weights shown in Figure 10-13. Since each signal $x_i(t)$ is just a time-delayed version of $x_1(t)$, it follows that

$$\left.\begin{aligned} x_2(t) &= x_1(t - \Delta) \\ x_2(t) &= x_1(t - 2\Delta) \\ &\vdots \\ x_L(t) &= x_1[t - (L-1)\Delta] \end{aligned}\right\} \tag{10.17}$$

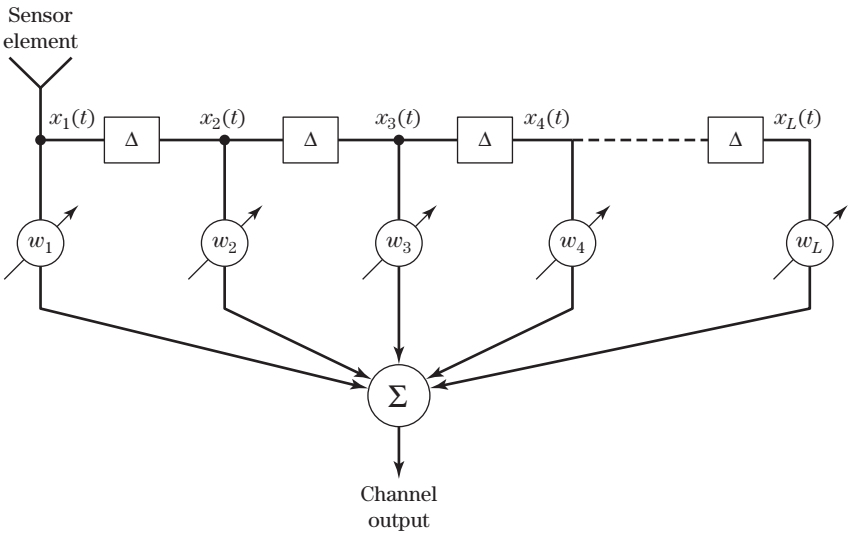


FIGURE 10-13 ■ Tapped delay line processor for a single-element channel having real adaptive weights.

Now since the elements of $\mathbf{R}_{xx}$ are given by

$$r_{x_i x_j} \triangleq E\{x_i(t)x_j(t)\} \tag{10.18}$$

it follows from (10.17) that

$$r_{x_i x_j} = r_{x_1 x_1}(\tau_{ij}) \tag{10.19}$$

where $r_{x_1 x_1}(\tau_{ij})$ is the autocorrelation function of $x_1(t)$, and τ_{ij} is the time delay between $x_i(t)$ and $x_j(t)$. Furthermore, $r_{x_i x_i}(\tau_{ij})$ is the sum of three autocorrelation functions—those of the desired signal, the interference, and the thermal noise so that

$$r_{x_1 x_1}(\tau_{ij}) = r_{dd}(\tau_{ij}) + r_{II}(\tau_{ij}) + r_{nn}(\tau_{ij}) \tag{10.20}$$

For the elements of $\mathbf{R}_{xx}$ corresponding to $x_i(t)$ and $x_j(t)$ from different element channels, $r_{x_i x_j}$ consists only of the sum of the autocorrelation functions of the desired signal and the interference signal (with appropriate delays) but not the thermal noise since the element noise from channel to channel is uncorrelated. Thus, for signals in different element channels

$$r_{x_i x_j}(\tau_{ij}) = r_{dd}(\tau_{d_{ij}}) + r_{II}(\tau_{I_{ij}}) \tag{10.21}$$

where $\tau_{d_{ij}}$ denotes the time delay between $x_i(t)$ and $x_j(t)$ for the desired signal, and $\tau_{I_{ij}}$ denotes the time delay between $x_i(t)$ and $x_j(t)$ for the interference signal (these two time delays will in general be different due to the different angles of arrival of the two signals). Only when $x_i(t)$ and $x_j(t)$ are from the same array element channel will $\tau_{d_{ij}} = \tau_{I_{ij}}$ (which may then be denoted by τ_{i_j}).

Next, consider the quadrature hybrid array processor depicted in Figure 10-14. Let $x_1(t)$ and $x_3(t)$ denote the in-phase signal components and $x_2(t)$ and $x_4(t)$ denote the quadrature-phase signal components of each of the elements output signals. Then the in-phase and quadrature components are related by

$$\left.\begin{aligned} x_2(t) &= \check{x}_1(t) \\ x_4(t) &= \check{x}_3(t) \end{aligned}\right\} \tag{10.22}$$

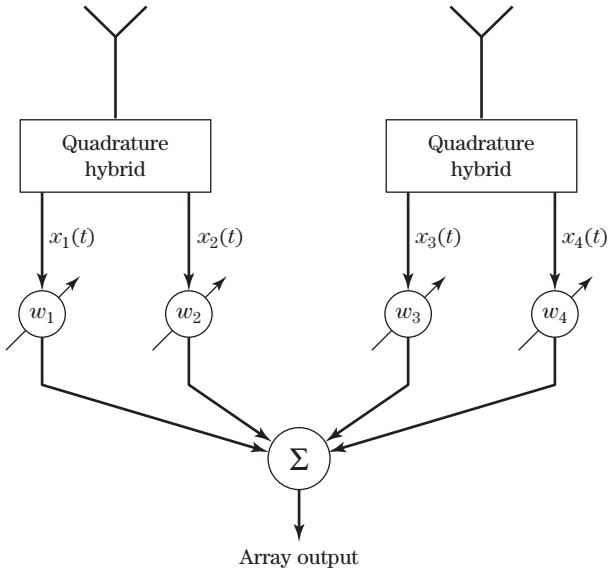


FIGURE 10-14 ■ Quadrature hybrid processing for a two-element array.

The symbol ˇ denotes the Hilbert transform

$$\check{x}(t) \triangleq \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau \tag{10.23}$$

where the previous integral is regarded as a Cauchy principal value integral. The various elements of the correlation matrix

$$r_{x_i x_j} = E\{x_i(t)x_j(t)\} \tag{10.24}$$

can then be found by making use of certain Hilbert transform relations as follows [14,15]:

$$E\{\check{x}(t)\check{y}(s)\} = E\{x(t)y(s)\} \tag{10.25}$$
$$E\{\check{x}(t)y(s)\} = -E\{x(t)\check{y}(s)\} \tag{10.26}$$

so that

$$E\{\check{x}(t)x(t)\} = 0 \tag{10.27}$$
$$E\{x(t)\check{y}(s)\} = \check{E}\{x(t)y(s)\} \tag{10.28}$$

where $\check{E}\{x(t)y(s)\}$ denotes the Hilbert transform of $r_{xy}(\tau)$ where $\tau = s - t$. With the previous relations and from (10.22) it then follows that

$$r_{x_1x_1} = E\{x_1(t)x_1(t)\} = r_{x_1x_1}(0) \tag{10.29}$$
$$r_{x_1x_2} = E\{x_1(t)x_2(t)\} = E\{x_1(t)\check{x}_1(t)\} = 0 \tag{10.30}$$
$$r_{x_2x_2} = E\{x_2(t)x_2(t)\} = E\{\check{x}_1(t)\check{x}_1(x)\} \tag{10.31}$$
$$= E\{x_1(t)x_1(t)\} = r_{x_1x_1}(0)$$

where $r_{x_1x_1}(\tau)$ is the autocorrelation function of $x_1(t)$ given by (10.20).

When two different sensor element channels are involved [as with $x_1(t)$ and $x_3(t)$, for example], then

$$E\{x_1(t)\, x_3(t)\} = r_{dd}(\tau_{d_{13}}) + r_{II}(\tau_{I_{13}}) \tag{10.32}$$

where $\tau_{d_{13}}$ and $\tau_{I_{13}}$ represent the spatial time delays between the sensor elements of Figure 10-14 for the desired and interference signals, respectively. Similarly

$$\begin{aligned} E\{x_1(t)x_4(t)\} = E\{x_1(t)\check{x}_3(t)\} &= \check{E}\{x_1(t)x_3(t)\} \\ &= \check{r}_{dd}(\tau_{d_{13}}) + \check{r}_{II}(\tau_{I_{13}}) \qquad (10.33) \\ E\{x_2(t)x_3(t)\} = E\{\check{x}_1(t)x_3(t)\} &= -E\{\check{x}_1(t)x_3(t)\} \\ &= -\check{E}\{x_1(t)x_3(t)\} = -\check{r}_{dd}(\tau_{d_{13}}) - \check{r}_{II}(\tau_{I_{13}}) \qquad (10.34) \\ E\{x_2(t)x_4(t)\} = E\{\check{x}_1(t)\check{x}_3(t)\} &= x\{x_1(t)x_3(t)\} \\ &= r_{dd}(\tau_{d_{13}}) + r_{II}(\tau_{I_{13}}) \qquad (10.35) \end{aligned}$$

Now consider the cross-correlation vector $\mathbf{r}_{xd}$ defined by

$$\mathbf{r}_{xd} \triangleq E\begin{bmatrix} x_1(t)d(t) \\ x_2(t)d(t) \\ \vdots \\ x_{2N}(t)d(t) \end{bmatrix} \tag{10.36}$$

where N is the number of sensor elements. Each element of $\mathbf{r}_{xd}$, denoted by r_{x_id}, is just the cross-correlation between the reference signal $d(t)$ and signal $x_i(t)$. Since the reference signal is just a replica of the desired signal and is statistically independent of the interference and thermal noise signals, the elements of $\mathbf{r}_{xd}$ consist only of the autocorrelation function of the desired signal so that

$$r_{x_id} = E\{x_i(t)d(t)\} = r_{dd}(\tau_{d_i}) \tag{10.37}$$

where τ_{d_i} represents the time delay between the reference signal and the desired signal component of $x_i(t)$. For an array with tapped delay line processing, each element of $\mathbf{r}_{xd}$ is the autocorrelation function of the desired signal evaluated at a time-delay value that reflects both the spatial delay between sensor elements and the delay line delay to the tap of interest. For an array with quadrature hybrid processing, the elements of $\mathbf{r}_{xd}$ corresponding to an in-phase channel yield the autocorrelation function of the desired signal evaluated at the spatial delay appropriate for that element as follows:

$$r_{x_id}(\text{in-phase channel}) = E\{x_i(t)d(t)\} = r_{dd}(\tau_{d_i}) \tag{10.38}$$

The elements of $\mathbf{r}_{xd}$ corresponding to quadrature-phase channels can be evaluated using (10.27) and (10.28) as follows:

$$\begin{aligned} r_{x_{i+1}d}(\text{quadrature-phase channel}) &= E\{x_{i+1}(t)d(t)\} \\ &= E\{\check{x}_i(t)d(t)\} = -E\{x_i(t)\check{d}(t)\} \qquad (10.39) \\ &= -\check{E}\{x_i(t)d(t)\} = -\check{r}_{x_id}(\tau_{d_i}) \end{aligned}$$

Once $\mathbf{R}_{xx}$ and $\mathbf{r}_{xd}$ have been evaluated for a given signal environment, the optimal LMS weights can be computed from (10.16), and the steady-state response of the entire array can then be evaluated.

The tapped delay line in the element channel of Figure 10-13 has a channel transfer function given by

$$H_1(\omega) = w_1 + w_2 e^{-j\omega\Delta} + w_3 e^{-j2\omega\Delta} + \ldots + w_L e^{-j(L-1)\omega\Delta} \tag{10.40}$$

Likewise, the quadrature hybrid processor of Figure 10-14 has a channel transfer function

$$H_1(\omega) = w_1 - jw_2 \tag{10.41}$$

The array transfer function for the desired signal and the interference accounts for the effects of spatial delays between array elements. A two-element array transfer function for the desired signal is

$$H_d(\omega) = H_1(\omega) + H_2(\omega)e^{-j\omega\tau_d} \tag{10.42}$$

whereas the transfer function for the interference is

$$H_I(\omega) = H_1(\omega) + H_2(\omega)\, e^{-j\omega\tau_I} \tag{10.43}$$

The spatial time delays associated with the desired and interference signals are represented by τ_d and τ_I, respectively, between element 1 [with channel transfer function $H_1(\omega)$] and element 2 [with channel transfer function $H_2(\omega)$]. With two sensor elements spaced apart by a distance d as in Figure 10-11, the two spatial time delays are given by

$$\tau_d = \frac{d}{\mathfrak{v}} \sin\theta_s \tag{10.44}$$

$$\tau_I = \frac{d}{\mathfrak{v}} \sin\theta_I \tag{10.45}$$

The output signal-to-total-noise ratio is defined as

$$\text{SNR} \triangleq \frac{P_d}{P_I + P_n} \tag{10.46}$$

where P_d, P_I, and P_n represent the output desired signal power, interference signal power, and thermal noise power, respectively. The array output power for each of the foregoing three signals may now be evaluated. Let $\phi_{dd}(\omega)$ and $\phi_{II}(\omega)$ represent the power spectral densities of the desired signal and the interference signal, respectively; then the desired signal output power is given by

$$P_d = \int_{-\infty}^{\infty} \phi_{dd}(\omega)|H_d(\omega)|^2 d\omega \tag{10.47}$$

where $H_d(\omega)$ is the overall transfer function seen by the desired signal, and the interference signal output power is

$$P_I = \int_{-\infty}^{\infty} \phi_{II}(\omega)|H_1(\omega)|^2 d\omega \tag{10.48}$$

where $H_I(\omega)$ is the overall transfer function seen by the interference signal. The thermal noise present in each element output is statistically independent from one element to the next. Let $\phi_{nn}(\omega)$ denote the thermal noise power spectral density; then the noise power

contributed to the array output by element 1 is

$$P_{n_1} = \int_{-\infty}^{\infty} \phi_{nn}(\omega)|H_1(\omega)|^2 d\omega \tag{10.49}$$

whereas that contributed by element 2 is

$$P_{n_2} = \int_{-\infty}^{\infty} \phi_{nn}(\omega)|H_2(\omega)|^2 d\omega \tag{10.50}$$

Consequently, the total thermal noise output power from a two-element array is

$$P_n = \int_{-\infty}^{\infty} \phi_{nn}(\omega)[|H_1(\omega)|^2 + |H_2(\omega)|^2]\, d\omega \tag{10.51}$$

The foregoing expressions may now be used in (10.46) to obtain the output signal-to-total-noise ratio.

10.3.3 Performance Comparison of Four Array Processors

In this subsection, four adaptive arrays—one with quadrature hybrid processing and three with tapped delay line processing (using real weights)—are compared for signal bandwidths of 4, 10, 20, and 40%. Tapped delay lines use real weights to preserve as much simplicity as possible in the hardware implementation, although this sacrifices the available degrees of freedom with a consequent degradation in tapped delay line performance relative to combined amplitude and phase weighting. The results obtained will nevertheless serve as an indication of the relative effectiveness of tapped delay line processing compared with quadrature hybrid processing for broadband signals.

The four array processors to be compared are shown in Figure 10-15, where each array has two sensor elements and the elements are spaced one-half wavelength apart at the center frequency of the desired signal bandwidth. Figure 10-15a shows an array having quadrature hybrid processing, whereas Figure 10-15b–10-15d exhibit tapped delay line processing. The processor of Figure 10-15b has one delay element corresponding to one-quarter wavelength at the center frequency and two associated taps. The processor of Figure 10-15c has two delay elements, each corresponding to one-quarter wavelength at the center frequency, and three associated taps. The processor of Figure 10-15d has four delay elements, each corresponding to one-eighth wavelength at the center frequency, and five associated taps. Note that the total delay present in the tapped delay line of Figure 10-15d is the same as that of Figure 10-15c, so the processor in Figure 10-15d may be regarded as a more finely subdivided version of the processor in Figure 10-15c.

Assume that the desired signal is biphase modulated of the form

$$s_d(t) = A\cos[\omega_0 t + \phi(t) + \theta] \tag{10.52}$$

where $\phi(t)$ denotes a phase angle that is either zero or π over each bit interval, and θ is an arbitrary constant phase angle (within the range $[0, 2\pi]$) for the duration of any signal pulse. The nth bit interval is defined over $T_0 + (n-1)T \le t \le T_0 + nT$, where n is any integer, T is the bit duration, and T_0 is a constant that determines where the bit transitions occur, as shown in Figure 10-16.

Assume that $\phi(t)$ is statistically independent over different bit intervals and is zero or π with equal probability and that T_0 is uniformly distributed over one bit interval;

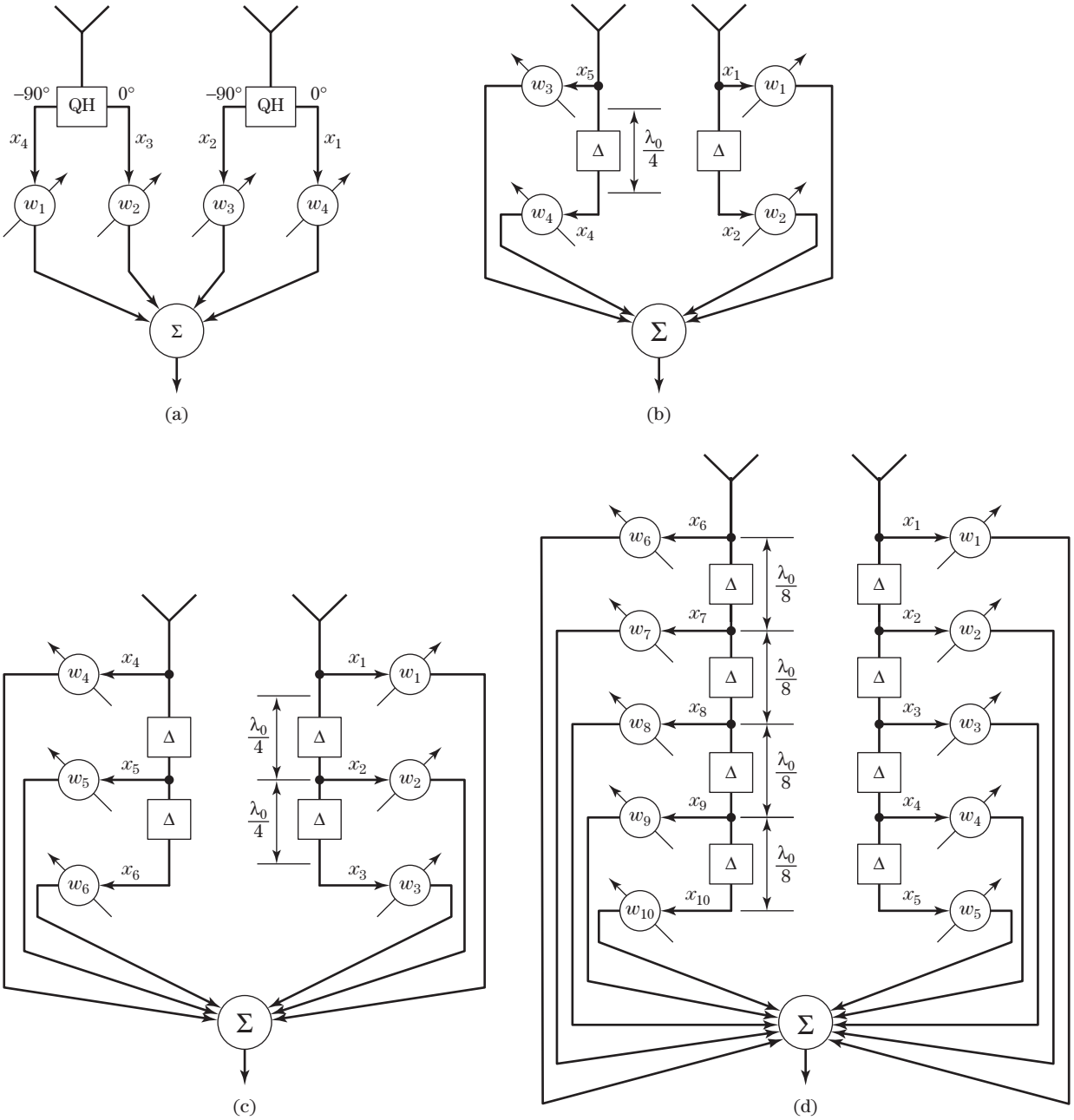


FIGURE 10-15 ■ Four adaptive array processors for broadband signal processing comparison. a: Quadrature hybrid. b: Two-tap delay line. c: Three-tap delay line. d: Five-tap delay line. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst*., January 1979 [13].

then, $s_d(t)$ is a stationary random process with power spectral density given by

$$\phi_{dd}(\omega) = \frac{A^2 T}{2}\left[\frac{\sin(T/2)(\omega-\omega_0)}{(T/2)\,(\omega-\omega_0)}\right]^2 \tag{10.53}$$

This power spectral density is shown in Figure 10-17.

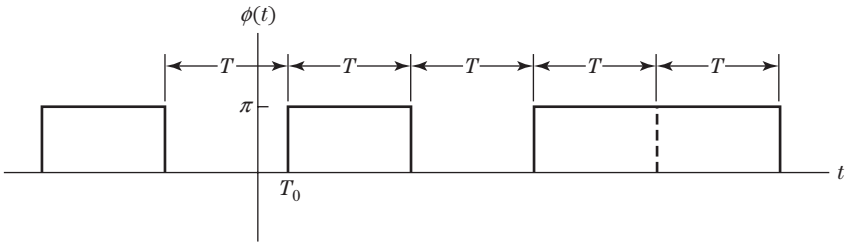


FIGURE 10-16 ■ Bit transitions for biphase modulated signal.

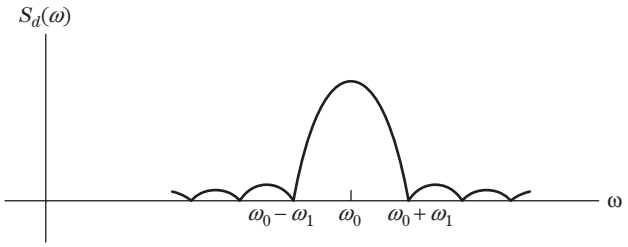


FIGURE 10-17 ■ Desired signal power spectral density.

The reference signal equals the desired signal component of $x_1(t)$ and is time aligned with the desired component of $x_2(t)$. The desired signal "bandwidth" will be taken to be the frequency range defined by the first nulls of the spectrum given by (10.53). With this definition, the fractional bandwidth then becomes

$$\text{desired signal bandwidth} = \frac{2\omega_1}{\omega_0} \tag{10.54}$$

where ω_1 is the frequency separation between the center frequency ω_0 and the first null

$$\omega_1 = \frac{2\pi}{T} \tag{10.55}$$

Assume that the interference signal is a Gaussian random process with a flat, bandlimited power spectral density over the range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$; then the interference signal spectrum appears in Figure 10-18. Finally, the thermal noise signals present at each element are statistically independent between elements, having a flat, bandlimited, Gaussian spectral density over the range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$ (identical with the interference spectrum of Figure 10-18).

With the foregoing definitions of signal spectra, the integrals of (10.48) and (10.51) yielding interference and thermal noise power are taken only over the frequency range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$. The desired signal power also is considered only over the frequency range $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$ to obtain a consistent definition of signal-to-noise ratio (SNR). Therefore, the integral of (10.47) is carried out only over $\omega_0 - \omega_1 < \omega < \omega_0 + \omega_1$.

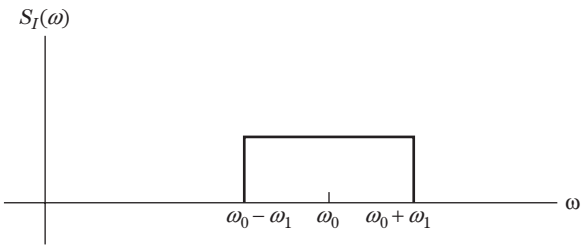


FIGURE 10-18 ■ Interference signal power spectral density.

To compare the four adaptive array processors of Figure 10-15, the output SNR performance is evaluated for the aforementioned signal conditions. Assume the element thermal noise power p_n is 10 dB below the element desired signal power p_s so that $p_s/p_n = 10$ dB. Furthermore, suppose that the element interference signal power p_i is 20 dB stronger than the element desired signal power so that $p_s/p_i = -20$ dB. Now assume that the desired signal is incident on the array from broadside. The output SNR given by (10.46) can be evaluated from (10.47), (10.48), and (10.49) by assuming the processor weights satisfy (10.16) for each of the four processor configurations. The resulting output signal-to-total noise ratio that results using each processor is plotted in Figures 10-19–10-22 as a function of the interference angle of arrival for 4, 10, 20, and 40% bandwidth signals, respectively.

In all cases, regardless of the signal bandwidth, when the interference approaches broadside (near the desired signal) the SNR degrades rapidly, and the performance of

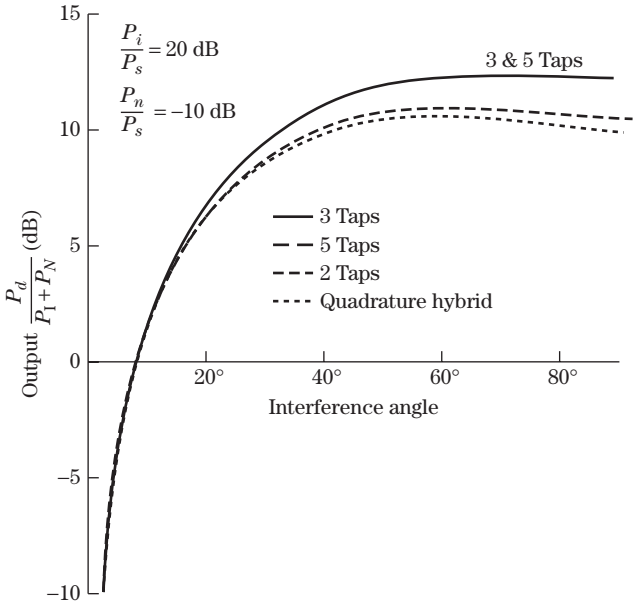


FIGURE 10-19 ■ Output signal-to-interference plus noise ratio interference angle for four adaptive processors with 4% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

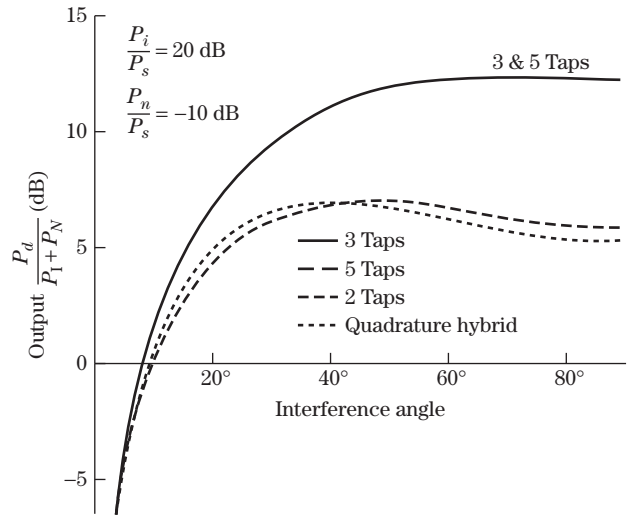


FIGURE 10-20 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 10% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

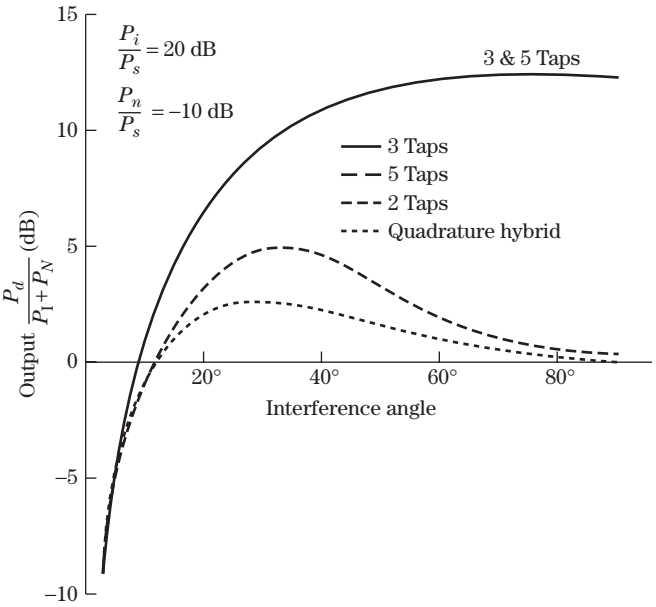


FIGURE 10-21 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 20% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

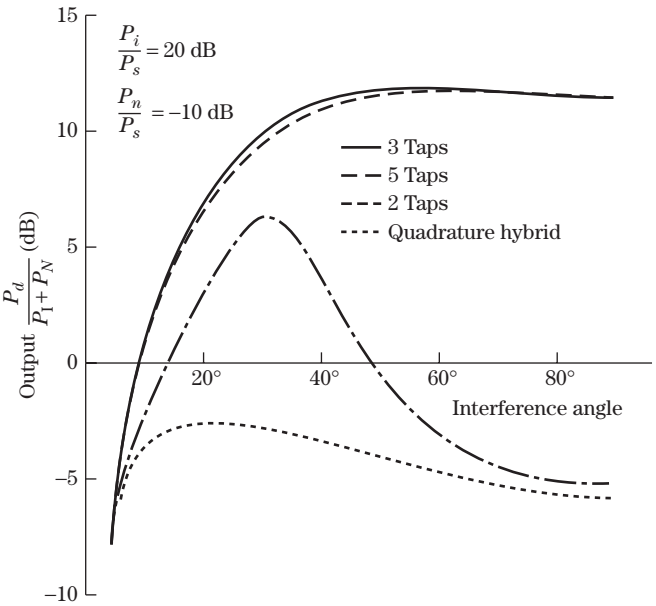


FIGURE 10-22 ■ Output signal-to-interference plus noise ratio versus interference angle for four adaptive processors with 40% bandwidth signal. From Rodgers and Compton, *IEEE Trans. Aerosp. Electron. Syst.*, January 1979 [13].

all four processors becomes identical. This SNR degradation is expected since, when the interference approaches the desired signal, the desired signal falls into the null provided to cancel the interference, and the output SNR consequently falls. Furthermore, as the interference approaches broadside, the interelement phase shift for this signal approaches zero. Consequently, the need to provide a frequency-dependent phase shift behind each array element to deal with the interference signal is less, and the performance of all four processors becomes identical.

When the interference signal is widely separated from the desired signal, then the output SNR is different for the four processors being considered, and this difference becomes more pronounced as the bandwidth increases. For 20 and 40% bandwidth signals,

for example, neither the quadrature hybrid processor nor the two-tap delay line processor provides good performance as the interference signal approaches endfire. The performance of both the three- and five-tap delay line processors remains quite good in the endfire region, however. If 20% or more bandwidth signals are accommodated, then tapped delay line processing becomes a necessity. Figure 10-22 shows that there is no significant performance advantage provided by the five-tap processor compared with the three-tap processor, so a three-tap processor is adequate for up to 40% bandwidth signals in the case of a two-element array.

Figures 10-21 and 10-22 show that the output SNR performance of the two-tap delay line processor peaks when the interference signal is 30° off broadside, because the interelement delay time is $\lambda/4$ (since the elements are spaced apart by $\lambda/2$). Consequently, the single-delay element value of $\lambda/4$ provides just the right amount of time delay to compensate exactly for the interelement time delay and to produce an improvement in the output SNR.

The three-tap and five-tap delay line processors both produce a maximum SNR of about 12.5 dB at wide interference angles of 70° or greater. For ideal channel processing, the interference signal is eliminated, the desired signal in each channel is added coherently to produce $P_d = 4p_s$, and the thermal noise is added noncoherently to yield $P_N = 2p_n$. Thus, the best possible theoretical output SNR for a two-element array with thermal noise 10 dB below the desired signal and no interference is 13 dB. Therefore, the three-tap and five-tap delay line processors are successfully rejecting nearly all the interference signal power at wide off-boresight angles.

10.3.4 Processor Transfer Functions

Ideally, the array transfer function for the desired signal should be constant across the desired signal bandwidth, thereby preventing desired signal distortion. The interference transfer function should be a low array response over the interference bandwidth.

The transfer functions for the four processors and the two-element array are evaluated using (10.40)–(10.45). Using the same conditions adopted in computing the SNR performance, Figures 10-23–10-26 show $|H_d(\omega)|$ and $|H_I(\omega)|$ for the four processors of

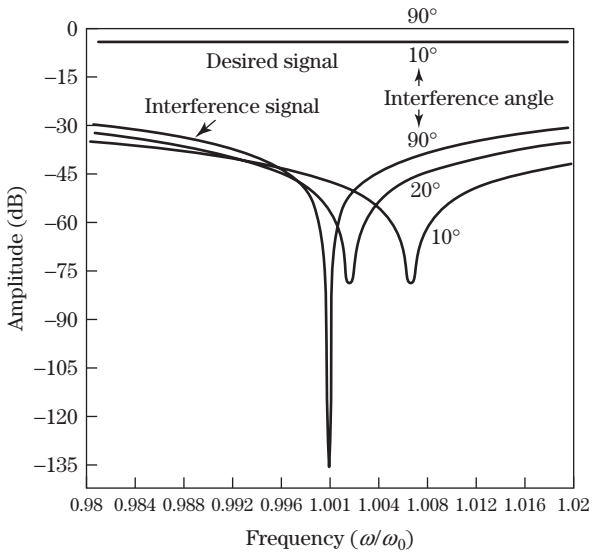


FIGURE 10-23 ■ Quadrature hybrid transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

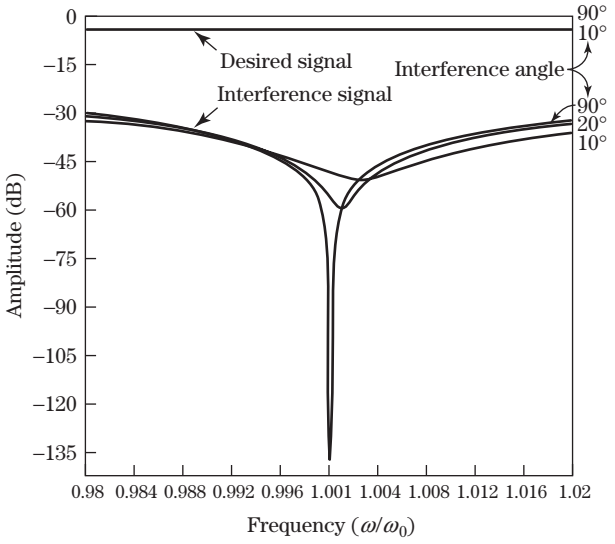


FIGURE 10-24 ■ Two-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

Figure 10-15 with a 4% signal bandwidth and various interference signal angles. The results shown in these figures indicate that for all four processors and for all interference angles the desired signal response is quite flat over the signal bandwidth. As the interference approaches the desired signal angle at broadside, however, the (constant) response level of the array to the desired signal drops because of the desired signal partially falling within the array pattern interference null.

The results in Figure 10-23 for quadrature hybrid processing show that the array response to the interference signal has a deep notch at the center frequency when the interference signal is well separated ($\theta_i > 20°$) from the desired signal. As the interference signal approaches the desired signal ($\theta_i < 20°$), the notch migrates away from the center frequency, because the processor weights must compromise between rejection of the interference signal and enhancement of the desired signal when the two signals are close. Migration of the notch improves the desired signal response (since the desired signal power spectral density peaks at the center frequency) while affecting interference rejection only slightly (since the interference signal power spectral density is constant over the signal band).

The array response for the two-tap processor is shown in Figure 10-24. The response to both the desired and interference signals is very similar to that obtained for quadrature hybrid processing. The most notable change is the slightly different shape of the transfer function notch presented to the interference signal by the two-tap delay line processor compared with the quadrature hybrid processor.

Figure 10-25 shows the three-tap processor array response. The interference signal response is considerably reduced, with a minimum rejection of the interference signal of about 45 dB. When the interference signal is close to the desired signal, the array response has a single mild dip. As the separation angle between the interference signal and the desired signal increases, the single dip becomes more pronounced and finally develops into a double dip at very wide angles. It is difficult to attribute much significance to the double-dip behavior since it occurs at such a low response level (of more than 75 dB attenuation). The five-tap processor response of Figure 10-26 is very similar to the three-tap processor response except slightly more interference signal rejection is achieved.

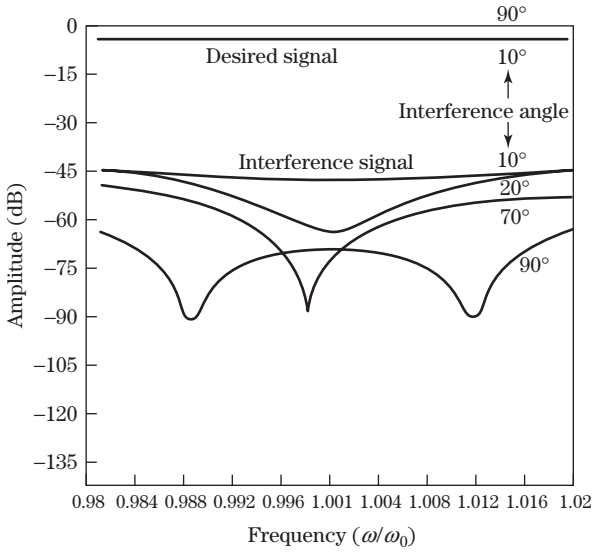


FIGURE 10-25 ■ Three-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

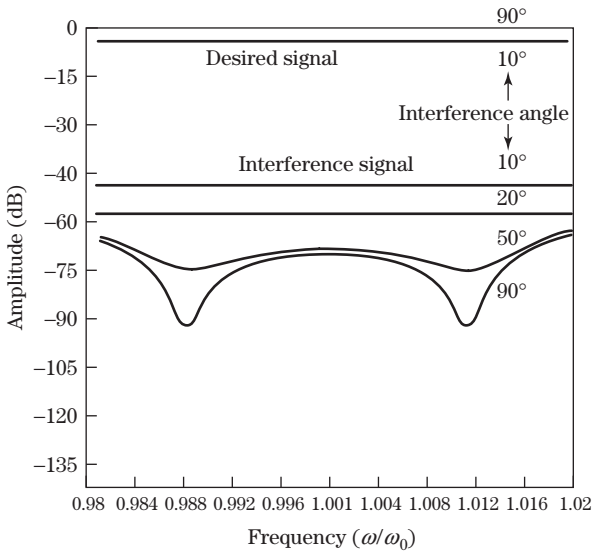


FIGURE 10-26 ■ Five-tap delay line transfer functions at 4% bandwidth. From Rodgers and Compton, Technical Report ESL 3832-3, 1975 [12].

As the signal bandwidth increases, the processor response curves remain essentially the same as in Figures 10-23–10-26 except the following:

1. As the interference signal bandwidth increases, it becomes more difficult to reject the interference signal over the entire bandwidth, so the minimum rejection level increases.
2. The desired signal response decreases because the array feedback reduces all weights to compensate for the presence of a greater interference signal component at the array output, thereby resulting in greater desired signal attenuation.

The net result is that as the signal bandwidth increases, the output SNR performance degrades, as confirmed by the results of Figures 10-19–10-22.

10.4 COMPENSATION FOR MUTUAL COUPLING

In many applications, the limited space available for mounting an antenna motivates the use of a small array. As the array size decreases, the array element spacing becomes less than a half-wavelength, and mutual coupling effects become more of a factor in degrading the array performance. When an array consists of single-mode elements (meaning that the element aperture currents may change in amplitude but not in shape as a function of the signal angle of arrival), then it is possible to modify the element weights to compensate for the pattern distortion caused by the mutual coupling at a particular angle [16]. These weight adjustments may work for more than one angle.

Let the vector $\underline{v}$ denote the coupling perturbed measured voltages appearing at the output of the array elements, and let $\underline{v}^{d}$ represent the coupling unperturbed voltages that would appear at the array element outputs if no mutual coupling were present. The effect of mutual coupling on single-mode elements is written as

$$\mathbf{v}(u) = \mathbf{C}\,\mathbf{v}^{d}(u) \tag{10.56}$$

where $u = \sin\theta$, θ is the angle of arrival, and the matrix $\mathbf{C}$ describes the effects of mutual coupling and is independent of the signal scan angle. If the array is composed of multimode elements, then the matrix $\mathbf{C}$ would be scan angle dependent.

It follows that the unperturbed signal vector, $\mathbf{v}^{d}$ can be recovered from the perturbed signal vector by introducing compensation for the mutual coupling

$$\mathbf{v}^{d} = \mathbf{C}^{-1}\mathbf{v} \tag{10.57}$$

Introducing the compensation network $\mathbf{C}^{-1}$ as shown in Figure 10-27 then allows all subsequent beamforming operations to be performed with ideal (unperturbed) element signals, as are customarily assumed in pattern synthesis.

This mutual coupling compensation is applied to an eight-element linear array having element spacing $d = 0.517\ \lambda$ consisting of identical elements. Figure 10-28(a) shows the effects of mutual coupling by displaying the difference in element pattern shape between a central and an edge element in the array.

Figure 10-28 displays a synthesized 30 dB Chebyshev pattern both without (a) and with (b) mutual coupling compensation. It is apparent from this result that the compensation network gives about a 10 dB improvement in the sidelobe level.

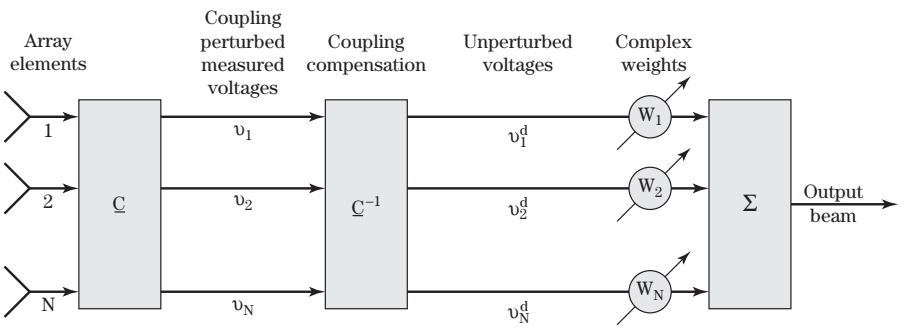


FIGURE 10-27 ■ Coupling Compensation and Beamforming in an Array Antenna. From Steyskal & Herd, IEEE Trans. Ant & Prop., Dec. 1995.

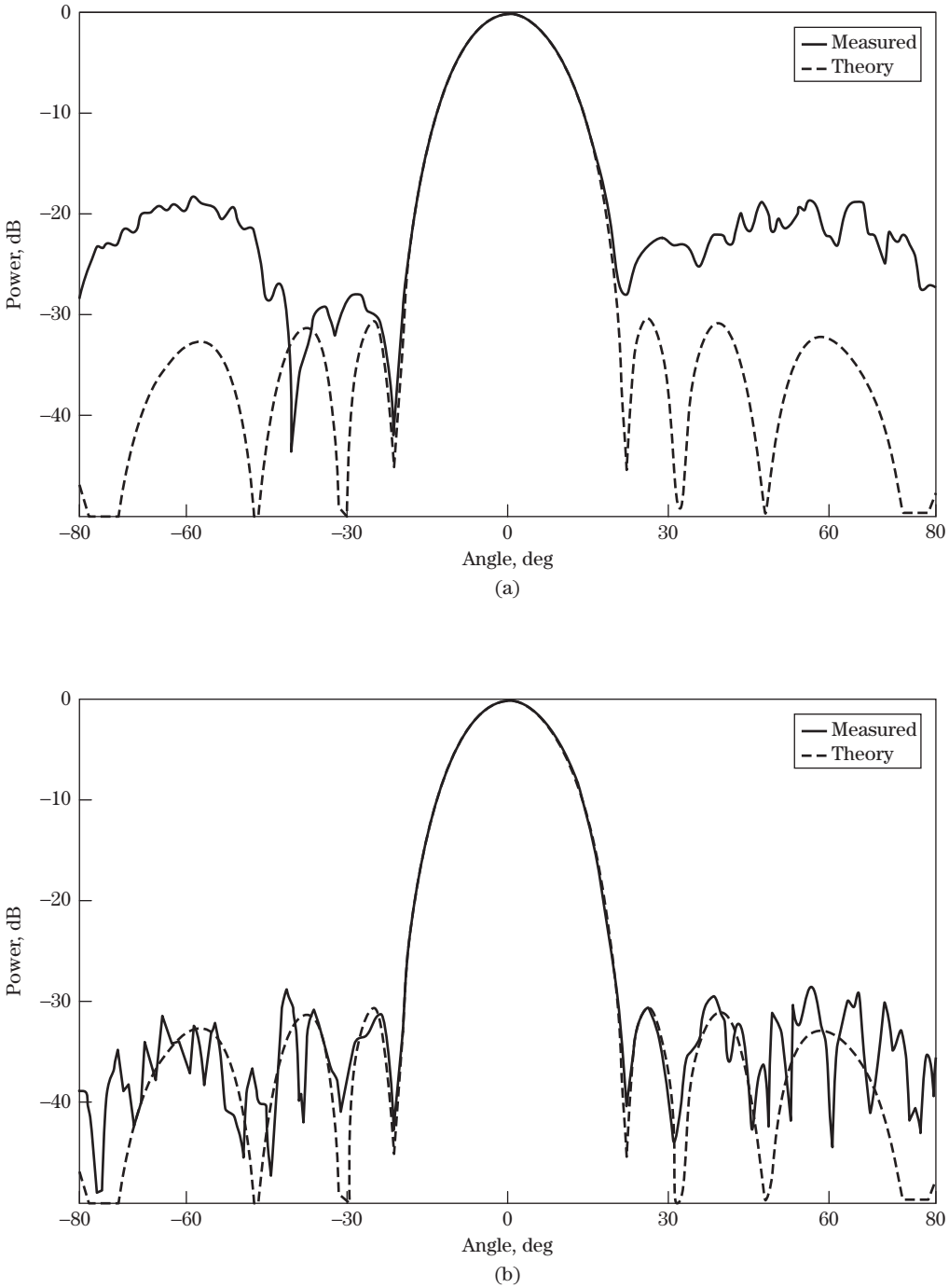


FIGURE 10-28 ■ 30 dB Chebyshev pattern (a) without and (b) with Coupling Compensation with a Scan Angle of 0°. From Steyskal & Herd, IEEE Trans. Ant. & Prop. Dec. 1995.

10.5 MULTIPATH COMPENSATION

In many operating environments, multipath rays impinge on the array shortly after the direct path signal arrives at the sensors. Multipath distorts any interference signal that may appear in the various element channels, thereby severely limiting the interference cancellation. A tapped delay line processor combines delayed and weighted replicas of the input signal to form the filtered output signal and thereby has the potential to compensate for multipath effects, since multipath rays also consist of delayed and weighted replicas of the direct path ray.

10.5.1 Two-Channel Interference Cancellation Model

Consider an ideal two-element adaptive array with one channel's (called the "auxiliary" channel) response adjusted so that any jamming signal entering the other channel through the sidelobes (termed the "main" channel) is canceled at the array output. A system designed to suppress sidelobe jamming in this manner is called a coherent sidelobe canceller (CSLC), and Figure 10-29 depicts a two-channel CSLC system in which the auxiliary channel employs tapped delay line compensation involving L weights and $L-1$ delay elements of value Δ seconds each. A delay element of value $D=(L-1)\Delta/2$ is included in the main channel so the center tap of the auxiliary channel corresponds to the output of the delay D in the main channel, thereby permitting compensation for both positive and negative values of the off-broadside angle θ. This ideal two-element CSLC system model exhibits all the salient characteristics that a more complex system involving several auxiliary channels would have, so the two-element system serves as a convenient model for performance evaluation of multipath cancellation [17].

The system performance measure is the ability of the CSLC to cancel an undesired interference signal through proper design of the tapped delay line. In actual practice, an adaptive algorithm adjusts the weight settings. To eliminate the effect of algorithm selection from consideration, only the steady-state performance is evaluated. Since the steady-state solution can be found analytically, it is necessary to determine only the resulting solution for the output residue power. This residue power is then a direct measure of the interference cancellation ability of the two-element CSLC model.

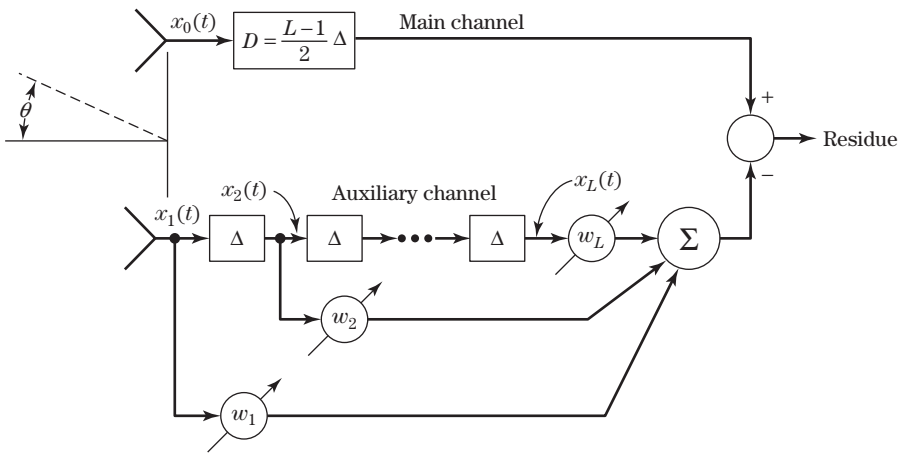


FIGURE 10-29 ■ Ideal two-element CSLC model with auxiliary channel compensation involving L weights and $L-1$ delay elements.

