NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DOA ESTIMATION BY EIGENDECOMPOSITION
USING SINGLE VECTOR
LANCZOS ALGORITHM

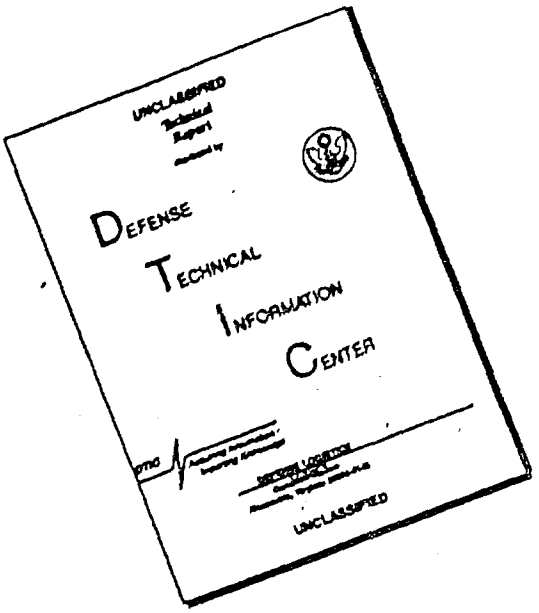
by

Daniel E. Gear

June 1989

Thesis Advisor Murali Tummala

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified			1b Restrictive Markings
2a Security Classification Authority			3 Distribution Availability of Report: Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule			
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)
6a Name of Performing Organization: Naval Postgraduate School	6b Office Symbol (if applicable): 62		7a Name of Monitoring Organization: Naval Postgraduate School
6c Address (city, state, and ZIP code): Monterey, CA 93943-5000			7b Address (city, state, and ZIP code): Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization	8b Office Symbol (if applicable)		9 Procurement Instrument Identification Number
8c Address (city, state, and ZIP code)			10 Source of Funding Numbers: Program Element No | Project No | Task No | Work Unit Accession No

11 Title (include security classification): DOA ESTIMATION BY EIGENDECOMPOSITION USING SINGLE VECTOR LANCZOS ALGORITHM

12 Personal Author(s): Daniel F. Gear

13a Type of Report	13b Time Covered	14 Date of Report (year, month, day)	15 Page Count
Master's Thesis	From To	June 1989	79

16 Supplementary Notation: 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.

17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)
Field	Group	Subgroup	direction of arrival, Lanczos, eigendecomposition.

19 Abstract (continue on reverse if necessary and identify by block number)

Subspace methods of solving spectral estimation and direction of arrival (DOA) problems involve finding the eigenvalues and eigenvectors of correlation matrices. Using the Lanczos algorithm some of the extreme eigenvalues and eigenvectors can be approximated without requiring the entire matrix decomposition theoretically saving many computations.

This thesis develops a model and a form of the Lanczos algorithm to solve the DOA problem. The relationship of the number of eigenvectors used to the accuracy of the results in a low signal to noise ratio example are examined.

20 Distribution Availability of Abstract	21 Abstract Security Classification
X unclassified/unlimited ☐ same as report ☐ DTIC users	Unclassified
22a Name of Responsible Individual: Murali Tummala	22b Telephone (include Area code): (408) 646-2645 | 22c Office Symbol: 62Tu

DD FORM 1473,84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

DOA ESTIMATION BY EIGENDECOMPOSITION
USING SINGLE VECTOR
LANCZOS ALGORITHM

by

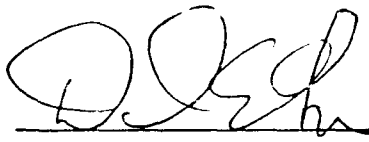
Daniel E. Gear
Lieutenant Commander, United States Navy
A.B., University of North Carolina, 1977

Submitted in partial fulfillment of the
requirements for the degree of

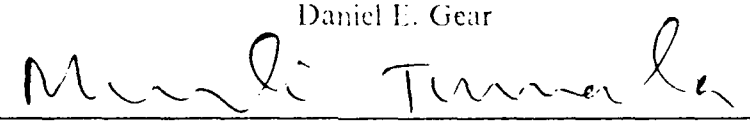
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

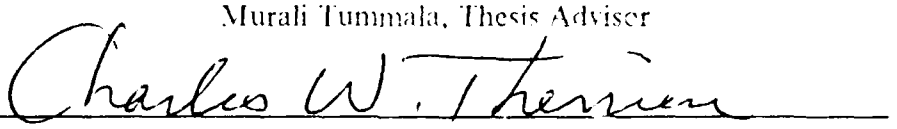
NAVAL POSTGRADUATE SCHOOL
JUNE 1989

Author: 

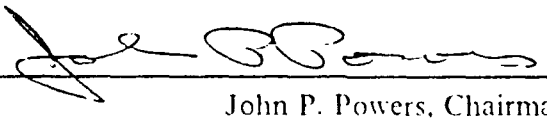
Daniel E. Gear

Approved by: 

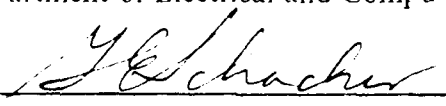
Murali Tummala, Thesis Adviser



Charles W. Therrien, Second Reader



John P. Powers, Chairman,
Department of Electrical and Computer Engineering



Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

Subspace methods of solving spectral estimation and direction of arrival (DOA) problems involve finding the eigenvalues and eigenvectors of correlation matrices. Using the Lanczos algorithm some of the extreme eigenvalues and eigenvectors can be approximated without requiring the entire matrix decomposition theoretically saving many computations.

This thesis develops a model and a form of the Lanczos algorithm to solve the DOA problem. The relationship of the number of eigenvectors used to the accuracy of the results in a low signal to noise ratio example are examined.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Recently, a number of signal processing techniques have been developed that involve resolving a received signal into so-called signal and noise subspaces. The ability to perform spectral estimation or direction of arrival from the determination of the noise subspace has been shown in the works of Pisarenko, Schmidt, Kay, Kailath, and others. The methods vary in the manner in which the subspace is reached and how the resulting eigenvectors are calculated. [References 1, 2 , and 3].

To achieve better results (detection at lower signal-to-noise ratios, better resolution, finer accuracy, less bias), more samples are required. This leads to larger, more complex matrices that better describe the signal and noise subspaces. Determination of the subspaces requires an eigendecomposition of an estimated correlation matrix. The general eigendecomposition of a matrix requires computations $O(n^3)$, thus the larger matrices require many more operations. Once the matrix is decomposed into its eigenvalues and eigenvectors the proper set of eigenvectors must be used to find the resulting spectrum. Hence, estimation is required to split the eigenvalues into signal and noise subsets.

The difficulties in the procedures can be stated with two questions.

- Where is the threshold between signal and noise eigenvalues (and thus which, or how many eigenvectors are used)?
- What method should be used to find those eigenvectors in a timely fashion?

Proposed here is a method which will accurately estimate some of the eigenvalues and eigenvectors of a matrix without performing the entire decomposition. Computational savings are realized when only a partial decomposition is required. The eigenvectors used correspond to the minimum eigenvalues of the matrix. With an over-specified matrix (dimension much larger than the number of signals present), these minimum eigenvalues will all correspond to the noise subspace. Each noise eigenvector contains all the information to find the actual spectrum, although spurious results will also be apparent (because the matrix is over specified). Several methods of handling these spurious peaks are illustrated, including eigenvector averaging, spectral averaging and using the spectral product.

This thesis is organized in five chapters. Chapter II, Direction of Arrival, discusses classical spectral estimation theory and how it applies to the linear array problem (beamforming). Subspace methods starting with Pisarenko Harmonic Decomposition

and proceeding to MUSIC and ESPRIT are discussed in detail. Chapter III, The Lanczos Algorithm, includes all the basic theory required to describe this eigendecomposition method. There we also compare several methods to negate the spurious peaks with the proposed spectral product giving the best results. Results of the algorithm for numerous cases are given in Chapter IV. The last chapter summarizes the results and advantages of this Lanczos subspace method. This chapter also includes some recommendations for future work and lists some possible applications.

II. DIRECTION OF ARRIVAL

Direction of arrival is a form of spectral analysis, performing frequency detection and resolution in the spatial domain vice the conventionally considered time domain. The signals incident on an array are analyzed, and, if the presumptions of the analysis are valid, the correct bearings to the sources are determined. Formerly it was only possible to analyze the output of an array by conventional Fourier techniques. More recently numerous methods have been developed which enhance the ability to accurately determine the spectral and angular resolution.

This chapter summarizes some of the salient points of spectral estimation before discussing classical direction of arrival array processing (Bartlett beamforming). Projection-type superresolution subspace methods are then discussed, starting with Pisarenkos Harmonic Decomposition. MUSIC and ESPRIT are discussed in detail and several other subspace methods are mentioned.

A. SPECTRAL ESTIMATION

Spectral estimation is the term used to describe efforts to find the frequency components of a signal sampled in time. Two conditions that are required for the remainder of this thesis are that the processes considered are wide sense stationary (WSS) and ergodic. The assumption of WSS means that the process has finite mean and that the autocorrelation is a function of the distance, or lag, between two samples and not of the position of the samples themselves. Ergodicity allows time averages to be used to determine ensemble averages.

The autocorrelation function of the signal x(t) is

$$R_{xx}(m) = < x(n+m), x(n) > = \lim_{N \to \infty} \left\{ \frac{1}{2N+1} \sum_{n=-N}^{N} x(n+m)x(n) \right\} \qquad (1)$$

where $x(n)$ are the individual samples of the signal. When only a finite number of samples, M, are taken, the above definition must be modified. The sample autocorrelation function is defined as

$$\hat{R}_{xx}(k) = \frac{1}{M} \sum_{n=1}^{M-1-k} x(n+k)x(n), \quad \text{for } 0 \le k \le (M-1) \tag{2}$$

It is easily shown that [Ref. 4: pp. 56-58]

$$\begin{aligned} \hat{R}_{xx}(k) &= \hat{R}_{xx}(-k), \quad \text{for } -(M-1) \le k \le 0 \\ &\text{and} \\ \hat{R}_{xx}(0) &\ge \hat{R}_{xx}(k), \quad \text{for all } k \end{aligned} \tag{3}$$

The sample autocorrelation matrix $\hat{\mathbf{R}}_{xx}$ is

$$\hat{\mathbf{R}}_{xx} = \begin{bmatrix} \hat{R}_{xx}(0) & \hat{R}_{xx}(-1) & \cdot & \cdot & \cdot & \hat{R}_{xx}(-M) \\ \hat{R}_{xx}(1) & \hat{R}_{xx}(0) & \cdot & \cdot & \cdot & \hat{R}_{xx}(-M+1) \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ \hat{R}_{xx}(M) & \hat{R}_{xx}(M-1) & \cdot & \cdot & \cdot & \hat{R}_{xx}(0) \end{bmatrix} \tag{4}$$

The power spectral density (PSD) is a measure of power per unit frequency. A plot of PSD versus frequency will show the relative power at all the frequencies present. It also describes the properties of the noise in the signal, i.e., white noise will have a flat PSD showing that all frequencies are equally represented [Ref. 1: pp. 53]. The PSD is given by

$$S_{xx}(f) = \frac{1}{T} \sum_{m=-\infty}^{\infty} R_{xx}(m) e^{-j2\pi f m T}$$

where T is the sampling period.

The periodogram method of estimating the true PSD is one of the earliest and most widely used [Ref. 5: pp. 5-8]. The periodogram is defined as the z-transform of the sample autocorrelation matrix evaluated on the unit circle [Ref. 4: pp.52-53].

$$\hat{S}_{xx}(z) = \sum_{k=-(M-1)}^{M-1} \hat{R}_{xx}(k) z^{-k} \tag{6}$$

It may also be found by directly z-transforming the original data sequence x(n)

$$\hat{S}_{xx}(z) = \frac{1}{M} X(z) X(z^{-1}) \quad \text{where} \quad X(z) = \sum_{n=1}^{N} x(n) z^{-n} \tag{7}$$

The periodogram spectrum is found by substituting $z = e^{j2\pi fT}$.

$$\hat{S}_{xx}(f) = \frac{1}{N} |X(f)|^2 = \frac{1}{N} \left| \sum_{n=1}^{N} x(n) e^{-j2\pi fnT} \right|^2 \tag{8}$$

Data is often run through a computationally efficient Fast Fourier Transform (FFT) to find the periodogram spectrum.

The measures of effectiveness of a spectral estimator are based on comparisons of resolution, detectability, bias and variance. *Resolution* relates to the ability of the estimator to separate two separate spectral lines that are closely spaced in frequency. The capacity to locate a low energy signal is measured in detectibility. *Bias* is the tendency of the observed peak to be shifted off the true spectral line. *Variance* is a measure of the propensity to keep the true spectral shape, or how quickly the PSD falls off away from the true peak. Different spectral estimators are sometimes compared in terms of *robustness*, or ability to function well with various types of signals and noise.

If individual signals are narrowband the resolution criterion is said to be achieved when direct observation of the spectrum leads to the correct determination of the number of signals. Separate peaks are not required to determine that two signals are present [Ref. 6]. Resolution is inversely proportional to the length of the data samples. With f_s as the sampling rate and T the sample period, or time between samples, $T = \frac{1}{f_s}$, the frequency resolution is given by

$$\Delta f = \frac{f_s}{M} = \frac{1}{MT} \tag{9}$$

Thus, with the periodogram, better resolution can only be achieved with longer record lengths.

The above description is based on no windowing (or rectangular windowing) of the data samples. The windowing in the time domain is seen as convolution in the frequency domain. The rectangular windows, for example, transform into sinc functions in the frequency domain resulting in high sidelobe distortion known as leakage in the frequency domain. High sidelobes result in many false peaks, making it difficult to discern the true spectral peaks, so detectibility suffers.

Nonrectangular windows are used to taper the data to minimize the discontinuities that cause the high sidelobes. Common windows include the Bartlett and Hamming. The lower sidelobes come at a cost of resolution, so two or more signals close to each other in frequency may merge into one. Resolution may be boosted by lengthening the sample time, but the increased record length may violate the requirement for stationarity. More can be found on the subject of windows in References 7 and 8.

Because of the difficulty in meeting the requirements for both detectibility and resolution, parametric methods have been developed that produce increased resolution with short data lengths. The parametric methods are based on determining an appropriate model for the process that produced the data. If the process can be effectively modeled, then more reasonable assumptions may be made about the data's behavior outside of the window (i.e., these data points do not have to be set to zero). Once the appropriate model is chosen to represent the process, the parameters are estimated from the available data, inserted into the model, and the power spectral density expression for the respective model is determined. A few common parametric methods are summarized below. [Ref 1: pp. 106-108, Ref 5: pp. 172-174]

Many random processes that are encountered in practice may be represented by the linear difference equation

$$x(n) = -\sum_{k=1}^{p} a(k)x(n-k) + \sum_{k=0}^{q} b(k)u(n-k) \tag{10}$$

where $u(n)$ is the input driving sequence and $x(n)$ is the output sequence. The $a(k)$ parameters are the autoregressive coefficients and the $b(k)$ are the moving average coefficients. Equation 10 is thus known as the *autoregressive-moving average* (ARMA) model or ARMA(p,q) process and is the most general model with a rational transfer function. The power spectral density that results from the ARMA model exhibits both sharp peaks and deep nulls. If the autoregressive parameters of equation 10 are all set to zero except $a(0) = 1$, the resultant model is the *moving average* (MA) process of order q. The MA spectrum will have deep nulls, but relatively broad peaks. With the $b(k)$ coefficients of equation 10 set to zero except $b(0) = 1$, the *autoregressive* (AR) process of order p results. The AR spectrum will exhibit sharp peaks but will have broad nulls. [Ref 5: pp. 174-181] Each of the models will exhibit greater accuracy and flexibility as the order is increased. With a high enough order the AR method can approximate an ARMA or MA process, and, likewise, a very high order MA model can be used to approximate an ARMA or AR process. But if the model order is too high for the actual process, estimation errors will lead to nonzero coefficients and spurious peaks will result. Thus proper estimation of model order is important. [Ref 1: pp. 112-113, pp. 198-207].

The parameters of these three models may be estimated by using the Yule-Walker equations utilizing the autocorrelation matrix of the available data stream [Ref 1: pp. 115-118]

$$\mathbf{R}_{xx}\mathbf{a} = -\mathbf{r} \tag{11}$$

While the true autocorrelation function is impossible to determine, the *maximum likelihood* estimator (ML) is one means to find good approximations of the parameters for the AR model. The ML method uses a suitable estimate of the autocorrelation or covariance matrix and then solves [Ref 1: pp. 185-190]

$$\mathbf{C}\hat{\mathbf{a}} = -\mathbf{c} \tag{12}$$

for the parameters $\hat{\mathbf{a}}$ where $\mathbf{C}$ is the symmetric covariance matrix and $\mathbf{c}$ is an autocorrelated vector.

The Burg method (maximum entropy) estimates reflection coefficients from the process and then uses the Levinson recursion to find the estimated parameters [Ref 1: pp. 228-231].

Generally, the various AR spectral estimators except the autocorrelation method are unbiased and have similar variance. The covariance and Burg methods are restricted to

ranges that keep the summation within the available data and do not assume zero pads outside of the samples, thus taking advantage of the basis which led to the creation of the parametric methods in the first place. [Ref 1: pp. 240-252]

B. BEAMFORMING

A conventional approach to the direction of arrival (DOA) problem is through the classical beamforming (Bartlett) technique. Basically, this is a measure of coherency of the signals arriving at an array of sensors. The characteristics of an array are described in terms of array gain, directivity, resolution, beamwidth, and sidelobes. These are based on array size (number of sensors), sensor sensitivity, intersensor spacing, and post reception processing.

Assuming a narrowband point source in the far field, a plane wave will pass through a linear array in a specified order, where the magnitude of the excitation on any individual sensor will be related to the phase of the signal at the instant of sampling. The individual sensors will see different instantaneous magnitudes based on this phase difference which is a function of the frequency (or wavelength), the DOA, and the spacing between sensors. The difference in phase for two successive sensors for a linear array with equally spaced sensors is

$$\Delta\phi = \frac{2\pi d}{\lambda} \sin\theta \tag{13}$$

where d is the distance between sensors, λ is the signal wavelength, and θ is the angle between the wavefront and the array axis. This phase difference $\Delta\phi$ is also known as the normalized wavenumber, k [Ref. 4: pp. 341-343]. Figure 1 illustrates the array-wavefront interaction.

The output of a simple narrowband delay-and-sum beamformer is

$$c(t) = \sum_{m=1}^{M} x_m(t - \tau_m) \tag{14}$$

where $x_m(t)$, $m = 1, 2, \ldots, M$ is the signal at the mth sensor. The time lag, τ_m, between two adjacent sensors is to make up for the propagation delay caused by the wavefront having to travel the extra distance $d \sin\theta$. One can adjust the magnitude of the output to plane waves arriving from a particular direction θ by introducing appropriate time

delays at each sensor prior to summing. This is known as "steering the array" or "steering the beam". An illustration of a typical beamformer arrangement is shown in Figure 2.

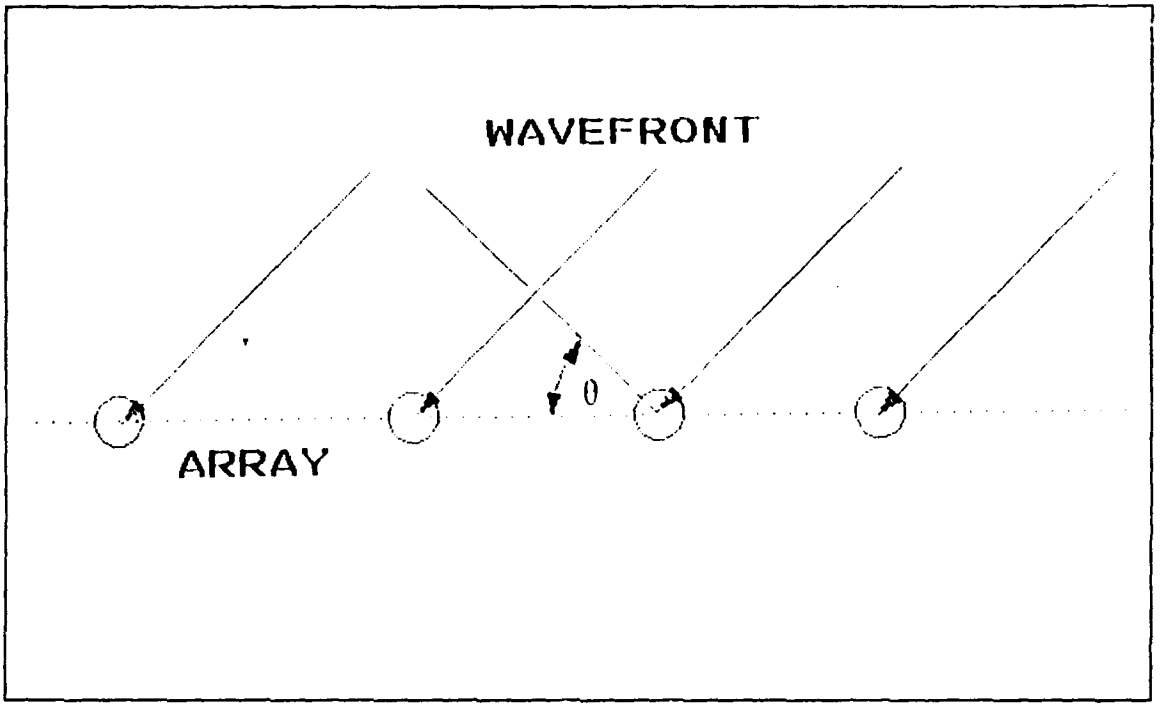


Figure 1. Wavefront

In the multiple source case, especially if the undesired signals are interfering with the detection of other sources, this technique may be modified to steer nulls instead of beams thus minimizing output from "jammers". Nulls may be directed toward a single known source to minimize the strength of the signal with respect to that source, with the output being analyzed to determine what other sources are present. Another implementation is to steer the nulls to minimize the total output. The analysis of the delays will indicate the directions of multiple targets. [Ref. 4: pp. 351-355, Ref. 9]

Beamforming may also be analyzed in the frequency domain by transforming equation 14 into the frequency domain

$$c(f) = \sum_{m=1}^{M} x_m(f) e^{-j2\pi f \tau_m} \tag{15}$$

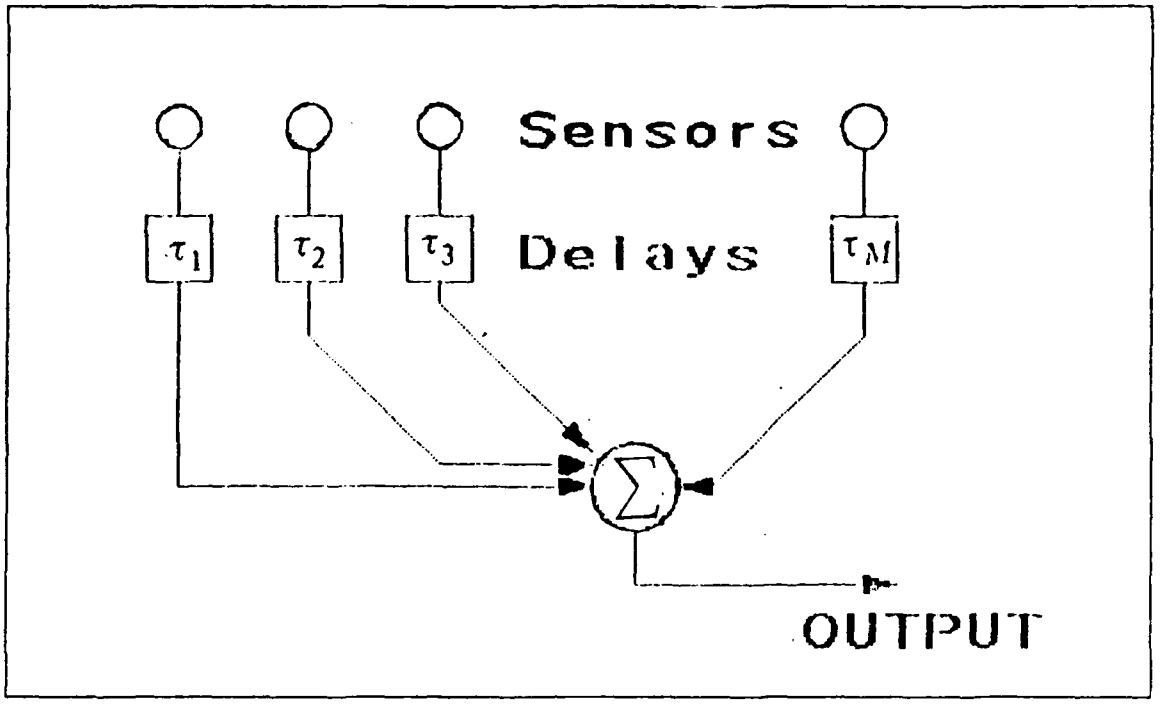


Figure 2. Simple delay-and-sum beamformer

In vector notation we can write $e = \mathbf{w}^T\mathbf{x}$ where w are the weights and x are the outputs of M sensors

$$\text{where} \quad \mathbf{w} = \begin{bmatrix} e^{-j2\pi f\tau_1} \\ e^{-j2\pi f\tau_2} \\ \cdot \\ \cdot \\ \cdot \\ e^{-j2\pi f\tau_M} \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1(f) \\ x_2(f) \\ \cdot \\ \cdot \\ \cdot \\ x_M(f) \end{bmatrix} \tag{16}$$

The steering vector $\mathbf{s}_k$ is the phase relationship between the angle θ and the normalized wavenumber k given in equation 13

$$\mathbf{s}_k = \begin{bmatrix} 1 \\ e^{jk} \\ e^{2jk} \\ \cdot \\ \cdot \\ \cdot \\ e^{(M-1)jk} \end{bmatrix} \quad \text{and} \quad k = \frac{2\pi d}{\lambda} \sin\theta \tag{17}$$

It can be shown that the steering vector from an array with weights **w**, directed toward an arbitrary direction θ, can be expressed in terms of the steering vector as $\mathbf{a} = \mathbf{s}_k$ [Ref. 4: pp. 343-344].

Frequency domain beamforming is directly analogous to spectral estimation decribed above. Spatial sampling has the requirement that sensor distances d must be less than $\lambda/2$ apart to prevent "grating lobes" (or aliasing) corresponding to the Nyquist rate in the frequency domain of $f_{max} \leq f/2$. Longer arrays, containing more sensors, will give better resolution and smoothing. This is similar to frequency resolution being proportional to the data record length ($\Delta f \approx \frac{1}{MT}$). [Ref. 4: pp. 341-345, Ref. 10: pp. 4-8, Ref. 11 : pp. 293-299]

The DOA is actually a relative comparison of observed spatial frequency and known signal frequency. The spatial frequency is greatest on *endfire*, when the wavefront is perpendicular to the array ($\theta = 90°$ or $\frac{\pi}{2}$). Here the phase difference between adjacent sensors is at a maximum. A simultaneous sampling of all sensors at one instant in time, or *snapshot*, will indicate the maximum spatial frequency. Observation of the spatial wave over time (with a known temporal sample rate) will indicate the end of the array where the source is located.

