"One And Two Dimensional Maximum Entropy Spectral Estimation"

Naveed Akhtar Malik

TECHNICAL REPORT 493
November 1981

Massachusetts Institute of Technology
Research Laboratory of Electronics
Cambridge, Massachusetts 02139

One and Two Dimensional Maximum Entropy Spectral Estimation

by

Naveed A. Malik

Submitted to the Department of Electrical Engineering and Computer Science on November 9, 1981 in partial fulfillment of the requirements for the degree of Doctor of Science.

Abstract

The Maximum Entropy power spectrum estimation problem for two dimensional signals is discussed. The problem involves obtaining a correlation matching power spectrum estimate that can be represented as the reciprocal of the spectrum of a two dimensional polynomial, whose order is the same as that of the known correlations.

The differences between the one dimensional and two dimensional problems are pointed out. A new algorithm for solving the highly non-linear two dimensional problem is developed. The "alternating projections" type of algorithm iterates between the correlations and the polynomial coefficients applying the requisite constraints in each domain to obtain the desired solution.

Implementations of the algorithm for main frame and mini-computers are described. The algorithm is used to investigate the properties of two dimensional Maximum Entropy power spectra, and the performance of the Maximum Entropy estimator is compared to the Maximum Likelihood and Bartlett estimates.

The algorithm is also shown to be applicable and useful for obtaining the power spectrum estimates for one-dimensional signals with missing correlation values.

A computer program for implementing the algorithm on a mini-computer is provided.

Thesis Supervisor: Jae S. Lim

Title: Assistant Professor of Electrical Engineering

To my parents

for their love, understanding and support

Acknowledgements

I would like to thank Prof. Jae Lim for enthusiastically guiding this thesis research. Working with him has been a genuine pleasure, both academically and personally. I would also like to thank Prof. A. V. Oppenheim who originally motivated this research, and whose encouragement was always timely and deeply appreciated. Special thanks are also due to Prof. J. H. McClellan for finding the time to serve as a reader for this thesis. His constructive criticism and helpful comments have helped to smooth out the rough edges of this manuscript.

Webster Dove deserves praise for maintaining the computer in a very reliable state, an undoubtedly mammoth undertaking.

Special thanks are due to my brother Saleem and my friends, Babar Khan, Javed Baqai and Fuad Khan, whose crazy humor mixed with patience and understanding made these long years bearable.

Finally, I would like to thank my parents whose love and support have made all this possible.

TABLE OF CONTENTS

FIGURES

TABLES

Chapter 1

Introduction

The problem of power spectrum estimation (PSE) arises in various fields such as speech processing,[1] seismic signal processing,[2] image restoration,[3] radar,[4] sonar,[5] radio-astronomy etc. and its applications range from identifying signal source parameters and transmission channel characteristics, to removing noise from images.[3] Consequently, this problem has received considerable attention in the literature, and a wide variety of techniques for PSE have been developed. One technique that has been widely investigated due to its high resolution characteristics is the Maximum Entropy method. The primary interest of this thesis is to investigate the Maximum Entropy method of power spectrum estimation for two-dimensional signals.

1.1 Background

The power spectrum of a zero-mean N-dimensional signal $x[\underline{n}]$ is defined as the N-dimensional Fourier transform of its autocorrelation function (ACF); that is,

$$P_x(\underline{\omega}) \;\underline{\Delta}\; F_N(R_x[\underline{n}])$$

where

$P_x(\underline{\omega})$: power spectrum of $x[\underline{n}]$

$R_x[\underline{n}]$: ACF of $x[\underline{n}]$

and

$$\underline{n} \;\underline{\Delta}\; (n_1, n_2, \ldots n_N)$$
$$\underline{\omega} \;\underline{\Delta}\; (\omega_1, \omega_2, \ldots \omega_N)$$

and F_N denotes the N-dimensional Fourier transform operation. Thus, to obtain the power spectrum of a signal, its autocorrelation function must be known over all "time"*.

The basic problem in PSE is that of forming the "best" estimate of the power spectrum of a signal given only a finite segment of the signal or its ACF. The commonly used approaches to PSE include:

Classical or conventional PSE:

- Autocorrelation estimates[6]
- Periodograms, averaged and/or modified periodograms[7,8]

Parametric signal modeling[9]

Data or ACF extension techniques[10,11]

The Maximum likelihood (ML) method[12]

The Maximum Entropy (ME) method[13]

The motivation behind and shortcomings of the traditional or classical approaches are well known. When the power spectrum estimate is obtained as the Fourier transform of the sample ACF, the implicit assumption made is that the data are zero outside the known region. Further, if unbiased estimates of the ACF are used, the power spectrum can display meaningless negative regions. In addition, when the periodogram is obtained via the Discrete Fourier Transform (DFT), the data are assumed to be periodic. Further, the variance of the periodogram does not decrease with increasing data length.[7] The variance can only be traded off against the resolution of the estimate. The major advan-

*Time is used here as a general concept. Signals of dimensionality higher than one will usually have spatial as well as temporal dependence, and what is implied here is that to obtain $P_x(\omega)$, $x[\underline{n}]$ must be known for all $\underline{n}$.

tages associated with the conventional approaches are that the power spectrum estimates are easy to compute and the methods extend in a straightforward manner to two and higher dimensional signals. Further, the classical methods offer good performance when a large amount of data are available for analysis.

Another general approach to PSE is to obtain a rational model for the signal generating process. The PS estimate is then obtained directly from the model parameters. Various modeling techniques, such as autoregressive (AR), moving average (MA) and autoregressive-moving average (ARMA), have been successfully developed for the one-dimensional (1-D) case.[9] These techniques can also be extended in a straightforward manner to some two-dimensional (2-D) signals but they have not achieved the same degree of success.[14]

If the original signal is assumed to be bandlimited and sampled sufficiently fast to avoid aliasing problems, then various iterative extrapolation schemes[11, 15] may be tried. The major problems associated with these methods are the typically slow convergence rates and the problems encountered in handling noisy data.[10] Although these techniques offer a viable alternative in the 1-D case, the computational burden involved in handling 2-D data can become enormous as the storage and computational requirements increase quadratically.

The Maximum Likelihood approach to PSE has its basis in array processing.[12] It was originally motivated by one of the shortcomings of the conventional PSE methods. In the conventional methods the windowing of the ACF implies that the spectrum is smoothed by a fixed window. The ML method smooths the spectrum with an adaptive window that attempts to minimize the leakage or interference from neighboring frequencies, thereby achieving better resolution. The extension of the ML method to 2-D problems is straightforward.

Another approach to the PSE problem is the Maximum Entropy method. This approach imposes far fewer constraints on the data than the conventional methods and assumes that all the information that can be obtained about the signal generating process, is contained in the available data segment. A measure of the information, usually the entropy, is then chosen and maximized under the constraint that it must be consistent with the known information. This procedure then leads to the so-called "Maximum Entropy" method of power spectrum estimation, first expounded by Burg[13] in 1967. Pendrell[14] has shown that the ML method can also be considered as an information maximization method, with a different approximation used for the entropy of the process than is employed in the ME procedure. The resolution of the ML approach lies between that of the conventional and ME methods, a fact that was very neatly quantified by Burg.[16]

The ME method which has better resolution characteristics than the ML method, can also be viewed as a technique for extending the ACF beyond the known limits. It always leads to an all-pole or autoregressive model for the spectrum, and in 1-D the problem is identical to autoregressive modeling of the signal.[17] This leads to a linear problem formulation that is analytically tractable and computationally attractive. Unfortunately, the corresponding 2-D problem is highly non-linear, and although the form of the spectral estimate is still autoregressive, no closed form solution to the problem has been found so far. In fact, the previously proposed solutions are computationally expensive, and are usually approximations to the true solution. The main objective of this thesis research was to develop and characterize a new algorithm for obtaining the true 2-D Maximum Entropy power spectrum estimates. Although the algorithm was developed primarily for the 2-D problem, it has also proved applicable and useful for the case of 1-D signals with missing correlation values. In such cases, the 1-D ME PSE problem is also non-linear and a closed form solution does not exist, and therefore the algorithm developed in this thesis provides a viable alternative for obtaining the power spectral estimates in these cases.

1.2 Thesis Outline

The maximum entropy problem is defined and discussed in Chapter 2, and the form of the PS estimate is obtained. The differences between the 1-D and 2-D problems are outlined, and the previously proposed solutions to the 2-D problem are discussed.

Chapter 3 is devoted to developing the new algorithm for 2-D ME PSE. The algorithm is introduced at the conceptual level, and then developed into a practical technique.

Chapter 4 discusses the implementation of the algorithm on large (mainframe) computers with unlimited on-line storage capability, and on minicomputers with limited on-line storage. Three different implementations are compared.

Chapter 5 characterizes the properties of the ME PS estimates of 2-D sinusoids buried in white Gaussian noise (WGN). The characterization is done in terms of the effects of signal to noise (S/N) ratio, size of the known ACF array, and the starting phase of the sinusoids, on the ME PS estimates. The resolution properties and the accuracy of peak location estimation of the ME estimates are compared to conventional and ML estimates. In addition, the application of the algorithm to a practical example using real data is shown.

Chapter 6 discusses the application of the algorithm to the case of 1-D signals with missing correlation points. This situation can arise when the data are sampled over a non-uniform grid. In such cases, even the 1-D ME PSE problem is highly non-linear and no closed form solution has been proposed.

Chapter 7 summarizes the thesis and indicates directions for future research.

Appendix A contains the source program for a mini-computer (PDP-11, UNIX system).

1.3 Notation

Standard notation is used throughout this thesis. The under-bar indicates vector valued quantities. Thus,

$\underline{n} \ \underline{\Delta} \ (n_1, n_2, \ldots, n_N)$
$\underline{\omega} \ \underline{\Delta} \ (\omega_1, \omega_2, \ldots, \omega_N)$
$x[\underline{n}]$: N—dimensional time signal.
In the context of this thesis, N will generally be equal to one or two.

$R_x[\underline{n}]$: ACF of $x[\underline{n}]$
$P_x(\underline{\omega})$: Power Spectrum of $x[\underline{n}]$
$\hat{P}_x(\underline{\omega})$: An Estimate of the Power Spectrum of $x[\underline{n}]$

$\lambda[\underline{n}]$: AR coefficients corresponding the power spectrum, that is
$\lambda[\underline{n}] = F^{-1}[\hat{P}_x(\underline{\omega})]$
F : Fourier Transform operation
F^{-1} : Inverse Fourier Transform operation
The dimensionality of the Fourier transform will usually be clear from the context.

The region of support for the known ACF segment is referred to as the region "A". Due to the Hermetian symmetry of the ACF, the region A is always symmetric about the origin, and is assumed to contain the origin for both 1-D and 2-D signals.

Power spectrum estimates of the form $\frac{1}{F[\lambda[\underline{n}]]}$ are referred to as all-pole or autoregressive (AR) estimates regardless of dimensionality.