Let $x_0(t)$, $x_1(t)$, and $e(t)$ represent the complex envelope signals of the main channel input signal, the auxiliary channel input signal, and the output residue signal, respectively. Define the complex signal vector

$$\mathbf{x}^T \triangleq [x_1(t), x_2(t), \ldots, x_L(t)] \tag{10.58}$$

where

$$\begin{aligned} x_2(t) &\triangleq x_1(t - \Delta) \\ &\vdots \\ x_L(t) &\triangleq x_1\,[t - (L\ -\ 1)\Delta] \end{aligned}$$

Also, define the complex weight vector

$$\mathbf{w}^T \triangleq [w_1, w_2, \ldots, w_L] \tag{10.59}$$

The output of the tapped delay line may then be expressed as

$$\text{filter output} = \sum_{i=1}^{L} x_1[t - (i-1)\Delta]w_i^* = \mathbf{w}^\dagger \mathbf{x}(t) \tag{10.60}$$

The residue (complex envelope) signal is given by

$$e(t) = x_0(t - D) + \mathbf{w}^\dagger \mathbf{x}(t) \tag{10.61}$$

The weight vector $\mathbf{w}$ minimizes the residue signal in a mean square error (MSE) sense. For stationary random processes, this is equivalent to minimizing the expression

$$R_{ee}(0) = E\ \{e(t)e^*(t)\} \tag{10.62}$$

From (10.61) and the fact that

$$E\{x_0(t - D)x_0^*(t - D)\} = r_{x_0x_0}(0) \tag{10.63}$$

$$E\{\mathbf{x}(t)x_0^*(t - D)\} = r_{xx_0}(-D) \tag{10.64}$$

$$E\{\mathbf{x}(t)\mathbf{x}^\dagger(t)\} = \mathbf{R}_{xx}(0) \tag{10.65}$$

it follows that

$$\begin{aligned} R_{ee}(0) = r_{x_0x_0}(0) &- \mathbf{r}_{xx_0}^\dagger(-D)\mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \\ &+ [\mathbf{r}_{xx_0}^\dagger(-D) + \mathbf{w}^\dagger\mathbf{R}_{xx}(0)] \bullet \mathbf{R}_{xx}^{-1}(0)\,[\mathbf{r}_{xx_0}(-D) + \mathbf{R}_{xx}(0)\mathbf{w}] \end{aligned} \tag{10.66}$$

Minimize (10.66) by appropriately selecting the complex weight vector $\mathbf{w}$. Assume the matrix $\mathbf{R}_{xx}(0)$ is nonsingular: the value of $\mathbf{w}$ for which this minimum occurs is given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \tag{10.67}$$

The corresponding minimum residue signal power then becomes

$$R_{ee}(0)_{\min} = r_{x_0x_0}(0) - \mathbf{r}_{xx_0}^\dagger(-D)\mathbf{R}_{xx}^{-1}(0)\mathbf{r}_{xx_0}(-D) \tag{10.68}$$

Interference cancellation performance of the CSLC model of Figure 10-27 is determined by evaluating (10.66) using selected signal environment assumptions.

10.5.2 Signal Environment Assumptions

Let $s_1(t, \theta_1)$ represent the interference signal arriving from direction θ_1, and let $s_m(t, \rho_m, D_m, \theta_{m+1})$ for $m = 2, \ldots, M$ represent the multipath structure associated with the interference signal that consists of a collection of $M - 1$ correlated plane wave signals of the same frequency arriving from different directions so that $\theta_{m+k} \neq \theta_1$ and $\theta_{m+k} \neq \theta_{m+l}$ for $k \neq l$. The multipath rays each have an associated reflection coefficient ρ_m and a time delay with respect to the direct ray D_m. The structure of the covariance matrix for this multipath model can then be expressed as [18]

$$\mathbf{R}_{ss} = \mathbf{V}_s \mathbf{A} \mathbf{V}_s^{\dagger} \tag{10.69}$$

where $\mathbf{V}_s$ is the $N \times M$ signal matrix given by

$$\mathbf{V}_s = \left[\mathbf{v}_{s_1} \,\middle|\, \mathbf{v}_{s_2} \,\middle|\, \cdots \,\middle|\, \mathbf{v}_{s_M} \right] \tag{10.70}$$

whose components are given by the $N \times 1$ vectors

$$\mathbf{v}_{s_m} = \sqrt{P_{s_m}} \begin{bmatrix} 1 \\ \exp\left[j2\pi (d/\lambda_0) \sin\theta_m \right] \\ \exp\left[j2\pi (d/\lambda_0) 2 \sin\theta_m \right] \\ \vdots \\ \exp\left[j2\pi (d/\lambda_0) (N-1) \sin\theta_m \right] \end{bmatrix} \tag{10.71}$$

where $P_{s_m} = \rho_m^2$ denotes the power associated with the signal s_m, and $\mathbf{A}$ is the multipath correlation matrix. When $\mathbf{A} = \mathbf{I}$, the various signal components are uncorrelated whereas for $\mathbf{A} = \mathbf{U}$ (the $M \times M$ matrix of unity elements) the various components are perfectly correlated. For purposes of numerical evaluation the correlation matrix model may be selected as [18]

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{M-1} \\ \alpha & 1 & \alpha & \cdots & \alpha^{M-1} \\ \vdots & & & & \\ \alpha^{M-1} & . & . & \cdots & 1 \end{bmatrix} \quad 0 \leq \alpha \leq 1 \tag{10.72}$$

Note that channel-to-channel variations in θ_m, D_m, and ρ_m cannot be accommodated by this simplified model. Consequently, a more general model must be developed to handle such variations, which tend to occur where near-field scattering effects are significant. The input signal covariance matrix may be written as

$$\mathbf{R}_{xx} = \mathbf{R}_{nn} + \mathbf{V}_s \mathbf{A} \mathbf{V}_s^{\dagger} \tag{10.73}$$

where $\mathbf{R}_{nn}$ denotes the noise covariance matrix.

If only a single multipath ray is present, then $s(t, \theta_1)$ denotes the direct interference signal, and $s_m(t, \rho_m, D_m, \theta_2)$ represents the multipath ray associated with the direct interference signal. The received signal at the main channel element is then given by

$$x_0(t) = s(t, \theta_1) + s_m(t, \rho_m, D_m, \theta_2) \tag{10.74}$$

Denote $s(t, \theta_1)$ by $s(t)$; then $s_m(t, \rho_m, D_m, \theta_2)$ can be written as $\rho_m s(t - D_m) \times \exp(-j\omega_0 D_m)$ so that

$$x_0(t) = s(t) + \rho_m s(t - D_m) \exp(-j\omega_0 D_m) \tag{10.75}$$

where ω_0 is the center frequency of the interference signal. It then follows that

$$\begin{aligned} x_1(t) &= s(t-\tau_{12})\,\exp(-j\omega_0\tau_{12}) \\ &\quad + \rho_m s(t-D_m-\tau_{22})\,\exp\left[-j\omega_0(D_m+\tau_{22})\right] \end{aligned} \tag{10.76}$$

where τ_{12} and τ_{22} represent the propagation delay between the main channel element and the auxiliary channel element for the wavefronts of $s(t, \theta_1)$ and $s_m(t, \rho_m, d_m, \theta_2)$, respectively.

Assuming the signals $s(t, \theta_1)$ and $s_m(t, \rho_m, D_m, \theta_2)$ possess flat spectral density functions over the bandwidth B, as shown in Figure 10-30*a*, then the corresponding auto- and cross-correlation functions of $x_0(t)$ and $x_1(t)$ can be evaluated by recognizing that

$$\mathbf{R}_{xx}(\tau) = \mathfrak{F}^{-1}\{\mathbf{\Phi}_{xx}(\omega)\} \tag{10.77}$$

where $\mathfrak{F}^{-1}\{\cdot\}$ is the "inverse Fourier transform," and $\mathbf{\Phi}_{xx}(\omega)$ denotes the cross-spectral density matrix of $\mathbf{x}(t)$.

From (10.74), (10.76), and (10.77) it immediately follows that

$$r_{x_0x_0}(0) = 1 + |\rho_m|^2 + \frac{\sin\pi BD_m}{\pi BD_m}\left(\rho_m e^{-j\omega_0 D_m} + \rho_m^* e^{j\omega_0 D_m}\right) \tag{10.78}$$

Likewise, defining $f[\psi, \text{sgn1}, \text{sgn2}] \triangleq \dfrac{\sin\pi B\,[\psi + \text{sgn1}\cdot(i-1)\Delta + \text{sgn2}\cdot D]}{\pi B\,[\psi + \text{sgn1}\cdot(i-1)\Delta + \text{sgn2}\cdot D]}$

and $\quad g[\psi, \text{sgn}] \triangleq \dfrac{\sin\pi B\,[\psi + \text{sgn}\cdot(i-k)\Delta]}{\pi B[\psi + \text{sgn}\cdot(i-k)\,\Delta]}$, $\quad$ then

$$\begin{aligned} r_{x_ix_0}(-D) &= f[\tau_{12}, +, -]\exp\{-j\omega_0[\tau_{12} + (i-1)\Delta]\} \\ &\quad + f[D_m+\tau_{22}, +, -]\rho_m \exp\{-j\omega_0[\tau_{22} + (i-1)\Delta + D_m]\} \\ &\quad + f[D_m-\tau_{12}, -, +]\rho_m^* \exp\{-j\omega_0[\tau_{12} + (i-1)\Delta - D_m]\} \\ &\quad + f[\tau_{22}, +, -]|\rho_m|^2 \exp\{-j\omega_0[\tau_{22} + (i-1)\Delta]\} \end{aligned} \tag{10.79}$$

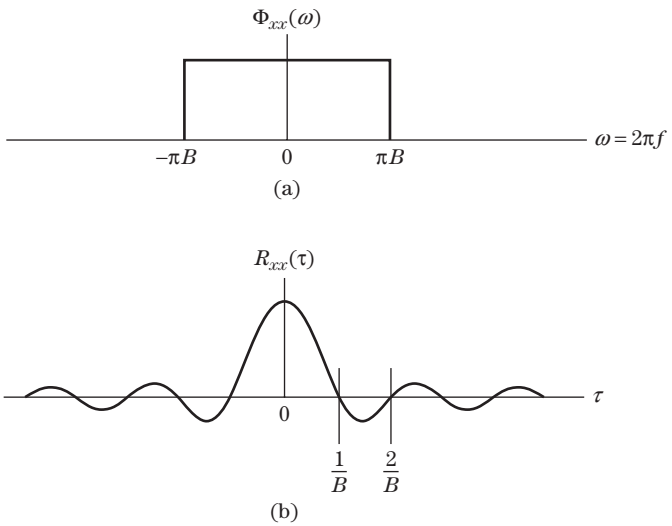


FIGURE 10-30 ■ Flat spectral density function and corresponding autocorrelation function for interference signal. a: Spectral density function. b: Autocorrelation function.

$$
\begin{aligned}
r_{x_i x_k}(0) = {} & g[0,+][1+|\rho_m|^2]\exp[-j\omega_0(i-k)\Delta] \\
& + g[\tau_{12}-\tau_{22}-D_m,-]\rho_m \exp\{j\omega_0[\tau_{12}-\tau_{22}-D_m-(i-k)\Delta]\} \\
& + g[\tau_{12}-\tau_{22}-D_m,+]\rho_m^* \exp\{-j\omega_0[\tau_{12}-\tau_{22}-D_m+(i-k)\Delta]\}
\end{aligned} \tag{10.80}
$$

The vector $\mathbf{r}_{xx_0}(-D)$ is then given by

$$
\mathbf{r}_{xx_0}(-D) = \begin{bmatrix} r_{x_1x_0}(-D) \\ r_{x_2x_0}(-D) \\ \vdots \\ r_{x_Nx_0}(-D) \end{bmatrix} \tag{10.81}
$$

and the matrix $\mathbf{R}_{xx}(0)$ is given by

$$
\mathbf{R}_{xx}(0) = \begin{bmatrix} r_{x_1x_1}(0) & r_{x_1x_2}(0) & \cdots & r_{x_1x_N}(0) \\ \vdots & r_{x_2x_2}(0) & & \\ & & \ddots & \\ r_{x_1x_N}(0) & \cdots & & r_{x_Nx_N}(0) \end{bmatrix} \tag{10.82}
$$

To evaluate (10.68) for the minimum possible value of output residue power (10.78), (10.79), and (10.80), show that it is necessary to specify the following parameters:

N = number of taps in the transversal filter
ρ_m = multipath reflection coefficient
ω_0 = (radian) center frequency of interference signal
D_m = multipath delay time with respect to direct ray
τ_{12} = propagation delay between the main antenna element and the auxiliary antenna element for the direct ray
τ_{22} = propagation delay between the main antenna element and the auxiliary antenna element for the multipath ray
Δ = transversal filter intertap delay
B = interference signal bandwidth
D = main channel receiver time delay

The quantities τ_{12} and τ_{22} are related to the CSLC array geometry by

$$
\left.\begin{aligned} \tau_{12} &= \frac{d}{\mathfrak{v}}\sin\theta_1 \\ \tau_{22} &= \frac{d}{\mathfrak{v}}\sin\theta_2 \end{aligned}\right\} \tag{10.83}
$$

where

d = interelement array spacing
$\mathfrak{v}$ = wavefront propagation speed
θ_1 = angle of incidence of direct ray
θ_2 = angle of incidence of multipath ray

10.5.3 Example: Results for Compensation of Multipath Effects

An interference signal has a direct ray angle of arrival is $\theta_1 = 30^\circ$, the multipath ray angle of arrival is $\theta_2 = -30^\circ$, and the interelement spacing is $d = 2.25\lambda_0$. Some additional

signal and multipath characteristics are

$$\begin{aligned} \text{center frequency} \quad & f_0 = 237 \text{ MHz} \\ \text{signal bandwidth} \quad & B = 3 \text{ MHz} \\ \text{multipath reflection coefficient} \quad & \rho_m = 0.5 \end{aligned} \tag{10.84}$$

Referring to (10.76), (10.79), and (10.80), we see that the parameters ω_0, τ_{12}, τ_{22}, D_m, and Δ enter the evaluation of the output residue power in the form of the products $\omega_0\tau_{12}$, $\omega_0\tau_{22}$, $\omega_0 D_m$, and $\omega_0\Delta$. These products represent the phase shift experienced at the center frequency ω_0 as a consequence of the four corresponding time delays. Likewise, the parameters B, D, D_m, τ_{12}, τ_{22}, and Δ enter the evaluation of the output residue power in the form of the products BD, BD_m, $B\tau_{12}$, $B\tau_{22}$, and $B\Delta$; these time–bandwidth products are phase shifts experienced by the highest frequency component of the complex envelope interference signal as a consequence of the five corresponding time delays. Both the intertap delay Δ and the multipath delay D_m are important parameters that affect the CSLC system performance through their corresponding time–bandwidth products; thus, the results are given here with the time–bandwidth products taken as the fundamental quantity of interest.

Since for this example $\theta_1 = -\theta_2$, the product $\omega_0\tau_{12}$ is specified as

$$\left.\begin{aligned} & \omega_0\tau_{12} = \frac{\pi}{4} \\ \text{then the product} \quad & \\ & \omega_0\tau_{22} = -\frac{\pi}{4} \end{aligned}\right\} \tag{10.85}$$

Furthermore, let the products $\omega_0 D_m$ and $\omega_0\Delta$ be given by

$$\left.\begin{aligned} \omega_0 D_m &= 0 \pm 2k\pi, \qquad k \text{ any integer} \\ \omega_0\Delta &= 0 \pm 2l\pi, \qquad l \text{ any integer} \end{aligned}\right\} \tag{10.86}$$

For the element spacing $d = 2.25\lambda_0$ and $\theta_1 = 30^\circ$, then specify

$$B\tau_{12} = -B\tau_{22} = \frac{1}{P}, \qquad P = 72 \tag{10.87}$$

Finally, specifying the multipath delay time to correspond to 46 meters yields

$$BD_m = 0.45 \tag{10.88}$$

Since

$$D = \frac{N-1}{2}\Delta \tag{10.89}$$

Only N and $B\Delta$ need to be specified to evaluate the output residue power by way of (10.68).

To evaluate the output residue power by way of (10.68) resulting from the array geometry and multipath conditions specified by (10.84)–(10.89) requires that the cross-correlation vector $\mathbf{r}_{xx_0}(-D)$, the $N \times N$ autocorrelation matrix $\mathbf{R}_{xx}(0)$, and the autocorrelation function $r_{x_0x_0}(0)$ be evaluated by way of (10.78)–(10.80). A computer program to evaluate (10.68) for the multipath conditions specified was written in complex, double-precision arithmetic.

Figure 10-31 shows a plot of the output residue power where the resulting minimum possible value of canceled power output in dB is plotted as a function of $B\Delta$ for various

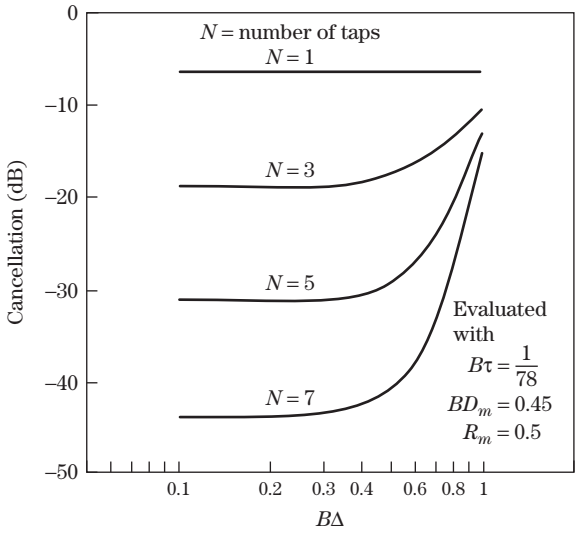


FIGURE 10-31 ■
Decibel cancellation versus $B\Delta$ for multipath.

specified values of N. It will be noted in Figure 10-31 that for $N = 1$ the cancellation performance is independent of $B\Delta$ since no intertap delays are present with only a single tap. As explained in Appendix B, the transfer function of the tapped delay line transversal filter has a periodic structure with (radian) frequency period $2\pi B_f$, which is centered at the frequency f_0. It should be noted that the transversal filter frequency bandwidth B_f is not necessarily the same as the signal-frequency bandwidth B. The transfer function of a transversal filter within the primary frequency band ($|f - f_0| < B_f/2$) may be expressed as

$$F(f) = \sum_{k=1}^{N} [A_k e^{j\phi_k}] \exp[-j2\pi(k-1)\delta f \Delta] \tag{10.90}$$

where $A_k e^{j\phi_k}$ represents the kth complex weight, $\delta f = f - f_0$, $f_0 =$ center frequency, and the transversal filter frequency bandwidth is

$$B_f = \frac{1}{\Delta} \tag{10.91}$$

Since the transversal filter should be capable of adjusting the complex weights to achieve appropriate amplitude and phase values over the entire signal bandwidth B, it follows that B_f should satisfy

$$B_f \geq B \tag{10.92}$$

Consequently, the maximum intertap delay spacing is given by

$$\Delta_{\max} = \frac{1}{B} \tag{10.93}$$

It follows that values of $B\Delta$ that are greater than unity should not be considered for practical compensation designs; however, values of $B_f > B$ (resulting in $0 < B\Delta < 1$) are sometimes desirable.

Figure 10-31 shows that, as $B\Delta$ decreases from 1, for values of $N > 1$ the cancellation performance rapidly improves (the minimum canceled residue power decreases) until $B\Delta = BD_m$ (0.45 for this example), after which very little significant improvement occurs. As $B\Delta$ becomes very much smaller than BD_m (approaching zero), the cancellation performance degrades since the intertap delay is effectively removed. The simulation could not compute this result since as $B\Delta$ approaches zero the matrix $\mathbf{R}_{xx}(0)$ becomes singular and matrix inversion becomes impossible. Cancellation performance of -30 dB is virtually assured if the transversal filter has at least five taps and Δ is selected so that $\Delta = D_m$.

Suppose for example that the transversal filter is designed with $B\Delta = 0.45$. Using the same set of selected constants as for the previous example, we find it useful to consider what results would be obtained when the actual multipath delay is different from the anticipated value corresponding to $BD_m = 0.45$. From the results already obtained in Figure 10-31, it may be anticipated that, if $BD_m > B\Delta$, then the cancellation performance would degrade. If, however, $BD_m \ll B\Delta$, then the cancellation performance would improve since in the limit as $D_m \to 0$ the system performance with no multipath present would result.

10.5.4 Results for Compensation of Array Propagation Delay

In the absence of a multipath ray, the analysis presented in the preceding section includes all the features necessary to account for array propagation delay effects. When we set $\rho_m = 0$ and let $\tau_{12} = \tau$ represent the element-to-element array propagation delay, (10.78)–(10.80) permit (10.68) to be used to investigate the effects of array propagation delay on cancellation performance. On the basis of the behavior already found for multipath compensation, it would be reasonable to anticipate that with $B\Delta = B\tau$ then maximum cancellation performance would obtain, whereas if $B\Delta > B\tau$ then the cancellation performance would degrade. Figure 10-32 gives the resulting cancellation performance as a function of $B\Delta$ for fixed $B\tau$. The number of taps N is an independent parameter, and all other system constants are the same as those in the example of Section 10.4.3. It is seen that the results confirm the anticipated performance noted already.

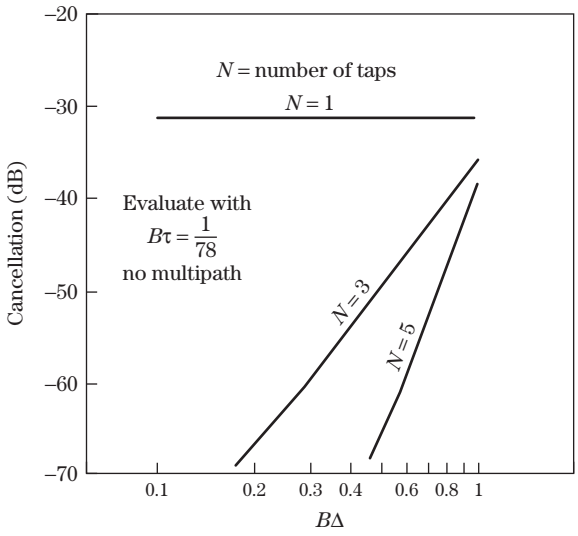


FIGURE 10-32 ■ Decibel cancellation versus $B\Delta$ for array propagation delay.

10.6 | ANALYSIS OF INTERCHANNEL MISMATCH EFFECTS

Any adaptive array processor is susceptible to unavoidable frequency-dependent variations in gain and phase between the various element channels. Additional degrees of freedom provided by a tapped delay line compensate for such frequency-dependent "channel mismatch" effects. Since a simple two-element CSLC system exhibits all the salient characteristics of channel mismatching present in more complex systems, the two-element model is again adopted as the example for performance evaluation of channel mismatch compensation.

Figure 10-33 is a simplified representation of a single auxiliary channel CSLC system in which the single complex weight is a function of frequency. The transfer function $T_0(\omega, \theta)$ reflects all amplitude and phase variations in the main beam sidelobes as a function of frequency as well as any tracking errors in amplitude and phase between the main and auxiliary channel electronics. Likewise, the equivalent transfer function for the auxiliary channel (including any auxiliary antenna variations) is denoted by $T_1(\omega, \theta)$. The spectral power density of a wideband jammer is given by $\phi_{JJ}(\omega)$. The signal from the auxiliary channel is "multiplied" by the complex weight $w_1 = \alpha e^{j\phi}$, and the "cancelled" output of residue power spectral density is represented by $\phi_{rr}(\omega, \theta)$.

The objective of the CSLC is to minimize the residue power, appropriately weighted, over the bandwidth. Since the integral of the power spectral density over the signal frequency spectrum yields the signal power, the requirement to minimize the residue power is expressed as

$$\underset{w_1}{\text{Min}} \int_{-\infty}^{\infty} \phi_{rr}(\omega, \theta) d\omega \tag{10.94}$$

where

$$\phi_{rr}(\omega, \theta) = |T_0(\omega, \theta) - w_1 T_1(\omega, \theta)|^2 \phi_{JJ}(\omega) \tag{10.95}$$

Now replace the complex weight w_1 in Figure 10-33 by a tapped delay line having $2N + 1$ adaptively controlled complex weights each separated by a time delay Δ as in Figure 10-34. A delay element of value $N\Delta$ is included in the main channel (just as in the preceding section) so that compensation for both positive and negative angles of arrival is provided. The main and auxiliary channel transfer functions are written in terms of the output of the main channel, so no delay terms occur in the resulting main channel transfer

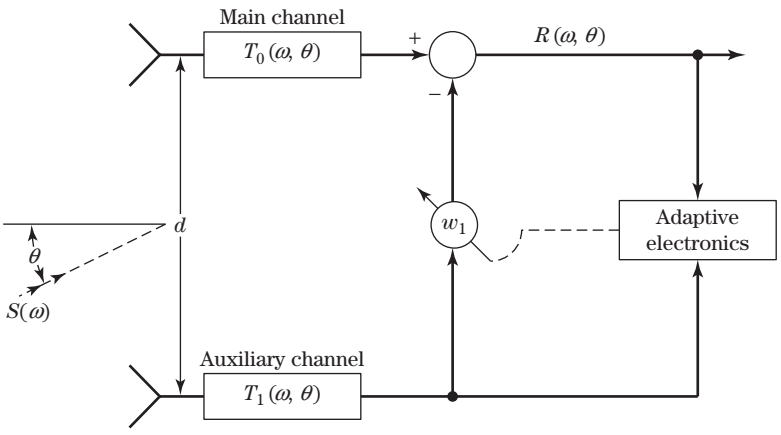


FIGURE 10-33 ■ Simplified model of single-channel CSLC.

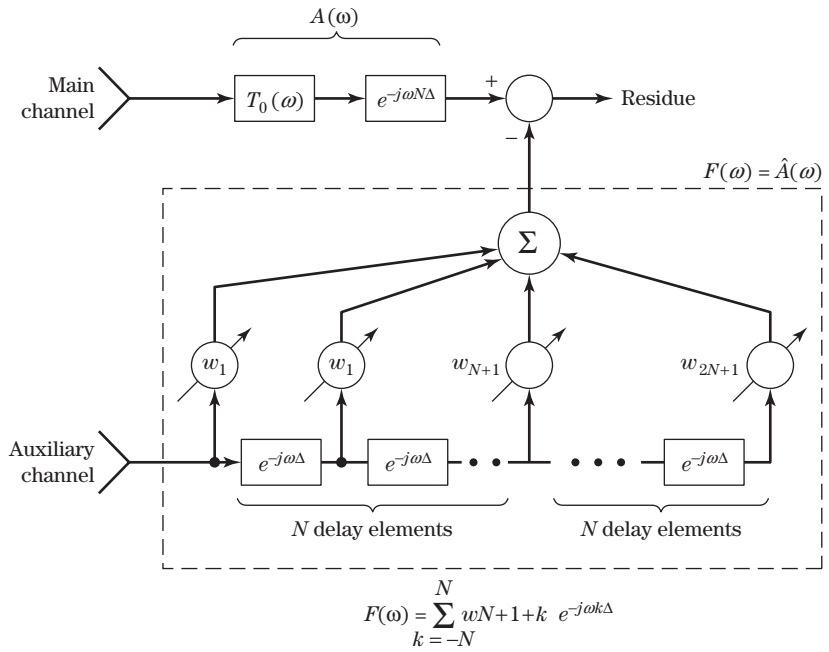


FIGURE 10-34 ■ Single-channel CSLC having main channel distortion and tapped delay line auxiliary channel compensation.

function, $A(\omega)$. Assume for analysis purposes that all channel distortion is confined to the main channel and that $T_1(\omega, \theta) = 1$. The transversal filter transfer function, $F(\omega)$, can be expressed as

$$F(\omega) = \sum_{k=-N}^{N} w_{N+1+k} e^{-j\omega k\Delta} \tag{10.96}$$

where the w_{N+1+k}'s are nonfrequency-dependent complex weights.

We want to minimize the output residue power over the signal bandwidth by appropriately selecting the weight vector **w**. Assuming the jammer power spectral density is constant over the frequency region of interest, then minimizing the output residue power is equivalent to selecting the $F(\omega)$ that provides the "best" estimate (denoted by $\hat{A}(\omega)$) of the main channel transfer function over that frequency range. If the estimate $\hat{A}(\omega)$ is to be optimal in the MSE sense, then the error in this estimate $e(\omega) = A(\omega) - F(\omega)$ must be orthogonal to $\hat{A}\,(\omega) = F(\omega)$, that is,

$$E\{[A(\omega) - F(\omega)]F^*(\omega)\} = 0 \tag{10.97}$$

where the expectation $E\{\cdot\}$ is taken over frequency and is therefore equivalent to

$$E\{\cdot\} = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\{\ \}\,d\omega \tag{10.98}$$

where all frequency-dependent elements in the integrand of (10.98) are reduced to baseband. Letting $A(\omega) = A_0(\omega)e^{-j\phi_0(\omega)}$, substituting (10.96) into (10.97), and requiring the error to be orthogonal to all tap outputs to obtain the minimum MSE estimate $\hat{A}(\omega)$ then yields the condition

$$E\{[A_0(\omega)\exp[-j\phi_0(\omega)] - F(\omega)]\exp(j\omega k\,\Delta)\} = 0 \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.99}$$

Equation (10.99) can be rewritten as

$$E\{A_0(\omega)\exp[j(\omega k\,\Delta-\phi_0(\omega)]\}-E\left\{\left[\sum_{l=-N}^{N}W_{N+1+l}\exp(-j\omega l\Delta)\right]\cdot\exp(j\omega k\Delta)\right\}=0\qquad\text{for } k=-N,\ldots,0,\ldots,N \tag{10.100}$$

Note that

$$E\{\exp[-j\omega(l-k)\Delta]\}=\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.101}$$

it follows that

$$E\left\{\left[\sum_{l=-N}^{N}W_{N+1+l}\exp(-j\omega l)\right]\exp(j\omega l\Delta)\right\}=\sum_{l=-N}^{N}W_{N+1+l}\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.102}$$

so that (10.100) can be rewritten in matrix form as

$$\mathbf{v}=\mathbf{C}\mathbf{w} \tag{10.103}$$

where

$$v_k=E\{A_0(\omega)\exp[j(\omega k\Delta-\phi_0(\omega))]\} \tag{10.104}$$

$$C_{k,l}=\frac{\sin[\pi B\Delta(l-k)]}{\pi B\Delta(l-k)} \tag{10.105}$$

Consequently, the complex weight vector must satisfy the relation

$$\mathbf{w}=\mathbf{C}^{-1}\mathbf{v} \tag{10.106}$$

Using (10.106) to solve for the optimum complex weight vector, we can find the output residue signal power by using

$$R_{ee}(0)=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}|A(\omega)-F(\omega)|^2\phi_{JJ}(\omega)d\omega \tag{10.107}$$

where $\phi_{JJ}(\omega)$ is the constant interference signal power spectral density. Assume the interference power spectral density is unity across the bandwidth of concern; then the output residue power due only to main channel amplitude variations is given by

$$R_{ee_A}=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}|A_0(\omega)-F(\omega)|^2d\omega \tag{10.108}$$

Since $A(\omega)-F(\omega)$ is orthogonal to $F(\omega)$, it follows that [15]

$$E\{|A(\omega)-F(\omega)|^2\}=E\{|A(\omega)|^2\}-E\{|F(\omega)|^2\} \tag{10.109}$$

and hence

$$R_{ee_A}=\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[A_0^2(\omega)-|F(\omega)|^2]\,d\omega \tag{10.110}$$

It likewise follows from (10.107) that the output residue power contributed by main channel phase variations is given by

$$R_{ee_p} = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} |e^{-j\phi_0(\omega)} - F(\omega)|^2 \phi_{JJ}(\omega)\, d\omega \tag{10.111}$$

where $\phi_0(\omega)$ represents the main channel phase variation. Once again assuming that the input signal spectral density is unity across the signal bandwidth and noting that $[e^{-j\phi_0(\omega)} - F(\omega)]$ must be orthogonal to $F(\omega)$, it immediately follows that

$$\begin{aligned} R_{ee_p} &= \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} [1 - |F(\omega)|^2]\, d\omega \\ &= 1 - \sum_{j=-N}^{N}\sum_{k=-N}^{N} w_k w_j^* \frac{\sin[\pi B\Delta(k-j)]}{\pi B\Delta(k-j)} \end{aligned} \tag{10.112}$$

where the complex weight vector elements must satisfy (10.103)–(10.106).

If it is desired to evaluate the effects of both amplitude and phase mismatching simultaneously, then the appropriate expression for the output residue power is given by (10.107), which (because of orthogonality) may be rewritten as

$$R_{ee}(0) = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} \{|A(\omega)|^2 - |F(\omega)|^2\}\phi_{JJ}(\omega)\, d\omega \tag{10.113}$$

where the complex weights used to obtain $F(\omega)$ must again satisfy (10.102)–(10.106), which now involve both a magnitude and a phase component and it is assumed that $\phi_{JJ}(\omega)$ is a constant.

10.6.1 Example: Effects of Amplitude Mismatching

To evaluate (10.110) it is necessary to adopt a channel amplitude model corresponding to $A(\omega)$. One possible channel amplitude model is given in Figure 10-35 for which

$$A(\omega) = \begin{cases} 1 + R\cos\omega T_0 & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases} \tag{10.114}$$

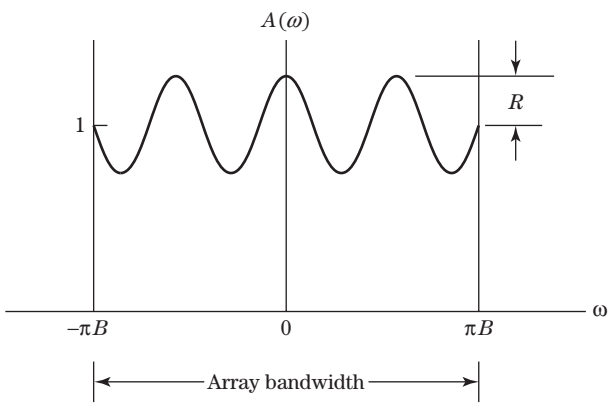


FIGURE 10-35 ■ Channel amplitude model having $3\frac{1}{2}$ cycles of ripple for evaluation of amplitude mismatch effects.

where

$$T_0 = \frac{2n+1}{2B} \qquad \text{for } n = 0, 1, 2, \ldots$$

and the integer n corresponds to $(2n+1)/2$ cycles of amplitude mismatching across the bandwidth B. Letting the phase error $\phi_0(\omega) = 0$, it follows from (10.104) that

$$v_k = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]e^{j\omega k\Delta}\,d\omega \tag{10.115}$$

or

$$v_k = \frac{\sin(\pi Bk\Delta)}{\pi Bk\,\Delta} + \frac{R}{2}\left[\frac{\sin(\pi B[T_0 + k\Delta])}{\pi B\,[T_0 + k\Delta]} + \frac{\sin(\pi B[T_0 - k\Delta])}{B[T_0 - k\Delta]}\right] \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.116}$$

Evaluation of (10.116) permits the complex weight vector to be found, which in turn may be used to determine the residue power by way of (10.110).

Now

$$|F(\omega)|^2 = F(\omega)\,F^*(\omega) = \mathbf{w}^\dagger\boldsymbol{\beta}\boldsymbol{\beta}^\dagger\mathbf{w} \tag{10.117}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} e^{j\omega N\Delta} \\ e^{j\omega(N-1)\,\Delta} \\ \vdots \\ e^{-j\omega N\Delta} \end{bmatrix} \tag{10.118}$$

Carrying out the vector multiplications indicated by (10.117) then yields

$$|F(\omega)|^2 = \sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta} \tag{10.119}$$

The output residue power is therefore given by [see equation (10.110)]

$$R_{ee_A} = \int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]^2\,d\omega - \int_{-\pi B}^{\pi B}\sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta}\,d\omega \tag{10.120}$$

Equation (10.120) may be evaluated using the following expressions:

$$\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}[1 + R\cos\omega T_0]^2\,d\omega = \left(1 + \frac{R^2}{2}\right) + 2R\frac{\sin\pi[(2n+1)/2]}{\pi[(2n+1)/2]} + \frac{R^2}{2}\frac{\sin\pi(2n+1)}{\pi(2n+1)} \tag{10.121}$$

$$\frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^* e^{j\omega(k-i)\Delta}\,d\omega = \sum_{i=1}^{2N+1}\sum_{k=1}^{2N+1} w_i w_k^*\frac{\sin\pi(k-i)B\Delta}{\pi(k-i)B\Delta} \tag{10.122}$$

10.6.2 Results for Compensation of Selected Amplitude Mismatch Model

The evaluation of (10.120) requires knowing the ripple amplitude R, the number of cycles of amplitude mismatching across the bandwidth, and the product of $B\Delta$ (where B is the cancellation bandwidth and Δ is the intertap delay spacing). The results of a computer evaluation of the output residue power are summarized in Figures 10-36–10-39 for $B\Delta = 0.25$, 0.5, 0.75, and 1, and $R = 0.09$. Each of the figures presents a plot of the decibel cancellation (of the undesired interference signal) achieved as a function of the number of taps in the transversal filter and the number of cycles of ripple present across the cancellation bandwidth. No improvement (over the cancellation that can be achieved with only one tap) is realized until a sufficient number of taps is present in the transversal filter to achieve the resolution required by the amplitude versus frequency variations in

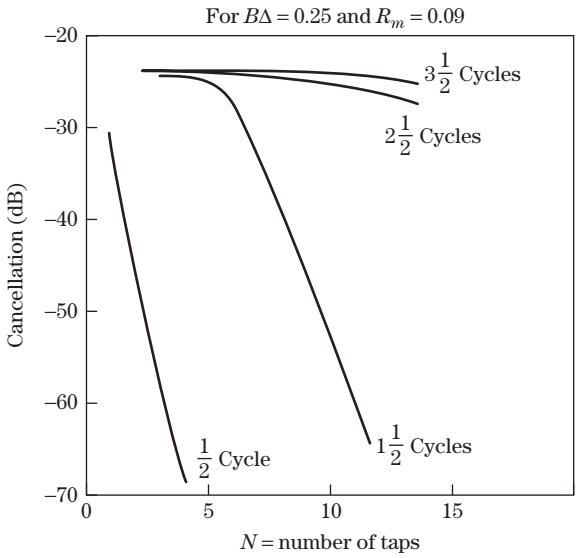


FIGURE 10-36 ■ Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.25$.

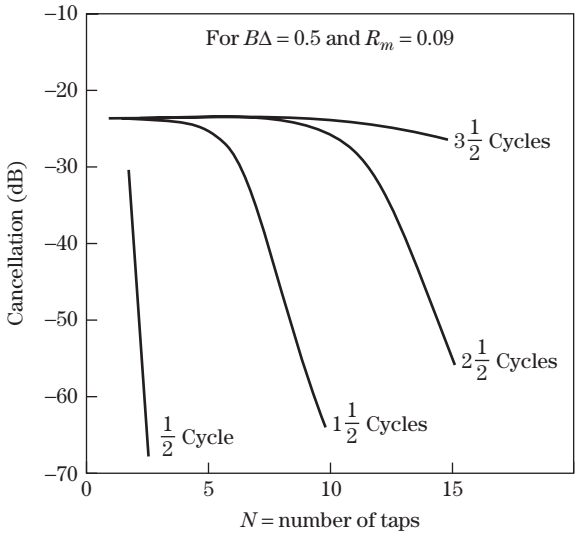


FIGURE 10-37 ■ Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.5$.

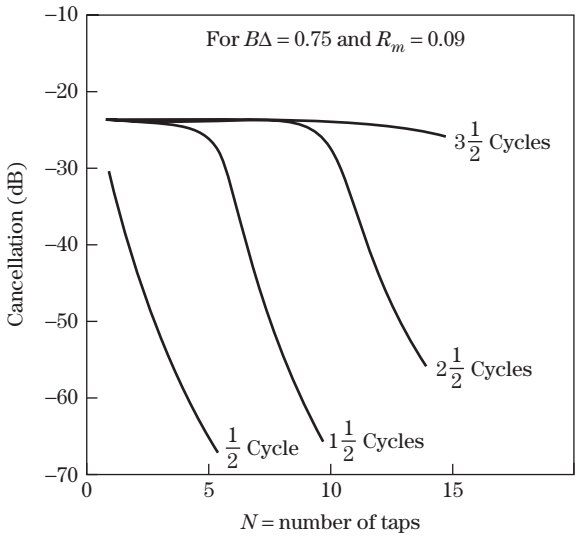


FIGURE 10-38 ■
Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 0.75$.

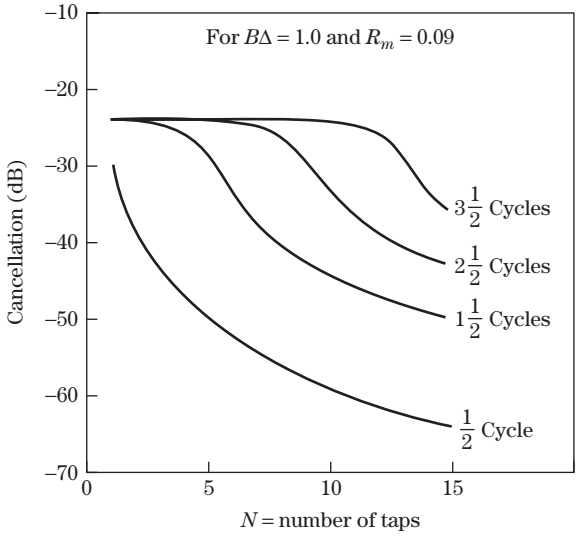


FIGURE 10-39 ■
Decibel cancellation versus number of taps for selected amplitude mismatch models with $B\Delta = 1.0$.

the amplitude mismatch model. The sufficient number of taps for the selected amplitude mismatch model was found empirically to be given by

$$N_{\text{sufficient}} \approx \left(\frac{N_r - 1}{2}\right)[7 - 4(B\Delta)] + 1 \tag{10.123}$$

where N_r is the number of half-cycles of ripple appearing in the mismatch model.

If there are a sufficient number of taps in the transversal filter, the cancellation performance improves when more taps are added depending on how well the resulting transfer function of the transversal filter matches the gain and phase variations of the channel mismatch model. Since the transversal filter transfer function resolution depends in part on the product $B\Delta$, a judicious selection of this parameter ensures that providing additional taps provides a better match (and hence a significant improvement in cancellation performance), whereas a poor choice results in very poor transfer function matching even with the addition of more taps.

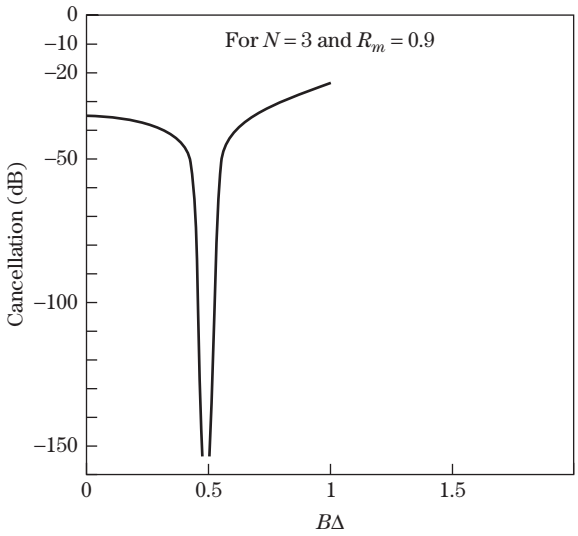


FIGURE 10-40 ■ Decibel cancellation versus $B\Delta$ for one-half-cycle amplitude mismatch model.

Taking the inverse Fourier transform of (10.114) $\mathfrak{F}^{-1}\{A(\omega)\}$ yields a time function corresponding to an autocorrelation function $f(t)$ that can be expressed as

$$f(t) = s(t) + Ks(t \pm T_0) \tag{10.124}$$

The results of Section 10.5.3 and equation (10.124) imply that $\Delta = T_0$ (or equivalently, $B\Delta =$ number cycles of ripple mismatch) if the product $B\Delta$ is to "match" the amplitude mismatch model. This result is illustrated in Figure 10-40 where decibel cancellation is plotted versus $B\Delta$ for a one-half-cycle ripple mismatch model. A pronounced minimum occurs at $B\Delta = \frac{1}{2}$ for $N = 3$ and $R_m = 0.9$.

When the number of cycles of mismatch ripple exceeds unity, the foregoing rule of thumb leads to the spurious conclusion that $B\Delta$ should exceed unity. Suppose, for example, there were two cycles of mismatch ripple for which it was desired to compensate. By setting $B\Delta = 2$ (corresponding to $B_f = \frac{1}{2}B$), two complete cycles for the transversal filter transfer function are found to occur across the cancellation bandwidth. By matching only one cycle of the channel mismatch, quite good matching of the entire mismatch characteristic occurs but at the price of sacrificing the ability to independently adjust the complex weights across the entire cancellation bandwidth, thereby reducing the ability to appropriately process broadband signals. Consequently, if the number of cycles of mismatch ripple exceeds unity, it is usually best to set $B\Delta = 1$ and to accept whatever improvement in cancellation performance can be obtained with that value, or increase the number of taps.

10.6.3 Example: Effects of Phase Mismatching

Let $\phi(\omega)$ corresponding to the phase error be characterized by

$$\phi(\omega) = \begin{cases} A\cos\omega T_0 & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases} \tag{10.125}$$

where A represents the peak number of degrees associated with the phase error ripples. This model corresponds to the error ripple model of (10.112) (with zero average value present).

Since

$$v_k = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B} \exp(j\{A\cos\omega T_0 + \omega k\Delta\})\,d\omega \qquad \text{for } k = -N, \ldots, 0, \ldots, N \tag{10.126}$$

it can easily be shown by defining

$$f(K, \text{sgn}) \triangleq \frac{\sin\pi[K + \text{sgn}\cdot(i - (N+1))B\Delta]}{\pi[K + \text{sgn}\cdot(i - (N+1))B\Delta]}$$

and $g(K) \triangleq f(K, +) + f(K, -)$ that

$$v_i = J_0(A)f(0, +) + jJ_1(A)g\left[\frac{2n+1}{2}\right] + \sum_{k=1}^{\infty}(-1)^k\left\{J_{2k}(A)g[k(2n+1)] + jJ_{2k+1}(A)g\left[(2k+1)\left(\frac{2n+1}{2}\right)\right]\right\} \tag{10.127}$$

where $J_n(\cdot)$ denotes a Bessel function of the nth order for $i = 1, 2, \ldots, 2N+1$.

10.6.4 Results for Compensation of Selected Phase Mismatch Model

The computer evaluation of the output residue power resulted in the performance summarized in Figures 10-41–10-43 for $B\Delta = 0.2$, 0.45, and 1.0 and $A = 5^\circ$. These figures present the decibel cancellation achieved as a function of the number of taps in the transversal filter and the number of cycles of phase ripple present across the cancellation bandwidth. The general nature of the curves appearing in Figures 10-41–10-43 is the same as that of Figures 10-36–10-39 for amplitude mismatching. Furthermore, just as in the amplitude mismatch case, a better channel transfer function fit can be obtained with the transversal filter when the mismatch characteristic has a fewer number of ripples.

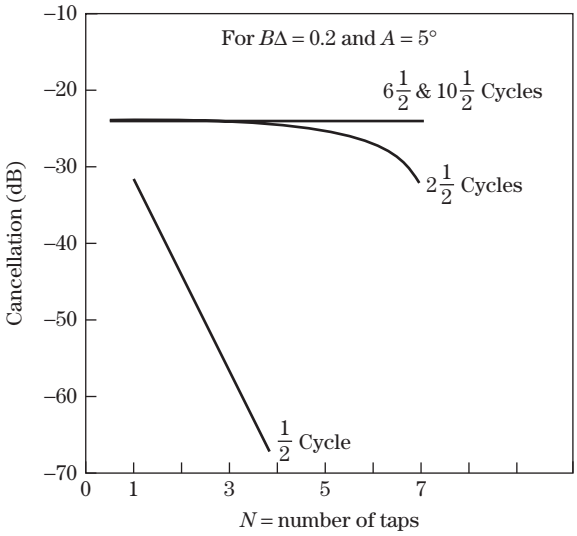


FIGURE 10-41 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 0.2$.

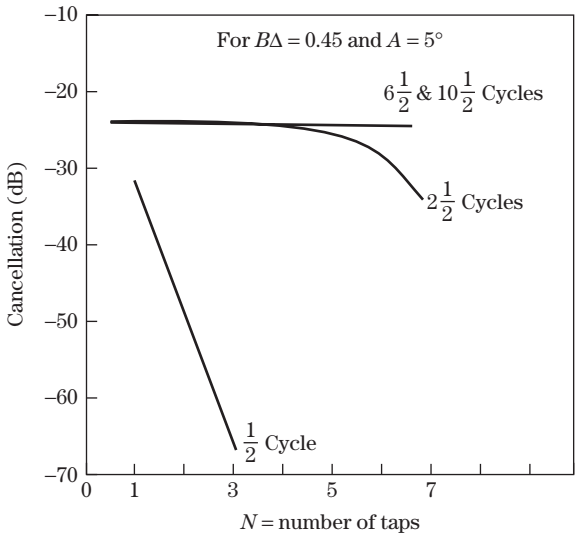


FIGURE 10-42 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 0.45$.

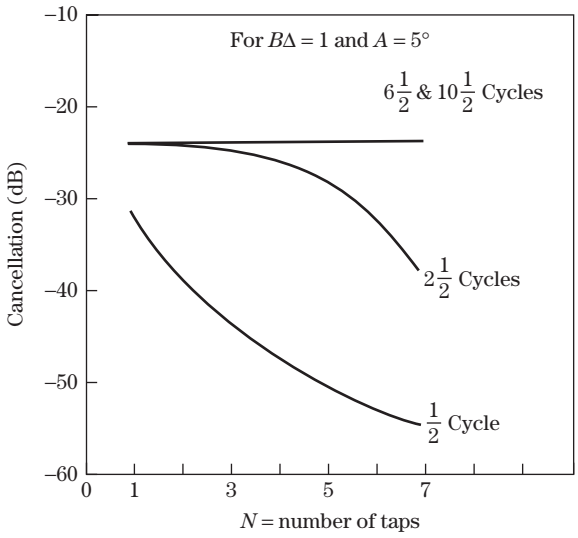


FIGURE 10-43 ■ Decibel cancellation versus number of taps for selected phase mismatch models with $B\Delta = 1.0$.