On *broadside* ($\theta = 0$ or π), the wavefront excites each sensor identically. Spatial sampling indicates no phase difference along the entire array, except for the effects of additive noise, resulting in the computation of zero spatial frequency. Unfortunately, linear arrays have an inherent ambiguity with broadside signals. The side of the array at which the source is located cannot be determined without further information. Spatial frequency is illustrated in Figure 3.

An extra requirement in the standard DOA problem is *a priori* knowledge of incoming signal frequencies. Typically, this is handled via a bank of bandpass filters on the output of the sensors. Data streams from the sensors at each desired center fre-

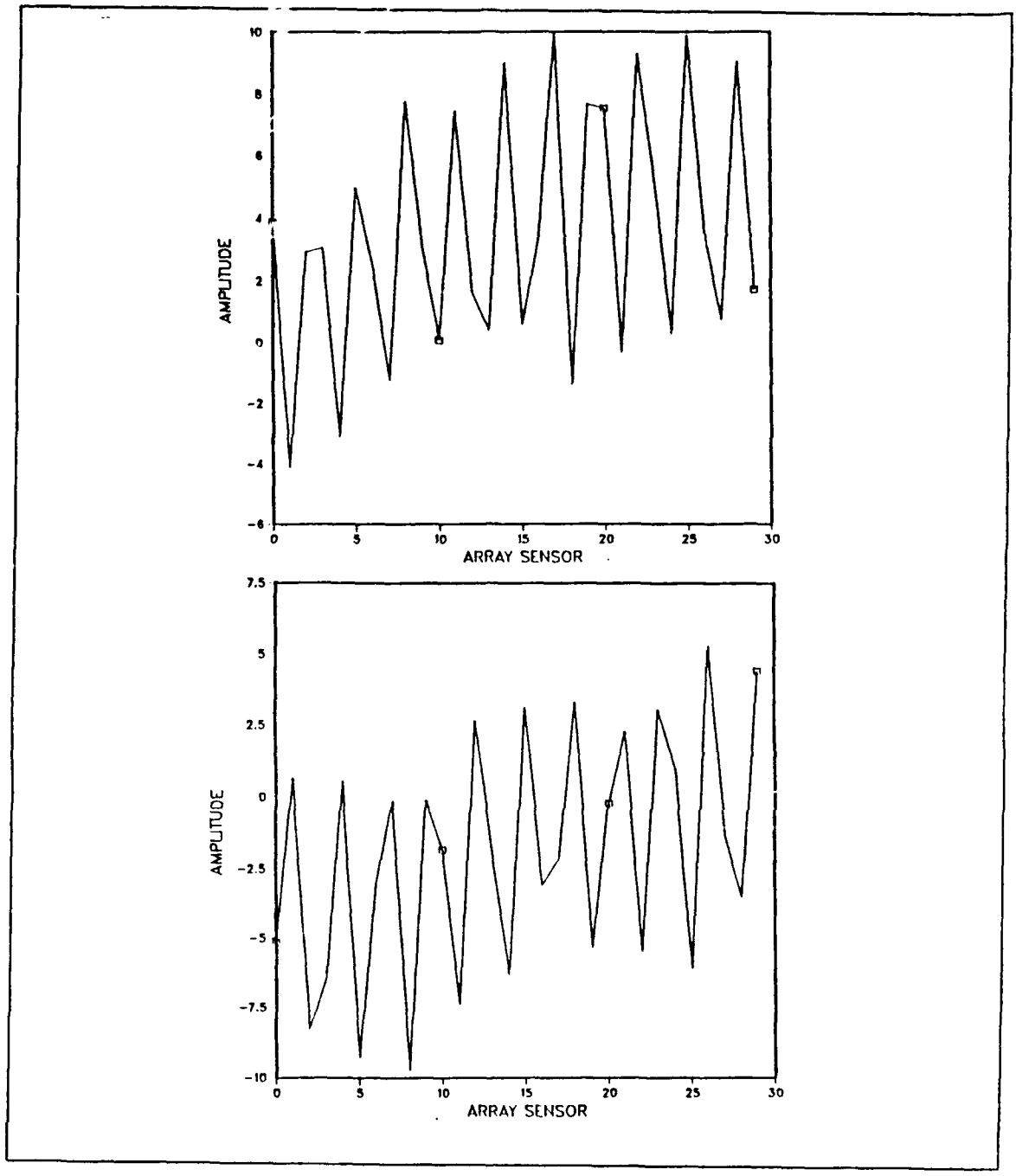


Figure 3. **Spatial Frequency:** Two signals with SNR = 2dB, $\theta_1 = 1°$ and $\theta_2 = 45°$ for two snapshots at time = (a) t_0, (*b*) t_1. Note the variation in 'DC level' as the snapshots sample the nearly broadside signal at different phases.

quency (f_1, f_2, ...) are processed in parallel, resulting in spatial frequencies for each temporal frequency. The DOAs are calculated by comparing these spatial frequencies with the center frequencies of their respective filter banks.

Improvement in beamforming may be seen through the use of windows, weighing each sensor output by the appropriate amount to narrow the beamwidth or lower sidelobes, but, as discussed earlier, at a cost of lowering overall resolution.

C. SUBSPACE METHODS

1. Introduction

The projection-type subspace method utilizes the properties of the autocorrelation, covariance, or modified covariance data matrices and their eigenvalue/eigenvector decomposition into signal components and noise components in estimating the DOA. Generally, subspace methods use an assumed property of the data to provide good resolution if the data fits the particular assumptions, even for extremely short data sets. If the data (and signals) do not meet the assumptions, the results can be quite misleading. The assumptions here call for white noise and a signal whose estimated autocorrelation matrix converges to the true autocorrelation matrix as the order is increased.

For p complex sinusoids in additive complex white noise the combined autocorrelation function of the signal plus noise is given by

$$R_{xx}(m) = \sum_{i=1}^{p} P_i e^{j2\pi f_i m} + \sigma_v^2 \delta(m) \tag{18}$$

We define $\mathbf{R}_{ss}$ as the signal autocorrelation matrix of rank p. $\mathbf{R}_{vv} = \sigma_v^2 \mathbf{I}$ of full rank M and give the autocorrelation matrix as

$$\mathbf{R}_{xx} = \sum_{i=1}^{p} P_i \mathbf{e}_i \mathbf{e}_i^H + \sigma_v^2 \mathbf{I} = \mathbf{R}_{ss} + \mathbf{R}_{vv} \tag{19}$$

where $\mathbf{I}$ is an M by M identity matrix, $\mathbf{R}_{vv}$ is of full rank M due to the noise contribution, P_i is the power of the ith complex sinusoid, σ_v^2 is the noise variance, and $\mathbf{e}_i = [1, e^{j2\pi f_i}, \ldots, e^{j2\pi f_i (M-1)}]^T$. Unfortunately, this decomposition is not possible without

absolute knowledge of the noise. The p signal vectors $\mathbf{e}_i$ contain the frequency information and are related to the eigenvectors of $\mathbf{R}_{xx}$ by $\mathbf{v}_i = \frac{1}{\sqrt{M}}\mathbf{e}_i$ for $i \leq p$. The eigendecomposition of $\mathbf{R}_{xx}$ is

$$\mathbf{R}_{xx} = \sum_{i=1}^{p} (\lambda_i + \sigma_v^2)\mathbf{v}_i\mathbf{v}_i^H + \sum_{i=p+1}^{M} \sigma_v^2\mathbf{v}_i\mathbf{v}_i^H \tag{20}$$

The principal eigenvectors of $\mathbf{R}_{xx}$ are a function of both the signal and noise. If no signal is present the autocorrelation matrix is simply a diagonal matrix with the eigenvalues equal to the variance of the noise [Ref. 1: pp. 422-423]:

$$\mathbf{R}_{xx} = \begin{bmatrix} \sigma_v^2 & 0 & & & & \\ 0 & \sigma_v^2 & 0 & & 0 & \\ & 0 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & 0 & & \cdot & \cdot & 0 \\ & & & & 0 & \sigma_v^2 \end{bmatrix} \tag{21}$$

The data generated from taking M samples of p sinusoids in white noise can be used to generate an autocorrelation matrix with the following properties:

- The theoretical autocorrelation matrix will be composed of a signal autocorrelation matrix and a noise autocorrelation matrix.
- The signal autocorrelation matrix is not full rank if $M > p$.
- The p principal eigenvectors of the signal autocorrelation matrix may be used to find the sinusoidal frequencies.
- The p principal eigenvectors of the signal autocorrelation matrix are identical to the p principal eigenvectors of the total autocorrelation matrix.

The matrix will have a minimum eigenvalue $\lambda = \sigma_v^2$. [Ref. 1: pp. 422-434]

Furthermore, the noise subspace eigenvectors are orthogonal to the signal eigenvectors, or any linear combination of signal eigenvectors. For the *theoretical* autocorrelation matrix, $\mathbf{R}_M$, all eigenvalues not associated with the signals are associated with the noise and are identical in magnitude at $\lambda = \sigma_v^2$, the minimum eigenvalue of $\mathbf{R}_M$. Thus,

$$\mathbf{R}_M \mathbf{v}_M = \lambda_{\min} \mathbf{v}_M \tag{22}$$

The zeros of this minimum eigenvector polynomial

$$\sum_{j=0}^{M-1} \mathbf{v}_{p+1}(j+1) z^{-j} \tag{23}$$

will have p zeros on the unit circle corresponding to signal frequencies [Ref. 4: pp. 335-337, Ref. 5: pp. 371-372].

For the simple case of $M - 1$ complex sinusoids, the autocorrelation matrix $\mathbf{R}_M$ will have a single noise subspace eigenvector $\mathbf{v}_M$ with its associated eigenvalue $\lambda = \sigma_v^2$, the minimum eigenvalue of $\mathbf{R}_M$. Thus, the resulting zeros correspond exactly to the sinusoidal frequencies.

2. MUSIC

MUltiple SIgnal Classification (MUSIC) is a form of a noise subspace frequency estimator, based on noise subspace eigenvectors with uniform weighting. The MUSIC algorithm finds a pseudo spectral estimate from [Ref. 5: p. 373]

$$P_{MUSIC}(f) = \frac{1}{\mathbf{e}^H(f)\left(\sum_{j=p+1}^{M} \mathbf{v}_j \mathbf{v}_j^H\right)\mathbf{e}(f)} \tag{24}$$

where $\mathbf{e} = [1, e^{2\pi f}, \ldots, e^{2\pi f(M-1)}]^T$. The advantage of this method is in its generalized nature. There is no longer is a requirement for uniform spatial sampling. Any array geometry will provide a solution. A well-designed array will eliminate bearing ambiguities and provide unique solutions [Ref. 2: pp. 19-28].

R.O. Schmidt [Refs. 2, 12, 13] has shown that a group of sensors excited by a stationary point source emitter of known frequency will have a magnitude and phase relationship (or correspondence) based on the DOA of the plane wave. This correspondence depends on the array geometry, as well as individual sensors characteristics (i.e., directivity, gain, and frequency response), and may not be unique for that one direction of arrival.

By examining the signal in terms of an M dimensional complex field, where each sensor provides an orthonormal axis, one can see that a single signal will result in one steering vector. This steering vector describes the relationship between the individual sensors in terms of phase and magnitude differences, thus for any unique signal frequency and direction of arrival there is one unique steering vector. The magnitude of the vector will vary with time, but its direction in M space is a constant determined by the amplitude and phase relationship of the sensors for that particular signal as illustrated in Figure 4.

The theory of superposition applies, thus with two signals present the instantaneous received steering vector is a linear combination of the individual steering vectors. The time varying steering vector will move in a plane that is spanned by the individual steering vectors. Figure 5 shows the subspace plane spanned by two steering vectors. The same idea can be expanded to a case of p independent signals causing the steering vector to move through a p dimensional subspace (as long as $M > p$).

Unfortunately, the steering vector may not determine the actual DOA uniquely. In the example of a one-dimensional linear array, a signal gives an infinite number of

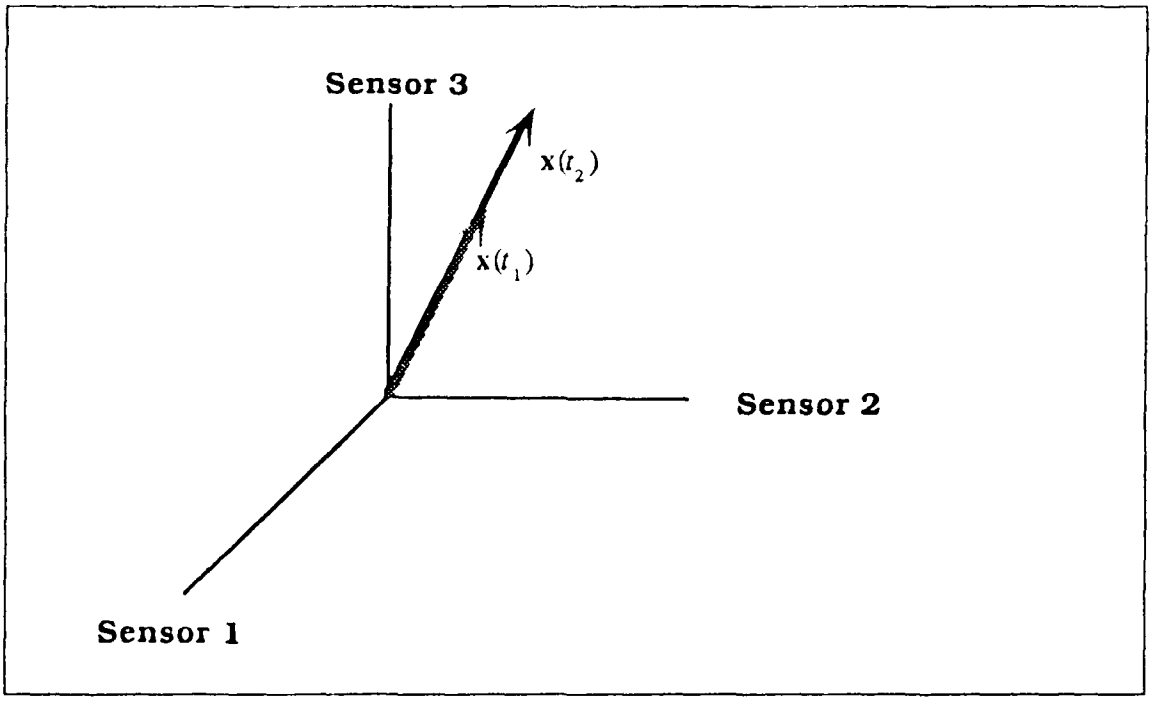


Figure 4. Steering vector for 3 sensors and 1 signal

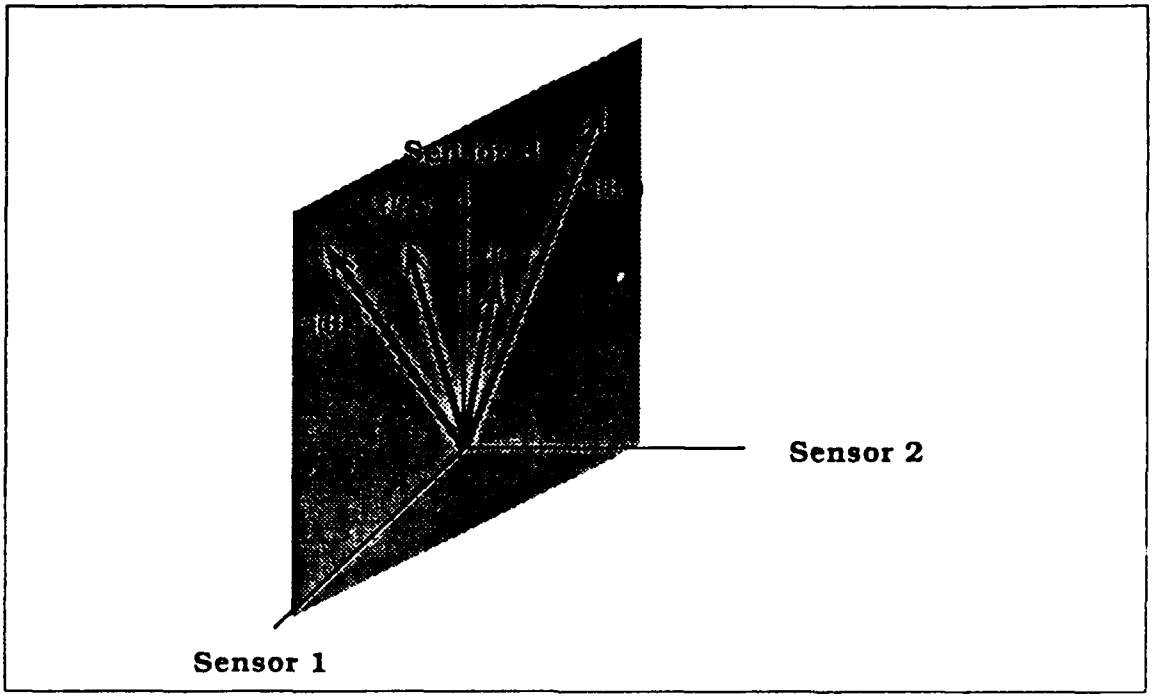


Figure 5. Signal Subspace for 2 signals

DOAs that lie in a cone that is formed by revolving the actual DOA about the axis of the array. Thus the array design and its manifold (expected response) will play a large part in achieving optimum results. The array manifold describes the predicted response of the array to any possible signal or combination of signals. The manifold may be estimated analytically or through physical calibration.

Analytically describing an array is a complex mathematical procedure for all but the simplest arrays (i.e., equally spaced sensors in a linear array or a 3 element orthogonal array). It also assumes that absolute knowledge of the array geometry is available -- an assumption which can lead to errors if the differences are too large.

Calibration with test sources requires a known, low noise environment while the test sources of each desired frequency are placed in a finite number of possible locations. The estimated response to actual signals is an interpolation of these responses. Each set of calibration parameters requires storage in memory; this results in overall massive storage requirements for a comprehensive grid.

Several parameters such as the number of signals present, the directions of arrival of those signals, and the signal strengths can be determined from the signal sub-

space information. More complex models, however, can be developed that can determine signal frequency and polarization as described in References 2 and 13.

We will now describe the basic steps in the MUSIC algorithm for the DOA problem. First, we sample the signals and compute the autocorrelation matrix of the data **R**. Then, we determine the ordered set of eigenvalues and eigenvectors of **R**. In particular, the eigenvectors associated with the minimum eigenvalues must be accurate. In the theoretical case the signal eigenvalues are composed of signal strength (P_i) and noise variance (σ_n^2) and the nonprincipal eigenvalues will all be identically σ_n^2.

We now determine the number of signals by eigenvalue comparison. A simple counting of the eigenvalues greater than σ_n^2 will give the number of signals present in the theoretical case. However, the sample autocorrelation estimates does not lead to such simple results. Small power level signals may have small eigenvalues (perhaps smaller than σ_n^2), and the noise eigenvalues will not actually be identical but will group or cluster near an approximation of σ_n^2. Methods of solution include likelihood ratio tests where the gaps between the eigenvalues determine threshold placement [Ref. 2: pp. 77-79].

Once we find the number of signals present we can determine the signal subspace by the span of the first p eigenvectors

$$\mathbf{V}_S = [\mathbf{v}_1 \; \mathbf{v}_2 \; \cdots \; \mathbf{v}_p] \tag{25}$$

The signal nullspace is its orthogonal complement

$$\mathbf{V}_N = [\mathbf{v}_{p+1} \; \mathbf{v}_{p+2} \; \cdots \; \mathbf{v}_M] \tag{26}$$

We now find the intersection of the signal subspace with the array manifold. The intersection is given in equation 24 for the case of the uniformly spaced linear array. In actuality the intersection is estimated with a "best-fit" method used to determine the optimum result. In the nonlinear case the array manifold is much more difficult to represent.

Two major areas of difficulty with the MUSIC algorithm are the calculation and storage of the array manifold and performing the eigendecomposition of the autocorrelation matrix that results from very large arrays.

3. ESPRIT

In Reference 14, Paulraj, Roy, and Kailath describe Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), a subspace method which utilizes two identical subarrays X and Y. A known distance called a displacement vector

separates the two parallel subarrays, but no rotation can be present. Each sensor in a matching pair (doublet) must be identical, but knowledge of individual sensor and array response characteristics is not required.

The N elements of both arrays are sampled simultaneously with the signal at each sensor being given by

$$x_i(t) = \sum_{k=1}^{p} s_k(t) a_i(\theta_k) + n_{x_i(t)}$$

$$y_i(t) = \sum_{k=1}^{p} s_k(t) e^{j\frac{2\pi d}{\lambda}\sin\theta_k} a_i(\theta_k) + n_{y_i(t)} \tag{27}$$

where the sampled signal at each sensor in a doublet differs only by a phase term and additive noise. With p signals present, s_i is the wavefront at the reference sensor in the X array, θ_i is the DOA relative to the displacement vector, $a(\theta_i)$ is the response of the ith sensor in a subarray relative to the reference sensor in that array for a signal from bearing θ_i, d is the magnitude of the displacement vector and n the noise term. In vector notation the signals at the subarrays are

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{n}_x(t)$$

$$\mathbf{y}(t) = \mathbf{H}\Phi\mathbf{s}(t) + \mathbf{n}_y(t) \tag{28}$$

where

$$\begin{aligned} \mathbf{x}(t) &= [x_1(t), x_2(t), \ldots, x_m(t)]^T, \\ \mathbf{y}(t) &= [y_1(t), y_2(t), \ldots, y_m(t)]^T, \\ \mathbf{n}_x(t) &= [n_{x_1}(t), n_{x_2}(t), \ldots, n_{x_m}(t)]^T, \\ \text{and} \quad \mathbf{n}_y(t) &= [n_{y_1}(t), n_{y_2}(t), \ldots, n_{y_m}(t)]^T \end{aligned} \tag{29}$$

The vector of wavefronts at the reference sensor in array X is represented by $\mathbf{s}(t)$. All displacements and phase differences are based on this sensor. The p by p diagonal matrix, Φ, contains the phase delays that occur with each set of doublets

$$\Phi = \mathrm{diag}\left[e^{j\phi_1}, e^{j\phi_2}, \ldots, e^{j\phi_p}\right] \tag{30}$$

where $\phi_i = \frac{2\pi d}{\lambda}\sin\theta_i$, as shown in equation 13.

If $\mathbf{R}_{pp}$ is the signal autocorrelation matrix, the subarray autocorrelation matrix is given by

$$\mathbf{R}_{xx} = \mathbf{H}\mathbf{R}_{pp}\mathbf{H}^T + \sigma_v^2\mathbf{I} \tag{31}$$

The cross correlation between subarrays is

$$\mathbf{R}_{xy} = \mathbf{H}\mathbf{R}_{pp}\Phi^T\mathbf{H}^T \tag{32}$$

where $\mathbf{H}$ is the direction matrix whose columns are the direction vectors for the p wavefronts

$$\mathbf{h}(\theta_k) = [h_1(\theta_k),\ h_2(\theta_k),\ \ldots,\ h_m(\theta_k)]^T \tag{33}$$

After some manipulation [Ref. 15: pp. 251-253], we obtain the matrix

$$\begin{aligned}\Gamma &= (\mathbf{R}_{xx} - \lambda_{\min}\mathbf{I}) - \gamma\mathbf{R}_{xy} \\ &= \mathbf{H}\mathbf{R}_{pp}\mathbf{H}^T - \gamma\mathbf{H}\mathbf{R}_{pp}\Phi^T\mathbf{H}^T \\ &= \mathbf{H}\mathbf{R}_{pp}(\mathbf{I} - \gamma\Phi^T)\mathbf{H}^T\end{aligned} \tag{34}$$

In general, there will be p eigenvalues of this matrix. But when $\gamma = e^{\frac{2\pi i}{\lambda}\sin\theta_k}$, the kth row of $(\mathbf{I} - \gamma\Phi^T)$ becomes zero, leaving p-1 eigenvalues. Each value of γ where this occurs is called a *generalized eigenvalue (GE)*. Now, the DOAs can be determined by

$$\theta_k = \arcsin\left(\frac{\lambda_k\phi_k}{2\pi d}\right) \tag{35}$$

Due to estimation errors in the calculation of the correlation matrices, some error will be induced. Generally, the GE's will be moved off the unit circle and out from the origin, but in the case of strong signals, they will still be easily discernible. It should be noted that poor array design may result in possible ambiguities (similar to MUSIC).

Advantages to note over the MUSIC algorithm come with respect to the array manifold. With ESPRIT, no manifold is required. The considerable calibration efforts and storage requirements are nonexistent. However, the two subarrays must be identical in all respects and must be positioned exactly parallel to each other [Ref. 14].

4. Other Subspace Methods

Large variance effects may be seen in the above methods due to the poor reliability of the estimation of the eigenvectors associated with the minimum eigenvalues of the estimated autocorrelation matrix $\hat{\mathbf{R}}_{xx}$. A different method of spectral estimation is the use of the principal components where only the eigenvectors associated with the largest eigenvalues are used. Some methods have tried to minimize the effect of noise by using $\mathbf{R}_{xx} - \sigma_v^2\mathbf{I}$ but the estimation errors in both $\mathbf{R}_{xx}$ and σ_v^2 have limited their success [Ref. 1: pp. 425]. Other spatial spectral methods include the parametric methods such as AR and ML [Ref. 1: pp. 426-427].

The structured matrix approximation of Kumaresan and Shaw [Ref. 16] uses K snapshots in time of an N element uniformly spaced linear array with each snapshot in time forming a column of a data matrix $\mathbf{X}$, which is then approximated by $\hat{\mathbf{X}}_M$ in the least squares sense. The bearings information is then calculated using the properties of the Vandermonde matrix.

A combination of frequency domain beamforming and autoregressive modeling techniques have been employed in Reference 17 to estimate the direction of arrival in a multiple source localization situation using a planar array.

Halpeny and Childers [Ref. 18] use frequency-wavenumber filters to break the multiple wave case down to a succession of single wave problems.

Reference 19 explains an algorithm that uses non-noise eigenvectors from a covariance matrix to obtain high resolution direction of arrival for narrow band sources.

An adaptive beamforming method similar to a minimum energy approach is decribed in Reference 20. The eigenstructure of the correlation matrix is analyzed and the computations are modified to optimize resolution but at a cost of array gain.

III. THE LANCZOS ALGORITHM

Lanczos algorithms provide a method to find some of the extreme eigenvalues (smallest or largest) and their associated eigenvectors from large matrices with fewer operations than required in an entire matrix decomposition. The procedure is a tridiagonalization of the original matrix based on an iteration developed by Cornelius Lanczos [Ref. 21: pp. 49-206]. Once the matrix is in a tridiagonalized form the eigenvalues can be easily computed using a symmetric QR algorithm [Ref. 15: pp. 278-279] or bisection (with or without Sturm sequencing) [Ref. 15: pp. 305-308]. The algorithm takes advantage of "minimized iterations" providing quick convergence to the final tridiagonal matrix and avoiding accumulation of roundoff error. [Ref. 22]

The method fell from favor as a tridiagonalization technique with the advent of the Givens and Householder transformations later on in the 1950s. Givens transformations [Ref. 15: pp. 38-47] use plane rotations (orthogonal matrices) to zero out undesired entries in the matrix undergoing tridiagonalization. The Householder methods [Ref. 15: pp. 43-47] use elementary reflectors to accomplish the same end. Both methods are inherently stable, with the Householder method being slightly superior in both speed and accuracy. Both methods outperformed Lanczos as a complete tridiagonalization procedure in the general case where all eigenvalues are required [Ref. 23: pp. 42-43].

