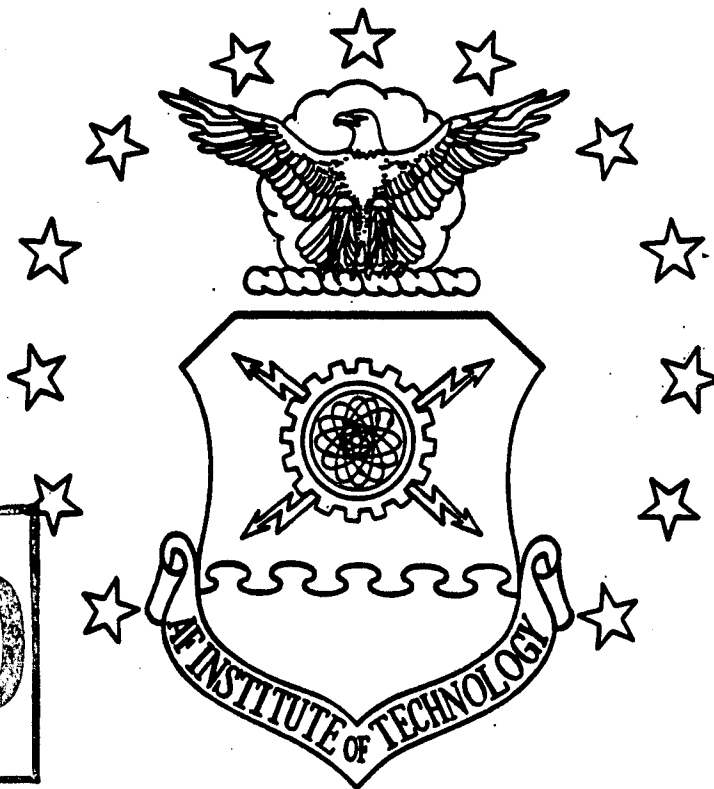
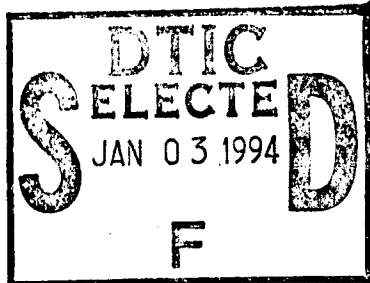




ADAPTIVE BEAMFORMING IN FREQUENCY-DISPERSIVE
MULTIPATH ENVIRONMENTS

THESIS

David A. Murray
Captain, USAF

AFIT/GE/ENG/94D-22

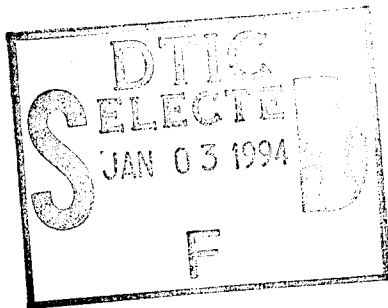
This document has been approved
for public release and sale; its
distribution is unlimited.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

19941228 019

AFIT/GE/ENG/94D-22



ADAPTIVE BEAMFORMING IN FREQUENCY-DISPERSIVE
MULTIPATH ENVIRONMENTS

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

THESIS

David A. Murray
Captain, USAF

AFIT/GE/ENG/94D-22

DTIC QUALITY INSPECTED 2

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

AFIT/GE/ENG/94D-22

ADAPTIVE BEAMFORMING IN FREQUENCY-DISPERSIVE
MULTIPATH ENVIRONMENTS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

David A. Murray, B.A., B.S.E.E.

Captain, USAF

December, 1994

Approved for public release; distribution unlimited

Acknowledgements

This work was sponsored by Wright Laboratories Radar Branch, WL/AARM, at Wright-Patterson AFB. I would like to thank Emil Martinsek, Ed Culpepper, and Richard Kitzerow for their support, and for providing background material which was helpful in completing this thesis. It has been a pleasure working with my advisor, Dr. Michael P. Clark. I thank him for unselfishly devoting much time to me as we worked through various challenges encountered in the course of this thesis. He made it an exciting and positive part of my AFIT experience. A special thanks goes to Dr. R. T. Compton for generously supplying certain information to me, and for giving of his time, both on the phone and in person, to answer questions concerning his work in this area. I would also like to acknowledge the other members of my thesis committee, who reviewed this work and offered constructive criticism: Captains Ronald DeLap and Joseph Sacchini, and Dr. Bruce Suter. My deepest appreciation is reserved for my family: my wife Terry, and my children Austin and Andrea. The innumerable sacrifices they made during the past 18 months, together with their unending love and support, made it possible for me to finish this thesis on schedule.

David A. Murray

Table of Contents

	Page
Acknowledgements	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1-1
II. Background	2-1
2.1 Notation	2-1
2.2 Scenario	2-1
2.3 Problem	2-2
2.4 Approach	2-3
2.5 Time Delays and Phase Shifts	2-5
2.6 Linearly Constrained Minimum Variance Beamforming	2-6
2.6.1 Linear Constraints	2-6
2.6.2 LCMV Approach	2-7
2.6.3 Generalized Sidelobe Canceler	2-8
III. Literature Review	3-1
IV. Analysis	4-1
4.1 Assumptions	4-1
4.1.1 Beamformer Model	4-1
4.1.2 Signals	4-2
4.2 The Covariance Matrices	4-7
4.2.1 Exact Covariance Matrix	4-7

	Page
4.2.2 Estimated Covariance Matrix	4-11
4.3 Eigenvalue Decomposition and Rank-One Updating	4-12
4.4 Exploiting Structure in the Covariance Matrix	4-16
4.5 Computation Time Considerations	4-19
4.5.1 Step 1—Simplify and Deflate	4-22
4.5.2 Step 2—Find the Eigenvalues	4-23
4.5.3 Step 3—Perform Noise Averaging	4-23
4.5.4 Step 4—Calculate the Eigenvectors	4-23
4.5.5 Step 5—Update the Eigenvectors	4-24
4.5.6 Step 6—Stabilize	4-24
4.5.7 Step 7—Compute the Weight Vector	4-24
4.5.8 Step 8—Calculate the Output	4-25
V. Simulations	5-1
5.1 Proof-of-Concept	5-1
5.2 The General Case	5-6
VI. Conclusions and Recommendations	6-1
Appendix A. Block Structure of Eigenvector Update	A-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure	Page
2.1. N-element Array	2-4
2.2. Generalized Sidelobe Canceler	2-9
4.1. N-element Array	4-1
4.2. Power Spectrum of Desired and Jamming Signals	4-3
4.3. Generalized Sidelobe Canceler	4-17
4.4. The Algorithm	4-20
5.1. Deviation from Optimal for the Best Case	5-4
5.2. Optimal vs. Actual for the Worst Case	5-5
5.3. Optimal vs. Actual for the Typical Case	5-5
5.4. Optimal vs. Actual for the Best Case	5-6
5.5. Optimal vs. Actual for the Extreme Case	5-7
5.6. Optimal vs. Actual for the Extreme Case with Forgetting Factor	5-7
5.7. Two Elements–Six Interferers	5-8
5.8. Five Elements–Six Interferers	5-9
5.9. Beampattern for Several Successive Samples	5-10
5.10. Five Elements–18 Interferers	5-10
5.11. Beampattern of Successive Samples	5-11

List of Tables

Table	Page
5.1. Variables Used in the Simulations	5-3

Abstract

It is well known that an antenna array with N degrees of freedom (DOF) can cancel $N - 1$ interferers if they approach the array from directions other than that of the desired signal [3]. Compton [5] has shown that there are cases in which it is possible for an array with N DOF to effectively cancel *more* than $N - 1$ interferers. He shows specifically how this can be done when there is incident upon a two-element array not only the desired signal and a jammer signal, but two Doppler-shifted multipath copies of the jammer signal as well. In this situation the received signal is nonstationary; therefore, updates of the weight vector must use only as many samples as correspond to the time interval over which the signal can be considered locally stationary. In this frequency-dispersive environment, it is desirable to recalculate the weight vector often; in some cases, we would like to do so every sample. However, recalculating the weight vector involves estimating and inverting the covariance matrix of the received signal; using traditional methods, this is an $\mathcal{O}(N^3)$ process, where N represents the degrees of freedom in the array. For large N this becomes time-prohibitive. In this thesis we examine the application of eigenvalue decomposition and rank-one updating [7] of the covariance matrix to reduce the computation. By exploiting the structure of the matrix we find that we are able to significantly reduce the time required for its estimation and inversion. Using these techniques, we can update the weight vector every sample, enabling us to operate effectively in a frequency-dispersive environment.

ADAPTIVE BEAMFORMING IN FREQUENCY-DISPERSIVE MULTIPATH ENVIRONMENTS

I. Introduction

Air combat situations commonly involve the enemy's use of jamming signals. The ability of a radar receiving system to cancel the effect of jammers can easily determine the difference between a successful mission and a failed mission. In addition to the direct jamming signal, there are often copies of this signal which arrive via multiple random paths. This multipath environment further degrades the receiver's ability to discern between desired and undesired signals.

The number of jamming signals may be many, but often may be only one or two. However, the number of multipath reflections of the direct signals may be quite large. It is well-known that beamforming may be used to cancel unwanted signals which arrive from directions other than that of the desired signal. Furthermore, it is understood that in order to cancel such interference, the number of degrees of freedom in the antenna array must be greater than the number of undesired signals. However, it has been shown that in the case of correlated signals, such as we have in a multipath environment, it is possible to cancel more signals than we have DOF by the appropriate weighting and combining of the correlated signals [5]. But, there is a price involved. The computation time required to calculate the weight vector in these situations is not trivial. The use of the standard Wiener solution in the sample matrix inverse, as developed in [15] requires an $\mathcal{O}(N^3)$ calculation, where N represents the number of elements in the array. This is not necessarily a problem in situations where the weight vector needs to be updated infrequently. However, when a time-varying signal with nonstationary characteristics is involved, the weight vector may need to be recalculated at short intervals. This presents us with a challenge.

Compton [4] realized this problem, and approached it from the viewpoint of using frequency-shifted weights in the beamformer in order to lengthen the time over which a particular weight vector would be useful. His results are impressive, and provide one method of attacking this problem.

We have chosen to take a different approach, utilizing some recent (and some not so recent) advances in signal subspace processing and matrix computations. We assume that our scenario includes a single direct jamming signal, together with a potentially large number of multipath reflections of the direct jammer. These multipath signals are characterized by, among other things, induced Doppler shifts, which result in a nonstationary received signal. This, in turn, imposes tight restrictions on the amount of time between weight vector updates. We will solve the problem through the application of rank-one eigenvalue decomposition [1, 7, 12, 17] updates of the estimated covariance matrix. In the process we will find that not only is it possible to operate in a frequency-dispersive multipath environment with more interferers than degrees of freedom, but it is feasible to update the covariance matrix at every sample in real-time.

II. Background

2.1 Notation

Since vectors and matrices are used throughout this thesis, it is important to establish the notation that will be used. Vectors are always represented with lowercase boldface symbols, and are assumed to be column vectors; matrices are denoted by uppercase symbols. A superscript asterisk, $[\cdot]^*$ denotes complex conjugate transpose. In the case of a scalar this reduces to simply the complex conjugate, and in the case of a real vector or matrix to simply the transpose; the context will determine the use. To signify the unconjugated transpose, we will use the standard notation of superscript T , $[\cdot]^T$. Diagonal matrices are often encountered in this thesis; rather than showing full matrix notation for these cases, we will use the shorthand, $\text{diag}(d_1 \cdots d_N)$, to mean the same thing as,

$$\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_N \end{bmatrix}. \quad (2.1)$$

Finally, A^{-1} will signify the inverse of the matrix A .

2.2 Scenario

We are concerned with the situation in which an N -element adaptive beamformer is used to receive some desired signal arriving from a known direction. However, in addition to the desired signal, a jammer signal is impinging upon the array. Furthermore, several multipath copies of the jamming signal are arriving at the beamformer as well. In order to begin to quantify this situation, we will assume that there is precisely one desired signal coming from direction θ_d , and $P + 1$ jamming, or interfering, signals arriving from directions $\theta_0, \theta_1, \dots, \theta_P$. θ_0 is the direction of arrival (DOA) of the direct jamming signal, while θ_1 through θ_P are the DOA's of the P multipath

jamming signals. Also, there is present at the input of each element of the array some amount of thermal noise.

A description of our assumptions about these signals will be detailed later. For now our focus is on one area of concern regarding the interfering signals—induced Doppler shifts. We assume that each multipath interference signal experiences some amount of Doppler shift relative to the direct jamming signal. Thus, in general, a composite signal consisting of one desired signal, $P + 1$ interfering signals, and additive white Gaussian noise will be impinging upon our N -element array. This composite signal will be nonstationary due to the effect of the induced Doppler shifts on the P multipath interferers.

It is well known that, with an N -element array, as many as $N - 1$ interfering signals may be effectively canceled at the output of the array by appropriate adjustment of the weight vector. However, what can be done in the case when there are more, possibly many more, than $N - 1$ interferers? This is a basic question of interest in this thesis.

2.3 Problem

As just stated, our main concern is the case where we have impinging upon an N -element array more than $N - 1$ interfering signals. We also noted that the received signal will be nonstationary due to the Doppler shifts on the multipath signals. This nonstationarity is the heart of the problem. The weight vector of the beamformer adapts itself based upon an estimation of the covariance matrix of the received signal (as shown in more detail later). When this covariance matrix is nonstationary, it must be estimated using only “locally stationary” samples of the received signal. The number of samples that can be used will certainly depend on the degree of degradation that can be tolerated. Fundamentally, it will depend on the amount of Doppler spread¹ that exists in the received signal, as well as upon the sampling rate of the system. The smaller the number of

¹Doppler spread is defined to be the maximum difference in frequency that exists between all possible pairs of interfering signals.

samples that are locally stationary, the less accurate will be the estimate of the received covariance matrix. This is a fundamental limitation in the presence of Doppler-shifted multipath signals and, aside from increasing the sampling rate, cannot be overcome.

We will assume that the Doppler spread is such that we can accumulate a sufficient number of samples to estimate the covariance matrix, and use this estimate to recalculate the new weight vector for use in the beamformer. This being the case, it remains to be shown that it is possible to construct a weight vector that will result in a useful beamformer output. If there are fewer than N interferers, well-known principles of beamforming can be used to cancel the interferers; however, we will consider the general case where there are more than $N - 1$ interferers. Thus, according to the terminology above, there are N elements and $P + 1$ interferers, where $P + 1$ is in general larger than $N - 1$. This represents the first problem—is it possible to use adaptive beamforming in a frequency-dispersive multipath environment (where there are more interfering signals than degrees of freedom), and still produce an acceptable output?

It has been shown [5] that the weight vector will quickly become outdated, with potentially disastrous results, if it is not recalculated often. However, the calculation of the weight vector depends upon the estimation and inversion of the covariance matrix. The inversion process is time-intensive; for an N -element array, it is an $\mathcal{O}(N^3)$ process [2]. This is the second problem encountered—if the answer to the question above is in the affirmative, then is it possible to recalculate the weight vector (in real-time) often enough to make the system practical?