10.7 SUMMARY AND CONCLUSIONS

Array errors due to manufacturing tolerances distort the array pattern. To minimize these errors, the array must be calibrated at the factory and at regular intervals once deployed.

The transversal filter consisting of a sequence of weighted taps with intertap delay spacing offers a practical means for achieving the variable amplitude and phase weighting as a function of frequency that is required if an adaptive array system is to perform well against wideband interference signal sources. The distortionless channel transfer functions for a two-element array were derived. It was found that to ensure distortion-free response to a broadband signal the channel phase is a linear function of frequency, whereas the channel amplitude function is nearly flat over a 40% bandwidth. Quadrature hybrid processing provides adequate broadband signal response for signals having as much as 20% bandwidth. Tapped delay line processing is a practical necessity for 20% or more

bandwidth signals. A transversal filter provides an attractive means of compensating the system auxiliary channels for the undesirable effects of the following:

1. Multipath interference
2. Interchannel mismatch
3. Propagation delay across the array

For multipath interference, the value of the intertap delay is in the neighborhood of the delay time associated with the multipath ray. If the intertap delay time exceeds the multipath delay time by more than about 30% and the multipath delay time is appreciable, a severe loss of compensation capability is incurred. If the intertap delay is too small, then an excessive number of taps will be required for effective cancellation to occur. Since multipath delay having "small" values of associated time delay do not severely degrade the array performance, it is reasonable to determine the most likely values of multipath delay that will occur for the desired application and base the multipath compensation design on those delay times (assuming $B\Delta \leq 1$). For reflection coefficients of 0.5 and $BD_m = 0.45$, the use of five taps will ensure a -30 dB cancellation capability.

The results shown in Figures 10-31 and 10-32 indicate that array propagation delay effects are usually much easier to compensate than are multipath effects. This result occurs because multipath in effect introduces two (or more) signals in each channel (that are essentially uncorrelated if $BD_m \gg 1$), which require more degrees of freedom to adequately compensate.

The problem presented by interchannel mismatch is to obtain a transfer function with the transversal filter that succeeds in matching the amplitude and phase error characteristics exhibited among the various sensor channels. As might be expected, the more severe the mismatching between channels, the more difficult it is to achieve an acceptable degree of compensation. In particular, it is highly undesirable for more than $2\frac{1}{2}$ cycles of mismatch ripple to occur over the cancellation bandwidth; even this degree of mismatch requires seven taps on the transversal filter before a truly effective degree of compensation can be achieved. It may very well result that the best choice of intertap delay spacing for the interchannel mismatch characteristic of concern is far different from the optimum choice of intertap delay selected for multipath compensation; should this actually occur, it is necessary to adopt a compromise value for the intertap delay spacing. Such a compromise value for the intertap delay spacing hopefully results in an acceptable degree of compensation for both multipath and interchannel mismatch effects.

10.8 | PROBLEMS

Distortionless Transfer Functions

1. From (10.11) and (10.12) it immediately follows that $|H_1(\omega) = |H_2(\omega)|$, thereby yielding the pair of equations

$$f_1\{|H_1|, \alpha_1, \alpha_2, \theta_s\} = \exp(-j\omega T_1)$$

and

$$f_2\{|H_1|, \alpha_1, \alpha_2, \theta_i\} = 0$$

(a) Show from the previous pair of equations that $\alpha_1(\omega)$ and $\alpha_2(\omega)$ must satisfy

$$\alpha_2(\omega) - \alpha_1(\omega) = \frac{\pi\omega}{\omega_0} \sin\theta_i \pm n\pi$$

where n is any odd integer.

(b) Since the magnitude of $\exp(-j\omega T_1)$ must be unity, show using $f_1\{\ \} = \exp(-j\omega T_1)$ that (10.13) results.

(c) Show that the angle condition associated with $f_1\{\ \} = \exp(-j\omega T_1)$ yields (10.14).

(d) Show that substituting (10.14) into the results from part (a) yields (10.15).

2. For a three-element linear array, the overall transfer function encountered by the desired signal in passing through the array is

$$H_d(\omega) = H_1(\omega) + H_2(\omega)\exp\left(-j\frac{\omega d}{c}\sin\theta_s\right) + H_3(\omega)e\left(-j\frac{\omega 2d}{c}\sin\theta_s\right)$$

and the overall transfer function seen by the interference signal is

$$H_I(\omega) = H_1(\omega) + H_2(\omega)\exp\left(-j\frac{\omega d}{c}\sin\theta_i\right) + H_3(\omega)e\left(-j\frac{\omega 2d}{c}\sin\theta_i\right)$$

What does imposing the requirements (10.9) and (10.10) now imply for the three-channel transfer functions?

Hilbert Transform Relations

3. Prove the Hilbert transform relations given by (10.25)–(10.28).

4. Using (10.61), (10.62), and the results of (10.63)–(10.65), show that $\mathbf{R}_{ee}$ is given by (10.66).

5. Derive the correlation functions given by (10.78)–(10.80) for the signal environment assumptions (10.75) and (10.76)

6. Show that as the time–bandwidth product $B\Delta$ approaches zero, then the matrix $\mathbf{R}_{xx}(0)$ [whose elements are given by (10.80)] becomes singular so that matrix inversion cannot be accomplished.

Compensation for Channel Phase Errors

7. For the phase error $\phi(\omega)$ given by (10.125), show that v_k given by (10.127) follows from the application of (10.126).

8. Let $\phi(\omega)$ correspond to the phase error model be given by

$$\phi(\omega) = \begin{cases} A\left[1 - \cos\frac{2\omega}{B}\right] & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases}$$

Show that v_k of (10.126) is given by

$$v_k = \int_{-\pi B}^{\pi B}\left[\cos\left\{A\left(1-\cos\omega\frac{2}{B}\right)\right\} + j\sin\left\{A\left(1-\cos\omega\frac{2}{B}\right)\right\}\right]\exp(j\omega k\Delta_{d\omega})$$

Use the trigonometric identities

$$\cos\left[A - A\cos\omega\frac{2}{B}\right] = \cos A\cos\left[A\cos\omega\frac{2}{B}\right] + \sin A\sin\left[A\cos\omega\frac{2}{B}\right]$$

$$\sin\left[A - A\cos\omega\frac{2}{B}\right] = \sin A\cos\left[A\cos\omega\frac{2}{B}\right] - \cos A\sin\left[A\cos\omega\frac{2}{B}\right]$$

and the fact that

$$\cos(A\cos\omega T_0) = J_0(A) + 2\sum_{k=1}^{\infty}(-1)^k \cdot J_{2k}(A)\cos[(2k)\omega T_0]$$

$$\sin(A\cos\omega T_0) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(A)\cdot\cos[(2k+1)\omega T_0]$$

where $J_n(\cdot)$ denotes a Bessel function of the nth order and define

$$f(n, \text{sgn}) \triangleq \frac{\sin\pi[n + \text{sgn}\cdot(i-(N+1))B\Delta]}{\pi[n+\text{sgn}\cdot(i-(N+1))B\Delta]}$$

$$g(n) \triangleq f(n,+) + f(n,-)$$

to show that

$$\begin{aligned} v_i = {} & J_0(A)\cdot f(0,+)[\cos A + j\sin A] + J_1(A)\cdot g(2)[\sin A - j\cos A] \\ & + \sum_{k=1}^{\infty}(-1)^k\{J_{2k}(A)\cdot g(4k)[\cos A + j\sin A] \\ & + J_{2k+1}(A)\cdot g[(2k+1)2][\sin A - j\cos A]\} \\ \text{for } i = {} & 1, 2, \ldots, 2N+1 \end{aligned}$$

9. Let $\phi(\omega)$ corresponding to the phase error model be given by

$$\phi(\omega) = \begin{cases} b\omega^2(\pi B - |\omega|) & \text{for } |\omega| \le \pi B \\ 0 & \text{otherwise} \end{cases}$$

As before, it follows that

$$v_i = \frac{1}{2\pi B}\int_{-\pi B}^{\pi B}\exp\{j[b\omega^2(\pi B - |\omega|) + \omega i\Delta]\}\,d\omega$$

Letting $u = \omega/\pi B$, applying Euler's formula, and ignoring all odd components of the resulting expression, show that

$$v_i = \int_0^1 \exp\left\{j\left[\frac{27A}{4}u^2(1-u)\right]\right\}\cos\pi[u(i-(N+1))B\Delta]du$$

where $A = 4b(\pi B/3)^3$ for $i = 1, 2, \ldots, 2N+1$. The foregoing equation for v_i can be evaluated numerically to determine the output residue power contribution due to the previous phase error model.

Computer Simulation Problems

10. A 30-element linear array ($d = 0.5\lambda$) has a 20 dB, $\overline{n} = 2$ Taylor taper applied at the elements. Plot the array factor when $\delta_n^a = 0.1$ and $\delta_n^a = 0.1$.

11. A 30-element linear array ($d = 0.5\lambda$) has a 30 dB, $\overline{n} = 7$ low sidelobe taper. Plot the array factors for a single element failure at (1) the edge and (2) the center of the array.

12. Find the location and heights of the quantization lobes for a 20-element array with $d = 0.5\lambda$ and the beam steered to $\theta = 3^\circ$ when the phase shifters have three, four, and five bits.

10.9 REFERENCES

[1] E. Brookner, *Antenna Array Fundamentals—Part 2, Practical Phased Array Antenna Systems,* E. Brookner, ed., Norwood, MA, Artech House, 1991

[2] R.J. Mailloux, "Array Grating Lobes Due to Periodic Phase, Amplitude, and Time Delay Quantization," *IEEE AP-S Trans.*, Vol. 32, December 1984, pp. 1364–1368.

[3] A. J. Boonstra, and A. J. van der Veen, "Gain Calibration Methods for Radio Telescope Arrays," *IEEE Signal Processing Trans.,* Vol. 51, No. 1, 2003, pp. 25–38.

[4] R. Sorace, "Phased Array Calibration," *IEEE AP-S Trans.,* Vol. 49, No. 4, 2001, pp. 517–525.

[5] T. Takahashi, Y. Konishi, S. Makino, et al., "Fast Measurement Technique for Phased Array Calibration," *IEEE AP-S Trans.,* Vol. 56, No. 7, 2008, pp. 1888–1899.

[6] M. Borkowski, "Solid-State Transmitters," *Radar Handbook*, M. I. Skolnick, ed., pp. 11.1–11.36, New York, McGraw Hill, 2008.

[7] N. Tyler, B. Allen, and H. Aghvami, "Adaptive Antennas: The Calibration Problem," *IEEE Communications Magazine,* Vol. 42, No. 12, 2004, pp. 114–122.

[8] N. Tyler, B. Allen, and A. H. Aghvami, "Calibration of Smart Antenna Systems: Measurements and Results," *IET Microwaves, Antennas & Propagation,* Vol. 1, No. 3, 2007, pp. 629–638.

[9] R. L. Haupt, *Antenna Arrays: A Computational Approach*, New York, Wiley, 2010.

[10] W. T. Patton and L. H. Yorinks, "Near-Field Alignment of Phased-Array Antennas," *IEEE AP-S Trans.,* Vol. 47, No. 3, 1999, pp. 584–591.

[11] W. E. Rodgers and R. T. Compton, Jr., "Tapped Delay-Line Processing in Adaptive Arrays," Report 3576-3, April 1974, prepared by The Ohio State University Electro Science Laboratory, Department of Electrical Engineering under Contract N00019-73-C-0195 for Naval Air Systems Command.

[12] W. E. Rodgers and R. T. Compton, Jr., "Adaptive Array Bandwidth with Tapped Delay-Line Processing," Report 3832-3, May 1975, prepared by The Ohio State University Electro Science Laboratory, Department of Electrical Engineering under Contract N00019-74-C-0141 for Naval Air Systems Command.

[13] W. E. Rodgers and R. T. Compton, Jr., "Adaptive Array Bandwidth with Tapped Delay-Line Processing," *IEEE Trans. Aerosp. Electron. Syst*., Vol. AES-15, No. 1, January 1979, pp. 21–28.

[14] T. G. Kincaid, "The Complex Representation of Signals," General Electric Report No. R67EMH5, October 1966, HMED Publications, Box 1122 (Le Moyne Ave.), Syracuse, NY, 13201.

[15] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York, McGraw-Hill, 1965, Ch. 7.

[16] H. Steyskal & J. Herd, "Mutual Coupling Compensation in Small Array Antennas," *IEEE Trans. Ant. & Prop.,* Vol. AP-38, No. 12, December 1995, pp. 603–606.

[17] R. A. Monzingo, "Transversal Filter Implementation of Wideband Weight Compensation for CSLC Applications," unpublished Hughes Aircraft Interdepartmental Correspondence Ref. No. 78-1450.10/07, March 1978.

[18] A. M. Vural, "Effects of Perturbations on the Performance of Optimum/Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst*., Vol. AES-15, No. 1, January 1979, pp. 76–87.

CHAPTER 11

Direction of Arrival Estimation and Related Topics

Chapter Outline

An array's ability to resolve signals depends on the beamwidth of the array, so high-resolution algorithms have been developed, in order for small arrays can resolve closely spaced signals by using narrow nulls in place of the wide main beam. A linear array with elements separated by half-wavelength spacing has $N - 1$ nulls in which to locate up to $N - 1$ signals.

In this chapter, we apply maximum likelihood (ML) estimation methods to estimate the direction of arrival (DOA), or angle of arrival (AOA), of one or more signal sources, using data received by the elements of an N-element antenna array. The Cramer–Rao (CR) lower bound on angle estimation error is derived under several different signal assumptions. The CR bound helps determine system performance versus signal-to-noise ratio (SNR) and array size.

The advantage of optimal array estimation methods (processing the antenna element signals in an optimum fashion) lies in its application to multiple signal environments and in conditions where interfering signals are present. If only one signal is present in a white noise background, conventional monopulse processing achieves the same AOA estimation accuracy. High angular resolution of desired signals is achieved beyond the one

beamwidth resolution limit of conventional monopulse processing. Angular accuracy and angular resolution depends not only on array beamwidth but also on the SNR of the desired signals and the number of snapshots of the data vector. Fractional beamwidth resolution, or "super-resolution," of two or more signal sources incident on an array offers a potential solution to a wide variety of radar and communications problems, including fractional beamwidth resolution of multiple interference sources, reduction of multipath-induced angle of arrival estimation error, improved angle tracking accuracy in the presence of main beam and sidelobe interference, spatial isolation of two closely spaced signals, and improved ability to separate slow-moving and ground clutter in a radar that is moving, as in an airborne radar.

The ML AOA estimation method is one of several useful super-resolution techniques developed in the literature. While the ML methods come very close to theoretical bounds on AOA estimation error (e.g., the Cramer–Rao bound), they are computationally intensive when the AOAs of multiple signals are estimated at the same time. As a result, computationally simpler methods have emerged that perform nearly as well. Further drawbacks of the ML method are that the desired signal sources (sources whose AOAs are to be estimated) are represented as point sources and that the number of sources is known. In cases where the spatial spectrum contains spatially distributed sources (e.g., heavy multipath environment) or where an accurate estimate of the number of sources is difficult (dense signal environment, heavy multipath), some of the minimum variance methods are more appropriate, at the expense of AOA accuracy and resolution.

The AOA techniques presented here apply to a wide range of parameter estimation problems. Examples include time of arrival estimation (e.g., target distance in a radar), frequency estimation (e.g., target Doppler frequency, estimation of sinusoids), and signal intensity.

11.1 PERIODOGRAM

The simplest approach to finding the direction of a signal is to scan the main beam of the array by adjusting the steering vector until the signal is detected. The relative output power of a linear array lying along the x-axis is given by

$$P(\theta) = \mathbf{A}^{\dagger}(\theta)\mathbf{R}_{xx}\mathbf{A}(\theta) \tag{11.1}$$

and $\mathbf{R}_T$ is the signal plus noise correlation matrix where the subscript "T" denotes where the uniform array steering vector is given by

$$\mathbf{A}(\theta) = \left[e^{jkx_1\cos\theta}\cdots e^{jkx_N\cos\theta}\right]^T, \theta_{\min} \le \theta \le \theta_{\max} \tag{11.2}$$

A periodogram is a plot of the output power versus angle, where a window function that is independent of the data being analyzed must be adopted. By weighting all angles equally, a rectangular window function is in effect being adopted. Peaks in the periodogram correspond to signal locations. Large arrays have a narrower beamwidth than smaller arrays and can resolve closely spaced signals better. Figure 11-1 shows the periodigram for a 12-element array with $\lambda/2$ spacing and three sources incident at $\theta = -50^\circ$, 10°, and 20°. The source at $\theta = -50^\circ$ is easy to distinguish, but the sources at $\theta = 10^\circ$, and 20° appear to be a single source, because the beamwidth is too wide. This example demonstrates the need for super-resolution techniques.

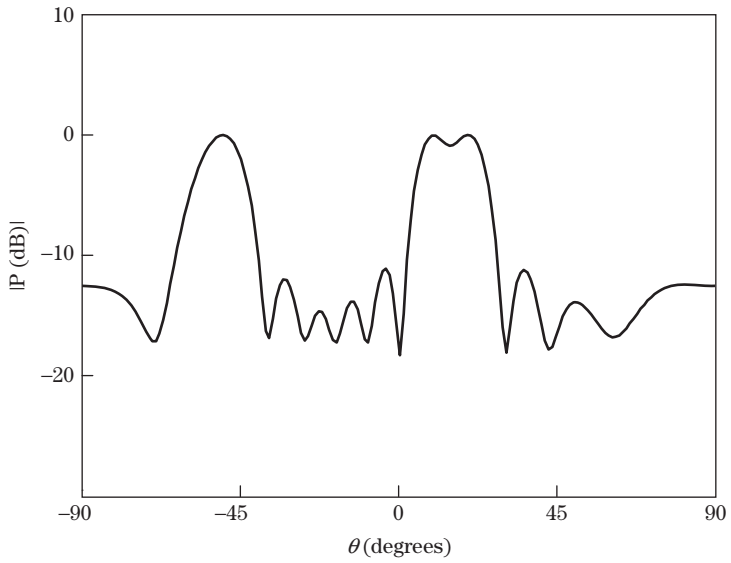


FIGURE 11-1 ■ Plot of the periodogram of a 12-element uniform array when three sources are incident at $\theta = -50^\circ, 10^\circ,$ and 20°.

11.2 CAPON'S METHOD

The array beamwidth, which is inveresely proportional to the array size, limits periodogram resolution. Nulls are better suited to locate a signal than the main beam. The maximum likelihood estimate of the power arriving from a desired direction, while all the other sources are considered interference, is known as Capon's method (or the Maximum Likelihood Method [MLM])[1]. Both the MLM and Maximum Entropy Method (to be discussed later) do not have fixed window functions associated with them, a fact that ameliorates the windowing problem (in effect, the window function adapts itself to the data under analysis). The desired signal remains constant while the output power is minimized. The array weights that maximize the signal-to-interference ratio are

$$\mathbf{w} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{A}}{\mathbf{A}^\dagger \mathbf{R}_{xx}^{-1}\mathbf{A}} \tag{11.3}$$

The Capon spectrum is the denominator of (11.3)

$$P(\theta) = \frac{1}{\mathbf{A}^\dagger(\theta)\mathbf{R}_{xx}^{-1}\mathbf{A}(\theta)} \tag{11.4}$$

Figure 11-2 shows the periodigram for a 12-element array with $\lambda/2$ spacing and three sources incident at $\theta = -50^\circ$, 10°, and 20°. Capon's method distinguishes between two closely spaced sources much better than the periodogram.

11.3 MUSIC ALGORITHM

*MU*ltiple *SI*gnal *C*lassification (MUSIC) assumes the noise is uncorrelated and the signals have little correlation [2,3]. The MUSIC spectrum is

$$P(\theta) = \frac{\mathbf{A}^\dagger(\theta)\mathbf{A}(\theta)}{\mathbf{A}^\dagger(\theta)\mathbf{V}_\lambda \mathbf{V}_\lambda^\dagger \mathbf{A}(\theta)} \tag{11.5}$$

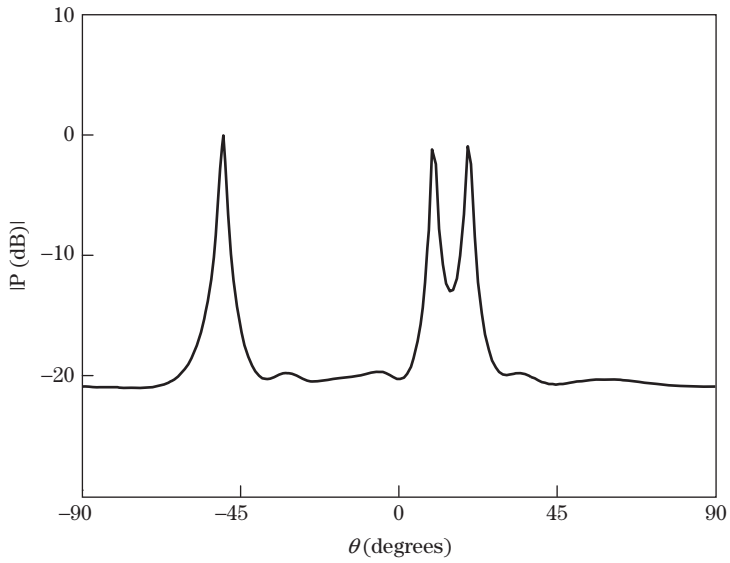


FIGURE 11-2 ■ Plot of the Capon spectrum of a 12-element uniform array when three sources are incident at $\theta = -50^\circ, 10^\circ,$ and 20°.

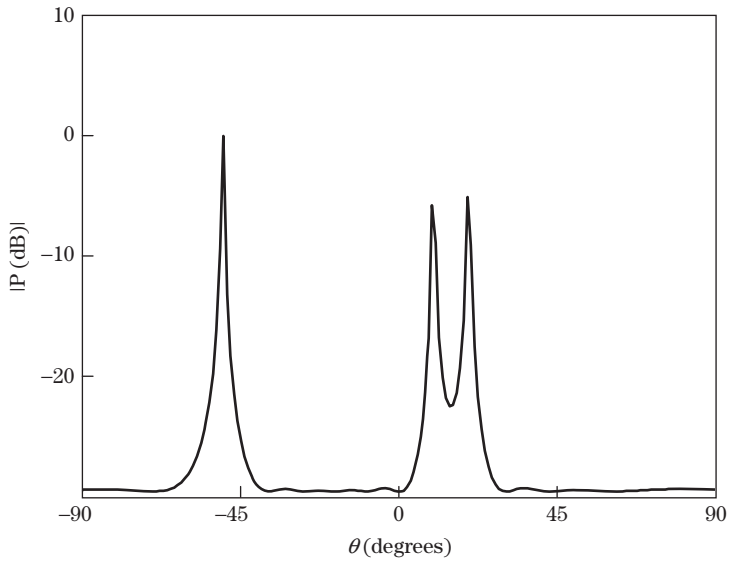


FIGURE 11-3 ■ Plot of the MUSIC spectrum of a 12-element uniform array when three sources are incident at $\theta = -50^\circ, 10^\circ,$ and 20°.

The signal plus noise correlation matrix in the denominator of (11.3) is replaced by $\mathbf{V}_\lambda \mathbf{V}_\lambda^\dagger$ where the columns of V_λ are the eigenvectors of the noise subspace. In the numerator, $\mathbf{A}^\dagger(\theta)$ replaces $\mathbf{R}_{xx}^{-1}$ and corresponds to the $N - N_s$ smallest eigenvalues of the correlation matrix. Figure 11-3 shows the MUSIC spectrum for a 12-element array with $\lambda/2$ spacing and three sources incident at $\theta = -50^\circ, 10^\circ,$ and 20°. The MUSIC spectrum is similar to the Capon spectrum, except the floor between the peaks is much lower for the MUSIC spectrum.

The root-MUSIC algorithm is a more robust alternative that accurately locates the direction of arrival by finding the roots of the array polynomial that corresponds to the

denominator of (11.5) [4].

$$\mathbf{A}^{\dagger}(\theta)\mathbf{V}_{\lambda}\mathbf{V}_{\lambda}^{\dagger}\mathbf{A}(\theta) = \sum_{n=M+1}^{M-1} c_n z^n \tag{11.6}$$

where

$$z = e^{j\frac{2\pi}{\lambda}nd\sin\theta}$$
$$c_n = \sum_{r-c=n} \mathbf{V}_{\lambda}\mathbf{V}_{\lambda}^{\dagger}$$

The c_n results from summing the diagonal $n = r - c$ of $V_{\lambda}V_{\lambda}^{\dagger}$ in (11.6) with r and c indicating the row and column, respectively, of the matrix. Polynomial roots, z_m, on the unit circle (Chapter 2) are the poles of the MUSIC spectrum. The phase of the polynomial roots in (11.6) are given by

$$\theta_m = \sin^{-1}\left(\frac{\arg(z_m)}{kd}\right) \tag{11.7}$$

Roots on the unit circle correspond to the signals. Roots off the unit circle are spurious. The $2N-1$ diagonals of $V_{\lambda}V_{\lambda}^{\dagger}$ form a polynomial with $2N-2$ roots. Table 11-1 contains the roots of the polynomial for a 12-element array with $\lambda/2$ spacing and three sources incident at $\theta = -50^{\circ}$, 10°, and 20°. Roots on the unit circle correspond to signals and have a "yes" in column 3. Spurious roots are off the unit circle and have a "no" in column 3. All roots appear in the unit circle plot in Figure 11-4. Note that each root on the unit circle is actually a double root (see Table 11-1), so it appears that there are only 19 roots in Figure 11-4 when there are actually 22 roots.

A unitary (real-valued) root-MUSIC algorithm reduces the computational complexity of the root-MUSIC algorithm by exploiting the eigen decomposition of a real-valued correlation matrix. Unitary root MUSIC improves threshold and asymptotic performances relative to conventional root MUSIC.

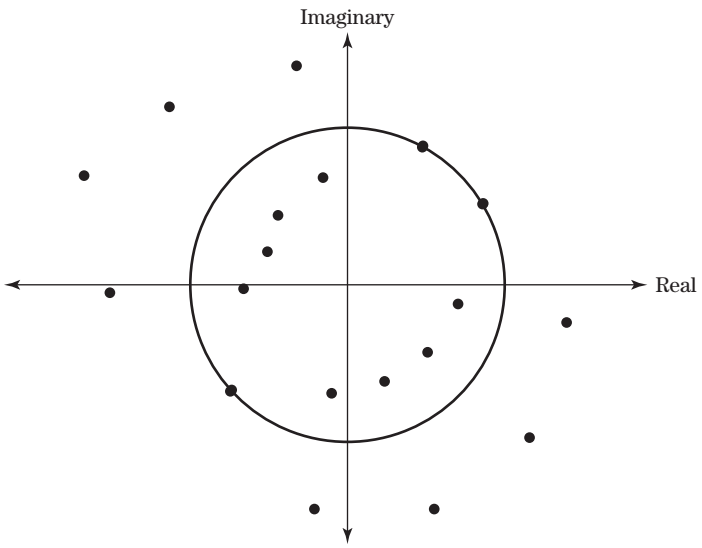


FIGURE 11-4 ■ Unit circle representation of all the roots found using root MUSIC.

TABLE 11-1 ■ The roots found using root MUSIC. The ones close to the unit circle represent the correct signal directions.

m	Roots	Signal Present
1	−1.6766 +j0.6959	*no*
2	0.5468 −j1.4237	*no*
3	1.1541 −j0.9681	*no*
4	1.3909 −j0.2378	*no*
5	−0.2116 −j1.4237	*no*
6	−1.1328 +j1.1311	*no*
7	−1.5093 −j0.0508	*no*
8	−0.3246 +j1.3931	*no*
9	0.8585 +j0.5181	*yes*
10	0.8538 +j0.5153	*yes*
11	0.4787 +j0.8843	*yes*
12	0.4735 +j0.8746	*yes*
13	−0.7461 −j0.6749	*yes*
14	−0.7372 −j0.6668	*yes*
15	0.6986 −j0.1194	*no*
16	0.5086 −j0.4266	*no*
17	0.2351 −j0.6121	*no*
18	−0.1021 −j0.6872	*no*
19	0.1587 +j0.6809	*no*
20	−0.6618 −j0.0223	*no*
21	−0.4421 +j0.4414	*no*
22	−0.5088 +j0.2112	*no*

11.4 THE MAXIMUM ENTROPY METHOD

The maximum entropy method (MEM) is sometimes called the all poles model or the autoregressive model [5–8]. MEM spectral estimation is widely used in geophysics, speech processing, sonar, and radar [9–12]. In addition to spectrum analysis, MEM applies to bearing estimation problems for signals received by an array of sensors [13,14]. High resolution comes from extrapolating the partially known autocorrelation function beyond the last known lag value in a manner that maximizes the entropy of the corresponding power spectrum at each step of the extrapolation [5,15]. If the autocorrelation function is unknown, then the power spectrum is estimated directly from the available time series data using a method devised by Burg [16]. Excellent power spectrum estimates are obtained from relatively short time series data record lengths, and the approach has a rapid convergence rate [17].

In its most elementary form the maximum entropy principle for estimating the power spectrum of a single-channel, stationary, complex time series can be stated as a problem of finding the spectral density function $\phi_{xx}(f)$ that maximizes

$$\text{entropy} = \int_{-W}^{W} \ln \phi_{xx}(f) df \tag{11.8}$$

under the constraint that $\phi_{xx}(f)$ satisfies a set of N linear measurement equations

$$\int_{-W}^{W} \phi_{xx}(f) G_n(f) df = g_n, \quad n = 1, \ldots, N \tag{11.9}$$

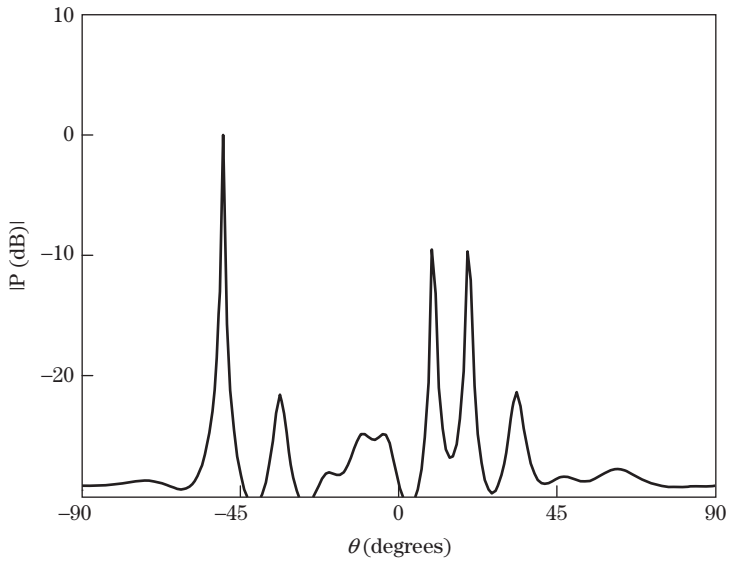


FIGURE 11-5 ■ Plot of *the 12-element uniform array MEM spectrum with three signals present.*

where the time series is sampled with the uniform period Δt so the Nyquist fold-over frequency is $W = 1/2\Delta t$, and the power spectrum of the time series is bandlimited to $\pm W$. The functions $G_n(f)$ in the measurement equations are known test functions, and the g_n are the observed values resulting from the measurements.

MEM is based on a rational function model of the spectrum that has only poles and not zeros [18]. The MEM spectrum is given by [19]

$$P(\theta) = \frac{1}{\mathbf{A}^{\dagger}(\theta)\mathbf{R}_{xx}^{-1}[:, n]\mathbf{R}_{xx}^{\dagger -1}[:, n]\mathbf{A}(\theta)} \tag{11.10}$$

where n corresponds to the nth column of the inverse correlation matrix. Results depend on which n is chosen. Figure 11-5 shows the MEM spectrum for a 12-element array with $\lambda/2$ spacing and three sources incident at $\theta = -50^{\circ}$, 10°, and 20°. Very sharp peaks in the spectrum occur in the signal directions.

Two cases can now be considered: the first where the autocorrelation function is partially known; and the second where the autocorrelation function is unknown.

11.4.1 Partially Known Autocorrelation Function

Let $x(t)$ represent the time series of a stationary (scalar) random process with an associated autocorrelation function $r(\tau)$ for which N discrete lag values $\{r(0), r(1), \ldots, r(N-1)\}$ are known. An estimate of $r(N)$ lies outside the interval of known autocorrelation function values. The basic autocorrelation function theorem states that $r(N)$ must have a value such that the $(N+1) \times (N+1)$ Hermitian Toeplitz autocorrelation matrix given by

$$\mathbf{R}_{nn} \triangleq \begin{bmatrix} r(0) & r(-1) & \ldots & r(-N) \\ r(1) & r(0) & \ldots & r(1-N) \\ \cdot & \ldots & \ldots & \cdot \\ r(N) & r(N-1) & \ldots & r(0) \end{bmatrix} \tag{11.11}$$

is positive semidefinite (i.e., all subdeterminants of $\mathbf{R}_{nn}$ must be nonnegative). Since $\det(\mathbf{R}_N)$ is a quadratic function of $r(N)$, two values of $r(N)$ make the determinant equal to zero. These two values of $r(N)$ define boundaries within which the predicted value of $r(N)$ must fall. The MEM procedure seeks to select the value of $r(N)$ that maximizes $\det(\mathbf{R}_N)$. For a Gaussian random process this procedure is equivalent to maximizing (11.8) subject to the constraint equations [16]:

$$r(n) = \int_{-W}^{W} \phi_{xx}(f) \exp(j2\pi f n \Delta t) df, \quad -N \leq n \leq N \tag{11.12}$$

Maximizing $\det(\mathbf{R}_N)$ is equivalent to finding coefficients for a prediction error filter, and these coefficients play an important role in finding the MEM spectral estimate. Before finding these coefficients, it is instructive to examine the role played by a prediction error filter in obtaining the MEM spectral estimate for a partially known autocorrelation function. Suppose there are N samples of $x(t)$ denoted by $x_0, x_1, \ldots, x_{N-1}$ where each of the samples is taken Δt seconds apart. A linear predicted estimate of x_N based on the previous sampled values of $x(t)$ is obtained from an $(N+1)$-point prediction filter as follows:

$$\hat{x}_N = -\sum_{i=1}^{N} a(N, i) x_{N-i} \tag{11.13}$$

The error associated with $\hat{x}_N$ is then given by

$$\varepsilon_N = x_N - \hat{x}_N = x_N + \sum_{i=1}^{N} a(N, i) x_{N-i} \tag{11.14}$$

Equation (11.14) can be written in matrix form as

$$\varepsilon_N = \mathbf{a}_N^T \mathbf{x} \tag{11.15}$$

where $\mathbf{x}^T = [x_N, x_{N-1}, \ldots, x_0]$, and $\mathbf{a}_N^T = [1, a(N,1), a(N,2), \ldots, a(N,N)]$, where the coefficient $a(N, N) = C_N$ is called the reflection coefficient of order N. The error ε_N is regarded as the output of an Nth order prediction error filter whose coefficients are given by the vector $\mathbf{a}_N$ and whose power output is

$$P_N = E\{\varepsilon_N^2\} \tag{11.16}$$

It is desirable to minimize the MSE of (11.16) by appropriately selecting the prediction filter coefficients contained in $\mathbf{a}_N$. To obtain the minimum mean square estimate $\hat{x}_N$, the error ε_N must be orthogonal to the past data so that

$$E\{x_i \varepsilon_N\} = 0 \qquad \text{for } i = 0, 1, \ldots, N-1 \tag{11.17}$$

Furthermore, the error ε_N must be uncorrelated with all past estimation errors so that

$$E\{\varepsilon_N \varepsilon_{N-k}\} = 0 \qquad \text{for } k = 1, 2, \ldots, N-1 \tag{11.18}$$

Equations (11.16) and (11.8) are the conditions required for a random process to have a white power spectrum of total power P_N (or a power density level of $P_N/2W$ where $W = 1/2\Delta t$). The prediction error filter is regarded as a whitening filter that operates

FIGURE 11-6 ■ Whitening filter representation of prediction error filter.

on the input data $\{x_0, x_1, \ldots, x_{N-1}\}$ to produce output data having a white power density spectrum of level $P_N/2W$ as indicated in Figure 11-6. It immediately follows that the estimate $\hat{\phi}_{xx}(f)$ of the input power spectrum $\phi_{xx}(f)$ is given by

$$\hat{\phi}_{xx}(f) = \frac{P_N/2W}{\left|1 + \sum_{n=1}^{N} a(N,n)\exp(-j2\pi fn\Delta t)\right|^2} \tag{11.19}$$

where the denominator of (11.19) is recognized as the power response of the prediction error filter. Equation (11.19) yields the MEM estimate of $\phi_{xx}(f)$ provided that the coefficients $a(N,n)$, $n = 1, \ldots, N$, and the power P_N can be determined.

A relationship among the coefficients $a(N,n)$, $n = 1, \ldots, N$, the power P_N, and the autocorrelation function values $r(-N), r(-N+1), \ldots, r(0), \ldots, r(N-1), r(N)$ is provided by the well-known prediction error filter matrix equation [14]

$$\begin{bmatrix} r(0) & r(-1) & \ldots & r(-N) \\ r(1) & r(0) & \ldots & r(-N+1) \\ \vdots & \vdots & & \vdots \\ r(N) & r(N-1) & \ldots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a(N,1) \\ \vdots \\ a(N,N) \end{bmatrix} = \begin{bmatrix} P_N \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11.20}$$

Equation (11.20) can be derived [as done in [16] by maximizing the entropy of (11.8)] subject to the constraint equations (11.12). If we know the autocorrelation values $\{r(-N), r(-N+1), \ldots, r(-1), r(0), r(1), \ldots, r(N-1), r(N)\}$, the coefficients $a(N,n)$ and the power P_N may then be found using (11.20). Equation (11.20) can be written in matrix form as

$$\mathbf{R}_{nn}\mathbf{a}_N = \begin{bmatrix} P_N \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad \mathbf{a}_N = \mathbf{R}_{nn}^{-1} \begin{bmatrix} P_N \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11.21}$$

Let

$$\mathbf{R}_{nn}^{-1} = \begin{bmatrix} z_{11} & z_{12} & \cdots \\ z_{21} & z_{22} & \\ \vdots & & \\ z_{N1} & & \end{bmatrix} \tag{11.22}$$

It then follows that

$$\mathbf{a}_N^T = \left[1, \frac{z_{21}}{z_{11}}, \frac{z_{31}}{z_{11}}, \ldots, \frac{z_{N1}}{z_{11}}\right] \tag{11.23}$$

Ulrych and Bishop [6] also give a convenient recursive procedure for determining the coefficients in $\mathbf{a}_N$.

Having determined the prediction error filter coefficients and the corresponding MEM spectral estimate, we must now consider how the autocorrelation function can be extended beyond $r(N)$ to $r(N+1)$ where the autocorrelation values $r(0), r(1), \ldots, r(N)$ are all known. Suppose for example that $r(0)$ and $r(1)$ are known, and it is desired to extrapolate the autocorrelation function to the unknown value $r(2)$. The prediction error filter matrix equation for the known values $r(0)$ and $r(1)$ is given by

$$\begin{bmatrix} r(0) & r(-1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a(1,1) \end{bmatrix} = \begin{bmatrix} P_1 \\ 0 \end{bmatrix} \tag{11.24}$$

where $r(-1) = r^*(1)$. To determine the estimate $\hat{r}(2)$, append one more equation to the previous matrix equation by incorporating $\hat{r}(2)$ into the autocorrelation matrix to yield [6]

$$\begin{bmatrix} r(0) & r(-1) & \hat{r}(-2) \\ r(1) & r(0) & r(-1) \\ \hat{r}(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a(1,1) \\ 0 \end{bmatrix} \begin{bmatrix} P_1 \\ 0 \\ 0 \end{bmatrix} \tag{11.25}$$

Solving (11.25) for $\hat{r}(2)$ then yields $\hat{r}(2) + a(1,1)r(1) = 0$ or

$$\hat{r}(2) = -a(1,1)r(1) = -C_1 r(1) \tag{11.26}$$

Continuing the extrapolation procedure to still more unknown values of the autocorrelation function simply involves the incorporation of these additional $r(n)$ into the autocorrelation matrix along with additional zeros appended to the two vectors to give the appropriate equation set that yields the desired solution. Since the prediction error filter coefficients remain unchanged by this extrapolation procedure, it follows that the spectral estimate given by (11.19) remains unchanged by the extrapolation as well.

11.4.2 Unknown Autocorrelation Function

The discussion of the previous section assumed that the first N lag values of the autocorrelation function were precisely known. Quite often, however, any information concerning the autocorrelation function is estimated from the time series data. One estimate of the autocorrelation function is the average given by

$$\hat{r}(\tau) = \frac{1}{N} \sum_{i=1}^{N-\tau} x_i x_i^* + \tau \tag{11.27}$$

With finite data sets, however, (11.27) implicitly assumes that any data outside the finite data interval are zero. The application of Fourier transform techniques to a finite data interval likewise assumes that any data that may exist outside the data interval are periodic with the known data. These unwarranted assumptions about unknown data represent "end effect" problems that may be avoided using the MEM approach, which makes no assumptions about any unmeasured data.

The MEM approach to the problem of spectral estimation when the autocorrelation function is unknown estimates the coefficients of a prediction error filter that never runs off the end of a finite data set, thereby making no assumptions about data outside the data interval. The prediction error filter coefficients are used to estimate the maximum entropy spectrum. This approach exploits the autocorrelation reflection–coefficient theorem, which

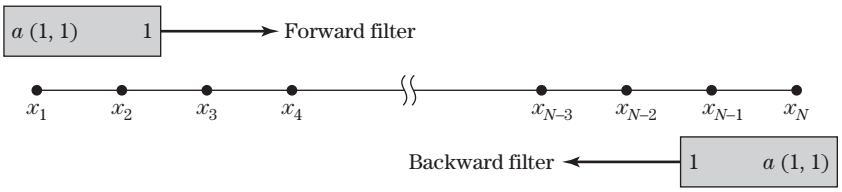


FIGURE 11-7 ■ Two-point prediction error filter operating forward and backward over an N-point data set.

states that there is a one-to-one correspondence between an autocorrelation function and the set of numbers $\{r(0), C_1, C_2, \ldots, C_N\}$ [16].

To obtain the set $\{r(0), C_1, C_2, \ldots, C_N\}$ first consider estimating $r(0)$ as the average square value of the data set, that is,

$$\hat{r}(0) = \frac{1}{N}\sum_{i=1}^{N} |x_i|^2 \tag{11.28}$$

Now consider how a two-point prediction error filter coefficient can be estimated from an N-point long data sample. The problem is to determine the two-point filter (having a first coefficient of unity) that has the minimum average power output where the filter is not run off the ends of the data sample. For a two-point filter running forward over the data sample as shown in Figure 11-7, the average power output is given by

$$P_1^f = \frac{1}{N-1}\sum_{i=1}^{N-1} |x_{i+1} + a(1,1)x_i|^2 \tag{11.29}$$

Since a prediction filter operates equally well running backward over a data set as well as forward, the average power output for a backward running two-point filter is given by

$$P_1^b = \frac{1}{N-1}\sum_{i=1}^{N-1} |x_i + a^*(1,1)x_{i+1}|^2 \tag{11.30}$$

Since there is no reason to prefer a forward-running filter over a backward-running filter and since (11.29) and (11.30) represent different estimates of the same quantity, averaging the two estimates should result in a better estimator than either one alone [and also guarantees that the estimate of the reflection coefficient $a(1, 1)$ is bounded by unity—a fact whose significance will be seen shortly] so that

$$\begin{aligned} P_1 &= \frac{1}{2}\left(P_1^f + P_1^b\right) \\ &= \frac{1}{2(N-1)}\left[\sum_{i=1}^{N-1} |x_{i+1} + a(1,1)x_i|^2 + \sum_{i=1}^{N-1} |x_i + a^*(1,1)x_{i+1}|^2\right] \end{aligned} \tag{11.31}$$

Now, minimize P_1 by selecting the coefficient $a(1, 1)$. Setting the derivative of P_1 with respect to $a(1, 1)$ equal to zero shows that the minimizing value of $a(1, 1)$ is given by [18]

$$a(1,1) = \frac{-2\sum_{i=1}^{N-1} x_i^* x_{i+1}}{\sum_{i=1}^{N-1} (|x_i|^2 + |x_{i+1}|^2)} \tag{11.32}$$

Having $\hat{r}(0)$ and $a(1, 1)$, we can find the remaining unknown parts of the prediction error filter matrix equation using (11.20), that is,

$$\begin{bmatrix} \hat{r}(0) & \hat{r}(-1) \\ \hat{r}(1) & \hat{r}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a(1, 1) \end{bmatrix} = \begin{bmatrix} P_1 \\ 0 \end{bmatrix} \tag{11.33}$$

so that

$$\hat{r}(1) = -a(1, 1)\hat{r}(0) \tag{11.34}$$

and the output power of the two-point filter is given by

$$P_1 = \hat{r}(0)[1 - |a(1, 1)|^2] \tag{11.35}$$

The result expressed by (11.35) implies that $|a(1, 1)| \leq 1$, which is also the necessary and sufficient condition that the filter defined by $\{1, a(1, 1)\}$ be a prediction error filter.

Next, consider how to obtain the coefficients for a three-point prediction error filter from the two-point filter just found. The prediction error filter matrix equation takes the form

$$\begin{bmatrix} \hat{r}(0) & \hat{r}(-1) & \hat{r}(-2) \\ \hat{r}(1) & \hat{r}(0) & \hat{r}(-1) \\ \hat{r}(2) & \hat{r}(1) & \hat{r}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a(2, 1) \\ a(2, 2) \end{bmatrix} = \begin{bmatrix} P_2 \\ 0 \\ 0 \end{bmatrix} \tag{11.36}$$

From the middle row it follows that

$$\hat{r}(1) + a(2, 1)\hat{r}(0) + a(2, 2)\hat{r}^*(1) = 0 \tag{11.37}$$

On substituting (11.34) into (11.37) it immediately follows that

$$a(2, 1) = a(1, 1) + a(2, 2)a^*(1, 1) \tag{11.38}$$

The corresponding output power P_2 is then

$$P_2 = P_1(1 - |a(2, 2)|^2) \tag{11.39}$$

Consequently, the coefficient vector for the three-point filter takes the form

$$\mathbf{a}_2^T = [1, a(1, 1) + a(2, 2)a^*(1, 1), a(2, 2)] \tag{11.40}$$

An equation for the estimated power output of a three-point filter can be written corresponding to (11.31) as $P_2 = \frac{1}{2}\left(P_2^f + P_2^b\right)$ where now

$$P_2^f = \frac{1}{N-2} \sum_{i=1}^{N-2} |x_{i+2} + a(2, 1)x_{i+1} + a(2, 2)x_i|^2 \tag{11.41}$$

$$P_2^b = \frac{1}{N-2} \sum_{i=1}^{N-2} |x_i + a^*(2, 1)x_{i+1} + a^*(2, 2)x_{i+2}|^2 \tag{11.42}$$

Since $a(2, 1)$ is given by (11.38) and $a(1, 1)$ is already known, it follows that the minimization of P_2 is carried out by varying only $a(2, 2) = C_2$. As was the case with the two-point filter, the magnitude of the three-point filter coefficient $a(2, 2)$ must not exceed unity. On minimizing of P_2, it turns out that $|a(2, 2)| \leq 1$, which is the necessary and sufficient condition that the filter defined by $\{1, a(1, 1) + a(2, 2)a^*(1, 1), a(2, 2)\}$ be a prediction

error filter. The corresponding Hermitian Toeplitz autocorrelation matrix is nonnegative definite as required by the basic autocorrelation function theorem.

The four-point filter can likewise be formed from the three-point filter by carrying out the minimization of P_3 with respect to $a(3, 3)$ only. The aforementioned procedure continues until all the coefficients $a(N, n), n = 1, \ldots, N$, are found, and (11.19) then yields the MEM spectral estimate. The general solution for the reflection coefficients $C_N = a(N, N)$ is given in [18]. Equations (11.38) and (11.39) may be expressed in general terms as

$$a(N, k) = a(N-1, k) + a(N, N)a^*(N-1, N-k) \tag{11.43}$$

$$P_N = P_{N-1}[1 - |a(N, N)|^2] \tag{11.44}$$

Since $|a(N, N)| \leq 1$, it follows that $0 \leq P_N \leq P_{N-1}$, so the error decreases with increasing filter order N. The choice of N is determined by the desired resolution of the estimated spectrum.

Burg [8] showed that the inverse of the covariance matrix $\mathbf{R}_{xx}$ is determined from the prediction error filter coefficients $a(m, k)$ and their corresponding error powers P_k for $k = 1, 2, \ldots, L$, where $L \leq N - 1$ is the filter length. The resulting MEM estimate of $\mathbf{R}_{xx}^{-1}$ differs from the inverse of the sample covariance matrix used in Chapter 6, and some evidence suggests that the MEM estimate $\hat{\mathbf{R}}_{xx}^{-1}$ converges to $\mathbf{R}_{xx}^{-1}$ more rapidly than the direct matrix inversion (DMI) estimate. This fast convergence feature makes the Burg algorithm particularly attractive in situations where only a small number of independent data samples are available.

11.4.3 Extension to Multichannel Spectral Estimation

Since the processing of array sensor outputs involves multichannel complex signals, it is important to generalize the single-channel Burg process to multiple channels to exploit the MEM procedure for adaptive array applications. Several multichannel generalizations of the scalar MEM approach have been proposed [20–26], and the generalization provided by Strand [27,28] is outlined here.