The real power of the Lanczos method however lies in obtaining the extreme values quickly. The entire matrix need not be tridiagonalized before solutions start to converge. Also, if the original matrix is sparse, the Lanczos method maintains that property, saving even more computations. Thus for large matrices (order > 100) the Lanczos method will converge on extreme eigenvalues in many fewer operations. Recently Tufts and Melissinos [Ref. 24: pp. 43-47] have derived a variation of the Lanczos method for high resolution spectral estimation and showed that their method outperforms both the singular value decomposition and the power method of principal eigenvector and eigenvalue computation. Later in this chapter, it will be shown that storage requirements are also minimized.

This chapter starts with an explanation of the classical Lanczos algorithm as developed by Lanczos and modified by Paige [Ref. 25]. Then we will discuss the unorthodox Lanczos algorithm of Cullum and Willoughby where no reorthogonalization is performed [Ref. 23]. Finally, the algorithm used in the direction of arrival problem are

discussed in detail. Also, we present some results of the algorithm in the form of spectral estimation of multiple tones.

A. CLASSICAL LANCZOS AND ITS VARIANTS

The original algorithm was designed as a means to directly tridiagonalize the symmetric matrix **A**. The development of Lanczos algorithm requires the knowledge of Krylov sequences and subspaces. For an n by n matrix **A** and any arbitrary non-null n by 1 vector $\mathbf{v}_1$ the Krylov sequence of vectors is defined as:

$$\mathbf{v}_{i+1} = \mathbf{A}\mathbf{v}_i = \mathbf{A}^i\mathbf{v}_1 \quad \text{for } k = 1, 2, \ldots, n \tag{36}$$

In the above sequence there will be a vector, say $\mathbf{v}_{m+1}$, which can be expressed as a linear combination of all the preceding vectors. The Krylov matrix of rank m is then given by

$$\mathbf{V}_m = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3 \ \ldots \ \mathbf{v}_m] = [\mathbf{v}_1, \ \mathbf{A}\mathbf{v}_1, \ \mathbf{A}^2\mathbf{v}_1, \ldots, \ \mathbf{A}^{m-1}\mathbf{v}_1] \tag{37}$$

and the Krylov subspace is the space that spans the vectors $[\mathbf{v}_1, \ \mathbf{v}_2, \ldots, \mathbf{v}_m]$,

$$K^m(\mathbf{A}, \mathbf{v}_1) = \text{span}\left\{\mathbf{v}_1, \ \mathbf{A}\mathbf{v}_1, \ \mathbf{A}^2\mathbf{v}_1, \ldots, \ \mathbf{A}^{k-1}\mathbf{v}_1\right\} \tag{38}$$

The columns of the n by m Krylov matrix $\mathbf{V}_m$ are orthogonal. The tridiagonalization of the given matrix **A** is then obtained as

$$\mathbf{T} = \mathbf{V}_m^T\mathbf{A}\mathbf{V}_m, \tag{39}$$

where **T** is an m by m tridiagonal matrix. Thus, the given matrix **A** can be tridiagonalized provided we have an efficient way to compute the orthogonal matrix $\mathbf{V}_m$ or to compute the elements of matrix **T** by performing the decomposition of equation 39 directly as proposed by Lanczos [Ref. 22, Ref. 15].

The most direct way of performing the tridiagonalization assumes that $\mathbf{T} = \mathbf{V}^T\mathbf{A}\mathbf{V}$ where $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \ldots \ \mathbf{v}_n]$. Note that **A** is a symmetric n by n matrix and **V** is an n by m orthogonal matrix constructed from the Krylov sequence of vectors. The basic recursion for a real n by n matrix **A** starts with a randomly generated unit Lanczos vector $\mathbf{v}_1$. Define scalar $\beta_1 = 0$ and $\mathbf{v}_0 = \mathbf{0}$. Define Lanczos vectors $\mathbf{v}_i$ and elements α_i and β_{i-1} for $i = 1, 2, \ldots, M$,

$$\beta_{i+1}\mathbf{v}_{i+1} = \mathbf{A}\mathbf{v}_i - \alpha_i\mathbf{v}_i - \beta_i\mathbf{v}_{i-1} \tag{40}$$

$$\alpha_i \equiv \mathbf{v}_i^T \mathbf{A} \mathbf{v}_i \tag{41}$$

$$\beta_{i+1} \equiv \mathbf{v}_{i+1}^T \mathbf{A} \mathbf{v}_i \tag{42}$$

This is referred to as the basic Lanczos iteration or recursion. The actual Lanczos matrix $\mathbf{T}_j$, $j = 1, 2, \ldots, M$, is a symmetric tridiagonal matrix with α_i, $1 \le i \le j$, on the diagonal, and β_{i-1}, $1 \le i \le (j-1)$, above and below the diagonal.

$$\mathbf{T}_j = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & & & \\ \beta_2 & \alpha_2 & \beta_3 & 0 & & \\ 0 & \beta_3 & \cdot & \cdot & \cdot & \\ & 0 & \cdot & \cdot & \cdot & 0 \\ & & \cdot & \cdot & \cdot & \beta_j \\ & & & 0 & \beta_j & \alpha_j \end{bmatrix} \tag{43}$$

Each new Lanczos vector $\mathbf{v}_{i+1}$ is obtained from orthogonalizing the vector $\mathbf{A}\mathbf{v}_i$ with $\mathbf{v}_i$ and $\mathbf{v}_{i-1}$. The elements α_i and β_{i-1} are the scalar coefficients that make up the Lanczos matrix. The basic Lanczos recursion may be condensed into matrix notation by defining $\mathbf{V}_m = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m)$. Then from equations 40, 41, and 42

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m \mathbf{T}_m + \beta_{m+1} \mathbf{v}_{m+1} \mathbf{e}_m^T \tag{44}$$

where $\mathbf{e}_m$ is the coordinate vector whose mth component is 1 while the rest of the components are 0. $\mathbf{T}_m$ is the m by m Lanczos matrix, the diagonal entries are

$$\mathbf{T}_m(k,k) = \alpha_k \quad \text{for} \quad 1 \le k \le m, \tag{45}$$

and the entries along the two sub-diagonals on either side of the principal diagonal are

$$\mathbf{T}_m(k,k+1) \equiv \mathbf{T}_m(k+1,k) \equiv \beta_{k+1} \quad \text{for} \quad 1 \le k \le m-1 \tag{46}$$

Note that $\mathbf{A}$ is never modified (unlike in Givens and Householder transformations), thus advantage may be taken of any existing sparsity. Also, the only storage requirements are for the three Lanczos vectors (n dimension), the Lanczos matrix $\mathbf{T}_j$, and the original matrix $\mathbf{A}$.

We summarize the actual steps:

- Use the basic recursion of equations 40, 41, and 42 to transform the symmetric matrix $\mathbf{A}$ into a family of symmetric tridiagonal matrices $\mathbf{T}_j$, $j = 1,2,\ldots,M$.

- For $m \leq M$ find the eigenvalues of $\mathbf{T}_m$, μ (also known as the Ritz values of $\mathbf{T}_m$).
- Use some or all of these eigenvalues, μ, as approximations of the eigenvalues of $\mathbf{A}$, λ.
- For each eigenvalue μ find a unit eigenvector $\mathbf{u}$ so that $\mathbf{T}_m\mathbf{u} = \mu\mathbf{u}$.

The Ritz vector $\mathbf{y}$ is the approximation of the eigenvector of $\mathbf{A}$. It is found from mapping the eigenvector $\mathbf{u}$ of the Lanczos matrix.

$$\mathbf{y} \equiv \mathbf{V}_M\mathbf{u} \tag{47}$$

So the eigendecomposition of $\mathbf{T}_m$ finally results in the Ritz pair $(\mu, \mathbf{y})$ which approximates the eigenvalues and eigenvectors of $\mathbf{A}$. [Ref. 23: pp. 32-33., Ref. 15: pp. 322-325.]

Unfortunately, the effects of finite precision arithmetic prevent the theoretical Lanczos algorithm from working. The eigenvalues of $\mathbf{A}$ and the eigenvalues of $\mathbf{T}_m$ no longer converge. Roundoff errors are partially to blame, but the dominant effect is from the loss of orthogonality of the Lanczos vectors $\mathbf{v}_j$. From equation 40 it can be seen that a small $\hat{\beta}_{j-1}$ will have great effect on $\hat{\mathbf{v}}_{j+1}$. Paige showed that the algorithm runs within allowable error constraints until it starts to converge on the first eigenvalue. At this point $\hat{\beta}_{j-1}$ becomes small and the Lanczos vectors lose orthogonality. The loss of orthogonality is not random, however. It always occurs in the direction of the Ritz vector $\mathbf{y}$.

This trouble can be dealt with through reorthogonalization. But questions that we must answer include:

- How much reorthogonalization is required?
- Where should it be performed?
- Reorthogonalize with respect to what?

Complete reorthogonalization is the first choice, inserting a Householder matrix computation into the Lanczos algorithm. This enforces orthogonality among all the Lanczos vectors and is effective at keeping the system stable. But the extra computations it requires negate any advantage of the Lanczos algorithm, making the number of computations on the same order as a complete decomposition. Numerous vectors have to be stored and compared requiring many swaps into and out of storage. [Ref. 15: pp. 334-335]

Selective reorthogonalization is clearly more efficient. The extra computations are performed only if orthogonality checks indicate loss is imminent. Paige shows that the loss of orthogonality occurs only when the algorithm converges on a Ritz pair. Thus, instead of reorthogonalizing against every other Lanczos vector, using the less numerous Ritz vectors will be sufficient. Storage is therefore minimized. Only when all eigenvalues of the matrix are required does this method become computationally more expensive than other techniques. [Ref. 26]

The last option is *no reorthogonalization*. The explanation above would seem to indicate that maintaining orthogonality is required. However by analyzing the causes and effects of the original loss of orthogonalization one can insert corrections into the solution to give valid eigenvalues and eigenvectors. This is the procedure that will be examined in the next section.

One other property will be mentioned. Here the single vector Lanczos recursion described above will not find duplicate eigenvalues of the matrix **A**. Parlett's proof uses the power method to expand **v** to compute the columns of the Krylov matrix $\mathbf{K}^n(\mathbf{v}, \mathbf{A})$. If $\mathcal{I}(\mathbf{v})$ is the smallest invariant subspace of $\mathbb{R}^n$ which contains **v**, then the projection of **A** onto $\mathcal{I}(\mathbf{v})$ has simple eigenvalues. Also, the Krylov subspaces fill up $\mathcal{I}(\mathbf{v})$ and for some $l \leq n$ we have

$$span\{\mathbf{v}\} \subset \mathbf{K}^2(\mathbf{A}, \mathbf{v}) \subset \mathbf{K}^3(\mathbf{A}, \mathbf{v}) \subset \ \ldots \subset \mathbf{K}^l(\mathbf{A}, \mathbf{v}) = \mathbf{K}^{l+1}(\mathbf{A}, \mathbf{v}) = \mathcal{I}(\mathbf{v}) \tag{48}$$

The result is that some eigenvectors of **A** may be missed, and any repeated eigenvalues will not be detected. [Ref. 27: pp. 235-239]

The multiple eigenvalue problem can be treated by using a *Block Lanczos* algorithm. Instead of obtaining a tridiagonal matrix, the result is a banded block matrix, where the diagonal is $\mathbf{M}_n$, an l by l matrix, and the above and below the principal diagonal are upper triangular matrices $\mathbf{B}_n$. Each block must be dimensioned $l \geq p$, where p is the estimated multiplicity of the desired eigenvalue.

$$\mathbf{T}_j = \begin{bmatrix} \mathbf{M}_1 & \mathbf{B}_2^T & 0 & & & \\ \mathbf{B}_2 & \mathbf{M}_2 & \mathbf{B}_3^T & 0 & & \\ 0 & \mathbf{B}_3 & \cdot & \cdot & \cdot & \\ & 0 & \cdot & \cdot & \cdot & 0 \\ & & 0 & \cdot & \cdot & \mathbf{B}_j^T \\ & & & 0 & \mathbf{B}_j & \mathbf{M}_j \end{bmatrix} \tag{49}$$

This is analogous to the general case [Ref. 15: pp. 337-339]. The block Lanczos algorithm is briefly reviewed at the end of this chapter.

The above discussion assumes that the given matrix A is real and symmetric. Besides the algorithms summarized in this section for a real and symmetric matrix case, there are other general Lanczos algorithms proposed in the literature [Ref. 23] that are suitable for Hermitian matrices, non-symmetric matrices, and for rectangular matrices.

B. IMPLEMENTATION

The Lanczos phenomenon states that a few, not all, of the eigenvalues of a very large matrix **A** can be computed using the Lanczos recursion with no reorthogonalization. But to find most of the eigenvalues, the Lanczos matrix, $\mathbf{T}_m$, will grow in size approaching that of **A**, causing the loss of orthogonality of the Lanczos vectors. The loss of orthogonality results in many spurious eigenvalues, as well as extra copies of good eigenvalues. In any case, a test is required to confirm either:

- a "found" eigenvalue is good, or
- the eigenvalue that appears is spurious.

Golub and Van Loan, Parlett, and Paige [Refs. 15, 27, and 28] describe procedures that look at the eigenvalues for each $\mathbf{T}_m$ as m is stepped up in size. All the eigenvalues of $\mathbf{T}_m$ are computed at each step. The good eigenvalues will repeat at each larger $\mathbf{T}_m$, while the spurious eigenvalues jump around. If an eigenvalue does not match at consecutive $\mathbf{T}_m$'s it may be considered spurious and thrown out. If a good eigenvalue is mistakenly deleted (due to numerical roundoff), it can be counted on to reappear in the next step.

Cullum and Willoughby [Ref. 23] take a different tack by developing a test to find and eliminate bad eigenvalues and retain the rest. The advantage here is in utilizing the machine's tolerance to drop bad eigenvalues, while not discarding good eigenvalues that have yet to converge. As a result a larger spectrum is available sooner, even though it may only be a rough estimate of where the eigenvalues will finally converge.

In practice, parts of the Lanczos recursion (equations 41 and 42) are replaced by

$$\alpha_i = \mathbf{v}_i^T(\mathbf{A}\mathbf{v}_i - \beta_i\mathbf{v}_{i-1}) \tag{50}$$

and

$$\beta_{i+1} = \|\mathbf{A}\mathbf{v}_i - \alpha_i\mathbf{v}_i - \beta_i\mathbf{v}_{i-1}\| \tag{51}$$

Computation of the element α_i is a modified Gram-Schmidt orthogonalization procedure. The new β_{i-1} is equivalent to the β_{i-1} of equation 42 but now it directly controls the size of the Lanczos vector.

In what follows we describe two Lanczos algorithms, namely the single vector Lanczos algorithm which is modified and analyzed by Paige [Ref. 28] and the block Lanczos algorithm described by Cullum and Willoughby [Ref. 23]. Both algorithms have been considered for the estimation of the directions-of-arrival of multiple targets in noisy environments in this thesis.

1. Single Vector Lanczos Algorithm

The first procedure to be described is the Paige's single vector Lanczos algorithm [Ref. 28] for real symmetric matrices. The single vector Lanczos procedure is one of the most straightforward implementations of the theory. This procedure will find some or many of the eigenvalues and eigenvectors of a real symmetric matrix **A** such that $\mathbf{Ax} = \lambda\mathbf{x}$. It will not detect repeated eigenvalues. However, it may be noted that for many problems of interest in practice we do not have strictly multiple eigenvalues. For example, in the direction-of-arrival problems the smallest eigenvalues of the autocorrelation matrix corresponding to the noise associated with the target signals are spread over a small range rather than coinciding on the same value [Ref. 2].

No reorthogonalization is performed as part of the single vector Lanczos algorithm. As mentioned earlier, the Lanczos vectors begin to lose their orthogonality when we seek to estimate all or most of the eigenvalues of the real symmetric matrix **A**. For the application under consideration, however, we are generally interested in only a few of the minimum eigenvalues and the corresponding eigenvectors. It is mainly for this reason that we have not attempted the complete or selective reorthogonalization of the Lanczos vectors in this study.

Now we shall outline the basic steps involved in the single vector Lanczos algorithm. This is based on the recursion described by equations 40, 50 and 51. Based on these equations Paige [Ref. 28] presented four different single vector algorithms. We have adapted one of in this study. The complete eigenvalue eigenvector problem actually consists of three parts: (a) obtaining the tridiagonal matrix $\mathbf{T}_n$ from the given symmetric and real matrix **A** using Paige's recursion, (b) determining the smallest eigenvalues of $\mathbf{T}_n$ using the bisection method and Sturm sequencing, and (c) estimating the corresponding eigenvectors by computing the Ritz vectors. The details are presented in the following.

Step 1: As shown in equation 43 the symmetric tridiagonal matrix $\mathbf{T}_j$ has entries α_j and β along its principal, and the adjacent sub and super diagonals, respectively. The following recursive expressions are then used to compute the entries of the tridiagonal matrix, and also the Lanczos vectors $\mathbf{v}_j$ [Ref. 28]:

Initial conditions: $\mathbf{v}_1$ is an arbitrary n by 1 vector such that $\|\mathbf{v}\|_2 = 1$

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{A}\mathbf{v}_1 \\ \alpha_0 &= \beta_0 = 0 \\ \text{for } j &= 1, 2, \ldots, m \\ \alpha_j &= \mathbf{v}_j^T \mathbf{u}_j \\ \mathbf{w}_j &= \mathbf{u}_j - \alpha_j \mathbf{v}_j \\ \beta_j &= \|\mathbf{w}_j\|_2 \\ \mathbf{v}_{j+1} &= \mathbf{w}_j / \beta_j \\ \mathbf{u}_{j+1} &= \mathbf{A}\mathbf{v}_{j+1} - \beta_j \mathbf{v}_j \end{aligned} \tag{52}$$

where $\mathbf{w}_j$ and $\mathbf{u}$ are some intermediate vectors. The vector $\mathbf{v}_1$ is obtained by filling its entries with a random number sequence and then normalizing it with respect to its Euclidean norm. Now $\mathbf{T}_m$ is obtained by simply filling its entries as in equation 43. Note that $m \ll n$ in the above. One quick test to ensure that we have obtained a fairly accurate estimate of $\mathbf{T}_m$ is to compute the product $\mathbf{v}_j^T \mathbf{v}_{j'}$. The result should be equal to $\delta_{jj'}$, where $\delta_{jj'}$ is the Kronecker delta function.

Step 2: The eigenvalues of the tridiagonal matrix $\mathbf{T}_m$, denoted by μ_j, can be computed using the bisection method and Sturm sequencing. Actually one could obtain both eigenvalues and eigenvectors of $\mathbf{T}_m$ by employing such methods as the QR algorithm. However, when only a few eigenvalues are required, the bisection method seems appropriate. For the given m by m matrix $\mathbf{T}_m$ we define the characterstic polynomial

$$p_m(\mu) = \det(\mathbf{T}_m - \mu \mathbf{I}) \tag{53}$$

which can be recursively computed as follows

$$\begin{aligned} p_0(\mu) &= 1 \\ p_1(\mu) &= \alpha_1 - \mu \\ p_j(\mu) &= (\alpha_j - \mu) p_{j-1}(\mu) - \beta_{j-1}^2 p_{j-2}(\mu) \\ &\quad \text{for } j = 2, 3, \ldots, m \end{aligned} \tag{54}$$

The roots of the polynomial $p_m(\mu)$ are the required eigenvalues. For our application, we are only interested in a small range of eigenvalues at the lowest end of the eigenvalue spectrum. We make use of the Sturm sequencing property that the eigenvalues of $\mathbf{T}_{j-1}$ strictly separate those of $\mathbf{T}_j$ [Ref. 15: pp. 305-307] and implement the following iteration:

$$\mu = \frac{a+b}{2}$$

$$b = \mu \quad \text{if} \quad p_m(a)p_m(b) < 0 \tag{55}$$

$$a = \mu \quad \text{if} \quad p_m(a)p_m(b) \geq 0$$

and we repeat the above as long as $|b-a| > \varepsilon(|b|+|a|)$, where ε is the machine unit round-off error and $[a,b]$ is the range of our required eigenvalues. Determining the range of interest in our application may require some a priori knowledge about the signal to noise ratios (SNR) and it may take a couple of iterations to do this. Some alternatives to the iteration given in equation 55 are to use a straightforward polynomial root finder and then pick the roots of interest, or to employ the well known L-D-U factorization, both of which may not be computationally efficient.

Step 3: There are two ways to obtain the eigenvectors of $\mathbf{A}$ knowing its eigenvalues, λ_j. Note that μ_j are the estimates of λ_j. In the first method, we compute the eigenvectors of T_m, denoted by $\mathbf{t}_j$, and then obtain the eigenvectors of $\mathbf{A}$ given by

$$\mathbf{x}_j = L_m \mathbf{t}_j \tag{56}$$

where $L_m = [\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_m]$ is the Lanczos matrix which ideally is the same as the Krylov matrix of equation 37. Note that $(\mu_j, \mathbf{t}_j) \in \mathbf{T}_m$.

The second method involves computing the Ritz vectors either from the Rayleigh quotient interation or by the orthogonal iteration. Here we assume that we have good estimates of λ_j from the previous step, and proceed to obtain the eigenvector $\mathbf{x}_j$ by minimizing the cost function

$$J = \|(\mathbf{A} - \lambda_j \mathbf{I})\mathbf{x}_j\|_2 \tag{57}$$

It can be shown that a simple minimization of J with respect to $\mathbf{x}_j$ yields the Rayleigh quotient of $\mathbf{x}_j$.

$$r(\mathbf{x}_j) = \lambda_j = \frac{\mathbf{x}_j^T \mathbf{A} \mathbf{x}_j}{\mathbf{x}_j^T \mathbf{x}_j} \tag{58}$$

Therefore, given λ_j and using equation 58 we can formulate the Rayleigh quotient iteration as follows [Ref. 15, 27]

Initial condition: $\mathbf{x}_0$ is an arbitrary vector such that $\|\mathbf{x}_0\|_2 = 1$

for $k = 0, 1, 2, \ldots$

$$r(\mathbf{x}_j) = \frac{\mathbf{x}_k^T \mathbf{A} \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k} \tag{59}$$

$$\text{solve} \quad (\mathbf{A} - r(\mathbf{x}_k)\mathbf{I})\mathbf{y}_{k+1} = \mathbf{x}_k \quad \text{for} \quad \mathbf{y}_{k+1}$$

$$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\|_2$$

where $\mathbf{y}_{k+1}$ is some intermediate vector. We stop the iteration when $\|\mathbf{y}_{k+1}\|_2$ converges to a constant or when $r(\mathbf{x}_j) \simeq \lambda_j$, one of the known eigenvalues. At each iteration step we need to solve an n by n system of equations in this method. One advantage with this method, however, is that it converges very quickly. Besides the above iteration, some other methods are outlined in References 15 , 23 and 27. We remark that if only a few eigenvalues and eigenvectors (say, five) are required, it may be more direct to use equation 56.

We now present an example of the ability of the single vector Lanczos algorithm to estimate the direction of arrival, or to find spectral lines in noise, and the advantages in extracting more than one eigenvalue and eigenvector in this process. We consider a signal with three sinusoids present in noise

$$x(n) = \sum_{i=1}^{3} A_i \cos(2\pi f_i n) + n(n) \tag{60}$$

where A_i are the amplitudes of sinusoids, f_i are the normalized spatial frequencies ($0 \le f \le 0.5$ corresponding to $0 \le \theta \le \frac{\pi}{2}$) , and $n(n)$ is the zero mean white noise with a variance of σ_n^2.

We have computed a 25 by 25 autocorrelation matrix of $x(n)$, $\mathbf{R}_{xx}$, by using 100 data samples. We have used the covariance method for this purpose, hence $\mathbf{R}_{xx}$ is real and symmetric. The eigenvectors $\mathbf{x}_j$ of $\mathbf{R}_{xx}$ corresponding to the lowest eigenvalues are computed by employing the single vector Lanczos algorithm. The power spectral density estimates are computed as follows:

$$S_{xx}^{(j)}(f) = \left| \frac{1}{\sum_{l=0}^{24} x_{jl} z^{-l}} \right|_{z = e^{j2\pi f}}^{2} \tag{61}$$

where x_{jl} are the elements of the jth eigenvector, $\mathbf{x}_j$.

Figure 6 and Figure 7 show the power spectral density (PSD) estimates of equation 60 with an SNR of 10 dB for $j = 1$ and 2, respectively.

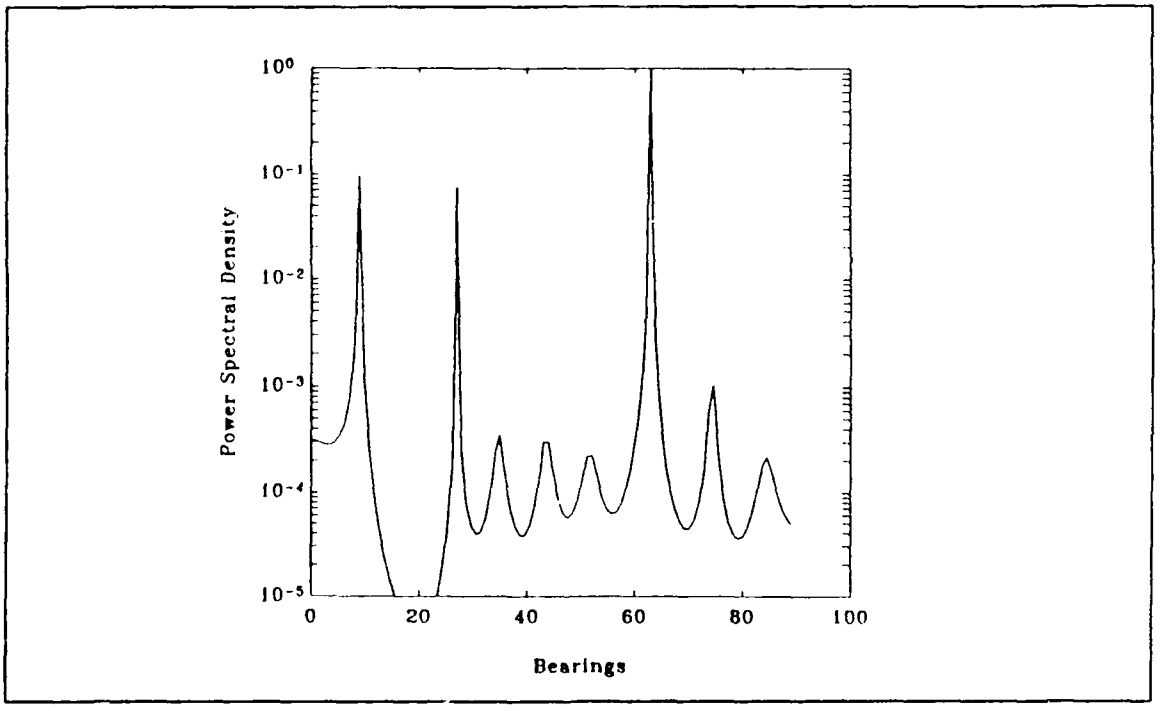


Figure 6. PSD of first eigenvector

Note that the index j indicates the increasing magnitude of the eigenvalues. Thus, $(\lambda_1, \mathbf{x}_1)$ are the lowest eigenvalue and the corresponding eigenvector. In both figures we have the peaks at the correct locations (9°, 27°, 63°). However, they both have spurious

smaller peaks at different locations. We can observe the same trend for the first five eigenvectors as shown in Figure 8, where the spectral estimates are over laid on each other.