Chapter 2

The Maximum Entropy PSE Problem

2.1 The Maximum Entropy Method

The formulation of the Maximum Entropy method of power spectrum estimation was first expounded by Burg[13] in 1967. In order to circumvent the artificial assumptions made about the data by the conventional methods of PSE, Burg suggested that the data, or equivalently, its ACF, be assumed non-zero outside the known segment. The assumption was that the entropy-density was not a function of the unknown correlation values. Since the entropy is a measure of the information about the process, this is equivalent to assuming that all the information about the generating process is contained in the available segment, and thus, that the value of the entropy is the maximum possible that is consistent with the known data. This is the reason why the technique is referred to as the Maximum Entropy method.

2.2 Entropy of a 2-D Process

The term "entropy" used in the context of power spectrum estimation represents a measure of the information content of a signal. The amount of information contained in a signal can be quantified by the length of the message required for its transmission. It is easily seen that encoding of information requires a number of digits proportional to the logarithm of the inverse of the probability of occurrence of the event.[18] When not all events are equally probable, the average information is measured by

$$H = \frac{-1}{\log r} \sum_j p_j \log p_j$$

a quantity termed the "entropy" by Shannon.[19] Here, p_j represenʟs the probability of occurrence of the j'th event, and 'r' is the base of the number system used to measure the entropy (e.g. r = 2 means binary digits). The unsubscripted logarithm is taken to be the natural logarithm.

When the random variable can take on a continuum of values, the sum in the definition of the entropy is replaced by an integral. Further, when one deals with the realizations of a process, the probability is replaced by the corresponding joint probability density function (PDF). The expression for the entropy therefore becomes:

$$H = -\int f(\underline{z}) \log f(\underline{z})\, d\underline{z} \qquad (2.1)$$

where $f(\underline{z})$ represents the PDF of the process $\underline{z}$. In order to obtain the power spectrum estimate of a wide sense stationary process, only its first and second order statistics are required. Therefore, in this context, one cannot distinguish a given time series from one which can be fully characterized by its first and second order statistics only, namely, a Gaussian process.

Suppose we have a 2-D Gaussian process $\underline{z}_{NM}$:

$$\underline{z}_{NM} \;\underline{\Delta}\; \begin{array}{cccc} z_{11} & z_{12} & \cdots & z_{1M} \\ z_{21} & z_{22} & \cdots & z_{2M} \\ \cdot & & & \\ \cdot & & & \\ z_{N1} & z_{N2} & \cdots & z_{NM} \end{array}$$

The PDF of the process is given by

$$f(\underline{z}_{NM}) = \frac{1}{(2\pi)^{NM/2} |\Phi_{NM}|^{1/2}} e^{\frac{-1}{2} \sum_1^N \sum_1^N \sum_1^M \sum_1^M \varphi_{ijkl}^{-1} z_{ik}^{*} z_{jl}} \qquad (2.2)$$

where the symbol '*' represents the complex conjugate. Φ_{NM} is the NM x NM matrix of autocorrelations. It consists of an M x M equidiagonal array of N x N

Toeplitz (equidiagonal) matrices. It is often referred to as a "block Toeplitz" matrix. φ_{ijkl} represents the element in the i'th row, j'th column of the block matrix which lies in the k'th row and l'th column of the array of block matrices. Its value is

$$\varphi_{ijkl} = E[z_{ik}z_{jl}^{*}] = \varphi(i-j,k-l)$$

Similarly, φ_{ijkl}^{-1} represents an element of the inverse of Φ_{NM}.

Substituting eq.(2.2) into eq.(2.1), we obtain

$$H(\underline{z}_{NM}) = \frac{1}{2}\ln|\Phi_{NM}|$$

In the limit, $N,M\to\infty$, $H(\underline{z}_{NM})$ diverges, and it becomes necessary to define the entropy density:

$$H \triangleq \lim_{N,M\to\infty}\frac{1}{NM}H(\underline{z}_{NM})$$

$$= \lim_{N,M\to\infty}\frac{1}{2NM}\ln|\Phi_{NM}|$$

The entropy density can now be related to the power spectrum by the use of Szego's theorem.[20] In 2-D, the theorem is:

$$\lim_{N,M\to\infty}\frac{1}{NM}\sum_{i=1}^{NM}F(\lambda_i) = \frac{1}{4f_N^2}\int\int_{-f_N}^{f_N}F(4f_N^2P(f_x,f_y))\,df_x\,df_y$$

where the λ_i are the NM eigenvalues of the matrix Φ_{NM}, $P(f_x,f_y)$ is the spectral density at the frequency (f_x,f_y), and f_N is the Nyquist frequency (assumed to be identical for the x and y directions). F(.) is an arbitrary function. Choosing F(.) = ln(.), we obtain

$$\lim_{N,M\to\infty}\frac{1}{NM}\ln(\prod_{i=1}^{NM}\lambda_i) = \frac{1}{4f_N^2}\int\int_{-f_N}^{f_N}\ln(4f_N^2P(f_x,f_y))\,df_x\,df_y$$

which gives

$$\lim_{N,M\to\infty}\frac{1}{NM}\ln|\Phi_{NM}| = \frac{1}{4f_N^2}\int\int_{-f_N}^{f_N}\ln(4f_N^2P(f_x,f_y))\,df_x\,df_y$$

or

$$H = \frac{1}{4f_N^2}\int\int_{-f_N}^{f_N} \ln(4f_N^2 P(f_x,f_y))\,df_x\,df_y \qquad (2.3)$$

which is the expression relating the entropy density to the log of the power spectrum. This is the 2-D analog of the expression derived by Smylie et al.[18]

2.3 The Problem

The technique of ME PSE can be quantified by considering the expression for the entropy, eq.(2.3). Rewriting eq.(2.3) in terms of the radian frequency $\underline{\omega}$, and dropping the constants which do not affect the methodology, we get

$$H = \int_{\underline{\omega}} \ln P_x(\underline{\omega})\, d\underline{\omega}$$

Writing the power spectrum as the Fourier transform of the ACF, we obtain

$$H = \int_{\underline{\omega}} \ln\sum_{\underline{n}} R_x[\underline{n}]e^{-j\underline{\omega}\cdot\underline{n}}d\,\underline{\omega} \qquad (2.4)$$

Now, suppose that $R_x[\underline{n}]$ is known only over a finite region "A". Separating the summation in eq.(2.4) into the known and unknown regions, we obtain

$$H = \int_{\underline{\omega}} \ln\left[\sum_{\underline{n}\in A} R_x[\underline{n}]e^{-j\underline{\omega}\cdot\underline{n}} + \sum_{\underline{n}\not\subset A} R_x[\underline{n}]e^{-j\underline{\omega}\cdot\underline{n}}\right] d\underline{\omega} \qquad (2.5)$$

Now, maximizing the entropy with respect to the unknown ACF values, we obtain

$$\frac{dH}{dR_x[\underline{m}]} = 0 = \int_{\underline{\omega}} \frac{e^{-j\underline{\omega}\cdot\underline{m}}}{\sum_{\underline{n}} R_x[\underline{n}]e^{-j\underline{\omega}\cdot\underline{n}}} d\underline{\omega} \quad \text{for } m \not\subset A$$

or

$$\int_{\underline{\omega}} \frac{e^{-j\underline{\omega}\cdot\underline{n}}}{P_x(\underline{\omega})} d\underline{\omega} = 0 \quad \text{for } m \not\subset A \qquad (2.6)$$

But this is just the inverse Fourier transform of one over the power spectrum. Hence, if we write,

$$G_x(\underline{\omega}) = \frac{1}{P_x(\underline{\omega})} \qquad (2.7)$$

then $G_x(\underline{\omega})$ is also a valid power spectrum, since $P_x(\underline{\omega}) > 0$ for all $\underline{\omega}$. Therefore, it can be represented as the Fourier transform of some ACF, say $\lambda[\underline{n}]$. That is,

$$G_x(\underline{\omega}) = \sum_{\underline{n}} \lambda[\underline{n}] e^{-j\underline{\omega}\cdot\underline{n}} \qquad (2.8)$$

or equivalently

$$\int_{\underline{\omega}} G_x(\underline{\omega}) e^{j\underline{\omega}\cdot\underline{n}} d\omega = \lambda[\underline{n}] \qquad (2.9)$$

But eq.(2.6) then indicates that $\lambda[\underline{n}]$ is zero outside the region A. This leads to the very important conclusion that the ME PS estimate can be represented by an autoregressive model, whose coefficients have finite support. Further, this support is exactly the same as that of the known ACF of the signal $x[\underline{n}]$. Hence,

$$P_{MEM}(\underline{\omega}) = \frac{1}{\sum_{\underline{n}\in A} \lambda[\underline{n}] e^{-j\underline{\omega}\cdot\underline{n}}} \qquad (2.10)$$

Since the above discussion was completely general, eq.(2.10) shows that regardless of signal dimensionality, the form of the ME power spectrum estimate is always autoregressive in nature.

The above derivation implies that the ME problem can equivalently be stated as follows. Given $R_x[\underline{n}]$ for $\underline{n} \in A$, determine $\hat{P}_x(\underline{\omega})$ such that $\hat{P}_x(\underline{\omega})$ is of the form:

$$\hat{P}_x(\underline{\omega}) = \frac{1}{\sum_{\underline{n}\in A} \lambda[\underline{n}] e^{-j\underline{\omega}\cdot\underline{n}}} \qquad (2.11)$$

and

$$F^{-1}\left[\hat{P}_x(\underline{\omega})\right] = R_x[\underline{n}] \qquad \text{for } \underline{n} \in A \qquad (2.12)$$

2.4 Signal Dimensionality And The Maximum Entropy Problem

The above problem statement applies to all signals regardless of their dimensionality. The solutions to the problem, however, depend strongly on the signal dimensionality. For 1-D signals, when the ACF is known over a connected region, the mean- square error minimization of the prediction filter based on autoregressive signal modeling requires solving a set of linear equations for the filter coefficients. Further, the power spectrum estimate obtained from these filter coefficients is identical to the ME power spectrum estimate.[21] For 2-D signals, unfortunately, this is no longer the case. Specifically, even though minimizing the mean-square error of the autoregressive filter still requires solving a set of linear equations, the power spectrum obtained from the estimated filter coefficients is no longer the ME PS estimate. The reason for this can be seen by examining the form of the normal equations for the filter coefficients in the autoregressive signal modeling case. The derivation of the general form of the normal equations for 2-D signals is completely analogous to the 1-D case. Minimizing the squared prediction error with respect to the filter coefficients gives

$$\sum_{(i,j)\in B}\sum a_{ij}R_x(r-i,s-j) = R_x(r,s) \qquad \text{for } (r,s)\in B \qquad (2.13)$$

where the a_{ij} represent the autoregressive filter coefficients to be estimated, and the set B consists of all the points where the filter mask has non-zero values. The power spectrum obtained from the a_{ij} is given by

$$\hat{P}_x(\omega_1,\omega_2) = \frac{1}{\left|\sum_{(k,l)\in B}\sum a_{kl}.e^{-j\omega_1 k}.e^{-j\omega_2 l}\right|^2} \qquad (2.14)$$

From eq.(2.13) for any non-trivial choice of B, that is, if B does not consist of a set of collinear points, the number of independent values of $R_x(n_1,n_2)$