These two questions constitute the specific areas of interest in this thesis.

2.4 Approach

The questions that we seek to answer, from the previous section are:

1. Is it possible to use adaptive beamforming in a frequency-dispersive multipath environment and produce an acceptable output?

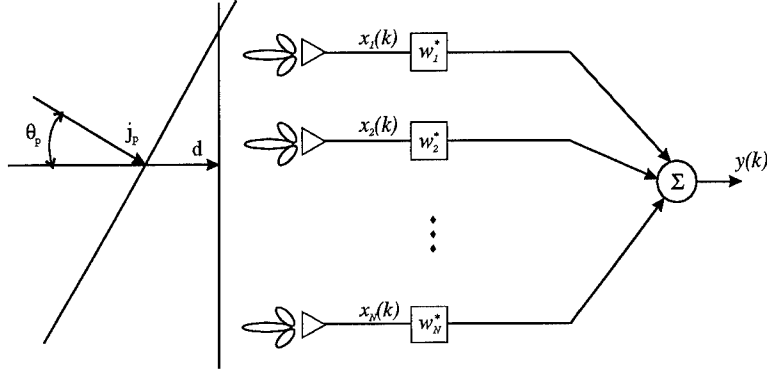


Figure 2.1. N-element Array

2. If so, then is it possible to recalculate the weight vector (in real-time) frequently enough to make the system practical?

We will investigate the first problem by using the N -element model of a narrowband beamformer as shown in Fig. 2.1. During this phase of our investigation, we will demonstrate the viability of using an N -element array in the presence of more than $N - 1$ interferers. Therefore, we will assume that some practical method exists to recalculate the covariance matrix estimate and its inverse at each sample. These calculations will be used to form the weight vector, using the Wiener solution,

$$\mathbf{w}(k) = \hat{R}_{xx}^{-1}(k)\mathbf{d} \quad (2.2)$$

where $\mathbf{w}(k)$ is the weight vector at sample k , $\hat{R}_{xx}(k)$ is the estimated covariance matrix of the received signal at sample k , and $\mathbf{d}$ represents the direction vector for the desired signal, to be described later.

Our approach will be to use linearly constrained, minimum variance (LCMV) beamforming to minimize the output due to the undesired signals. The objective of LCMV beamforming is to minimize the output variance (power) of the array under the constraint that signals from certain directions are processed with specified gains and phases [19].

After demonstrating that the array can give an acceptable output in a frequency-dispersive environment, we will consider the viability of real-time processing for the time-intensive process of estimated covariance matrix inversion.

As Fig. 2.1 implies, our consideration will be limited to narrowband signals. The beamformer model is the standard narrowband beamformer used to receive signals which have a bandwidth that is small relative to the carrier frequency. Thus, there are no tapped delay lines behind the array elements. This limiting of the problem enables us to make an important assumption which we discuss in the following section and use throughout this paper.

2.5 Time Delays and Phase Shifts

This section briefly discusses two types of time delays that are encountered, and how and why they are modeled differently. First, there is an *interelement* delay which occurs due to the finite distance that incoming signals must travel in propagating from one element to the next. This delay is dependent upon the spacing between the array elements and the direction of arrival of the impinging waves. These delays will be modeled as simple phase shifts, which, as Compton [5] points out, is valid for signal bandwidths up to 1% of the carrier frequency. Since narrowband signals are assumed throughout this thesis, we will use Compton's approximation. Secondly, there is a *spatial* delay associated with the jammer signal and its multipath copies as they propagate from the source to the array. We will initially include these delays in our theoretical analysis of the situation. However, the simplifying assumption will eventually be made that all of the multipath copies arrive at the array at the same instant as the direct jammer signal. Again, Compton observes [5] that there will always be some correlation among the multipath signals, and in any case, the nonstationarity of the received signal is not caused by the time delay differences. When the spatial delays are included in our theoretical discussions, they are *not* modeled as simple phase

shifts, because it is assumed possible for these delays to be large enough to decorrelate the jammer signals.

2.6 *Linearly Constrained Minimum Variance Beamforming*

As mentioned previously the LCMV beamformer will be used in our discussions and simulations. Therefore, a short review of its essential elements is in order.

A beamformer weights and sums data from an array of sensor elements to perform spatial filtering. Spatial filtering concerns the ability to distinguish among several signals, depending upon their direction of arrival (DOA) and their temporal frequency content. The LCMV beamformer seeks to exploit this spatial discrimination capability in order to minimize the output power (variance), subject only to some set of linear constraints.

Signals may be classified as either narrowband or broadband; our interest in this thesis is confined to narrowband signals. Narrowband is defined in terms of the fractional bandwidth of the signal. The fractional bandwidth is the signal bandwidth as a percentage of the carrier frequency. Signals whose fractional bandwidths are much less than 1% will be characterized as narrowband, while those with fractional bandwidths much greater than 1% will be called broadband [20].

A narrowband beamformer can be compared to a finite impulse response (FIR) filter. While the response of an FIR filter is a function of frequency, a beamformer's response is a function of both angle of arrival and frequency. Narrowband beamformers are easier to analyze than broadband; since only narrowband beamformers are used in this thesis, our discussion is limited to these.

2.6.1 Linear Constraints. Several types of linear constraints may be used with an LCMV beamformer. We will use what are known as point constraints, in which the beamformer response at a specified frequency and DOA must meet some gain and phase requirements. The response is

constrained at radian frequency ω_o and angle θ_s as

$$\mathbf{w}^* \mathbf{d}(\theta_s, \omega_o) = f_1, \quad (2.3)$$

where f_1 is the desired complex response. Several linear constraints can be expressed as,

$$C^* \mathbf{w} = \mathbf{f}, \quad (2.4)$$

where C is the constraint matrix and $\mathbf{f}$ the response vector. The matrix C is chosen to contain r linear constraints so the above equation represents r linearly independent columns in N unknowns. The weight vector $\mathbf{w}$ has length N . while C is an $N \times r$ matrix. If $N = r$, then $\mathbf{w}$ is uniquely determined by the constraints. To ensure there is a $\mathbf{w}$ which satisfies the constraints, r is chosen to be less than N . Throughout this thesis we will invoke only one constraint—signals arriving from the direction of the desired signal must be passed with unity gain and no phase change.

2.6.2 LCMV Approach. The ultimate goal of LCMV beamforming is to pass all desired signals with some prescribed effect while minimizing the resultant output power due to all other signals. This is accomplished by constraining the beamformer response in desired directions while minimizing the overall output power (variance). Formally stated, we seek to,

$$\min_{\mathbf{w}}(P_o), \text{ subject to } C^* \mathbf{w} = \mathbf{f}. \quad (2.5)$$

P_o represents the output power, and may be expressed as,

$$P_o = E[|y|^2] = E[yy^*] = \mathbf{w}^* E[\mathbf{x}\mathbf{x}^*] \mathbf{w}. \quad (2.6)$$

Therefore, the problem may be recast as,

$$\min_{\mathbf{w}}(\mathbf{w}^* R_{xx} \mathbf{w}), \text{ subject to } C^* \mathbf{w} = \mathbf{f}, \quad (2.7)$$

where $R_{xx} = E(\mathbf{x}\mathbf{x}^*)$ represents the exact covariance matrix of the received signal $\mathbf{x}$. To solve this problem, first make an educated guess that the solution is given by,

$$\mathbf{w}_g = R_{xx}^{-1} C [C^* R_{xx}^{-1} C]^{-1} \mathbf{f}. \quad (2.8)$$

Eq. 2.8 is a reasonable guess since C is full rank and R_{xx} is positive definite, thus ensuring that the inverses exist [20]. Other solutions of the form $\mathbf{w} = \mathbf{w}_g + \mathbf{a}$ meet the constraint so long as $C^* \mathbf{a} = 0$. Substituting this solution back into the objective function gives (momentarily suppressing the subscript on R_{xx}),

$$\begin{aligned} \mathbf{w}^* R \mathbf{w} &= (\mathbf{w}_g^* + \mathbf{a}^*) R (\mathbf{w}_g + \mathbf{a}) \\ &= \mathbf{w}_g^* R \mathbf{w}_g + \mathbf{w}_g^* R \mathbf{a} + \mathbf{a}^* R \mathbf{w}_g + \mathbf{a}^* R \mathbf{a} \\ &= \mathbf{w}_g^* R \mathbf{w}_g + \mathbf{g}^* [C^* R^{-1} C]^{-1} C^* R^{-1} R \mathbf{a} + \mathbf{a}^* R R^{-1} C [C^* R^{-1} C]^{-1} \mathbf{f} + \mathbf{a}^* R \mathbf{a} \\ &= \mathbf{w}_g^* R \mathbf{w}_g + \mathbf{f}^* [C^* R^{-1} C]^{-1} C^* \mathbf{a} + \mathbf{a}^* C [C^* R^{-1} C]^{-1} \mathbf{f} + \mathbf{a}^* R \mathbf{a} \\ &= \mathbf{w}_g^* R \mathbf{w}_g + \mathbf{a}^* R \mathbf{a}, \end{aligned} \quad (2.9)$$

since $C^* \mathbf{a} = 0$. We see that all other possible solutions which do not violate the original constraint make the objective function larger. Therefore, $\mathbf{w}_g$ must be a minimum.

2.6.3 Generalized Sidelobe Canceler. The generalized sidelobe canceler (GSC) is a convenient way to represent the function of an LCMV beamformer. Consider Fig. 2.2. The derivation that follows is adapted from Clark [2]. Without a loss of generality, we can force the constraint matrix C to be composed of orthonormal columns. We can then find some matrix, C_n , such that

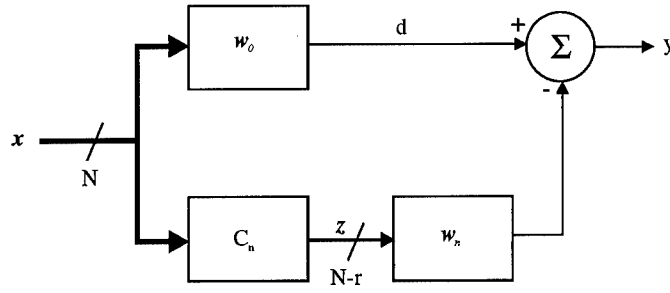


Figure 2.2. Generalized Sidelobe Canceler

$[C \mid C_n]$ is a unitary matrix. Now the subspaces defined by C and C_n are orthogonal and together span all of $C^{N \times N}$ space. Split w into two orthogonal components w_0 and $-v$ such that,

$$w = w_0 - v. \quad (2.10)$$

Choose w_0 so that it lies in the range of C , and $-v$ so that it lies in the range of C_n . Then the constraint equation can be written as,

$$C^*(w_0 - v) = f. \quad (2.11)$$

But $C^*v = 0$, since there exists some $(N - r)$ -element vector w_n , such that $v = C_n w_n$; therefore,

$$C^*w_0 = f. \quad (2.12)$$

Now C is $N \times r$, with $r < N$, and w_0 is $r \times 1$; therefore, this equation represents an underdetermined problem. Out of all possible solutions, we seek the one that will minimize the white noise gain, w^*w , of the beamformer. This is the minimum least squares solution given by,

$$w_0 = C(C^*C)^{-1}f = Cf. \quad (2.13)$$

Now, $\mathbf{w}$ can be expressed as,

$$\mathbf{w} = \mathbf{w}_0 - C_n \mathbf{w}_n. \quad (2.14)$$

The first term on the right side of this equation is often called the quiescent weight vector. It is given by $C\mathbf{f}$, depending only on the constraints and not on the data; it is implemented in the upper half of Fig. 2.2. The second term is incorporated in the lower branch of the GSC and is essentially an unconstrained adaptive beamformer. The number of adaptive weights is $N - r$. The constraint matrix C preserves the desired signal. But the matrix C_n blocks any portion of the data that is contained in the space spanned by the columns of C (i.e., the signal subspace). Consequently, no part of the received signal that falls in the signal subspace is allowed to pass through the lower branch of the GSC. The matrix C_n is for this reason often called the *blocking* matrix. The $(N - r)$ -element vector, $\mathbf{z} = C_n^* \mathbf{x}$, thus contains none of the desired signal, but only interference and noise. On the other hand the signal d , in the upper branch, contains both signal and noise subspace information. We can therefore use the weight vector $\mathbf{w}_n$ to modify $\mathbf{z}$ in such a manner as to cancel the noise subspace portion of d without affecting the signal subspace part. This is the goal of the GSC.

Finally, the output of the beamformer may be expressed as,

$$y = \mathbf{w}^* \mathbf{x} = (\mathbf{w}_0 - C_n \mathbf{w}_n)^* \mathbf{x} = \mathbf{w}_0^* \mathbf{x} - \mathbf{w}_n^* C_n^* \mathbf{x}. \quad (2.15)$$

The standard diagram of a GSC (Fig. 2.2) is generated by this equation.

Based upon this explanation of the GSC, we can now express the LCMV beamformer with the following *unconstrained* equation:

$$\min_{\mathbf{w}_n} (\mathbf{w}_0 - C_n \mathbf{w}_n)^* R_{xx} (\mathbf{w}_0 - C_n \mathbf{w}_n). \quad (2.16)$$

The solution [20] to this equation is given by,

$$\mathbf{w}_n = (C_n^* R_{xx} C_n)^{-1} C_n^* R_{xx} \mathbf{w}_0. \quad (2.17)$$

Examination of this expression reveals a similarity with Eq. 2.2. Consider the autocovariance of $\mathbf{z}$,

$$R_{zz} = E\mathbf{z}\mathbf{z}^* = EC_n^* \mathbf{x}\mathbf{x}^* C_n = C_n^* E\mathbf{x}\mathbf{x}^* C_n = C_n^* R_{xx} C_n. \quad (2.18)$$

Now consider the cross-covariance of the signals $\mathbf{z}$ and d in Fig. 2.2,

$$\mathbf{r}_{zd} = E\mathbf{z}d^* = EC_n^* \mathbf{x}\mathbf{x}^* \mathbf{w}_0 = C_n^* R_{xx} \mathbf{w}_0. \quad (2.19)$$

Comparing Eqs. 2.18 and 2.19 with Eq. 2.17, we can rewrite Eq. 2.17 as,

$$\mathbf{w}_n = R_{zz}^{-1} \mathbf{r}_{zd}, \quad (2.20)$$

which shows the similarity between the standard Wiener solution, and the solution as implemented with a GSC.

Clearly, the unconstrained expression (Eq. 2.16) of the GSC represents a simplification over the constrained one (Eq. 2.7). However, both accurately portray the operation of an LCMV beamformer. We will use the GSC implementation throughout the remainder of this paper; the reader should now be able to discern the differences between the two signals $\mathbf{x}$ and $\mathbf{z}$, as well as between the covariances, R_{xx} , and R_{zz} . We will have occasion to refer to and manipulate each of these signals at various points in our discussions, depending on which is more convenient or effective in a particular situation.