Based on the above results one feels that we could employ some kind of averaging to get rid of the spurious peaks and improve the estimation performance. We have implemented two such methods: the eigenvector averaging and the spectral averaging. Figure 9 shows the result of the algebraic averaging of the first 5 eigenvectors, and Figure 10 shows the result of the algebraic averaging of the spectral estimates of the same eigenvectors. As seen from Figures 9 and 10, eigenvector averaging yields better results than spectral averaging.

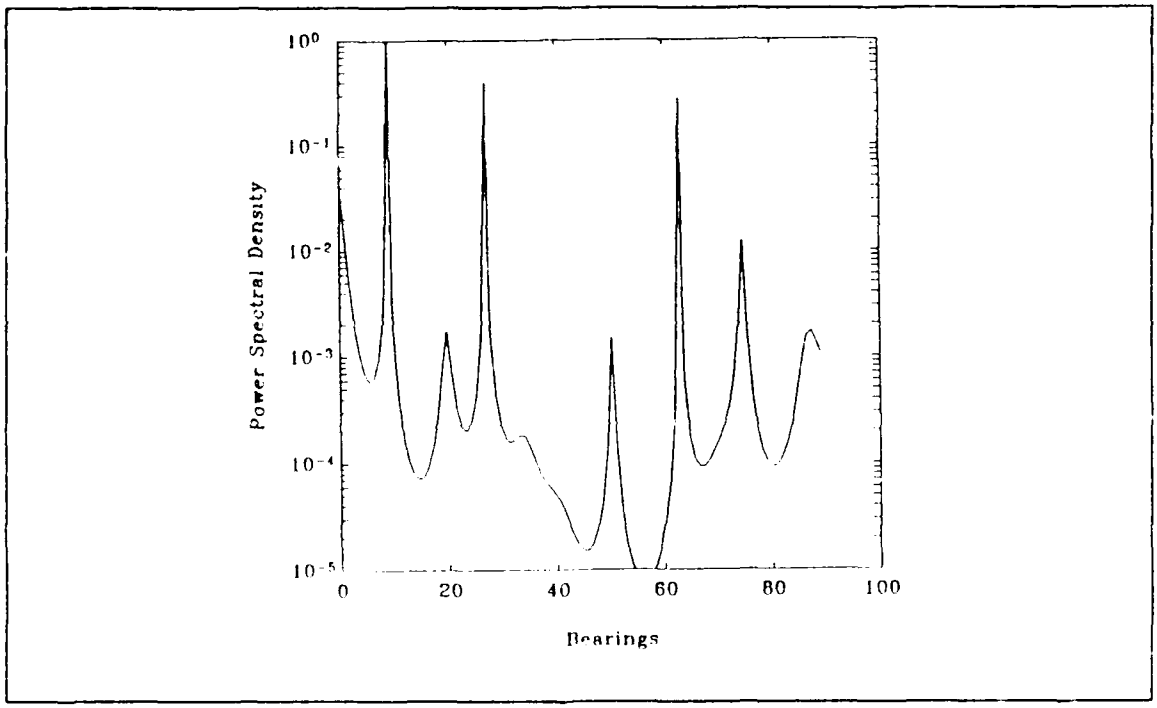


Figure 7. PSD of second eigenvector

Improved results, however, were obtained by using what is called spectral multiplication which is obtained by taking the product of the individual spectra, given by

$$S_{xx}(f) = \prod_{j=1}^{J} S_{xx}^{(j)}(f) \tag{62}$$

where J is a predetermined number ($J < m < n$). Figures 11 and 12 show the results of spectral multiplication for $J = 2$ and $J = 5$, respectively. As can be seen in these figures, using more spectra in equation 62 greatly improves the performance. Also, even for $J = 2$, spectral multiplication outperforms the eigenvector averaging method.

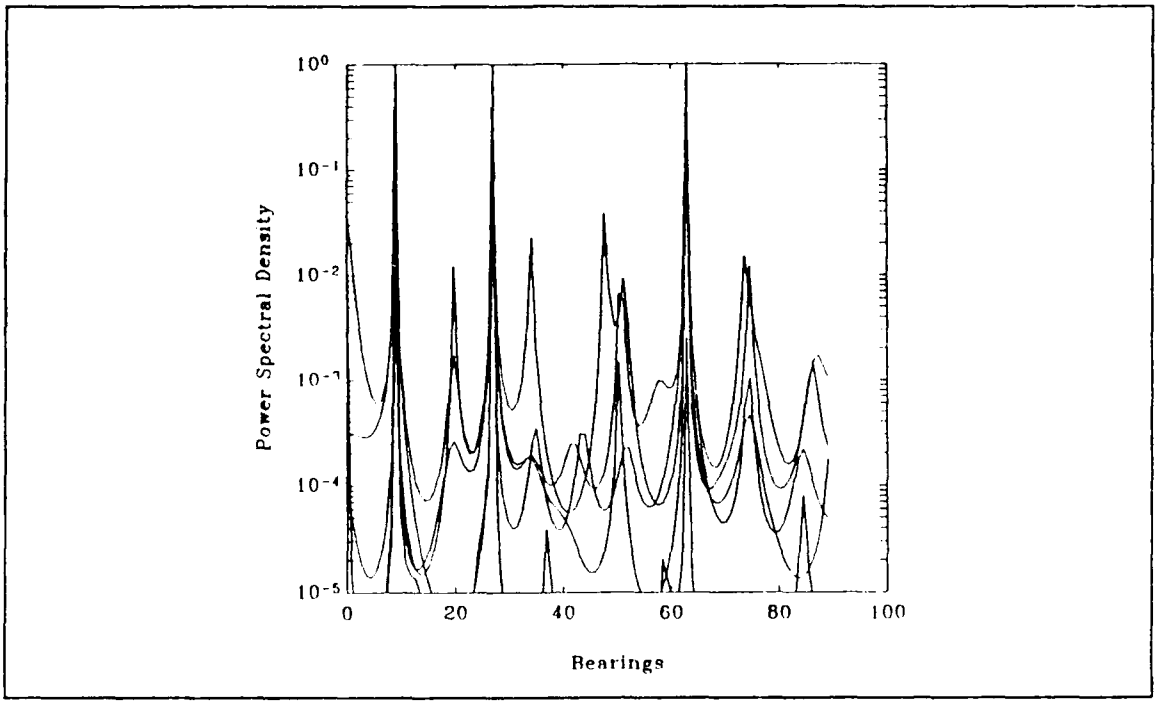


Figure 8. Overlayed PSDs of first 5 eigenvector

In the remainder of the thesis, we have used spectral multiplication in preference to the eigenvector or spectral averaging. Figure 13 shows the multiplication of 5 spectra for the case when the SNR = 0 dB. We notice a spurious peak around $\theta = 74°$. More spurious peaks are observed when the SNR is decreased to -5 dB (see Figure 14) and Figure 15 shows the spectrum for the SNR but we have used the eigenvectors 6-10 in this case. Improved performance is obtained as shown in Figure 16 (J = 10) and Figure 17 (J = 15) by using more and more eigenvectors in the spectral multiplication.

In all the above cases we always observed the signal spectral peaks at the right places. The spurious peaks, however, did not appear at the same location as we used a different eigenvector to compute the spectrum, $S_{xx}^{(i)}(f)$.

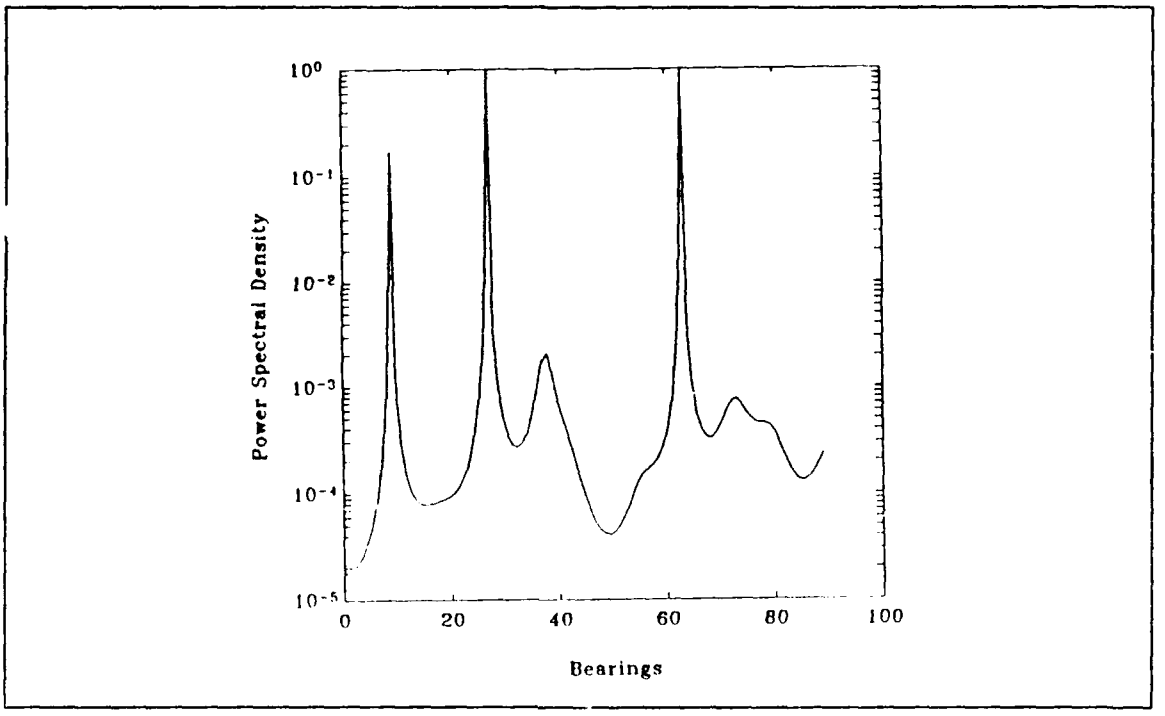


Figure 9. Eigenvector averaging

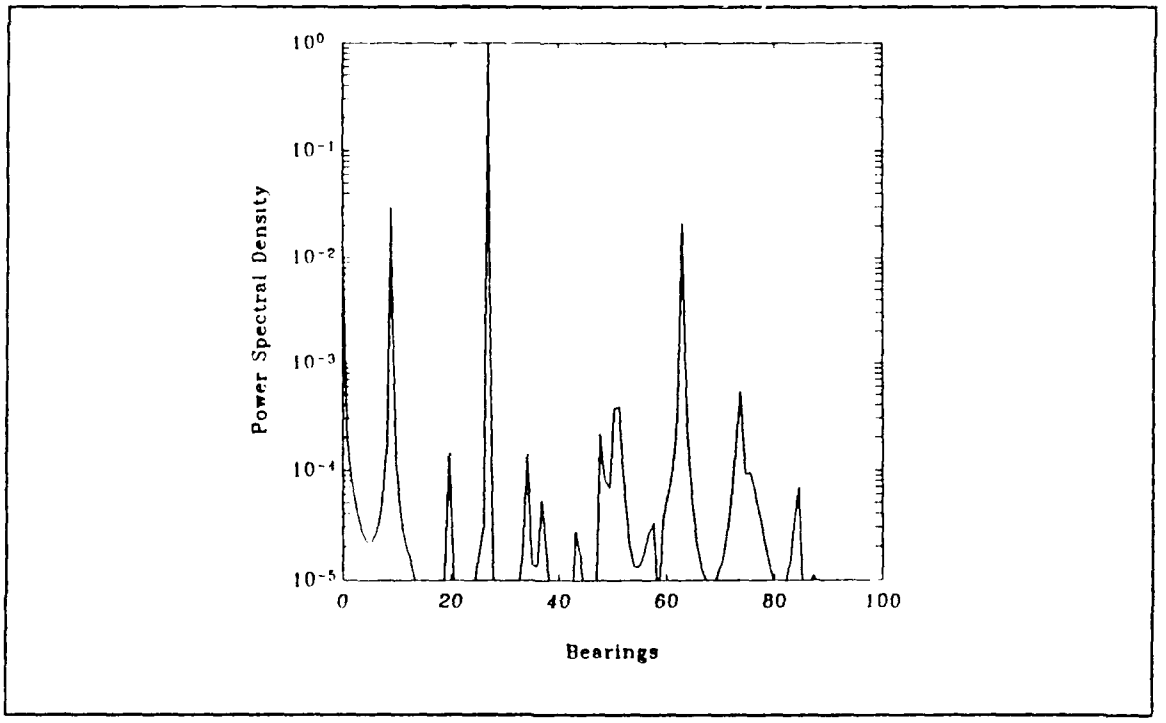


Figure 10. Spectral averaging

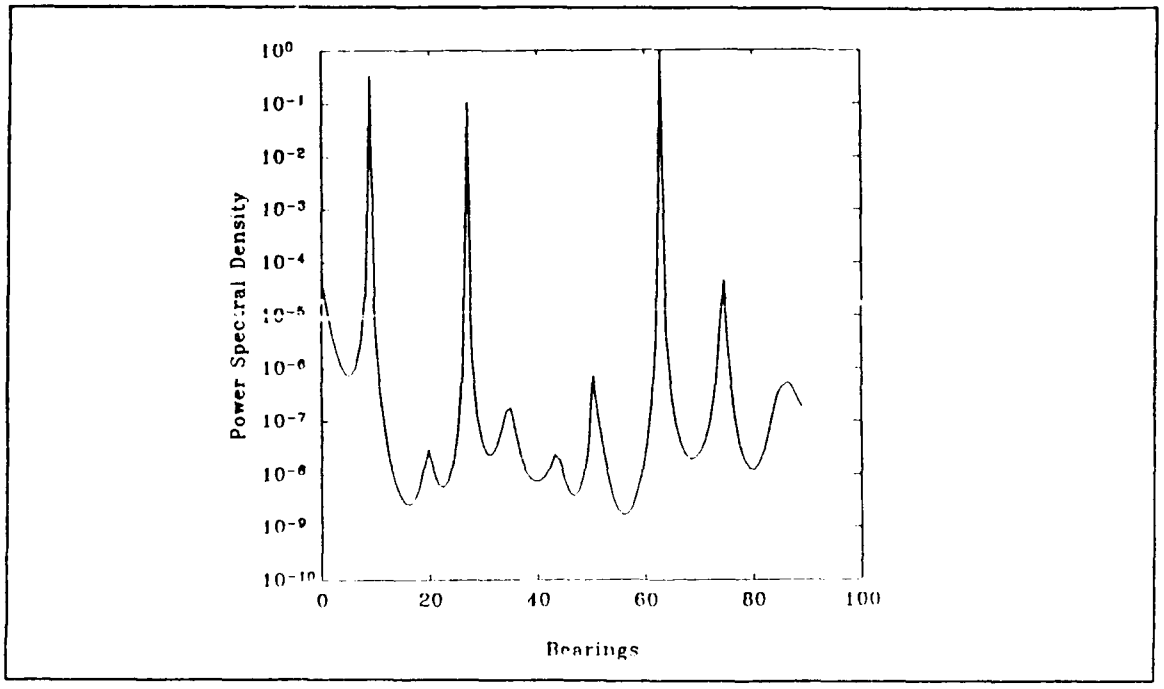


Figure 11. Spectral product for 2 eigenvectors

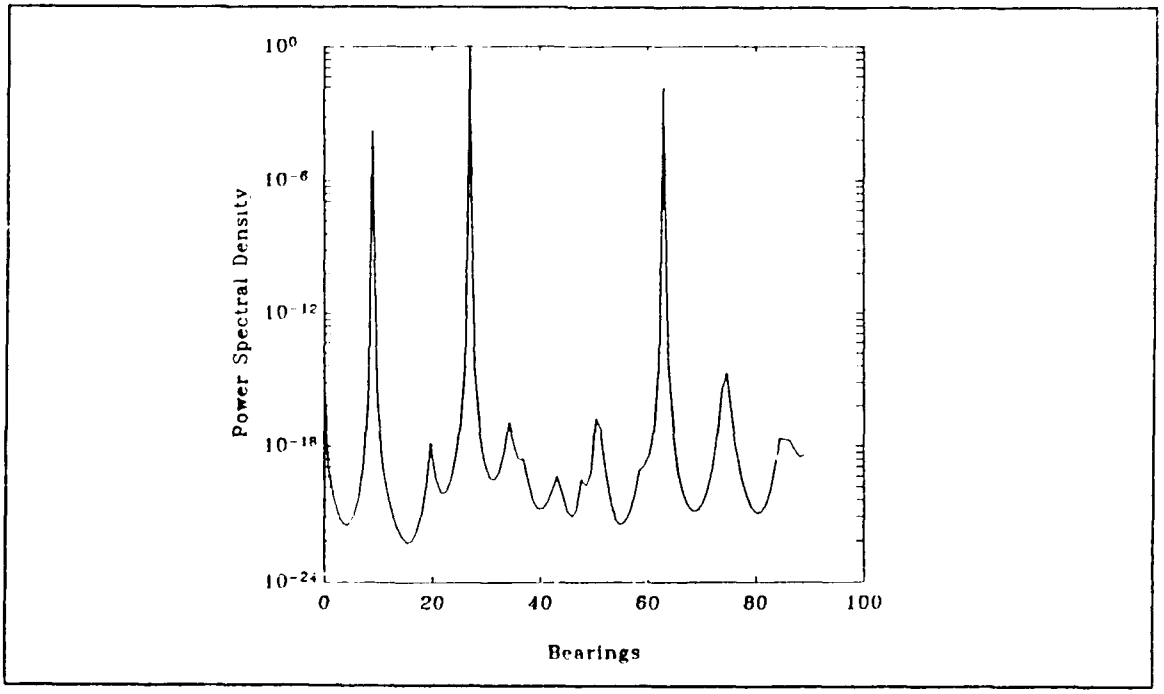


Figure 12. Spectral product for 5 eigenvectors

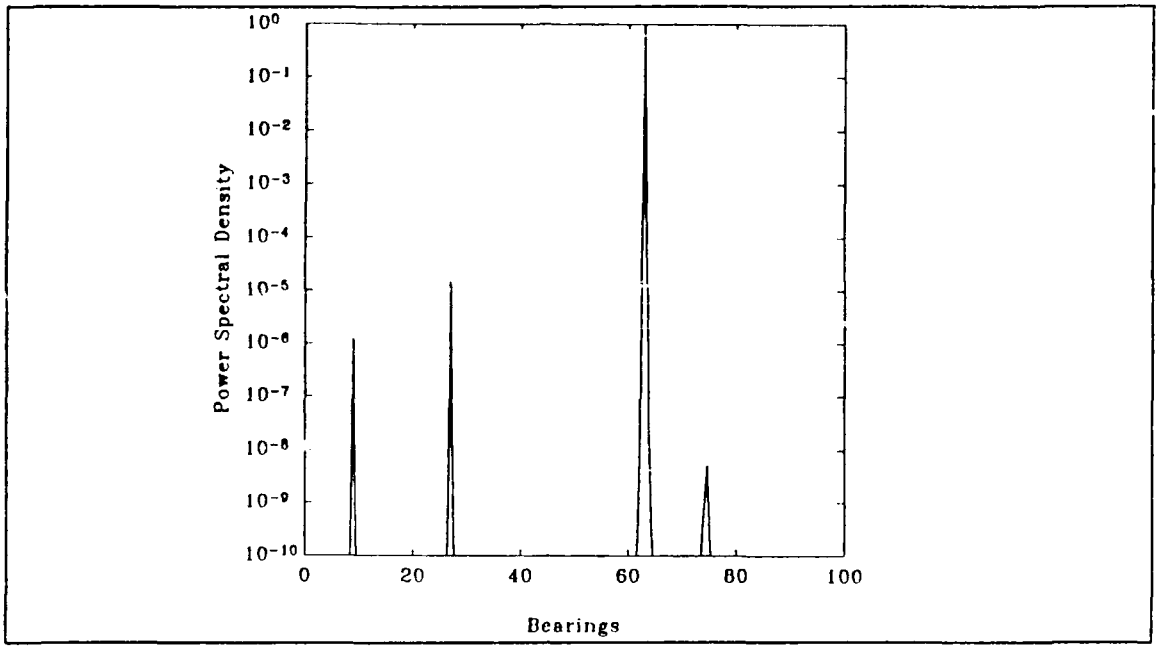


Figure 13. Spectral product for 5 eigenvectors, 0 dB

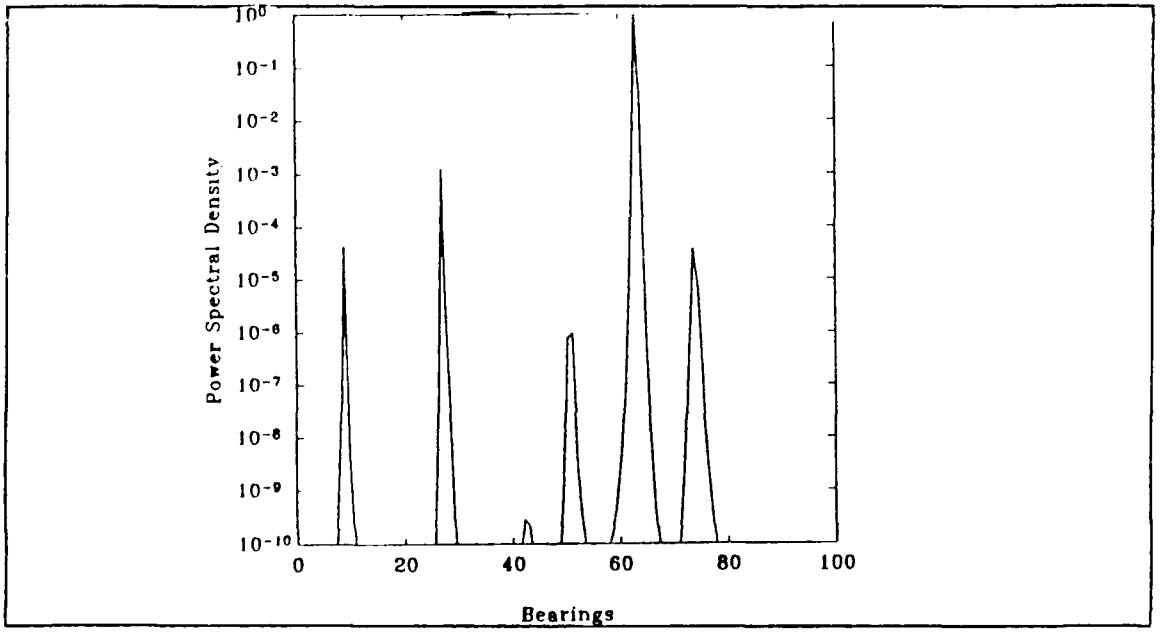


Figure 14. Spectral product for 5 eigenvectors, -5 dB: Using second through sixth eigenvectors

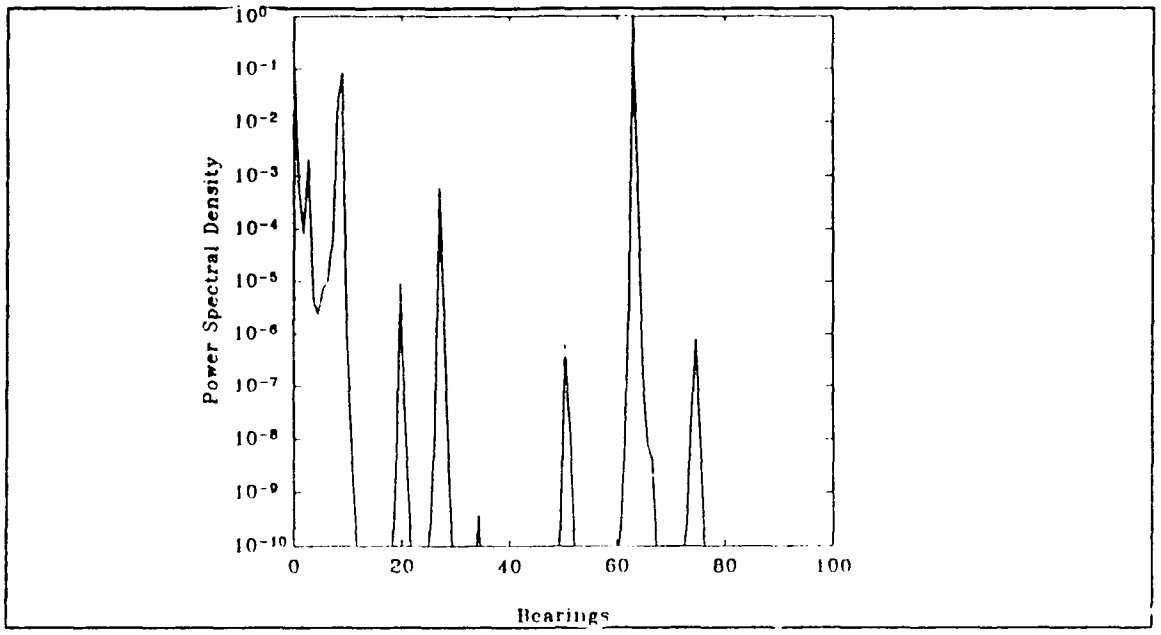


Figure 15. Spectral product for 5 eigenvectors, -5 dB: Using sixth through tenth eigenvectors