Suppose there are N observations of the p-channel vector $\mathbf{x}(t)$ denoted by $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}\}$. A linear prediction of $\mathbf{x}_N$ based on the previous observations of $\mathbf{x}(t)$ can be obtained as

$$\hat{\mathbf{x}}_N = -\sum_{i=1}^{N} \mathbf{A}^\dagger(N, i)\mathbf{x}_{N-i} \tag{11.45}$$

where $\mathbf{A}(N, i)$ now denotes the matrix of N-long forward prediction filter coefficients. The error associated with $\hat{\mathbf{x}}_N$ is then given by

$$\boldsymbol{\varepsilon}_N = \mathbf{x}_N - \hat{\mathbf{x}}_N = \mathbf{x}_N + \sum_{i=1}^{N} \mathbf{A}^\dagger(N, i)\mathbf{x}_{N-i} = \mathbf{A}_N^\dagger \mathbf{x}_f \tag{11.46}$$

where

$$\mathbf{A}_N^\dagger = [\mathbf{I}, \mathbf{A}^\dagger(N, 1), \mathbf{A}^\dagger(N, 2), \ldots, \mathbf{A}^\dagger(N, N)]$$

$$\mathbf{x}_f^T = \left[\mathbf{x}_N^T, \mathbf{x}_{N-1}^T, \ldots, \mathbf{x}_0^T\right]$$

A prediction error filter running backward over a data set in general will not have the same set of matrix prediction filter coefficients as the forward-running prediction filter, so the error vector associated with a backward-running prediction filter is denoted by

$$\mathbf{b}_N = \mathbf{x}_0 - \hat{\mathbf{x}}_0 = \mathbf{x}_0 + \sum_{i=1}^{N} \mathbf{B}^{\dagger}(N, i)\mathbf{x}_i = \mathbf{B}_N^{\dagger}\mathbf{x}_b \tag{11.47}$$

The fact that $\mathbf{A}(N, i) \neq \mathbf{B}^{\dagger}(N, i)$ reflects the fact that the multichannel backward prediction error filter is not just the complex conjugate time reverse of the multichannel forward prediction error filter (as it was in the scalar case).

The matrix generalization of (11.20) is given by

$$\begin{aligned}\mathbf{R}^f\mathbf{A}_N &= \begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(-1) & \dots & \mathbf{R}(-N) \\ \mathbf{R}(1) & \mathbf{R}(0) & \dots & \mathbf{R}(-N+1) \\ \vdots & & & \\ \mathbf{R}(N) & \mathbf{R}(N-1) & \dots & \mathbf{R}(0) \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A}(N,1) \\ \vdots \\ \mathbf{A}(N,N) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}_N^f \\ 0 \\ \vdots \\ 0 \end{bmatrix}\end{aligned} \tag{11.48}$$

where the $p \times p$ block submatrices $\mathbf{R}(k)$ are defined by

$$\mathbf{R}(k) = E\{\mathbf{x}(t)\mathbf{x}^{\dagger}(t - k\Delta t)\}, \qquad k = 0, 1, \dots, N \tag{11.49}$$

so that

$$\mathbf{R}(-k) = \mathbf{R}^{\dagger}(k) \tag{11.50}$$

The forward power matrix $\mathbf{P}_N^f$ for the prediction error filter satisfying (11.48) is given by

$$\mathbf{P}_N^f = E\{\boldsymbol{\varepsilon}_N\boldsymbol{\varepsilon}_N^{\dagger}\} = \mathbf{A}_N^{\dagger}\mathbf{R}^f\mathbf{A}_N \tag{11.51}$$

The optimum backward prediction error filter likewise satisfies

$$\begin{aligned}\mathbf{R}^b\mathbf{B}_N &= \begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(1) & \dots & \mathbf{R}(N) \\ \mathbf{R}(-1) & \mathbf{R}(0) & \dots & \mathbf{R}(N-1) \\ \vdots & & & \\ \mathbf{R}(-N) & \mathbf{R}(-N+1) & \dots & \mathbf{R}(0) \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{B}(N,1) \\ \vdots \\ \mathbf{B}(N,N) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}_N^b \\ 0 \\ \vdots \\ 0 \end{bmatrix}\end{aligned} \tag{11.52}$$

The backward power matrix $\mathbf{P}_N^b$ for the prediction error filter satisfying (11.52) is then

$$\mathbf{P}_N^b = E\{\mathbf{b}_N\mathbf{b}_N^{\dagger}\} = \mathbf{B}_N^{\dagger}\mathbf{R}^b\mathbf{B}_N \tag{11.53}$$

The matrix coefficients $\mathbf{A}(N, N)$ and $\mathbf{B}(N, N)$ are referred to as the forward and backward reflection coefficients, respectively, as follows:

$$\mathbf{C}_N^f = \mathbf{A}(N, N) \qquad \text{and} \qquad \mathbf{C}_N^b = \mathbf{B}(N, N) \tag{11.54}$$

The maximum entropy power spectral density matrix can be computed either in terms of the forward filter coefficients using

$$\mathbf{\Phi}_{xx}(f) = \Delta t \left[\mathbf{A}^{-1}\left(\frac{1}{z}\right)\right]^{\dagger} \mathbf{P}_N^f \left[\mathbf{A}^{-1}\left(\frac{1}{z}\right)\right] \tag{11.55}$$

where

$$\mathbf{A}(z) = \mathbf{I} + \mathbf{A}(N, 1)z + \cdots + \mathbf{A}(N, N)z^N \tag{11.56}$$

and $z \triangleq e^{-j2\pi f \Delta t}$, or in terms of the backward filter coefficients using

$$\mathbf{\Phi}_{xx}(f) = \Delta t[\mathbf{B}^{-1}(z)]^{\dagger}\mathbf{P}_N^b[\mathbf{B}^{-1}(z)] \tag{11.57}$$

where

$$\mathbf{B}(z) = \mathbf{I} + \mathbf{B}(N, 1)z + \cdots + \mathbf{B}(N, N)z^N \tag{11.58}$$

The forward and backward reflection coefficients are furthermore related by

$$\mathbf{C}_N^f = \left(\mathbf{P}_{N-1}^b\right)^{-1}\mathbf{C}_N^{b\dagger}\mathbf{P}_{N-1}^f \qquad \text{or} \qquad \mathbf{C}_N^b = \left(\mathbf{P}_{N-1}^f\right)^{-1}\mathbf{C}_N^{f\dagger}\mathbf{P}_{N-1}^b \tag{11.59}$$

If an observation process has collected N_d consecutive vector samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_d}\}$, then a prediction error filter of length N will have $M = N_d - N$ consecutive $(N+1)$-tuples from the data set to operate on. It is convenient to array each $(N+1)$-tuple of data as an extended vector defined by

$$\mathbf{x}_m^N \triangleq \begin{bmatrix} \mathbf{x}_{m+N} \\ \mathbf{x}_{m+N-1} \\ \vdots \\ \mathbf{x}_{m+1} \\ \mathbf{x}_m \end{bmatrix}, \qquad \begin{matrix} N = 0, 1, \ldots, N_d - 1 \\ m = 1, 2, \ldots, M = N_d - N \end{matrix} \tag{11.60}$$

The residual outputs of the forward error filter (denoted by $\mathbf{u}_m$) and the backward error filter (denoted by $\mathbf{v}_m$) when these filters are applied to the mth $(N+1)$-tuple of data are given by

$$\mathbf{u}_m = \left[\mathbf{A}_{N-1}^{\dagger} \vdots 0\right]\mathbf{x}_m^N + \mathbf{C}_N^{f\dagger}\left[0 \vdots \mathbf{B}_{N-1}^{\dagger}\right]\mathbf{x}_m^N = \boldsymbol{\varepsilon}_m^N + \mathbf{C}_N^{f\dagger}\mathbf{b}_m^N \tag{11.61}$$

$$\mathbf{v}_m = \mathbf{C}_N^{b\dagger}\left[\mathbf{A}_{N-1}^{\dagger} \vdots 0\right]\mathbf{x}_m^N + \left[0 \vdots \mathbf{B}_{N-1}^{\dagger}\right]\mathbf{x}_m^N = \mathbf{C}_N^{b\dagger}\boldsymbol{\varepsilon}_m^N + \mathbf{b}_m^N \tag{11.62}$$

Using (11.59), however, we can rewrite (11.62) as

$$\mathbf{v}_m = \mathbf{P}_{N-1}^b\mathbf{C}_N^f\mathbf{P}_{N-1}^{-1}\boldsymbol{\varepsilon}_m^N + \mathbf{b}_m^N \tag{11.63}$$

Equations (11.61) and (11.63) show that the forward and backward prediction error filter residual outputs depend only on the forward reflection coefficient $\mathbf{C}_N^f$. The coefficient $\mathbf{C}_N^f$ are chosen to minimize a weighted sum of squares of the forward and backward residual

outputs of the filter of length N; that is, minimize $SS(\mathbf{C}_N^f)$ where

$$SS(\mathbf{C}_N^f) \triangleq \frac{1}{2}\sum_{m=1}^{M} K_m \left[\mathbf{u}_m^\dagger \left(\mathbf{P}_{N-1}^f\right)^{-1}\mathbf{u}_m + \mathbf{v}_m^\dagger \left(\mathbf{P}_{N-1}^b\right)^{-1}\mathbf{v}_m\right] \tag{11.64}$$

and K_m is a positive scalar weight $= 1/M$. The equation that yields the optimum value of $\mathbf{C}_N^f$ for (11.64) is then

$$\mathbf{B}\mathbf{C}_N^f + \mathbf{P}_{N-1}^b \mathbf{C}_N^f \left(\mathbf{P}_{N-1}^f\right)^{-1}\mathbf{E} = -2\mathbf{G} \tag{11.65}$$

where

$$\mathbf{B} \triangleq \sum_{m=1}^{M} K_m \mathbf{b}_m^N \mathbf{b}_m^{N\dagger} \tag{11.66}$$

$$\mathbf{E} \triangleq \sum_{m=1}^{M} K_m \boldsymbol{\varepsilon}_m^N \boldsymbol{\varepsilon}_m^{N\dagger} \tag{11.67}$$

$$\mathbf{G} \triangleq \sum_{m=1}^{M} K_m \mathbf{b}_m^N \boldsymbol{\varepsilon}_m^{N\dagger} \tag{11.68}$$

After obtaining $\mathbf{C}_N^f$ from (11.65) (which is a matrix equation of the form $\mathbf{AX} + \mathbf{XB} = \mathbf{C}$) then $\mathbf{C}_N^b$ can be obtained from (11.59), and the desired spectral matrix estimate can be computed from (11.55) or (11.57).

11.5 COMPARING DOA ALGORITHM SPECTRA

The various high-resolution DOA approaches discussed so far have much better resolution than a periodogram. Figure 11-8 compares the DOA algorithms when the three signals incident on the 12-element uniform array at $\theta = -50^\circ$, 10°, and 20° have relative signals

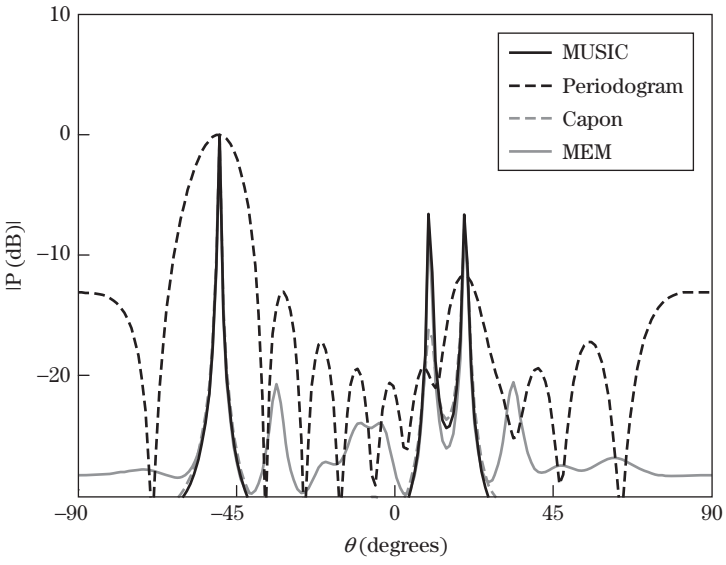


FIGURE 11-8 ■ Comparison of the DOA algorithms when the three signals incident on the 12-element uniform array at $\theta = -50^\circ$, 10°, and 20° have relative signals strengths of 16, 1, and 4, respectively.

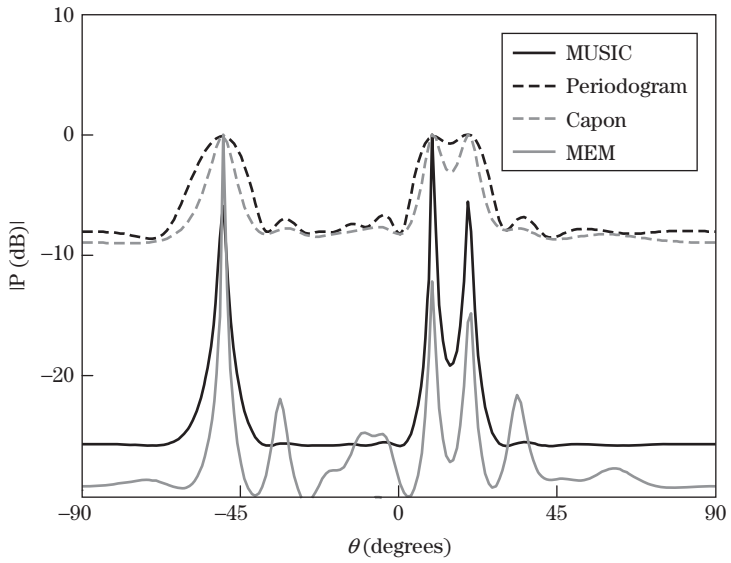


FIGURE 11-9 ■ Comparing four DOA techniques when the standard deviation of the noise is 2.

strengths of 16, 1, and 4, respectively. The lowest amplitude signal is almost lost in the periodogram and is off by several degrees from the true signal position.

Increasing the noise variance causes the noise floor of the periodogram and Capon spectrum to rise as shown in Figure 11-9. The Capon spectrum barely distinguishes between the two closely spaced signals, and the peaks no longer accurately reflect the signal strengths. The noise floor of the MUSIC spectrum goes up, but it has stronger peaks than the MEM spectrum.

11.6 | SEQUENTIAL IMPLEMENTATION OF BAYES OPTIMAL ARRAY PROCESSOR

It was noted in Chapter 3 that when the overall goal of good signal detection is the principal concern then array processors based on the likelihood ratio are optimum in the sense of minimizing the risk associated with an incorrect decision concerning signal presence or absence. Furthermore, when the signal to be detected has one or several uncertain parameters (due to, e.g., location uncertainty) then the sufficient test statistic for decision can be obtained from the ratio of marginal probability density functions in the following manner.

Let the observation vector $\mathbf{x}$ consist of the observed random processes appearing at the array elements (or the Fourier coefficients representing the random processes). If any uncertain signal parameters are present, model these parameters as additional random variables, and summarize any prior knowledge about them with an a priori probability density function $p(\boldsymbol{\theta})$. The likelihood ratio is written as the ratio of the following marginal probability density functions:

$$\Lambda(\mathbf{x}) = \frac{\int_{\Theta} p(\mathbf{x}/\boldsymbol{\theta}, \text{signal present})\, p(\boldsymbol{\theta}) d\boldsymbol{\theta}}{p(\mathbf{x}/\text{signal absent})} \tag{11.69}$$

where $\boldsymbol{\theta} \in \Theta$. Equation (11.69) can be implemented by means of a suboptimal "one-shot array processor" [29] for which a block diagram is given in Figure 11-10.

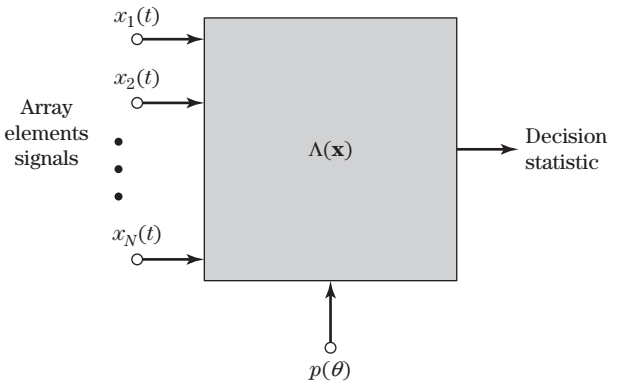


FIGURE 11-10 ■ Block diagram for one-shot array processor.

11.6.1 Estimate and Plug Array Processor

An intuitively appealing approach to the array detection problem when uncertain signal parameters exist is to directly estimate these parameters and plug them into the conditional likelihood ratio as though they were exactly known. A block diagram of such an estimate and plug array processor is given in Figure 11-11. Quite naturally, the merit of any estimate and plug structure depends on the accuracy of the estimates generated by the parameter estimation scheme, and the question of the processor's sensitivity to mismatch between the true parameter values and those assumed (or estimated) has received some attention [30,31].

Two popular approaches to obtaining estimates of random signal parameters for use in an estimate and plug array processor are the ML approach and the Bayes optimum approach. In the ML approach, the maximum likelihood estimate (MLE) of the unknown signal parameters is formed by solving

$$\left.\frac{\partial p(\mathbf{x}/\boldsymbol{\theta}, \text{ signal present})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MLE}}} = 0 \tag{11.70}$$

and then using the resulting estimate in the likelihood ratio test statistic as if it were known exactly. The Bayes approach to the parameter estimation problem incorporates any a priori knowledge concerning the signal parameters in the form of a probability density function

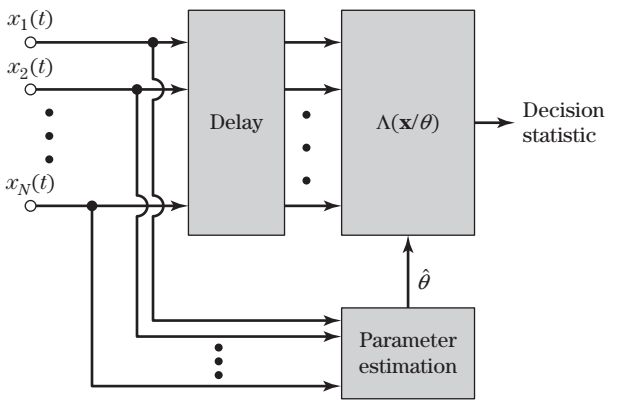


FIGURE 11-11 ■ Block diagram of estimate and plug array processor.

$p(\boldsymbol{\theta}/\text{signal present})$. To obtain a signal parameter estimate for an estimate and plug array processor, note that

$$\frac{\Lambda(\mathbf{x})}{\Lambda(\mathbf{x}/\hat{\boldsymbol{\theta}})} = \frac{\int_{\Theta} p(\mathbf{x}/\boldsymbol{\theta}, \text{signal present})\, p(\boldsymbol{\theta}/\text{signal present}) d\boldsymbol{\theta}}{p(\mathbf{x}/\hat{\boldsymbol{\theta}}, \text{signal present})} \tag{11.71}$$

The optimal Bayes processor explicitly incorporates a priori knowledge about the unknown signal parameters $\boldsymbol{\theta}$ into the likelihood ratio $\Lambda(\mathbf{x})$ through the averaging process expressed by the numerator of (11.71). To obtain an estimate $\hat{\boldsymbol{\theta}}$ to use in a suboptimal estimate and plug structure, require

$$\frac{\Lambda(\mathbf{x})}{\Lambda(\mathbf{x}/\hat{\boldsymbol{\theta}})} = 1 \tag{11.72}$$

Having evaluated $\Lambda(\mathbf{x})$ using the averaging process, we find $\hat{\boldsymbol{\theta}}$ as the solution to (11.72), and it is referred to as a "pseudo-estimate" $\hat{\boldsymbol{\theta}}_{\text{PSE}}$ [32]. The performance of the Bayes optimal processor for the case of a signal known except for direction (SKED) was investigated by Gallop and Nolte [33]. A comparison between the two estimate and plug structures obtained with a MLE and a Bayes pseudo-estimate is given in [34] for the case of target location unknown. The results indicated that the ML detector performs the same as the Bayes pseudo-estimate detector when the a priori knowledge about the uncertain parameter is uniformly distributed. When the a priori knowledge available is more precise, however, the performance of the Bayes pseudo-estimate detector improves whereas that of the ML detector does not, and this difference between the two processors becomes more pronounced as the array size becomes larger.

11.6.2 Sequential Optimal Array Processor

When implemented in the form of an estimate and plug structure, the Bayes optimal array processor processes all the observed data at the same time. By implementing the same processor sequentially, the resulting array processor will exhibit adaptive (or learning) features naturally.

To see how to implement a Bayes optimal processor sequentially, let $\mathbf{x}^i$ denote the vector of observed outputs (or the Fourier coefficients thereof) from the array elements for the ith sample period. The sequence $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L\}$ then represents L different observation vector samples. The joint probability density function of the observed vector sequence can be written as

$$p(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L) = \prod_{i=1}^{L} p(\mathbf{x}^i/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1) \tag{11.73}$$

When unknown signal parameters are present, application of the averaging process described in the previous section to $p(\boldsymbol{\theta})$ yields

$$P(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L) = \int_{\Theta} \prod_{i=1}^{L} p(\mathbf{x}^i/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1, \boldsymbol{\theta})\, p(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{11.74}$$

Now assume parameter conditional independence of the various $\mathbf{x}^i$ so that

$$p(\mathbf{x}^i/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1, \boldsymbol{\theta}) = p(\mathbf{x}^i/\boldsymbol{\theta}) \tag{11.75}$$

and then it follows that (11.74) can be rewritten as

$$\mathbf{P}(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L) = \int_\Theta \prod_{i=1}^{L} p(\mathbf{x}^i/\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{11.76}$$

According to Bayes's rule it follows that

$$p(\boldsymbol{\theta}/\mathbf{x}^1) = \frac{p(\mathbf{x}^1/\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{x}^1)} \tag{11.77}$$

$$p(\theta/\mathbf{x}^1, \mathbf{x}^2) = \frac{p(\mathbf{x}^2/\boldsymbol{\theta})\, p(\boldsymbol{\theta}/\mathbf{x}^1)}{p(\mathbf{x}^2/\mathbf{x}^1)} \tag{11.78}$$

$$\vdots$$

$$p(\boldsymbol{\theta}/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1) = \frac{p(\mathbf{x}^{i-1}/\boldsymbol{\theta})\, p(\boldsymbol{\theta}/\mathbf{x}^{i-2}, \ldots, \mathbf{x}^1)}{p(\mathbf{x}^{i-1}/\mathbf{x}^{i-2}, \ldots, \mathbf{x}^1)} \tag{11.79}$$

so that

$$p(\mathbf{x}^1, \ldots, \mathbf{x}^L) = \prod_{i=1}^{L} \int_\Theta p(\mathbf{x}^i/\boldsymbol{\theta})\, p(\boldsymbol{\theta}/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1)\, d\boldsymbol{\theta} \tag{11.80}$$

where $p(\boldsymbol{\theta}/\mathbf{x}^{i-1}, \ldots, \mathbf{x}^1)$ represents the updated form of the a priori information contained in $p(\boldsymbol{\theta})$.

A sequential processor may now be implemented using (11.79) and (11.80) to form the marginal density functions required in the likelihood ratio as follows:

$$\Lambda(\mathbf{x}^1, \ldots, \mathbf{x}^L) = \frac{p(\mathbf{x}^1, \ldots, \mathbf{x}^L/\text{signal present})}{p(\mathbf{x}^1, \ldots, \mathbf{x}^L/\text{signal absent})} \tag{11.81}$$

A block diagram of the resulting sequential array processor based on (11.79)–(11.81) is given in Figure 11-12. The sequential Bayesian updating of $p(\boldsymbol{\theta})$ represented by (11.79) results in an optimal processor having an adaptive capability.

Performance results using an optimal sequential array processor were reported in [29] for a detection problem involving a signal known exactly imbedded in Gaussian noise where the noise has an additive component arising from a noise source with unknown direction. The adaptive processor in this problem must both succeed in detecting the presence or absence of the desired signal and in "learning" the actual direction of the

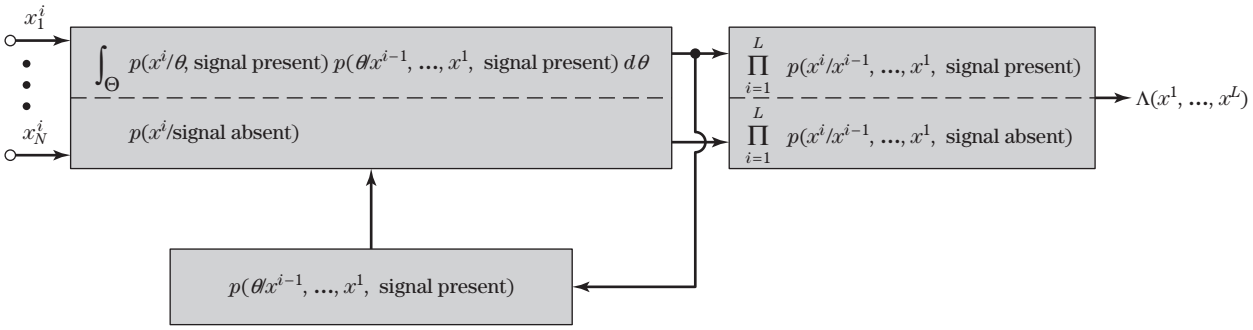


FIGURE 11-12 ■ Bayes optimal sequential array processor.

noisy signal source. The results obtained indicated that even though the directional noise to thermal noise ratio was relatively low, the optimal sequential processor could nevertheless determine the directional noise source's location.

11.7 | PARAMETER ESTIMATION VIA SUBSPACE FITTING

The estimation of signal parameters via rotational invariance techniques (ESPRIT) [35–37] is a DOA estimation exploiting a known array structure. The method described here is a total least square (TLS) version of the ESPRIT approach to the problem, and extensions of this method may readily be seen when the problem formulation is cast in a subspace fitting framework [38]. It has been found that this approach provides estimates for which the parameter estimate variance coincides with the Cramer–Rao bound.

The output of the *i*th sensor of an array can be represented by

$$x_i(t) = \sum_{j=1}^{d} a_i(\theta_j)s_j(t) + n_i(t), i = 1, \ldots, k \tag{11.82}$$

where $a_i(\theta_j)$ is a complex scalar representing the sensor response to the *j*th emitter signal (there are a total of "*d*" signals). The *j*th emitter signal is denoted by $s_j(t)$, and the additive noise is $n_i(t)$. In matrix notation, (11.82) can be written as

$$\mathbf{x}(t) = [\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_d)]\mathbf{s}(t) + \mathbf{n}(t) = \mathbf{A}(\boldsymbol{\theta}_{()})\mathbf{s}(t) + \mathbf{n}(t) \tag{11.83}$$

where $\boldsymbol{\theta}_{()}$ is a d-dimensional parameter vector corresponding to the actual DOAs, and $\mathbf{A}(\theta_{()})$ is a $k \times d$ response matrix where the vector $\mathbf{a}(\theta_j) = [a_1(\theta_j) \ldots a_k(\theta_j)]^T$ contains the sensor responses to a unit wavefront from the direction θ_j. The array output is sampled at N time instances, and this collection of samples are arranged in the columns of a $k \times N$ data matrix, $\mathbf{X}_N$, given by

$$\mathbf{X}_N = [\mathbf{x}(1), \ldots, \mathbf{x}(N)] = \mathbf{A}(\boldsymbol{\theta}_{()})\mathbf{s}_N + \mathbf{N}_N. \tag{11.84}$$

The emitter covariance matrix is give by $\mathbf{S} = E[\mathbf{s}(t)\mathbf{s}^H(t)]$, where "E" denotes the expected value. Likewise, the output signal covariance matrix is given by

$$\mathbf{R}_{xx} = E[\mathbf{x}(t)\mathbf{x}^H(t)] = \mathbf{A}(\boldsymbol{\theta}_{()})\mathbf{S}\mathbf{A}(\boldsymbol{\theta}_{()})^H + \sigma^2\mathbf{I} \tag{11.85}$$

The eigendecomposition of $\mathbf{R}_{xx}$ is given by

$$\mathbf{R}_{xx} = \mathbf{E}_s\boldsymbol{\Lambda}_s\mathbf{E}_s^H + \mathbf{E}_n\boldsymbol{\Lambda}_n\mathbf{E}_n^H \tag{11.86}$$

where $\lambda_1 > \cdots > \lambda_d > \lambda_{d+1} = \cdots = \lambda_k = \sigma^2$. The matrix $\mathbf{E}_s = [\mathbf{e}_1, \ldots, \mathbf{e}_d]$ contains the d eigenvectors corresponding to the distinct signals. The range space of $\mathbf{E}_s$ is called the *signal subspace*, and its orthogonal complement is the *noise subspace*, which is spanned by the columns of $\mathbf{E}_n = [\mathbf{e}_{d+1}, \ldots, \mathbf{e}_k]$. The eigendecomposition of the sample covariance matrix estimate of $\mathbf{R}_{xx}$ then corresponds to (11.86) with estimates for each of the matrices replacing the actual matrix values.

Equations (11.84)–(11.86) show that the parameter estimation problem is regarded as a subspace fitting problem in which the subspace spanned by $\mathbf{A}(\boldsymbol{\theta})$ is fitted to the

measurements $\mathbf{X}_N$ in a least squares sense. The basic subspace fitting problem is defined by

$$\hat{\mathbf{A}}, \hat{\mathbf{T}} = \arg \min_{\mathbf{A}, \mathbf{T}} \|\mathbf{M} - \mathbf{AT}\|_F^2 \tag{11.87}$$

where $\|\mathbf{A}\|_F^2 = \text{trace}(\mathbf{A}^H\mathbf{A})$ is the Frobenius norm, and $\hat{\theta} = \arg \min_{\theta} \mathbf{V}(\boldsymbol{\theta})$ is the minimizing argument of $\mathbf{V}(\boldsymbol{\theta})$ The $k \times q$ matrix $\mathbf{M}$ in (11.87) represents the data, whereas $\mathbf{T}$ is any $p \times q$ matrix. For a fixed $\mathbf{A}$, the minimum with respect to $\mathbf{T}$ is a measure of how well the range spaces of $\mathbf{A}$ and $\mathbf{M}$ match. The subspace fitting estimate selects $\mathbf{A}$ so these subspaces are as close as possible. The estimate of $\boldsymbol{\theta}$ is then obtained from the parameters of $\hat{\mathbf{A}}$. It is of some practical interest to note that the subspace fitting problem is separable in $\mathbf{A}$ and $\mathbf{T}$. By substituting the pseudo-inverse solution $\hat{\mathbf{T}} = \mathbf{A}^p\mathbf{M}$ back into (11.87) where $\mathbf{A}^p = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$, one obtains the following equivalent problem

$$\hat{\mathbf{A}} = \arg \max_{A} \text{tr}\{\mathbf{P}_A\mathbf{M}\mathbf{M}^H\} \tag{11.88}$$

where $\mathbf{P}_A = \mathbf{A}\mathbf{A}^p$ is a projection matrix that projects onto the column space of $\mathbf{A}$. The subspace fitting problem then resolves into a familiar parameter optimization problem described by

$$\hat{\theta} = \arg \max_{\theta} \text{tr}\{\mathbf{P}_{\boldsymbol{A}}(\boldsymbol{\theta})\hat{\mathbf{R}}_{xx}\} \tag{11.89}$$

where $\hat{\mathbf{R}}_{xx}$ is the sample covariance matrix. Notice from (11.88) that the same result could be obtained by simply taking $\hat{\mathbf{R}}_{xx} = \mathbf{M}\mathbf{M}^H$. This approach is the deterministic maximum likelihood method for obtaining direction-of-arrival estimates.

The ESPRIT algorithm assumes that the array is composed of two identical subarrays, each having $k/2$ elements (so the total number of array elements is even). The subarrays are displaced from each other by a known displacement vector so the propagation between the subarrays can be described by the diagonal matrix $\boldsymbol{\Phi} = \text{diag}[e^{j\omega\tau_1}\ e^{j\omega\tau_2}\ \ldots\ e^{j\omega\tau_e}]$, where τ_i is the time delay in the propagation of the ith emitter signal between the two subarrays, and ω is the center frequency of the emitters. The time delay is then related to the angle of arrival by $\tau_i = |\Delta| \sin\theta_i / c$, where c is the speed of propagation, and Δ is the displacement vector between the two subarrays. The output of the array is then modeled as

$$\mathbf{x}(t) = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{\Gamma}\boldsymbol{\Phi} \end{bmatrix} s(t) + \begin{bmatrix} n_1(t) \\ n_2(t) \end{bmatrix} \tag{11.90}$$

where the $k/2 \times d$ matrix $\boldsymbol{\Gamma}$ contains the common array manifold vectors of the two subarrays, and $\boldsymbol{\Phi}$ is the diagonal matrix discussed already. Since the matrices defined by $[\mathbf{E}_1^T\mathbf{E}_2^T]^T$ (eigenvectors for the two subarrays) and $[\boldsymbol{\Gamma}^T\boldsymbol{\Phi}^T\boldsymbol{\Gamma}^T]$ have the same range space, there is a full rank $d \times d$ matrix $\mathbf{T}$ such that

$$\begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{\Gamma}\boldsymbol{\Phi} \end{bmatrix} \mathbf{T} \tag{11.91}$$

Eliminating $\boldsymbol{\Gamma}$ in (12.91) then yields

$$\mathbf{E}_2 = \mathbf{E}_1\mathbf{T}^{-1}\boldsymbol{\Phi}\mathbf{T} = \mathbf{E}_1\boldsymbol{\Psi} \tag{11.92}$$

Using the eigendecomposition

$$\begin{bmatrix} \hat{\mathbf{E}}_1^H \\ \hat{\mathbf{E}}_2^H \end{bmatrix} \begin{bmatrix} \hat{\mathbf{E}}_1 & \hat{\mathbf{E}}_2 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \mathbf{L} \begin{bmatrix} V_{11}^H & V_{21}^H \\ V_{12}^H & V_{22}^H \end{bmatrix} \tag{11.93}$$

where $\mathbf{L} = \text{diag}[l_1, l_2, \ldots, l_{2d}]$. Since $\boldsymbol{\Psi} = \mathbf{T}^{-1}\boldsymbol{\Phi}\mathbf{T}$, the elements of $\boldsymbol{\Phi}$ are estimated by the eigenvalues of $\hat{\Psi}_{TLS}$. The principal angles of these eigenvalues give estimates of the time delays τ_i, which in turn give the DOA estimates. Solving for $\boldsymbol{\Psi}$ and equating its eigenvalues to $\boldsymbol{\Phi}$ results in the signal angle estimates of

$$\theta_m = \sin^{-1}\left(\frac{\arg(\lambda_m^{\Psi})}{kd}\right) \tag{11.94}$$

where λ_m^{Ψ} = eigenvalues of $\boldsymbol{\Psi}$.

$$V_s = \begin{bmatrix} -0.0816 + \text{j}0.0000 & -0.3214 - \text{j}0.0000 & 0.3239 - \text{j}0.0000 \\ -0.0214 + \text{j}0.1413 & 0.1728 - \text{j}0.1651 & 0.2693 - \text{j}0.3461 \\ 0.0775 + \text{j}0.2593 & -0.0428 + \text{j}0.3203 & -0.0252 - \text{j}0.2482 \\ 0.3061 + \text{j}0.0678 & -0.1822 - \text{j}0.1796 & -0.2547 - \text{j}0.1985 \\ 0.3649 - \text{j}0.1551 & 0.2694 + \text{j}0.1027 & -0.0855 + \text{j}0.0202 \\ 0.0744 - \text{j}0.3899 & -0.2220 + \text{j}0.1963 & -0.0774 + \text{j}0.0945 \\ -0.1733 - \text{j}0.3619 & 0.1385 - \text{j}0.2273 & -0.0089 - \text{j}0.1292 \\ -0.3708 - \text{j}0.0374 & 0.1865 + \text{j}0.2484 & -0.0916 - \text{j}0.0330 \\ -0.2907 + \text{j}0.1344 & -0.2014 - \text{j}0.1534 & -0.3170 + \text{j}0.0412 \\ -0.0037 + \text{j}0.2384 & 0.2845 - \text{j}0.1651 & -0.0919 + \text{j}0.2447 \\ 0.0553 + \text{j}0.1390 & -0.1668 + \text{j}0.1730 & 0.0681 + \text{j}0.4281 \\ 0.0542 - \text{j}0.0455 & -0.1122 - \text{j}0.2981 & 0.3375 + \text{j}0.1231 \end{bmatrix}$$

Next, compute $\boldsymbol{\Psi}$ to get

$$\boldsymbol{\Psi} = \begin{bmatrix} 0.6322 - \text{j}0.6792 & -0.1035 + \text{j}0.2722 & 0.2247 - \text{j}0.0474 \\ -0.2434 - \text{j}0.0108 & -0.6666 + \text{j}0.5904 & -0.3027 + \text{j}0.3979 \\ -0.1525 + \text{j}0.2374 & -0.2235 + \text{j}0.1595 & 0.6221 - \text{j}0.6479 \end{bmatrix}$$

The eigenvalues of ψ are found and substituted into (11.94) to find an estimate of the angle of arrival.

$$\theta_m = \begin{bmatrix} -50.15^\circ & 9.99^\circ & 20.02^\circ \end{bmatrix}$$

11.8 THE MAXIMUM LIKELIHOOD ESTIMATE

The ML estimator has importance in estimation theory because it provides signal parameter estimates that are in some sense optimal; that is, under certain conditions, it provides the "best" (most accurate) estimate of a particular signal parameter. Examples of parameters of interest include signal AOA, amplitude, phase, frequency, and time delay of arrival. The following treatment is developed for AOA, although an estimate for the signal intensity (received signal amplitude) is a by-product.

The precise form of the ML estimate depends on certain assumptions regarding the signal model. Two signal models are considered: (1) the first model assumes the desired

signals to be Gaussian zero mean ("stochastic" signal model); and (2) the second model assumes that signals are "deterministic" but unknown. We start with the stochastic model.

11.8.1 Mathematical Preliminaries

The maximum likelihood estimator and the derivation of the Cramer–Rao bound are based on the Gaussian probability density function. First, the notion of the Gaussian density function for the stochastic signal model is established.

Let $\underline{x}_k$ be an N-dimensional complex vector representing the kth data sample of the signals received by an N-element antenna array. The sample consists of a desired signal component $\underline{s}_i$ and a noise component $\underline{n}_i$ so that

$$\underline{x}_k = \underline{s}_k + \underline{n}_k \tag{11.95}$$

The noise $\underline{n}_k$ is assumed to be a sample function from a zero-mean N-variate complex Gaussian process, with full rank covariance matrix

$$R_{nn} = E\left\{\underline{n}_k \underline{n}_k^{\dagger}\right\} \tag{11.96}$$

For the stochastic signal model, the desired signal $\underline{s}_k$ is also assumed to be a sample function from a zero-mean N-variate complex Gaussian process, with covariance

$$R_{ss} = E\left\{\underline{s}_k \underline{s}_k^{\dagger}\right\} \tag{11.97}$$

Under these assumptions, the probability density for the data sample $\underline{x}_k$ is given by [7]

$$p(\underline{x}_k) = (\pi)^{-N} |R|^{-1} \exp\left(-\underline{x}_k^{\dagger} R^{-1} \underline{x}_k\right) \tag{11.98}$$

where R is the $N \times N$ covariance matrix

$$R = E\{\underline{x}_k \underline{x}_k^{\dagger}\} \tag{11.99}$$

It is assumed that the desired signal and noise are uncorrelated, so that

$$R = R_{ss} + R_{nn} \tag{11.100}$$

Our objective is to estimate the angle of arrival of the signal $\underline{s}_k$ in the presence of internal and external interference denoted by $\underline{n}_k$, based on K independent data samples ("snapshots") of $\underline{x}_k$, $k = 1, \ldots, K$. The ML estimation procedure is based on determining the value of the unknown parameters (parameters to be estimated) that maximizes the conditional joint density function of the K independent data samples, which from (11.97) has the form

$$p(\underline{x}_1, \underline{x}_2, \ldots \underline{x}_K \,|\theta, S) = (\pi)^{-NK} |R|^{-K} \exp\left[-\sum_{k=1}^{K} \underline{x}_k^{\dagger} R^{-1} \underline{x}_k\right] \tag{11.101}$$

The density in (11.100) is conditional on the values of the unknown signal parameters to be estimated, namely, the AOA (θ) and the signal intensity (S). To maximize (11.100) with respect to θ and S, Equation (11.100) must be reformulated to show explicit dependence on these variables. For narrowband signals, where the element-to-element time delay experienced by the desired signal can be represented as phase shifts of the signal, this

equation reduces to

$$P(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_K \,|\theta, S) = C \left[\frac{1}{1+SB}\right]^K \exp\left[\frac{S}{1+SB}\sum_{k=1}^{K} \underline{x}_k^\dagger R_{nn}^{-1} \underline{d}(\theta)\underline{d}^\dagger(\theta) R_{nn}^{-1} \underline{x}_k\right] \tag{11.102}$$

where

$$C = (\pi)^{-NK} \, |R_{nn}|^{-K} \exp\left[-\sum_{k=1}^{K} \underline{x}_k^\dagger R_{nn}^{-1} \underline{x}_k\right] \tag{11.103}$$

$$R_{ss} = S\underline{d}(\theta)d^\dagger(\theta) \tag{11.104}$$

$$B = \underline{d}^\dagger(\theta) R_{nn}^{-1} \underline{d}(\theta) \tag{11.105}$$

Here the scalar C is a constant, the *scalar* B depends on θ, and the $N \times N$ covariance matrix R_{ss} depends on both S and θ. Equation (11.102) was derived under the narrowband signal assumption, which permits the signal covariance matrix to be written as in (11.104). The N-component vector $\underline{d}(\theta)$ is the (unknown) vector of phase delays corresponding to the signal angle of arrival. For the case of a linear array of identical isotropic antenna elements (received signal power is the same in each element), S represents the (unknown) received signal power at each element so that $\underline{d}(\theta)$ can be written as

$$\underline{d} = [e^{j\beta_1\mu} e^{j\beta_2\mu} \cdots e^{j\beta_N\mu}]^T \tag{11.106}$$

where

$$\mu = \sin\theta \tag{11.107}$$

$$\beta_n = 2\pi\alpha_n/\lambda \tag{11.108}$$

The parameter α_n is the relative location of the nth element along the line array, and λ is the wavelength of the desired signal. Here, μ is the angle of arrival in sine space and θ is the AOA relative to broadside of the linear array. A more general definition for $\underline{d}(\theta)$ will be given later in this section.

Equation (11.102) is the conditional joint density of the observations $\underline{x}_k, k = 1, \ldots, K$, given the unknown parameters S and θ. When viewed as a function of the unknown parameters, it is known as the likelihood function. The maximum likelihood estimate is the value of the parameters that maximize the likelihood function, so the objective is to obtain the values for S and θ that maximize the density p in (11.102). As will be shown, closed-form solutions for θ and S are difficult to obtain in general, so numerical search methods must often be used. Although one could search all possible values of θ and S in Equations (11.102), (11.104), and (11.105) to find those values that maximize p, a computationally intensive multidimensional search would be required. Fortunately, it turns out that in this case the dependence of p on θ and S is separable, reducing to a search only over θ. This result is developed subsequently.

Before proceeding further with the derivation, the treatment will now be generalized to allow for multiple desired signals. The subsequent development follows that of Jaffer [38]. Define M narrowband signal sources with angles of arrival denoted by $\theta_1, \theta_2, \ldots \theta_M$. The received data vector $\underline{x}_k$ can be written as the sum of the M signals plus the noise term

$$\underline{x}_k = \sum_{m=1}^{M} b_{km}\underline{d}_m + \underline{n}_k \tag{11.109}$$

where b_{km} is the (complex) amplitude of the kth sample of the mth signal, and $\underline{d}_m = \underline{d}_m(\theta_m)$ is the direction delay vector for the mth signal. The general form of $\underline{d}$ for the mth narrowband signal source is given by [38]

$$\underline{d}_m(\theta_m) = \left[g_1(\theta_m)e^{-j\phi_1(\theta_m)}, g_2(\theta_m)e^{-j\phi_2(\theta_m)} \ldots g_N(\theta_m)e^{-j\phi_N(\theta_m)}\right] \tag{11.110}$$

where θ_m is the angular location (angle of arrival) of the mth source, g_n is the complex gain of the nth antenna element (generally a function of θ), and $\phi_n(\theta_m)$ is the phase delay of the m^{th} mth source at the nth antenna element relative to a suitable reference point (e.g., the array phase center).

It is convenient to rewrite (11.107) in vector/matrix notation as

$$\underline{x}_k = D(\underline{\theta})\underline{b}_k + \underline{n}_k \tag{11.111}$$

The M-dimensional complex vector $\underline{b}_k$ denotes the signal amplitude and phase in complex notation for each of the M signals for the kth snapshot. Under the stochastic signal model, $\underline{b}_k$ is assumed to be a zero-mean Guassian random vector with covariance

$$R_{bb} = E\{\underline{b}_k\underline{b}_k^{\dagger}\} \tag{11.112}$$

The direction vectors for each of the M signals, $\underline{d}_m(\theta_m)$, $m = 1, \ldots M$, form the columns of the $N \times M$ matrix $D(\theta)$.

11.8.2 Maximum Likelihood Estimation of the Direction of Arrival for Stochastic Signals

In the multiple signal case, Equation (11.100) retains its form

$$p(\underline{x}_1, \underline{x}_2, \ldots \underline{x}_K \,|\theta_1,\ \theta_2, \ldots, \theta_M, R_{bb}) = (\pi)^{-NK}|R|^{-K} \exp\left[-\sum_{k=1}^{K} \underline{x}_k^{\dagger} R^{-1} \underline{x}_k\right] \tag{11.113}$$

Maximizing (11.111) with respect to $\underline{\theta}$ and R_{bb} is equivalent to maximizing $\ln(p)$, denoted the log likelihood function

$$\ln L^{'}(\underline{\theta}_1, \underline{\theta}_2, \ldots, \underline{\theta}_M, R_{bb}) = C^{'} - K \log|R| - \sum_{k=1}^{K} \underline{x}_k^{\dagger} R^{-1} \underline{x}_k \tag{11.114}$$

where the dependencies on $\underline{\theta}$ and R_{bb} are contained in R

$$R = D(\underline{\theta})R_{bb}D^{\dagger}(\underline{\theta}) + R_{nn} \tag{11.115}$$

The likelihood function in Equation (1.18) can be simplified by dropping the terms that do not depend on either $\underline{\theta}$ or R_{bb} and by rearranging terms

$$\ln L(\underline{\theta}, R_{bb}) = -K \log|R| - KTr(R^{-1}\hat{R}) \tag{11.116}$$

where $\hat{R}$ is the $N \times N$ sample covariance matrix

$$\hat{R} = \frac{1}{K}\sum_{k=1}^{K} \underline{x}_k x_k^{\dagger} \tag{11.117}$$

As discussed already for single-signal case, maximization of (11.114) directly is computationally intractable for all but the simplest cases, because it requires an $M^2 + M$ dimensional search to jointly optimize the components of $\underline{\theta}$ and R_{bb}. Jaffer [38] reduced this to an M-dimensional search of $\underline{\theta}$ by showing that the maximum likelihood estimate $\underline{\theta} = \underline{\hat{\theta}}_{ML}$ that maximizes $L(\underline{\theta}, R_{bb})$ in Equation (11.114) can be obtained by maximizing $J_{sto}(\underline{\theta})$ with respect to $\theta_1, \theta_2, \ldots \theta_M$

$$J_{sto}(\underline{\theta}) = \frac{1}{\sigma^2}\mathrm{Tr}\left[G(\underline{\theta})\hat{R}\right] - \ln\left|G(\underline{\theta})\hat{R}G(\underline{\theta}) + \sigma_n^2(I_N - G(\underline{\theta}))\right| \tag{11.118}$$

where

$$G(\underline{\theta}) = D(\underline{\theta})[D^{\dagger}(\underline{\theta})D(\underline{\theta})]^{-1}D^{\dagger}(\underline{\theta}) \tag{11.119}$$

As noted in [38], $G(\theta)$ is the orthogonal projection matrix. For the purpose of simplification, the derived result in (11.116) made the assumption that the noise covariance matrix R_{nn} is known and is given by $R_{nn} = \sigma_n^2 I_N$; that is, the only noise terms are due to thermal noise. The result in (11.116) is generalized to handle an arbitrary positive definite matrix R_{nn} by transforming $\underline{x}_k$ into a new coordinate system so that the noise covariance matrix in the new coordinate system is $\sigma^2 I_N$, that is,

$$\underline{y}_k = R_{nn}^{-1/2}\underline{x}_k \tag{11.120}$$

Once $\underline{\hat{\theta}}_{ML}$ is determined, Jaffer [38] wrote the maximum likelihood estimate of R_{bb} as

$$\hat{R}_{bb}\left(\underline{\hat{\theta}}_{ML}\right) = \left[D^{\dagger}\left(\underline{\hat{\theta}}_{ML}\right)D\left(\underline{\hat{\theta}}_{ML}\right)\right]^{-1}D^{\dagger}\left(\underline{\hat{\theta}}_{ML}\right)\left[\hat{R} - \sigma_n^2 I_N\right]D\left(\underline{\hat{\theta}}_{ML}\right)\left[D^{\dagger}\left(\underline{\hat{\theta}}_{ML}\right)D\left(\underline{\hat{\theta}}_{ML}\right)\right]^{-1} \tag{11.121}$$

In summary to this point, we have shown that the maximum likelihood estimate of the angles of arrival of M signal sources is determined by finding those values of $\theta_1, \theta_2, \ldots, \theta_M$ that maximize $J(\underline{\theta})$ in Equation (11.116). This requires an M-dimensional search over the angular regions of interest. To evaluate $J(\theta)$, it is assumed that the following parameters are known a priori: (1) antenna element gain versus θ; (2) antenna element location; and (3) the noise covariance matrix $R_{nn} = \sigma_n^2 I_N$. The sample covariance matrix is computed from the data samples (11.115). Note the result in (11.116) assumes that the signals are narrowband and propagate as plane waves. Also note that equation (11.116) assumes that R_{nn} is known a priori: in most practical situations, the external interference is not known beforehand and must be estimated from the data samples. In radar, it is often possible to estimate R_{nn} by averaging the sample covariance matrix using adjacent range cells that do not contain the target (desired signal). Such an estimate is more difficult in a communications system in which the signal is present in all the data samples. In this case, it may be necessary to use an a priori estimate of R_{nn}. For example, if it is assumed that R_{nn} is made up of only internal noise ($R_{nn} = \sigma_n^2 I_N$), then all external sources must be estimated using Equation (11.116).