III. Literature Review

The usefulness of adaptive arrays has been shown time and again in the literature. However, while the problem addressed in this thesis finds much support in the literature in terms of surrounding issues and potential solutions, very little was found dealing with the specific problem presented in the previous chapter. Compton [4] was the notable exception. In fact, we became convinced of the need for this thesis as a direct result of a briefing given by Dr. R. T. Compton [4] at the Wright Laboratories Radar Branch in the early summer of 1994. The briefing covered the effects that multipath jamming has on an adaptive array. It became clear to us as a result of the briefing and ensuing discussion that some recent discoveries in signal subspace processing could be profitably applied to problems inherent in frequency-dispersive environments. Thus began a search of the literature for relevant information regarding this problem and its solutions.

Compton [4] describes the jammer multipath model for the particular case of a two-element array in an environment consisting of the desired signal, three interfering signals (two multipath copies in addition to the direct jamming signal), and noise. He begins his analysis by considering the special case where the multipath interferers have no Doppler shift associated with them, but arrive at the array with varying amounts of delay (time-dispersive only). He then proceeds to consider the opposite special case where each multipath interferer is Doppler-shifted, but arrives at the array at the same time as the direct interfering signal (frequency-dispersive only). Finally, he considers the general case where both time and frequency dispersion occur (doubly-dispersive). In each case he is able to show how it is theoretically possible to obtain perfect cancellation of the jammer signals.

In the frequency-dispersive environment Compton shows that a nonstationary received signal causes the calculated weight vector to be useful for only a short period of time. He defines the *3-dB time duration* as the length of time over which a given calculated weight vector (based on the exact covariance matrix) will continue to result in a signal-to-interference-plus-noise ratio (SINR)

that is within 3 dB of the optimum value. The fundamental problem is the inability to estimate the received covariance matrix, calculate its inverse, and determine the resulting weight vector, within a short enough time-span to avoid obsolete weight vectors. Compton demonstrates how this time duration can be improved (i.e., lengthened) by using frequency-shifted weights as part of the weight vector.

Our approach in this paper is to seek to use existing signal processing techniques in a manner that enables us to update the weight vector often enough and rapidly enough that the nonstationary received signal does not pose a problem. Several papers were reviewed that provided helpful insight into the problem.

We are indebted to Bunch *et al.* [1] for their work in developing and explaining eigenvalue decomposition and rank-one updating of symmetric matrices. DeGroat and Roberts [7] also added insight in this area and expanded the work of Bunch to include a method whereby the eigenvalues and eigenvectors remain stable regardless of the number of updates performed—this is a critical need in our application. DeGroat and Roberts [6] also provided help on noise averaging which is a crucial step in simplifying the computation required in our scenario.

Compton [5] provided insight into the problems associated with operating in a frequency-dispersive environment. He also enabled us to have a point of reference for comparison purposes, giving us confidence at the outset that we were on the right track. His work, however, deals strictly with the use of frequency-shifted weights, whereas we were pursuing the feasibility of calculating the weight vector often enough to not need such weights.

Since our desire was to work with linearly constrained, minimum variance (LCMV) beamforming, Van Veen's tutorial [19] on this subject proved quite helpful. He also provided insight into the use of the generalized sidelobe canceler (GSC) which transforms the LCMV approach from a constrained problem into a simpler unconstrained problem. Van Veen and Buckley [20] were likewise helpful in this regard.

For help with the basics of signal processing, Oppenheim and Schaffer [13] were indispensable. Noble and Daniel [11] were helpful in reviewing the principles of linear algebra, especially in the area of the singular value decomposition. Similarly, Golub and Van Loan's text [9] was good for both matrix computation intricacies, and the issue of the number of operations required for various algorithms.

Scharf's [16] section dealing with Householder transformations was beneficial in understanding how to implement the deflation portion of the EVD problem. Golub [9] also gave some direction in this area. Scharf [16] has an excellent section dealing with structured covariance matrices which guided us in the exploitation of the structure of the covariance matrix in the environment under study.

IV. Analysis

In order to analyze the problems outlined in Chapter II, it is important to state in a lucid manner the assumptions being made regarding the various signals which comprise the total input to our array. The first section in this chapter considers those assumptions. Then both the exact covariance matrix and the estimated covariance matrix are examined, with an ensuing discussion on the manner in which the aforementioned problems are solved.

4.1 Assumptions

Important assumptions are made concerning the beamformer itself, as well as the signals impinging upon the beamformer; all are covered below.

4.1.1 Beamformer Model. An N-element array is shown in Fig. 4.1. The elements of the array are assumed to be directional with interelement spacing $\lambda/2$, where λ is the wavelength of the carrier signal received by the array. The output of each element, $x_n(k)$, is a discrete-time sample

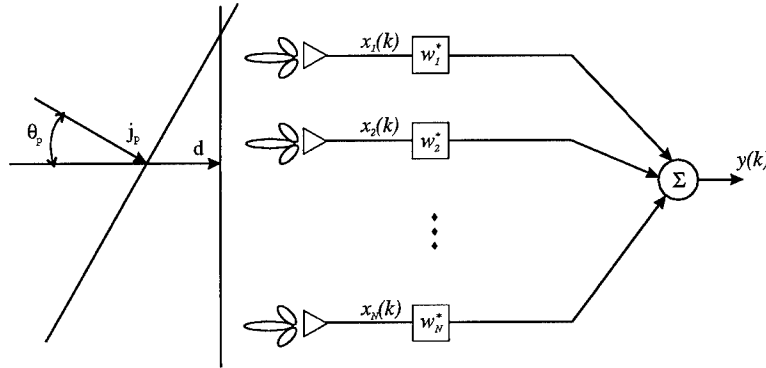


Figure 4.1. N-element Array

of its input. These signals are processed by the adaptive weights, $w_n^*(k)$, and then summed to form the output,

$$y(k) = w_1^*(k)x_1(k) + w_2^*(k)x_2(k) + \cdots + w_N^*(k)x_N(k)$$

$$= \sum_{n=1}^N w_n^*(k) x_n(k) \quad (4.1)$$

where we have used the conventional method of multiplying the signal by the complex conjugate of the weights in order to simplify later notation.¹

4.1.2 Signals. All signals incident upon the array are assumed to be *continuous*-time waveforms, while the signals at the output of the elements are baseband *discrete*-time signals. The desired signal, $d(t)$, is incident on the array broadside, i.e. $\theta_d = 0$ deg, while $P + 1$ interferers arrive from directions $\theta_0, \theta_1, \dots, \theta_P$ respectively. These interferers originate from a single source, with $j_0(t)$ representing the direct jammer signal, and $j_1(t), \dots, j_P(t)$ representing multipath copies of $j_0(t)$. Each multipath copy has a different amount of time delay, frequency shift, and phase shift relative to $j_0(t)$, as well as a different amplitude. Also present at the input to each element is additive white Gaussian noise, which we shall label $n_1(t), \dots, n_N(t)$.

Let the signal at the output of the elements be denoted by the vector,

$$\mathbf{x}(k) = [x_1(k) \ \cdots \ x_N(k)]^T, \quad (4.2)$$

and let the weight vector, $\mathbf{w}(k)$, be given by

$$\mathbf{w}(k) = [w_1(k) \ \cdots \ w_N(k)]^T. \quad (4.3)$$

This gives an expression for the output of the array the inner product of $\mathbf{w}(k)$ and $\mathbf{x}(k)$,

$$y(k) = \mathbf{w}^*(k) \mathbf{x}(k). \quad (4.4)$$

¹We will use the superscript * to indicate complex conjugate transpose. Note in the case of scalars that this is simply the complex conjugate.

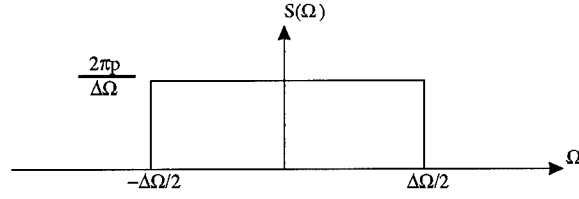


Figure 4.2. Power Spectrum of Desired and Jamming Signals

Here we note the dependence of the weight vector $\mathbf{w}$ on the sample value k , since $\mathbf{w}$ will be updated at every sample in our model.

4.1.2.1 Desired Signal. Numerical subscripts will be used when appropriate to indicate a signal associated with a particular element. Thus, for example, $d_1(t)$ refers to the signal impinging upon element 1 due to the desired signal $d(t)$.

Now, since $d(t)$ arrives broadside to the array, $d_i(t)$ equals $d_j(t)$, for $1 \leq i, j \leq N$, and $d_i(t)$ may be denoted simply as $d(t)$. We assume that $d(t)$ is a complex, circularly-symmetric [18], narrowband Gaussian random process with zero mean. It is formed in the transmitter by modulating a carrier of frequency Ω_c with a baseband narrowband random process, $u_d(t)$, and may be expressed as,

$$d(t) = u_d(t)e^{j\Omega_c t}. \quad (4.5)$$

We assume that $u_d(t)$, the complex envelope of $d(t)$, has a power spectral density, $S(\Omega)$, as shown in Fig. 4.2.

This portion of the received signal can be expressed in vector form as,

$$\mathbf{x}_d(t) = [d(t) \cdots d(t)]^T = d(t)[1 \cdots 1]^T = d(t)\mathbf{1}, \quad (4.6)$$

where $\mathbf{1}$ is an N -element column vector of ones. We assume that the receiving array performs the operations of down-converting, sampling, and digitizing the signal. The sampling rate is assumed

to satisfy the well-known Nyquist criterion. Thus, at the output of the array elements we have,

$$\mathbf{x}_d(k) = u_d(k)\mathbf{1}. \quad (4.7)$$

4.1.2.2 Jamming Signals. The direct jamming signal, $j_0(t)$, is also a complex, circularly-symmetric, narrowband Gaussian random process with zero mean. It may be expressed as,

$$j_0(t) = u_j(t)e^{j\Omega_c t} \equiv j(t), \quad (4.8)$$

where $u_j(t)$, also having a spectrum as in Fig. 4.2, is the complex envelope of $j(t)$.

The multipath jammer signals are amplitude-changed, time-delayed, frequency- and phase-shifted versions of $j(t)$. Consider the multipath signal from direction θ_i ; it may be expressed in terms of $j(t)$ as,

$$j_i(t) = A_i j(t - T_i) e^{j(\Omega_i t + \psi_i)}, \quad (4.9)$$

where $j(t - T_i)$ represents a copy of the direct jammer signal, $j(t)$, delayed by an amount of time T_i , with a change in amplitude represented by A_i , some amount of Doppler shift Ω_i , and a change in phase relative to $j(t)$ of ψ_i . The amplitude change and the phase angle can be combined into a single complex amplitude term, which we will denote $s_i(t)$,

$$s_i(t) = A_i e^{j\psi_i}. \quad (4.10)$$

Now the i^{th} jammer multipath signal can be expressed as,

$$j_i(t) = s_i j(t - T_i) e^{j\Omega_i t}. \quad (4.11)$$

We will assume that each component of this equation is deterministic, with the obvious exception of the $j(t - T_i)$ term, which represents a delayed version of the direct jamming signal. In other words the amplitude variations, phase shifts, and Doppler shifts induced on the multipath signals will be set at arbitrary values, rather than randomly changing.

In considering the portion of the signal at the input of each element due to these jamming signals, we must take into account the direction of arrival (DOA) of the signals. From the geometry of Fig. 4.1 it can be shown that a wavefront arriving at element n from direction θ_i must travel an additional distance l_i given by

$$l_i = \frac{\lambda}{2} \sin \theta_i, \quad (4.12)$$

in order to arrive at element $n + 1$. This will produce a phase lag of signal $j(t - T_i)$ at element $n + 1$ relative to element n given by,²

$$\phi_i = \frac{2\pi l_i}{\lambda} = \pi \sin \theta_i. \quad (4.13)$$

Taking element 1 as our zero-phase reference point, we have as the total jamming signal incident on element 1,

$$x_{1j}(t) = j(t) + s_1 j(t - T_1) e^{j\Omega_1 t} + \dots + s_P j(t - T_P) e^{j\Omega_P t}, \quad (4.14)$$

and, in general, on element n ,

$$x_{nj}(t) = e^{-j(n-1)\phi_0} j(t) + e^{-j(n-1)\phi_1} s_1 j(t - T_1) e^{j\Omega_1 t} + \dots + e^{-j(n-1)\phi_P} s_P j(t - T_P) e^{j\Omega_P t}. \quad (4.15)$$

We can rewrite this using matrix and vector notation as,

$$\mathbf{x}_j(t) = A D_\Omega(t) D_s \mathbf{j}(t), \quad (4.16)$$

²We are assuming that the interelement time delay can be accurately modeled as simply a phase shift, as discussed in Section 2.5. This is valid as long as the bandwidth of the signal is less than approximately 1% of the carrier frequency [5].

where the following definitions apply. A is an $N \times (P + 1)$ matrix whose columns represent steering vectors which account for the phase shifts of the various interferers due to each interferer's direction of arrival. It is given by

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ e^{-j\phi_0} & e^{-j\phi_1} & \dots & e^{-j\phi_P} \\ \vdots & & \ddots & \vdots \\ e^{-j(N-1)\phi_0} & e^{-j(N-1)\phi_1} & \dots & e^{-j(N-1)\phi_P} \end{bmatrix}. \quad (4.17)$$

$D_\Omega(t)$ is a $(P + 1) \times (P + 1)$ diagonal matrix of Doppler shifts, and is given by,

$$D_\Omega(t) = \text{diag}(1 \quad e^{j\Omega_1 t} \quad \dots \quad e^{j\Omega_P t}), \quad (4.18)$$

where the first element of the matrix is unity since, by definition, the direct jammer signal experiences no Doppler shift. $D_s(t)$ is also a $(P + 1) \times (P + 1)$ diagonal matrix, representing the amplitude and phase changes incurred by the multipath interferers relative to the direct jammer signal. It has the form,

$$D_s = \text{diag}(s_0 \quad s_1 \quad \dots \quad s_P), \quad (4.19)$$

where s_0 clearly equals unity, since the direct jammer signal has no amplitude change or phase shift relative to itself. Finally, $\mathbf{j}(t)$ is a $(P + 1)$ -element vector representing each interfering signal, with its respective spatial time delay,

$$\mathbf{j}(t) = [j(t) \quad j(t - T_1) \quad \dots \quad j(t - T_P)]. \quad (4.20)$$

Then, with $\mathbf{x}_j(t)$ present at the input, we get at the output of the array,

$$\mathbf{x}_j(k) = A D_\omega(k) D_s \mathbf{j}(k), \quad (4.21)$$

where the conversion to discrete-time signals results in the normal changes from analog frequencies in Hertz to digital frequencies in radians according to the familiar relationship,

$$\omega_i = \Omega_i T_s. \quad (4.22)$$

T_s is the sampling period in the continuous-to-discrete conversion process, and is chosen so that Nyquist's criterion is met; both the bandwidth, $\Delta\omega$, of the original baseband signal, and the Doppler spread³ must be taken into account in meeting this criterion.

Considering Eq. 4.21, it is clear that the only element of $\mathbf{x}_j(k)$ which is not deterministic is $\mathbf{j}(k)$.