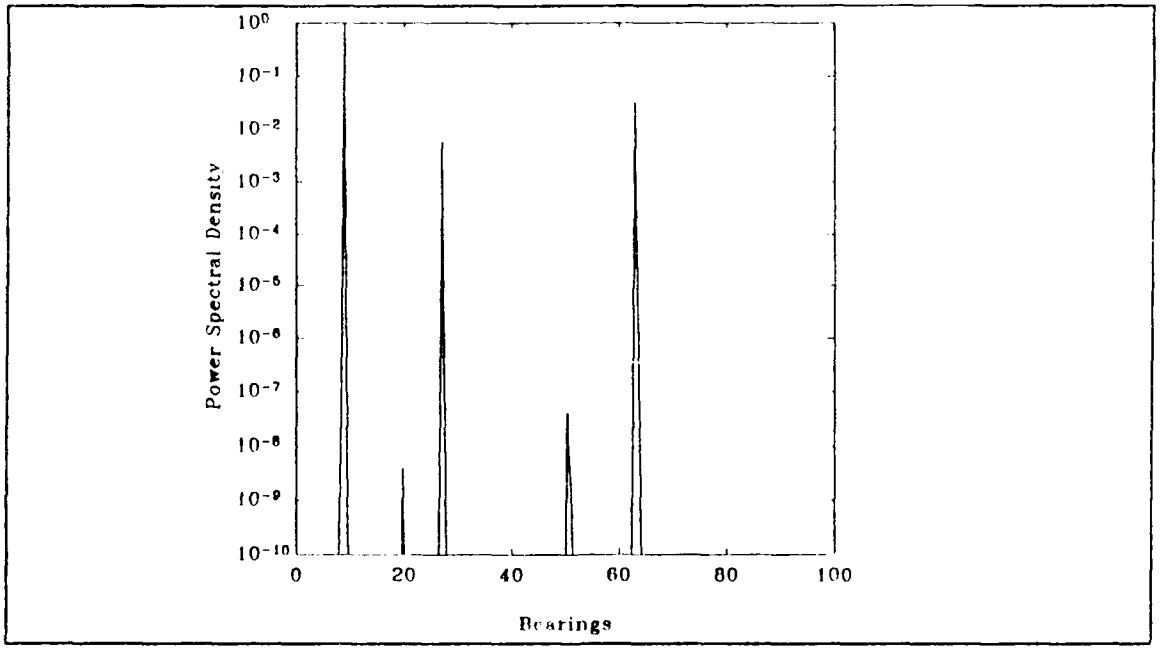


Figure 16. Spectral product for 10 eigenvectors, -5 dB

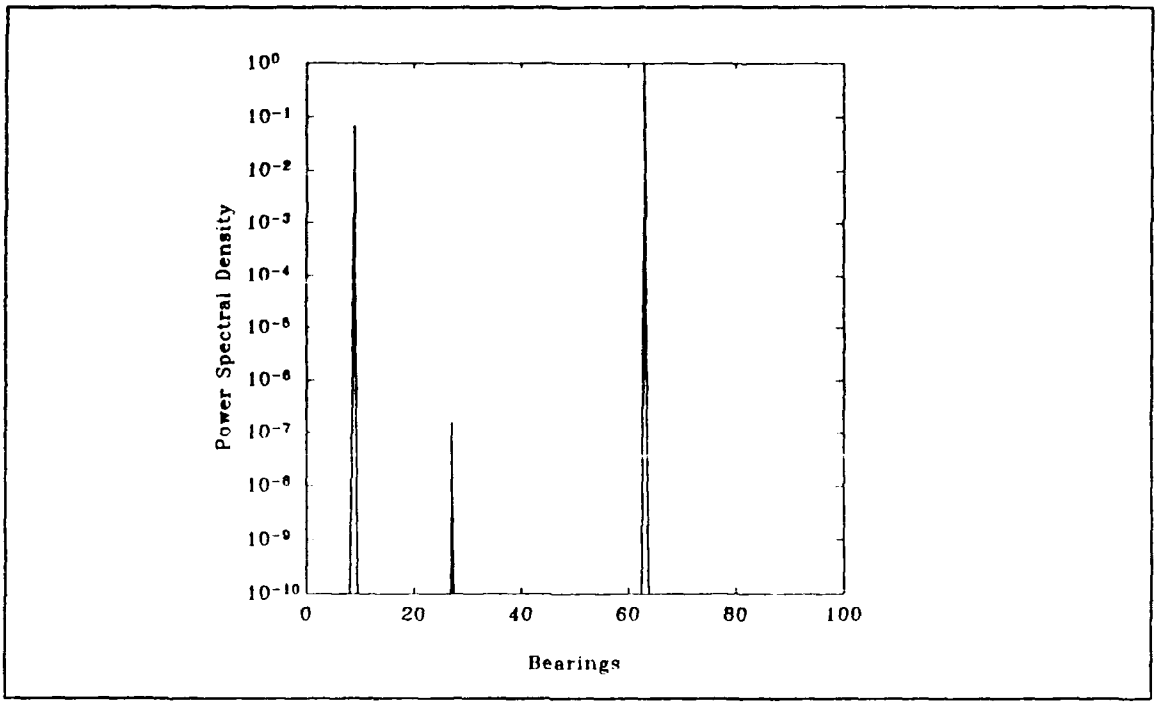


Figure 17. Spectral product for 15 eigenvectors, -5 dB

2. Other Methods

The single vector Lanczos algorithm will not determine that repeating eigenvalues exist, thus it cannot find the corresponding eigenvectors. The subspace that results has an incomplete basis as it is described only by the eigenvectors that are computed. The solution is to use a block method that is analogous to the single vector Lanczos algorithm. As we mentioned earlier, the block form of the Lanczos algorithm does find eigenvectors with multiplicity p as long as the blocks are dimensioned $l \geq p$. We attempted to incorporate the Cullum and Willoughby hybrid block Lanczos algorithm (Ref. 29 Chapter 8) into our direction of arrival model. We postulated that it would be desirable to compute a few of the extreme smallest repeating eigenvalues and their respective eigenvectors. However we were never able to get the program to reliably compute good eigenvalues and eigenvectors for the autocorrelation matrix. This has not posed a problem for our model as the covariance matrix does not appear to have repeating eigenvalues, but a larger order matrix may indeed include duplicating noise eigenvalues and require an algorithm that will accurately operate with that perturbation.

The algorithm we attempted to use is actually a hybrid approach to finding the eigenvalues and eigenvectors of a real symmetric matrix **A**. For insight into the problem look at the block analogy of equations 40, 41, and 42. Define matrices $\mathbf{B}_1 \equiv \mathbf{0}$ and $\mathbf{V}_0 \equiv \mathbf{0}$. The n by q matrix $\mathbf{V}_1$ has columns that are orthonormal random vectors. The value of q must be greater than or equal to the number of eigenvalues to be found.

$$\begin{aligned} &\text{for } i = 2, 3, \ldots, s \\ &\mathbf{V}_{i+1}\mathbf{B}_{i+1} = \mathbf{P}_i \equiv \mathbf{A}\mathbf{V}_i - \mathbf{V}_i\mathbf{M}_i - \mathbf{V}_{i-1}\mathbf{B}_i^T \end{aligned} \tag{63}$$

$$\mathbf{M}_i \equiv \mathbf{V}_i^T(\mathbf{A}\mathbf{V}_i - \mathbf{V}_{i-1}\mathbf{B}_i^T) \tag{64}$$

$$\mathbf{V}_{i+1}\mathbf{B}_{i+1} \equiv \mathbf{P}_i \tag{65}$$

The matrix $\mathbf{B}_{i+1}$ is a modified Gram-Schmidt orthonormalization of the columns of $\mathbf{P}_i$. Also note that the matrix $\mathbf{M}_i$ correspond to the α's of the single vector Lanczos.

The block analogy to the Krylov subspace approach can be performed with

$$K^s(\mathbf{A}, \mathbf{V}_1) = span\left\{\mathbf{V}_1, \mathbf{A}\mathbf{V}_1, \mathbf{A}^2\mathbf{V}_1, \ldots, \mathbf{A}^{s-1}\mathbf{V}\right\} \tag{66}$$

The blocks $\mathbf{V}_j$, for $j = 1, 2, \ldots, s$ form the orthonormal basis of the Krylov subspace.

It can be shown that for a symmetric n by n matrix **A** and an orthonormal n by q starting matrix $\mathbf{V}_1$, that the block recursion equations 63, 64, and 65 will generate blocks $\mathbf{V}_2, \mathbf{V}_3, \ldots, \mathbf{V}_s$ where $qs < n$. It is these blocks that form an orthonormal basis of the subspace $K^s(\mathbf{A}, \mathbf{V}_1)$. In much the same way as the single vector Lanczos algorithm generates the tridiagonal Lanczos matrix, the block variant generates *blocks*, but these are now nontridiagonal. At the end of each iteration the Lanczos matrix is of the form

$$
\mathbf{T}_s \equiv \left[\begin{array}{c|cccccccc}
\mathbf{A}_1 & b_2^T & & & & & \tilde{\mathbf{M}}_1^T\mathbf{A}\overline{\mathbf{M}} & & \\
\hline
b_2 & \alpha_2 & \beta_3 & & & & & & \\
 & \beta_3 & \alpha_3 & \beta_4 & & & & & \\
\overline{\mathbf{M}}^T\mathbf{A}\tilde{\mathbf{M}}_1 & & \beta_4 & \alpha_4 & \cdot & & & & \\
 & & & \cdot & \cdot & \cdot & & & \\
 & & & & \cdot & \cdot & \cdot & & \\
 & & & & & \cdot & \cdot & \beta_t & \\
 & & & & & & \beta_t & \alpha_t &
\end{array}\right] \tag{67}
$$

The submatrix $\tilde{\mathbf{M}}_1^T\mathbf{A}\overline{\mathbf{M}}$ consists of the reorthogonalized terms and $\tilde{\mathbf{M}}_1$ is the portion of the first block that is not generating descendants. Ritz vectors are computed on every iteration and are used as the starting blocks for the next iteration. Each block is required to be reorthogonalized with respect to the all the vectors in first block which is not being allowed to generate descendants. It is apparent that the block procedure requires a great amount of storage and is very computationally intensive.

IV. RESULTS

Using the Lanczos algorithm it is possible to find some of the eigenvalues and eigenvectors of a matrix without going through an entire matrix decomposition. The smallest eigenvalue of the autocorrelation matrix and its corresponding eigenvector will have the required spectral information to determine a source's bearing (direction of arrival) from an array. Multiplying several of the resultant eigenvectors' power spectral densities will tend to reinforce the true spectral peak and zero out spurious peaks that do not occur with every eigenvector.

The problem with finding the split between the noise and signal eigenvalues disappears as only a few of the smallest eigenvalues of a large matrix (in relation to the number of sources) are used.

A. APPROACH

The received signal is modeled by sinusoids at normalized spatial frequencies proportional to their bearings from endfire (.0 = 0°, .5 = 90°). The sum of these sinusoids is sampled at a rate based on the interelement spacing of $\lambda_{min}/2$. Thus a source at endfire is sampled at the Nyquist rate and the sample rate increases as the bearing shifts toward the array broadside. The simulation uses

$$ss(n) = \sum_{i=1}^{T} A\cos(2\pi f_i n) + n(n) \tag{68}$$

where $ss(n)$ is the instaneous excitation for the sensor at location n, A is the amplitude of each of the T signals, f_i is the normalized spatial frequency of the ith source (dependent upon bearing), and $n(n)$ is white gaussian noise. The relationship between A and the noise variance σ_v^2 is determined by the desired signal to noise ratio (SNR), where

$$SNR = 10\log\left(\frac{\frac{A^2}{2}}{\sigma_v^2}\right) \tag{69}$$

The experiment consists of simulating a linear array with equally spaced sensors receiving signals of known temporal frequency from various bearings. One possible

physical implementation would place a bank of bandpass filters on each sensor with the outputs from each similar filter tied into a correlator. Advantages of this method include the processing gain found by prefiltering the noise and simple parallel implementation with separate channels for each frequency band. The lowering of the noise bandwidth will raise the SNR at the correlator. As more filters are used (smaller bandwidth) the noise power decreases and the SNR is increased. The algorithm creates an autocorrelation matrix with the output of the correlator. The Lanczos tridiagonalization and eigendecomposition provides the eigenvectors that are estimates of the spatial PSD. The PSD corresponds to the sources directions of arrival. A possible implementation is shown in Figure 18.

B. EXPERIMENT SET UP

The first three cases show the effect of different signal strengths on the ability to accurately determine the number of targets and the bearing resolution for various directions and target spacing. In each of these cases, the number of sensors is 100, a matrix size of 25 is used and 15 iterations (the number of eignevalues eigenvectors found) are performed. Case 4 uses three 5 dB sources at 18°, 36° and 41.4° to illustrate the effects on changing the number of sensors (samples), the size of the autocorrelation matrix, and the number of eigenvectors used. The noise is randomly generated white gaussian noise with a standard deviation selected to provide the desired SNR. Each figure shows, (a) the PSD of selected eigenvectors overlayed and plotted versus bearing and (b) the product of selected PSDs of those eigenvectors.

Case 1 is with all sources at a signal strength of 5 dB. Figure 19 shows results from the second through sixth eigenvectors of a single source at 18°. Note that some of the eigenvectors have individual peaks as high as the true signal peak, but only at the true bearing do all have a common peak. Figure 20 illustrates the other end of the spectrum, at 81°. Once again the second through sixth eigenvectors are overlayed to show that the correct bearing is consistently displayed, but in this case one eigenvector has an individual peak higher than the true signal peak. The product of these PSDs provides sufficient resolution. Figure 21 has two closely spaced sources at 36° and 38°. Resolution is achieved but the PSD product of the second through sixth eigenvectors shows a spurious peak near 75°. Figure 22 is from three sources at 0°, 36° and 88.2°. The individual eigenvector PSDs clearly show the excellent performance at broadside.

Case 2 lowers the signal strength of all sources to 0 dB. Figure 23 shows results from the second through sixth eigenvectors of a single source at 18°. Many more peaks

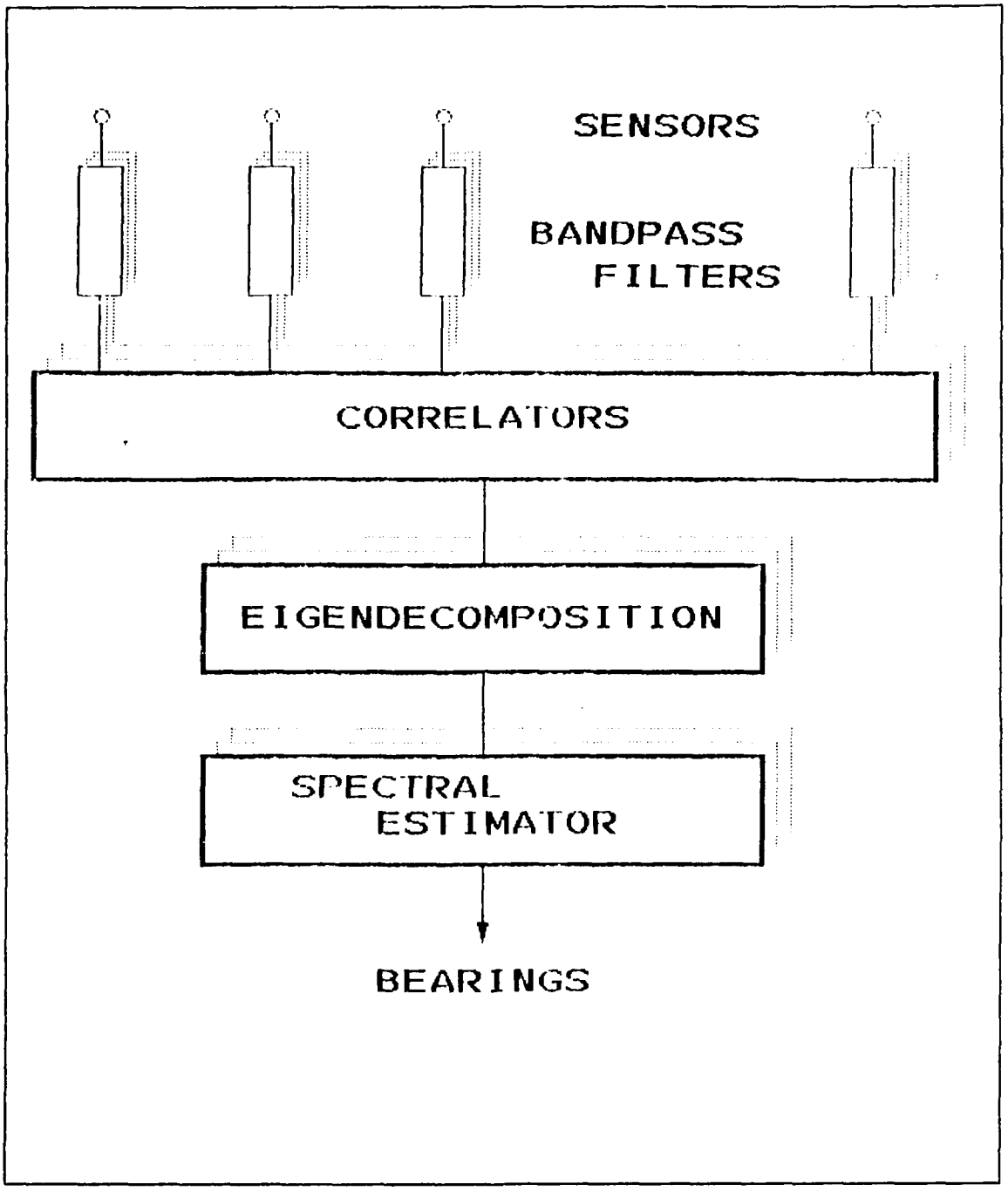


Figure 18. A physical implementation

are visible in the PSD product, making the decision of how many targets more difficult. Figure 24 shows that at the the other end of the spectrum (at 81°) the situation is

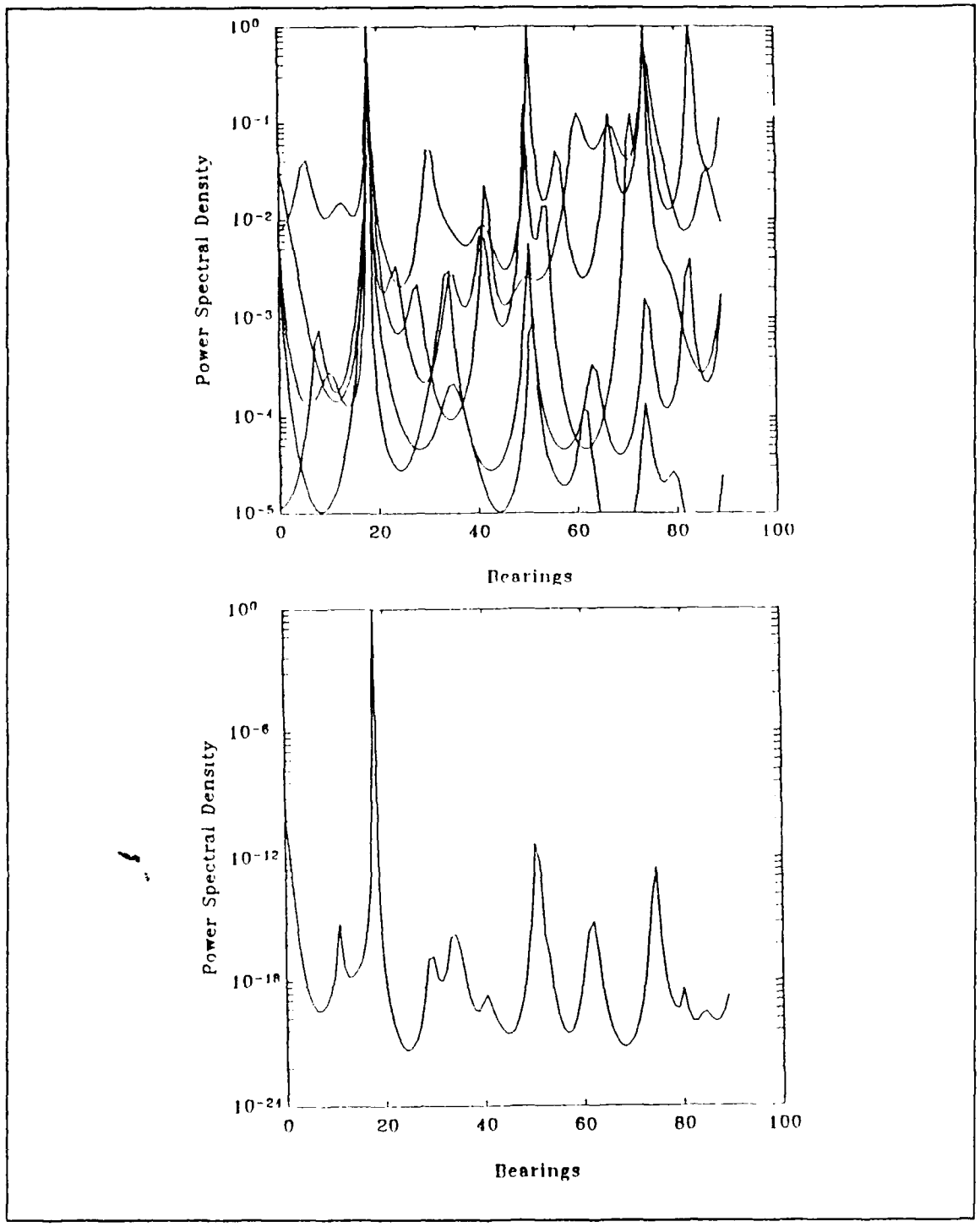


Figure 19. Case 1 5 dB, 1 target at 18 °: (a) overlay of PSDs from second through sixth eigenvectors, (b) product of the above PSDs

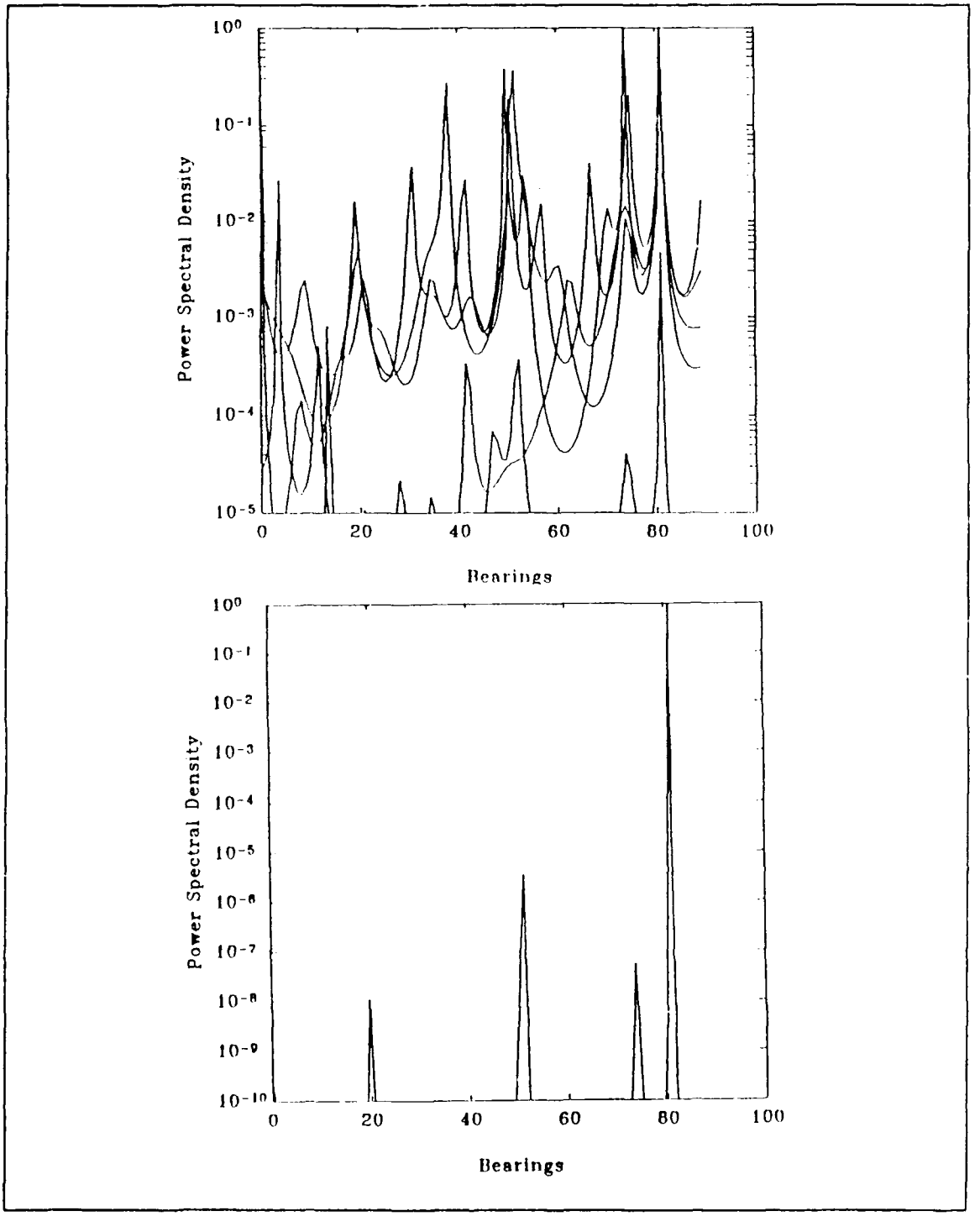


Figure 20. Case 1 5 dB, 1 target at 81 °: (a) overlay of PSDs from second through sixth eigenvectors, (b) product of the above PSDs

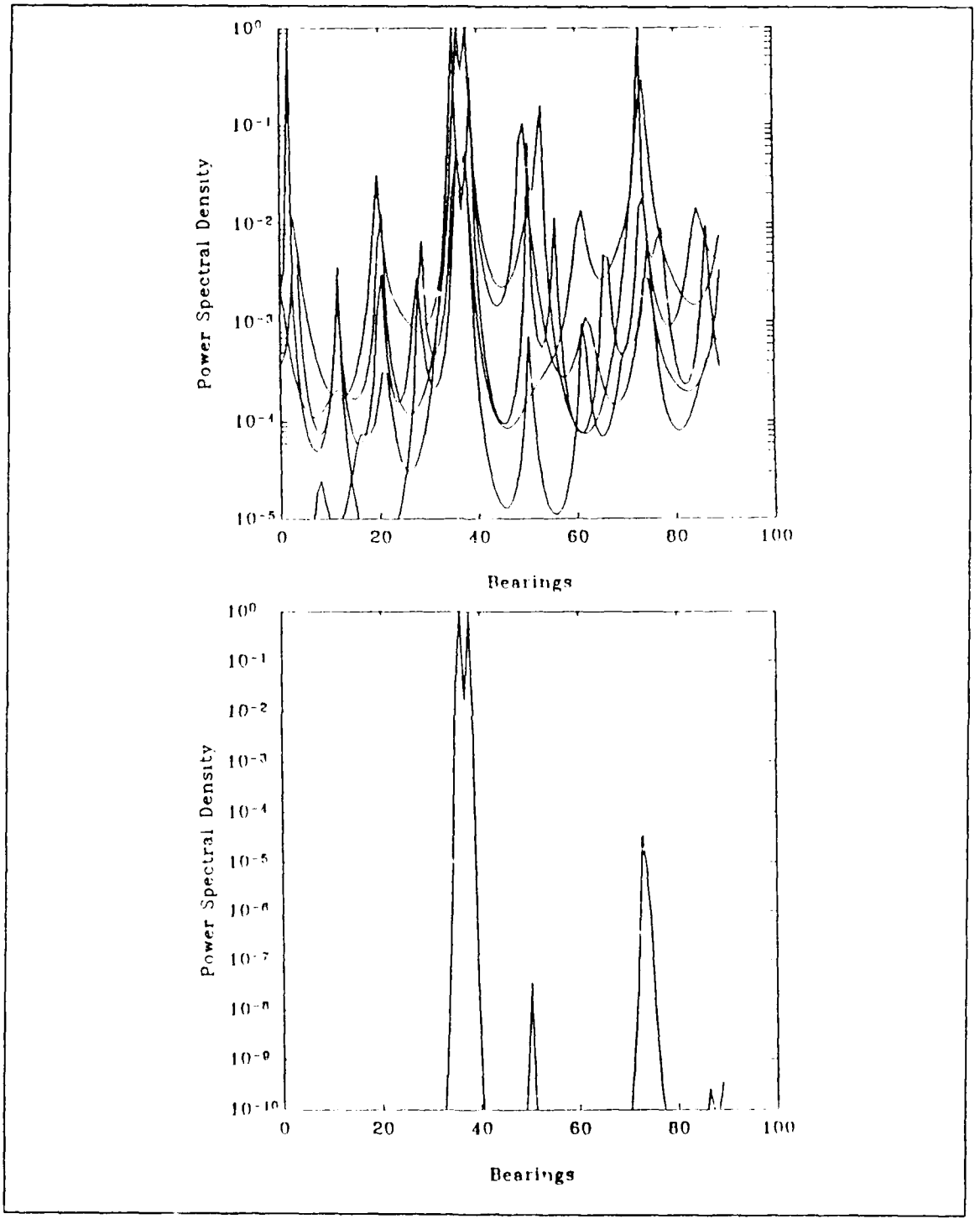


Figure 21. Case 1 5 dB, 2 targets at 36 ° and 38 °: (a) overlay of PSDs from third through seventh eigenvectors, (b) product of the above PSDs