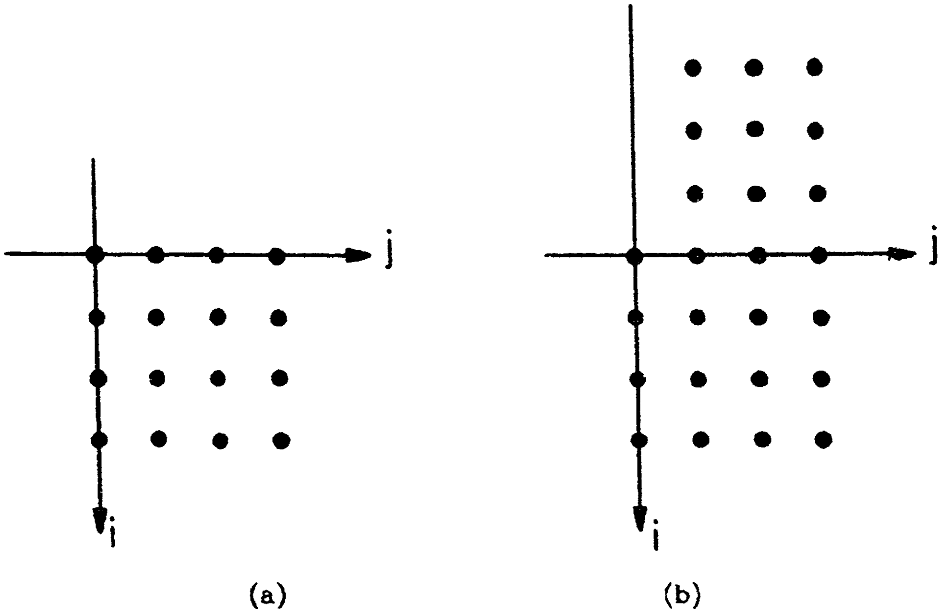


Fig. 2.1: Twc dimensional autoregressive signal modeling. (a) First quadrant autoregressive mask of size 4x4. (b) Independent autocorrelation points required to solve the normal equations for the mask of (a).

required to solve the abov- set of equations for the filter coefficients is greater than the number of filter coefficients. For example, consider the filter mask shown in Fig. 2.1(a), in which the dots represent the region for which the a_{ij} are non-zero. Fig. 2.1(b) shows the independent values of $R_x(n_1,n_2)$ required to solve for the a_{ij} in Fig. 2.1(a) by eq.(2.10). Clearly, the number of correlation points is greater than the number of filter coefficients. Since the estimated power spectrum given by eq.(2.14) is completely determined by the filter coefficients alone, it does not possess enough degrees of freedom to satisfy eq.(2.11) which is required for the ME PS estimate. Thus we see that the AR model obtained by the linear least squares formulation does not coincide with the ME problem in 2-D. In fact, the 2-D ME PSE problem is highly non-linear and a closed form solution has not yet been found.

In the absence of a closed form solution, it is important to know the conditions for the existence and uniqueness of the solution. In this regard, Woods[2] has obtained the theoretical result that if the given $R_x(n_1,n_2)$ is a part of some positive-definite correlation function, [meaning that its Fourier transform is positive for all (ω_1,ω_2)], a unique solution to the ME problem does exist. In general, it is difficult[22] to determine if the given segment of the ACF is a part of some positive-definite correlation function, even though this is generally the case in most practical problems. For example, if the ACF is obtained from the data using the biased estimator for the ACF[6] as is done generally, the correlation function is always positive-definite. The problem of the extendibility of the given segment of the ACF has been discussed by Lang,[23] and a mathematical characterization of the problem has been offered. In this thesis, we assume that the given segment of the correlation function is indeed extendible, so that the ME PS estimate exists, and is unique.

2.5 Proposed Solutions

In his Ph.D. thesis Burg[24] describes a general variational approach to estimating any function, which can be applied to the problem of 2-D ME power spectrum estimation. The proposed solution is iterative in nature and requires the inversion of a matrix in each iteration where the dimension of the matrix is of the order of the number of given autocorrelation values. However, no results using the technique have been presented.

Wernecke and D'Addario[25] have attempted the Maximum Entropy reconstruction of images from noisy data. Again, the algorithm proposed is iterative and attacks the primal problem, that is, the entropy is maximized under appropriate constraints. The maximizatiou is done by continuously adjusting the power spectrum estimate and evaluating the expression for the entropy and its gradient. A few optimization algorithms are proposed, which are computationally expensive and require large amounts of storage. However, the results shown are encouraging.

Woods[2] has given a constructive proof for the existence and uniqueness of the 2-D ME spectral estimates, under the condition that the known ACF values form part of a positive definite autocorrelation function. He has also proposed a solution to the problem based on the assumption that the power spectrum can be expressed as a convolutional power series (a power series in the frequency domain). However, the algorithm is not proved to converge, and the very nature of the formulation leads to an approximation to the true solution. Further, the approach is limited to power spectra which can be expressed in the form of the proposed convolutional power series.

Lang[23] has tackled the general multi-dimensional ME problem for sensor arrays. The problem of the extendibility of the given ACF segment is characterized. The ME power spectrum estimation problem is reduced to the solution of a finite-dimensional convex optimization problem and iterative algorithms using the method of steepest descent are proposed. The algorithms are proven to converge, and results for both the one-dimensional and two-dimensional problems are shown.

Jain and Ranganath[26] have also developed iterative algorithms for the solution of the ME problem. However, their algorithms do not use the method of steepest descent, and are typically slower than those proposed by Lang.

Other attempts to solve the 2-D ME PSE problem have been made by Ong,[27] Roucos and Childers,[28] Newman[29] and others, but none of them has achieved any high degree of success.

Chapter 3

The Algorithm

In this chapter, a new iterative algorithm for obtaining ME PS estimates for 2-D signals is developed. This algorithm has also proved useful for the case of one dimensional signals with missing correlation points. That application is discussed in Chap. 6.

3.1 The Conceptual Algorithm

Recall from Sec. 2.1.1 that the 2-D ME PSE problem can be stated as follows:

Given $R_x[\underline{n}]$ for $\underline{n} \in A$, determine $\hat{P}_x(\underline{\omega})$ such that $\hat{P}_x(\underline{\omega})$ is of the form:

$$\hat{P}_x(\underline{\omega}) = \frac{1}{\sum\limits_{\underline{n}\in A} \lambda[\underline{n}]e^{-j\underline{\omega}.\underline{n}}} \tag{3.1}$$

and satisfies the consistency constraint

$$F^{-1}[\hat{P}_x(\underline{\omega})] = R_x[\underline{n}] \quad \text{for } \underline{n} \in A \tag{3.2}$$

In this section we develop a new iterative algorithm for obtaining the 2-D ME power spectrum estimate.

Suppose we are given $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$ such that $R_x(n_1,n_2)$ is a segment of some positive definite correlation function. To find the unique ME PS estimate, we express a power spectrum $P_y(\omega_1,\omega_2)$ as follows.

$$\begin{aligned} P_y(\omega_1,\omega_2) &= F[R_y(n_1,n_2)] \\ &= \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} R_y(n_1,n_2).e^{-j\omega_1 n_1}.e^{-j\omega_2 n_2} \end{aligned} \tag{3.3}$$

and

$$\frac{1}{P_y(\omega_1,\omega_2)} = F[\lambda(n_1,n_2)]$$
$$= \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} \lambda(n_1,n_2).e^{-j\omega_1 n_1}.e^{-j\omega_2 n_2} \qquad (3.4)$$

From eq.(3.3) and eq.(3.4) it is clear that $R_y(n_1,n_2)$ can be obtained from $\lambda(n_1,n_2)$ and vice versa through direct and inverse Fourier transform operations. Now, from eq.(3.1) and eq.(3.2), $P_y(\omega_1,\omega_2)$ is the unique ME PS estimate if and only if $\lambda(n_1,n_2) = 0$ for $(n_1,n_2) \notin A$ and $R_y(n_1,n_2) = R_x(n_1,n_2)$ for $(n_1,n_2) \in A$. Thus, we see that for $P_y(\omega_1,\omega_2)$ to be the desired ME PS estimate, we have a constraint on $R_y(n_1,n_2)$ (consistency) and a constraint on $\lambda(n_1,n_2)$ (finite support). Recognizing this, it is straightforward to develop a simple iterative algorithm to find the unique ME PS estimate. Specifically, we go back and forth between $R_y(n_1,n_2)$, (the correlation domain) and $\lambda(n_1,n_2)$ (the coefficient domain) and each time, impose the requisite constraints on the correlation and coefficient values. Thus, starting with some initial estimate for $\lambda(n_1,n_2)$ we obtain an estimate for $R_y(n_1,n_2)$. This estimate is then corrected by the given $R_x(n_1,n_2)$ over the region A and is used to generate a new $\lambda(n_1,n_2)$. The new $\lambda(n_1,n_2)$ is then truncated to the desired limits and this procedure is repeated. The above iterative procedure is illustrated in Fig. 3.1 and forms the basis of the new iterative algorithm for 2-D ME PS estimation.

The iterative procedure discussed above is very similar in form to other iterative techniques,[30,31] that have been successfully used in image processing. Even though the conditions under which the algorithm converges are not yet known, if the algorithm converges, then the solution satisfies both eq.(3.1) and eq.(3.2) and consequently is the desired ME PS estimate.

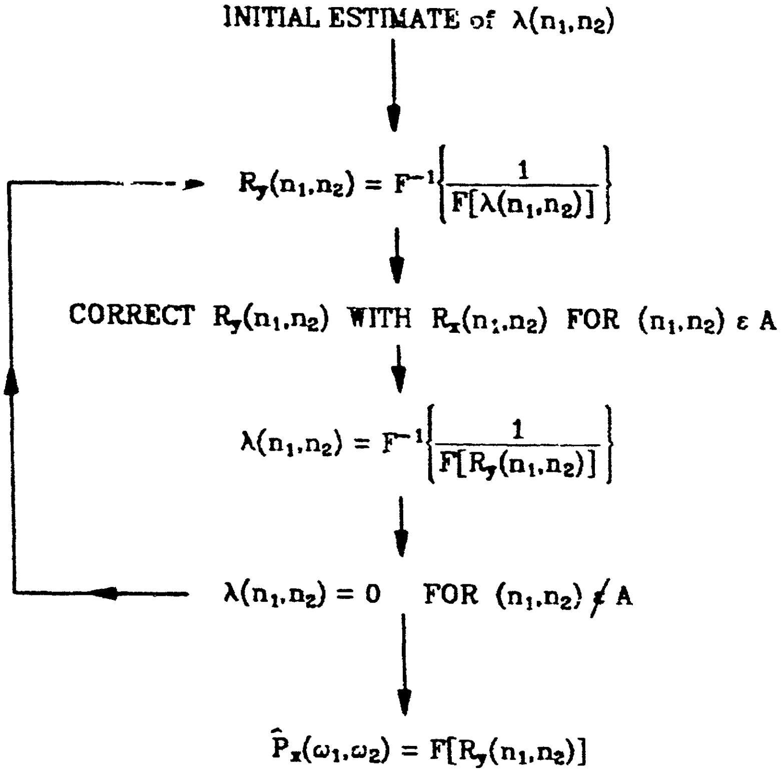

INITIAL ESTIMATE of $\lambda(n_1,n_2)$
$R_y(n_1,n_2) = F^{-1}\left\{\frac{1}{F[\lambda(n_1,n_2)]}\right\}$
CORRECT $R_y(n_1,n_2)$ WITH $R_x(n_1,n_2)$ FOR $(n_1,n_2) \varepsilon A$
$\lambda(n_1,n_2) = F^{-1}\left\{\frac{1}{F[R_y(n_1,n_2)]}\right\}$
$\lambda(n_1,n_2) = 0$ FOR $(n_1,n_2) \not\varepsilon A$
$\hat{P}_x(\omega_1,\omega_2) = F[R_y(n_1,n_2)]$

Fig. 3.1: A new approach to 2-D maximum entropy power spectrum estimation.