The M-dimensional search required to find the values of θ_m that maximize $J(\theta)$ can become computationally intensive for large M. In practice, M is limited to two or three sources by limiting the angular search region, typically one or two beamwidths in extent. Any sources outside the search region (e.g., in the sidelobes) are treated as interference and therefore must be included in R_{nn}, which as noted already must be known or estimated. Methods for obtaining an accurate estimate of R_{nn} depend on the particular situation

(e.g., radar vs. communications vs. pulsed signals, amplitude of the desired signals), which is beyond the scope of the current discussion.

11.8.3 Maximum Likelihood Estimation of the Direction of Arrival for Deterministic Signals

The preceding section developed the maximum likelihood estimate of the AOAs of multiple signals under the assumption that the signals are zero-mean Gaussian distributed. This assumption is most appropriate when the received signal amplitude and phase tend to vary sample to sample in a random manner, as in radar or signal monitoring systems.

In this section, we consider the case where the signal is modeled as "deterministic." This assumption is more appropriate for communications signals where sample-to-sample variations are to be estimated and for preamble signals where the signal is known a priori.

As in the stochastic case, the data vector $\underline{x}_k$ is given by Equation (11.111)

$$\underline{x}_k = D(\underline{\theta})\underline{b}_k + \underline{n}_k$$

where the noise component $\underline{n}_k$ is, as in the preceding section, assumed to be a sample function from a zero-mean, Gaussian process with covariance matrix $R_{nn} = E\{\underline{n}_k \underline{n}_k^\dagger\}$. No assumption is made about the statistics of $\underline{s}_k = D(\underline{\theta})\underline{b}_k$. The conditional joint density of K independent data samples $\underline{x}_k$, $k = 1, 2, \ldots K$, is then expressed as

$$p(\underline{x}_1, \ldots \underline{x}_K \,|\underline{b}_1\, , \ldots \underline{b}_M, \underline{\theta}_1, \ldots \underline{\theta}_M) = (\pi)^{-MK}\,|R_{nn}|^{-K}$$
$$\exp\left[-\sum_{k=1}^{K}(\underline{x}_k - D(\theta)\underline{b}_k)^\dagger R_{nn}^{-1}(\underline{x}_k - D(\theta)\underline{b}_k)\right] \tag{11.122}$$

In this case, the likelihood function p is maximized by minimizing the exponent, which is a real scalar given by

$$H = \frac{1}{K}\sum_{k=1}^{K} H_k \tag{11.123}$$

where

$$H_k = [\underline{b}_k - T^{-1}V\underline{x}_k]^\dagger T[\underline{b}_k - T^{-1}V\underline{x}_k] + \underline{x}_k^\dagger R_{nn}^{-1}\underline{x}_k + \underline{x}_k^\dagger V^\dagger T^{-1}V\underline{x}_k \tag{11.124}$$
$$T = D^\dagger(\underline{\theta})R_{nn}^{-1}D(\underline{\theta}) \tag{11.125}$$
$$V = D^\dagger(\underline{\theta})R_{nn}^{-1}$$

V is an $M \times N$ matrix, and the $M \times M$ matrix T is assumed positive definite so that H_k is minimized with respect to $\underline{b}_k$ when

$$\underline{b}_k(\underline{\theta}) = (D^\dagger(\underline{\theta})R_{nn}^{-1}D(\underline{\theta}))^{-1}D^\dagger(\underline{\theta})R_{nn}^{-1}\underline{x}_k \tag{11.126}$$

Substituting $\underline{b}_k$ into Equation (11.123) and deleting terms that do not depend on $\underline{\theta}$, one obtains an expression for H that depends only on $\underline{\theta}$

$$J_{\text{det}}(\underline{\theta}) = \frac{1}{K}\sum_{k=1}^{K}\underline{x}_k^\dagger R_{nn}^{-1}D(\underline{\theta})(D^\dagger(\underline{\theta})R_{nn}^{-1}D(\underline{\theta}))^{-1}D^\dagger(\underline{\theta})R_{nn}^{-1}\underline{x}_k \tag{11.127}$$

The ML estimates of the AOA of M narrowband signal sources received by an N-element array, under the deterministic signal model, are the values of $\theta_1, \theta_2, \ldots \theta_M$ that maximize

$J_{\text{det}}(\underline{\theta})$. $\underline{b}_k$ is then found by substituting the ML estimates of θ_k, $k = 1, \ldots, M$ into Equation (11.126). Note that the noise covariance matrix R_{nn} assumed in the derivation of $J_{\text{det}}(\underline{\theta})$ allows for external interference and thus is a more general result than $J_{sto}(\underline{\theta})$.

The result for deterministic signals, $J_{\text{det}}(\underline{\theta})$ in Equation (11.127), can be compared with the stochastic case, $J_{sto}(\underline{\theta})$ in Equation (11.116). For $R_{nn} = \sigma_n^2 I_N$, which was the assumption in deriving $J_{sto}(\underline{\theta})$ in (11.116), $J_{\text{det}}(\underline{\theta})$ in (11.117) is the same as the first term in $J_{sto}(\underline{\theta})$. The only difference between $J_{\text{det}}(\underline{\theta})$ and $J_{sto}(\underline{\theta})$, under the assumption that $R_{nn} = \sigma_n^2 I_N$, is the second term on the right side of Equation (11.116). Experience has shown that the second term in (11.116) is small relative to the first term for moderate to high SNR, so accuracy of the two methods is very nearly the same. In general, AOA accuracy is relatively insensitive to sample-to-sample signal variations.

It is instructive to reformulate $J_{\text{det}}(\underline{\theta})$ as follows. Let

$$\underline{u}(\underline{\theta}) = R_{nn}^{-1/2} D(\underline{\theta}) \tag{11.128}$$

and

$$\underline{y}_k = R_{nn}^{-1/2} \underline{x}_k \tag{11.129}$$

Then Equation (11.127) becomes

$$J_{\text{det}}(\underline{\theta}) = \frac{1}{K} \sum_{k=1}^{K} \underline{y}_k^{\dagger} U(\underline{\theta}) (U^{\dagger}(\underline{\theta}) U(\underline{\theta}))^{-1} U^{\dagger}(\underline{\theta}) \underline{y}_k \tag{11.130}$$

This has the same form as the first term in the expression for $J_{sto}(\underline{\theta})$, and indicates how the first term in (11.116) can be transformed to generalize it to any positive definite R_{nn}.

11.9 | CRAMER–RAO LOWER BOUND ON AOA ESTIMATION ERROR

The Cramer–Rao bound is derived for the general case of estimating the parameters of a signal corrupted by Gaussian noise and interference. We will derive a generalized result and then will specialize it to the estimation of the AOA. The CR bound, along with other bounds such as Barankin and Ziv–Zakai, is useful for determining system design requirements (e.g., antenna size, SNR, number of antenna elements N, number of snapshots K) necessary to achieve a certain level of estimation accuracy. The CR bound is the best known in that it often leads to closed-form solutions and provides a relatively tight bound under a broad range of practical conditions [39,40], most notably in cases where the optimum array SNR exceeds a certain level. The development will follow that of Ballance and Jaffer [39], starting with a generalized version of the N-variate Gaussian probability density function derived under the deterministic signal model.

Define the kth snapshot of the N-element antenna array as

$$\underline{x}_k = \underline{s}_k(\underline{\upsilon}) + \underline{n}_k \tag{11.131}$$

where $\underline{\upsilon}$ is an M-dimensional vector of real-valued, unknown, nonrandom parameters. The Cramer–Rao lower bound on the variance of any unbiased estimate of the signal parameter υ_m, $m = 1, \ldots M$ is given by the m, mth component of the inverse of the $M \times M$ Fisher information matrix (FIM) [40]

$$\sigma_{\underline{\upsilon}_m}^2 \geq [\Gamma^{-1}]_{m,m} \tag{11.132}$$

The $M \times M$ Fisher information matrix Γ is a symmetric matrix given by the expected value of the second partial derivatives of the log likelihood function with respect to the unknown parameters $\underline{\upsilon}_1, \underline{\upsilon}_2, \ldots \underline{\upsilon}_M$

$$\Gamma_{ij} = E\left\{-\frac{\partial^2 \ln L(\upsilon_1, \ldots \upsilon_M)}{\partial \upsilon_i \partial \upsilon_j}\right\} \tag{11.133}$$

From Equation (11.100), the log likelihood function for the stochastic signal model is given by

$$\ln L(\underline{\upsilon}) = -NK \ln(\pi) - K \ln|R| - \sum_{k=1}^{K} \underline{x}_k^{\dagger} R_{nn}^{-1} \underline{x}_k \tag{11.134}$$

Now to simplify the development (a more general case will be derived later), assume a single narrowband source with $N \times 1$ direction vector $\underline{d}(\mu)$, as in Equation (11.95). Then the likelihood function reduces to Equation (11.101) and the log likelihood function reduces to

$$\ln L = \ln C - K \ln(1 + SB) + \frac{S}{(1 + SB)} \sum \underline{x}_k^{\dagger} R_{nn}^{-1} \underline{d}(\mu) \underline{d}^{\dagger}(\mu) R_{nn}^{-1} \underline{x}_k \tag{11.135}$$

where C is given by (11.103), B is given by (11.105), and S is the received signal power as defined in (11.102). Taking the partial derivatives in Equation (11.102) with respect to μ and S, one obtains for the 2×2 Fisher information matrix

$$\Gamma_{11} = -E\left\{\frac{\partial^2 \ln L}{\partial \mu^2}\right\} = \frac{KS^2}{(1 + SB)^2}\left\{F^2 + 2(1 + SB)B^2 D\right\}$$

$$\Gamma_{21} = -E\left\{\frac{\partial^2 \ln L}{\partial S \partial \mu}\right\} = \frac{KSB}{(1 + SB)^2} F = \Gamma_{12}$$

$$\Gamma_{22} = -E\left\{\frac{\partial^2 \ln L}{\partial S^2}\right\} = \frac{KB^2}{(1 + SB)^2}$$

Inverting Γ, we obtain the final results for Γ^{-1}

$$\sigma_{\mu}^2 \geq \Gamma_{11}^{-1} = \frac{1 + SB}{2KS^2 B^2 Q} \tag{11.136}$$

$$\Gamma_{12}^{-1} = \Gamma_{11}^{-1} \frac{S}{B} F = \Gamma_{21}^{-1} \tag{11.137}$$

$$\sigma_S^2 \geq \Gamma_{22}^{-1} = \Gamma_{11}^{-1} \frac{S^2}{B^2}\left\{F^2 + 2(1 + SB)B^2 Q\right\} \tag{11.138}$$

where Q and F are real scalar quantities given by

$$Q = \frac{1}{B^2}\left\{B\underline{d}^{\dagger}(\mu) A^{\dagger} R_{nn}^{-1} A \underline{d}(\mu) - \left|\underline{d}^{\dagger}(\mu) A^{\dagger} R_{nn}^{-1} \underline{d}(\mu)\right|^2\right\}$$

$$F = \underline{d}^{\dagger}(\mu) R_{nn}^{-1} A \underline{d}(\mu) + \underline{d}^{\dagger}(\mu) A^{\dagger} R_{nn}^{-1} \underline{d}(\mu)$$

and A is defined as the $M \times M$ diagonal matrix with diagonal elements given by $A_{nn} = \beta_n = 2\pi\alpha_n/\lambda; n = 1, \ldots N$.

Equation (11.136) gives the CR lower bound on the variance of any unbiased estimate of the angle of arrival θ, and Equation (11.137) gives the lower bound on the variance of any unbiased estimate of the signal power S.

For an N-element line array of identical and equally spaced antenna elements, each with unity gain, and assuming $R_{nn} = \sigma_n^2 I_N$ (so $S/\sigma_n^2 =$ the signal-to-noise ratio of the signal received by each element), the lower bound on the variance of the AOA estimate about its true value is given by

$$\sigma_\mu^2 \geq E(\hat{\mu} - \mu_{true})^2 = \Gamma_{11}^{-1} = \frac{6}{(2\pi)^2}\frac{1}{K}\frac{\left(1 + NS/\sigma_n^2\right)}{\left(NS/\sigma_n^2\right)^2}\frac{1}{(N^2-1)}\left(\frac{\lambda}{\alpha}\right)^2 \tag{11.139}$$

where α is the separation between antenna elements. Note that NS/σ_n^2 is the array SNR. Equation (11.101) applies to the deterministic signal case, where the signal is unknown but nonrandom. Ballance and Jaffer [39] showed that for the stochastic signal model

$$\sigma_\theta^2 \geq E(\hat{\mu} - \mu_{true})^2 = \Gamma_{11}^{-1} = \frac{6}{(2\pi)^2}\frac{1}{K}\frac{1}{\left(NS/\sigma_n^2\right)}\frac{1}{(N^2-1)}\left(\frac{\lambda}{\alpha}\right)^2 \tag{11.140}$$

which is smaller than Equation (11.139) by the ratio

$$\frac{\text{stochastic bound}}{\text{deterministic bound}} = \frac{\left(NS/\sigma_n^2\right)}{\left(1 + NS/\sigma_n^2\right)} \tag{11.141}$$

For moderate to large SNR (greater than approx 10 dB), the two bounds are nearly the same.

11.10 | FISHER INFORMATION MATRIX AND CR BOUND FOR GENERAL CASES

This section gives an overview of results derived by Ballance and Jaffer [39] that give the FIM for the time-varying, nonlinear case (i.e., $\underline{s}_k$ may be time varying and be a nonlinear transformation of $\underline{\upsilon}$). The FIM can then be inverted to determine the Cramer–Rao bound.

The specific result given here applies to the deterministic signal model. Starting with the general equation for the data vector $\underline{x}_k = \underline{s}_k(\underline{\upsilon}) + \underline{n}_k$, and the general form for the likelihood function in Equation (11.122), the log likelihood function is given by

$$\ln L = -NK\ln(\pi) - K\ln|R_{nn}| - \sum_{k=1}^{K}(\underline{x}_k - \underline{s}_k(\underline{\upsilon}))^\dagger R_{nn}^{-1}(\underline{x}_k - \underline{s}_k(\underline{\upsilon})) \tag{11.142}$$

Note the difference between the previous deterministic case and the stochastic case in Equation (11.114). The deterministic case is easier to deal with because only the exponent depends on the unknown parameter vector $\underline{\upsilon}$. Taking the partial derivatives defined in Equation (11.133) and following the derivation in [39], the general $M \times M$ symmetric FIM is given by

$$\Gamma = 2\sum_{k=1}^{K}\text{Re}\left\{H_k^\dagger R_{nn}^{-1} H_k\right\} \tag{11.143}$$

where H_k is the $N \times M$ matrix

$$H_k = \left[\frac{\partial \underline{s}_k}{\partial \underline{\upsilon}_1} \cdots \frac{\partial \underline{s}_k}{\partial \underline{\upsilon}_M} \right] \tag{11.144}$$

The AOA is determined from one of terms of Γ^{-1}. Consider the case of a single emitter, with $R_{nn}^{-1} = \sigma_n^2 I_N$. Let the kth snapshot $\underline{x}_k = \underline{d}(\theta) b_k$ where b_k is a 1×1 complex vector, and $\underline{d}(\theta)$ is the $N \times 1$ vector defined as

$$\underline{d}(\theta) = \left[g_1(\theta) e^{-j\varphi_1(\theta)}, g_2(\theta) e^{-j\varphi_2(\theta)} \ldots g_N(\theta) e^{-j\varphi_N(\theta)} \right] \tag{11.145}$$

where $\varphi_n(\theta) = \frac{2\pi}{\lambda} \tau_n(\theta)$. $\tau_n(\theta)$ is the time delay experienced by the desired signal relative to fixed reference delay. The components of the FIM are determined by letting $\upsilon_1 = \theta$, $\upsilon_2 = \angle b_1$, $\upsilon_3 = |b_1|$, $\upsilon_4 = \angle b_2$, $\upsilon_5 = |b_2|, \ldots \upsilon_{2K+1} = |b_K|$, and then taking the partial derivatives in (11.144). Following [39], Γ_{11}^{-1} is found to be given by

$$\sigma_{\hat{\mu}}^2 \geq \Gamma_{11}^{-1} = \frac{1}{2K(SNR_1)} \left[\left\| \underline{d}' \right\|^2 - \frac{\left| \underline{d}'^{\dagger} \underline{d} \right|^2}{\|\underline{d}\|^2} \right] \tag{11.146}$$

where $SNR_1 = \dfrac{1}{K} \displaystyle\sum_{k=1}^{N} \frac{|b_k|^2}{\sigma_n^2}$ is the average array signal-to-noise ratio that would be received by antenna elements with unity gain.

11.11 SUMMARY AND CONCLUSIONS

Using the main beam of an array to locate signals generates the periodogram. The periodogram does not accurately locate targets or resolve closely spaced targets. Superresolution techniques, such as Capon's method, MUSIC, root MUSIC, and MEM, use nulls to accurately determine a target's position as well as to resolve closely spaced targets. The MEM approach to spectral estimation was introduced for obtaining high-resolution power spectral density estimates of stationary time series from limited data records. Extension of Burg's scalar MEM algorithm to the multiple-channel problem is required for array processing problems, and the extension provided by Strand was outlined for this purpose. The Problems section introduces a modification of the Burg algorithm introduced by Marple that alleviates the line-splitting problem occasionally occurring with this algorithm. When both signal detection and parameter estimation must be accomplished, then the Bayes optimal array processor can be implemented sequentially to provide adaptive capabilities in a natural manner. The notions of subspace fitting and eigenspace beamformers were introduced using the concept of eigendecomposition as the point of departure.

11.12 PROBLEMS

1. ***Computer Simulation of Direction of Arrival Estimation Algorithms.***

   a. Use a periodogram to demonstrate the effect of separation angle between two sources using an 8 element uniform array with $\lambda/2$ spacing when $\theta_1 = -30^\circ$, 10°, 20° and $\theta_2 = 30^\circ$.

b. An 8 element uniform array with $\lambda/2$ spacing has 3 signals incident upon it: $s_1(-60^\circ) = 1$, $s_2(0^\circ) = 2$, and $s_3(10^\circ) = 4$. Find the Capon spectrum.

c. An 8 element uniform array with $\lambda/2$ spacing has three signals incident upon it: $s_1(-60^\circ) = 1$, $s_2(0^\circ) = 2$, and $s_3(10^\circ) = 4$. Find the MEM spectrum.

d. An 8 element uniform array with $\lambda/2$ spacing has three signals incident upon it: $s_1(-60^\circ) = 1$, $s_2(0^\circ) = 2$, and $s_3(10^\circ) = 4$. Find the MUSIC spectrum.

e. An 8 element uniform array with $\lambda/2$ spacing has three signals incident upon it: $s_1(-60^\circ) = 1$, $s_2(0^\circ) = 2$, and $s_3(10^\circ) = 4$. Find the location of the signals using the root MUSIC algorithm.

2. Prediction Error Filter Equations The prediction error filter matrix equation for a scalar random process may be developed by assuming that two sampled values of a random process x_0 and x_1 are known, and it is desired to obtain an estimate of the next sampled value $\hat{x}_2$ using a second-order prediction error filter

$$\hat{x}_2 = a(2,2)x_0 + a(2,1)x_1$$

so that $\varepsilon = x_2 - \hat{x}_2 = x_2 - a(2,1)x_1 - a(2,2)x_0$

a. Using the fact that the optimal linear predictor must provide estimates for which the error is orthogonal to the data (i.e., $\overline{x_0\varepsilon} = 0$ and $\overline{x_1\varepsilon} = 0$), show that

$$\begin{bmatrix} r(1) & r(0) & r(2) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a(2,1) \\ -a(2,2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where $r(n) \triangleq \overline{x_i x_j}$, $|i - j| = n$.

b. If $P_2 \triangleq \overline{\varepsilon^2} = \overline{(x_2 - \hat{x}_2)(x_2 - \hat{x}_2)}$, use the fact that the error in the estimate $\hat{x}_2$ is orthogonal to the estimate itself (i.e., $\overline{(x_2 - \hat{x}_2)\hat{x}_2} = 0$) and the previously given expression for $\hat{x}_2$ to show that

$$P_2 = r(0) - a(2,1)r(1) - a(2,2)r(2)$$

The results of part (a) combined with the result from part (b) then yield the prediction error filter matrix equation for this case.

3. *The Relationship between MEM Spectral Estimates and ML Spectral Estimates [19]* Assume the correlation function of a random process $\mathbf{x}(t)$ is known at uniformly spaced, distinct sample times. Then the ML spectrum (for an equally spaced line array of N sensors) is given by

$$\text{MLM}(k) = \frac{N\Delta x}{\mathbf{v}^\dagger(k)\mathbf{R}_{xx}^{-1}\mathbf{v}(k)}$$

where

k = wavenumber (reciprocal of wavelength)
Δx = spacing between adjacent sensors
$\mathbf{v}(k)$ = beam steering column vector where $v_n(k) = e^{-j2\pi nk\Delta x}$, $n = 0, 1, \ldots, N-1$
$\mathbf{R}_{xx}$ = $N \times N$ correlation matrix of $\mathbf{x}(t)$

a. First, define the lower triangular matrix $\mathbf{L}$ by

$$\mathbf{L} \triangleq \begin{bmatrix} 1 & 0 & \cdots & 0 \\ c(2,N) & 1 & \cdots & 0 \\ c(3,N) & c(2,N-1) & \cdots & 0 \\ \vdots & & & \\ c(N,N) & c(N-1,N-1) & \cdots & 1 \end{bmatrix}$$

where $1, c(2,M), \ldots, c(M,M)$ are the weights of the M-long prediction error filter whose output power is $P(M)$. Note that

$$\mathbf{R}_{xx}\mathbf{L} = \begin{bmatrix} P(N) & — & — & — & — \\ 0 & P(N-1) & — & — & — \\ 0 & 0 & & — & — \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & & & P(1) \end{bmatrix}$$

The maximum entropy spectrum estimate corresponding to the M-long prediction error filter is given by

$$\text{MEM}(k,M) = \frac{P(M)\Delta x}{\left|\sum_{i=1}^{M} c(i,M)\exp(j2\pi k(i-1)\Delta x)\right|^2}$$

where $c(1,M) \equiv 1$. Define the matrix $\mathbf{P}$ according to

$$\mathbf{P} \equiv \mathbf{L}^\dagger \mathbf{R}_{xx}\mathbf{L}$$

Show that $\mathbf{P}$ is an $N \times N$ diagonal matrix whose diagonal elements are given by $P(N)$, $P(N-1), \ldots, P(1)$.

b. Using the fact that $\mathbf{R}_{xx}^{-1} = \mathbf{L}\mathbf{P}^{-1}\mathbf{L}^\dagger$, show that

$$\mathbf{v}^\dagger \mathbf{R}_{xx}^{-1}\mathbf{v} = (\mathbf{L}^\dagger\mathbf{v})^\dagger \mathbf{P}^{-1}(\mathbf{L}^\dagger\mathbf{v}) = \sum_{n=1}^{N} \frac{\Delta x}{\text{MEM}(k,n)}$$

this result then gives the desired relationship

$$\frac{1}{\text{MLM}(k)} = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{\text{MEM}(k,n)}$$

Therefore, the reciprocal of the ML spectrum is equal to the average of the reciprocals of the maximum entropy spectra obtained from the one-point up to the N-point prediction error filter. The lower resolution of the ML method therefore results from the "parallel resistor network averaging" of the lowest- to the highest-resolution maximum entropy spectra.

4. ***Equivalence of MEM Spectral Analysis to Least-Squares Fitting of a Discrete-Time All-Pole Model to the Available Data [15]*** Assume that the first $(N+1)$ points $\{r(0), r(1), \ldots, r(N)\}$

of the autocorrelation function of a stationary Gaussian process are known exactly, and it is desired to estimate $r(N+1)$. Consider the Toeplitz covariance matrix

$$\mathbf{R}_{N+1} = \begin{bmatrix} r(0) & r(1) & \cdots & r(N) & r(N+1) \\ r(1) & r(0) & \cdots & r(N-1) & r(N) \\ \cdots & \cdots & \cdots & & \\ r(N+1) & r(N) & & r(1) & r(0) \end{bmatrix}$$

The basic autocorrelation function theorem states that $\mathbf{R}_{N+1}$ must be semipositive definite if the quantities $r(0), r(1), \ldots, r(N+1)$ are to correspond to an autocorrelation function. Consequently $\det[\mathbf{R}_{N+1}]$ must be nonnegative.

MEM spectral analysis seeks to select that value of $r(N+1)$ that maximizes $\det[\mathbf{R}_{N+1}]$. The entropy of the $(N+2)$ dimensional probability density function with covariance matrix $\mathbf{R}_{N+1}$ is given by

$$\text{entropy} = 1n(2\pi e)^{[N+2/2]}\det[\mathbf{R}_{N+1}]^{1/2}$$

and the choice for $r(N+1)$ maximizes this quantity. To obtain $r(N+2)$, the value of $r(N+1)$ just found is substituted into $\mathbf{R}_{N+2}$ to find $\det[\mathbf{R}_{N+2}]$, and the corresponding entropy is maximized with respect to $r(N+2)$. Likewise, substituting the values of $r(N+1)$ and $r(N+2)$ found already into $\det[\mathbf{R}_{N+3}]$ and maximizing yields $r(N+3)$. The estimates for additional values $r(N+4), r(N+5), \cdots$ may then be evaluated by following the same procedure.

a. Show that maximizing $\det[\mathbf{R}_{N+1}]$ with respect to $r(N+1)$ is equivalent to the relation

$$\det \begin{bmatrix} r(1) & r(0) & \cdots & r(N-1) \\ r(2) & r(1) & \cdots & r(N-2) \\ \vdots & \vdots & & \vdots \\ r(N+1) & r(N) & \cdots & r(1) \end{bmatrix} = 0$$

b. Consider the all-pole data prediction error model given by

$$y(n) + a_1' y(n-1) + \cdots + a_N' y(n-N) = e(n)$$

or

$$\mathbf{y}^T \mathbf{a}' = e(n)$$

where

$\mathbf{a}'^T = [1, a_1', a_2', \ldots, a_N']$
$\mathbf{y}^T = [y(n), y(n-1) \cdots y(n-N)]$
$N =$ order of all-pole model
$n =$ number of data samples, $n > N$

and where $e(n)$ is a zero-mean random variable with $E\{e(i)e(j)\} = 0$ for $i \neq j$. Assuming that $E\{e(n)y(n-k)\} = 0$ for $k > 0$, show that multiplying both sides of the previous equation for $e(n)$ by $y(n-k)$ and taking expectations yields

$$r'(k) + a_1' r'(k-1) + a_2' r'(k-2) + \cdots + a_N' r'(k-N) = 0, \qquad \text{for } k > 0$$

where $r'(k) \triangleq E\{y(n)y(n-k)\}$.

c. Using the results of part (b) and the fact that $r(\tau) = r(-\tau)$, it follows that

$$\begin{aligned} &r'(1) + a'_1 r'(0) + \cdots + a'_N r'(N-1) = 0 \\ &r(2) + a'_1 r'(1) + \cdots + a'_N r'(N-2) = 0 \\ &\vdots \\ &r'(N+1) + a'_1 r'(N) + \cdots + a'_N r'(1) = 0 \end{aligned}$$

or $\mathbf{R}'_{N+1}\mathbf{a}' = \mathbf{0}$ in matrix notation. Use this result to show that

$$\det \begin{bmatrix} r'(1) & r'(0) & \cdots & r'(N-1) \\ r'(2) & r'(1) & \cdots & r'(N-2) \\ \vdots & \vdots & & \vdots \\ r'(N+1) & r'(N) & \cdots & r'(1) \end{bmatrix} = 0$$

If the first $N+1$ exact values $\{r(0), r(1), \ldots, r(N)\}$ of any autocorrelation function are available, then substituting these values into the first N of the simultaneous linear equations corresponding to $\mathbf{R}_{N+1}\mathbf{a} = \mathbf{0}$ yields a unique solution for the coefficients $\{a_1, a_2, \ldots, a_N\}$. Consequently, the value for $r(N+1)$ for a discrete-time all-pole model having the coefficients $\{a_1, a_2, \ldots, a_N\}$ is uniquely determined by

$$\det \begin{bmatrix} r(1) & r(0) & \cdots & r(N-1) \\ r(2) & r(1) & \cdots & r(N-2) \\ \vdots & \vdots & & \\ r(N+1) & r(N) & \cdots & r(1) \end{bmatrix} = 0$$

This result is identical to the relation obtained in part (a); hence, the same solution would have been obtained from maximum entropy spectral analysis.

5. ***Angle of Arrival Estimation [37]*** The MEM technique has superior capability for resolving closely spaced spectral peaks that may be exploited for estimating the angular distribution of received signal power.

a. Using a time–space dualism, reformulate equation (11.19) to give a spatial spectrum $\hat{\phi}_{xx}(\mu)$, where $\mu = \cos\theta$, and θ is the angle from array endfire for an $N+1$ element linear array. Assume narrowband signals with spacing d between elements.

b. Reformulate equation (11.20) for the spatial estimation problem of part (a). What correspondence exists between the prediction error filter coefficients and the weights of a coherent sidelobe canceller with N auxiliary antennas?

6. ***The Marple Algorithm [41]*** A new autoregressive (AR) spectral analysis algorithm has been proposed that yields spectral estimates with no apparent line splitting (the occurrence of two or more closely spaced peaks in the AR spectral estimate where only one peak should be present) and reduced spectral peak frequency estimation biases. It exploits forward and backward linear prediction and therefore is closely related to the Burg algorithm. With the Burg algorithm the forward linear prediction error, $f_{M,k}$, in the single-channel case is given by

$$f_{M,k} = x_{k+M} + \sum_{i=1}^{M} a_{M,i} x_{k+M-i} = \sum_{i=0}^{M} a_{M,i} x_{k+M-i} \quad \text{for } 1 \le k \le N-M$$

where M is the order of the all-pole AR model, x_k is the kth sample output of the AR model, and $a_{M,m}$ is the AR parameter m of the Mth order process. Note that $a_{n,0}$ is defined as unity.

Likewise, the backward linear prediction error is given by

$$b_{M,k} = \sum_{i=0}^{M} a_{M,i}^{*} x_{k+i} \quad \text{also for } 1 \le k \le N - M$$

Since stationarity is assumed, the backward AR coefficients are the conjugates of the forward AR coefficients. To obtain estimates of the AR parameters, Burg minimized the sum of the backward and forward prediction error energies

$$e_M = \sum_{k=1}^{N-M} \left| f_{M,k} \right|^2 + \sum_{k=1}^{N-M} \left| b_{M,k} \right|^2$$

Substituting $f_{M,k}$ and $b_{M,k}$ into e_M and setting the derivatives of e_M with respect to the parameters $a_{M,1}$ through $a_{M,M}$ to zero, one obtains

$$2\sum_{j=0}^{M} a_{M,j} r_M(i, j) = 0 \quad \text{for} \quad i = 1, \ldots, M (a_{M,0} = 1 \text{ by definition})$$

where $r_M(i, j) = \sum_{k=1}^{N-M} (x_{k+M-j} x_{k+M-i}^{*} + x_{k+j}^{*} x_{k+i})$ for $0 \le i, j \le M$. The minimum prediction energy is then given by

$$e_M = \sum_{j=0}^{M} a_{M,j} r_M(0, j).$$

a. Show that the previous three expressions can be written in matrix form as

$$\mathbf{R}_M \mathbf{A}_M = \mathbf{E}_M$$

$$\text{where } \mathbf{A}_M = \begin{bmatrix} 1 \\ a_{M,1} \\ \vdots \\ a_{M,M} \end{bmatrix}, \mathbf{E}_M = \begin{bmatrix} e_M \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\text{and } \mathbf{R}_M = \begin{bmatrix} r_M(0,0) & \cdots & r_M(0, M) \\ \vdots & & \vdots \\ r_M(M,0) & \cdots & r_M(M, M) \end{bmatrix}$$

b. The matrix expression found in part (a) has a structure that can be exploited to produce an algorithm requiring a number of operations $\propto M^2$ rather than M^3. $\mathbf{R}_M$ has both Hermitian symmetry $[r_M(i, j) = r_M^{*}(j, i)]$ and Hermitian persymmetry $[r_M(i, j) = r_M^{*}(M-i, M-j)$; it does not have Toeplitz symmetry $[r_M(i, j) = r_M(i - j)]$, as the covariance matrix does. However, $\mathbf{R}_M$ is composed of Toeplitz matrices. Show that

$$\mathbf{R}_M = (\mathbf{T}_M)^H \mathbf{T}_M + (\mathbf{T}_M^{v})^H \mathbf{T}_M^{v}$$

$$\text{where } \mathbf{T}_M = \begin{bmatrix} x_{M+1} & x_M & \cdots & x_1 \\ x_{M+2} & x_{M+1} & \cdots & x_2 \\ \vdots & & & \vdots \\ x_N & x_{N-1} & \cdots & x_{N-M} \end{bmatrix}$$

$$\text{and } \mathbf{T}_M^{v} = \begin{bmatrix} x_1^{*} & \cdots & x_{M+1}^{*} \\ \vdots & & \vdots \\ x_{N-M}^{*} & \cdots & x_N^{*} \end{bmatrix} \quad \text{(conjugate and reversed matrix)}$$

c. To exploit this structure, introduce two new prediction error energy terms

$$e'_M = \sum_{k=1}^{N-M-1} \left[\left| f_{M,k+1} \right|^2 + \left| b_{M,k} \right|^2 \right] \quad \text{and} \quad e''_M = \sum_{k=1}^{N-M-1} \left[\left| f_{M,k} \right|^2 + \left| b_{M,k+1} \right|^2 \right]$$

which represent time-index-shifted variants of the definition for e_M. As a result of these new prediction error energy terms, show that
$\mathbf{R}'_M \mathbf{A}'_M = \mathbf{E}'_M$ and $\mathbf{R}''_M \mathbf{A}''_M = \mathbf{E}''_M$ where the elements of $\mathbf{R}'_M$ and $\mathbf{R}''_M$ are now given by

$$r'_M(i, j) = \sum_{k=1}^{N-M-1} \left[x_{k+M+1-j} x^*_{k+M+1-i} + x^*_{k+j} x_{k+i} \right]$$

$$\text{and } r''_M(i, j) = \sum_{k=1}^{N-M-1} \left[x_{k+M-j} x^*_{k+M-i} + x^*_{k+1+j} x_{k+1+i} \right]$$

As a result of these expressions, show that a persymmetry relationship exists:

$$r'_M(i, j) = [r''_M(M - i, M - j)]^*$$

d. Show that the following relationships exist among the correlation matrices $\mathbf{R}_M$, $\mathbf{R}'_M$, and $\mathbf{R}''_M$:

$$\mathbf{R}'_M = \mathbf{R}_M - \begin{bmatrix} x^*_{M+1} \\ \vdots \\ x^*_1 \end{bmatrix} \begin{bmatrix} x_{M+1}, \cdots, x_1 \end{bmatrix} - \begin{bmatrix} x_{N-M} \\ \vdots \\ x_N \end{bmatrix} \begin{bmatrix} x^*_{N-M}, \cdots, x^*_N \end{bmatrix}$$

$$\mathbf{R}''_M = \mathbf{R}_M - \begin{bmatrix} x^*_N \\ \vdots \\ x^*_{N-M} \end{bmatrix} \begin{bmatrix} x_N, \cdots, x_{N-M} \end{bmatrix} - \begin{bmatrix} x_1 \\ \vdots \\ x_{M+1} \end{bmatrix} \begin{bmatrix} x^*_1, \cdots, x^*_{M+1} \end{bmatrix}$$

$$\mathbf{R}_{M+1} = \left[\begin{array}{ccc} \mathbf{R}'_M & | & r_{M+1}(0, M+1) \\ \text{— — — — —} & | & \vdots \\ r_{M+1}(M+1, 0) & \cdots & r_{M+1}(M+1, M+1) \end{array}\right]$$

$$\mathbf{R}_{M+1} = \left[\begin{array}{ccc} r_{M+1}(0, 0) & \cdots & r_{M+1}(0, M+1) \\ \vdots & | & \text{— — — — —} \\ r_{M+1}(M+1, 0) & | & \mathbf{R}''_M \end{array}\right]$$

Now define the $(M + 1)$ element auxiliary column vectors

$$\mathbf{C}_M = \begin{bmatrix} c_{M,0} \\ \vdots \\ c_{M,M} \end{bmatrix}, \mathbf{D}_M = \begin{bmatrix} d_{M,0} \\ \vdots \\ d_{M,M} \end{bmatrix}$$

with analogous definitions for $\mathbf{C}''_M$ and $\mathbf{D}''_M$. The auxiliary column vector elements are defined by the matrix–vector products

$$\mathbf{R}_M \mathbf{C}_M = \begin{bmatrix} x^*_{M+1} \\ \vdots \\ x^*_1 \end{bmatrix}, \mathbf{R}_M \mathbf{D}_M = \begin{bmatrix} x_{N-M} \\ \vdots \\ x_N \end{bmatrix}$$

with analogous products for $\mathbf{R}''_M \mathbf{C}''_M$ and $\mathbf{R}''_M \mathbf{D}''_M$. Introduce the notation $\mathbf{A}^I_M$ to denote the vector formed by reversing the element order, and introduce the notation $\mathbf{A}^I_M$ to denote the

vector formed by reversing the element order and conjugating

$$\mathbf{A}_M^I = \begin{bmatrix} a_{M,M}^* \\ \vdots \\ a_{M,1}^* \\ 1 \end{bmatrix}$$

Likewise, define $\mathbf{E}_M^I$, $\mathbf{C}_M^I$, and $\mathbf{D}_M^I$. From the expressions of forward and backward prediction errors given for the Burg algorithm, show that these errors can be expressed in vector notation as

$$f_{M,1} = \begin{bmatrix} x_{M+1} \cdots, x_1 \end{bmatrix} \mathbf{A}_M \quad \text{and} \quad b_{M,N-M} = \begin{bmatrix} x_{M+1}, \cdots, x_N \end{bmatrix} \mathbf{A}_M^*.$$

At this point, we are approaching the development given by Burg but now using time-shifted AR parameters rather than the AR parameters employed before. The complete derivation is rather lengthy and will not be pursued here. A complete block diagram of the Maple algorithm is given in [41] along with a similar diagram for the Burg algorithm for ease in comparison.

7. **Computer Simulation Problem** An eight-element uniform array with $\lambda/2$ spacing has three signals incident upon it: $s_1(-60^\circ) = 1$, $s_1(-0^\circ) = 2$, and $s_3(-10^\circ) = 4$. Estimate the incident angles using ESPRIT.

11.13 REFERENCES

[1] J. Capon, "High-Resolution Frequency-Wavenumber Spectrum Analysis," *Proceedings of the IEEE,* Vol. 57, No. 8, 1969, pp. 1408–1418.

[2] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Transactions on Antennas and Propagation,* Vol. 34, No. 3, 1986, pp. 276–280.

[3] R. Schmidt and R. Franks, "Multiple Source DF Signal Processing: An Experimental System," *IEEE Transactions on Antennas and Propagation,* Vol. 34, No. 3, 1986, pp. 281–290.

[4] A. Barabell, "Improving the Resolution Performance of Eigenstructure-Based Direction-Finding Algorithms." pp. 336–339.

[5] R. T. Lacoss, "Data Adaptive Spectral Analysis Methods," *Geophysics*, Vol. 36, August 1971, pp. 661–675.

[6] T. J. Ulrych and T. N. Bishop, "Maximum Entropy Spectral Analysis and Autoregressive Decompositions," *Rev. Geophys. Space Phys.*, Vol. 13, 1975, pp. 183–200.

[7] J. P. Burg, "Maximum Entropy Spectral Analysis," Paper presented at the 37th Annual Meeting of the Society of Exploration Geophysicists, October 31, 1967, Oklahoma City, OK.

[8] J. P. Burg, "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, Vol. 1, Paper No. 15, Enschede, The Netherlands, 1968.

[9] T. J. Ulrych and R. W. Clayton, "Time Series Modelling and Maximum Entropy," *Phys. Earth Planet. Inter.*, Vol. 12, 1976, pp. 188–200.

[10] S. B. Kesler and S. Haykin, "The Maximum Entropy Method Applied to the Spectral Analysis of Radar Clutter," *IEEE Trans. Inf. Theory*, Vol. IT-24, No. 2, March 1978, pp. 269–272.

[11] R. G. Taylor, T. S. Durrani, and C. Goutis, "Block Processing in Pulse Doppler Radar," Radar-77, *Proceedings of the 1977 IEE International Radar Conference,* October 25–28, London, pp. 373–378.

[12] G. Prado and P. Moroney, "Linear Predictive Spectral Analysis for Sonar Applications," C. S. Draper Report R-1109, C. S. Draper Laboratory, Inc., 555 Technology Square, Cambridge, MA, September 1977.

[13] R. N. McDonough, "Maximum-Entropy Spatial Processing of Array Data," *Geophysics*, Vol. 39, December 1974, pp. 843–851.

[14] D. P. Skinner, S. M. Hedlicka, and A. D. Matthews, "Maximum Entropy Array Processing," *J. Acoust. Soc. Am.*, Vol. 66, No. 2, August 1979, pp. 488–493.

[15] A. Van Den Bos, "Alternative Interpretation of Maximum Entropy Spectral Analysis," *IEEE Trans. Inf. Theory*, Vol. IT-17, No. 4, July 1971, pp. 493–494.

[16] J. P. Burg, "Maximum Entropy Spectral Analysis," Ph.D. Dissertation, Stanford University, Department of Geophysics, May 1975.

[17] J. H. Sawyers, "The Maximum Entropy Method Applied to Radar Adaptive Doppler Filtering," Proceedings of the 1979 RADC Spectrum Estimation Workshop, October 3–5, Griffiss AFB, Rome, NY.

[18] S. Haykin and S. Kesler, "The Complex Form of the Maximum Entropy Method for Spectral Estimation," *Proc. IEEE*, Vol. 64, May 1976, pp. 822–823.

[19] J. P. Burg, "The Relationship Between Maximum Entropy Spectra and Maximum Likelihood Spectra," *Geophysics*, Vol. 37, No. 2, April 1972, pp. 375–376.

[20] R. H. Jones, "Multivariate Maximum Entropy Spectral Analysis," Paper presented at the Applied Time Series Analysis Symposium, May 1976, Tulsa, OK.

[21] R. H. Jones, "Multivariate Autoregression Estimation Using Residuals," Paper presented at Applied Time Series Analysis Symposium, May 1976, Tulsa, OK; also published in *Applied Time-Series Analysis*, edited by D. Findley, Academic Press, New York, 1977.

[22] A. H. Nutall, "Fortran Program for Multivariate Linear Predictive Spectra Analysis Employing Forward and Backward Averaging," Naval Underwater System Center, NUSC Tech. Doc. 5419, New London, CT, May 9, 1976.

[23] A. H. Nutall, "Multivariate Linear Predictive Spectral Analysis Employing Weighted Forward and Backward Averaging: A Generalization of Burg's Algorithm," Naval Underwater System Center, NUSC Tech. Doc. 5501, New London, CT, October 13, 1976.

[24] A. H. Nutall, "Positive Definite Spectral Estimate and Stable Correlation Recursion for Multivariate Linear Predictive Spectral Analysis," Naval Underwater System Center, NUSC Tech. Doc. 5729, New London, CT, November 14, 1976.

[25] M. Morf, A. Vieira, D. T. Lee, and T. Kailath, "Recursive Multichannel Maximum Entropy Spectral Estimation," *IEEE Trans. Geosci. Electron.*, Vol. GE-16, No. 2, April 1978, pp. 85–94.

[26] R. A. Wiggins and E. A. Robinson, "Recursive Solution to the Multichannel Filtering Problem," *J. Geophys. Res.*, Vol. 70, 1965, pp. 1885–1891.

[27] O. N. Strand, "Multichannel Complex Maximum Entropy Spectral Analysis," *1977 IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 736–741.

[28] O. N. Strand, "Multichannel Complex Maximum Entropy (Autoregressive) Spectral Analysis," *IEEE Trans. Autom. Control*, Vol. AC-22, No. 4, August 1977, pp. 634–640.

[29] L. W. Nolte and W. S. Hodgkiss, "Directivity or Adaptivity?" EASCON 1975 Record, *IEEE Electronics and Aerospace Systems Convention,* September 1975, Washington, DC, pp. 35.A–35.H.

[30] H. Cox, "Sensitivity Considerations in Adaptive Beamforming," Proceedings of the NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics,

August 21–September 1, 1972, Loughborough, UK; also in *Signal Processing*, edited by J. W. R. Griffiths, P. L. Stocklin, and C. van Schooneveld, Academic Press, New York, 1973.

[31] H. Cox, "Resolving Power and Sensitivity to Mismatch of Optimum Array Processors," *J. Acoust. Soc. Am.*, Vol. 54, No. 3, September 1973, pp. 771–785.

[32] D. Jaarsma, "The Theory of Signal Detectability: Bayesian Philosophy, Classical Statistics, and the Composite Hypothesis," Cooley Electronics Laboratory, University of Michigan, Ann Arbor, Technical Report TR-200, 1969.

[33] M. A. Gallop and L. W. Nolte, "Bayesian Detection of Targets of Unknown Location," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-10, No. 4, July 1974, pp. 429–435.

[34] W. S. Hodgkiss and L. W. Nolte, "Bayes Optimum and Maximum-Likelihood Approaches in an Array Processing Problem," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-11, No. 5, September 1975, pp. 913–916.

[35] R. H. Roy, "ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques," Ph.D. Dissertation, Stanford University, Stanford, CA, August 1987.

[36] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT—A Subspace Rotation Approach to Estimation of Parameters of Cissoids in Noise," *IEEE Trans. Acoust., Speech, & Signal Processing,* Vol. ASSP-34, No. 4, October 1986, pp. 1340–1342.

[37] R. Roy and T. Kailath, "ESPRIT—Estimation of Signal Parameters via Rotation Invariance Techniques," *IEEE Trans. Acoust., Speech, & Signal Processing,* Vol. ASSP-37, No. 7, July 1989, pp. 984–995.

[38] A. G. Jaffer, "Maximum Likelihood Direction Finding of Stochastic Sources: A Separable Solution," *1988 IEEE International Conf. Acoustics, Speech, and Signal Processing (ICASSP),* New York, April 1988, pp. 2893–2896.

[39] W. P. Ballance and A. G. Jaffer, "The Explicit, Analytic Cramer-Rao Bound on Angle Estimation," *Proc. Twenty-Second Asilomar Conf. Signals, Systems and Computers,* Pacific Grove, CA, November 1988, pp. 345–351.

[40] Harry L. Van Trees, *Detection, Estimation & Modulation Theory*, New York, Wiley, 1986, p. 66.

[41] L. Marple, "A New Autoregressive Spectrum Analysis Algorithm," *IEEE Trans. Acoustics, Speech & Signal Processing,* Vol. ASSP-28, No. 4, August 1980, pp. 441–454.

CHAPTER 12

Recent Developments in Adaptive Arrays

Chapter Outline

This chapter presents several innovations that have taken place since the first edition of this book. Wireless communication applications often resort to very simple beam switching, in which multiple beams simultaneously exist and the one with the best signal reception is selected. Moving radars or sonars must deal with clutter as well as interfering signals. Space-time adaptive processing (STAP) combines a spatial adaptive array with a temporal adaptive array to improve clutter cancellation and null placement. Another relatively recent development is multiple input, multiple output (MIMO) antenna array systems where an adaptive array is used for both transmit and receive to increase channel capacity. Reconfigurable antennas change their physical layout using switches to adapt for example the pattern, frequency response, and polarization response to match the desired signal. Partial adaptivity is of interest when only a portion of the total number of elements is controlled, thereby reducing the number of processors required to achieve an acceptable level of adaptive array performance.

12.1 BEAM SWITCHING

Multiple overlapping beam patterns pointing in slightly different directions can be formed in an array using hardware or software. For instance, Figure 12-1 shows a 10-element uniform array with elements spaced $\lambda/2$ apart with five orthogonal beams. Beams are orthogonal when the angle at which one beam has a peak and the rest have nulls (indicated

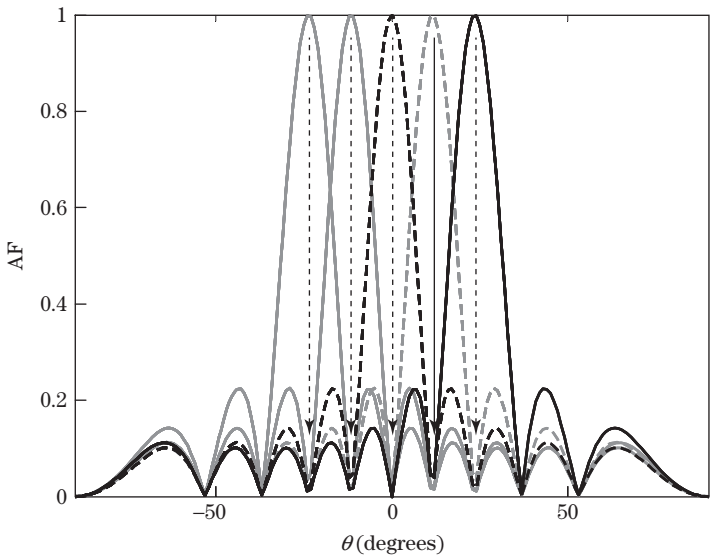


FIGURE 12-1 ■ Five orthogonal beams from a 10-element uniform array with half-wavelength spacing.