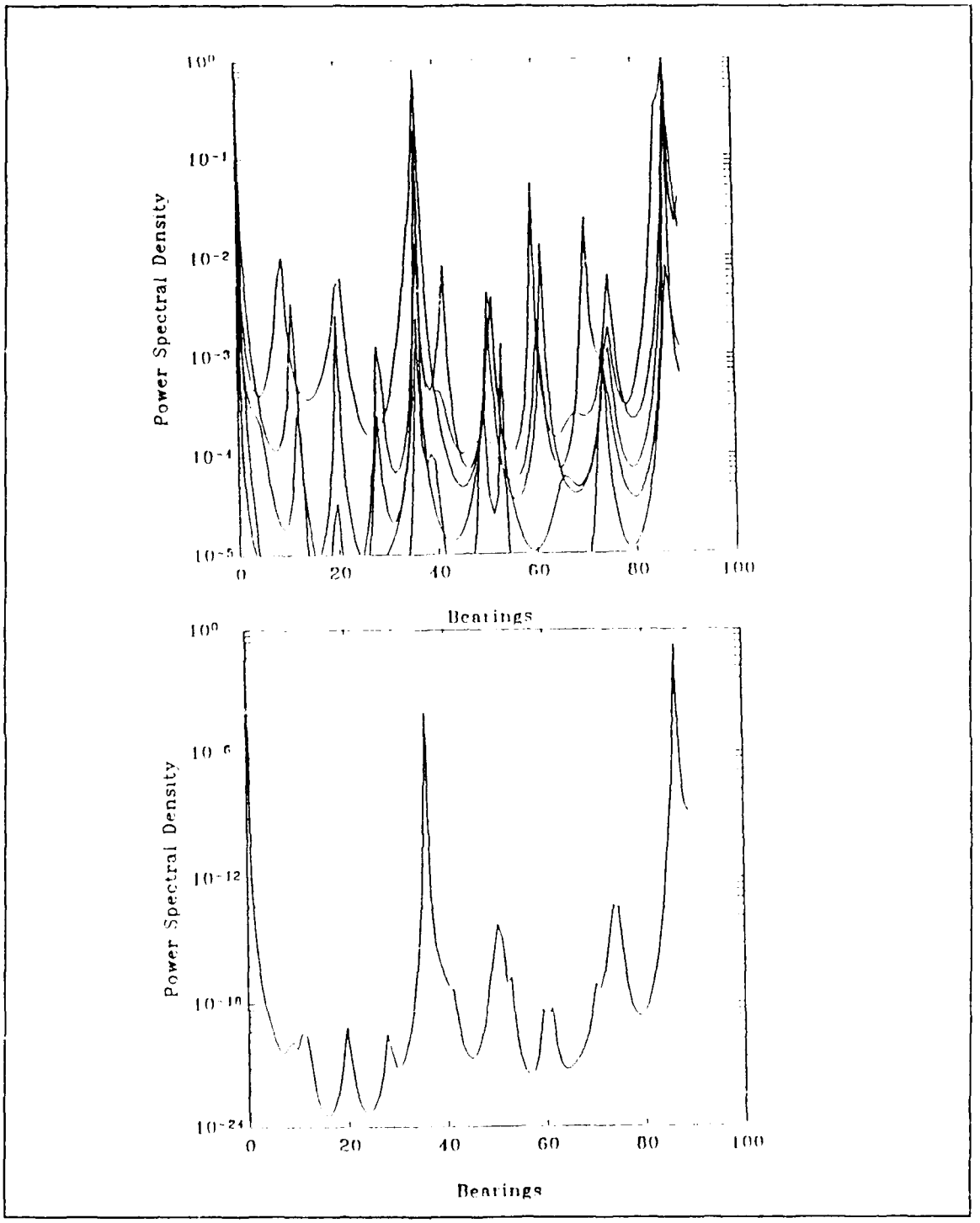


Figure 22. Case 1 5 dB, 3 targets at 0 °, 36 ° and 88.2 °: (a) overlay of PSDs from second through sixth eigenvectors, (b) product of the above PSDs

slightly worse (due to a lower sampling rate). The overlays of the second through sixth eigenvectors show that the correct bearing is consistently displayed, but in this case enough spurious peaks reinforce one another, resulting in the PSD product that has not zeroed out the bad peaks. Figure 25 illustrates that the proper choice of eigenvectors will resolve this problem. Here the PSDs of the first through fifth eigenvectors are used, giving a product that is easier to determine correctly. Figure 26 shows the 0 dB case for two close spaced sources at 36° and 38°. Resolution is achieved but the PSD product of the first through fifth eigenvectors shows several spurious peaks, including the same one as in Case 1 near 75°. Figure 27 is the three source example at 0 dB. The individual eigenvector PSDs are repeating at the proper bearings but the performance at broadside is resulting in the product at the other bearings actually being driven down.

A signal strength of -5 dB is used for Case 3. Figure 28 shows results from the first 10 eigenvectors of a single source at 18°. Using more good eigenvectors increases the likelihood that all spurious peaks will be diminished. Figure 29 illustrates results at the other end of the spectrum (at 81°). Resolution is looked at in Figure 30. At -5 dB the algorithm cannot separate the two close spaced sources at 36° and 38°. A number of spurious peaks are higher than the bump at 36° making it impossible to accurately determine the number of sources as well as both locations. Resolution is tried again in Figure 31 with 2 sources at 36° and 40°. Using five eigenvector PSD products produces good results. Figure 32 shows 3 sources at -5 dB. Good performance is seen both in the overlays and in the PSD product.

Case 4 starts with 100 sensors and a 25 by 25 covariance matrix shown in Figure 33. As the number of sensors is decreased and the number of eigenvectors used is held constant, more spurious peaks start to occur (Figure 34 and Figure 35). Figure 36 shows the spectral improvement as more eigenvectors are used. As the number of sensors is decreased to 40 (Figure 37) and seven eigenvectors are used, the results are still acceptable. Using only 3[illegible] sensors we can no longer see resolve between the two closely spaced sources. With a sufficient number of eigenvectors no spurious peaks are present but the true number of targets is nondeterminable (Figure 38 and Figure 39).

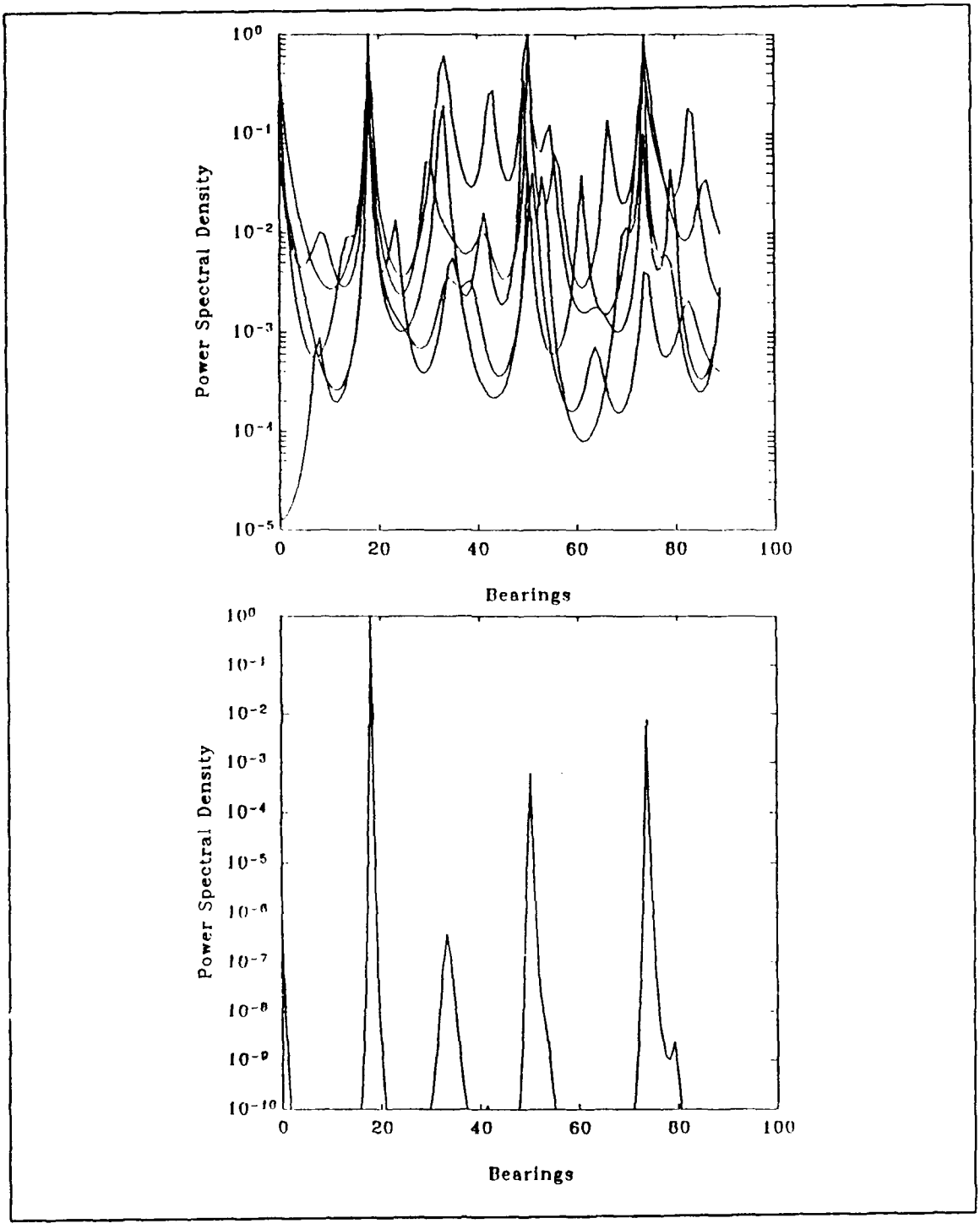


Figure 23. Case 2 0 dB, 1 target at 18 °: (a) overlay of PSDs from second through sixth eigenvectors, (b) product of the above PSDs

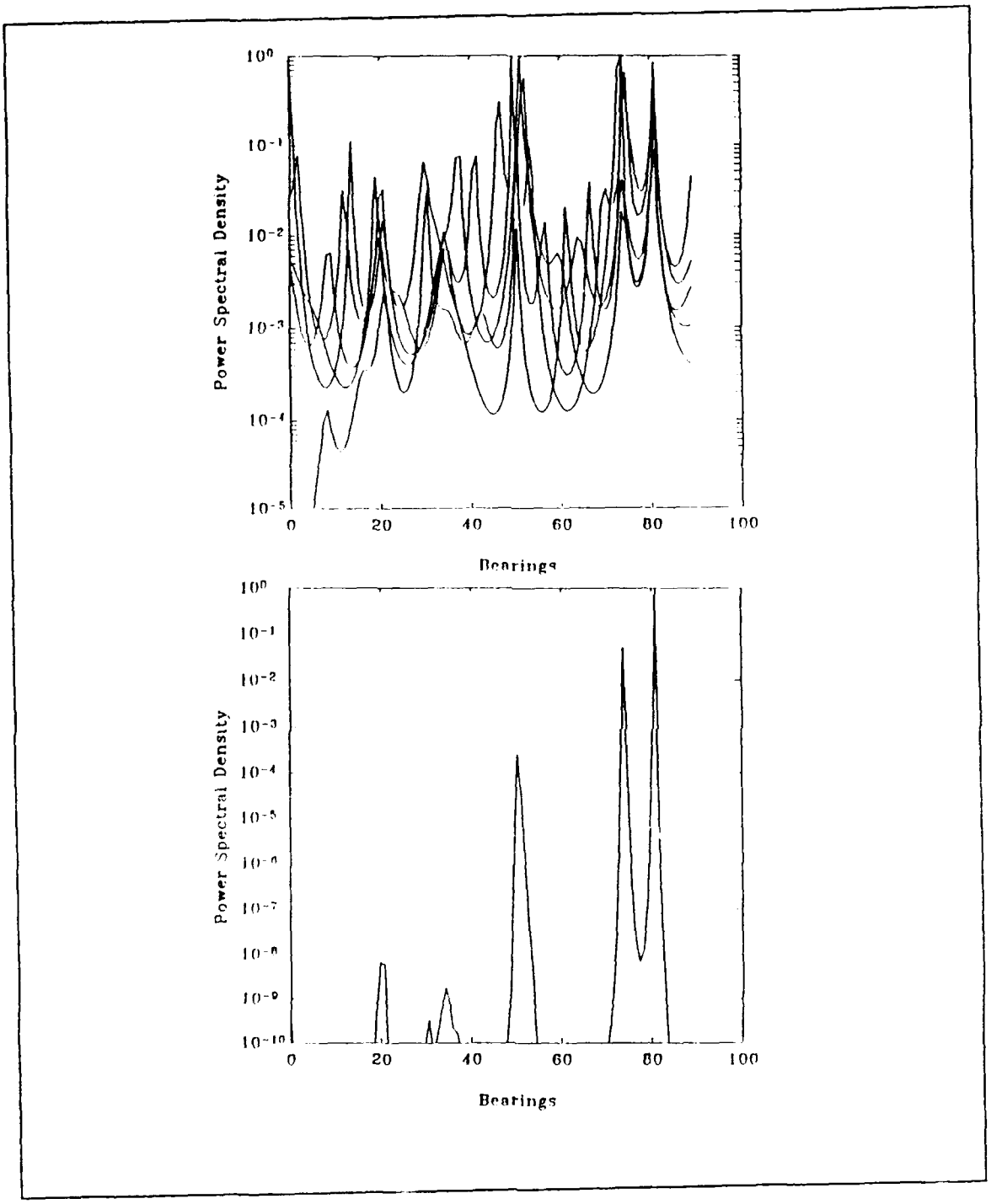


Figure 24. Case 2 0 dB, 1 target at 81 °: (a) overlay of PSDs from first through fifth eigenvectors, (b) product of the second through sixth eigenvector PSDs

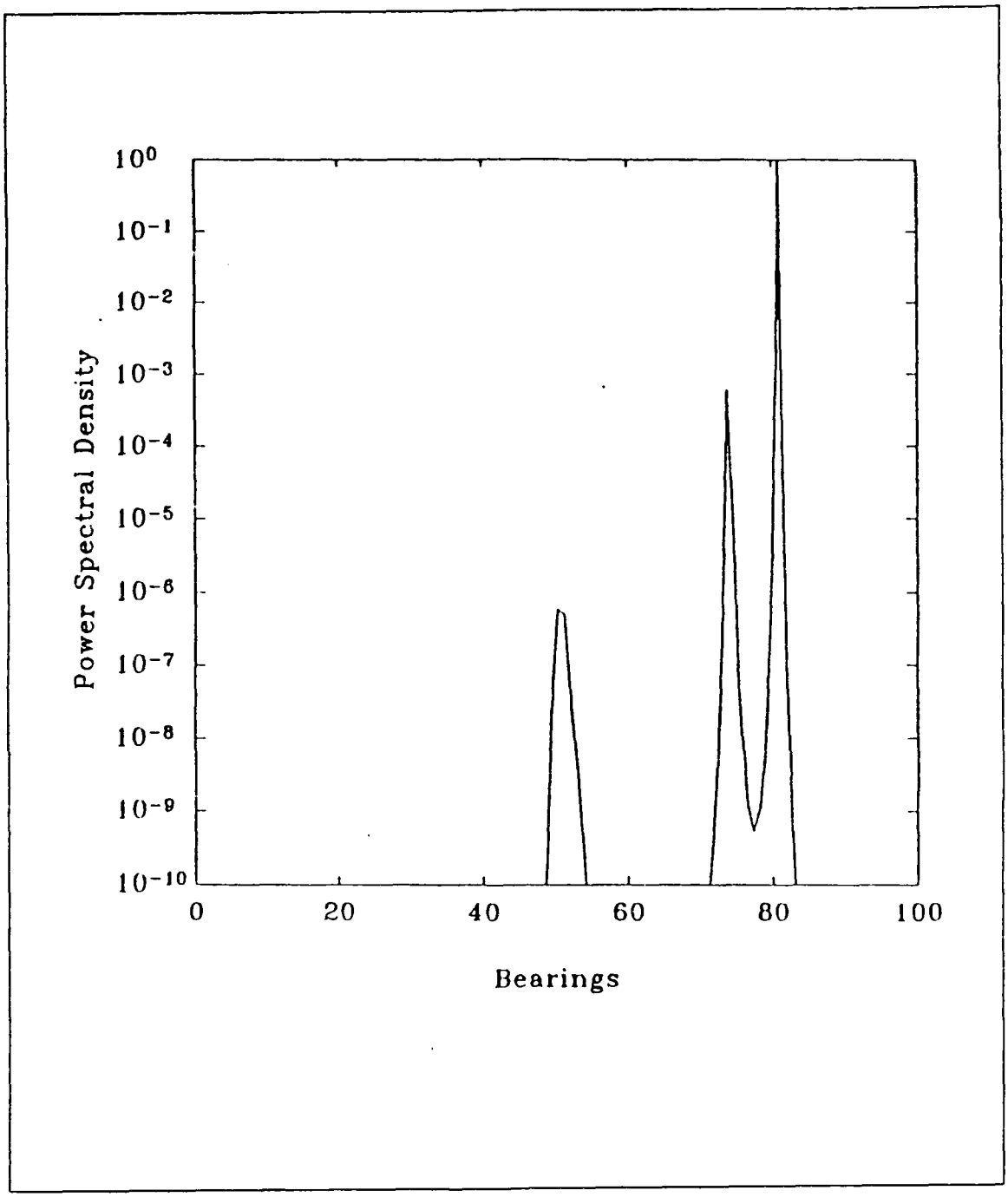


Figure 25. Case 2 0 dB, 1 target at 81 °: product of the first through fifth eigenvector PSDs

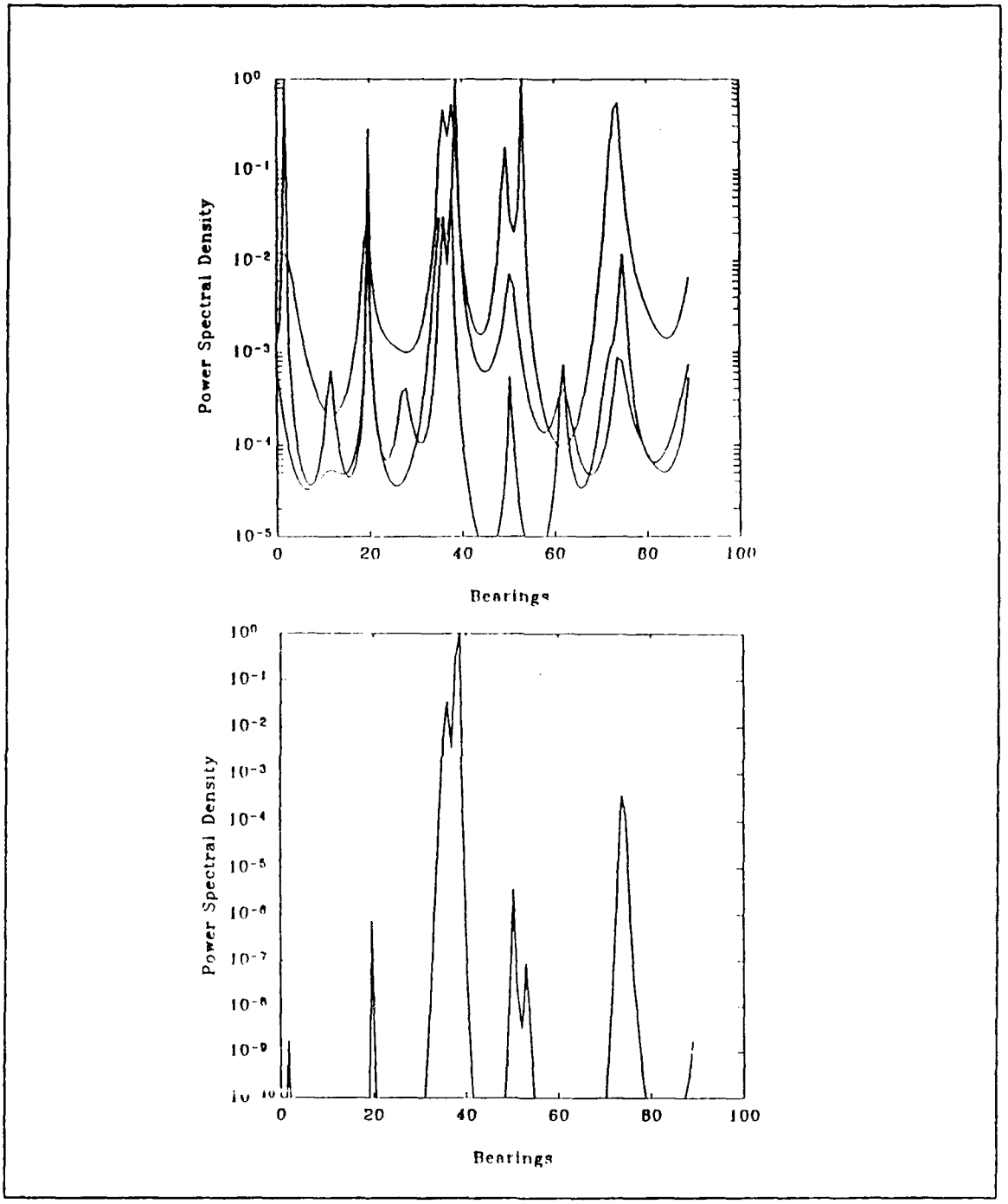


Figure 26. Case 2 0 dB, 2 targets at 36 ° and 38 °: (a) overlay of PSDs from first through third eigenvectors, (b) product of the first through fifth eigenvector PSDs

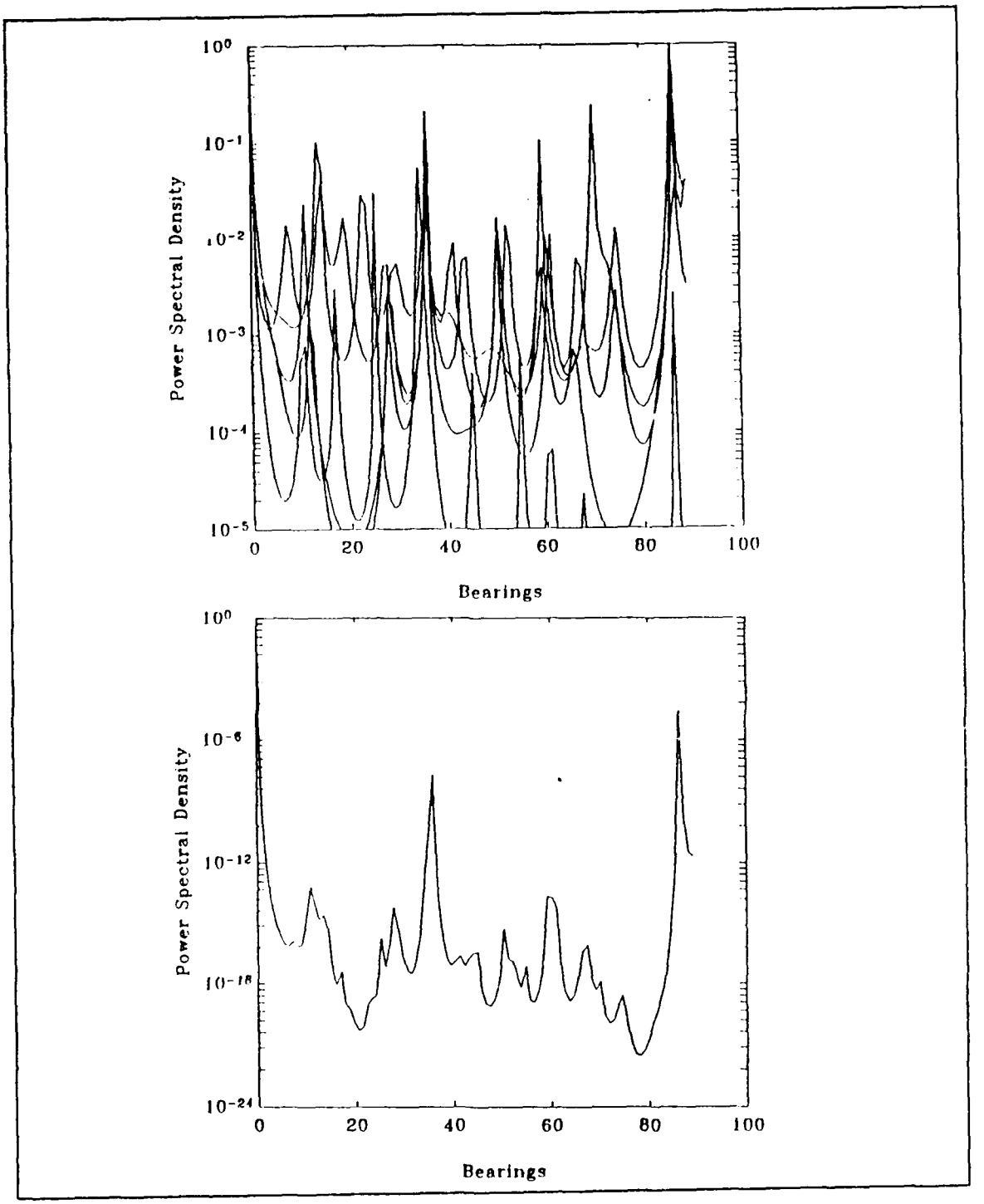


Figure 27. Case 2 0 dB, 3 targets at 0 °, 36 ° and 88.2 °: (a) overlay of PSDs from second through sixth eigenvectors, (b) product of the above PSDs