3.2 The Practical Algorithm

The conceptual algorithm outlined above and illustrated in Fig. 3.1 cannot, in general, be used directly to obtain the ME PS estimates. Issues such as the spectral zero-crossing problem, choice of adequate DFT length, error criterion for convergence decisions etc. arise, and require elucidation before the algorithm can assume a practical form. These issues are discussed below.

3.2.1 The Spectral Zero Crossing Problem

The algorithm shown in Fig. 3.1 requires two inversions of the spectral estimates in each iteration, and thus the iterations cannot be continued if the power spectrum estimate (or the inverse power spectrum) has any zero crossings at any stage in the iterative procedure. Unfortunately, zero crossings can occur in two different ways in each iteration. One is the truncation of the coefficients and the other is the correction of the ACF. To see this, let $\lambda^m(n_1,n_2)$ and $R_y^m(n_1,n_2)$ represent $\lambda(n_1,n_2)$ and $R_y(n_1,n_2)$ after the m'th iteration, and suppose that the following conditions hold:

$$F[\lambda^m(n_1,n_2)] > 0 \quad \text{for all } (\omega_1,\omega_2) \tag{3.5}$$

$$F[R_y^m(n_1,n_2)] > 0 \quad \text{for all } (\omega_1,\omega_2) \tag{3.6}$$

and

$$\lambda^m(n_1,n_2) = F^{-1}\left[\frac{1}{F[R_y^m(n_1,n_2)]}\right] w(n_1,n_2) \tag{3.7}$$

where $w(n_1,n_2)$ represents a rectangular type window such that

$$w(n_1,n_2) = \begin{cases} 1 & \text{for } (n_1,n_2) \in A \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Similarly, let $\lambda^{m+1}(n_1,n_2)$ and $R_y^{m+1}(n_1,n_2)$ represent $\lambda(n_1,n_2)$ and $R_y(n_1,n_2)$ after the m+1'th iteration. In the iterative algorithm of Fig. 3.1, $\lambda^{m+1}(n_1,n_2)$ and $R_y^{m+1}(n_1,n_2)$ are obtained from $\lambda^m(n_1,n_2)$ by

$$R'(n_1,n_2) = F^{-1}\left[\frac{1}{F[\lambda^m(n_1,n_2)]}\right] \tag{3.9}$$

$$\begin{aligned} R_y^{m+1}(n_1,n_2) &= \begin{cases} R_x(n_1,n_2) & \text{for } (n_1,n_2) \in A \\ R'(n_1,n_2) & \text{otherwise} \end{cases} \\ &= R'(n_1,n_2) + [R_x(n_1,n_2) - R'(n_1,n_2)]\,w(n_1,n_2) \end{aligned} \tag{3.10}$$

$$\lambda'(n_1,n_2) = F^{-1}\left[\frac{1}{F[R^{m+1}{}^{y}(n_1,n_2)]}\right] \tag{3.11}$$

and

$$\lambda^{m+1}(n_1,n_2) = \begin{cases} \lambda'(n_1,n_2) & \text{for } (n_1,n_2) \in A \\ 0 & \text{otherwise} \end{cases}$$

$$= \lambda'(n_1,n_2).w(n_1,n_2) \qquad (3.12)$$

From eq.(3.9) it is clear that $R'(n_1,n_2)$ is positive definite since $\lambda^m(n_1,n_2)$ is assumed to be positive definite. However, $R_y^{m+1}(n_1,n_2)$ may not be positive definite due to the rectangular windowing $w(n_1,n_2)$ in eq.(3.10). Furthermore, even if $R_y^{m+1}(n_1,n_2)$ were positive definite so that $\lambda'(n_1,n_2)$ is positive definite, $\lambda^{m+1}(n_1,n_2)$ may not be positive definite due to $w(n_1,n_2)$ in eq.(3.12).

In order to ensure that the resulting $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ are positive definite so that the iterations can be continued, we make modifications to eq.(3.10) and eq.(3.12). Specifically, suppose that $R_y^{m+1}(n_1,n_2)$ is obtained by using a relaxation parameter α to linearly interpolate between $R'(n_1,n_2)$ and the known values $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$, and suppose that $\lambda^{m+1}(n_1,n_2)$ is obtained by linearly interpolating between $\lambda'(n_1,n_2).w(n_1,n_2)$ and $\lambda^m(n_1,n_2)$ via the relaxation parameter β. Then, in the modified iterative algorithm, $\lambda^{m+1}(n_1,n_2)$ and $R_y^{m+1}(n_1,n_2)$ are obtained from $\lambda^m(n_1,n_2)$ by

$$R'(n_1,n_2) = F^{-1}\left[\frac{1}{F[\lambda^m(n_1,n_2)]}\right] \qquad (3.13)$$

$$R_y^{m+1}(n_1,n_2) = \begin{cases} \alpha.R'(n_1,n_2) + (1-\alpha).R_x(n_1,n_2) & \text{for } (n_1,n_2) \in A \\ R'(n_1,n_2) & \text{otherwise} \end{cases}$$

$$= R'(n_1,n_2) + (1-\alpha).[R_x(n_1,n_2) - R'(n_1,n_2)].w(n_1,n_2) \qquad (3.14)$$

$$\lambda'(n_1,n_2) = F^{-1}\left[\frac{1}{F[R_y^{m+1}(n_1,n_2)]}\right] \qquad (3.15)$$

and

$$\lambda^{m+1}(n_1,n_2) = \beta.\lambda^m(n_1,n_2) + (1-\beta).\lambda'(n_1,n_2).w(n_1,n_2) \qquad (3.16)$$

Comparing eq.(3.10) and eq.(3.14), the latter reduces to the former when $\alpha = 0$. With any other choice of α eq.(3.14) represents a nonideal correction of

$R'(n_1,n_2)$ with the known values $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$, with a larger deviation of α from zero corresponding to a more nonideal correction. The important consequence of introducing the relaxation parameters α and β is that the resulting $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ can be guaranteed to be positive definite with the proper choice of these parameters. This can be seen by noting that $\lambda^m(n_1,n_2)$ and therefore $R'(n_1,n_2)$ are assumed to be positive definite. Hence, by choosing α sufficiently close to unity, $R_y^{m+1}(n_1,n_2)$ can be brought arbitrarily close to $R'(n_1,n_2)$. Similarly, eq.(3.16) reduces to eq.(3.12) when $\beta = 0$. With any other choice of β, $\lambda^{m+1}(n_1,n_2)$ now corresponds to an autocorrelation function which is a kind of "parallel resistor average" of $R_y^m(n_1,n_2)$ and $R_y^{m+1}(n_1,n_2)$. With a proper choice of β, $\lambda^{m+1}(n_1,n_2)$ can also be guaranteed to be positive definite, which can be seen by noting that $\lambda^m(n_1,n_2)$ is assumed to be positive definite and by considering β sufficiently close to 1 so that $\lambda^{m+1}(n_1,n_2)$ can be brought arbitrarily close to $\lambda^m(n_1,n_2)$. Therefore, by correctly choosing α and β in the ranges $0 \leq \alpha < 1$, $0 \leq \beta < 1$, $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ can be guaranteed to be positive definite, and thus the spectral zero crossing problem can be avoided and the iterations continued.

From eqs.(3.9)-(3.12), it is clear that if $\lambda^m(n_1,n_2)$ and $R_y^m(n_1,n_2)$ satisfy eqs.(3.5)-(3.7), then so do $\lambda^{m+1}(n_1,n_2)$ and $R_y^{m+1}(n_1,n_2)$ obtained by the modified iterative algorithm. If $\lambda^0(n_1,n_2)$ and $R_y^0(n_1,n_2)$, the initial estimates of $\lambda(n_1,n_2)$ and $R_y(n_1,n_2)$, are chosen to satisfy eqs.(3.5)-(3.7), then with a proper choice of the relaxation parameters α and β, the iterations specified by eqs.(3.5)-(3.8) and eqs.(3.13)-(3.16) form an iterative algorithm. This algorithm is shown in Fig. 3.2.