4.1.2.3 Noise. Complex, additive Gaussian noise is assumed to exist at each element's input. Each noise signal is circularly symmetric, zero-mean, and uncorrelated with the other noise signals. This noise, as it appears at the output of the elements, is given by,

$$\mathbf{x}_n(k) = \begin{bmatrix} n_1(k) & n_2(k) & \cdots & n_N(k) \end{bmatrix}^T, \quad (4.23)$$

and is distributed as $N[0, \sigma^2 I]$, where σ^2 is the variance of the noise.

4.2 The Covariance Matrices

4.2.1 Exact Covariance Matrix. The expression for the received signal at the output of the array elements can now be written as,

$$\mathbf{x}(k) = \mathbf{x}_d(k) + \mathbf{x}_j(k) + \mathbf{x}_n(k). \quad (4.24)$$

³Doppler spread is defined as the maximum difference in frequency that exists between all possible pairs of interfering signals.

This is the signal of interest; we now consider its covariance matrix, $R_x(k)$. Since the individual components are all uncorrelated, the covariance matrix may be written as,

$$R_x(k) = R_d(k) + R_j(k) + R_n(k). \quad (4.25)$$

Consider first the desired signal. From Eq. 4.7,

$$R_d = E\mathbf{x}_d\mathbf{x}_d^* = Eu_d u_d^* \mathbf{1}\mathbf{1}^T = p_d \mathbf{1}\mathbf{1}^T, \quad (4.26)$$

where E is the expectation operator, $p_d \equiv |u_d|^2$ is the power (variance) in the desired signal, $\mathbf{1}$ is a column vector of ones, and superscript T is the transpose operator. Note that R_d is not a function of k . Now consider the noise. From Eq. 4.23, and the fact that n_i and n_j are uncorrelated for $i \neq j$, we have,

$$R_n = E\mathbf{x}_n\mathbf{x}_n^* = \sigma^2 I, \quad (4.27)$$

where σ^2 is the variance of each noise signal, and I is the identity matrix.

The jamming signal is not as straightforward as the noise and the desired signal. From Eq. 4.21,

$$\begin{aligned} R_j(k) &= E\mathbf{x}_j(k)\mathbf{x}_j^*(k) \\ &= E[AD_\omega(k)D_s\mathbf{j}(k)\mathbf{j}^*(k)D_s^*D_\omega^*(k)A^*]. \end{aligned} \quad (4.28)$$

Considering Eqs. 4.17 through 4.20, and earlier comments upon these equations, it is clear that $\mathbf{j}(k)$ is the only random element under the expectation operator in Eq. 4.28. Therefore, the equation may be rewritten as,

$$R_j(k) = AD_\omega(k)D_s E[\mathbf{j}(k)\mathbf{j}^*(k)]D_s^*D_\omega^*(k)A^*. \quad (4.29)$$

Now from Eq. 4.20, the term within the expectation portion of this equation is given by,

$$\mathbf{j}(k)\mathbf{j}^*(k) = \begin{bmatrix} |j(k)|^2 & j(k)j^*(k-T_1) & \cdots & j(k)j^*(k-T_P) \\ j^*(k)j(k-T_1) & |j(k-T_1)|^2 & \cdots & j(k-T_1)j^*(k-T_P) \\ \vdots & & & \vdots \\ j^*(k)j(k-T_P) & j^*(k-T_1)j(k-T_P) & \cdots & |j(k-T_P)|^2 \end{bmatrix}, \quad (4.30)$$

where for convenience the notation $j(k-T_i)$ is used to indicate $j(kT_s-T_i)$. We limit our consideration to the special case where all of the spatial time delays, T_i , are equal to zero; i.e. the jammer signal and its copies all arrive at the array simultaneously.⁴ Thus Eq. 4.30 may be rewritten as,

$$\mathbf{j}(k)\mathbf{j}^*(k) = \begin{bmatrix} |j(k)|^2 & |j(k)|^2 & \cdots & |j(k)|^2 \\ |j(k)|^2 & |j(k)|^2 & \cdots & |j(k)|^2 \\ \vdots & & \ddots & \vdots \\ |j(k)|^2 & |j(k)|^2 & \cdots & |j(k)|^2 \end{bmatrix} = \mathbf{1}\mathbf{1}^T |j(k)|^2. \quad (4.31)$$

Then, recalling Eq. 4.8 and taking the expected value of Eq. 4.31 gives,

$$\mathbf{E}\mathbf{j}(k)\mathbf{j}^*(k) = \mathbf{1}\mathbf{1}^T \mathbf{E}|j(k)|^2 = p_j \mathbf{1}\mathbf{1}^T, \quad (4.32)$$

where $p_j \equiv |u_j|^2$ is the power in the direct jamming signal.

Now, using this result in Eq. 4.29 gives us,

$$R_j(k) = p_j A D_\omega(k) D_s \mathbf{1}\mathbf{1}^* D_s^* D_\omega^*(k) A^*. \quad (4.33)$$

This component of the covariance matrix is clearly a function of the sample time k , and dictates the use of time-varying weights. Let us summarize the results of this section thus far by displaying

⁴This simplification has no effect on the nonstationarity of the received signal. Compton [5] points out that this assumption is not far from the actual case, since the multipath signals will generally be highly correlated.

the individual covariances together with the expression for the overall covariance of the received signal:

$$R_d = p_d \mathbf{1}\mathbf{1}^T, \quad (4.34)$$

$$R_n = \sigma^2 I, \quad (4.35)$$

$$R_j(k) = p_j A D_\omega(k) D_s \mathbf{1}\mathbf{1}^* D_s^* D_\omega^*(k) A^*, \quad (4.36)$$

$$R_x(k) = R_d + R_j(k) + R_n. \quad (4.37)$$

The only term in Eq. 4.37 that changes with time is the $R_j(k)$ term, given by Eq. 4.36. Clearly the matrix $D_\omega(k)$ is the only time-varying element of Eq. 4.37; thus, all of the nonstationarity in the received signal is due to the presence of the multipath jammer signals with their associated Doppler shifts.

We are now in a position to determine the theoretically optimum weight vector for our array. Using the exact covariance matrix in the well-known Wiener solution gives,

$$\mathbf{w}(k) = R_x^{-1}(k) \mathbf{d}, \quad (4.38)$$

where $\mathbf{d}$ is the direction vector given by $\mathbf{d} = \mathbf{1}$, since our desired signal arrives broadside to the array [5]. So, substituting (4.37) into (4.38) gives us the optimal weight vector to use in attempting to cancel the interferers at the output of our beamformer. As discussed in Section 2.6.3, we will use the generalized sidelobe canceler implementation of an LCMV beamformer (see Fig. 2.2). Thus, we will use Eqs. 2.18 and 2.19 to form R_{zz} and $\mathbf{r}_{zd}$ from R_{xx} , and then use Eq. 2.20 to find the optimal $\mathbf{w}_n(k)$. Plots of the optimal SINR obtained using the exact covariance matrix are shown in Chapter V.

4.2.2 *Estimated Covariance Matrix.* In a practical system, of course, there is no access to the exact covariance matrix; it must be estimated using samples of the actual data. Reed *et al.* [15] used the maximum likelihood estimate of the covariance matrix (commonly called the sample covariance matrix, or SCM) given by,

$$\hat{R}(k) = \frac{1}{m} \sum_{i=k-m+1}^k \mathbf{z}_i \mathbf{z}_i^*, \quad (4.39)$$

to demonstrate that the Wiener solution could be used to obtain a rapidly converging adaptive algorithm. This approach works well and converges quite rapidly; however, the computation required to determine the inverse of the sample covariance matrix is $\mathcal{O}(N^3)$, where N is the number of degrees of freedom in the array. Since we are working in an environment in which the signal's statistics are time-varying, it would be wise to consider the use of an estimate which de-emphasizes samples in the estimate as they become further removed from the current sample. This suggests the use of an estimate such as,

$$\tilde{R}(k) = (1 - \alpha) \sum_{i=k-m+1}^k \alpha^{k-i} \mathbf{z}_i \mathbf{z}_i^*, \quad (4.40)$$

where α is a constant in $(0,1]$ that acts as the “forgetting factor” that de-emphasizes old data. It is easy to show that the expected value of this expression is given by,

$$\mathbb{E} \tilde{R}(k) = (1 - \alpha^m) R(k). \quad (4.41)$$

Thus, for large values of m this is an unbiased estimator of the covariance. The GSC implementation requires us to also estimate the cross-correlation $\mathbf{r}_{zd}$,

$$\tilde{\mathbf{r}}_{zd} = (1 - \alpha) \sum_{i=k-m+1}^k \alpha^{k-i} \mathbf{z}_i d_i^*. \quad (4.42)$$

Now, these relationships could be used directly; however, the $\mathcal{O}(N^3)$ computation requirement for the inverse of $\tilde{R}_{zz}$ is prohibitive. Since we want to update the estimates every sample, we consider a recursive definition of the above estimates, and then apply a rank-one eigenvalue decomposition update procedure to them. This leads naturally into our next topic.

4.3 Eigenvalue Decomposition and Rank-One Updating

Eq. 4.42 can be written recursively as,

$$\tilde{\mathbf{r}}_{zd}(k+1) = \alpha \tilde{\mathbf{r}}_{zd}(k) + (1-\alpha) \mathbf{z}(k+1) d^*(k+1). \quad (4.43)$$

with the recursion initialized to $\tilde{\mathbf{r}}_{zd}(1) = \mathbf{z}(1) d^*(1)$. We will use this recursion as it stands, since it is an $\mathcal{O}(N)$ operation. Eq. 4.40 can be written recursively as,

$$\tilde{R}_{zz}(k+1) = \alpha \tilde{R}_{zz}(k) + (1-\alpha) \mathbf{z}(k+1) \mathbf{z}^*(k+1). \quad (4.44)$$

with the recursion initialized to $\tilde{R}_{zz}(1) = \mathbf{z}(1) \mathbf{z}^*(1)$. Finding the inverse of $\tilde{R}$ is an $\mathcal{O}(N^3)$ process. We will use a rank-one eigenvalue decomposition (EVD) update procedure to reduce this computational requirement. The EVD of a symmetric matrix R has the form,

$$R = U D U^*, \quad (4.45)$$

where D is a diagonal matrix containing the eigenvalues of R in nonincreasing order, and U is the (unitary) matrix of corresponding eigenvectors. The rank-one EVD update of a symmetric matrix was first proposed by Golub in [8] and then developed into an algorithm by Bunch *et al.* in [1]. While Bunch's method is robust and guaranteed to converge, it suffers from a linear build-up of round-off error which becomes unacceptable when large numbers of updates are performed. DeGroat

and Roberts [7] developed an improved rank-one EVD update method that is more efficient than Bunch's, and more importantly, is immune to accumulated round-off errors.

If the EVD of R is given by UDU^* , then the basic algorithm of Bunch can be broken down into four distinct steps as follows:

1. $\tilde{R} = R + \mathbf{z}\mathbf{z}^* = U(D + \gamma\gamma^*)U^*$, $\gamma = U^*\mathbf{z}$ (Expose the underlying structure)
2. $\tilde{R} = UH(D + \beta\beta^*)H^*U^*$, $\beta = H^*\gamma$ (Deflate the problem)
3. $\tilde{R} = UH(Q\tilde{D}Q^*)H^*U^*$ (Find the eigenstructure of $D + \beta\beta^*$)
4. $\tilde{R} = \tilde{U}\tilde{D}\tilde{U}^*$, $\tilde{U} = UHQ$ (Update the eigensystem)

Step 1 is an attempt to expose a simpler underlying structure of the updated matrix, which is composed of a diagonal matrix modified by a rank one matrix. The second step is commonly called deflation and is a crucial part of the procedure. We take advantage of repeated roots to decrease the computation required in the update from R to $\tilde{R}$. It is easy to see that if $\beta_i = 0$ for some i , then the i^{th} eigenvalue and associated eigenvector do not change, since

$$(D + \beta\beta^*)\mathbf{e}_i = D\mathbf{e}_i + \beta\beta_i^* = d_i\mathbf{e}_i, \quad (4.46)$$

where $\mathbf{e}_i$ represents the i^{th} elementary column vector, and d_i is the i^{th} eigenvalue corresponding to the i^{th} column of D . If there are eigenvalues that have a multiplicity of two or more, then a block diagonal Householder transformation H may be used to zero some of the β_i 's, thus allowing us to deflate the size of the problem. Suppose, for example, an eigenvalue has a multiplicity $N - r$. Then all but one of these can be zeroed out, which reduces the size of the problem to $(r + 1) \times (r + 1)$ with a corresponding drop in computation time from $\mathcal{O}(N^3)$ to $\mathcal{O}(N(r + 1)^2)$. In our situation, $r = 1$, regardless of the value of N ; thus we have a drop in computation time from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$.

Step 3 involves finding the EVD of $(D + \beta\beta^*)$. This includes two processes. First, we must find the eigenvalues of the now simplified matrix. This essentially involves an iterative solution of

a nonlinear equation using rational interpolating functions—for details see [1] and [7]. Second, the associated eigenvectors must be constructed, which is a finite computation once the eigenvalues are known. As pointed out by DeGroat [7], the columns of $Q = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_N]$ are constructed by,

$$\mathbf{q}_i = \frac{(D - d_i I)^{-1} \boldsymbol{\beta}}{\|(D - d_i I)^{-1} \boldsymbol{\beta}\|} \quad (4.47)$$

where the d_i are the eigenvalues just found. The final step is then to simply update the eigenvectors, which involves multiplying the indicated orthogonal matrices.

The only iterative part of this algorithm is the updating of the eigenvalues of $P = D + \boldsymbol{\beta} \boldsymbol{\beta}^*$. These are found indirectly from the characteristic equation of P ,

$$\begin{aligned} p(\lambda) &= \det(D + \boldsymbol{\beta} \boldsymbol{\beta}^* - \lambda I) \\ &= \det(D - \lambda I) \det(I + (D - \lambda I)^{-1} \boldsymbol{\beta} \boldsymbol{\beta}^*) \\ &= \prod_{i=1}^n (d_i - \lambda) (1 + \boldsymbol{\beta}^* (D - \lambda I)^{-1} \boldsymbol{\beta}), \end{aligned} \quad (4.48)$$

where the new eigenvalues are the zeros of

$$w(\lambda) = 1 + \boldsymbol{\beta}^* (D - \lambda I)^{-1} \boldsymbol{\beta}. \quad (4.49)$$

See DeGroat [7] for a full explanation of the iterative method used to determine the eigenvalues. His algorithm is linear, which is a simplification of Bunch's quadratic algorithm [1]. With either method, convergence is asymptotically quadratic and guaranteed. The algorithm is highly parallel and can be easily adapted for implementation in a systolic array [17]. It is well-suited to signal processing problems that involve nonstationary covariance matrices. However, as the number of updates proliferates, so does the size of the accumulated round-off error. Errors arise in both the eigenvalue and eigenvector computations. The eigenvalues become slowly perturbed, while the eigenvectors

tend to become nonorthogonal. These effects are interrelated, and can grow to unacceptable levels when there are large numbers of updates (as there will be in our scenario). DeGroat's solution is to attack both sources of error. He uses exponential weighting of the eigenvalues,

$$\begin{aligned}\hat{R}_{zz}(k+1) &= \alpha \hat{R}_{zz}(k) + (1-\alpha)z(k+1)z^*(k+1) \\ &= U(k)(\alpha D(k) + (1-\alpha)\gamma(k+1)\gamma^*(k+1))U(k),\end{aligned}\tag{4.50}$$

which causes the accumulated numerical perturbation of each eigenvalue to be bounded by a constant, so long as no large errors affect the eigenvectors.