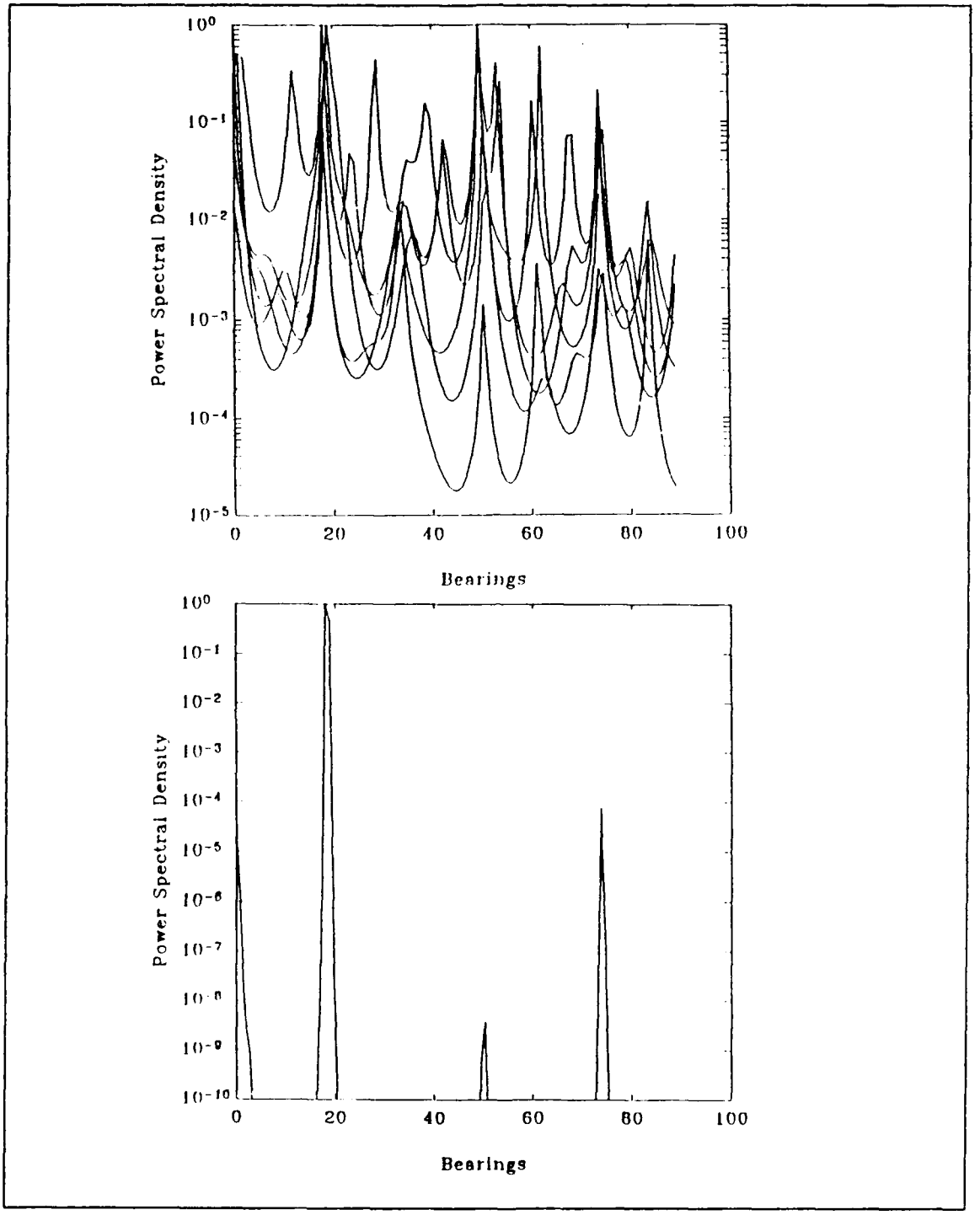


Figure 28. Case 3 -5 dB, 1 target at 18 °: (a) overlay of PSDs from first through sixth eigenvectors, (b) product of the first 10 eigenvector PSDs

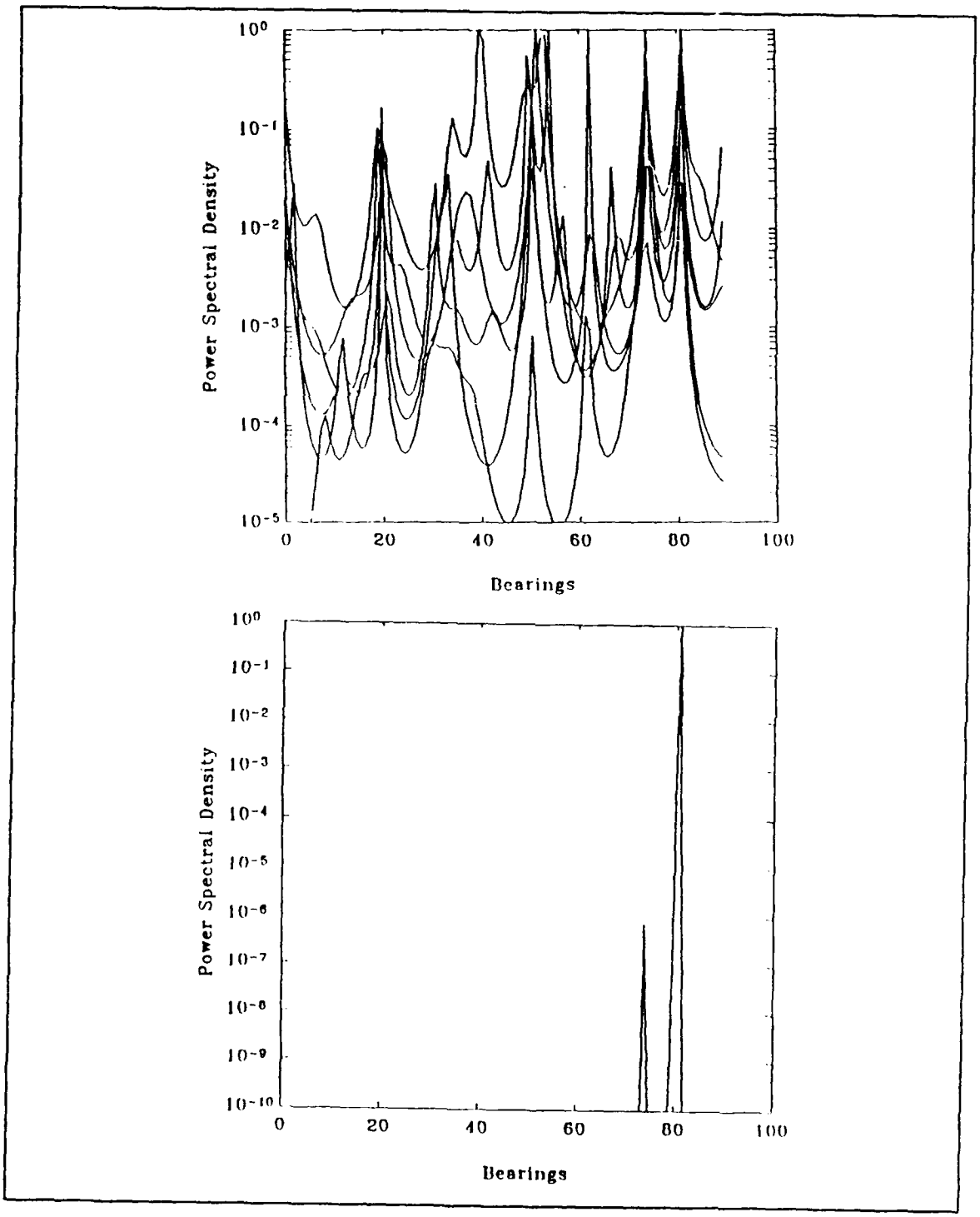


Figure 29. Case 3 -5 dB, 1 target at 81 °: (a) overlay of PSDs from first through sixth eigenvectors, (b) product of the first 10 eigenvector PSDs

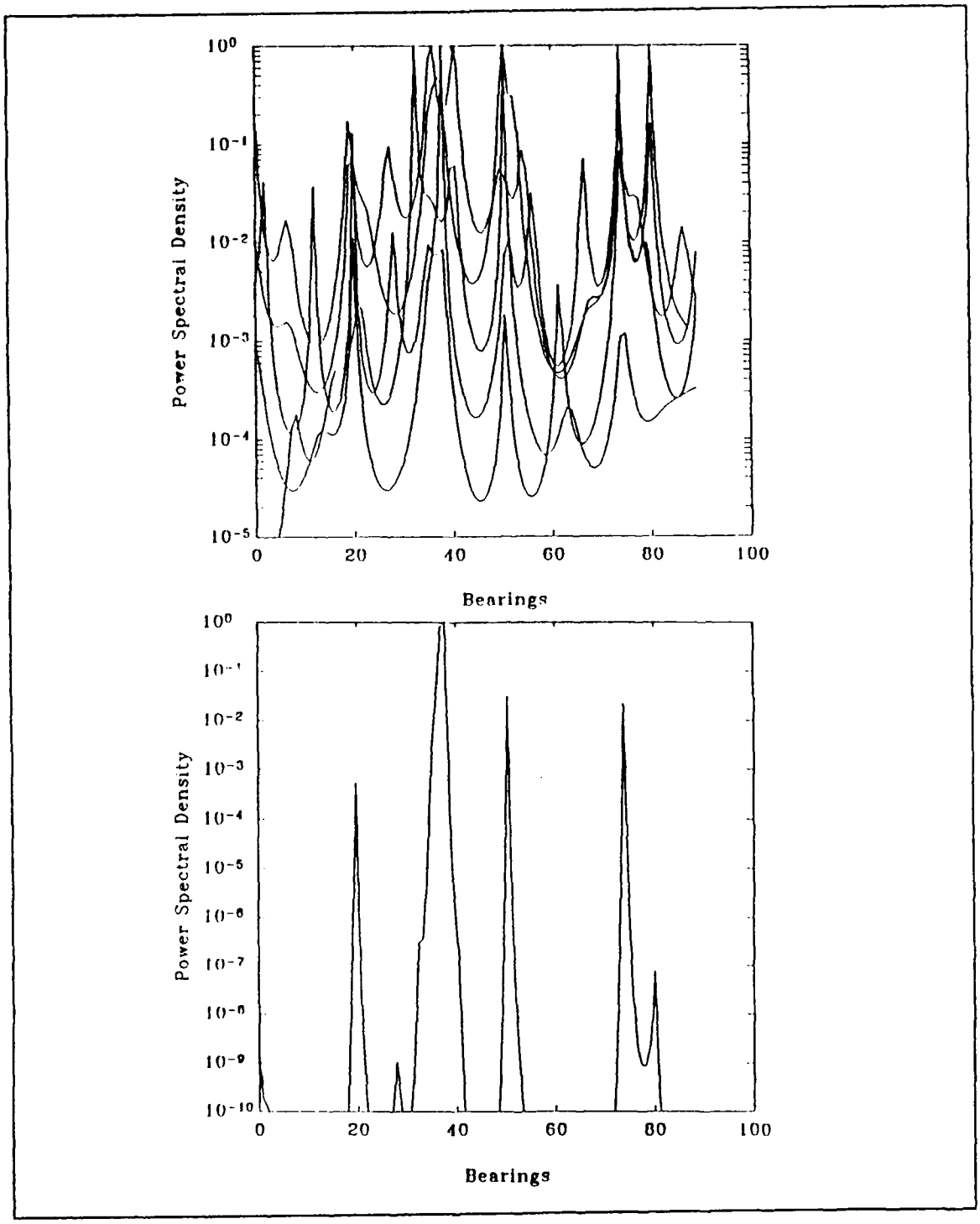


Figure 30. Case 3 -5 dB, 2 targets at 36 ° and 38 °: (a) overlay of PSDs from first through sixth eigenvectors, (b) product of the above PSDs

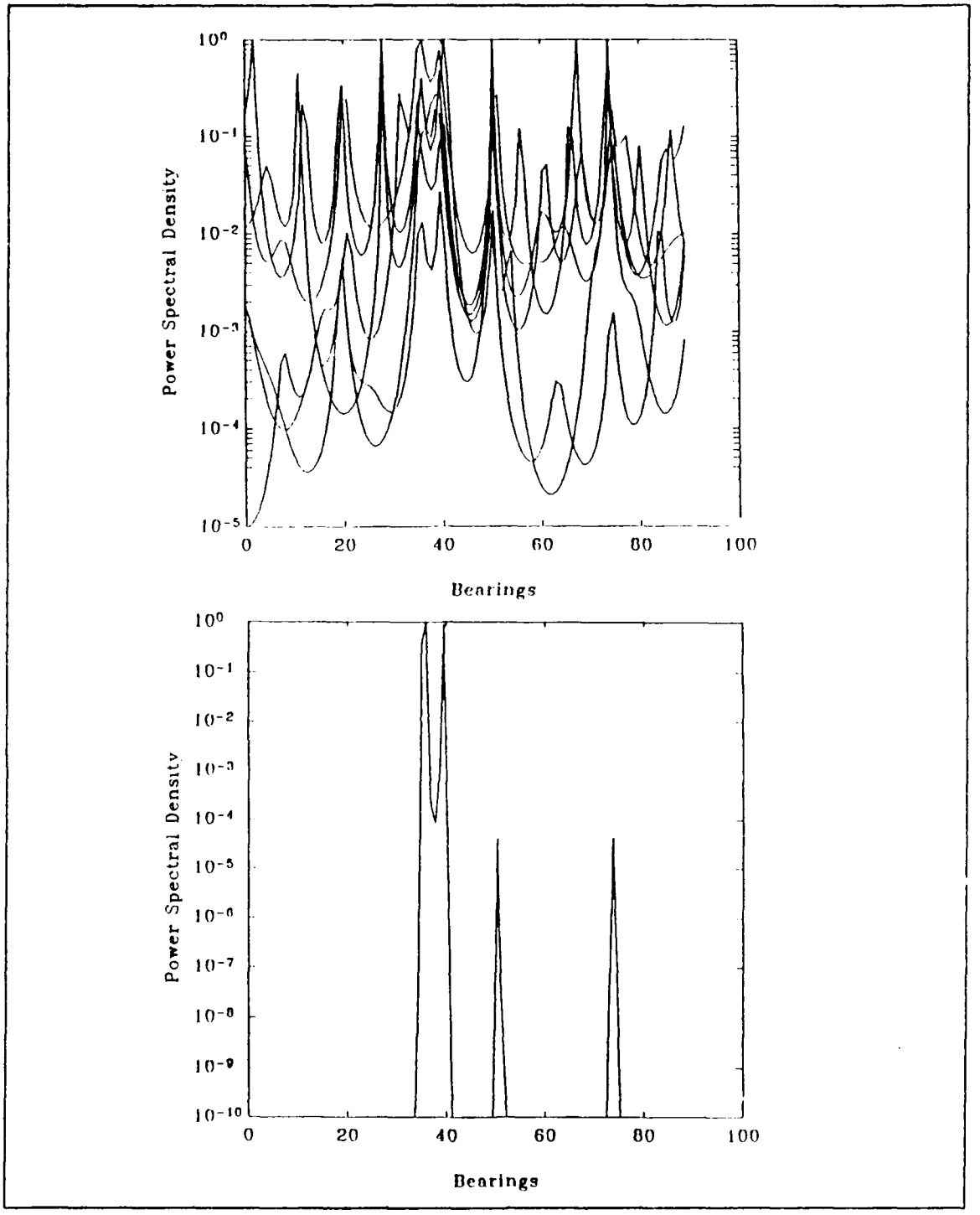


Figure 31. Case 3 -5 dB, 2 targets at 36 ° and 40 °: (a) overlay of PSDs from first through sixth eigenvectors, (b) product of the first 5 eigenvector PSDs

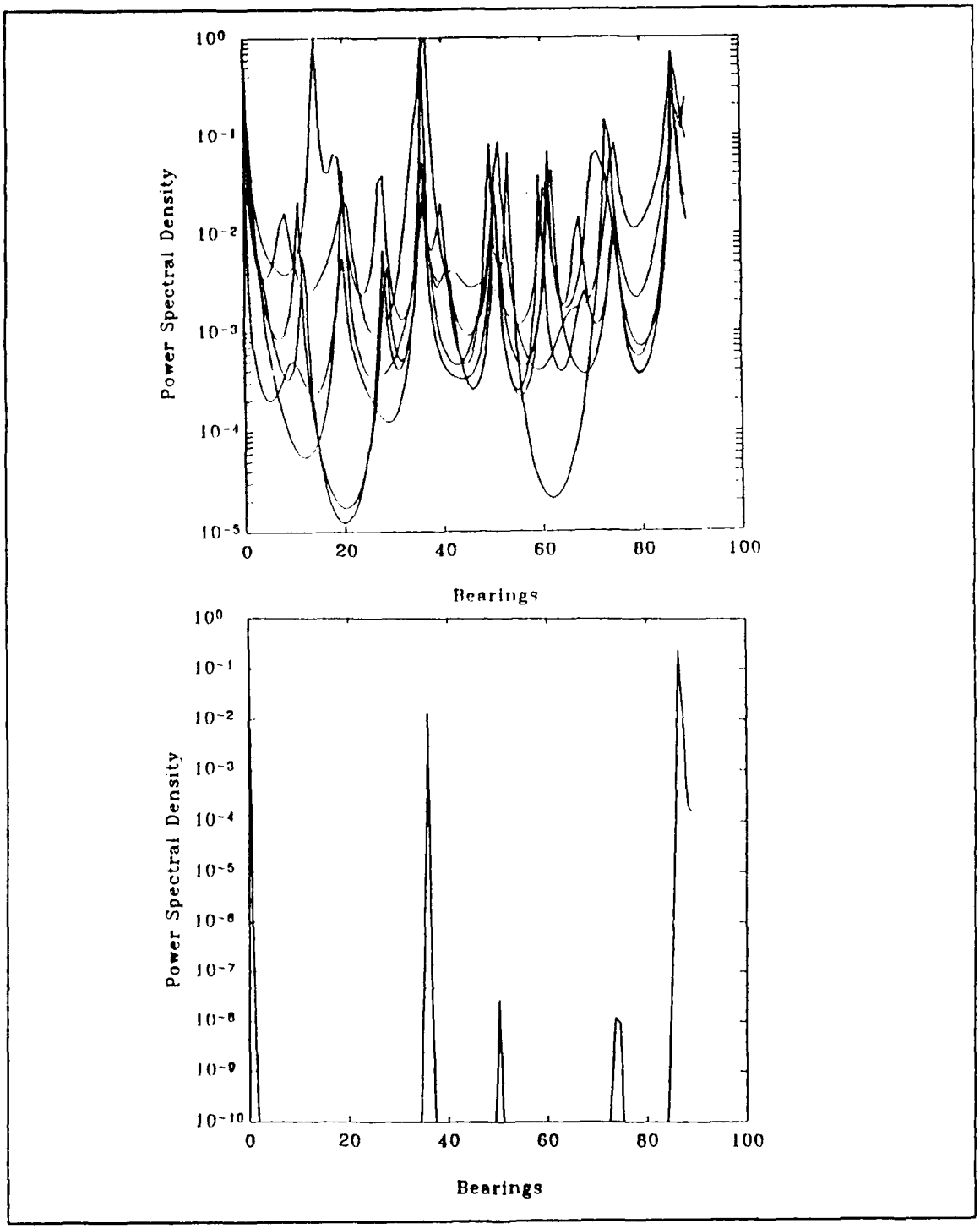


Figure 32. Case 3 -5 dB, 3 targets at 0 °, 36 ° and 88.2 °: (a) overlay of PSDs from first through sixth eigenvectors, (b) product of the first 10 PSDs

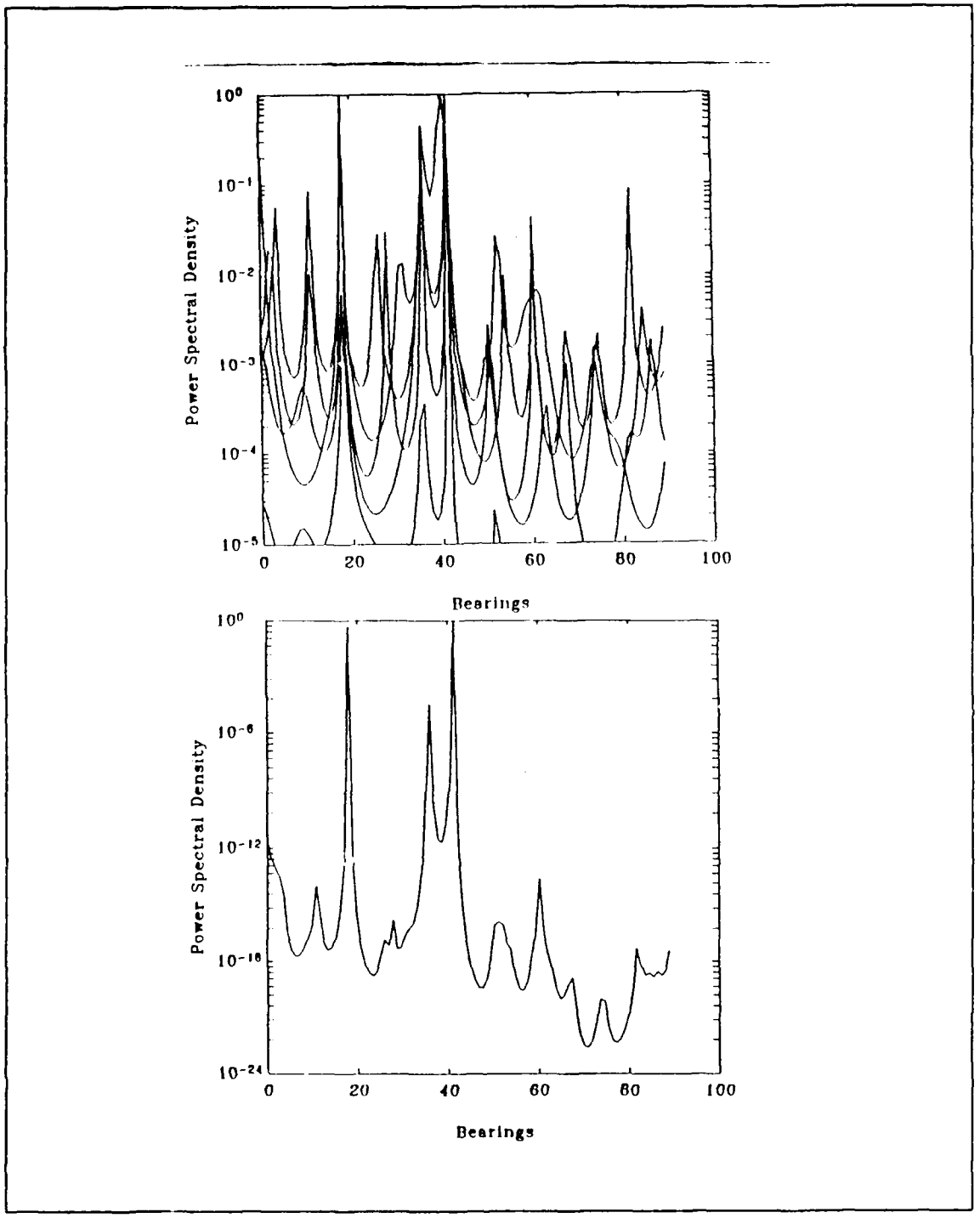


Figure 33. Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °: 100 sensors, (a) PSDs of second through sixth eigenvectors, (b) product of the above PSDs

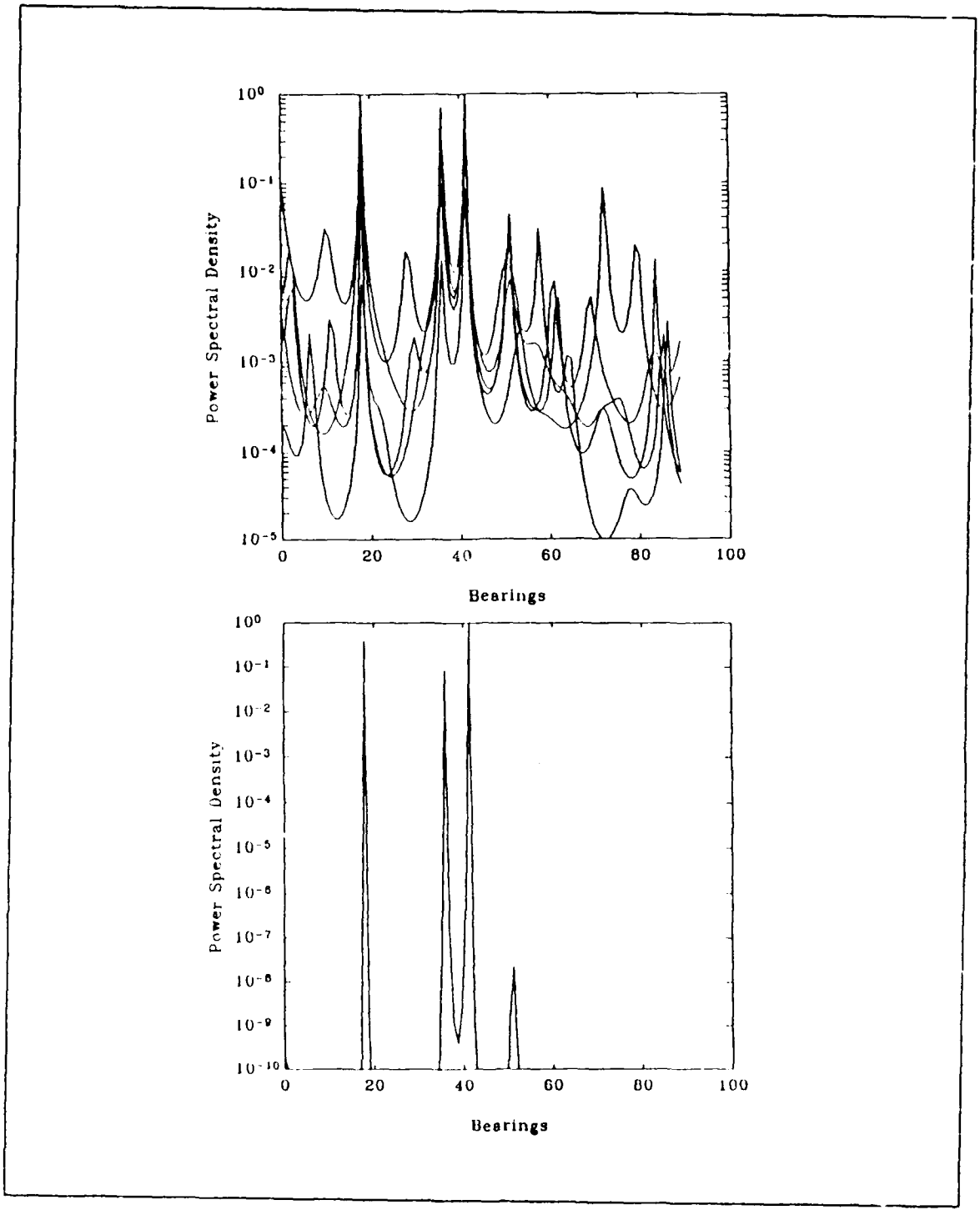


Figure 34. **Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °:** 75 sensors, (a) PSDs of second through sixth eigenvectors, (b) product of the above PSDs