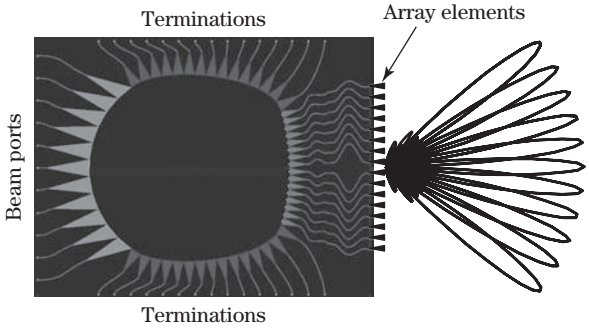


FIGURE 12-2 ■ Rotman lens with multiple beams (Courtesy of Remcom, Inc.). The lens is designed for microstrip implementation using the Rotman Lens Designer [2].

by dashed arrows in Figure 12-1). An algorithm continuously evaluates each beam and selects the one that maintains the highest signal quality. The system scans each beam output and selects the beam with the largest output power as well as suppresses interference.

Hardware configurations, such as the Rotman lens [1,2] (Figure 12-2) and Butler matrix [3] (Figure 12-3), have physical beam ports that correspond to a beam pointing in a direction determined by the passive feed network, element spacing, and number of elements. Each beam port in the Rotman lens receives a signal from all the elements. The different path lengths from the elements to the beam ports account for the phase shift that steers the beams. The Rotman lens in Figure 12-2 has a 16-element array with 11 beam ports. Each beam port has a pattern of a 16-element array steered to predetermined directions based on the geometry. The Butler matrix is a hardware version of a fast Fourier transform (FFT) [4]. Each port receives the signal from one of the beams. A switch selects the desired beam formed by these hardware beamformers. If an array has a digital beamformer, then the beams are formed and selected in software. Usually, the beams cover a desired azimuth range.

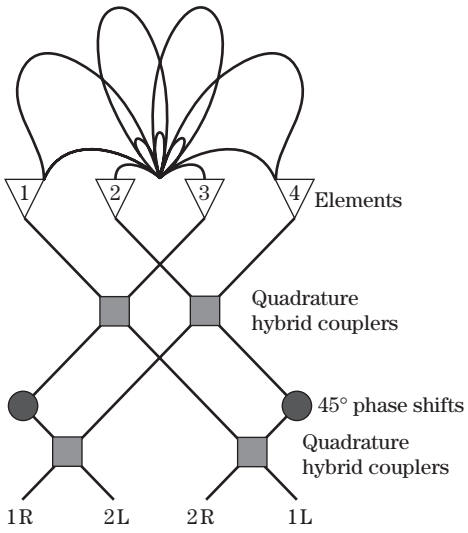


FIGURE 12-3 ■ Butler matrix with four beams.

12.2 SPACE-TIME ADAPTIVE PROCESSING

Not only environmental noise, jamming, and unintentional interfering signals but also clutter entering the sidelobes and main beam prevent detection of a radar signal. Airborne phased array radar data normally has moving target signals embedded in undesired signals, such as clutter, broadband noise jamming, and thermal noise. A moving target indicator (MTI) radar detects a moving target by isolating the Doppler frequency given by [5]

$$f_D = \frac{2v_a \cos\phi}{\lambda} \tag{12.1}$$

where v_a is the velocity of the aircraft, and ϕ is the angle measured from the velocity vector. Figure 12-4 is a plot of the received signal power as a function of azimuth angle and Doppler frequency. The peak of the Doppler clutter occurs normal to the direction of velocity and at zero Doppler frequency. The interference occurs at a single angle but over

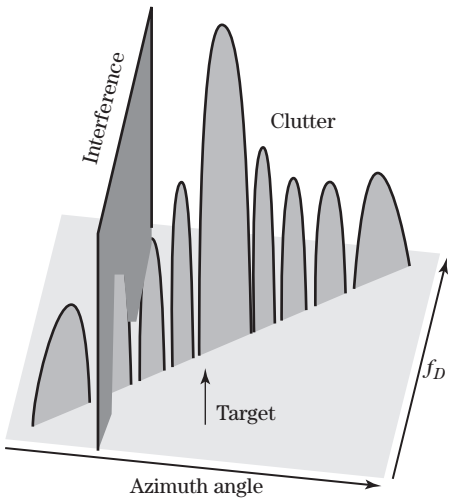


FIGURE 12-4 ■ A plot of the target, clutter, and interference returns as a function of angle and frequency.

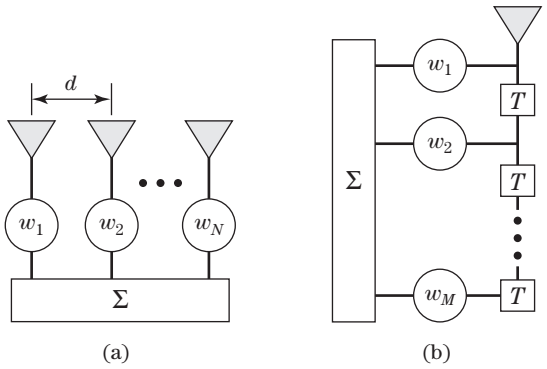


FIGURE 12-5 ■ Linear array types. a: Spatial. b: Temporal.

all frequencies. The motion of a radar platform spreads the clutter in Doppler frequency. The Doppler frequency from clutter at a specific point on the ground depends on the angle of the clutter position relative to the heading of the platform. Interference from a discrete source appears at one angle but is spread over all Doppler frequencies.

A spatial adaptive array weights and combines the signals received by the array elements at the same instant in time but at different spatial locations separated by distance d (Figure 12-5a). A temporal adaptive array combines signals received at the same spatial location but sampled at different instances in time separated by time T (Figure 12-5b).

A displaced phase center antenna (DPCA) cancels clutter induced by platform motion in an MTI radar [6]. The idea is to make the antenna appear stationary over the transmitted pulse train by electronically shifting the phase center of the receive aperture backward to compensate for the forward motion of the moving platform. The DPCA was first envisioned for a rotating monopulse radar [7]. By adding and subtracting the output from the azimuth difference channel to the output of the azimuth sum channel, a fore and aft beam are formed. If the output from the aft beam is subtracted from the output from the fore beam at the same pointing angle, then the clutter return would be canceled.

A better implementation is based on synthetic aperture radar (SAR). When the velocity vector of the platform is parallel to the linear array axis, then the pulse repetition frequency (PRF) is adjusted to the platform velocity so that the first, second, and subsequent elements at the current pulse appear to move to the respective positions of the second, third, and subsequent elements at the previous pulse [8]. Figure 12-6 shows a four-element array split into two three-element arrays: fore and aft. The full aperture transmits a pulse train with N_t pulses at $t = 0$. Both the fore and aft arrays receive one pulse, then the array moves in space a distance d, and the fore and aft arrays receive another pulse. Once the array moves forward by $(M - 1)d$, then the full aperture transmits another pulse train of N_t pulses. Figure 12-7 shows the phase centers of the full, fore, and aft apertures. They

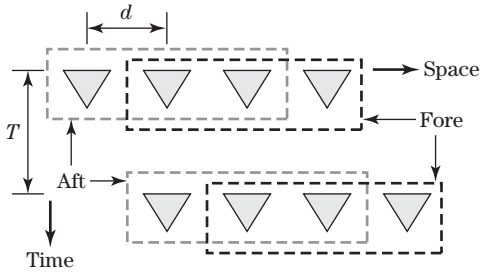


FIGURE 12-6 ■ The four-element array is split into fore and aft arrays. The array moves a distance equal to the element spacing in time T.

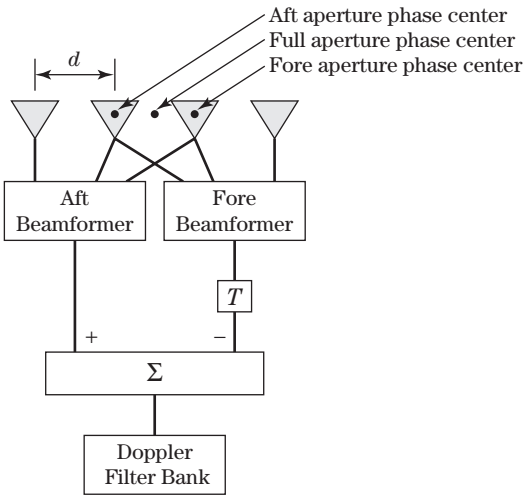


FIGURE 12-7 ■ The phase centers associated with the fore, aft, and full arrays.

are equally spaced by $d/2$. Delaying the fore aperture output by T effectively moves the fore aperture phase center to correspond to the aft aperture phase center. Thus, the array moves a distance $d/2$ in one pulse repetition interval (PRI), so [8]

$$v_a = \frac{d}{2PRI} \tag{12.2}$$

The voltage output from the aft and fore arrays are given by

$$\begin{aligned} \text{Aft beam:} \quad & AF_A(\phi) = e^{-j\frac{3\pi}{c} f d \cos\phi} + e^{-j\frac{\pi}{c} f d \cos\phi} + e^{j\frac{\pi}{c} f d \cos\phi} \\ \text{Fore beam:} \quad & AF_F(\phi) = e^{-j\frac{\pi}{c} f d \cos\phi} + e^{j\frac{\pi}{c} f d \cos\phi} + e^{j\frac{3\pi}{c} f d \cos\phi} \end{aligned} \tag{12.3}$$

Assuming that the clutter does not change from pulse to pulse, when the fore beam is delayed by T, then it is identical to the aft beam as shown by

$$\begin{aligned} AF_F(\phi)e^{-j2\pi f_d T} &= AF_F(\phi)e^{-j2\pi\left(\frac{2v_a\cos\phi}{\lambda}\right)\left(\frac{d}{2v_a}\right)} \\ &= AF_F(\phi)e^{-j\frac{2\pi}{c} f d \cos\phi} \\ &= e^{-j\frac{3\pi}{c} f d \cos\phi} + e^{-j\frac{\pi}{c} f d \cos\phi} + e^{j\frac{\pi}{c} f d \cos\phi} \\ &= AF_A(\phi) \end{aligned} \tag{12.4}$$

Thus, subtracting the fore aperture output delayed by T from the aft aperture output should cancel the clutter, which does not change much from pulse to pulse.

DCPA works perfectly when there are no errors. Some practical problems with DCPA are as follows [9]:

1. The platform velocity does not perfectly match the PRI of the radar.
2. Clutter can change from position to position.
3. Error tolerances in the antenna components cause the signal path at each element to be different.
4. Antenna calibration is necessary to compensate for thermal noise and aging of components.
5. Unwanted platform motion causes deviations from the desired velocity vector.

An adaptive DCPA algorithm (ADPCA) [10] is not an optimal processor, so it can have significant signal-to-interference plus noise ratio (SINR) loss. DCPA was implemented on an advanced development model called Pave Mover and then was followed by the development of the Joint Surveillance and Target Attack Radar System (Joint STARS) [11]. Joint STARS uses a 24 ft. long, 2 ft. high phased array antenna (Figure 12-8) mounted on the forward underfuselage of an Air Force E-8A (Figure 12-9).

STAP enables radars and sonars to detect targets obscured by clutter and jamming. It was first envisioned by Brennan and Reed [12,13] as an adaptive array for MTI radar. STAP is an improvement over DCPA, because it integrates the spatial adaptive array with the temporal adaptive array. It improves clutter cancellation performance and integrates spatial processing (sidelobe control and null placement) with clutter cancellation. STAP also has

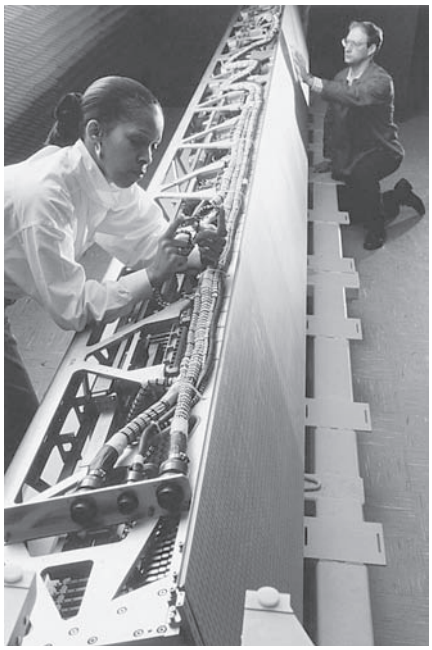


FIGURE 12-8 ■ Photograph of the phased array used for Pave Mover and Joint STARS (Courtesy of the National Electronics Museum).

FIGURE 12-9 ■ Picture of array mounted beneath the USAF E-8A (Courtesy of the USAF Strategic Air Command).

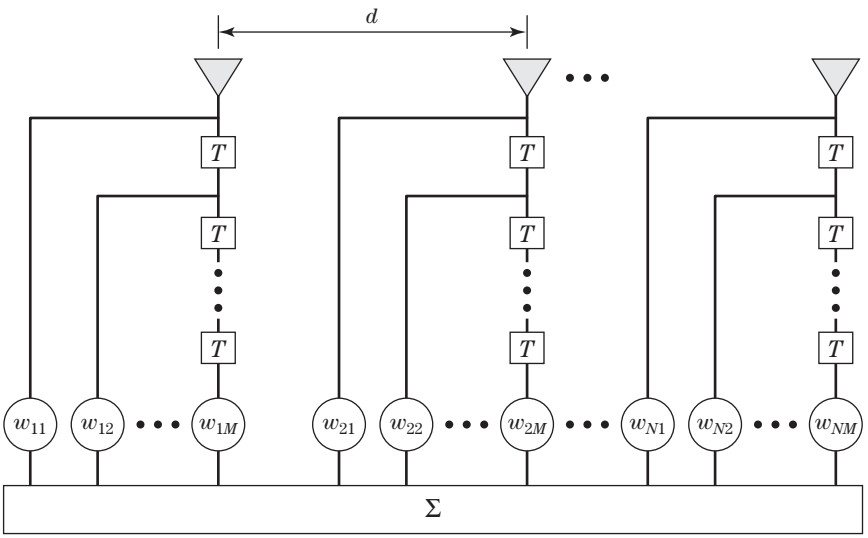


FIGURE 12-10 ■ Diagram of a STAP antenna.

application to sonar reverberation suppression and simultaneous position and Doppler estimation using high-resolution space-time matched field-processing techniques [8].

STAP is a two-dimensional (angle–frequency) adaptive filter that maximizes the output SINR by weighting the space and time samples of the array as shown in Figure 12-10. The element locations are the filter's spatial dimension, and the PRI is the temporal dimension. STAP forms a beam in the direction of the target at the same time it creates a frequency filter about the target spectrum and puts nulls in the directions and at the frequencies of interferers. It was originally developed to detect slow-moving targets from airborne radars. The radar transmits N_t coherent pulses that are received by N_r elements of an array and sampled at L range gates over several pulse repetition intervals. A snapshot is the $M \times N$ sample matrix collected at each range gate. The snapshots from each range gate are stacked into a data cube. STAP adaptively weights input signal samples to eliminate the noise, clutter, and interference received from the environment. These weights, at one instant of time (at the same time delay), form an antenna pattern in space with nulls placed in the directions of interfering sources. Applied to the signal samples at one antenna array element over the entire dwell, the STAP weights define a system impulse response. The clutter spectrum for ground-based radars has a peak at zero Doppler, whereas the clutter spectrum for moving platform radars depends on frequency. STAP adapts the frequency response to the clutter spectrum to receive the desired signal while rejecting clutter.

Figure 12-11 is an example of simulated radar returns as a function of angle and Doppler when both clutter and a jammer are present. The jammer appears stationary at one angle but exists over a wide frequency range. Clutter, on the other hand, has a maximum at zero Doppler and extends over all angles as a linear function of frequency. Applying a STAP filter lowers the clutter and jammer returns while increasing the noise and target as shown by the picture on the right in Figure 12-11.

STAP has the following advantages:

- Increased detection of low-velocity targets by better suppression of main beam clutter
- Increased detection of small cross section targets that might otherwise be obscured by sidelobe clutter

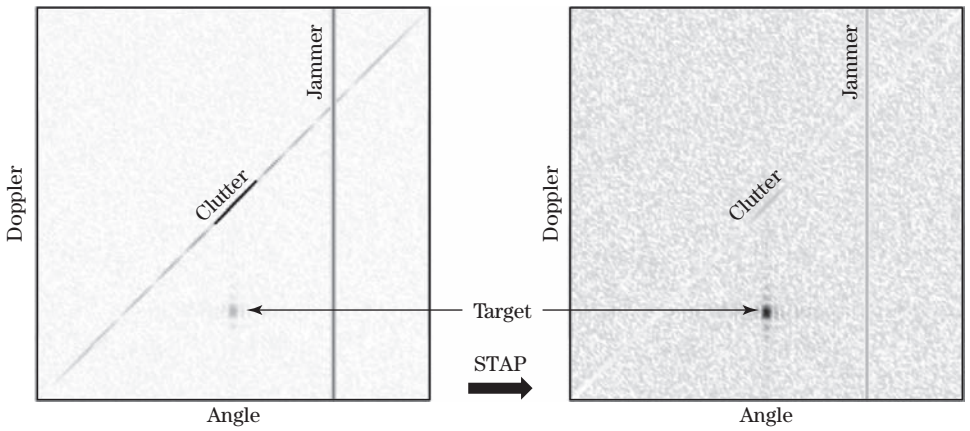


FIGURE 12-11 ■ STAP reduces clutter and jammer power as seen by the before (left) and after (right) angle–Doppler plots.

- Increased detection in combined clutter and jamming environments
- Robust to radar system errors
- Means to handle nonstationary interference

STAP's big disadvantage is the high computational cost [14]. There are two major parts of the computation: (1) formation of the covariance matrix; and (2) the inversion of that matrix. Data from a large number of adjacent range cells are required to compute the covariance matrix of the undesired signals. Assuming that the structure of the covariance matrix is known helps speed the calculation of the covariance matrix. Inverting the covariance matrix is the dominant computation in STAP, because the inversion is a nonlinear transform and is not conducive to parallel processing. Numerous algorithms that reduce the data sample size required or the computational cost have been proposed. Most of these approaches are only partially adaptive.

The STAP processor finds the weights in real time by solving an $NM \times NM$ system of linear equations. The new weights require on the order of $(NM)^3$ operations [14]. Airborne radars constantly scan space searching for targets, so computational throughputs on the order of hundreds of billions of floating-point operations per second with execution speeds of fractions of a second are needed. As a result, current research work on STAP focuses on developing algorithms that decompose the fully adaptive problem into reduced-dimension, real-time adaptive problems capable of being implemented in real time on reasonably sized processors. Space and power on an aircraft are premium commodities, so processing performance per unit size, power, and weight are important [14].

The following analysis follows that in [8]. An estimate of the covariance matrix from a set of radar data is called the training set. The covariance matrix is estimated from

$$\hat{\mathbf{R}} = \mathbf{X}^{\dagger}\mathbf{X} \tag{12.5}$$

where $\mathbf{X}$, the training set matrix, is a subset of the input data. The weights are computed using sample matrix inversion (SMI) and finds the weights using direct matrix inversion or a factorization approach that computes the Cholesky decomposition of $\mathbf{R}$ via QR decomposition of $\mathbf{X}$ (see Chapter 6).

If $\mathbf{X}$ can be factored so that $\mathbf{X} = \mathbf{UA}$, where $\mathbf{A}$ is upper triangular, then the covariance matrix is written as

$$\hat{\mathbf{R}} = \mathbf{A}^{\dagger}\mathbf{U}^{\dagger}\mathbf{UA} = \mathbf{A}^{\dagger}\mathbf{A} \tag{12.6}$$

Since $\mathbf{U}$ is an orthogonal, unitary matrix, $\mathbf{U}^{\dagger}\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Since $\mathbf{A}$ is triangular, The weights are easily computed from by back-solving

$$\begin{aligned} \mathbf{A}^{\dagger}\mathbf{u} &= \mathbf{s} \\ \mathbf{A}\mathbf{w} &= \mathbf{u} \end{aligned} \tag{12.7}$$

where $\mathbf{s}$ is the target response at one angle and Doppler frequency, and $\mathbf{u}$ is an intermediate computation vector. Each member of the STAP filter bank has a different $\mathbf{s}$. Beamforming is the product of the weight vector and the input data for a specific range gate.

$$\mathbf{z} = \mathbf{w}^{\dagger}\mathbf{y} \tag{12.8}$$

The covariance matrix of a fully adaptive STAP algorithm, in which a separate adaptive weight is applied to N_t pulse-repetition intervals as well as N_r elements, has dimensions $NM \times NM$ where $200 \leq NM \leq 10^5$.

The noise interference vector, $\mathbf{q}$, is a zero mean random vector with a multivariate complex Gaussian distribution represented by the sum of the uncorrelated noise and the correlated interference (e.g., jammer, clutter).

$$\mathbf{q} = \mathbf{c} + \mathbf{n} \tag{12.9}$$

The noise interference vector covariance matrix for a STAP array is given by [8]

$$\mathbf{R}_{qq} = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{M-2} & \mathbf{Q}_{M-1} \\ \mathbf{Q}_1 & \mathbf{Q}_0 & & & \mathbf{Q}_{M-2} \\ \vdots & & \ddots & & \vdots \\ \mathbf{Q}_{M-2}^{*} & & & \mathbf{Q}_0 & \mathbf{Q}_1 \\ \mathbf{Q}_{M-1}^{*} & \mathbf{Q}_{M-2}^{*} & \cdots & \mathbf{Q}_1^{*} & \mathbf{Q}_0 \end{bmatrix} \tag{12.10}$$

where $\mathbf{Q}_m$ are $N \times N$ spatial covariance submatrixes measured between pulses separated in time by mT [8].

$$\mathbf{Q}_m = E\{\mathbf{q}(nT)\mathbf{q}^{*}((n+m)T)\} = E\{\mathbf{q}(0)\mathbf{q}^{*}(mT)\} = \begin{bmatrix} q_{11}(mT) & \cdots & q_{1N}(mT) \\ \vdots & \ddots & \vdots \\ q_{N1}(mT) & \cdots & q_{NN}(mT) \end{bmatrix} \tag{12.11}$$

As with spatial adaptive nulling, the optimum weights are found from

$$\mathbf{w}_{\text{opt}} = \gamma \mathbf{R}_{qq}^{-1}\mathbf{s} \tag{12.12}$$

The initial nonadaptive filtering can be either a transformation into the frequency domain (e.g., by performing an FFT, over pulses in each channel) or a transformation into beam space (e.g., by performing nonadaptive beamforming in each pulse). We can perform both spatial and temporal transformations, if desired, or we can eliminate nonadaptive filtering altogether.

The nonadaptive filtering determines the domain (frequency or time, element or beam) in which adaptive weight computation occurs. Figure 12-12 is a diagram representing the data domain for a single range gate after a different type of nonadaptive transform [14,15]. For example, the upper right quadrant represents PRI data that have been transformed into Doppler space. Thus, each sample is a radar return for a specific Doppler frequency and receiver element. The lower left quadrant represents element data that have been transformed into beam space. Each sample is a radar return for a specific PIR and look

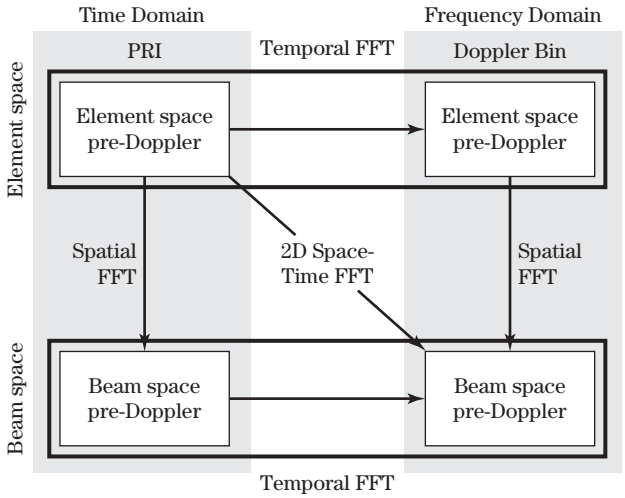


FIGURE 12-12 ■ STAP algorithm transformations involve either one-dimensional space and time transforms or two-dimensional space–time transform.

direction. For example, a STAP kernel that is adaptive in the frequency domain falls into the element-space post-Doppler quadrant of the taxonomy. A Doppler filter-bank FFT transforms the signals from each element. Low-sidelobe Doppler filtering localizes competing clutter in angle to reduce the extent of clutter to be adaptively canceled. The adaptation occurs over all elements and a number of Doppler bins. The number of Doppler bins is a parameter of the element-space post-Doppler algorithm. Factored post-Doppler algorithms perform spatial adaptation in a single bin and are not adaptive in time.

A partially adaptive STAP algorithm breaks down the fully adaptive STAP problem into a number of independent, smaller, and more computationally tractable adaptive problems while achieving near-optimum performance [8,14]. Figure 12-13 shows that a

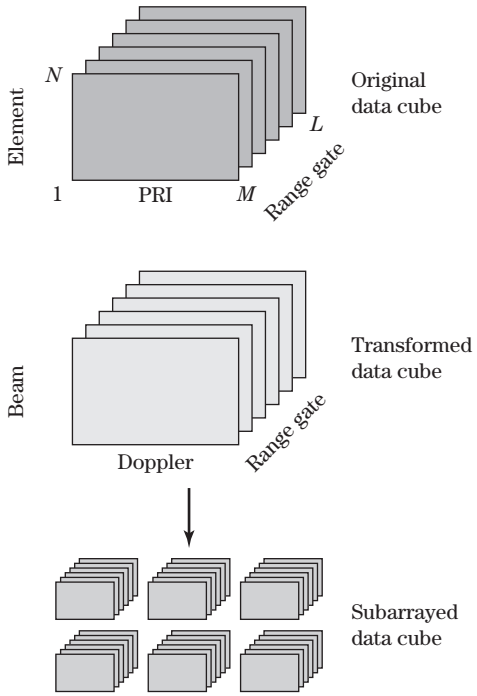


FIGURE 12-13 ■ STAP enhances the target while suppressing the clutter and interference. The picture to the left is before convergence, and the picture to the right is after convergence.

partially adaptive algorithm starts by nonadaptive filtering of the input signal data to reduce dimensionality. Once the input data are transformed and bins and beams (or channels and pulse repetition intervals) are selected to span the target and interference subspaces, multiple separate adaptive sample matrix inversion problems are solved, one for each Doppler frequency bin or pulse repetition interval, across either antenna elements or beams, depending on the domain of the adaptation.

Some current STAP research focuses on the following [16]:

1. Bistatic configurations where the transmit and receive platforms move separately to keep the receiving platform covert
2. Conformal arrays
3. Nonstationary received signals that arise from bistatic STAP, conformal arrays, terrain with different reflection coefficients, and clutter motion like vegetation blowing in the wind.
4. Knowledge-aided STAP, which attempt to remove as much of the heterogeneity from the snap shots prior to using conventional estimation methods. This is done by using a priori knowledge, typically stored in databases.
5. STAP applications in sonar, telecommunications, and detection of plastic landmines.

12.3 MIMO

A communications system falls into one of four categories shown in Figure 12-14 based on the number of antennas used for transmit and receive:

1. A single input, single output (SISO) system has one antenna for transmit and one antenna for receive. SISO is the most common communications system category and offers no spatial diversity.
2. A single input, multiple output (SIMO) system has one antenna for transmit and multiple antennas for receive. This configuration is common when an antenna array is used on receive.
3. A multiple input, single output (MISO) system has multiple antennas for transmit and one antenna for receive.
4. A MIMO system has multiple antennas for transmit and multiple antennas for receive. This configuration offers the greatest spatial diversity and control but also has the greatest hardware cost.

Adaptive antennas have traditionally been SIMO. Some work has been done in MISO systems, but today considerable attention has fallen on MIMO systems because they have

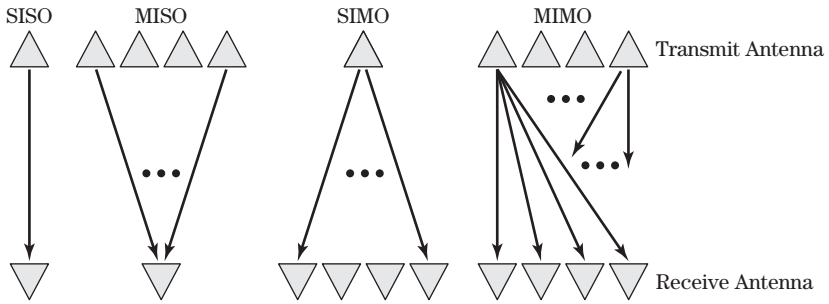


FIGURE 12-14 Communications system categories based on number of antennas used for transmit and receive.

the greatest promise to deliver the highest capacity. MIMO has also found use in acoustic arrays [17].

In a SISO system with two isotropic point sources separated by R and no multipath, if the transmitted signal is s, then the received signal, r, is given by

$$r = \frac{se^{-j\frac{2\pi}{\lambda}R}}{R} \tag{12.13}$$

If the signal also takes one bounce from an object with a reflection coefficient, Γ, then the received signal includes a second path of distance R_1.

$$r = \frac{se^{-j\frac{2\pi}{\lambda}R}}{R} + \frac{\Gamma se^{-j\frac{2\pi}{\lambda}R_1}}{R_1} \tag{12.14}$$

As the number of scattering objects increase, the number of paths increases until the multipath formulation is given by

$$r = \frac{se^{-j\frac{2\pi}{\lambda}R}}{R} + \sum_{m=1}^{M} \frac{\Gamma_m se^{-j\frac{2\pi}{\lambda}R_m}}{R_m} \tag{12.15}$$

Multiple paths turn the channel transfer function, h, into a random Gaussian variable, so (12.15) becomes

$$r = hs \tag{12.16}$$

When the transmit and receive antennas are arrays, then each of the N_t transmit elements send a signal to each of the N_r receive elements. Figure 12-15 shows a diagram of a MIMO system [18]. To take interactions between the elements in the transmit and receive arrays into account, (12.16) is written into matrix form

$$\mathbf{r} = \mathbf{Hs} + \mathbf{N} \tag{12.17}$$

where $\mathbf{N}$ is a noise vector, and $\mathbf{H}$ is the channel matrix given by (Figure 12-16)

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & & \vdots \\ & & \ddots & \\ h_{M1} & \cdots & & h_{MN} \end{bmatrix} \tag{12.18}$$

The SVD of $\mathbf{H}$ is given by

$$\mathbf{H} = \mathbf{U}_{SVD}\mathbf{D}\mathbf{V}_{SVD}^{\dagger} \tag{12.19}$$

$$\mathbf{D} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sqrt{\lambda_M} \end{bmatrix}$$

$$\sqrt{\lambda_m} = \text{singular values}$$

$$\mathbf{U}_{SVD}, \mathbf{V}_{SVD} = \text{singular vectors}$$

The columns of $\mathbf{U}_{SVD}$ are the receive weight vectors corresponding to the singular values, and columns of $\mathbf{V}_{SVD}$ are the transmit weight vectors corresponding to the singular values [19]. Since the channel is reciprocal, exactly the same weights may be used for transmission

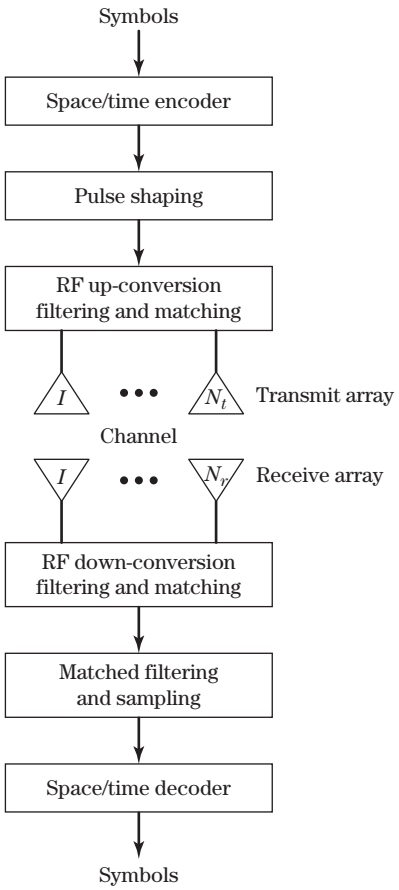


FIGURE 12-15 ■ Diagram of a MIMO system.

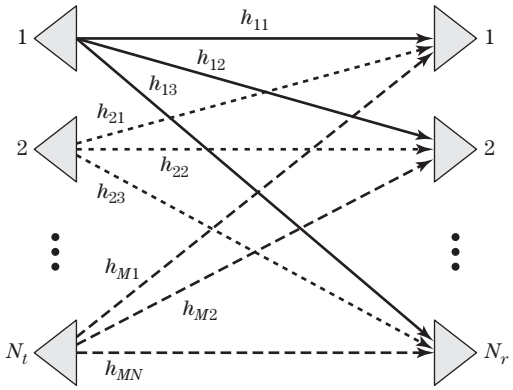


FIGURE 12-16 ■ Signal paths between transmit and receive antennas in a MIMO system.

as for reception. Beamforming using the singular vectors as array weights produces *eigenpatterns* that create independent (spatially orthogonal) parallel communication channels in the multipath environment.

Alternatively, (12.17) can be written in terms of power such that the power received by the array is given by

$$P = \mathbf{r}^{\dagger}\mathbf{r} = \mathbf{s}^{\dagger}\mathbf{H}\mathbf{H}^{\dagger}\mathbf{s} \tag{12.20}$$

An eigenvalue decomposition of the $N_r \times N_r$ correlation matrix $\mathbf{HH}^\dagger$ is written as

$$\mathbf{HH}^\dagger = \mathbf{R}_H = \mathbf{V}_\lambda \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_M \end{bmatrix} \mathbf{V}_\lambda^\dagger \tag{12.21}$$

where V_λ are the eigenvectors, and λ_m are the eigenvalues. The off-diagonal elements of $\mathbf{R}_H$ represent the correlation between the transmitted signal streams, with increased correlation resulting in decreased capacity. The eigenvalue represents the received signal power level in the the eigenchannel. Once an accurate estimate of $\mathbf{H}$ is established, then the transmitted data, $\mathbf{s}$, is calculated from the received data by inverting the channel matrix.

$$\mathbf{s} = \mathbf{H}^{-1}\mathbf{r} \tag{12.22}$$

In free space with no multipath, the elements of $\mathbf{H}$ are the free-space Greens function

$$\mathbf{H} = \begin{bmatrix} \dfrac{e^{-jkR_{11}}}{R_{11}} & \dfrac{e^{-jkR_{mn}}}{R_{mn}} & \cdots & \dfrac{e^{-jkR_{mn}}}{R_{mn}} \\ \dfrac{e^{-jkR_{21}}}{R_{21}} & \dfrac{e^{-jkR_{mn}}}{R_{mn}} & & \\ \vdots & & \ddots & \vdots \\ \dfrac{e^{-jkR_{mn}}}{R_{MN}} & & \cdots & \dfrac{e^{-jkR_{mn}}}{R_{mn}} \end{bmatrix} \tag{12.23}$$

The number of data streams supported must be less than or equal to the rank of $\mathbf{H}$. The rank of a matrix is the number of nonzero singular values. $\mathbf{H}$ is ill conditioned as presented in (12.23). Increasing the separation between array elements or adding random components to the matrix elements decreases the matrix condition number and improves the accuracy of inverting $\mathbf{H}$. Multipath adds random components, so it significantly improves the ability of inverting $\mathbf{H}$. Low-rank MIMO channels are associated with minimal multipath or large separation distance between the transmit and receive antennas [20]. The low-rank MIMO channel is equivalent to a SISO channel with the same total power. High-rank MIMO channels are associated with a high multipath environment, and the separation distance between transmit and receive antennas is small. As we pack more antennas into our array, the capacity per antenna drops due to higher correlation between adjacent elements. MIMO systems perform best with a full rank channel matrix, which means a low correlation between signals on the different antennas.

The instantaneous eigenvalues have limits defined by [19]

$$\left(\sqrt{N_t} - \sqrt{N_r}\right)^2 < \lambda_n < \left(\sqrt{N_t} + \sqrt{N_r}\right)^2 \tag{12.24}$$

When N_t is much larger than N_r, then all the eigenvalues cluster around N_t. Each eigenvalue is nonfading due to the high-order diversity. Thus, the uncorrelated asymmetric channel with many antennas has a very large theoretical capacity of N_r equal, constant channels with gains of N_t.

Each element of a MIMO transmit array uses the same frequency spectrum but may transmit symbols from different modulation schemes and carry independent data streams.

The propagation channel is assumed to be Rayleigh and unknown to the transmitter. Each burst contains a training sequence that allows the receiver to get an accurate estimate of the propagation conditions. The channel may change from one burst to the next due to motion of the transmitter or receiver.

When the channel is unknown at the transmitter, the power is uniformly distributed over the antennas

$$P_n = P/N_t \tag{12.25}$$

When the channel is known, then water filling determines the power allocation to the channels [21]. In water filling, more power is allocated to better subchannels with higher SNR to maximize the sum of data rates in all subchannels. Since the capacity is a logarithmic function of power, the data rate is usually insensitive to the exact power allocation, except when the SNR is low. Assuming all noise powers to be the same, water filling is the solution to the maximum capacity, where each channel is filled up to a common level ℓ

$$\frac{1}{\lambda_n} + P_n = \ell \quad \text{where} \quad \sum P_n = P \tag{12.26}$$

The capacity difference between the known and unknown channels is small for large P. Thus, the highest-gain channel receives the largest share of the power. In the limit where the SNR is small ($p < 1/\lambda_2 - 1/\lambda_1$), only one eigenvalue, the largest, is left.

The capacity of a MIMO system depends on the propagation environment, array geometry, and antenna patterns. If the sources are uncorrelated and equal power and the channel is random, then the ergodic (mean) capacity is given by [18]

$$C = E\left\{\log_2\left[\det\left(I_N + \frac{P}{N_t\sigma_n^2}\mathbf{H}\mathbf{H}^\dagger\right)\right]\right\} \text{ bits/s/Hz} \tag{12.27}$$

MIMO capacity increases linearly with the number of elements when the number of transmit and receive antennas are the same. Winters suggests that N_t should be of the order $2N_r$ [22]. When N_t and N_r are large and $N_t > N_r$, the capacity is [19]

$$C = N_r \log_2(1 + N_t P/N_r) \tag{12.28}$$

As long as the ratio of N_t/N_r is constant, then the capacity is a linear function of N_r.

The vertically layered blocking structure (V-BLAST) algorithm was developed at Bell Laboratories for spatial multiplexing [20]. It splits a single high data rate data stream into N_t lower rate substreams in which the bits are mapped to symbols. These symbols are transmitted from N_t antennas. Assuming the channel response is constant over the system bandwidth (flat fading), the total channel bandwidth is a fraction of the original data stream bandwidth. The receive array has an adaptive algorithm where each substream is a desired signal while the rest are nulled. As a result, the receive array forms N_t beams while simultaneously creating nulls. The received substreams are then multiplexed to recover the original transmitted signal. V-BLAST transmits spatial uncoded data streams without equalizing the signal at the receiver. VB cannot separate the streams and suffers from multistream interferences (MSI). Thus, the transmission is unsteady, and forward error coding is not always able to resolve this issue.

Space-time codes deliver orthogonal and independent data streams. Orthogonal frequency division multiplexing (OFDM) is commonly used in MIMO systems [21]. It splits

the high-speed serial data to be transmitted into many much lower-speed serial data signals that are sent over multiple channels. The bit or symbol periods are longer, so multipath time delays are not as deleterious. Increasing the subcarriers and bandwidth makes the signal immune from multipath. When the subcarrier spacing equals the reciprocal of the symbol period of the data signals, they are orthogonal. The resulting sinc frequency spectra have their first nulls at the subcarrier frequencies on the adjacent channels.

The channel matrix is found through measurement or computer model [18]. A switched array design employs a single transmitter and single receiver to measure **H** by using high-speed switches to sequentially connect all array combinations of array elements. Switching times range from 2 to 100 ms, so the measurement of all antenna pairs is possible before the channel appreciably changes for most environments. Virtual array instruments displace or rotate a single antenna element. This method eliminates mutual coupling effects, but a complete channel matrix measurement takes several seconds or minutes, so the channel must remain stationary over that time. As a result, virtual arrays work best for fixed indoor measurements when activity is low.

Channel models compute **H** based on a statistics. It is common practice to assume that the transfer function between one transmit and one receive antenna has Rayleigh magnitude and uniform phase distributions for a non-line-of-sight (NLOS) propagation environment. Ray tracing programs can be used to calculate **H**. Figure 12-17 shows an

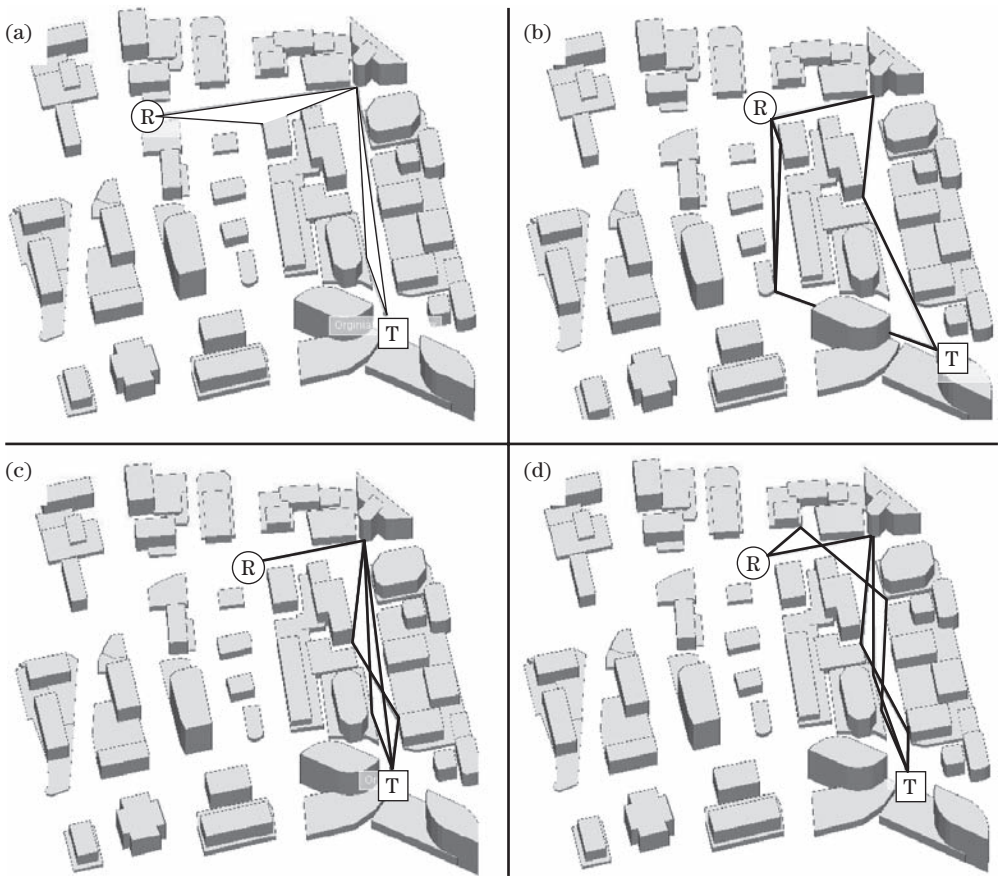


FIGURE 12-17 ■ Ray tracing in a MIMO system (Courtesy of Remcom, Inc.).

example of ray tracing from a transmit antenna to a receive antenna in a city when the transmit antenna and receive antenna are not line of sight [22]. Figure 12-17a is an example when only two bounces occur between the transmit and receive antenna. The signal paths in Figure 12-17b take two very different routes. Figure 12-17c and Figure 12-17d are examples of paths that take many bounces. Signal paths are highly dependent on the environment and the positions of the transmitters and receivers. Small changes in position as shown in Figure 12-17 create large changes in **H**.

12.4 | RECONFIGURABLE ANTENNAS AND ARRAYS

Antennas reconfigure or alter the current flow paths using switches to change, for example, frequency and polarization. For example, two ultra wide band (UWB) monopoles with reconfigurable band notch in the wireless LAN frequency range (5.150–5.825 GHz) are shown in Figure 12-18 [23]. The antenna is an elliptical patch fed with coplanar waveguide line. The U-shaped slot is approximately $\lambda/2$ long and has a frequency notch when the micro electro mechanical systems (MEMS) switch is open but not when the MEMS switch is closed. When the switch is open at 5.8 GHz, the currents in the inner and outer side of the slot flow in opposite directions and cancel each other. When the MEMS switch is closed, the slot is shorted at its center point, so the total length of the slot is cut in half. Consequently, the slot no longer supports the resonating currents and radiation occurs as if the slot was not present. The second antenna has two inverted L-shaped open stubs that connect and disconnect the stubs from the elliptical patch via MEMS switches. Shorting the stubs to the patch creates a rejection band. At the resonance frequency, the direction of the current on the stub flows in the opposite direction to the current along the nearby edge of the radiator, so the radiated fields cancel. When the stubs are not shorted, the antenna operates over the whole UWB range (3.1–10.6 GHz) without any rejection band. The MEMS switches actuate through the radiofrequency (RF) signal path, without any direct current (DC) bias lines, that might complicate the switch fabrication and integration while degrading the radiation performance due to RF leakage through the bias.

Reconfigurable antennas can also change the antenna polarization. For instance, placing MEMS switches in the feeds of microstrip antennas provides the ability to switch

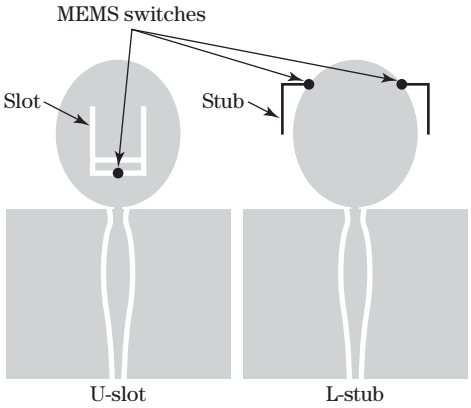


FIGURE 12-18 ■ Diagram of the reconfigurable U-slot and L-stub antennas.

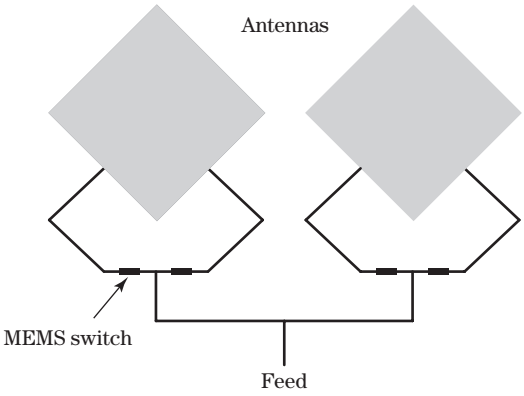


FIGURE 12-19 ■ Reconfigurable antenna that change polarization using MEMS.

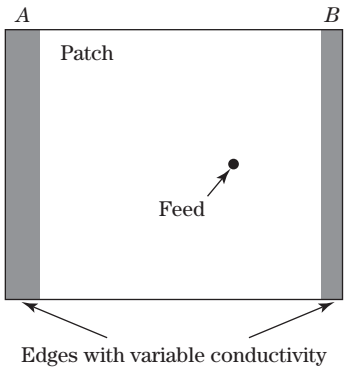


FIGURE 12-20 ■ Patch with edges made from variable conductivity material.

from one linear polarization to the orthogonal linear polarization or to circular polarization (Figure 12-19) [24].

A patch resonates at several frequencies if conducting extensions to the edges are added or removed [26]. The patch antenna in Figure 12-20 lies on a 3.17 mm thick polycarbonate substrate ($\varepsilon_r = 2.62$), and its edges at A and B made from a material with variable conductivity, such as silicon. The size of the copper patch as well as the width of A and B are optimized along with the feed location to produce a 50Ω input impedance at 1.715, 1.763, 1.875, and 1.930 GHz, depending on whether an edge is conductive. Graphs of the reflection coefficient, s_{11}, from 1.6 to 2.0 GHz for the four cases are shown in Figure 12-21. Each combination has a distinct resonant frequency.