To prevent the eigenvectors from becoming nonorthogonal, he applies a pairwise Gram-Schmidt (PGS) orthogonalization algorithm at each update. Define a measure of the nonorthogonality of the columns of U by,

$$F_U = \|UU^* - I\|_F.\tag{4.51}$$

$\|A\|_F$ signifies the Frobenius matrix norm of $A \in \Re^{m \times n}$ given by,

$$\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2.\tag{4.52}$$

DeGroat has shown that a PGS orthogonalization defined by

$$\tilde{q}_k = \frac{q_k - (q_k^* q_{k+1})q_{k+1}}{\|q_k - (q_k^* q_{k+1})q_{k+1}\|}, \quad k = 1 \dots N-1,\tag{4.53}$$

will always reduce the error measure of Eq. 4.51. In matrix form, Eq. 4.53 is given by the transformation,

$$\tilde{Q} = QB,\tag{4.54}$$

where B is a lower bidiagonal matrix. On any given update, most of the eigenvectors will be orthogonal within machine precision; thus, many of the subdiagonal elements will be zeros. This

contrasts sharply with a standard Gram-Schmidt orthogonalization in which B would represent a fully populated lower triangular matrix.

4.4 Exploiting Structure in the Covariance Matrix

Applying the results of the previous section to our problem proves quite productive. From Eqs. 4.34 through 4.37, we can see that we have *a priori* knowledge of the structure of our covariance matrix. R_j is clearly a rank one symmetric matrix; as such, it will have an EVD of the form,

$$R_j = U_j \left[\begin{array}{c|c} \lambda_j & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] U_j^* \quad (4.55)$$

where U_j is unitary. R_d is also rank one and symmetric, with an EVD of the same form. R_n has rank N , but it is a diagonal matrix; in fact, it is simply a scaled version of the identity matrix. It would have an EVD that looks like,

$$R_n = U_n(\sigma^2 I)U_n^*. \quad (4.56)$$

where σ^2 is the variance of each noise component.

Consider now the sum $R_d + R_j$. This will be a rank two matrix, with an EVD of the form,

$$R = U \left[\begin{array}{c|c} \lambda_1 & \mathbf{0} \\ \hline & \lambda_2 \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] U^*. \quad (4.57)$$

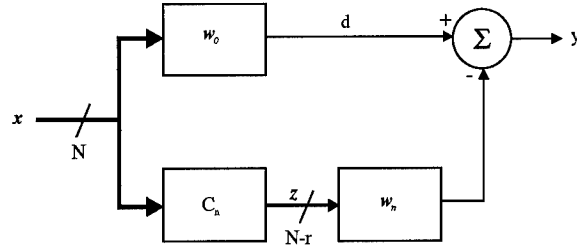


Figure 4.3. Generalized Sidelobe Canceler

Now we add the noise matrix to this sum, giving us a full rank matrix which has an EVD that looks like,

$$R_{xx} = U \begin{bmatrix} \lambda_1 + \sigma^2 & & \mathbf{0} \\ & \lambda_2 + \sigma^2 & \\ \mathbf{0} & & \sigma^2 I \end{bmatrix} U^*. \quad (4.58)$$

Recall that we are using an LCMV beamformer as described in Section 2.6.3, where the following relationships were developed (see Fig. 4.3):

$$R_{zz} = Ezz^* = C_n^* Exx^* C_n = C_n^* R_{xx} C_n \quad (4.59)$$

$$\mathbf{r}_{zd} = Ezd^* = C_n^* Exx^* \mathbf{w}_0 = C_n^* R_{xx} \mathbf{w}_0 \quad (4.60)$$

$$\mathbf{w}_n = R_{zz}^{-1} \mathbf{r}_{zd} \quad (4.61)$$

Therefore, the covariance matrix we will actually use to compute the weight vector, $\mathbf{w}_n$, is R_{zz} . R_{zz} is the covariance of the data vector, $\mathbf{z}$, that appears at the output of the blocking matrix C_n . Recall that C_n blocks any part of the received signal which lies in the signal subspace; thus R_{zz} will have the form,

$$R_{zz} = U \begin{bmatrix} \lambda_j + \sigma^2 & \mathbf{0} \\ \mathbf{0} & \sigma^2 I \end{bmatrix} U^*. \quad (4.62)$$

The structure of this matrix is critical for our application. Although it has full rank, we are aware that it is “rank-one plus noise.” Therefore, we know that the exact covariance matrix will have the

structure of Eq. 4.62. Based on our previous discussion of the rank-one EVD updating method, this is recognized as the pleasant case of an eigenvalue with a multiplicity greater than one; in fact, only two distinct eigenvalues exist — $\lambda_j + \sigma^2$ with a multiplicity of one, and σ^2 with a multiplicity of $N - 2$ (R_{zz} is an $(N - 1) \times (N - 1)$ matrix since a single constraint has been imposed upon the LCMV beamformer). Thus, deflation will be possible, with a resulting EVD update of size 2×2 , rather than $(N - 1) \times (N - 1)$. Now, our *estimated* covariance matrix will, in general, not have this structure, due to the error of the estimate. However, we can impose this structure upon it, since we know that this is in fact the structure of the true matrix. An EVD of the estimated covariance will have the structure,

$$\tilde{R}_{zz} = U \begin{bmatrix} \lambda_j + \sigma^2 & & & \mathbf{0} \\ & \sigma_1^2 & & \\ & & \ddots & \\ \mathbf{0} & & & \sigma_{N-2}^2 \end{bmatrix} U^*. \quad (4.63)$$

where λ_j would be some relatively large value due to the jamming signals, and the σ_i 's would be relatively small numbers due to the noise. Our knowledge of the structure of the exact covariance enables us to realize that all of the σ_i 's should have the same value. Clearly, then, one way to make the estimated matrix match the known structure is to average the last $N - 2$ eigenvalues, and substitute this average for each eigenvalue. Scharf [16] has shown that this approach gives the maximum likelihood estimate of the structured covariance. The maximum likelihood estimate of R_{zz} is given by,

$$\hat{R}_{zz} = U \left[\begin{array}{c|c} \lambda_j + \hat{\sigma}^2 & \mathbf{0} \\ \hline \mathbf{0} & \hat{\sigma}^2 I \end{array} \right] U^*, \quad (4.64)$$

with $\hat{\sigma}^2$ given by,

$$\hat{\sigma}^2 = \frac{1}{N - 2} \sum_{i=1}^{N-2} \sigma_i^2. \quad (4.65)$$

Let's now express Eq. 4.64 as,

$$R_{zz} = U D U^*, \quad (4.66)$$

where the definition of D is obvious by comparison of the two equations. The beauty of this expression for our covariance matrix is that its inverse is given by,

$$R_{zz}^{-1} = (U D U^*)^{-1} = U D^{-1} U^*. \quad (4.67)$$

Thus, the inversion process is simplified to finding the inverse of a diagonal matrix, which, for our matrix, is an $\mathcal{O}(1)$ process, independent of the number of elements in our array! This leads us to the final section of our present chapter, in which we consider explicitly the computation time required for the calculations involved in our system.

4.5 Computation Time Considerations

As has been shown in the last section, it is the calculation of R_{zz} and its inverse that we seek to reduce through the use of the rank-one EVD. We now know that given some initial $R_{zz} = U D U^*$, the following procedure must be performed at each sample of the incoming signal (we suppress the subscript on R_{zz} for convenience):

1. Simplify and deflate: $\tilde{R} = R + z z^* = U \tilde{H} (D + \beta \beta^*) \tilde{H}^* U^*$, $\beta = \tilde{H}^* U^* z$
2. Find the eigenvalues of $D + \beta \beta^* = Q \tilde{D} Q^*$
3. Noise average: $\hat{d}_i = \frac{1}{N-2} \sum_i d_i$, $i \in (1, N-2)$
4. Calculate the eigenvectors of $D + \beta \beta^* = Q \tilde{D} Q^*$
5. Update the eigenvectors: $\tilde{U} = U \tilde{H} Q$
6. Stabilize: $\tilde{U} := \tilde{U} B$, $D := \alpha D$
7. Compute the weight vector: $w_n = R^{-1} r = \tilde{U} D^{-1} \tilde{U}^* r$

8. Calculate the output: $y = d - w_n^* z$

The notation $\bar{H}$ has been used to indicate a block diagonal matrix which will contain as its lower right block the Householder transformation H , to be discussed in more detail momentarily. We now want to consider each of these steps, and the order of the number of operations necessary for each one. Fig. 4.4 summarizes the process we will use in our LCMV beamformer. To help simplify

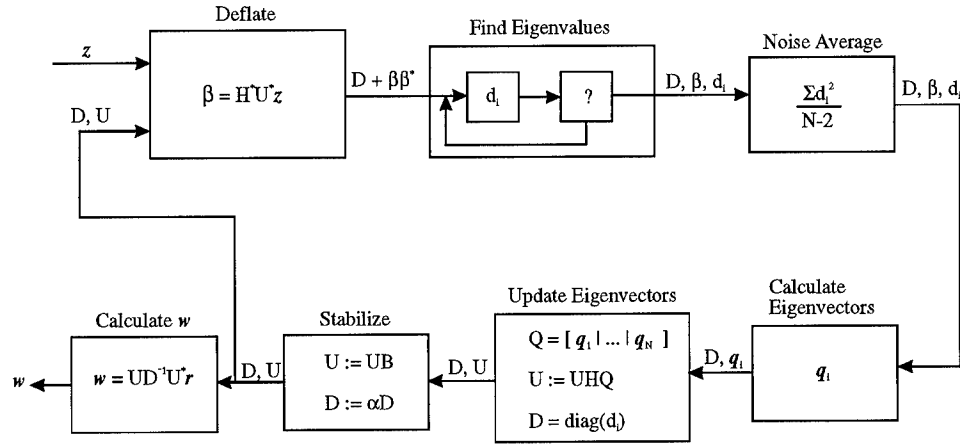


Figure 4.4. The Algorithm

our count, it is assumed that we have real, rather than complex, data. Extension to the complex case is not difficult.

Before examining the individual steps outlined above, a question must be addressed that the observant reader may have already asked. The GSC requires two transformations which are evident in Fig. 4.3 and stated here,

$$d = w_0^* x, \quad (4.68)$$

$$z = C_n^* x. \quad (4.69)$$

Certainly the computation times associated with these transformations must be addressed. Eq. 4.68 is an inner product which clearly is $\mathcal{O}(N)$. However, Eq. 4.69 involves a matrix-vector multiplication which appears to be an $\mathcal{O}(N^2)$ operation. We appeal to the known structure of our scenario in

order to reduce this to $\mathcal{O}(N)$. We have assumed that the desired signal arrives broadside to our array. The comments which follow apply specifically to this situation, but could easily be adapted for use with any DOA of the desired signal.

The blocking matrix, C_n , is constructed so that its columns are orthogonal to the null space of the constraint matrix, C , as discussed in Section 2.6.3. Since C in our case is simply the N -element vector $\mathbf{1}$, it is easy to show that the structure of C_n^* can always be made to look like the $(N-1) \times N$ matrix,

$$C_n^* = \begin{bmatrix} c & a & b & b & \cdots & b \\ c & b & a & b & \cdots & b \\ c & b & b & a & & b \\ \vdots & & & & \ddots & \vdots \\ c & b & b & b & \cdots & a \end{bmatrix}, \quad (4.70)$$

where the values of a, b , and c are easily found and not important at this point. In order to see the computation requirements for Eq. 4.69, it is helpful to rewrite this expression for C_n^* as the summation of three simpler matrices:

$$C_n^* = c \left[\mathbf{1} \mid \mathbf{0} \right] + (a-b) \left[\mathbf{0} \mid I \right] + b \left[\mathbf{0} \mid \mathbf{1}\mathbf{1}^T \right], \quad (4.71)$$

where the left side of each partitioned matrix is an $(N-1)$ -element vector, and the right side is an $(N-1) \times (N-1)$ square matrix. Now the order of computation of Eq. 4.69 is easily deduced by looking at these three separate components multiplied times the N -element vector $\mathbf{x}$. By inspection, in each case we have an $\mathcal{O}(N)$ operation. Thus, the overall order of this computation is N .

We now examine the eight steps of the algorithm discussed above. The algorithm will be shown to also have an overall computation requirement of $\mathcal{O}(N)$.

4.5.1 *Step 1—Simplify and Deflate.* This step contains two processes. First we must find the Householder transformation that will reflect our data vector appropriately. This requires a vector operation of the form,

$$H = I - \frac{2\xi\xi^*}{\xi^*\xi} \quad (4.72)$$

where ξ is an $(N-2)$ -element vector that is a function of the vector $\mathbf{z}$ (which is a function of the incoming data vector $\mathbf{x}$),

$$\xi = U_2^T \mathbf{z} - \|U_2^T \mathbf{z}\| \mathbf{e}_1. \quad (4.73)$$

U_2 is defined by partitioning the matrix U as,

$$U = [U_1|U_2], \quad (4.74)$$

where U is an $(N-1) \times (N-1)$ matrix, and U_2 represents the final $(N-2)$ columns of U . As before noted, $\mathbf{e}_1$ is the elementary column vector, $[1 \ 0 \ \cdots \ 0]^T$. This is clearly an $\mathcal{O}(N^2)$ process due to the outer product in Eq. 4.72. However, we only need to find the first two elements of the vector β since the other elements are being zeroed out by the Householder transformation. Thus we have,

$$\beta = \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & H \end{array} \right]^T \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ H^T \gamma_2 \end{bmatrix} = [\gamma_1 \quad -\|\gamma_2\| \quad 0 \quad \cdots \ 0]^T, \quad (4.75)$$

where we have partitioned $\gamma = U^T \mathbf{z}$ as $[\gamma_1 \ \gamma_2]^T$. Since H represents an elementary reflector, we know that the second element of β will be (the negative of) the norm of the vector γ_2 . The first element may be found by,

$$\gamma_1 = U_1^T \mathbf{z}, \quad (4.76)$$

where U_1 is the first column of U , as in Eq. 4.74. Clearly this is an $\mathcal{O}(N)$ operation. The second element is found most easily by realizing that,

$$\|\gamma\|^2 = \|U^T \mathbf{z}\|^2 = \|\mathbf{z}\|^2 = \gamma_1^2 + \|\gamma_2\|^2, \quad (4.77)$$

or, rearranging,

$$\|\gamma_2\| = (\|\mathbf{z}\|^2 - \gamma_1^2)^{\frac{1}{2}}. \quad (4.78)$$

This is also an $\mathcal{O}(N)$ process. The deflation which this step introduces is critical because it reduces the computation requirements of some of the remaining steps from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$.