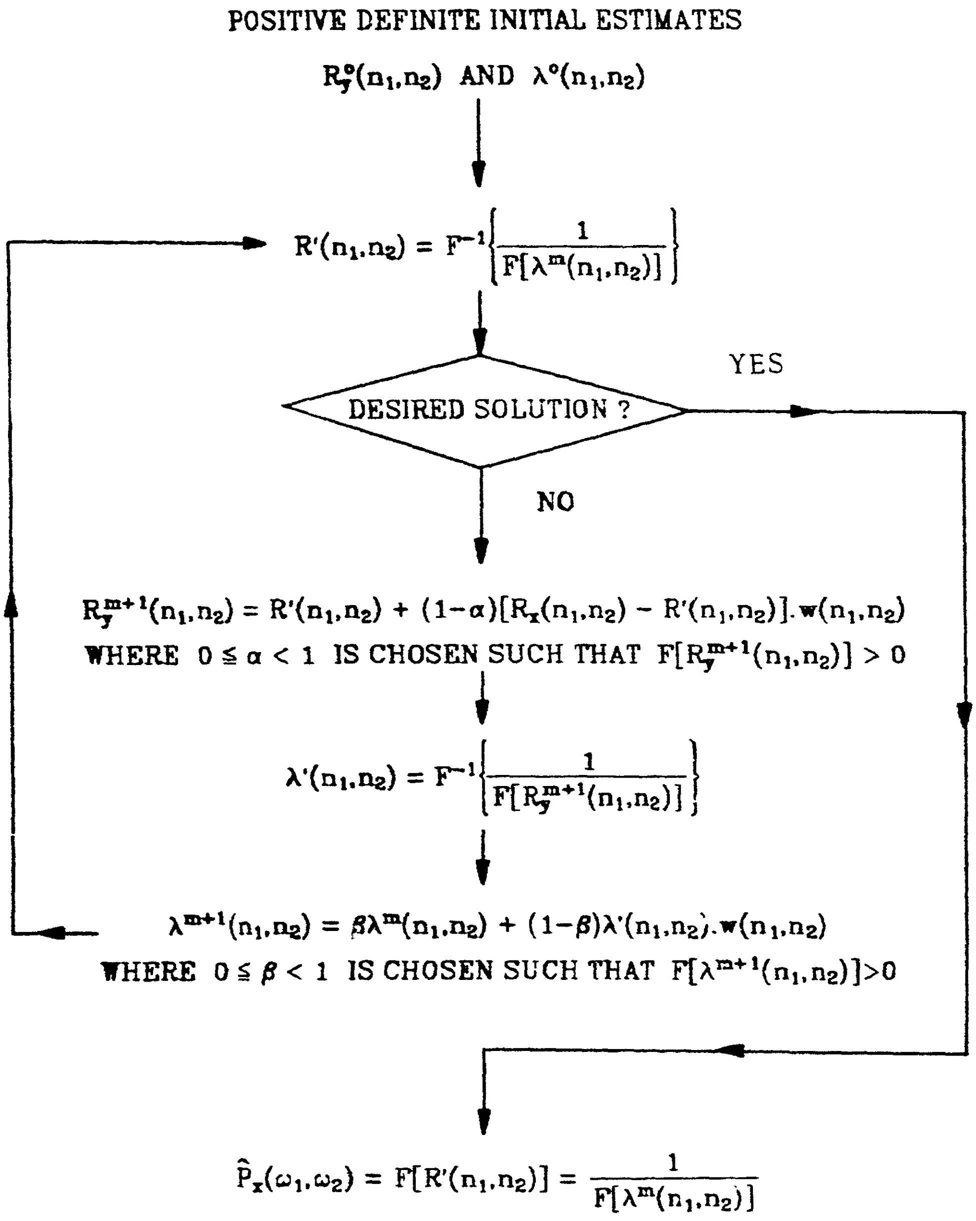

POSITIVE DEFINITE INITIAL ESTIMATES
$R_y^o(n_1,n_2)$ AND $\lambda^o(n_1,n_2)$
$R'(n_1,n_2) = F^{-1}\left\{\frac{1}{F[\lambda^m(n_1,n_2)]}\right\}$
DESIRED SOLUTION ?
YES
NO
$R_y^{m+1}(n_1,n_2) = R'(n_1,n_2) + (1-\alpha)[R_x(n_1,n_2) - R'(n_1,n_2)].w(n_1,n_2)$
WHERE $0 \leq \alpha < 1$ IS CHOSEN SUCH THAT $F[R_y^{m+1}(n_1,n_2)] > 0$
$\lambda'(n_1,n_2) = F^{-1}\left\{\frac{1}{F[R_y^{m+1}(n_1,n_2)]}\right\}$
$\lambda^{m+1}(n_1,n_2) = \beta\lambda^m(n_1,n_2) + (1-\beta)\lambda'(n_1,n_2).w(n_1,n_2)$
WHERE $0 \leq \beta < 1$ IS CHOSEN SUCH THAT $F[\lambda^{m+1}(n_1,n_2)]>0$
$\hat{P}_x(\omega_1,\omega_2) = F[R'(n_1,n_2)] = \frac{1}{F[\lambda^m(n_1,n_2)]}$

Fig. 3.2: An iterative algorithm for 2-D ME PSE based on Fig. 3.1.

3.2.2 The Relaxation Parameters

The relaxation parameters α and β were introduced above to transform the conceptual algorithm of Fig. 3.1 into the implementable algorithm of Fig. 3.2. The choice of α and β is dictated by two considerations. One is the requirement that the resulting $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ be positive definite. The second is our desire to choose α and β as close to zero as possible so that the maximum amount of correction is made at each iteration. In this regard, it has been empirically observed that choosing the smallest possible values of α and β consistent with the positive definite requirement on $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ can lead to a limit cycle behavior where the algorithm does not converge. A similar behavior has also been observed to occur if the value of α is decreased adaptively, over the course of the iterations. Further, it has been observed that the correction rates of the correlation function and the coefficients must be decreased if the normalized square error as defined in Sec. 3.4 below, between $R'(n_1,n_2)$ and $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$ increases from one iteration to the next.

In light of the above observations, the following method has been developed to obtain the values of α and β in the course of the iterations.

The initial values are chosen to be $\alpha_0 = \beta_0 = 0$. These values are updated in the following manner;

$$\alpha_{m+1} = \max\left[\alpha_m,\ 1-k.\frac{\min\limits_{(\omega_1,\omega_2)} F[R'(n_1,n_2)]}{\left|\min\limits_{(\omega_1,\omega_2)} F[(R_x(n_1,n_2)-R'(n_1,n_2)).w(n_1,n_2)]\right|}\right] \tag{3.17}$$

The denominator of the second term in eq.3.17 is simply the minimum of the correction spectrum, that is, the Fourier transform of the correction ACF. Since α is used to interpolate between the numerator, which is the spectral estimate at the m'th iteration, and the denominator, the term represents the maximum deviation from unity that can be used for α. This value is then scaled by the factor 'k', in keeping with the empirical observation that the minimum value of α is not the ideal value. Similarly,

$$\beta_{m+1} = \left[1 + (1-k).\left[\frac{1}{\beta_{\min}} - 1\right]\right] \cdot \beta_{\min} \qquad (3.18)$$

$$\text{if } \min_{(\omega_1,\omega_2)} F[\lambda'(n_1,n_2).w(n_1,n_2)] < 0$$

and

$$\beta_{m+1} = 0 \qquad \text{otherwise}$$

where

$$\beta_{\min} \triangleq \frac{\left| \min_{(\omega_1,\omega_2)} F[\lambda'(n_1,n_2).w(n_1,n_2)] \right|}{\min_{(\omega_1,\omega_2)} F[\lambda^m(n_1,n_2)] + \left| \min_{(\omega_1,\omega_2)} F[\lambda'(n_1,n_2).w(n_1,n_2)] \right|}$$

In eqs.(3.17) and (3.18), α_i and β_i represent the values of α and β in the i'th iteration; $\beta_{\min}$ represents the minimum value of β that results in a positive definite estimate for the coefficient set; max[,] represents the maximum of two arguments, $\min_{(\omega_1,\omega_2)}[]$ represents the minimum of the argument expression over (ω_1,ω_2), and "k" is the convergence rate parameter which governs how close α and β are to their minimum (ideal) values. The initial value of "k" is chosen to be moderately large ($k \simeq 0.5$) and then subsequently reduced if the error between $R'(n_1,n_2)$ and $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$ increases. Thus, the algorithm moves towards the desired solution rapidly at first, and, if necessary, it is slowed down as convergence is approached. When α and β are chosen according to eqs.(3.17) and (3.18), it is straightforward to show from eqs.(3.14) and (3.16) that the resulting $R_y^{m+1}(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ are guaranteed to be positive definite. Further, computing α_{m+1} and β_{m+1} by eqs.(3.17) and (3.18) requires little extra computation since the individual terms in the two equations are available in the course of the iterations.

3.3 Choice Of DFT Length

The ME method of PSE can be viewed as an attempt to extrapolate the correlation function beyond the limits of the known segment. The algorithm described above uses the Fourier transform to perform this extrapolation. Since the DFT is used in the implementation instead of the true Fourier transform, the length of the DFT used should be chosen such that the extended correlation function corresponding to the ME PS estimate is essentially zero beyond the DFT limits. If the DFT length is too short, the ACF estimates at each iteration will undergo severe aliasing and the true solution may not be obtained. Typically, the requirement that the estimated power spectrum be consistent with the known ACF values, cannot be achieved, and the ACF matching error as defined in Sec. 3.4 below, remains high. Another possibility that may result when too short a DFT length is used is that although the desired error level is achieved, the resulting coefficient set may not be positive-definite. This is because the DFT samples the true Fourier transform, and hence the DFT values obtained during the course of the iterations may be all positive while the true Fourier Transform of the coefficients, which is the reciprocal of the power spectral estimate, may not be positive for all frequency values. In this case, the coefficients cannot be used to form an acceptable PS estimate. On the other hand, using too large a DFT length will involve unnecessary computation. However, it is clear that if an error is to be made in choosing the DFT length, the error must be made in the direction of over estimation.

The choice of the proper DFT length is discussed further in Sec. 3.4 below and the length requirement as a function of S/N ratio is discussed in Chap. 5

3.4 Error Criterion

Another important issue to be considered in implementing the algorithm shown in Fig. 3.2 is the decision on when the algorithm converged so that the iterations can be stopped. Recall that the ME method imposes the consistency constraint on the ACF values and the finite support constraint on the coefficient set. The constraint on the coefficients is imposed by the algorithm at each iteration. On the other hand, the ACF is gradually corrected to achieve consistency with the known values. Thus, a reasonable approach is to consider that the algorithm has converged when the following condition is satisfied:

$$\frac{\sum\limits_{(n_1,n_2)\in A}\sum [R'(n_1,n_2) - R_x(n_1,n_2)]^2}{\sum\limits_{(n_1,n_2)\in A}\sum R_x^2(n_1,n_2)} \leq \varepsilon \tag{3.19}$$

Clearly, if $\varepsilon = 0$ with $R'(n_1,n_2)$ computed from $\lambda^m(n_1,n_2)$ using the discrete time Fourier transform rather than the DFT, the resulting solution corresponds to the ME PS estimate. However, due to a finite DFT length and finite precision arithmetic used, it may not be possible to reduce the error exactly to zero. On the other hand, the use of a short DFT length may reduce the error to a very small value without leading to the desired ME PS estimate. This again brings into sharp focus the fact that the DFT length must not be underestimated in implementing the algorithm. However, to avoid unnecessary computations, the algorithm can be started using a reasonable DFT length and a one-time test for the solution made at the end. Specifically, with a reasonable choice of the DFT length, the iterations are continued until the error ε reaches a very small value, typically 10^{-4}. If this error level cannot be attained, then the DFT length has to be increased. The coefficient set $\{\lambda\}$ obtained as the ME solution is then tested for positive definiteness over a much finer grid (much larger DFT length) than that used in the iterations. If the solution is not positive definite, the use of a longer DFT length is indicated. If the coefficient set λ is positive definite, then the error ε given by eq.(3.19) is rechecked by computing $R'(n_1,n_2)$ using a much longer DFT. If the new error is of the same order as that obtained during the iterations, the solution is declared to be good; otherwise, more iterations are required. Since the minimum error that can be achieved with a given DFT length is dictated by the amount of aliasing that is undergone by the ACF, if it becomes necessary to continue the iterations after the test, it is preferable to use a longer DFT length than that used in the iterations.

Fig. 3.3 shows a more detailed flowchart of the algorithm which incorporates the important implementation issues discussed above. It is not theoretically known under what conditions the algorithm in Fig. 3.3 converges. However, it has been observed empirically that the algorithm always converges to the ME PS estimate in the sense that the requisite constraints on the coefficients and the ACF values are satisfied, the former exactly and the latter to within the error criterion ε specified, when a sufficiently large DFT length is used. Further, the error has been observed to decrease very rapidly in the course of the first few iterations, and reasonable estimates of the power spectrum can usually be formed fairly rapidly.