The five-element array in Figure 12-22 has rectangular patches made from a perfect conductor that is 58.7 × 39.4 mm [25]. The 88.7 × 69.4 × 3 mm optically transparent fused quartz substrate ($\varepsilon_r = 3.78$) is backed by a perfectly conducting ground plane. The patch has a thin strip of silicon 58.7 × 2 mm with $\varepsilon_r = 11.7$ that separates the narrow right conducting edge of the patch (58.7 × 4.2 mm) from the main patch. A light-emitting diode (LED) beneath the ground plane illuminates the silicon through small holes in the ground plane or by making the ground plane from a transparent conductor. The silicon conductivity is proportional to the light intensity. A graph of the amplitude of the return loss is shown in Figure 12-23 when the silicon is 0, 2, 5, 10, 20, 50, 100,

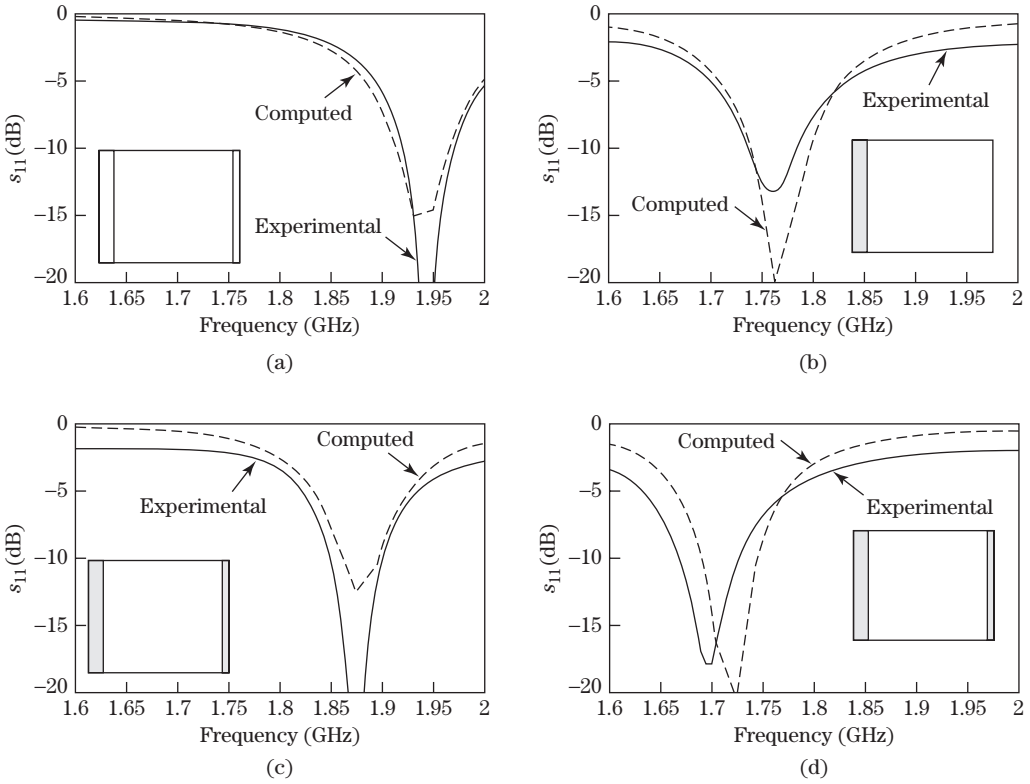


FIGURE 12-21 ■ Reflection coefficient (s_{11}) versus frequency for the following: a: No edge. b: Back edge. c: Front edge. d: Both edges are conductive.

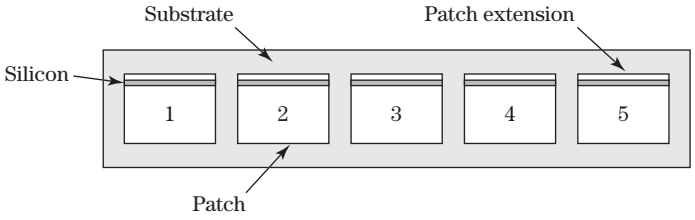


FIGURE 12-22 ■ A five-element linear array of photoconductive patches.

200, and 1,000 S/m. There is a distinct resonance at 2 GHz when the silicon has no conductivity. Increasing the conductivity gradually changes the resonance to 1.78 GHz. As the silicon conductivity increases, the resonant frequency of the patch changes, so the photoconductive silicon acts as an attenuator. The element spacing in the array is 0.5λ. Carefully tapering the illumination the LEDs creates so that the silicon conductivity at the five elements is [16 5 0 5 16] S/m results in the antenna pattern in Figure 12-24. It has a gain of 10.4 dB with a peak relative sidelobe level 23.6 dB.

An experimental adaptive array was constructed with photoconductive attenuators integrated with broadband monopole antennas (Figure 12-25) [26]. A close-up of one of the array elements appears in Figure 12-26. The attenuator consists of a meandering

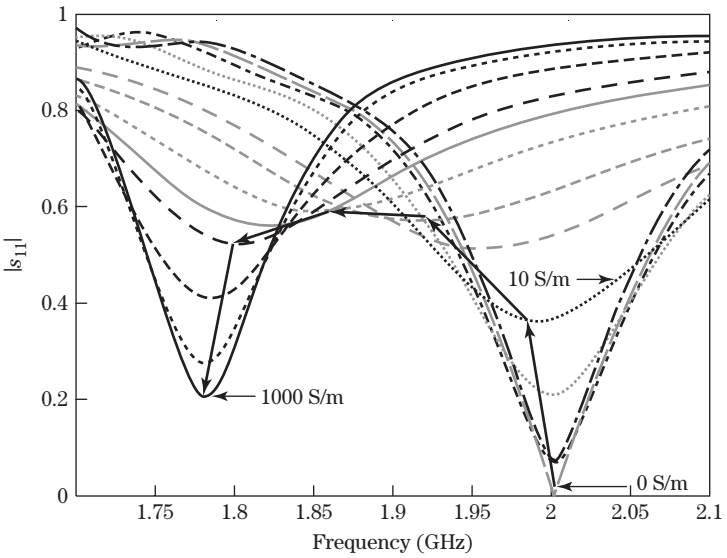


FIGURE 12-23 ■ Plots of the magnitude of s_{11} for silicon conductivities of 0, 1, 2, 5, 10, 20, 30, 50, 75, 100, 200, 500, and 1,000 S/m.

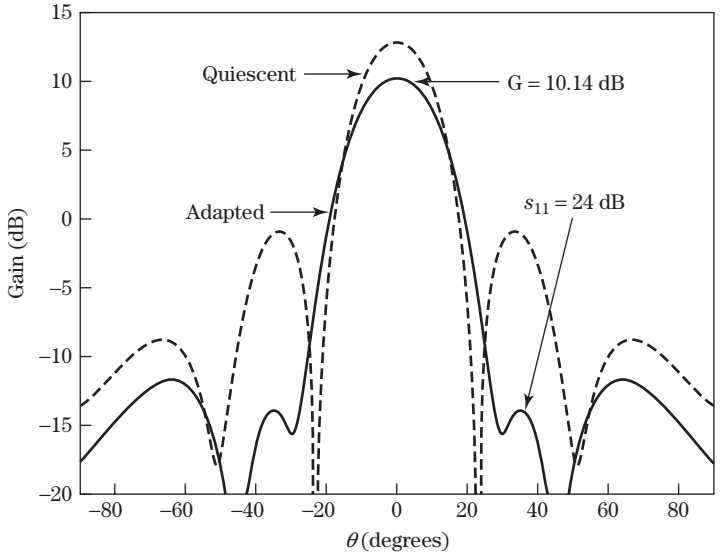


FIGURE 12-24 ■ The quiescent pattern is the dashed line and has all the conductivities set to 0. The adapted pattern is the solid line and has the silicon conductivities set to [16 8 0 8 16] S/m.

center conductor in coplanar waveguide mounted on a silicon substrate [27]. This attenuator is flip mounted over a hole in the coplanar waveguide feed of the broadband element (Figure 12-26). An infrared (IR) LED illuminates the meandering line in the center rectangular section to increase in conductivity of the silicon and to create a short between the center and outer conductors of the coplanar waveguide. Connecting this attenuator to an array element attenuates the signal received by the element. The element bandwidth was measured from 2.1 to 2.5 GHz ($S_{11} < -10$ dB). Figure 12-27 shows the array patterns the uniform array (all LEDs off) and for two Chebyshev amplitude tapers found by relating the LED current to the signal amplitude at the elements.

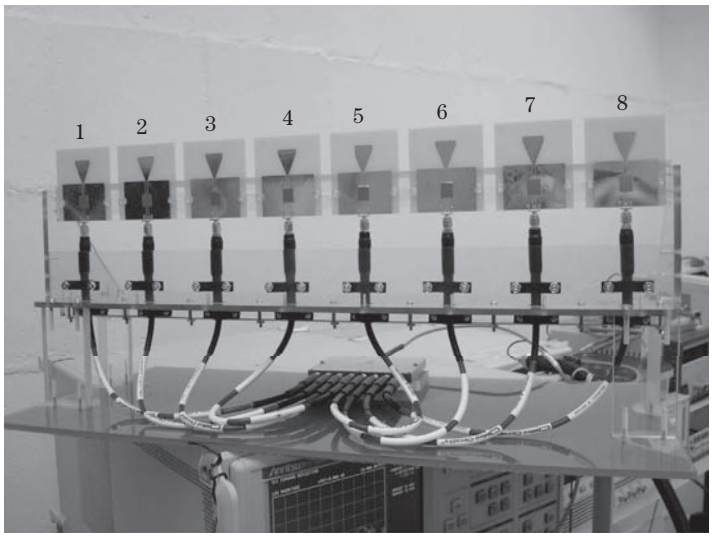


FIGURE 12-25 ■ Picture of experimental array with numbered elements.

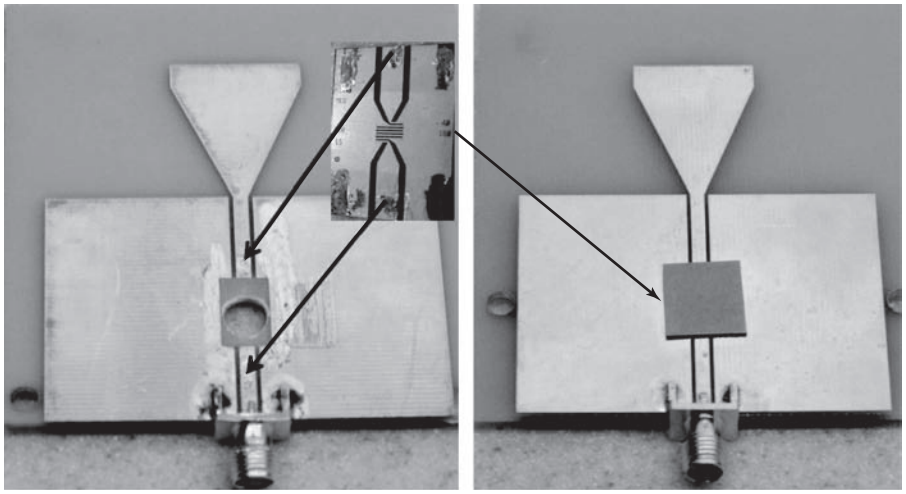


FIGURE 12-26 ■ The attenuator on the left is flipped and soldered to the antenna element on the right.

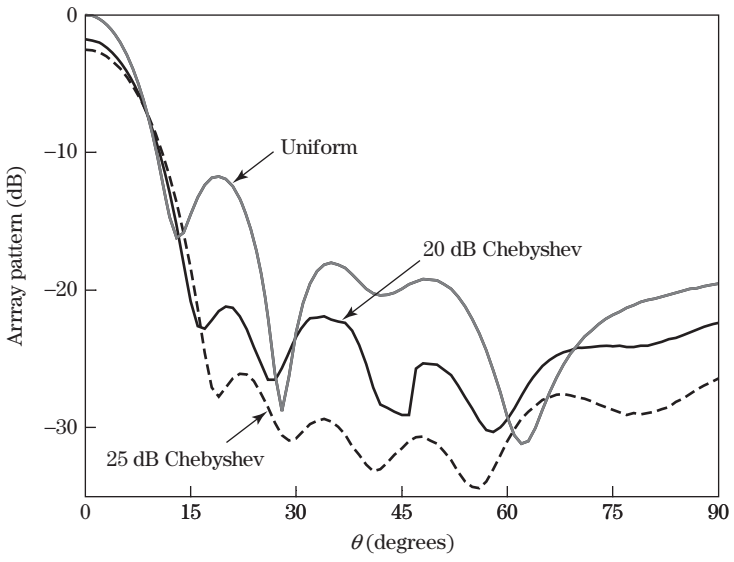


FIGURE 12-27 ■ Measured low sidelobe antenna patterns for the eight-element array when the diode currents are set for uniform and 20 and 25 dB Dolph–Chebyshev tapers.

12.5 | PERFORMANCE CHARACTERISTICS OF LARGE SONAR ARRAYS

Acoustic waves have much longer wavelengths and slower velocity of propagation than electromagnetic waves, so extracting information from the signals received by a sonar array is considerably different from extracting it from a radar. Large aperture sonar arrays contain hundreds of sensors. Such large arrays have some unexpected problems associated with adaptive algorithms and acoustic propagation. Since the dominant noise source in cluttered sonar environments is shipping, the signal environment is nonstationary with an associated time scale that has important implications for the ability to form array pattern nulls around moving ships [28].

Most adaptive algorithms either explicitly or implicitly form the sample covariance matrix

$$\hat{\mathbf{R}}_{xx} = \frac{1}{K}\sum_{m=1}^{K}\mathbf{x}_m\mathbf{x}_m^H \tag{12.29}$$

There are limits determining the number of time samples (snapshots), K, which are available in forming the estimate $\hat{\mathbf{R}}_{xx}$: (1) the time duration limit; and (2) the bandwidth limit, over which frequency averaging can be done. At broadside the main lobe of a sonar resolution cell has a cross-range extent given by

$$\Delta x \approx \frac{\lambda r}{L} \tag{12.30}$$

where r is the range to the source, L is the array aperture length, and λ is the wavelength of the transmitted sonar signal. For a source traveling with speed v, the associated bearing rate is

$$\dot{\phi} = \frac{v}{r} \tag{12.31}$$

Hence, the time spent within a single resolution cell is

$$\Delta \mathrm{T} = \frac{\Delta x}{v} = \frac{\left(\frac{\lambda r}{\mathrm{L}}\right)}{v} = \frac{\lambda}{L\dot{\phi}} \tag{12.32}$$

The transit time, $\mathrm{T}_{\mathrm{transit}}$, of an acoustic wave across the face of the array at endfire is given by

$$\mathrm{T}_{transit} = \frac{L}{v} \tag{12.33}$$

where v is the acoustic wave velocity in water (nominally 1,500 m/sec). The estimate of the phase in the cross-spectra will be smeared if one averages over more than one-eighth the transmit time bandwidth, so

$$\mathrm{B} < \frac{1}{8T_{transit}} = \frac{v}{8L} \tag{12.34}$$

The time–bandwidth product given by the product of (12.33) and (12.34) yields the approximate number of time samples (snapshots) available to form $\hat{\mathbf{R}}_{xx}$, or

$$\mathrm{K} < \Delta\mathrm{T} * \mathrm{B} = \frac{\lambda}{L\dot{\phi}} \times \frac{1}{8T_{transit}} = \frac{1}{8f\dot{\phi}} \left(\frac{v}{L}\right)^2 = \frac{f}{8\dot{\phi}} \left(\frac{\lambda}{L}\right)^2 \tag{12.35}$$

where $\lambda = \frac{v}{f}$.

Equation (12.35) expresses the inverse square law dependence on array lengths and thereby forces L to be quite small (e.g., a 200 Hz source moving at 20 knots and a 10 km distance transiting a 100 wavelength array limits the number of time samples to only 3). This time–bandwidth product limit dependence on propagation speed illustrates a big difference between sonar and radar.

The transit of a signal source across a resolution cell introduces eigenvalue spread into the signal source spectrum. To see how this occurs, introduce the parameter μ, representing the fraction of motion relative to a beamwidth:

$$\mu = \frac{L\Delta(\cos(\phi))}{\lambda} \tag{12.36}$$

where Δ denotes the beamwidth extent in radians, and ϕ is the angle off-broadside. When the source is within a beamwidth, there is a single large eigenvalue associated with $\hat{R}_{xx}$; however, as soon as the motion occupies approximately one resolution cell, the second eigenvalue becomes comparable. Splitting the source into two sources separated at one-half the distance traveled approximates this behavior. Solving this two-component eigenvalue problem leads to

$$\hat{\lambda}_2 = \frac{1}{4}\left[1 - \sin c^2\left(\frac{\pi\mu}{2}\right)\right] \approx \frac{(\pi\mu)^2}{48} \tag{12.37}$$

The resulting eigenvalue distribution for a 10λ array with $\mathrm{L} = 500$ is illustrated in Figure 12-28. A linear approximation for λ_2, denoted by "A", is about 1 dB below the actual value. A linear fit, denoted by "B," is also indicated.

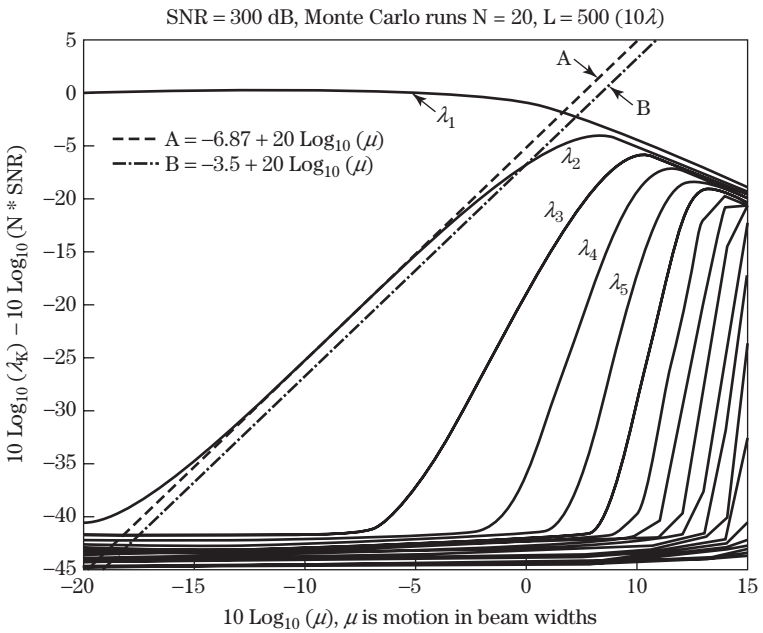


FIGURE 12-28 ■ Eigenvalues as a Function of Motion for a Moving Source Transiting a Sonar Resolution Cell. From Baggeroer & Cox, 1999 Asilomar Conference on Signals, Systems, and Computers, pp. 103–08

Most array processing algorithms assume that plane waves impinge on the array. For large apertures, however, wavefront curvature becomes significant, and plane wave models are no longer appropriate. The signal source range below which wavefront curvature must be taken into account is given by the Fresnel range,

$$\mathrm{R}_{\mathrm{Fresnel}} = \frac{L^2}{2\lambda} = \frac{\lambda}{2}\left(\frac{L}{\lambda}\right)^2 \tag{12.38}$$

Near-field effects become important even for modest frequencies (i.e., $f > 100$ Hz), and range-dependent processing must be introduced. The general solution to this problem, which is beyond the scope of this text, is known as matched-field processing, where the full-field propagation model occurs in the array processing [29].

One common way of dealing with the problem of a scarce number of samples in forming the sample covariance matrix is to introduce diagonal loading (discussed in Chapter 5). The introduction of diagonal loading improves the poor conditioning of the covariance matrix (although it hinders the detection of weak signals and introduces biases into the resulting direction-of-arrival estimates).

12.6 | ADAPTIVE PROCESSING FOR MONOPULSE TRACKING ANTENNAS

Monopulse array patterns are formed by synthesizing sum (Σ) and difference (Δ) beams. While it is straightforward to form adapted sum and difference beams, the resulting adapted monopulse pattern, Δ/Σ, has a highly distorted slope, rendering it ineffective for target angular location. To avoid this problem it is possible to use an approach that yields a controlled monopulse pattern [30]. We will consider this problem for a linear antenna array for which the angle off-boresight defines the tracking angle. The recommended procedure first forms the sum beam using the classic weight vector

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{s} \tag{12.39}$$

where $\mathbf{\Phi}$ is the interference signal covariance matrix, and $\mathbf{s}$ is the steering vector for the target of concern. The next step forms the difference beam $\Delta(\theta_0, f_0)$, where θ_0 is the target azimuth angle, and f_0 is the target Doppler frequency. The basic idea is to form the difference beam such that the received interference in minimized while satisfying three distinct constraints as follows:

1. $\Delta(\theta_0, f_0) = 0$ (the difference beam response is zero at the target location)
2. $\dfrac{\Delta(\theta_0 + \Delta\theta, f_0)}{\sum(\theta_0 + \Delta\theta, f_0)} = k_s \Delta\theta$ (maintain a constant slope where k_s is a slope constant, and $\Delta\theta$ is the angle excursion from θ_0 where the constraint holds)
3. $\dfrac{\Delta(\theta_0 + \Delta\theta, f_0)}{\sum(\theta_0 + \Delta\theta, f_0)} = -k_s \Delta\theta$ (negative angle excursion)

The steering vector can is the $NM \times 1$ vector defined as

$$\mathrm{s}^{\mathrm{T}} = [(\mathrm{s}_{11}\ \mathrm{s}_{12} \ldots \mathrm{s}_{1\mathrm{M}}) \cdots (\mathrm{s}_{\mathrm{N1}}\ \mathrm{s}_{\mathrm{N2}} \ldots \mathrm{s}_{\mathrm{NM}})] \tag{12.40}$$

reflecting an N-element array having a tapped delay line at each element and each tapped delay line has M time taps with a T sec time delay between each tap. The elements of s are

$$s_{nm} = e^{j2p\left[(n-1)\frac{2\pi}{\lambda}d(\sin\theta_0)-(m-1)f_0T\right]} \tag{12.41}$$

It is convenient to define the $NM \times 1$ vector $\mathbf{g}(\theta_0, f_0)$ as the previous vector $\mathbf{s}$. With these vectors, the constraints can be written in matrix notation as

$$\mathbf{H}^{\mathrm{T}}\mathbf{w}_{\Delta} = \boldsymbol{\rho} \tag{12.42}$$

where

$$\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \mathbf{g}^{\mathrm{T}}(\theta_0 + \Delta\theta, f_0) \\ \mathbf{g}^{\mathrm{T}}(\theta_0, f_0) \\ \mathbf{g}^{\mathrm{T}}(\theta_0 - \Delta\theta, f_0) \end{bmatrix} \tag{12.43}$$

$$\boldsymbol{\rho} = k_s \begin{bmatrix} \mathbf{w}^{\mathrm{T}}\mathbf{g}(\theta_0 + \Delta\theta, f_0) \\ 0 \\ -\mathbf{w}^{\mathrm{T}}\mathbf{g}(\theta_0 - \Delta\theta, f_0) \end{bmatrix} \Delta\theta \tag{12.44}$$

The weight vector $\mathbf{w}_{\Delta}$ that minimizes the difference beam interference power, $\mathbf{w}_{\Delta}^{H}\boldsymbol{\Phi}\mathbf{w}_{\Delta}$, subject to the constraint (12.42) is

$$\mathbf{w}_{\Delta} = \boldsymbol{\Phi}^{-1}\mathbf{H}^{*}(\mathbf{H}^{\mathrm{T}}\boldsymbol{\Phi}^{-1}\mathbf{H}^{*})^{-1}\boldsymbol{\rho} \tag{12.45}$$

To illustrate these results, consider a 13-element linear array with 14 taps in each tapped delay line. Assume that the clutter-to-noise ratio is 65 dB per element. The sum beam weight vector given by (12.1) produces a beam that has an interference plus noise power after adaptation that is close to the noise floor for all target speeds V such that $0.05 < \mathrm{V}/\mathrm{V}_b < 0.95$, where V_b is the radar blind speed. Likewise, the weight vector given by (12.47) also produces a difference beam with an adapted, interference plus noise power close to the noise floor for all target speeds. Figure 12-29 shows the adapted monopulse pattern, Δ/Σ, for two different target speeds where $\Delta\theta = 0.644\times$(3 dB Σ-beam angle).

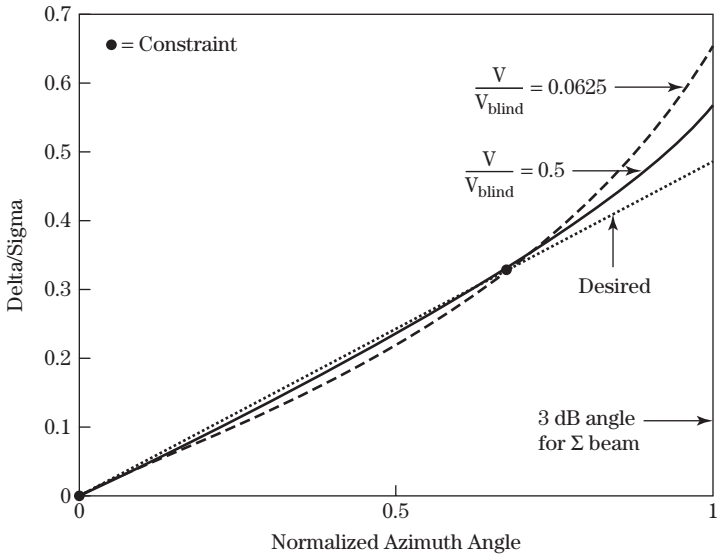


FIGURE 12-29 ■ Positive angle portion of adapted monopulse for three constraints. From Fante, IEEE Trans. Ant. & Prop. May 1999, pp. 773–74

12.7 PARTIALLY ADAPTIVE ARRAYS

Consider the block diagram of an interference cancelling adaptive array system given in Figure 12-30. For this system, a main array output is formed by merely combining the N element outputs. An adaptive processor combines only M of the element signals (where $M \leq N$) to form an adaptive output that is then subtracted from the main array output. In the event that $M = 1$, a simple sidelobe canceller system results. On the other hand if $M = N$, then the array is said to be fully adaptive. Cases that fall in between where $1 < M < N$ are referred to as a partially adaptive array (or as a multiple sidelobe canceller for the given configuration).

A fully adaptive array in which every element is individually adaptively controlled is obviously preferred since it affords the greatest control over the array pattern. For large arrays containing thousands of elements, individually controlling every element can prove a prohibitive implementation expense. Furthermore, signal processor implementation becomes much more difficult and costly as the dimensionality of the signal vector to be processed increases. Consequently, it is highly desirable to reduce the dimensionality of the signal processor while maintaining a high degree of control over the array response by adopting one of the following adaptive control philosophies:

1. Judiciously select only a fraction of the array elements to adaptively control, thereby resulting in elemental-level adaptivity [31,32].
2. Combine the N elements in the entire array into a collection of M subarrays by means of a subarray beamformer transformation $\mathbf{Q}$, and adaptively control each of the resulting subarray outputs [33,34].

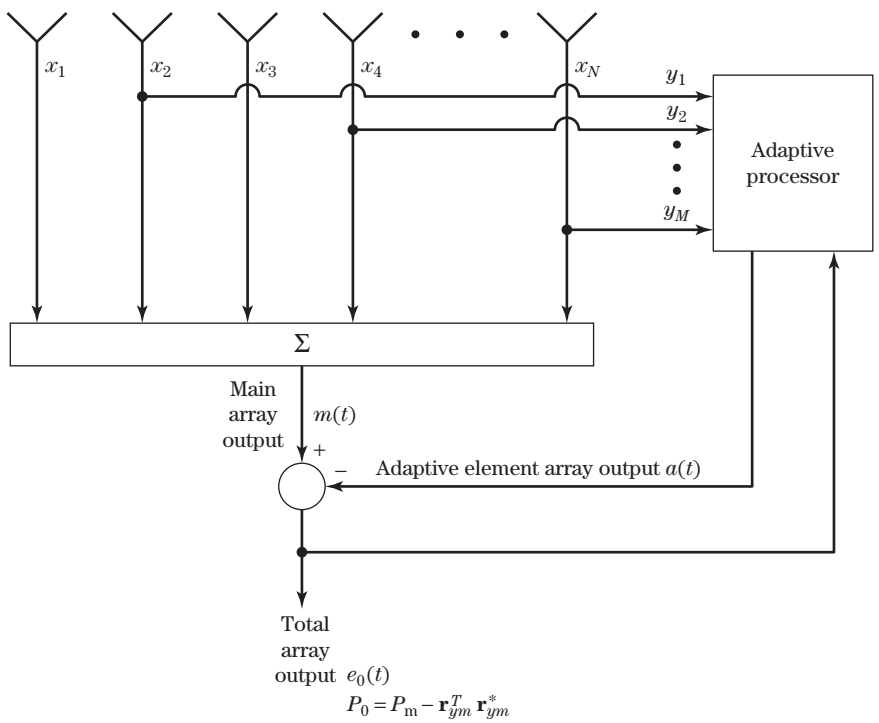


FIGURE 12-30 ■ Interference cancelling adaptive array block diagram.

Each of these approaches will now be discussed to determine the characteristics that typify these partially adaptive array concepts.

As shown in Chapter 2, null synthesis can be applied to analyzing partial adaptive nulling. A first step in picking the appropriate elements for a partially adaptive array is to look at null synthesis using a subset of the elements. The synthesized array factor when a subset of the elements has variable weights is given by

$$AF = \sum_{n=1}^{N} a_n e^{jkx_n \cos\phi} - \sum_{n=1}^{N_a} \Delta_{e(n)} a_n e^{j\left(kx_{e(n)} \cos\phi + \delta_{e(n)}\right)} \tag{12.46}$$

where

$0 \le a_n \le 1.0$

$0 \le \delta_n < 2\pi$

$N_a =$ number of adaptive elements

$e =$ vector containing indexes of the adaptive elements

Equation (12.46) can be placed in matrix form

$$\mathbf{A}'\boldsymbol{\Delta}' = \mathbf{b} \tag{12.47}$$

where

$$\mathbf{A}' = \begin{bmatrix} a_{e(1)} e^{jk(e(1)-1)d\cos\phi_1} & \cdots & a_{e(N_a)} e^{jk(e(N_a)-1)d\cos\phi_1} \\ \vdots & \ddots & \vdots \\ a_{e(1)} e^{jk(e(1)-1)d\cos\phi_M} & \cdots & a_{e(N_a)} e^{jk(e(N_a)-1)d\cos\phi_M} \end{bmatrix}$$

$$\boldsymbol{\Delta}'\mathbf{w} = \begin{bmatrix} \Delta_{e(1)} & \cdots & \Delta_{e(N_a)} \end{bmatrix}^T$$

$$\mathbf{b} = \begin{bmatrix} \sum_{n=1}^{N} w_n e^{jk(n-1)d\cos\phi_1} & \cdots & \sum_{n=1}^{N} w_n e^{jk(n-1)d\cos\phi_M} \end{bmatrix}^T$$

The N_a adaptive weights in an N element partial adaptive array are written as

$$w_n = \begin{cases} a_n(1 - \Delta_n)e^{j\delta_n} & \text{if element } n \text{ is adaptive} \\ a_n & \text{if element } n \text{ is not adaptive} \end{cases} \tag{12.48}$$

Four different subsets of four adaptive elements are considered:

(a) 3, 4, 5, 6

(b) 1, 2, 7, 8

(c) 1, 3, 5, 7

(d) 2, 4, 5, 8

All the configurations have a cancellation pattern peak at $\theta = -21^\circ$, but the rest of the cancellation patterns are very different. The choice of the adaptive elements determines the shape of the cancellation pattern thus the distortion to the adapted pattern. Consider an eight-element array with $\lambda/2$ spacing. If all the elements are adaptive, then the synthesized array factor and cancellation beam in Figure 12-31. Four contiguous adaptive elements have the cancellation beam shown in Figure 12-32a. This cancellation pattern is just a

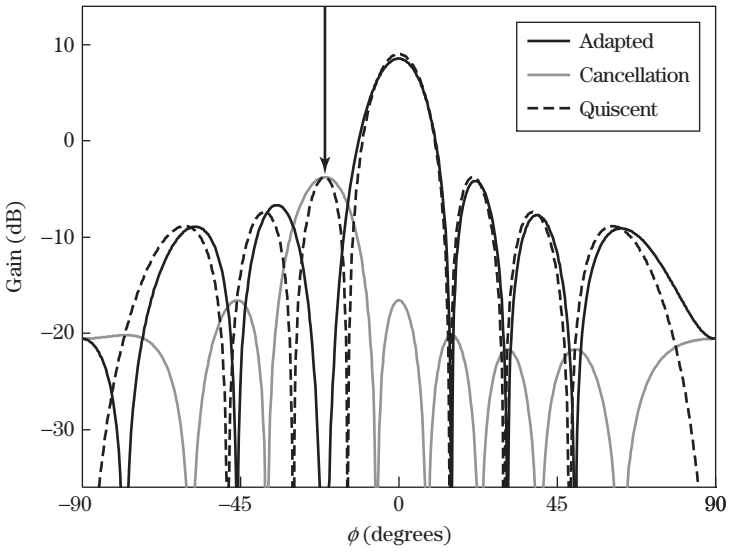
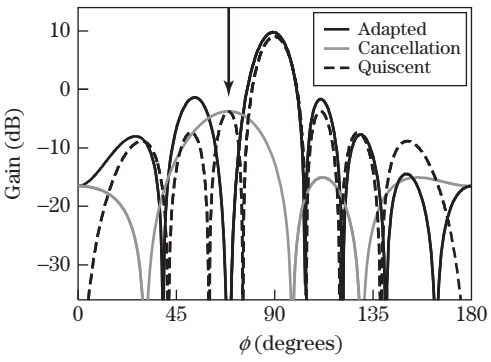
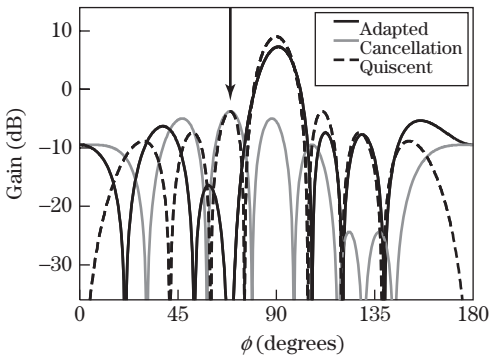


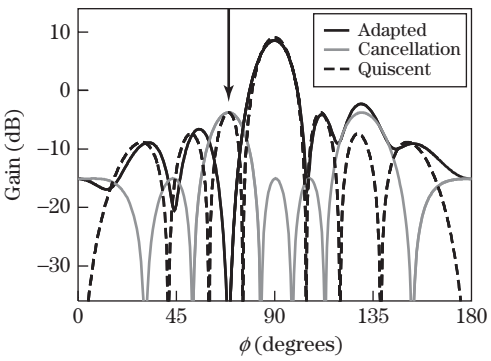
FIGURE 12-31 ■ Adapted and cancellation patterns superimposed on the quiescent pattern when a null is synthesized at $\theta = -21^\circ$ with all elements adaptive.



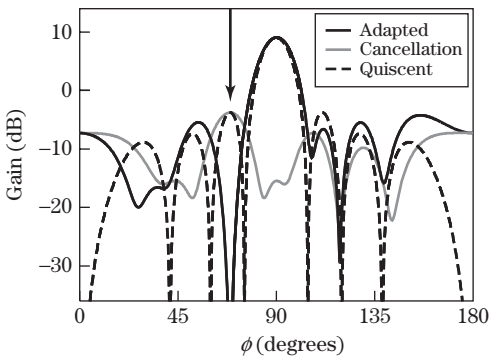
(a) Elements 3, 4, 5, and 6 adaptive.



(b) Elements 1, 2, 7, and 8 adaptive.



(c) Elements 1, 3, 5, and 7 adaptive.



(d) Elements 2, 4, 5, and 8 adaptive.

FIGURE 12-32 ■ Adapted and cancellation patterns superimposed on the quiescent pattern when a null is synthesized at $\theta = -21^\circ$ with four of the eight elements adaptive.

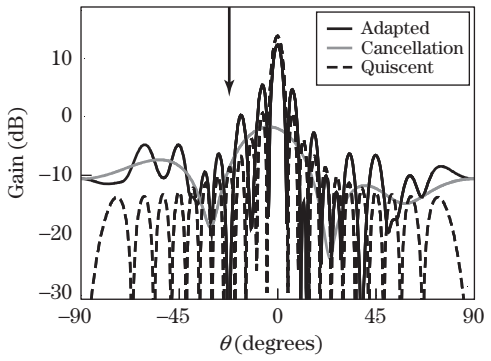
four-element uniform array factor with its main beam steered to $\theta = -21^\circ$. When two elements on each end of the array are adaptive, then the cancellation pattern has many lobes of approximately the same height (Figure 12-32b). Making every other element in the array adaptive induces grating lobes in the cancellation pattern as shown in Figure 12-32c. Random spacing of the adaptive elements produces the high sidelobe but narrow main beam cancellation pattern. Figure 12-32d shows the synthesized array factor and cancellation beam when the random elements are 2, 4, 5, and 8.

The cancellation pattern for any experimental or computed adapted array pattern is found by subtracting the quiescent electric field, $\mathbf{E}^{Quiescent}$, from the adapted electric field, $\mathbf{E}^{Adapted}$.

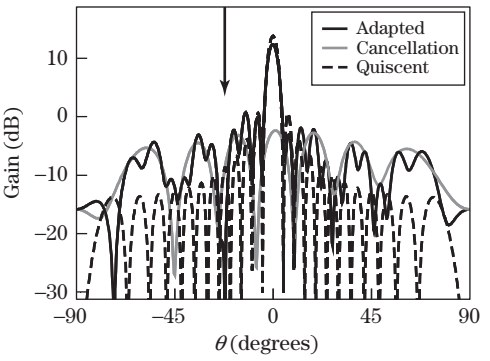
$$\mathbf{E}^{CancellationPattern} = \mathbf{E}^{Adapted} - \mathbf{E}^{Quiescent} \tag{12.49}$$

To plot the magnitude of the cancellation pattern correctly, $\mathbf{E}^{CancellationPattern}$ is normalized and adjusted relative to the peak gain of the adapted pattern.

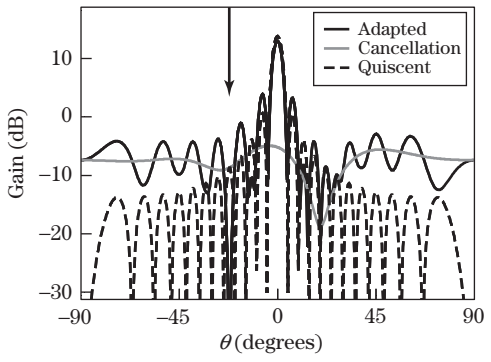
To test the usefulness of (12.49), a 24-element uniform array with $\lambda/2$ spacing has adaptive elements controlled by a genetic algorithm to minimize the total output power. A desired signal of 0 dB is incident at $\theta = 0^\circ$, whereas an interference signal is incident at $\theta = -21^\circ$. Figure 12-33a shows the case where the adaptive elements are four contiguous



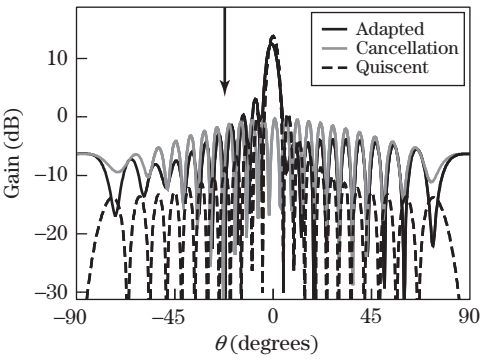
(a) Elements 11, 12, 13, and 14 adaptive complex weights.



(b) Elements 1, 2, 7, and 8 adaptive complex weights.



(c) Elements 11, 12, 13, and 14 adaptive phase weights.



(d) Elements 1, 2, 7, and 8 adaptive phase weights.

FIGURE 12-33 ■ Adapted and cancellation patterns for a 24-element array superimposed on the uniform quiescent pattern when a null is placed at $\theta = -22^\circ$.

elements in the center of the array. The nulling is accomplished with the side of the main beam of the cancellation beam rather than the peak. As a result, the loss in main beam gain and sidelobe distortion is significant but better than in the eight-element case. When the tow edge elements on either end of the array are adaptive, then the problems are similar to the previous case, but the distortion is different, because the cancellation pattern has many narrow lobes of nearly the same height rather than a uniform cancellation pattern (Figure 12-33b). Phase only adaptive nulling does not fare any better as shown in Figure 12-33c and Figure 12-33d.

Partial adaptive nulling experiments were done using the array in Figure 12-25. The output from the receiver goes to a computer with a genetic algorithm controller. The genetic algorithm varies the current fed to the LEDs to control signal attenuation at the adaptive elements. At least 15 dB of attenuation is available at each element. The largest possible decrease in gain occurs when the LEDs of the four adaptive elements are fed with 250 mA of current. The array becomes a four-element uniform array, so the main beam should decrease by 6 dB. Figure 12-34 shows a 5.2 dB decrease in the measured main beam of the far-field pattern. Thus, the adaptive array reduces the desired signal entering the main beam but cannot place a null in the main beam.

Figure 12-35a is the adapted pattern when one signal is incident on the array at -19° and elements 1, 2, 7, and 8 are adaptive. The adaptation lowered the main beam by 3.9 dB and the sidelobe level by 15.7 dB at -19°. Putting a -10 dBm desired signal at 0° and a 15 dBm signal incident at -19° produces the adapted pattern in Figure 12-35b. Its main beam goes down by -3.7 dB, and the sidelobe level at -19° goes down by 17.1 dB. These results show that adaptation with the desired signal present was approximately the same as when it was absent. Nulls can be placed in any sidelobe of the array pattern.

The adapted pattern when two 15 dBm signals are incident at -35° and -19° is shown in Figure 12-36. The main beam is reduced by 3.6 dB, while the sidelobe at -35° goes from -14 dB to -23 dB, and the sidelobe at -19° goes from -11.8 dB to -24.3 dB. Figure 12-37 shows the convergence for six independent random runs. Significant improvement occurs after five iterations.

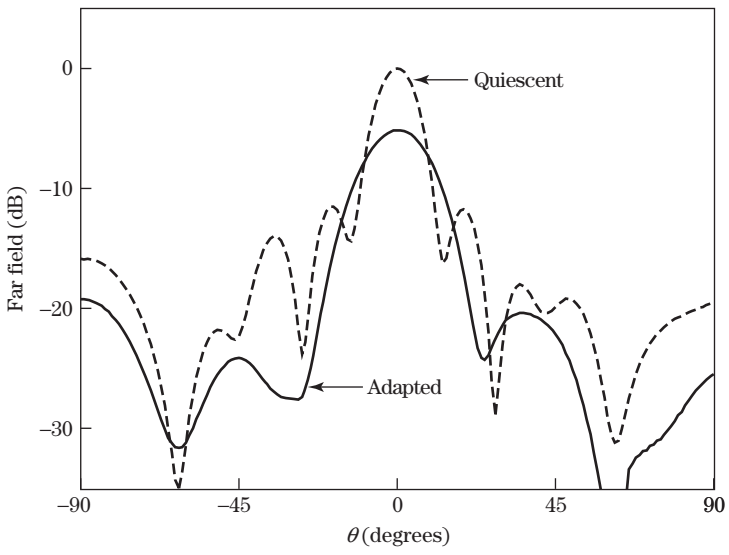
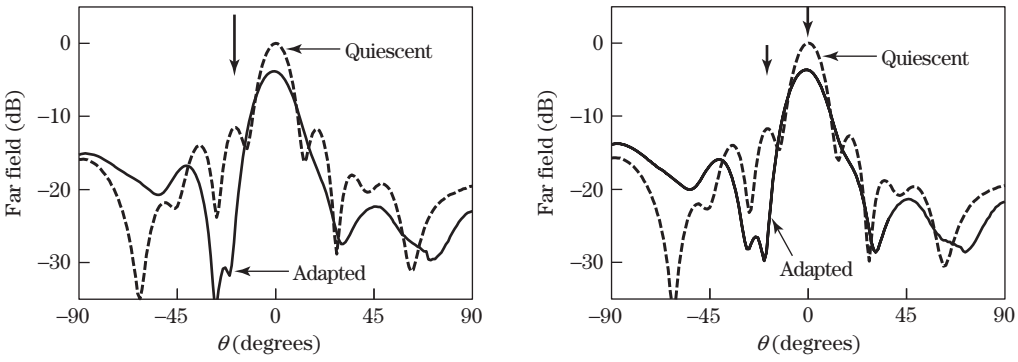


FIGURE 12-34 ■ Antenna pattern when the four adaptive elements are turned off.



(a) One signal at –19°. (b) Two signals at –19° and –0°.

FIGURE 12-35 ■ Adapted pattern (b) with and (a) without a signal present in the main beam.

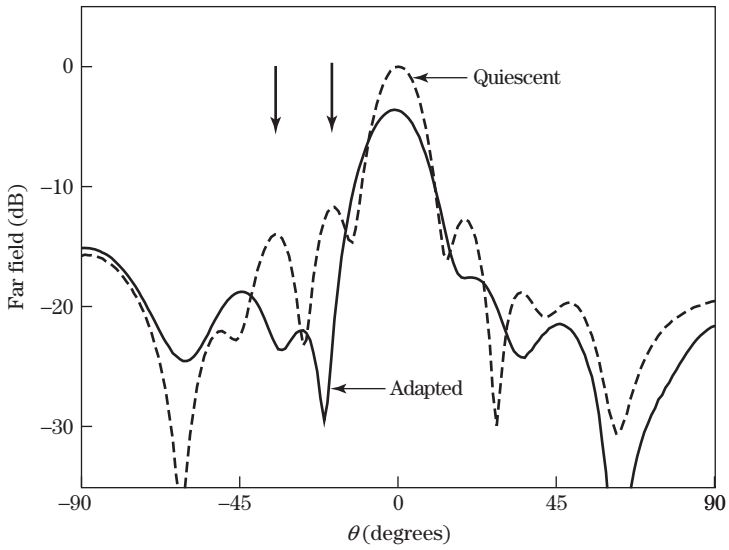


FIGURE 12-36 ■ Adapted pattern when signals are incident on the array at −35° and −19°.

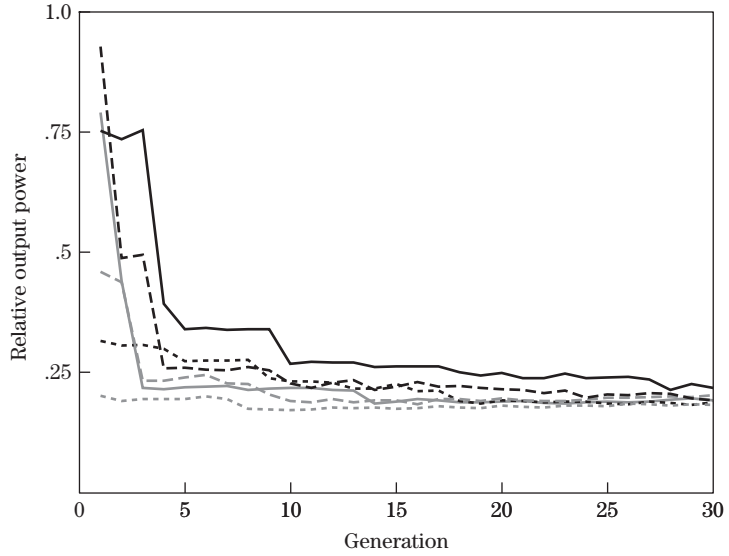


FIGURE 12-37 ■ Convergence of the adaptive algorithm when the signals are incident on the array at −35° and −19° for six independent random runs.

The adapted pattern for a signal is incident at 35° when elements 1 and 8 are adaptive is shown in Figure 12-38a. The main beam is reduced by 0.9 dB, whereas the sidelobe at 35° is reduced by 22 dB. A −10 dBm signal incident at 0° and a 15 dBm signal incident at −18° with elements 1 and 7 adaptive result in the adapted pattern in Figure 12-38b. The main beam reduction is 2.7 dB, and the sidelobe level at −18° is reduced by 13.1 dB. The main beam gain reduction is less when only two elements are adaptive compared with when four elements are adaptive.

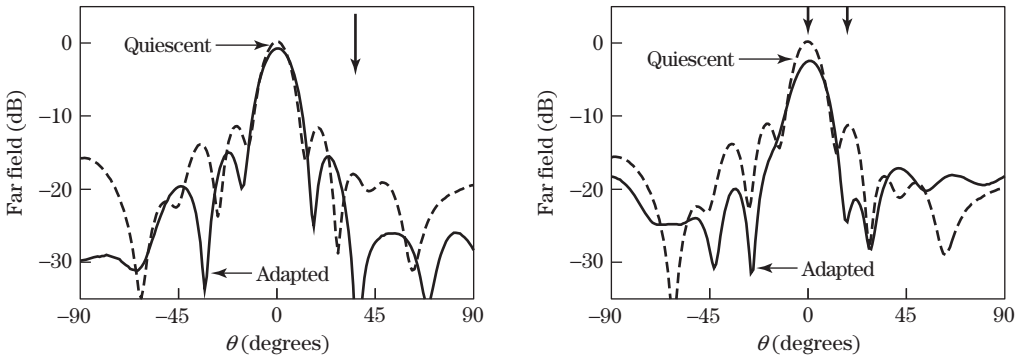
12.7.1 Adaptive Subarray Beamforming

The fundamental concept of adaptive subarray beamforming discussed by Chapman [33] is to reduce the required dimensionality of the signal processor by introduction of an $N \times M$ transformation matrix $\mathbf{Q}$ (called a subarray beamformer) as shown in Figure 12-39. The subarray signals resulting from the elements so combined in subarrays are then all adaptively controlled to produce the total array response. From Figure 12-39 it follows that

$$\mathbf{y} = \mathbf{Q}^T \mathbf{x} \tag{12.50}$$

Consequently, the covariance matrix of the subarray signal vector is given by

$$\mathbf{R}_{yy} = \mathbf{Q}^\dagger \mathbf{R}_{xx} \mathbf{Q} \tag{12.51}$$



(a) Elements 1 and 8 are adaptive.

(b) Elements 1 and 7 are adaptive.

FIGURE 12-38 ■ Adapted pattern when a signal is incident on the array at 35° and two elements are adaptive.

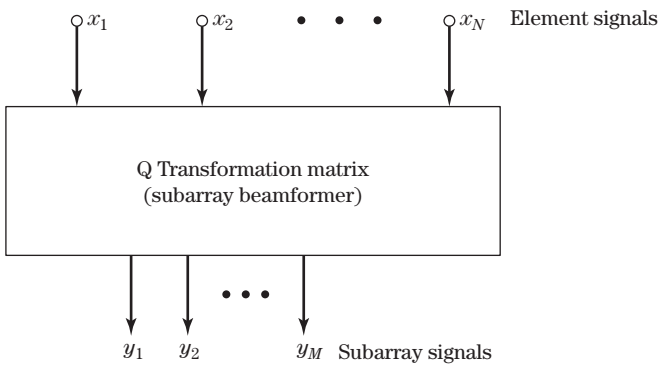


FIGURE 12-39 ■ Use of a subarray beamformer to reduce the signal vector dimensionality.