4.5.2 Step 2—Find the Eigenvalues. In this step we iteratively find the eigenvalues. This step has been greatly simplified in terms of computation time since the eigenstructure update has now been deflated to a 2×2 problem. See [7] for the implementation details.

4.5.3 Step 3—Perform Noise Averaging. This step represents an important part of our algorithm because it is what initially forces the covariance matrix to have repeated eigenvalues. These repeated eigenvalues are the key to the deflation of the problem, with the resulting reduction in computation time from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$. This step is $\mathcal{O}(N)$ since we simply add together the $N - 2$ eigenvalues and divide by $N - 2$.

4.5.4 Step 4—Calculate the Eigenvectors. This step is easy once we have performed the previous two steps. Eq. 4.47 shows the process required. It is evidently an $\mathcal{O}(1)$ process, since only the first two elements of β will change with each update, with a consequent change to only two eigenvectors. Again the deflation of Step 1 has made this step a simple process. See DeGroat [7] for details.

4.5.5 Step 5—Update the Eigenvectors. Updating the eigenvectors now requires the multiplication of unitary matrices, $\tilde{U} = UHQ$. Due to the deflation in Step 1, the block nature of the three matrices, and the large numbers of zeros and block identity matrices, this multiplication results in an overall computation process which is $\mathcal{O}(N)$. See Appendix A for details.

4.5.6 Step 6—Stabilize. This step involves the lower bidiagonal matrix multiplication discussed earlier, and is represented by the process shown in Eqs. 4.53 and 4.54. This would be an expensive process because of the inner products, but deflation has made the size of the vectors a constant size independent of N ; furthermore, the necessary calculations have already been performed in the iterative procedure to find the eigenvalues [7]. If the orthogonalization is performed on all adjacent pairs of eigenvectors, each element of U must be written; this implies an $\mathcal{O}(N^2)$ operation. To reduce this to $\mathcal{O}(N)$, we will orthogonalize only one pair of eigenvectors per update. Thus, on the first update the first two eigenvectors will be orthogonalized; on the second update, the second and third eigenvectors; and so on, until the last pair have been orthogonalized. At this point we will start over with the first and second eigenvectors, and continue in this fashion. The overall order of this step will thus be reduced to $\mathcal{O}(N)$.

4.5.7 Step 7—Compute the Weight Vector. The computation of the weight vector is given by,

$$\mathbf{w}_n = (UDU^*)^{-1} \mathbf{r}_{zd} = UD^{-1}U^* \mathbf{r}_{zd}. \quad (4.79)$$

This can be expressed in block matrix form as,

$$\mathbf{w}_n = \left[\begin{array}{c|c} U_1 & U_2 \end{array} \right] \left[\begin{array}{c|c} d_1^{-1} & \\ \hline & d_2^{-1}I \end{array} \right] \left[\begin{array}{c} U_1^* \\ U_2^* \end{array} \right] \mathbf{r}_{zd}, \quad (4.80)$$

where we have used the partitioning of U as before, and have taken advantage of the known structure of D . Now, multiplying the block structure through results in the following,

$$\mathbf{w}_n = \left[\begin{array}{c|c} d_1^{-1}U_1 & d_2^{-1}U_2 \end{array} \right] \left[\begin{array}{c} U_1^* \\ U_2^* \end{array} \right] \mathbf{r}_{zd} \quad (4.81)$$

$$= (d_1^{-1}U_1U_1^* + d_2^{-1}U_2U_2^*)\mathbf{r}_{zd} \quad (4.82)$$

$$= [d_1^{-1}U_1U_1^* + d_2^{-1}(I - U_1U_1^*)]\mathbf{r}_{zd} \quad (4.83)$$

$$= [(d_1^{-1} - d_2^{-1})U_1U_1^* + d_2^{-1}I]\mathbf{r}_{zd} \quad (4.84)$$

where we have used the identity $U_1U_1^* + U_2U_2^* = I$ for a partitioned unitary matrix. It is evident from Eq. 4.84 that the calculation of the weight vector is an $\mathcal{O}(N)$ process.

4.5.8 Step 8—Calculate the Output. The output is then computed as, $y = d - \mathbf{w}_n^* \mathbf{z}$, which is clearly an $\mathcal{O}(N)$ process.

In conclusion, it has been shown that by applying rank-one updating of the EVD of the covariance matrix, the computation requirements for updating the weight vector are significantly reduced. This makes a formerly intractable problem quite workable. Since the overall computation for our system is now an $\mathcal{O}(N)$ process, our algorithm should be capable of implementation in a practical system operating in real-time. The next chapter demonstrates that when the weight vector is updated every sample, the actual SINR at the beamformer output can be maintained close to the optimum possible SINR (which would be obtained if we had access to the exact covariance).

V. Simulations

5.1 Proof-of-Concept

In order to demonstrate the feasibility of successfully utilizing an N -element beamformer in an environment with more than $N - 1$ interfering signals, we start our simulations by assuming that the time required to calculate the inverse of the covariance matrix is not a problem. In fact it is *not* a problem when one is using Matlab to run a simulation and not operating in real-time. Furthermore, as shown in this thesis, the computation required to update the covariance matrix at each sample can be reduced to an $\mathcal{O}(N)$ process which may be realized in real-time. We begin by considering the simple case of a two-element array used in an environment where there are three interfering signals: the direct jamming signal, and two multipath copies thereof. This is the case Compton [5] examined, but he used only the exact covariance. The sample matrix inverse (SMI) method is used and updates of the covariance matrix, $\hat{R}_{xx}$, are performed every sample. The equations developed in Section 2.6.3 are then used to determine the weight vector that will minimize output power under the constraint that the desired signal pass with unity gain. Using this weight vector, then, the various components of the signal at the output of the beamformer are determined, and the signal-to-interference-plus-noise ratio (SINR) is calculated and plotted. We also compute the exact covariance matrix and its inverse so that we may plot the optimum SINR on the same graph for comparison with the actual SINR.

We must determine the number of samples to use in the calculation of the estimated covariance matrix. According to a well-known theorem of Reed *et al.* [15], at least $2N - 3$ samples are needed in the estimate in order to obtain a SINR within 3 dB of optimal; N represents the number of elements in the array. Since we are currently using a two-element array, this implies that only one sample is needed; however, two samples are used to avoid the problem of a singular covariance matrix.

Based upon work that Compton did with the two-element array [4], the following equation is used to approximate the maximum amount of Doppler spread that can be tolerated, and yet remain within 3 dB of the optimum SINR,

$$f_{dop} < \frac{2.3 * 10^{-3}}{T_s}. \quad (5.1)$$

We use a sampling frequency of 1 MHz; therefore the maximum Doppler spread we can expect to tolerate is,

$$f_{dop} < \frac{2.3 * 10^{-3}}{T_s} = \frac{2.3 * 10^{-3}}{1 * 10^{-6}} = 2300\text{Hz}. \quad (5.2)$$

This suggests that we examine the case with $f_{dop} \approx 2300$ Hz as a ‘typical’ case. Two other cases are also considered: one with twice as much Doppler spread (which we call the “worst” case) and another with half the Doppler spread (the “best” case). Thus, one simulation will be run with $f_{dop} \approx 1150$ Hz, and another with $f_{dop} \approx 4600$ Hz.

The Doppler spread is the difference in frequency between the extremes of the jammer and multipath signals. For example, if the Doppler shifts on the two multipath signals are both in the same direction (i.e. both positive or both negative), then the Doppler spread is simply given by the magnitude of the larger shift. If one Doppler shift is positive and the other negative, then the spread is given by the magnitude of the difference between the two shifts. It is the amount of spread itself, and not the signs of the shifts, that determines the effects of nonstationarity. Therefore, we consider only positive shifts, and use the following cases in our simulations:

	Worst	Typical	Best	
$f_2 :$	2004	1004	504	(5.3)
$f_3 :$	4604	2304	1154	

<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>
Signal-to-noise ratio	SNR	0 dB
Jammer-to-noise ratio	JNR	40 dB per jammer
Sampling frequency	f_s	1 MHz
# Elements	N	varies
# Jammers	P	varies
# Samples used in covariance estimate	m	varies
DOA of desired signal	θ_d	0 deg (broadside)
DOA of jammer signals	θ_j	varies (-75 to 75 deg)
Doppler shift of jammer signals	ψ_j	varies
Phase shift of multipath signals	ϕ_j	0 deg
Forgetting factor	α	varies (0 to 1)

Table 5.1. Variables Used in the Simulations

where f_2 and f_3 are the Doppler shifts on j_2 and j_3 respectively, and all values are in Hz. Other variables are involved in the simulations that deserve some explanation. We assume that the desired signal has the same relative strength as the noise signals, which is another way of saying that the signal-to-(thermal) noise ratio (SNR) is 0 dB. Although our development allows for the case of directional elements (by assigning various power levels depending upon the signals' direction of arrival), we have chosen to make each of the jamming signals contain 40 dB more power than the desired signal. Which is to say that the jammer-to-noise ratio (JNR) is 40 dB per jammer signal. The sampling frequency is assumed to be 1 MHz throughout. The number of samples used to compute the estimated covariance matrix varies with the parameters in each respective simulation, and may be seen listed above each graph, together with other pertinent information. In summary then, Table 5.1 contains the assumptions made.

Fig. 5.1 shows the amount of degradation in the output SINR in the best case, as a function of the number of samples included in the covariance matrix estimate. This shows graphically that there is some optimal length to use in the calculation of the covariance matrix, which, for a given sampling frequency, is dependent upon the Doppler spread. We used the optimum values found in all three cases to run the simulations, with the results shown in Figs. 5.2 through 5.4. Fig. 5.2 shows our worst case scenario; in this case the difference between the average optimal SINR and the average actual SINR is 4.9 dB. Note that the optimal SINR is plotted as a dashed line, while

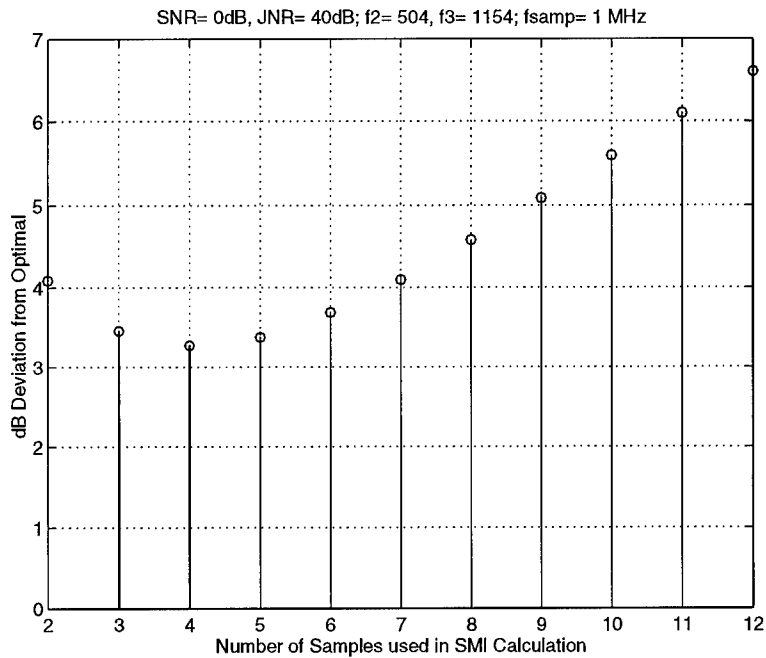


Figure 5.1. Deviation from Optimal for the Best Case

the actual SINR is plotted as a solid line. To generate the plot of the actual SINR, we ran the simulation for 100 different realizations of the data matrix, and then averaged over the ensemble for each sample.

Fig. 5.3 shows the typical case, where the difference between actual and optimal SINR averages is 4.2 dB. We used a sample length of three for this case. Finally, Fig. 5.4 shows the best case where the average difference is 3.4 dB. Here a sample length of four produced the best results. In every case the actual SINR clearly tracks the optimal case, i.e. the case in which the exact covariance matrix is used rather than an estimated covariance matrix. Furthermore, as the Doppler spread decreases, the more samples that can be used in the estimate, and the closer the actual SINR is to the optimal. This makes sense intuitively, since one would expect that a large number of samples would produce a better estimate than a small number of samples. The problem, as discussed earlier, is that the nonstationarity of the received signal limits the number of samples which can reasonably

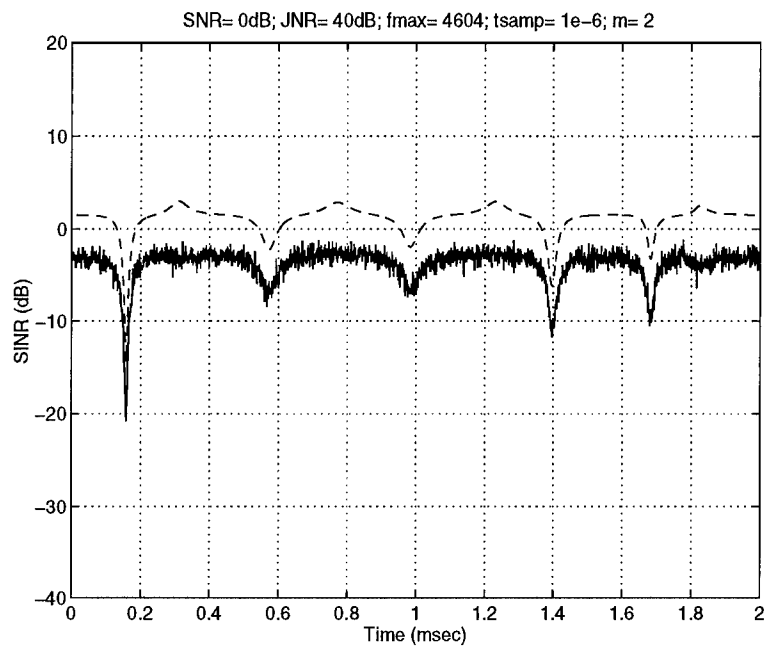


Figure 5.2. Optimal vs. Actual for the Worst Case

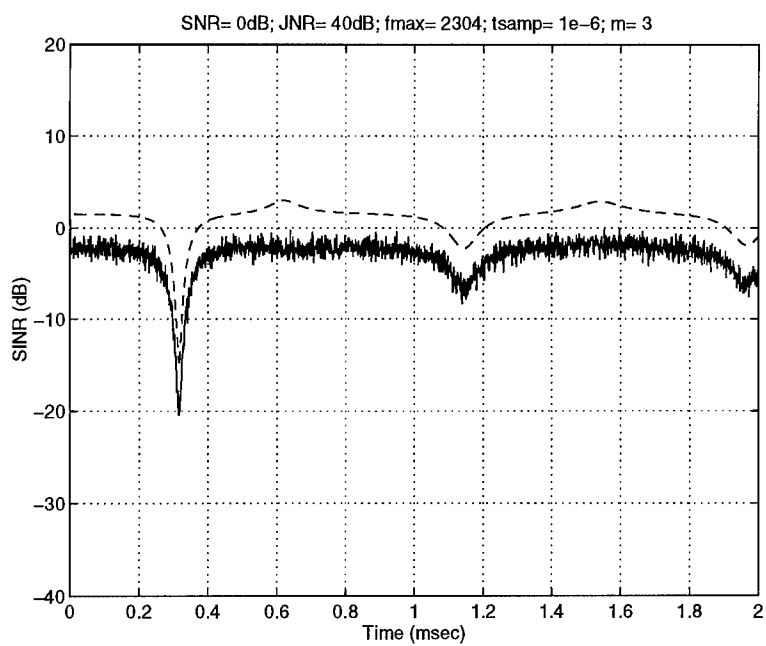


Figure 5.3. Optimal vs. Actual for the Typical Case

be used in the estimate. As the Doppler spread decreases, so does the degree of nonstationarity, and we can use more samples, with an improved SINR as a result.