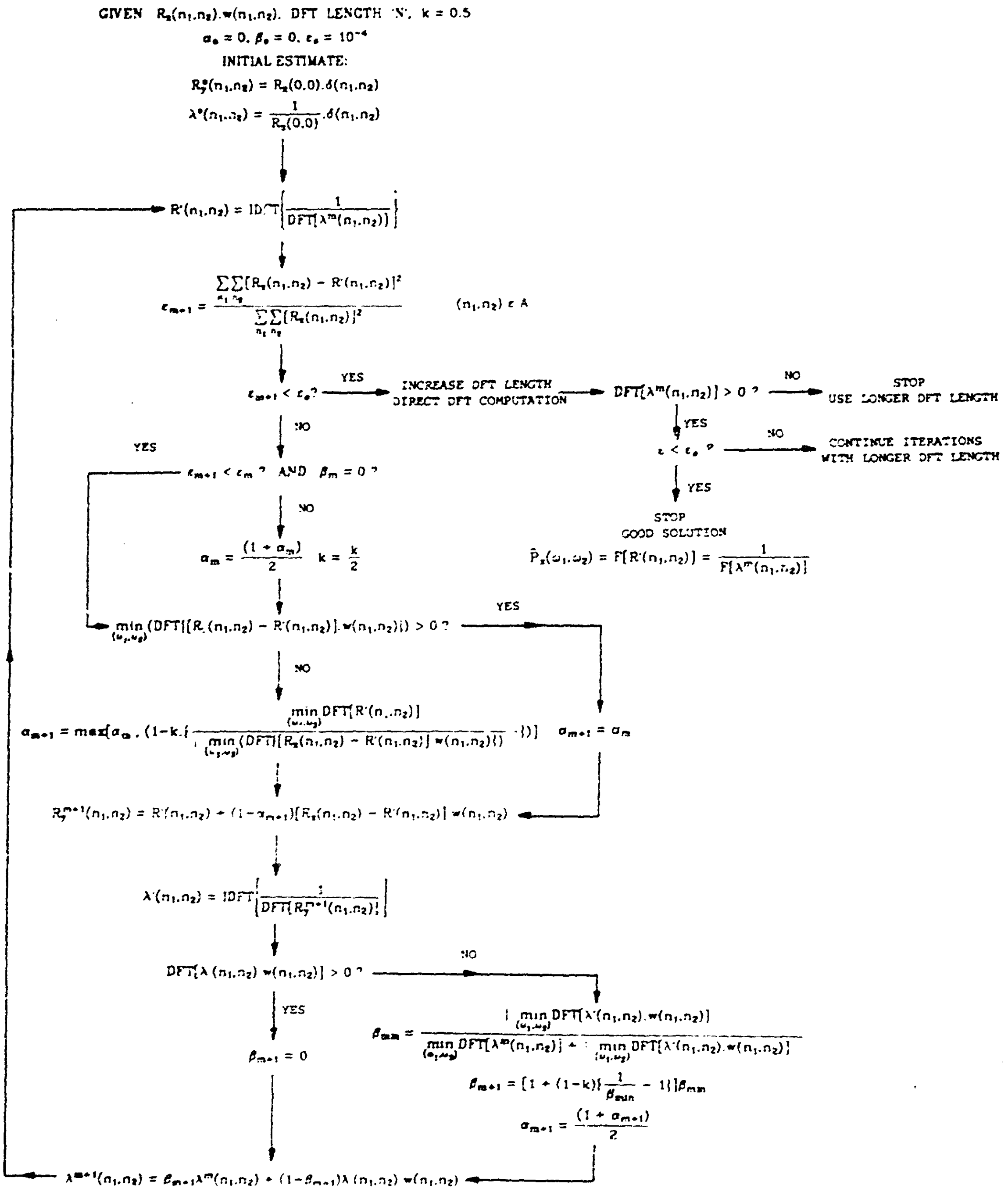

GIVEN $R_x(n_1,n_2).w(n_1,n_2)$, DFT LENGTH 'N', $k = 0.5$
$\alpha_0 = 0,\ \beta_0 = 0,\ \epsilon_0 = 10^{-4}$
INITIAL ESTIMATE:
$R_y^0(n_1,n_2) = R_x(0,0).\delta(n_1,n_2)$
$\lambda^0(n_1,n_2) = \frac{1}{R_x(0,0)}.\delta(n_1,n_2)$
$R'(n_1,n_2) = \mathrm{IDFT}\left\{\frac{1}{\mathrm{DFT}[\lambda^m(n_1,n_2)]}\right\}$
$\epsilon_{m+1} = \frac{\sum_{n_1}\sum_{n_2}[R_x(n_1,n_2) - R'(n_1,n_2)]^2}{\sum_{n_1}\sum_{n_2}[R_x(n_1,n_2)]^2}$
$(n_1,n_2) \epsilon A$
$\epsilon_{m+1} < \epsilon_0$?
YES
INCREASE DFT LENGTH
DIRECT DFT COMPUTATION
$\mathrm{DFT}[\lambda^m(n_1,n_2)] > 0$?
NO
STOP
USE LONGER DFT LENGTH
YES
$\epsilon < \epsilon_0$?
NO
CONTINUE ITERATIONS
WITH LONGER DFT LENGTH
YES
STOP
GOOD SOLUTION
$\hat{P}_x(\omega_1,\omega_2) = F[R'(n_1,n_2)] = \frac{1}{F[\lambda^m(n_1,n_2)]}$
NO
YES
$\epsilon_{m+1} < \epsilon_m$? AND $\beta_m = 0$?
NO
$\alpha_m = \frac{(1+\alpha_m)}{2}$ $k = \frac{k}{2}$
$\min_{(\omega_1,\omega_2)}(\mathrm{DFT}\{[R_x(n_1,n_2) - R'(n_1,n_2)].w(n_1,n_2)\}) > 0$?
YES
NO
$\alpha_{m+1} = \max[\alpha_m, (1-k.\{\frac{\min_{(\omega_1,\omega_2)}\mathrm{DFT}[R'(n_1,n_2)]}{\min_{(\omega_1,\omega_2)}(\mathrm{DFT}\{[R_x(n_1,n_2) - R'(n_1,n_2)]w(n_1,n_2)\})}\})]$
$\alpha_{m+1} = \alpha_m$
$R_y^{m+1}(n_1,n_2) = R'(n_1,n_2) + (1-\alpha_{m+1})[R_x(n_1,n_2) - R'(n_1,n_2)]w(n_1,n_2)$
$\lambda'(n_1,n_2) = \mathrm{IDFT}\left\{\frac{1}{\mathrm{DFT}[R_y^{m+1}(n_1,n_2)]}\right\}$
$\mathrm{DFT}[\lambda'(n_1,n_2)w(n_1,n_2)] > 0$?
NO
YES
$\beta_{m+1} = 0$
$\beta_{min} = \frac{|\min_{(\omega_1,\omega_2)}\mathrm{DFT}[\lambda'(n_1,n_2).w(n_1,n_2)]|}{\min_{(\omega_1,\omega_2)}\mathrm{DFT}[\lambda^m(n_1,n_2)] + |\min_{(\omega_1,\omega_2)}\mathrm{DFT}[\lambda'(n_1,n_2).w(n_1,n_2)]|}$
$\beta_{m+1} = [1 + (1-k)\{\frac{1}{\beta_{min}} - 1\}]\beta_{min}$
$\alpha_{m+1} = \frac{(1+\alpha_{m+1})}{2}$
$\lambda^{m+1}(n_1,n_2) = \beta_{m+1}\lambda^m(n_1,n_2) + (1-\beta_{m+1})\lambda'(n_1,n_2)w(n_1,n_2)$

Fig. 3.3: A detailed flowchart of the new iterative algorithm for 2-D ME power spectrum estimation.

3.5 Convergence Issues

The algorithm developed in this chapter is an empirical procedure for solving a highly non-linear problem. Although no proof of convergence is available so far, the algorithm has not failed to converge to the desired solution in a very large number of both one and two dimensional examples that have been tried so far.

The algorithm is an 'alternating-projection' type of algorithm. However, it is quite different from other alternating-projection algorithms such as those proposed by Papoulis[10] or by Gerchberg.[15] The two domains that the algorithm iterates between are the correlation domain and the domain of the autoregressive coefficients. Thus, the two domains are both 'time' domains rather than the 'time' and frequency domains of the other algorithms. Further, going from one domain to the other requires the obtaining of a spectrum (the frequency domain) and its reciprocal. This is what sets the algorithm apart from the others, and causes problems in trying to prove convergence. For example, all the operations performed locally on either the autocorrelation function or the coefficients can be shown to be non-expansive mappings. However, when complete mappings across one iteration are considered, it is no longer possible to show their non-expansiveness due to the reciprocal operation on the spectrum. Thus, it has not been possible to show that obtaining a new ACF estimate from the previous one through one complete iteration, is a non-expansive mapping. A similar situation holds for the filter coefficients. Given that the Maixmum Entropy solution exists and is unique, showing the non-expansiveness of either mapping would be sufficient to provide a proof for the convergence of the algorithm.

Chapter 4

Implementation Strategies

4.1 Introduction

A new algorithm for 2-D ME PSE was developed in Chap. 3. Several alternate strategies are available for the implementation of this algorithm. These strategies are dependent on the type and size of computer employed and make trade-offs between execution speed and on-line and off-line storage or memory requirements.

In comparing the different strategies it is simplest to consider the conceptual algorithm of Fig. 3.1. The salient features of the comparison that follows are then directly applicable to the practical algorithm depicted in Fig. 3.3. It is clear from Fig. 3.1 that the major expense involved in implementing the algorithm is the computation of the forward and inverse Fourier transforms. In the actual implementation, the true Fourier transform operations are approximated by the Discrete Fourier Transform (DFT). The strategies discussed below differ primarily in the specific method used to compute the 2-D DFT's. Three approaches are outlined: one which uses a conventional 2-D Fast Fourier Transform (FFT) algorithm to obtain the desired 2-D DFT's, another which makes more efficient use of 1-D FFT's to obtain the 2-D DFT's and the last which computes the 2-D DFT's by direct computation.

4.2 Main Frame Computer with Unlimited On-Line Storage

4.2.1 Conventional 2-D FFT Approach

When the algorithm is implemented on a main-frame (large) computer with unlimited on-line storage, the FFT algorithm can be used to compute the 2-D DFT's on-line and in place. The 2-D DFT is conventionally obtained by computing the 1-D FFT of the rows of the data, followed by the transforms of the columns, that is, for an FFT size of NxN, one needs to perform 2N N-length FFT's, each requiring on the order of NlogN operations. Thus to obtain a forward or inverse 2-D DFT of size NxN, $2N^2\log N$ operations are required. The resulting arrays are stored on-line.

4.2.2 Efficient use of the FFT

The use of the FFT outlined above, is simple to implement and is the most straightforward. However, it is an inefficient use of the FFT. A cursory study of the algorithm depicted in Fig. 3.2 reveals that in the forward transform operations, the starting array size is always MxM, which is the size of the region over which the ACF is known. In the case of the coefficient set $\{\lambda\}$ this is obvious. On the other hand, in the case of the ACF, since the transform of the current ACF estimate has been computed, only the transform of the correction ACF, $[R_x(n_1,n_2) - R'(n_1,n_2)].w(n_1,n_2)$ needs to be computed to update the spectral estimate. Thus, the starting data array in either case is of size MxM, and one needs to perform only (M+N) FFT's to obtain the entire 2-D DFT.

Similarly, in the case of the inverse transform operation, only an MxM array needs to be computed from the NxN spectrum arrays: the ACF estimate is required only over the region 'A' for correction, and the coefficient array size is also only MxM. Thus the inverse transform operations also require only (M+N) inverse FFT's, or on the order of (M+N)NlogN operations per inverse 2-D DFT.

Typically, the size of the DFT needed is much larger than the size of the ACF array, i.e. N>>M. Thus the efficient use of the FFT requires substantially fewer operations than the conventional 2-D FFT approach. The on-line storage requirements are the same for the two approaches, and no off-line or disk storage is used.