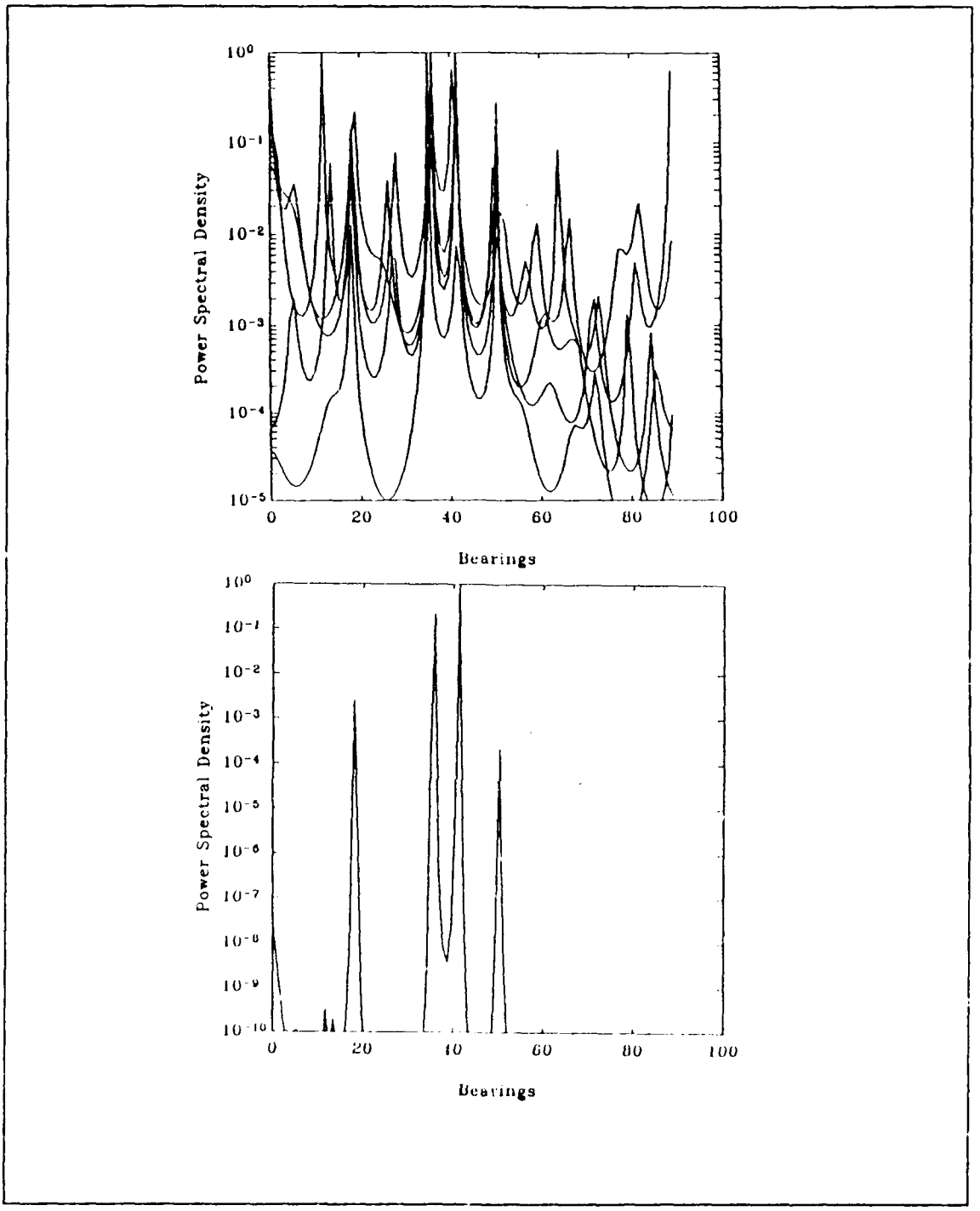


Figure 35. Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °: 50 sensors, (a) PSDs of second through sixth eigenvectors, (b) product of the above PSDs

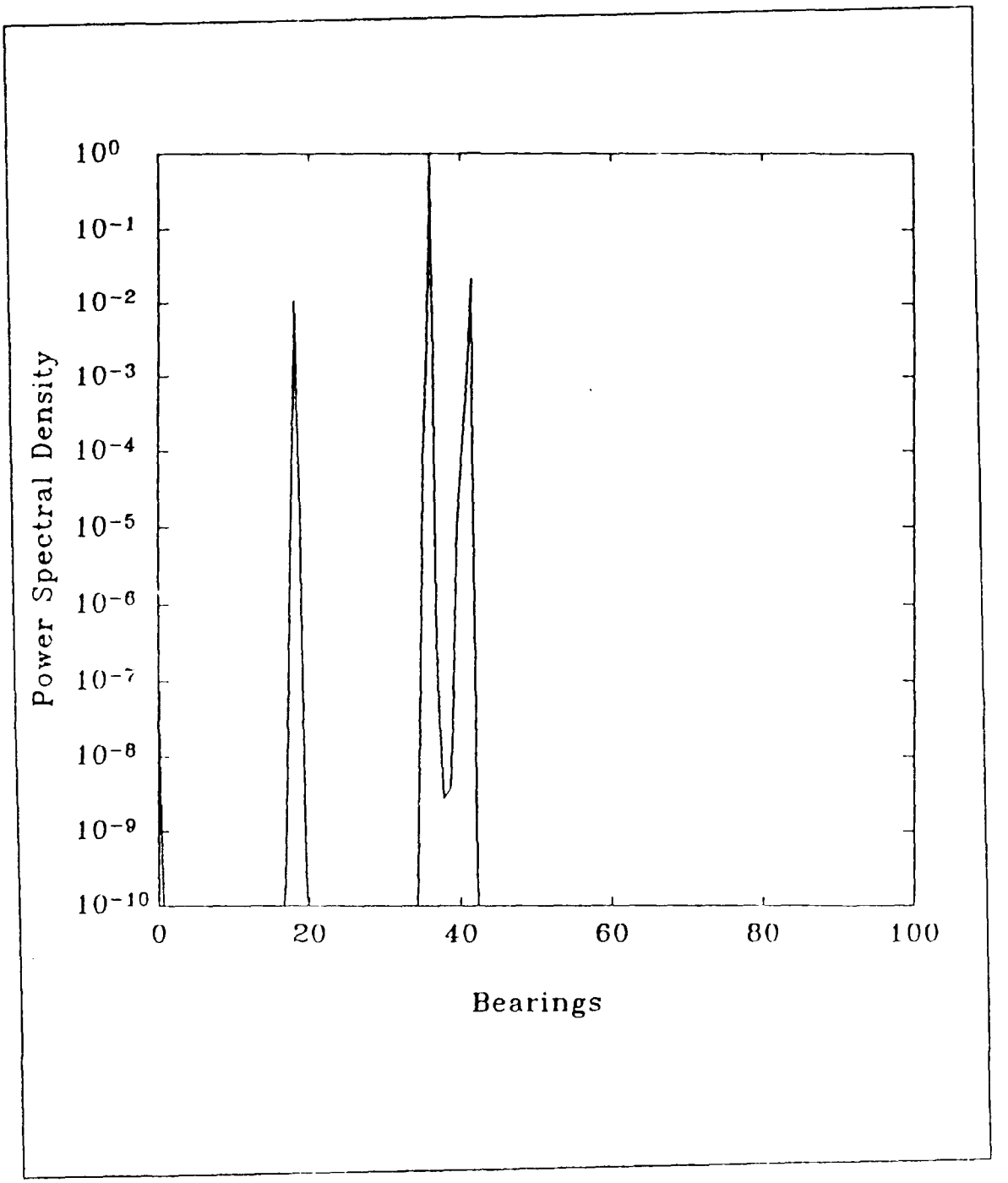


Figure 36. Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °: Product of the first through eighth eigenvector PSDs

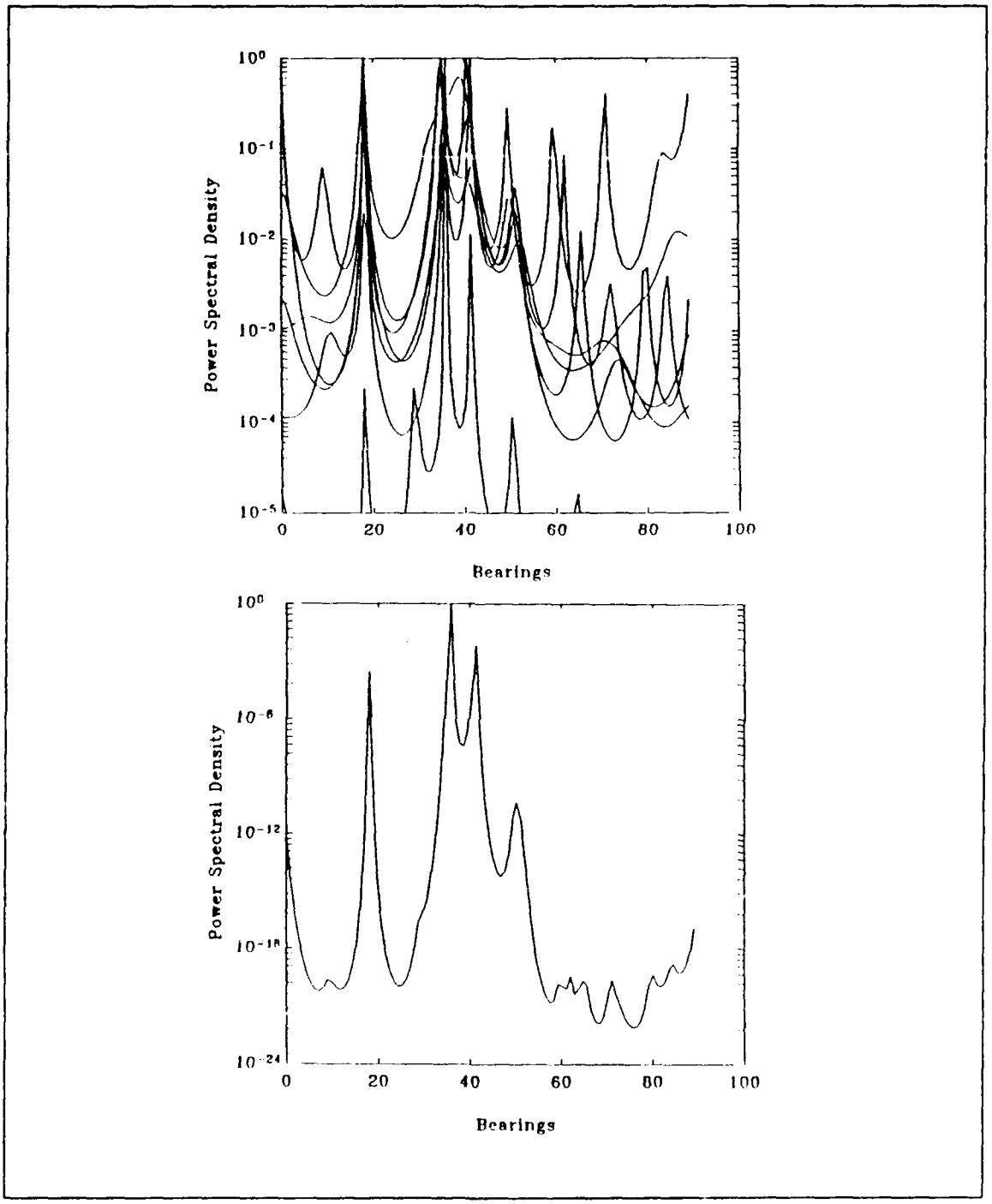


Figure 37. **Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °:** 40 sensors, 20 by 20 matrix, (a)PSDs of first through seventh eigenvectors, (b) product of the above PSDs

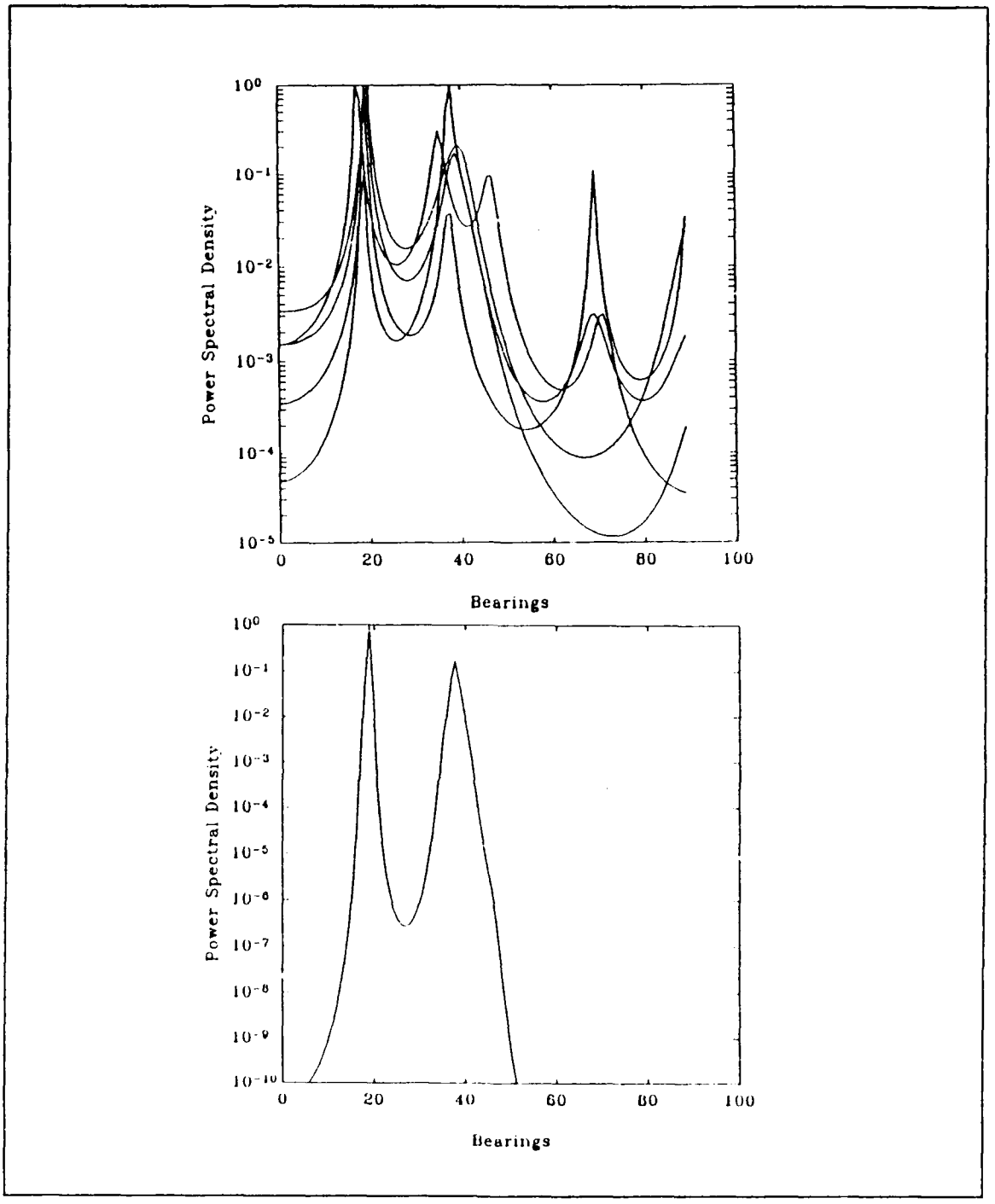


Figure 38. Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °: 30 sensors, 20 by 20 matrix, (a)PSDs of first through fifth eigenvectors, (b)Product of the first through fourth eigenvector PSDs

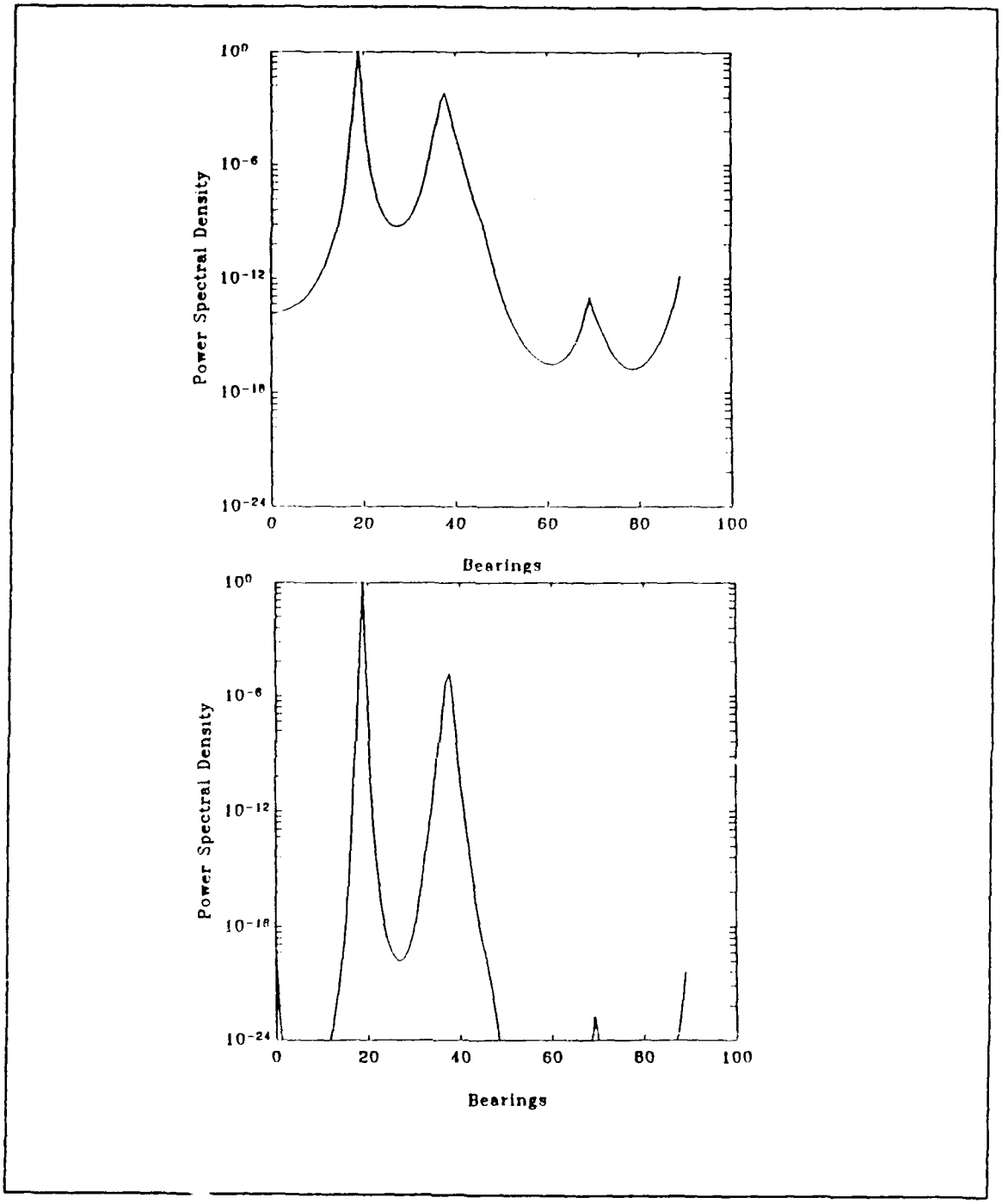


Figure 39. Case 4 5 dB, 3 targets at 18 °, 36 ° and 41.4 °: 30 sensors, 20 by 20 matrix, (a)Product of the first through fifth eigenvector PSDs, (b)Product of the first through twelfth eigenvector PSDs

V. CONCLUSIONS AND RECOMMENDATIONS

The results plotted in Chapter IV indicate that the eigenvectors found using the Lanczos algorithm are sufficiently accurate to determine the spectrum. Although no direct comparisons with other eigendecomposition methods are performed, the theory indicates that many fewer operations are required. We handle the other difficulty of conventional subspace methods by using only a few of the eigenvectors associated with the minimum eigenvalues of the autocorrelation matrix. No estimation of the noise subspace dimension is required or performed.

This theory may be applied to any system requiring rapid decomposition of the correlation matrix. Examples include phased array radar and passive acoustic arrays [Refs. 30, 31]. Reference 32 details an experimental system using the MUSIC algorithm for multiple source direction finding.

The following areas are recommended for future study.

- Use of the products of multiple spectra apparently resulted in good detection at low SNR. More research in this area to determine a physical interpretation of this method is required.
- Analysis and comparison of the results in terms of computational speed and accuracy with other eigendecomposition methods should be performed to find the true results.
- The Lanczos algorithm developed uses no reorthogonalization nor will it find repeating eigenvalues. Other forms of the Lanczos algorithm are available. Comparisons between these different methods to determine accuracy and speed may lead to more optimum results.
- A more detailed model should be developed that will simulate an array with a bank of bandpass filters to better forecast results of a physical implementation (as seen in Figure 18 of Chapter IV).
- A method which implements the algorithm in parallel fashion may be tried. Using one long linear array, several overlapping subarrays may be used to simultaneously create several autocorrelation matrices. The algorithm may be applied to these matrices in parallel. It is predicted that the greater number of available eigenvectors will more properly describe the noise subspace and therefore more accurately estimate the spectrum.

LIST OF REFERENCES

1. Kay, Steven M., *Modern Spectral Estimation*, Prentice-Hall, Englewood Cliffs, 1987

2. Schmidt, Ralph Otto, "A signal subspace approach to multiple emitter location and spectral estimation," Ph.D. Thesis, Stanford University, Stanford, California, Nov. 1981

3. Kay, Steven M., and Stanley Lawrence Marple, Jr., "Spectrum analysis--a modern perspective," *Proceedings of the IEEE*, Vol. 69, no. 11, Nov. 1981, pp. 1380-1419

4. Orphanidis, Sophocles J., *Optimum Signal Processing, An Introduction, 2d ed.* Macmillan, New York, 1985

5. Marple, S. Lawrence, Jr., *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, 1987

6. Kay, Steven, and Cedric Demeure, "The high-resolution spectrum estimator--a subjective entity," *Proceedings of the IEEE*, Vol. 72, no. 12, Dec. 1984, pp. 1815-1816

7. Harris, Fredric J., "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, Vol. 66, no. 1, Jan. 1978, pp. 51-83

8. Nuttall, Albert H., "Some windows with very good sidelobe behavior," *IEEE Trans. on ASSP*, Vol. ASSP-29, no.1, Feb. 1981, pp. 84-91

9. Van Veen, Barry D., and Richard A. Roberts, "Partially adaptive beamformer design via output power minimization," *IEEE Trans. on ASSP*, Vol. ASSP-35, no. 11, Nov. 1987, pp. 1524-1532

10. Van Veen, Barry D., and Kevin M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, Vol. 5, no. 2, Apr. 1988, pp. 4-24

11. Dudgeon, Dan E., and Russell M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, 1984

12. Schmidt, Ralph O., "New mathematical tools in direction finding and spectral analysis," *Proceedings of SPIE, Real Time Signal Processing VI*, Vol. 431, Aug. 23-25, 1983, San Diego, CA, pp. 7-19

13. Schmidt, Ralph O., "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Ant. and Prop.*, Vol. AP-34, no. 3, Mar. 1986, pp. 276-280

14. Paulraj, A., R. Roy, and T. Kailath, "Estimation of Signal Parameters via Rotational Invariance Techniques--ESPRIT," *Proceedings of the 19th Asilomar Conf. on Ckts., Syst., and Comp.*, Pacific Grove, 1986, pp. 83-89

15. Golub, Gene H., and Charles F. Van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, 1983

16. Kumaresan, Ramdas, and Arnab K. Shaw, "Superresolution by structured matrix approximation," *IEEE Trans. on ASSP*, Vol. ASSP-36, no. 1, Jan. 1988, pp. 34-44

17. Nikitakos, Nikitas V., "A comparison of two frequency domain adaptive beamforming algorithms for sonar signal processing," Master's thesis, Naval Postgraduate School, Monterey, California, 1988

18. Halpeny, Owen S., and Donald G. Childers, "Composite wavefront decomposition via multidimensional digital filtering of array data," *IEEE Trans. on Ckts. and Syst.*, Vol. CAS-22, Jun. 1975, pp. 276-286

19. Cadzow, James A., "A high resolution Direction-of-Arrival algorithm for narrowband coherent and incoherent sources," *IEEE Trans. on ASSP*, Vol. 36, no. 7, Jul. 1988, pp. 965-979

20. Johnson, Don H., and Stuart R. DeGraaf, "Improving the resolution bearing in passive sonar arrays by eigenvalue analysis," *IEEE Trans. on ASSP*, Vol. ASSP-30, no. 4, Aug. 1982, pp. 638-647

21. Lanczos, Cornelius, *Applied Analysis*, Prentice-Hall, Englewood Cliffs, 1956

22. Lanczos, Cornelius, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *Journal of Research of the National Bureau of Standards*, Vol. 45, no. 4, Oct. 1950, pp. 255-282

23. Cullum, Jane K., and Willoughby, R.A., *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1*, Birkhauser, Boston, 1985

24. Tufts, D. and Melissinos, C.D., "Simple, effective computation of principal eigenvectors and their eigenvalues and application to high-resolution estimation of frequencies," *IEEE Trans. on ASSP*, v. ASSP-34, no. 5, Oct. 1986, pp. 1046-1053

25. Paige, C. C., "Computational variants of the Lanczos method for the eigenproblem," *J. Inst. Math. Appl.*, **10**, 1972, pp. 373-381

26. Paige, C. C., "The computation of eigenvalues and eigenvectors of very large sparse matrices," Ph.D. Thesis, University of London, London, England, 1971

27. Parlett, B. N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, 1980

28. Paige, C. C., "Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem," *Linear Algebra and its Applications*, **34**, American Elsevier Publishing (1980), pp. 235-258

29. Cullum, Jane K., and Willoughby, R.A., *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 2*, Birkhauser, Boston, 1985

30. Nickel, U., "Angular superresolution with phased array radar: a review of algorithms and operational constraints," *IEE Proc.*, Vol. 134, PT. F, No. 1, Feb. 1987, pp. 53-59

31. Malpass, Robert L., "ATAS: Big ASW help for small ships," *U.S. Naval Institute Proceedings*, Vol. 115/1/1031, Jan. 1989, pp. 107-108

32. Schmidt, Ralph O., and Raymond E. Franks, "Multiple source DF signal processing: an experimental system," *IEEE Trans. on Ant. and Prop.*, Vol. AP-34, no. 3, Mar. 1986, pp. 281-290

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Murali Tummala, Code 62Tu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	3
5.	Professor C. W. Therrien, Code 62Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Professor Ralph Hippenstiel, Code 62Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Jeffrey M. Speiser, Code 743 Naval Ocean Systems Center San Diego, CA 92152-5000	1
8.	Dr. Rabi Madan Office of Naval Research Code 1114 800 North Quincy Street Arlington, VA 22217-5000	1
9.	Daniel E. Gear 104 Mervine Street Monterey, CA 93940	4
10.	Yong Joo Kim SMC 1017 Naval Postgraduate School Monterey, CA 93943-5000	1