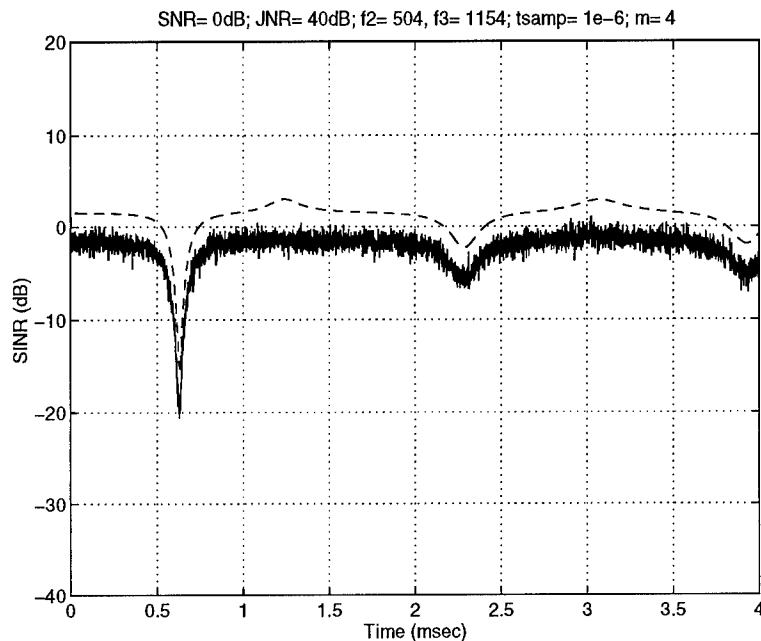


Figure 5.4. Optimal vs. Actual for the Best Case

We ran a simulation using the extreme values of $f_2 = 7004$ Hz, $f_3 = 18004$ Hz, in which we used a sample length of two. The results shown in Fig. 5.5 indicate an average difference in SINRs of about 8.0 dB. We ran this case again, using a forgetting factor of $\alpha = 0.1$, with the result in Fig. 5.6 showing a much improved average difference of about 5.0 dB. We cannot expect much better than this since we essentially have access to only one sample in determining the estimate of the covariance matrix.

5.2 The General Case

These results indicate that the concept of canceling more interferers than there are degrees of freedom is certainly possible. In order to demonstrate the feasibility in a somewhat more realistic environment, simulations were run using both more interferers and more array elements. Apart

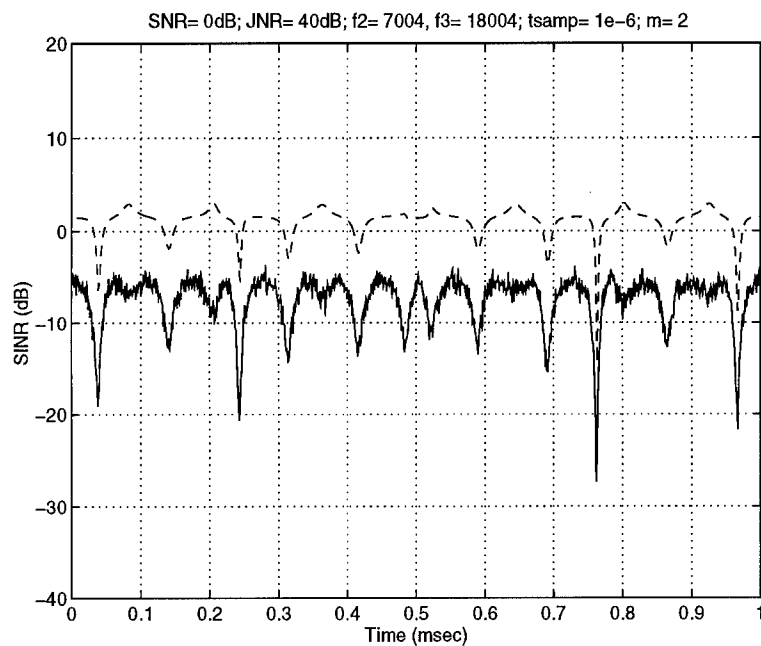


Figure 5.5. Optimal vs. Actual for the Extreme Case

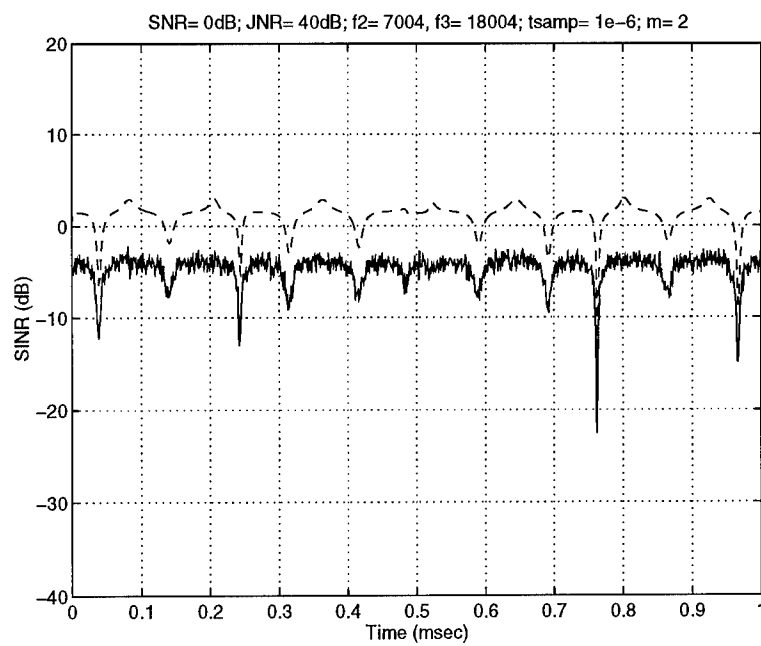


Figure 5.6. Optimal vs. Actual for the Extreme Case with Forgetting Factor

from explicit changes mentioned, we continue to use the parameters summarized in Table 5.1. We begin by considering again a two-element array. However, now there are six jamming signals, composed of the direct jammer and five multipath copies. The Doppler spread is 2045 Hz, and the generalized sidelobe canceler is being used with updates to w_n every sample. The result is shown in Fig. 5.7, where again both the optimal and actual SINR are plotted. Note that there are

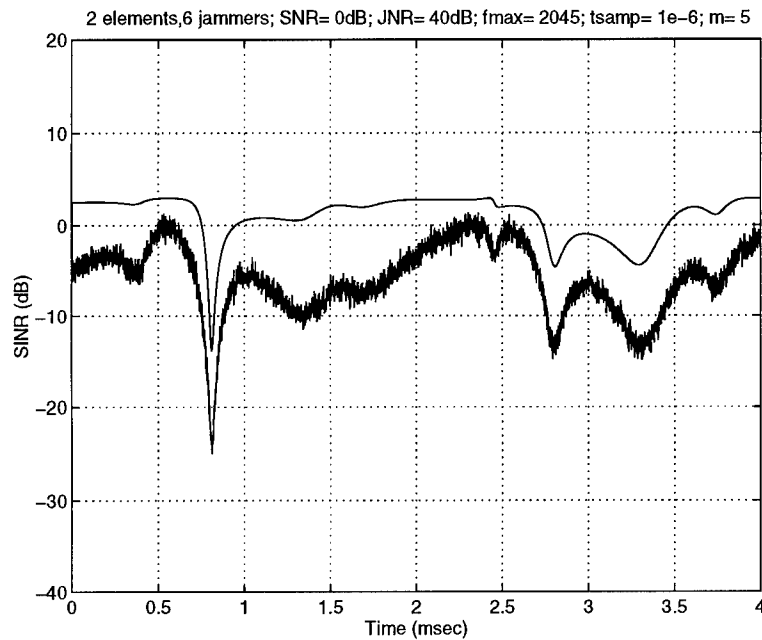


Figure 5.7. Two Elements–Six Interferers

points at which even the optimal SINR dips very low. This is due to the following phenomena [5]. The received interfering signals cycle through phase changes at different rates. As they do so, it happens at certain times that the jammer signals add coherently in a manner that makes them almost appear as if they were a single jammer signal arriving broadside to the array. Clearly, at these instants the beamformer cannot cancel the jammers since our imposed constraint is to pass all signals arriving broadside to the array with unity gain. We must simply live with this phenomena. The difference between actual and optimal SINR in this example is 6.9 dB.

Now the number of elements in the array is increased from two to five, while all other parameters are kept the same. The result is shown in Fig. 5.8. Note the improvement in the actual SINR,

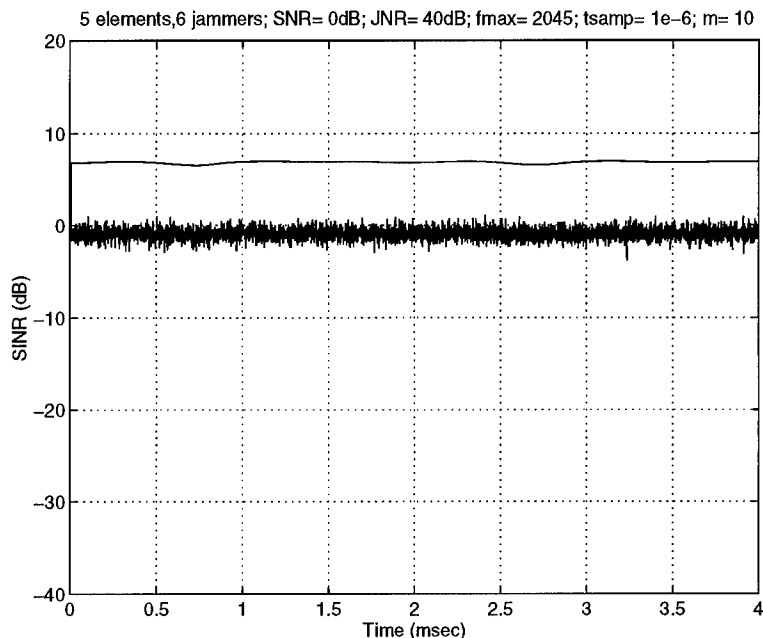


Figure 5.8. Five Elements–Six Interferers

which would be expected with more elements. The dB difference is now about 4.1 dB. We have also plotted the beam pattern for this example at several successive sample points (see Fig. 5.9). Interestingly, it is not a “good” pattern. If observed over long periods of time it changes considerably, the only constant being that it always passes the desired signal with unity gain. The beam pattern is not what has been optimized; rather, the pattern changes as necessary in order to reduce the total output power, with the only constraint being that the desired direction is unaffected. In terms of reducing the amount of interference that appears at the output, the beamformer appears to be doing a respectable job.

Now we use the same beamformer, but increase the number of interfering signals from six to 18 (Fig. 5.10). The dB difference with 18 interferers is still only 4.3 dB. This is somewhat surprising, in light of the large number of interferers, particularly when the total received jammer

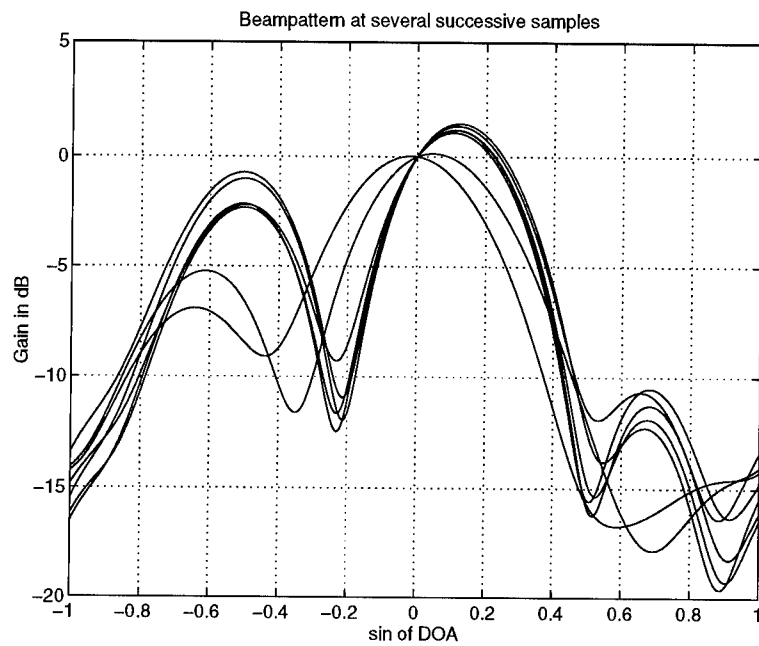


Figure 5.9. Beampattern for Several Successive Samples

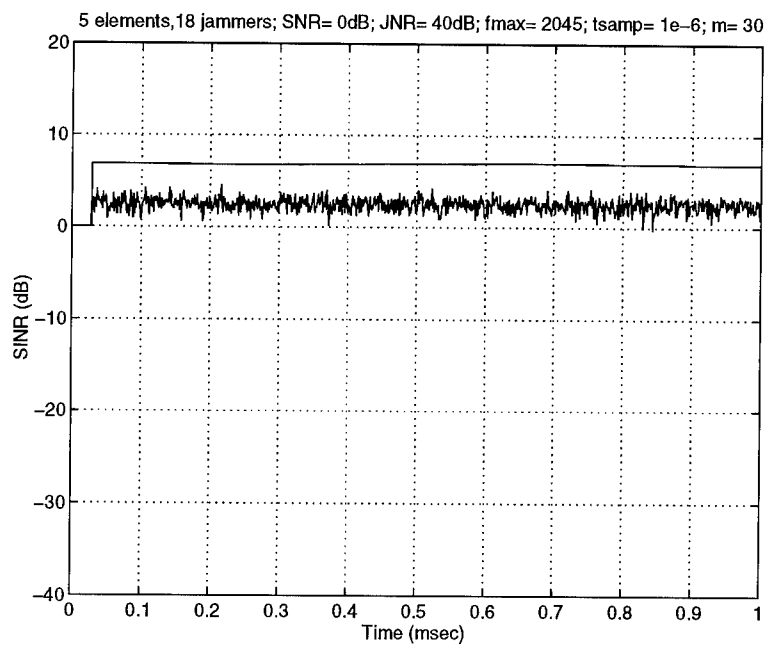


Figure 5.10. Five Elements-18 Interferers

power is considered. The reader may wonder why the dB difference in this case is better than in the previous example, since the only change was to go from six to 18 interferers. Two factors come into play. First, and most important, the sample length used in Fig. 5.8 was only 10, while in Fig. 5.10 the sample length was changed to 30. The improved estimate of the covariance matrix which results from the larger sample length is significant. Another factor is that the Doppler spread remained the same in both examples. Thus, although there is considerably more jammer power impinging upon the array in the latter simulation, there is the degree of nonstationarity involved.