4.3 Implementations for Mini-Computer

When implementing the algorithm on a mini-computer with limited on-line storage, it becomes necessary to use off-line or disk storage. Disk access times are typically much longer than machine cycle times, and therefore the amount of disk access demanded by a particular implementation strategy should be kept in mind.

There are two possible alternatives in the mini-computer implementation. One is to use the efficient FFT approach outlined above for the main-frame computers, and the other is to use a direct computation of the DFT's.

4.3.1 FFT Implementation

The FFT implementation on a mini-computer is basically the same as for a main-frame machine, except that for large DFT sizes, it is not possible to store the spectrum arrays on-line. Disk, or off-line storage becomes necessary. As was mentioned earlier, disk access times are typically much longer than machine cycle times and hence, it is advantageous to minimize disk access. The method of obtaining a large NxN size 2-D FFT on a mini-computer is usually as follows. N rows of the data are transformed and written out to disk. The disk array is then transposed, read in row by row, transposed, and written out to disk again. A further transpose is then required to obtain the result in correct order Thus, apart from the 2N FFT's required, a total of (4N + 4NlogN) disk accesses are required, where the number of disk accesses required to transpose a NxN array is of the order of 2NlogN.[30] In order to avoid this excessive amount of disk access, it is necessary to look at alternate strategies.

The efficient use of the FFT outlined above required only (M+N) FFT's per forward or inverse transform. If a complex array of size MxN can be configured in core, then this approach can be implemented on a mini-computer with only N disk accesses required per 2-D transform operation. This can be seen as follows. For the forward transform, M rows of the ACF or the coefficient array are transformed and the results saved in the MxN complex array, in core. The transpose of the data is then affected on-line, and the resulting N columns are transformed via the FFT and written out to disk in transposed form, thereby requiring only N writes. Similarly the inverse transform requires only N disk reads to read in the columns of the spectra (the array is in transposed form). Each column is inverse transformed and only the first M values are stored in the intermediate MxN array. M inverse transforms then result in the required MxM ACF array or the new coefficient estimates.

This implementation via the FFT is, of course, limited by the size of the available memory and the values of M (the ACF size) and N (the DFT size). However, the implementation is highly efficient, and although disk storage of size NxN is required for each spectrum array, the number of disk accesses is negligibly small.

4.3.2 Direct DFT Implementation

When the size MxM of the known ACF array (the region 'A') becomes large, or when the S/N ratio is high, DFT's and IDFT's of large size are typically required to implement the algorithm, and the efficient use of the FFT as discussed in section 4.3.1 above, is no longer feasible. It is then possible to fall back onto the conventional method of performing 2-D FFT's. However, the number of disk accesses involved in this approach are excessive, and hence the execution speed suffers. Further, the amount of disk storage required may also become prohibitive for modest sized systems. In this case, an alternative implementation is possible: compute the 2-D DFT by direct computation. This technique, described below, is specifically tailored towards the limitations of small mini-computer systems.

Referring back to Fig. 3.1, a careful observation shows that the direct computation of the DFT's and IDFT's does not significantly increase the computational burden. Specifically, let $\lambda^m(n_1,n_2)$ and $\lambda^{m+1}(n_1,n_2)$ represent the coefficient set $\{\lambda\}$ after the m'th and m+1'th iteration respectively. Using this notation, it is straightforward to show from Fig. 3.1 that

$\lambda^{m+1}(n_1,n_2)$ is related to $\lambda^m(n_1,n_2)$ by

$$\lambda^{m+1}(n_1,n_2) = F^{-1}\left[\frac{1}{\dfrac{1}{F[\lambda^m(n_1,n_2)]} + F[\{R_x(n_1,n_2) - R_y'(n_1,n_2)\}.w(n_1,n_2)]}\right].w(n_1,n_2) \tag{4.1}$$

where

$$R_y'(n_1,n_2) = F^{-1}\left[\frac{1}{F[\lambda^m(n_1,n_2)]}\right] \tag{4.2}$$

and

$$w(n_1,n_2) = \begin{cases} 1 & \text{for } (n_1,n_2) \in A \\ 0 & \text{otherwise} \end{cases} \qquad (4.3)$$

From eq.(4.1), $\lambda^{m+1}(n_1,n_2)$, $\lambda^{m}(n_1,n_2)$ and $\{R_x(n_1,n_2) - R_y'(n_1,n_2)\}.w(n_1,n_2)$ are finite extent sequences of size 'A'. As a consequence, directly computing each of the DFT's and IDFT's in eq.(4.1) requires on the order of M^2N^2 arithmetic operations where, as above, the size of 'A' is MxM and the DFT size is NxN. The direct 2-D FFT approach would require $2N^2\log N$ operations. Since N is typically much larger than M, direct computation does not significantly increase the number of arithmetic operations relative to using an FFT algorithm. As an example, when the size of 'A' is 5x5 and the DFT size is 512x512, direct computation requires about 40 percent additional arithmetic operations relative to using an FFT algorithm. However, it should be remembered that the FFT approach would require a large amount of disk access which could conceivably offset the computational advantage.

Although the comparison between the different implementation strategies has been made on the basis of the conceptual algorithm, it should be pointed out that in the practical implementation of the algorithm as shown in figure 3.3, the minimum value of the Fourier transform of the correction ACF is required to compute the value of the relaxation parameter α. Since α must be evaluated before the updated spectrum can be obtained, it becomes necessary to compute one extra two dimensional Fourier Transform as compared to the implementations which store the spectrum. Thus, the direct implementation is slightly more expensive computationally than a simple comparison between the various 2-D DFT computations would show.

The major advantage of direct computation is the significant reduction in the amount of memory required. Whereas the FFT algorithm requires $N^2 + M^2$

memory locations (some of which would be off-line for the mini-computer), the direct computation only requires memory locations on the order M^2, which are easily configured on-line. Additional advantages of the direct computation approach include the potential to exploit parallel processing, and no restrictions on DFT size. In fact, the DFT size need not be a power of two as is usually required by most FFT routines.

The implementation strategies discussed above are compared in Table 4.1 below. The memory requirements common to all schemes are not listed.

TABLE 4.1

Comparison of implementation strategies for main-frame and mini-computers.

		OPERATIONS PER ITERATION	ON-LINE MEMORY	OFF-LINE MEMORY	DISK ACC. PER ITERATION
MAIN FRAME COMPUTER	CONVENTIONAL 2-D FFT	$8N^2$logN	$2N^2$	None	None
	EFFICIENT FFT	4(M+N)logN	$2N^2$	None	None
MINI COMPUTER	CONVENTIONAL 2-D FFT	$8N^2$logN	None	$2N^2$	16N(logN+1)
	EFFICIENT FFT	4(M+N)logN	MN	$2N^2$	4N
	DIRECT COMPUTATION	$9M^2N^2$	None	None	None

Chapter 5

Experimental Results

5.1 Introduction

The algorithm developed in Chapter 3 has been applied to the ME power spectrum estimation problem for 2-D sinusoids buried in white Gaussian noise. For one set of experiments, it is assumed that the exact correlation values are available over the region 'A'. The region 'A' unless otherwise noted, is taken to be a square, symmetric about the origin in all cases. For the case of M real sinusoids, the exact ACF values are given by

$$R_x(n_1,n_2) = \sum_{i=1}^{M} a_i^2 \, \mathrm{Cos}(\omega_{i1} n_1 + \omega_{i2} n_2) + \sigma^2 \delta(n_1,n_2) \tag{5.1}$$

where $R_x(n_1,n_2)$ represent the ACF values, a_i^2 is the power of the i'th sinusoid, ω_{i1} and ω_{i2} give its frequency location, and σ^2 represents the noise power. For the case of M complex exponentials the exact ACF is given by

$$R_x(n_1,n_2) = \sum_{i=1}^{M} a_i^2 \, e^{j(\omega_{i1} n_1 + \omega_{i2} n_2)} + \sigma^2 \delta(n_1,n_2) \tag{5.2}$$

For both eqs. (5.1) and (5.2), $R_x(n_1,n_2)$ is assumed known for $(n_1,n_2) \in A$.

A parallel set of experiments uses ACF values estimated from synthetic data sets. In this case, the ACF is obtained via the biased estimator. That is if the data is available in a square array of size PxP, the ACF is estimated as

$$R_x(n_1,n_2) = \frac{1}{P^2} \sum_{k_1=1}^{P} \sum_{k_2=1}^{P} x^*[k_1 + n_1 + k_2 + n_2].x[k_1,k_2] \tag{5.3}$$

where $x[n_1,n_2]$ represents the synthetic data set given by:

$$x[n_1,n_2] = \sum_{i=1}^{M} a_i \mathrm{Cos}(\omega_{i1}.n_1 + \omega_{i2}.n_2 + \varphi_i) + w[n_1,n_2] \tag{5.4}$$

where the number of sinusoids is M, $w[n_1,n_2]$ represents white noise of power σ^2, φ_i is a random phase term associated with the i'th sinusoid, and the sums in eq.(5.3) run over known data values only. For the case of complex data, the cosine in eq.(5.4) is replaced by a complex exponential.

The unbiased estimate for the ACF was not used, because it can result in a non positive-definite ACF estimate (that is, the Fourier Transform of the ACF may not be greater than zero for all frequencies). In such cases, Woods' theorem[2] no longer holds, and the existence and uniqueness of the ME PS estimates can no longer be guaranteed.

The 2-D spectra are displayed in the form of contour plots and a few comments about their display are in order. All spectra are displayed with the highest contour level being normalized to zero dB. Wherever the length of the contours permit, they are labeled with the nearest integer value of the contour level, in dB below the maximum (0dB). The contours are always equally spaced and the increment between contours (CINC), in dB, is always noted. On all plots, the true peak location is marked with an 'x'.

For real data, the power spectra are symmetric about the origin, and thus only half the 2-D frequency plane is displayed. The full 2-D plane is displayed for spectra of complex signals Finally, the frequency axes, and all frequency values are in terms of the normalized frequency units of $\frac{\omega}{2\pi}$. Thus, for example, the interval $(-\pi,\pi)$ is represented by (-0.5,0.5), and the peak location of $(\omega_1,\omega_2) = (0.2\pi,0.3\pi)$ is represented by the ordered pair (0.1,0.15).

All pertinent data about the spectrum being displayed is included with the plot. Thus, the power and both the actual and the estimated frequency locations of the peaks are noted. Also, the S/N ratio (or the noise power) as well as the size of the ACF support region is indicated. Where necessary, the shape of the ACF support region 'A' is also shown. In the case of the ME PS estimates, the number of iterations (NITR) required and the size of the DFT used (NDFT), is also displayed.

The 2-D ME power spectrum estimates are compared with the Maximum Likelihood (ML) and the Bartlett estimates. The ML estimate for 2-D signals is obtained by inverting the matrix of 2-D autocorrelations Φ_{NM} defined in section 2.2 of chapter II. The estimate is given by

$$P_{ML}(\omega_1,\omega_2) \;\underline{\Delta}\; \frac{NM}{E^{\dagger}\,\Phi_{NM}^{-1}\,E}$$

where Φ_{NM}^{-1} represents the inverse of the block Toeplitz matrix of autocorrelations and $E^{\dagger}$ is the complex conjugate transpose of the vector

$$E \;\underline{\Delta}\; \begin{array}{l} COL\{1, e^{-j\omega_1}, e^{-j2\omega_1}, \cdots, e^{-j(N-1)\omega_1}, e^{-j\omega_2}, \\ \cdots, e^{-j\{(N-1)\omega_1+\omega_2\}}, \cdots, e^{-j(M-1)\omega_2}, \\ \cdots, e^{-j\{(N-1)\omega_1+(M-1)\omega_2\}}) \end{array}$$

The Bartlett estimate is obtained by taking the DFT of the ACF values which are known over the region 'A'. The ACF is first windowed by a 2-D separable triangular window to prevent the spectrum from displaying negative regions.