If the adaptive processor employs the Howells–Applebaum SNR performance measure algorithm, then the optimum weight vector solution is given by

$$\mathbf{w}_{y_{\text{opt}}} = \alpha \mathbf{R}_{yy}^{-1} \mathbf{v}_y^* \tag{12.52}$$

where the beam steering vector for the subarray $\mathbf{v}_y$ is related to the beam steering vector for the total array $\mathbf{v}_x$ by

$$\mathbf{v}_y = \mathbf{Q}^T \mathbf{v}_x \tag{12.53}$$

Consequently, the resulting array beam pattern can be computed by using the implied relationship

$$\mathbf{w}_x = \mathbf{Q}\mathbf{w}_{y_{\text{opt}}} \tag{12.54}$$

Note that $\mathbf{w}_x$ of (12.78) is not the same as $\mathbf{w}_{x_{\text{opt}}} = \alpha \mathbf{R}_{xx}^{-1} \mathbf{v}_x^*$ for the fully adaptive array but is rather a suboptimum solution that is constrained by the subarray configuration.

Now consider the effect of element-level amplitude and phase errors in a fully adaptive array and in a partially adaptive array. Amplitude and phase errors between any two channels is characterized by a mean offset signal level (over the bandwidth of interest) plus a variation about the mean as a function of frequency. The adjustment of a single complex weight in one channel removes any mean error that exists between a pair of channels, so a fully adaptive array is sensitive only to the error variance about the mean.

A subarray configuration, however, does not have as many degrees of freedom as the fully adaptive array and consequently may not succeed in removing all the mean errors that may exist among all the elements of the entire array. Thus, the performance of an array that is partitioned into subarrays is more susceptible to element-level errors than a fully adaptive array.

The effect of element random phase errors on a subarray structure are determined by appropriately modifying the $\mathbf{Q}$ subarray beamformer matrix. Define an "errored" $\mathbf{Q}$ matrix $\mathbf{Q}^{\varepsilon}$, which is given by

$$\mathbf{Q}^{\varepsilon} = \mathbf{E}^{\rho}\mathbf{Q} \tag{12.55}$$

where $\mathbf{E}^{\rho}$ is an $N \times N$ diagonal matrix with elements given by

$$e_{ii}^{\rho} = \exp[j2\pi(1 - \rho_e)z_i] \tag{12.56}$$

The parameter $\rho_e (0 \leq \rho_e \leq 1)$ represents the severity of the error (with no error corresponding to $\rho_e = 1$), whereas z_i is a uniformly distributed random variable $(-0.5 \leq z_i \leq 0.5)$.

This errored $\mathbf{Q}$ matrix model results in a random sidelobe structure (RSL) for the total array whose mean level (with respect to isotropic) is approximately given by [35]

$$|g_{\text{RSL}}(\theta)|^2 \cong \frac{\pi^2(1 - \rho_e)^2}{3} |g_e(\theta)|^2 \tag{12.57}$$

where $g_e(\theta)$ denotes the directional voltage gain of an array element.

Choosing subarrays by merely grouping physically contiguous elements is termed a *simple subarray* approach. Any simple subarray configuration has the subarray phase centers separated by several wavelengths produce grating lobes that cannot be altered by the adaptive processor [36]. A "beam-space" subarray, in which the full array aperture

is used for each subarray so the resulting structure can be regarded as a multibeam processor, avoids the grating lobe problem. Vural [37] likewise showed that a beam-based processor realizes superior performance in partially adaptive operations under diverse interference conditions. To introduce constraints into beam-space adaptive algorithms, the only difference compared with element-space data algorithms is in the mechanization of the constraint requirements [34].

Subarray groups for planar array designs are chosen by combining row (or column) subarrays. This choice is adaptive in only one principal plane of the array beam pattern and may therefore be inadequate. A more realistic alternative to the row–column subarray approach is a configuration called the row–column precision array (RCPA) [38], in which each element signal is split into two paths: a row path and a column path. All the element signals from a given row or column are summed, and all the row outputs and column outputs are then adaptively combined. The number of degrees of freedom in the resulting adaptive processor equals the number of rows plus the number of columns in the actual array.

When ideal operating conditions are assumed with perfect element channel matching, simulation studies have shown that the subarray configurations previously discussed yield array performance that is nearly the same as that of fully adaptive arrays [33]. When the array elements have independent random errors, however, the resulting random sidelobe structure severely deteriorates the quality of the array response. This performance deterioration results from the need for precision in the subarray beamforming transformation since this precision is severely affected by the random element errors.

12.7.2 Element-Level Adaptivity

Adaptive subarray beamforming is a highly attractive solution to the partially adaptive array problem, particularly if the application inherently requires multiple beamforming. However, the beamforming matrix transformation introduces an additional expense into the realization that can be avoided if multiple beams are not required by simply directly controlling only a fraction of the array elements in an element-level partial adaptivity scheme. With this approach, the question arises as to which elements of the original array should be adaptively controlled to achieve the best possible array performance. To gain some insight into the behavior of the system depicted in Figure 12-30, the approach taken by Morgan [32] is followed, and an explicit solution for a narrowband two-jammer problem is obtained.

Using the MMSE performance measure, the optimum weight vector for the adaptive processor is given by

$$\mathbf{W} = \mathbf{R}_{yy}^{*-1}\mathbf{r}_{ym} \tag{12.58}$$

where $\mathbf{R}_{yy} \triangleq E\{\mathbf{y}\mathbf{y}^{\dagger}\}$ is the $M \times M$ covariance matrix of the adaptive element signal vector, and $\mathbf{r}_{ym} \triangleq E\{m\mathbf{y}^{*}\}$ is the cross-correlation vector of the main array signal and the adaptive element signal vector. It is worth noting that the solution for $\mathbf{w}$ in (12.58) results instead of the more familiar expression $\mathbf{w} = \mathbf{R}_{yy}^{-1}\mathbf{r}_{ym}$, because Morgan defined $R_{yy} = E\{\mathbf{y}\mathbf{y}^{\dagger}\}$ instead of the more definition $R_{yy} = E\{\mathbf{y}^{*}\mathbf{y}^{T}\}$. The minimum total array output power that results when (12.58) is used is then

$$P_0 = P_m - \mathbf{r}_{ym}^{T}\mathbf{R}_{yy}^{-1}\mathbf{r}_{ym}^{*} \tag{12.59}$$

where P_m is the main array output power. The straightforward evaluation of (12.59) does not yield much insight into the relationship between adaptive performance and the array configuration and jamming environment. A concise mathematical expression of (12.59) is accordingly more desirable to elucidate the problem.

Consider two narrowband jammers of frequencies ω_1 and ω_2. The composite signal vector $\mathbf{y}$ is written as

$$\mathbf{y}(t) = \mathbf{n}(t) + J_1 \exp(j\omega_1 t)\mathbf{v}_1 + J_2 \exp(j\omega_2 t)\mathbf{v}_2 \tag{12.60}$$

where J_1 and J_2 represent the reference amplitude and phase, respectively, of the jammers, and $\mathbf{n}$ is the additive noise vector with independent components of equal power P_n. The spatial vectors $\mathbf{v}_1$, $\mathbf{v}_2$ have components given by

$$V_{k,i} = \exp\left\{ j\frac{2\pi}{\lambda}(\mathbf{r}_i \cdot \mathbf{u}_k) \right\}, \qquad i \in A, k = 1, 2 \tag{12.61}$$

where A denotes the subset of M adaptive elements, $\mathbf{r}_i$ is the ith element location vector, and $\mathbf{u}_k$ is a unit vector pointed in the direction of arrival of the kth jammer. For a linear array having elements aligned along the x-axis, (12.61) reduces to

$$V_{k,i} = \exp\left\{ j\frac{2\pi}{\lambda} x_i \sin\theta_k \right\} \tag{12.62}$$

where θ_k is the angle of arrival of the kth jammer measured from the array boresight.

The main array output signal can likewise be written as

$$m(t) = \sum_{i=1}^{N} n_i(t) + J_1 e^{j\omega_1 t} h_1 + J_2 e^{j\omega_2 t} h_2 \tag{12.63}$$

where

$$h_k = \sum_{i=1}^{N} v_{k,i}, \qquad k = 1, 2 \tag{12.64}$$

is the main array factor, which is computed by forming the sum of all spatial vector components for all the array elements $i = 1, \ldots, N$. It follows that the array output signal is written as

$$e_0(t) = m(t) - \mathbf{w}^T \mathbf{y}(t) \tag{12.65}$$

where $\mathbf{w}$ is the vector of adaptive weights.

The minimum total array output power given by (12.59) requires the covariance matrix $\mathbf{R}_{yy}$ and the cross-correlation vector $\mathbf{r}_{ym}$, both of which can be computed from (12.60) and (12.61), to yield

$$\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^\dagger\} = P_n\mathbf{I} + P_1\mathbf{v}_1\mathbf{v}_1^\dagger + P_2\mathbf{v}_2\mathbf{v}_2^\dagger \tag{12.66}$$

$$\mathbf{r}_{ym} = E\{\mathbf{y}^* m\} = P_n\mathbf{1} + P_1 h_1 \mathbf{v}_1^* + P_2 h_2 \mathbf{v}_2^* \tag{12.67}$$

where P_1 and P_2 denote the jammer power levels, and $\mathbf{1}$ is an $M \times 1$ vector of ones.

Since (12.58) requires the inverse of $\mathbf{R}_{yy}$, this inverse is explicitly obtained by the twofold application of the matrix inversion identity (D.10) of Appendix D to (12.66).

Substitution of (12.67) along with $\mathbf{R}_{yy}^{-1}$ into (12.58) then yields

$$\mathbf{w}_{\text{opt}} = \mathbf{1} + \gamma \left[F_1 - \rho^* F_2 + \frac{P_n}{M P_2} F_1 \right] \mathbf{v}_1^* + \gamma \left[F_2 - \rho F_1 + \frac{P_n}{M P_1} F_2 \right] \mathbf{v}_2^* \quad (12.68)$$

where

$$F_k = \sum_{i \notin A} v_{k,i} \quad (12.69)$$

denotes the complementary array factor that is computed by summing the spatial vector components for all the elements that are not adaptively controlled, the factor

$$\rho = \frac{1}{M} \sum_{i \subset A} v_{1,i}^* v_{2,i} \quad (12.70)$$

is the complex correlation coefficient of the adaptive element spatial vectors, and

$$\gamma = \left[M(1 - |\rho|^2) + \frac{P_n}{P_1} + \frac{P_n}{P_2} + \frac{P_n^2}{M P_1 P_2} \right]^{-1} \quad (12.71)$$

For a fully adaptive array where $N = M$, then (12.58) reduces to $\mathbf{w}_{\text{opt}} = \mathbf{1}$ in the absence of any constraints on the main lobe. If it is assumed that both jammers are much stronger than the noise so that $P_1, P_2 \gg P_n$, then (12.58) becomes

$$\mathbf{W}_{\text{opt}}|_{P_1, P_2 \to \infty} \to \mathbf{1} + \frac{F_1 - \rho^* F_2}{M(1 - |\rho|^2)} \mathbf{v}_1^* + \frac{F_2 - \rho F_1}{M(1 - |\rho|^2)} \mathbf{v}_2^* \quad (12.72)$$

The previous expression shows that the adaptive weights are ill conditioned whenever $|\rho| \approx 1$. This condition physically corresponds to the situation that occurs when the adaptive array pattern cannot resolve the two jammers. Such a failure to resolve occurs either because the two jammers are too close together or because they simultaneously appear on distinct grating lobes.

Recognizing that the output residue power is given by

$$P_0 = P_m - \mathbf{r}_{ym}^T \mathbf{w}_{\text{opt}}^* \quad (12.73)$$

we see that it follows that the normalized output residue power is expressed as

$$\frac{P_0}{P_n} = N - M + \gamma \left\{ |F_1|^2 + |F_2|^2 - 2\text{Re}(\rho F_1 F_2^*) + \frac{P_n}{M} \left(\frac{|F_1|^2}{P_2} + \frac{|F_2|^2}{P_1} \right) \right\} \quad (12.74)$$

When $N = M$ (a fully adaptive array) then, in the absence of main lobe constraints, $P_0 = 0$. Equation (12.64) immediately yields an upper bound for the maximum residue power as

$$P_0 \leq P_{0_{\max}} = \frac{P_1 P_2 (|F_1| + |F_2|)^2 + \frac{P_n}{M}(P_2 |F_1|^2 + P_1 |F_2|^2)}{P_1 + P_2 + \frac{P_n}{M}} \quad (12.75)$$

and $P_0 = P_{0_{\max}}$ when $\rho_{\max} = -\exp[\arg(F_1^* F_2)]$.

In the case of strong jammers for $|\rho| < 1$, then (12.64) reduces to

$$\left.\frac{P_0}{P_n}\right|_{P_1, P_2 \to \infty} \to N - M + \frac{|F_1|^2 + |F_2|^2 - 2\mathrm{Re}(\rho F_1 F_2^*)}{M(1 - |\rho|^2)}, \qquad |\rho| < 1 \tag{12.76}$$

This expression emphasizes the fact that the residue power takes on a large value whenever $|\rho| = 1$. Furthermore, (12.66) surprisingly does not depend on the jammer power levels P_1 and P_2 but instead depends only on the geometry of the array and the jammer angles. The maximum residue power upper bound (12.65) does depend on P_1 and P_2, however, since

$$\left.P_{0_{\max}}\right|_{P_1, P_2 \to \infty} \to (|F_1| + |F_2|)^2 \frac{P_1 P_2}{P_1 + P_2} \tag{12.77}$$

The foregoing expressions for output residue power demonstrate the central role the correlation coefficient plays between the adaptive element spatial vectors in characterizing the performance of a partially adaptive array. In simple cases, this correlation coefficient ρ can be related to the array geometry and jammer angles of arrival thereby demonstrating that it makes a significant difference which array elements are chosen for adaptive control in a partially adaptive array. In most cases, the nature of the relationship between ρ and the array geometry is so mathematically obscured that only computer solutions yield meaningful solutions. For the two-jammer case, it appears that the most favorable adaptive element locations for a linear array are edge-clustered positions at both ends of the array [32].

The previous results described for a partially adaptive array and a two-jammer environment did not consider the effect of errors in the adaptive weights, even though they are very important [31]. The variation in the performance improvement achieved with a partially adaptive array with errors in the nominal pattern weights is extremely sensitive to the choice of adaptive element location within the entire array. Consequently, for a practical design the best choice of adaptive element location depends principally on the error level experienced by the nominal array weights and only secondarily on the optimum theoretical performance that can be achieved. It has been found that spacing the elements in a partially adaptive array using elemental-level adaptivity so the adaptively controlled elements are clustered toward the center is desirable since then the adapted pattern interference cancellation tends to be independent of errors in the nominal adaptive weight values [31].

12.7.3 Eigenspace Beamformers

An eigendecomposition used by an eigenspace beamformer projects the input onto a reduced-rank subspace called the eigensubspace containing the signal and interference [39]. The projected input is then processed to form the beam. This technique is especially useful in designing a partially adaptive array whose response is nearly the same as the response of a fully adaptive array.

The eigendecomposition of an $N \times N$ signal covariance matrix may be described by

$$\mathbf{R}_{xx} = \sum_{i=1}^{N} \lambda_i \mathbf{\Phi}_i \mathbf{\Phi}_i^H = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H \tag{12.78}$$

where λ_i are the eigenvalues of $\mathbf{R}_{xx}$, $\mathbf{\Phi}_i$ are the corresponding eigenvectors, $\mathbf{E}$ is an $N \times N$ matrix of eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix of ordered eigenvalues. We may now

select D_r of these eigenvalues to form an $N \times D_r$ matrix, E_r, which is a "reduced" subspace since $D_r < N$. How are the eigenvectors selected to be used in E_r? There are two generally accepted approaches. One way postulates that there are D plan wave signals in the signal environment; then select the $D + 1$ largest eigenvalues. The signal plus interference subspace is then spanned by the eigenvector matrix $\mathbf{E}_{S+I}$, where

$$\mathbf{E}_{S+I} = [\mathbf{\Phi}_1 \mathbf{\Phi}_2 \ldots \mathbf{\Phi}_{D+1}] \tag{12.79}$$

Using this approach, the number "D" must be estimated.

An alternate approach using the same eigendecomposition (12.78) does not assume any structure for the interference and merely selects the D_{r} largest eigenvalues. Various tests can be used for determining if an eigenvalue is "large" enough to be included in D_{r} as follows:

(i) Consider the % of total power contained in λ_i

$$\frac{\lambda_i}{tr(\mathbf{R}_{xx})} > n_1$$

(ii) Consider the % of total power contained in D_r eigenvalues

$$\frac{\sum_{i=1}^{D_r} \lambda_i}{tr(\mathbf{R}_{xx})} > n_2$$

(iii) Consider the ratio of adjacent eigenvalues

$$\frac{\lambda_{i+1}}{\lambda_i} > n_3$$

Having selected D_r, define $\mathbf{E}_r = [\mathbf{\Phi}_1 \mathbf{\Phi}_2 \ldots \mathbf{\Phi}_{D_r}$ and $\mathbf{\Lambda} = \mathrm{diag}[\lambda_1 \ \lambda_2 \cdots \lambda_{D_r}]$. Then

$$\mathbf{R}_{xx} = \mathbf{E}_{S+I} \mathbf{\Lambda}_{S+I} \mathbf{E}_{S+I}^H + \mathbf{E}_N \mathbf{\Lambda}_N \mathbf{E}_N^H \tag{12.80}$$

where $\mathbf{E}_N$ is orthogonal to $\mathbf{E}_{S+I}$ ($\mathbf{E}_N^H \mathbf{E}_{S+I} = 0$) and, hence,

$$\mathbf{R}_{xx}^{-1} = \mathbf{E}_{S+I} \mathbf{\Lambda}_{S+I}^{-1} \mathbf{E}_{S+I}^H + \mathbf{E}_N \mathbf{\Lambda}_N^{-1} \mathbf{E}_N^H \tag{12.81}$$

Introducing the desired signal steering vector, $\mathbf{v}_d$, which satisfies $\mathbf{w}^H \mathbf{v}_d = 1$, the minimum power distortionless response (MPDR) beamformer is then given by

$$\mathbf{w}_{\mathrm{MPDR}} = \frac{\mathbf{R}_{xx}^{-1} \mathbf{v}_d}{\mathbf{v}_d^H \mathbf{R}_{xx}^{-1} \mathbf{v}_d} = \frac{\mathbf{E}_{S+I} \mathbf{\Lambda}_{S+I}^{-1} \mathbf{E}_{S+I}^H \mathbf{v}_d}{\mathbf{v}_d^H \mathbf{E}_{S+I} \mathbf{\Lambda}_{S+I}^{-1} \mathbf{E}_{S+I}^H \mathbf{v}_d} \tag{12.82}$$

since $\mathbf{v}_d$ is in $\mathbf{E}_{S+I}$ and hence is orthogonal to $\mathbf{E}_N$. A corresponding result for the model of (12.78) yields

$$\mathbf{W}_{\mathrm{MPDR}} = \frac{\mathbf{E}_r \mathbf{\Lambda}_r^{-1} \mathbf{E}_r^H \mathbf{v}_d}{\mathbf{v}_d^H \mathbf{E}_r \mathbf{\Lambda}_r^{-1} \mathbf{E}_r^H \mathbf{v}_d} \tag{12.83}$$

Several variations on the theme of eigendecomposition have been suggested [40–42], but the previously given development expresses the core ideas about the various schemes that are developed.

An interesting application of eigendecomposition to the interference nulling problems facing radio telescopes is discussed in [43]. In radio astronomy applications, detection of the desired signal cannot take advantage of traditional demodulation–detection algorithms,

and the signal of interest is so minute that it is many orders of magnitude weaker than the noise power spectral density, leading to minutes or even hours of integration to achieve a positive detection. Radio frequency interference (RFI) is a growing problem for radio astronomy; hence, there is strong motivation to exploit adaptive nulling techniques. Common sources of RFI in radio astronomy include nongeosynchronous satellites and land-mobile radio. To deal effectively with such dynamic signals, it is necessary to use a weight update period that is on the order of 10 msec. On the other hand, a self-calibration process updated about once per minute is highly important for radio astronomy, since it removes environmental and instrumental errors. Self-calibration methods require nearly stationary adapted array patterns between self-calibration updates; hence, most standard-weight update algorithms, which are subject to "weight jitter," are inappropriate for radio astronomy. To circumvent the weight jitter problem, the eigendecomposition of (12.80) is introduced, and the further decomposition of (12.80) is used. Since the desired signal in radio astronomy is miniscule, the signal term in (12.80) can safely be ignored, leaving

$$\mathbf{R}_{xx} = \mathbf{E}_I \mathbf{\Lambda}_I \mathbf{E}_I^H + \mathbf{E}_N \mathbf{\Lambda}_N \mathbf{E}_N^H \tag{12.84}$$

Equation (12.84) is another way of saying that only interference and noise are in the observations, since the signals are not visible within the timeframe of a weight update. The term $\mathbf{E}_I \mathbf{\Lambda}_I \mathbf{E}_I^H$ completely describes the interference, and the column span of $\mathbf{E}_I$ is referred to as the *interference subspace.* Likewise, column span of $\mathbf{E}_N$ is the *noise subspace.* Rather than form an estimate of $\mathbf{R}_{xx}$ as is done with most pattern nulling algorithms, an attractive approach is to use *subspace tracking,* in which an estimate of the interference subspace is formed directly using rank-ordered estimates of the eigenvectors and associated eigenvalues. This approach has been found to be able to form the desired interference nulls without experiencing any associated weight (or pattern) jitter.

The estimation of $\mathbf{E}_I$ is accomplished by recourse to the *projection approximation subspace tracking with deflation* (PASTd) method [44] in combination with the Gram–Schmidt orthonormalization procedure to ensure that the resulting eigenvectors are orthonormal. It is beyond the scope of this chapter to present the details of this approach; suffice it to say that accurate interference subspace identification has been achieved for an interference-to-noise ratio of unity.

12.7.4 The *J* Subbeam Approach

A large array with N elements suffers certain disadvantages when operated as a fully adaptive array—such as an excessively large number of computations and poor sidelobe performance in directions away from jammer locations. Whereas a fully adaptive array may significantly degrade the array sidelobes as well as the main lobe peak gain, a partially adaptive array consisting of $(J+1)$ beams, where J denotes the number of jammers present [45]. By forming an equivalent small array having $(J+1)$ ports, the output of each port is treated as an "element" output to which any standard adjustment algorithm can be applied. Whereas N may be very large (10,000 or more for large arrays), $(J+1)$ for the equivalent array are very small, resulting in significant performance improvement.

The technique of forming an equivalent small array uses a two-step process. First, the number of interfering jammers and their locations are estimated using such techniques as a spatial discrete Fourier transform of the array outputs, by maximum entropy spectral estimation techniques [44,45], or just by a search in angle with an auxiliary beam. Once the first step is completed, auxiliary beams are formed pointing at each of the jammers, and one auxiliary beam forms pointing at the target, and an adaptive algorithm applied to

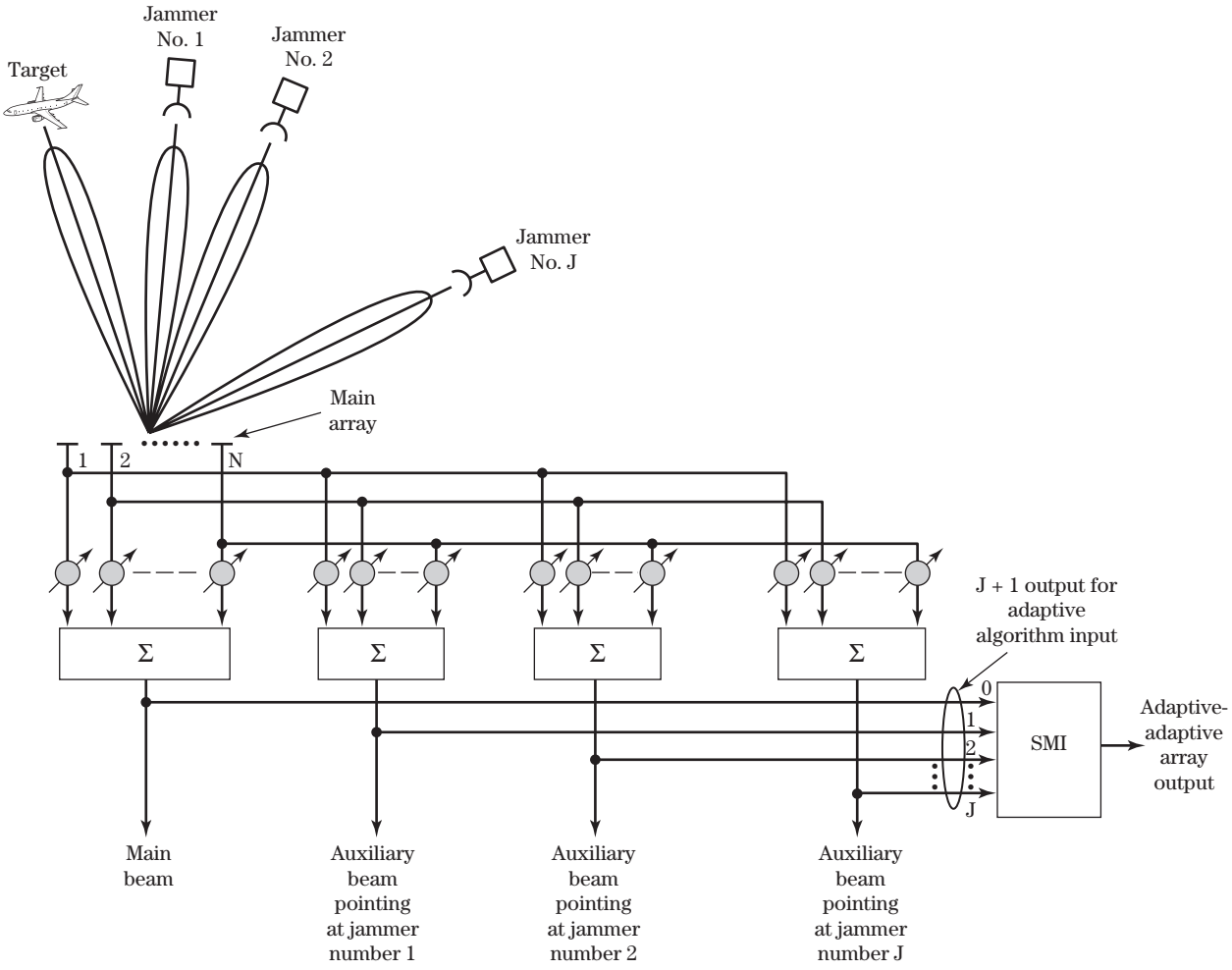


FIGURE 12-40 ■ Adaptive–adaptive array configuration. *N*-element array is reduced to $J + 1$ element array where J is equal to the number of jammers. From Brookner and Howell, Proc. IEEE, April 1986, pp. 602–604

the signals appearing at the $(J + 1)$ ports. The implementation of this approach is shown in Figure 12-40. The number of elements in the equivalent small array is only $(J + 1)$, so the equivalent signal covariance matrix is only $(J + 1) \times (J + 1)$.

The reason the adaptive–adaptive approach does not degrade the antenna sidelobes is that the equivalent small array subtracts one auxiliary beam pointing at the jammer from the main signal channel beam. The gain of the auxiliary beam in the direction of the jammer equals the gain of the main channel beam sidelobe in the direction of the jammer. As a result, the subtraction produces a null at the jammer location in the main channel sidelobe. Further variations of this basic scheme are discussed in the previously noted reference.

12.7.5 The Subarray Design Approach

One promising avenue to the design of a partially adaptive array is the subarray design approach [46]. The basic idea is to divide the original array into a series of tapered subarrays

resulting in a series of fixed beams using fixed weight vectors. Each of the resulting fixed beams may then be treated as a single element by weighting it with a single adaptive weight. Whereas the original array contains N elements, following the division into K fixed beams, there will be only K adaptive weights where $K < N$. It is not surprising that relatively good interference cancellation performance was observed to occur with subarrays comprised of elements clustered about the edge of the original array.

12.8 SUMMARY AND CONCLUSIONS

Beam switching is a very simple and cheap adaptive process that uses multiple beam arrays. In contrast, other techniques requiring extensive hardware and software complexity, such as DPCA, STAP, and MIMO, have received significant attention over the past 20 years. These techniques promise to overcome significant problems with clutter and multipath as well as rejecting interference. Reconfigurable antennas adapt by physically changing the antenna. The use of eigendecomposition in combination with modern spectral analysis techniques is yielding useful results in both sonar and radiotelescope applications, both of which experience unique technical problems not found in most radiofrequency applications. Compensation for mutual coupling in small arrays and the use of adaptive array concepts in monopulse tracking antennas are two areas that also offer promising results. The notions of subspace fitting and eigenspace beamformers were introduced using the concept of eigendecomposition as the point of departure. Partially adaptive array concepts are highly important because they offer the possibility of realizing near optimal array performance with only a fraction of the control elements (and hence the cost) required for a fully adaptive array.

12.9 PROBLEMS

1. **Multiple Beams.** Write a program in MATLAB to generate Figure 12-1.
2. **MIMO.** A MIMO system has a three-element array of isotropic point sources spaced d apart on transmit and receive. The system operates at 2.4 GHz, and the arrays are 100 m apart and face each other. Show how the condition number of H changes as the element spacing increases.
3. **Partial Adaptive Nulling.** Plot the cancellation patterns associated with placing a null at $\theta = 16.25^\circ$ in the array factor of a 32-element uniform array with $\lambda/2$ element spacing. Four different configurations of eight adaptive element are considered:
   a. 1,2,3,4,29,30,31,32
   b. 13,14,15,16,17,18,19,20
   c. 1,5,9,13,17,21,25,29
   d. 2,8,13,16,18,23,24,30
4. **Element Level Adaptivity** Recognizing that

$$(\mathbf{A} + \mathbf{b}\mathbf{b}^\dagger)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{b}^\dagger\mathbf{b}\mathbf{A}^{-1}}{(1 + \mathbf{b}^\dagger\mathbf{A}^{-1}\mathbf{b})}$$

   let $\mathbf{A} = P_n\mathbf{I} + P_1\mathbf{v}_1\mathbf{v}_1^\dagger$ in (12.66), and apply the previously given matrix identity once. Finally, apply the previous matrix identity to the resulting expression and show that (12.68) results from these operations on substitution of (12.66) into (12.58).

12.10 REFERENCES

[1] W. Rotman and R. Turner, "Wide-Angle Microwave Lens for Line Source Applications," *IEEE Transactions on Antennas and Propagation,* Vol. 11, No. 6, 1963, pp. 623–632.

[2] "Rotman Lens Designer," Remcom, Inc., 2009.

[3] J. Butler and R. Lowe, "Beam-Forming Matrix Simplifies Design of Electronically Scanned Antennas," *Electronic Design,* Vol. 9, April 1961, pp. 170–173.

[4] J. P. Shelton, "Fast Fourier Transforms and Butler Matrices," *Proceedings of the IEEE,* Vol. 56, No. 3, 1968, pp. 350–350.

[5] G. W. Stimson, *Introduction to Airborne Radar*, Raleigh, NC, SciTech Pub., 1998.

[6] C. E. Muehe and M. Labitt, "Displaced-Phase-Center Antenna Technique," Lincoln Lab Journal, Vol. 12, No. 2, 2000, pp. 281–296.

[7] J. K. Day and F. M. Staudaher, "Airborne MTI," *Radar Handbook*, M. I. Skolnik, ed., New York, McGraw Hill, 2008, pp. 3.1–3.34.

[8] R. Klemm and Institution of Engineering and Technology, *Principles of Space-Time Adaptive Processing*, 3rd ed., London, Institution of Electrical Engineers, 2006.

[9] Y. Dong, "A New Adaptive Displaced Phase Centre Antenna Processor," *2008 International Conference on Radar*, September 2–5, 2008, pp. 343–348.

[10] R. S. Blum, W. L. Melvin, and M. C. Wicks, "An Analysis of Adaptive DPCA," *Proceedings of the 1996 IEEE National Radar Conference*, 1996, pp. 303–308.

[11] H. Shnitkin, "Joint STARS Phased Array Radar Antenna," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 9, No. 10, October 1994, pp. 34–40.

[12] L. E. Brennan, and L. S. Reed, "Theory of Adaptive Radar," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-9, No. 2, 1973, pp. 237–252.

[13] L. Brennan, J. Mallett, and I. Reed, "Adaptive Arrays in Airborne MTI Radar," *IEEE Transactions on Antennas and Propagation,* Vol. 24, No. 5, 1976, pp. 607–615, 1976.

[14] K. Gerlach and M. J. Steiner, "Fast Converging Adaptive Detection of Doppler-Shifted, Range-Distributed Targets," *IEEE Transactions on Signal Processing,* Vol. 48, No. 9, 2000, pp. 2686–2690.

[15] J.O. McMahon, "Space-Time Adaptive Processing on the Mesh Synchronous Processor," *Lincoln Lab Journal,* Vol. 9, No. 2, 1996, pp. 131–152.

[16] W. Dziunikowski, "Multiple-Input Multiple-Output (MIMO) Antenna Systems," *Advances in Direction of Arrival Estimation*, S. Chandran (Ed.), Boston, MA, Artechhouse, 2006, pp. 3–19.

[17] R. Klemm, "STAP Architectures and Limiting Effects," in *ROT Lecture Series 228: Military Application of Space-Time Adaptive Processing,* April 2003, pp. 5-1–5-27.

[18] V. F. Mecca and J. L. Krolik, "MIMO STAP Clutter Mitigation Performance Demonstration Using Acoustic Arrays," in *42nd Asilomar Conference on Signals, Systems and Computers,* 2008, pp. 634–638.

[19] M. Rangaswamy, "An Overview of Space-Time Adaptive Processing for Radar," *Proceedings of the International Radar Conference,* 2003, pp. 45–50.

[20] J. B. Andersen, "Array Gain and Capacity for Known Random Channels with Multiple Element Arrays at Both Ends," *IEEE Journal on Selected Areas in Communications,* Vol. 18, No. 11, November 2000, pp. 2172–2178.

[21] Wireless InSite, 2.5, Remcom, Inc. (www.remcom.com), 2009.

[22] M. A. Jensen and J. W. Wallace, "A Review of Antennas and Propagation for MIMO Wireless Communications," *IEEE Transactions on Antennas and Propagation,* Vol. 52, No. 11, 2004, pp. 2810–2824.

[23] J. H. Winters, "On the Capacity of Radio Communication Systems with Diversity in a Rayleigh Fading Environment," *IEEE J. Select. Areas Commun.*, Vol. 5, June 1987, pp. 871–878.

[24] S. Nikolaou, N. D. Kingsley, G. E. Ponchak, J. Papapolymerou, and M. M. Tentzeris, "UWB Elliptical Monopoles With a Reconfigurable Band Notch Using MEMS Switches Actuated Without Bias Lines," *IEEE Transactions on Antennas and Propagation*, Vol. 57, No. 8, August 2009, pp. 2242–2251.

[25] R. L. Haupt, "Reconfigurable Patch with Switchable Conductive Edges," *Microwave and Optical Technology Letters,* Vol. 51, No. 7, pp. 1757–1760, July 2009.

[26] R. L. Haupt, "An Adaptive Rectangular Microstrip Patch Antenna Array Element Using Photonic Controls," IEEE Aerospace Conference, Big Sky, MT, March 2008.

[27] R. L. Haupt, J. R. Flemish, and D. W. Aten, "Broadband Linear Array with Photoconductive Weights," *IEEE AWPL*, Vol. 8, 2009, pp. 1288–1290.

[28] A. B. Baggeroer and H. Cox, "Passive Sonar Limits upon Nulling Multiple Moving Ships with Large Aperture Arrays," Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, & Computers, IEEE (Cat. No. CH37020), Vol. 1, Piscataway, NJ, 1999, pp. 103–08.

[29] A. Baggeroer, W. Kuperman, and H. Schmidt, "Matched Field Processing: Source Localization in Correlated Noise as an Optimum Parameter Estimation Problem," *J. Acoust. Soc. of Amer.,* Vol. 83, February 1988, pp. 571–587.

[30] R. L. Fante, "Synthesis of Adaptive Monopulse Patterns," IEEE Trans. Ant. & Prop., Vol AP-47. No. 5, May 1999, pp. 773–774.

[31] R.L. Haupt, "Adaptive nulling with weight constraints," *Progress In Electromagnetics Research B*, Vol. 26, 2010, pp. 23–38.

[32] D. R. Morgan, "Partially Adaptive Array Techniques," *IEEE Trans. Antennas & Prop.,* Vol. AP-26, No. 6, November 1978, pp. 823–833.

[33] D. J. Chapman, "Partial Adaptivity for the Large Array," *IEEE Trans. Antennas & Prop.,* Vol. Ap-24, No. 5, September 1976, pp. 685–696.

[34] A. M. Vural, "An Overview of Adaptive Array Processing for Sonar Applications," EASCON 1975 Record, *IEEE Electronics and Aerospace Systems Convention,* September 29–October 1, Washington, DC, pp. 34.A–34.M.

[35] M. Viberg and B. Ottersten, "Sensor Array Processing Based on Subspace Fitting," *IEEE Trans. Signal Processing,* Vol. 39, No. 5, May 1991, pp. 1110–1121.

[36] J. R. Flemish, H. W. Kwan, R. L. Haupt, and M. Lanagan, "A New Silicon-Based Photoconductive Microwave Switch," *Microwave and Optical Technology Letters,* Vol. 51, No. 1, January 2009.

[37] A. M. Vural, "A Comparative Performance Study of adaptive Array Processors," presented at the 1977 IEEE International Conference on Acoustics, Speech, and Signal Processing, May 9–11, Hartford, CT., paper 20.6.

[38] P. W. Howells, "High Quality Array Beamforming with a Combination of Precision and Adaptivity," prepared by Syracuse University Research Corp., SURC TN74-150, June 1974 (Unclassified).

[39] B. D. Van Veen, "Eigenstructure Based Partially Adaptive Array Design," *IEEE Trans. Ant. & Prop.,* Vol. AP-36, No. 3, March 1988, pp. 357–362.

[40] C. Lee and J. Lee, "Eigenspace-Based Adaptive Array Beamforming with Robust Capabilities," *IEEE Trans. Ant. & Prop.,* Vol. AP-45, No. 12, December 1997, pp. 1711–1716.

[41] N. L. Owsley, "Sonar Array Processing," *Array Signal Processing,* S. Haykin, ed., Englewood Cliffs, NJ, Prentice-Hall, 1985.

[42] I. P. Kirstens and D. W. Tufts, "On the Probability Density of Signal-to-Noise Ratio in an Improved Adaptive Detector," *Proc. ICASSP,* Tampa, FL, Vol. 1, 1985, pp. 572–575.

[43] S. W. Ellingson and G.A. Hampson, "A Subspace-Tracking Approach to Interference Nulling for Phased Array-Based Radio Telescopes," *IEEE Trans. Ant. & Prop.,* Vol. AP-50, No. 1, January 2002, pp. 25–30.

[44] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. Signal Processing,* Vol. 43, No. 1, January 1995, pp. 95–107.

[45] E. Brookner and J. Howell, "Adaptive-Adaptive Array Processing," *Proc. IEEE,* Vol. 74, No. 4, April 1986, pp. 602–604.

[46] H. Yang and M. A. Ingram, "Subarray Design for Adaptive Interference Cancellation," *IEEE Trans. Ant. & Prop.,* Vol. AP-50, No. 10, October 2002. pp. 1453–1459.

Appendix A: Frequency Response Characteristics of Tapped-Delay Lines

The frequency response characteristics of the tapped-delay line filter shown in Figure A-1 can be developed by first considering the impulse response $h(t)$ of the network. For the input signal $x(t) = \delta(t)$ it follows that

$$h(t) = \sum_{i=1}^{2n+1} w_i \delta[t - (i-1)\Delta] \tag{A.1}$$

where w_i, $i = 1, \ldots, 2n+1$ denotes the various complex weights located at the taps in the tapped-delay line having intertap delay spacing equal to Δ.

Taking the Laplace transform of (A.1) yields

$$\mathfrak{L}\{h(t)\} = H(s) = \sum_{i=1}^{2n+1} w_i e^{-s(i-1)\Delta} \tag{A.2}$$

Equation (A.1) represents a sequence of weighted impulse signals that are summed to form the output of the tapped-delay line. The adequacy of the tapped-delay line structure to represent frequency dependent amplitude and phase variations by way of (A.2) depends on signal bandwidth considerations.

Signal bandwidth considerations are most easily introduced by discussing the continuous input signal depicted in Figure A-2. With a continuous input signal, the signals appearing at the tapped-delay line taps (after time $t = t_0 + 2n\Delta$ has elapsed where t_0 denotes an arbitrary starting time) are given by the sequence of samples $x(t_0 + (i-1)\Delta)$, $i = 1, 2, \ldots, 2n+1$. The sample sequence $x(t_0 + (i-1)\Delta)$, $i = 1, \ldots, 2n+1$, uniquely characterizes the corresponding continuous waveform from which it was generated provided that the signal $x(t)$ is band-limited with its highest frequency component $f_{\max}$ less than or equal to one-half the sample frequency corresponding to the time delay, that is,

$$f_{\max} \leq \frac{1}{2\Delta} \tag{A.3}$$

Equation (A.3) expresses the condition that must be satisfied in order for a continuous signal to be uniquely reconstructed from a sequence of discrete samples spaced Δ seconds apart, and it is formally recognized as the "sampling theorem" [1]. Since the total (two-sided) bandwidth of a band-limited signal $x(t)$ is $BW = 2f_{\max}$, it follows that a tapped-delay line can uniquely characterize any continuous signal having $BW \leq 1/\Delta$ (Hz), so $1/\Delta$ can be regarded as the "signal bandwidth" of the tapped-delay line.

Since the impulse response of the transversal filter consists of a sequence of weighted impulse functions, it is convenient to adopt the z-transform description for the filter transfer

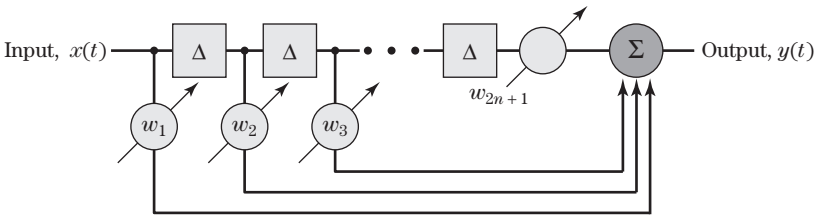


FIGURE A-1 ■ Transversal filter having $2n + 1$ complex weights.

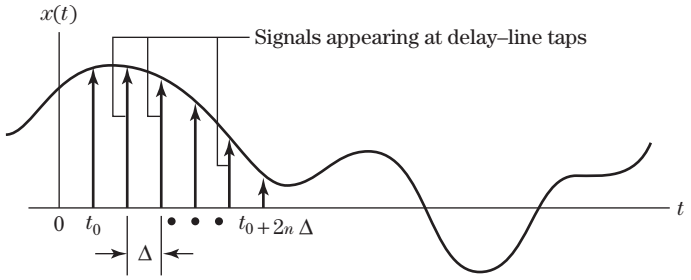


FIGURE A-2 ■ Continuous input signal.

function instead of the (more cumbersome) Laplace transform description of (A.2) by defining

$$z \triangleq e^{s\Delta} \tag{A.4}$$

so that

$$\mathfrak{Z}\{h(t)\} = H(z) = \sum_{i=1}^{2n+1} w_i z^{-(i-1)} \tag{A.5}$$

The frequency response of the transversal filter can then be obtained by setting $s = j\omega$ and considering how $H(j\omega)$ behaves as ω varies. Letting $s = j\omega$ corresponds to setting $z = e^{j\omega\Delta}$, which is a multiple-valued function of ω since it is impossible to distinguish $\omega\Delta = +\pi$ from $\omega\Delta = -\pi$, and consequently

$$H(j\omega) = H\left(j\omega \pm k\frac{2\pi}{\Delta}\right) \tag{A.6}$$

Equation (A.6) expresses the fact that the tapped-delay line transfer function is a periodic function of frequency having a period equal to the signal bandwidth capability of the filter. The periodic structure of $H(s)$ is easily seen in the complex s-plane, which is divided into an infinite number of periodic strips as shown in Figure A-3 [2]. The strip located between $\omega = -\pi/\Delta$ and $\omega = \pi/\Delta$ is called the "primary strip," while all other strips occurring at higher frequencies are designated as "complementary strips." Whatever behavior of $H(j\omega)$ obtains in the primary strip, this behavior is merely repeated in each succeeding complementary strip. It is seen from (A.5) that for $2n + 1$ taps in the tapped-delay line, there will be up to $2n$ roots of the resulting polynomial in z^{-1} that describes $H(z)$. It follows that there will be up to $2n$ zeros in the transfer function corresponding to the $2n$ delay elements in the tapped-delay line.

We have seen how the frequency response $H(j\omega)$ is periodic with period determined by the signal bandwidth $1/\Delta$ and that the number of zeros that can occur across the signal bandwidth is equal to the number of delay elements in the tapped-delay line. It remains

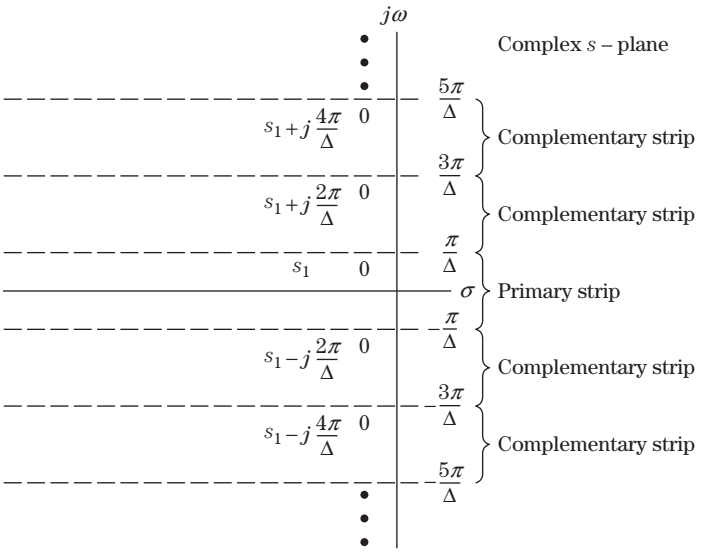


FIGURE A-3 ■ Periodic structure of $H(s)$ as seen in the complex s-plane.

to show that the resolution associated with each of the zeros of $H(j\omega)$ is approximately $1/N\Delta$, where $N =$ number of taps in the tapped-delay line. Consider the impulse response of a transversal filter having N taps and all weights set equal to unity so that

$$H(j\omega) = \sum_{i=0}^{N-1} e^{-j\omega i\Delta} = \frac{1 - e^{-j\omega N\Delta}}{1 - e^{-j\omega\Delta}} \tag{A.7}$$

Factoring $e^{-j\omega(\Delta/2)N}$ from the numerator and $e^{-j\omega(\Delta/2)}$ from the denominator of (A.7) then yields

$$H(j\omega) = \exp\left[j\omega\frac{\Delta}{2}(1-N)\right]\frac{\sin[\omega(\Delta/2)N]}{\sin[\omega(\Delta/2)]} \tag{A.8}$$

$$= N\exp\left[j\omega\frac{\Delta}{2}(1-N)\right]\frac{\left\{\frac{\sin[\omega(\Delta/2)N]}{[\omega(\Delta/2)N]}\right\}}{\left[\frac{\sin[\omega(\Delta/2)]}{[\omega(\Delta/2)]}\right]} \tag{A.9}$$

The denominator of (A.8) has its first zero occurring at $f = 1/\Delta$, which is outside the range of periodicity of $H(j\omega)$ for which $\frac{1}{2}BW = 1/2\Delta$. The first zero of the numerator of (A.8), however, occurs at $f = 1/N\Delta$ so the total frequency range of the principal lobe is just $2/N\Delta$; it follows that the 3 dB frequency range of the principal lobe is very nearly $1/N\Delta$ so the resolution (in frequency) of any root of $H(j\omega)$ may be said to be approximately the inverse of the total delay in the tapped-delay line. In the event that unequal weighting is employed in the tapped-delay line, the width of the principal lobe merely broadens, so the above result gives the best frequency resolution that can be achieved.

The discussion so far has assumed that the frequency range of interest is centered about $f = 0$. In most practical systems, the actual signals of interest are transmitted with a carrier frequency component f_0 as shown in Figure A-4. By mixing the actual transmitted signal with a reference oscillator having the carrier frequency, the transmitted signal can be reduced to baseband by removal of the carrier frequency component, thereby centering the spectrum of the information carrying signal component about $f = 0$. By writing all

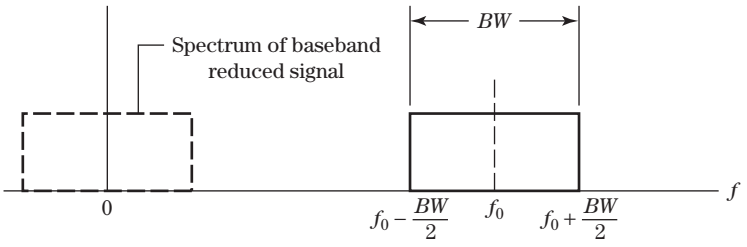


FIGURE A-4 ■ Signal bandwidth *BW* centered about carrier frequency f_0.

signals as though they had been reduced to baseband, no loss of generality results since this merely assumes that the signal spectrum is centered about $f = 0$; the baseband reduced information-carrying component of any transmitted signal is referred to as the "complex envelope," and complex envelope notation is discussed in Appendix B.

A.1 REFERENCES

[1] R. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of Pulse Code Modulation," *Proc. IRE*, Vol. 36, No. 11, November 1948, pp. 1324–1331.

[2] B. C. Kuo, *Discrete-Data Control System*, Prentice-Hall, Englewood Cliffs, NJ, 1970, Ch. 2.

INDEX

Q