We again show several successive beampatterns in Fig. 5.11. These look somewhat better, but are clearly not good beampatterns in the normal sense. However, we sacrifice the beampattern for the ability to operate in the presence of a large number of interfering signals with relatively few array elements, and maintain reasonable SINRs at the output of the beamformer. Furthermore, the beampattern could be improved if directional elements were used in the array; omnidirectional elements are being used in these simulations.

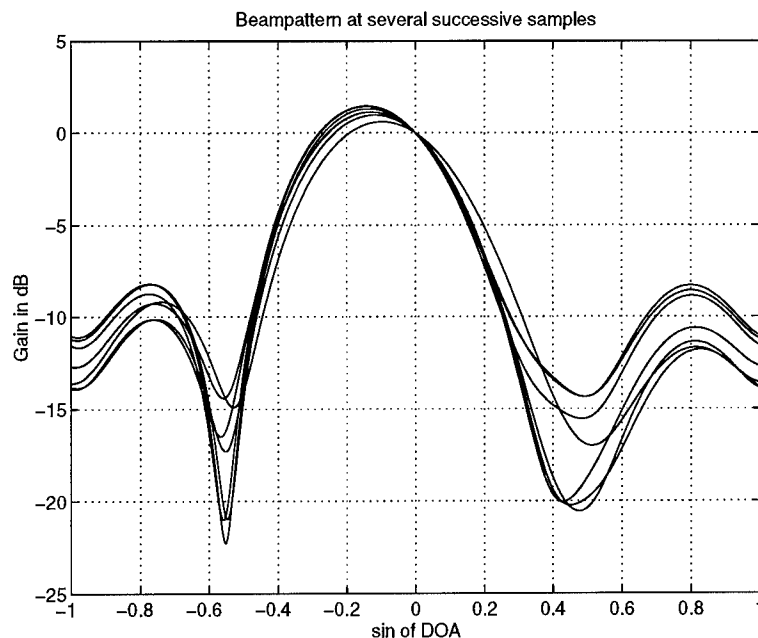


Figure 5.11. Beampattern of Successive Samples

This chapter has shown that SINRs reasonably close to optimal can be obtained with adaptive arrays which update their covariance estimate every sample. Coupled with the algorithm developed earlier which enables the real-time operation of such beamformers, it seems that the applicability of rank-one EVD to the frequency-dispersive multipath environment has been well-established.

VI. *Conclusions and Recommendations*

We have shown that it is possible to operate an adaptive array in a frequency-dispersive multipath environment in which there are more interferers than there are degrees of freedom in the array. We have outlined a method whereby the computation order is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$, where N represents the number of elements in the array.

Compton's [4] frequency-shifted weights provide a means of extending the length of time that a calculated weight vector may be used effectively; we have taken a separate path to arrive at a similar end. We used no frequency-shifted weights, but rather sought to show that it is possible to update the calculated weight vector often enough (every sample) to operate in a nonstationary environment in real-time.

The rank-one eigenvalue decomposition (EVD) played a vital role in the development of an algorithm to facilitate the necessary rapid update of the covariance matrix. The concept of noise averaging enabled us to greatly deflate the EVD problem, thus reducing the computation requirements drastically.

Clearly we made some simplifying assumptions which leave open the way for more work in this vital area. The broadband case should be considered, with taps being used in the beamformer model. The case in which there is time-dispersion as well as frequency-dispersion in the jammer signals should be examined. The algorithm itself could be a subject of closer scrutiny, and the further savings in time brought about through parallel processing could be explored. Future work should use the algorithm developed in this thesis, and apply it to actual simulations. Processing time could then be closely analyzed, and sequential operation versus parallel operation could be examined.

The environment should be expanded to consider the case of multiple jamming sources with the attendant increase in multipath signals. We have assumed throughout that the jammer signals are perfectly correlated. This is an area that could be expanded to include the more general case of

uncorrelated or partially correlated interfering signals. Investigation would also be warranted into the manner in which this technology could be adapted into present-day systems.

In any case, we are confident that the results of this paper can be applied in the ongoing effort to provide a more effective anti-jamming capability to the air combat warfighter, as well as applied to other areas of interest in array signal processing.

Appendix A. Block Structure of Eigenvector Update

The procedure required for most of the algorithm outlined in Fig. 4.4 is sketched here. The block structure makes evident the simplification of the problem as discussed in Section 4.5. We start with the Hermitian (symmetric) covariance matrix,

$$R = UDU^*. \quad (\text{A.1})$$

The incoming data vector $\mathbf{x}$ results in a GSC vector $\mathbf{z}$ which in turn produces an updated covariance,

$$\tilde{R} = R + \mathbf{z}\mathbf{z}^*. \quad (\text{A.2})$$

This may be written as,

$$\tilde{R} = U(D + \gamma\gamma^*)U^* = U\bar{H}(D + \beta\beta^*)\bar{H}^*U^* = \tilde{U}\tilde{D}\tilde{U}^*, \quad (\text{A.3})$$

where the following definitions apply,

$$\gamma = U^*\mathbf{z} \quad (\text{A.4})$$

$$\beta = \bar{H}^*\gamma \quad (\text{A.5})$$

As discussed in Section 4.5, the vector β is composed of only two nonzero elements. Therefore the problem of updating the eigenstructure of $\tilde{R}$ reduces to a 2×2 problem. The eigenvalue decomposition (EVD) of $D + \beta\beta^* = Q\tilde{D}Q^*$ thus reduces to the problem of finding the EVD of $D_1 + \beta_1\beta_1^* = Q_1\tilde{D}_1Q_1^*$ with the resulting block matrix representation:

$$R + \mathbf{z}\mathbf{z}^* = \left[\begin{array}{c|c} U_1 & U_2 \end{array} \right] \left(\left[\begin{array}{c|c} I & \mathbf{0} \\ \hline \mathbf{0} & H \end{array} \right] \left[\begin{array}{c|c} D_1 + \beta_1\beta_1^* & \mathbf{0} \\ \hline \mathbf{0} & D_2 \end{array} \right] \left[\begin{array}{c|c} I & \mathbf{0} \\ \hline \mathbf{0} & H^* \end{array} \right] \right) \left[\begin{array}{c} U_1^* \\ \hline U_2^* \end{array} \right]. \quad (\text{A.6})$$

Consider the following development of the above block representation (the U matrix is suppressed for convenience, the point being made without it):

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & H_{N-2} \end{array} \right] \left(\left[\begin{array}{c|c} D_1 & \mathbf{0} \\ \hline \mathbf{0} & dI_{N-2} \end{array} \right] + \beta\beta^* \right) \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & H_{N-2}^* \end{array} \right] \quad (\text{A.7})$$

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & H_{N-2} \end{array} \right] \left(\left[\begin{array}{c|c} Q_2 & \mathbf{0} \\ \hline \mathbf{0} & I_{N-3} \end{array} \right] \left[\begin{array}{c|c} \tilde{D}_2 & \mathbf{0} \\ \hline \mathbf{0} & dI_{N-3} \end{array} \right] \left[\begin{array}{c|c} Q_2^* & \mathbf{0} \\ \hline \mathbf{0} & I_{N-3} \end{array} \right] \right) \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & H_{N-2}^* \end{array} \right] \quad (\text{A.8})$$

$$\left[\begin{array}{cc|c} I_1 & 0 & \mathbf{0} \\ 0 & h & \mathbf{h}^T \\ \hline \mathbf{0} & \mathbf{h} & H_{N-3} \end{array} \right] \left(\left[\begin{array}{c|c} Q_2 & \mathbf{0} \\ \hline \mathbf{0} & I_{N-3} \end{array} \right] \left[\begin{array}{c|c} \tilde{D}_2 & \mathbf{0} \\ \hline \mathbf{0} & dI_{N-3} \end{array} \right] \left[\begin{array}{c|c} Q_2^* & \mathbf{0} \\ \hline \mathbf{0} & I_{N-3} \end{array} \right] \right) \left[\begin{array}{cc|c} I_1 & 0 & \mathbf{0} \\ 0 & h & \mathbf{h}^T \\ \hline \mathbf{0} & \mathbf{h} & H_{N-3}^* \end{array} \right] \quad (\text{A.9})$$

The subscript numbers signify the size of the square block partition in every case. Note that I_1 is just the scalar 1. The boldface $\mathbf{h}$ represents a column matrix made up of the obvious members of H . The scalar h represents the obvious member of the H matrix. The point to be made by this representation is that the order of the multiplications is greatly reduced due to deflation. Only one element of the Householder matrix becomes involved in multiplications that include something other than $\mathbf{0}$ or the scaled identity matrix. Furthermore, the EVD that occurs is only a 2×2 operation regardless of N . DeGroat [7] observes that when an eigenvalue of multiplicity $N - r$ is present, the total computational costs are reduced to $\mathcal{O}(N(r + 1)^2)$. In our scenario, $r = 1$; therefore, the order of computation is a constant times $\mathcal{O}(N)$, as desired.

Bibliography

1. Bunch, James R, Christopher P. Nielsen, and Danny C. Sorensen. "Rank-One Modification of the Symmetric Eigenproblem," *Numerische Mathematik*, Vol. 31, pp.31-48, 1978.
2. Clark, Michael P. and Richard A. Roberts. "Real-Time Adaptive Beamforming using Rank-One Eigenstructure Updating," Unpublished.
3. Compton, R. T., Jr. *Adaptive Antennas: Concepts and Performance*. New Jersey: Prentice-Hall, Inc., 1988.
4. Compton, R. T., Jr., "The Effects of Multipath Jamming on an Adaptive Array." Talk given to Wright Laboratories Radar Branch. Wright-Patterson Air Force Base Ohio, 12 July 1994.
5. Compton, R. T., Jr. *Unpublished Paper*.
6. DeGroat, Ronald D. and Richard A. Roberts. "A Family of Rank-One Eigenstructure Updating Methods," *Asilomar Conference on Signals, Systems, and Computers*, ACSSC-1/88/0564:564, 1988.
7. DeGroat, Ronald D. and Richard A. Roberts. "Efficient, Numerically Stabilized Rank-One Eigenstructure Updating," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-38, pp. 301-316, February 1990.
8. Golub, Gene H. "Some Modified Matrix Eigenvalue Problems," *SIAM Review*, Vol. 15, pp. 318-334, 1973.
9. Golub, Gene H. and C. F. Van Loan, *Matrix Computations*. Maryland: The Johns Hopkins University Press, 1989.
10. Johnson, Don H. and Dan E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. New Jersey: P T R Prentice-Hall, Inc., 1993.
11. Noble, Ben and James W. Daniel, *Applied Linear Algebra* (2nd edition). New Jersey: Prentice-Hall, Inc., 1977.
12. Karasalo, Ilkka. "Estimating the Covariance Matrix by Signal Subspace Averaging," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 8-12, February 1986.
13. Oppenheim, Alan V. and Ronald W. Schaffer, *Discrete-Time Signal Processing*. New Jersey: Prentice-Hall, Inc., 1989.
14. Papoulis, Athanasios, *Probability, Random Variables, and Stochastic Processes* (Third Edition). New York: McGraw-Hill, Inc., 1991.
15. Reed, Irving S., John D. Mallett, and Lawrence E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *IEEE Transactions on Aerospace and Electronic Systems*, AES-10(6):853, November 1974.
16. Scharf, Louis *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Massachusetts: Addison-Wesley Publishing Company, Inc., 1991.
17. Schreiber, Robert. "Implementation of Adaptive Array Algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp.1038-1045, October 1986.
18. Therrien, Charles W. *Discrete Random Signals and Statistical Signal Processing*. New Jersey: Prentice-Hall, Inc., 1992.
19. Van Veen, Barry D. "Minimum Variance Beamforming" in *Adaptive Radar Detection and Estimation*, Simon Haykin, ed. New York: John Wiley & Sons, 1992.
20. Van Veen, Barry D. and Kevin M. Buckley. "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE Acoustics, Speech, and Signal Processing Magazine*, pp. 4-24, April 1988.

Vita

Captain David Murray was born on 28 October 1958 in Cumberland, Maryland. He graduated from Keyser High School in Keyser, West Virginia in 1976 and attended Appalachian Bible College in Bradley, West Virginia, graduating with a Bachelor of Arts degree in the Pastoral-Greek studies program in May 1980. He was commissioned as a Second Lieutenant in the U.S. Air Force after finishing Officer Training School in April 1983. His first assignment was to attend Auburn University as part of the Undergraduate Engineer Conversion Program, during which he earned the degree Bachelor of Electrical Engineering with High Honor in May 1985. He then spent five years at the AFCC Systems Evaluation School at Wright-Patterson AFB, Ohio as the course director of a highly-acclaimed communications systems class. Following this, he worked for three years in headquarters Air Combat Command communications-computer directorate at Langley AFB, Virginia as a staff officer. In 1993, he was assigned to the Air Force Institute of Technology to earn his Master of Science Degree in Electrical Engineering.

Permanent address: 3713 Fincastle Drive
Beavercreek, Ohio 45431

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE ADAPTIVE BEAMFORMING IN FREQUENCY-DISPERSIVE MULTI-PATH ENVIRONMENTS		5. FUNDING NUMBERS		
6. AUTHOR(S) David A. Murray, Capt, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/94D-22		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Emil Martinsek, WL/AARM-2, Wright-Patterson AFB OH 45433		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) It is well known that an antenna array with N degrees of freedom (DOF) can cancel $N - 1$ interferers if they approach the array from directions other than that of the desired signal. It has been shown that there are cases in which it is possible for an array with N DOF to effectively cancel <i>more</i> than $N - 1$ interferers. Specifically, we show how this can be done when there is incident upon an N -element array not only the desired signal and a jammer signal, but more than $N - 1$ Doppler-shifted multipath copies of the jammer signal as well. In this situation the received signal is nonstationary; therefore, updates of the weight vector must use only as many samples as correspond to the time interval over which the signal can be considered locally stationary. In this frequency-dispersive environment, it is desirable to recalculate the weight vector often; in some cases, we would like to do so every sample. However, recalculating the weight vector involves estimating and inverting the covariance matrix of the received signal; using traditional methods, this is an $\mathcal{O}(N^3)$ process, where N represents the degrees of freedom in the array. For large N this becomes time-prohibitive. In this paper we examine the application of eigenvalue decomposition and rank-one updating of the covariance matrix to reduce the computation required. By exploiting the structure of the matrix we find that we are able to decrease the time required for its estimation and inversion from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$. Using these techniques, we can update the weight vector every sample, enabling us to operate effectively in a frequency-dispersive environment.				
14. SUBJECT TERMS Adaptive Beamforming, Eigenvalue Decomposition, Rank-One Updating, Linearly Constrained Minimum Variance Beamformer, Generalized Sidelobe Canceler, Structured Covariance Matrix		15. NUMBER OF PAGES 69		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	