5.2 Special Regions in the 2-D Frequency Plane

For 1-D real signals, it is well known[30,31] that the symmetry of the power spectrum causes errors in the peak location near $\omega = 0$ and the periodicity of the spectrum causes the same problem near $\omega = \pi$. This is due to the interference with the correlated mirror peaks that occur at negative frequencies for real data. For example, if the data is given by

$$x[n] \;=\; \mathrm{Sin}(\omega_o n) + \sigma^2\delta(n)$$

the power spectrum consists of two peaks located at $\omega = \omega_0$ and $\omega = -\omega_0$. If ω_0 is close in value to 0 or π, the interference between the peaks causes them to move closer in the PS estimate, initially causing errors in the location of the spectral peaks and eventually a complete merging of the two.

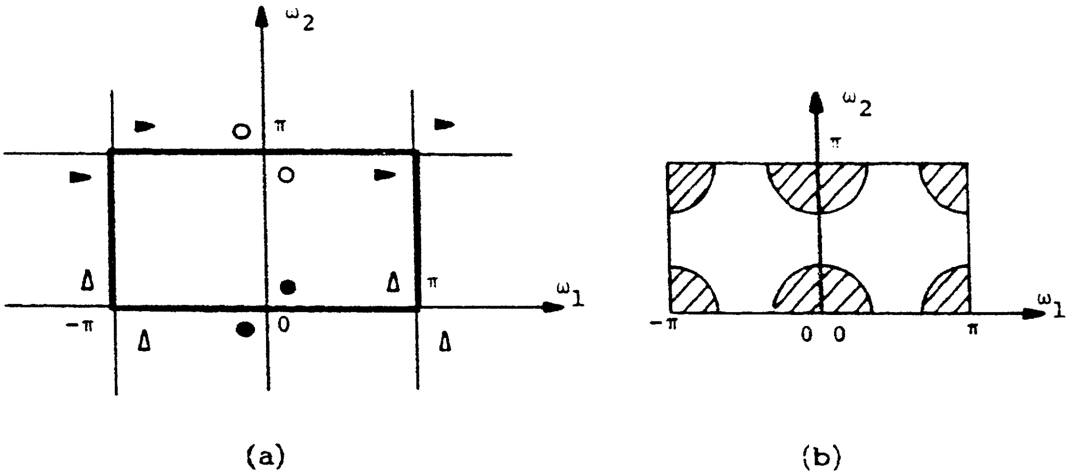


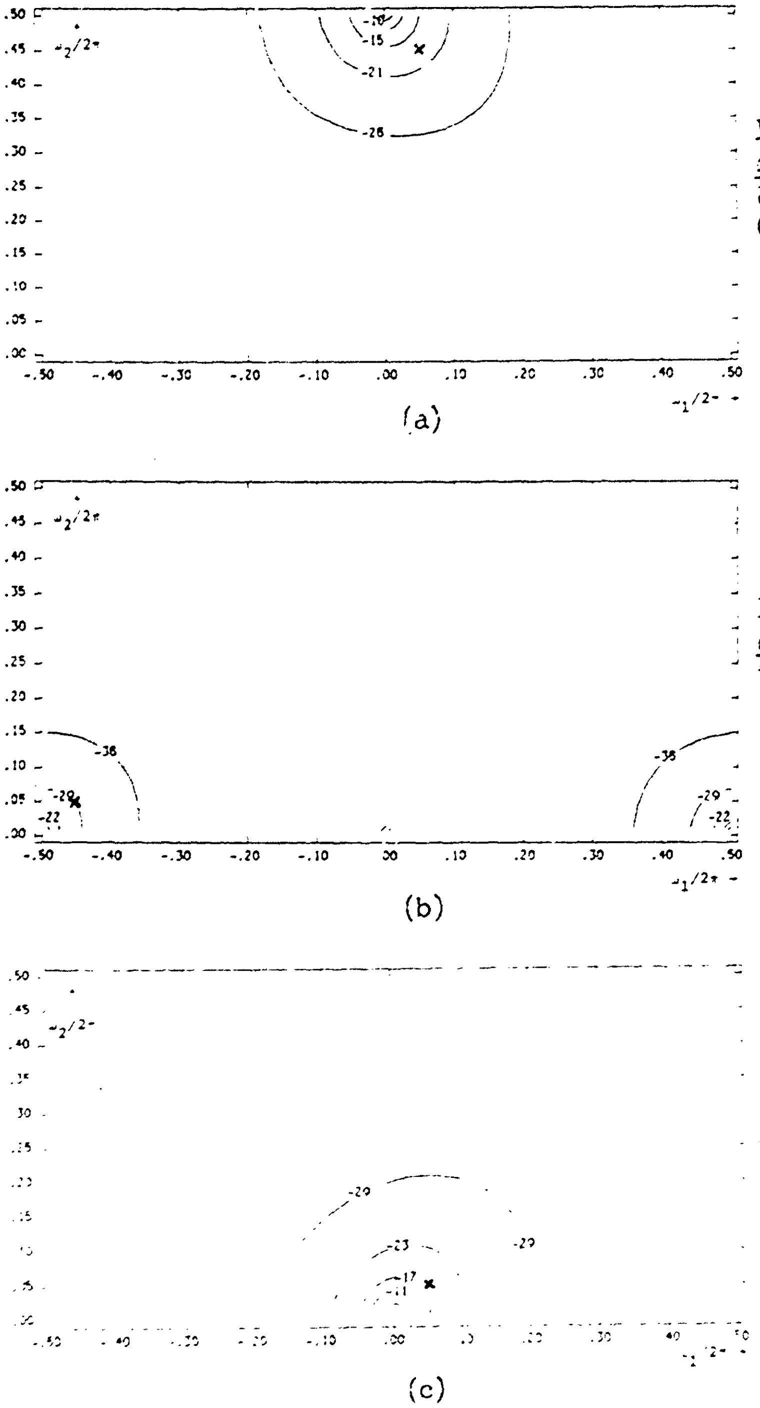
Fig. 5.1. The 2-D frequency plane for real data. (a) Symmetry and periodicity. (b) Special regions for real data

The case for 2-D real sinusoids was found to be similar except that the two-dimensional periodicity of the spectrum combined with its symmetry, results in errors in the peak location at several points in the 2-D frequency plane. Fig. 5.1(a) illustrates the symmetries and the location of the mirror peaks for the case of 2-D real sinusoids. The upper half plane, which completely specifies the power spectrum is indicated by bold lines, and the small geometric shapes show the locations of mirror peaks introduced by the symmetry and the periodicity. The shaded regions in Fig. 5.1(b) indicate the special regions in the upper half plane where the estimate of the peak location for real data can be expected to suffer.

This observation is supported by experimental data, as shown in Fig. 5.2. All the examples shown utilized a 3x3 region of support for the ACF (the region 'A'). For the ME method, Figs. 5.2(a) and (b) show that when the sinusoid is located within the special regions indicated in Fig. 5.1(b), the peak in the resulting power spectrum estimate gets pulled in towards the center of these regions, that is, towards $(\omega_1, \omega_2) = (0,0), (\pi,0), (\pi,\pi)$, or $(0,\pi)$. Figs. 5.2(c) and 5.2(d) show a similar result for the Maximum Likelihood and Bartlett methods. As the peak location moves out towards the centre of either quarter plane the single peak splits up into the desired pair of peaks, at first with erroneous peak location but eventually resulting in a good estimate of the power spectrum. Fig. 5.3 illustrates the ME PS estimate for one real sinusoid when the peak is located outside the special regions of Fig. 5.1(b).

Several examples have been obtained for the same ACF support but with different S/N ratios and the results are essentially the same for real sinusoids. As could be expected, the "region of resolution" increases in size with increasing S/N ratio, since the peaks in the estimate become sharper, thereby reducing the interference between mirror peaks.

Finally to verify that the errors in the peak location were indeed being caused by the interference of the mirror peaks in the real data, various examples using complex data were tried at the same S/N ratios and the ACF support region 'A' as were used for the previous examples. The results were similar to the 1-D case in that no special regions were discovered, (since there is no inherent symmetry in the power spectrum for complex data), and the resolution properties of the ME estimates were uniform throughout the



(a)

ME estimate; $\underline{\omega}=(0.1\pi,0.9\pi)$; 21 iterations,341 secs CPU time CINC = 5.32dB

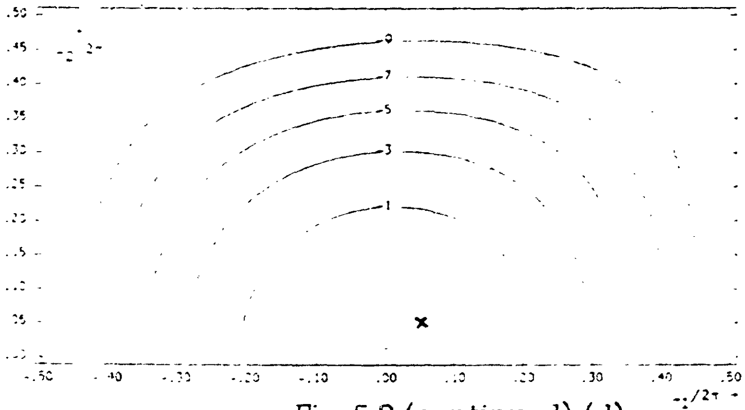
(b)

ME estimate; $\underline{\omega}=(-0.9\pi,0.1\pi)$; 21 iterations, CINC = 7.35dB

(c)

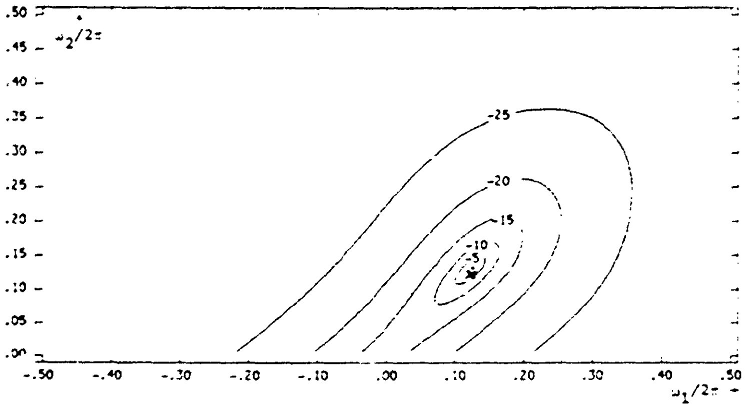
ML estimate; $\underline{\omega}=(0.1\pi,0.1\pi)$; CINC = 5.97dB

Fig. 5.2: Errors in spectral peak locations for real sinusoids in the regions shown in Fig. 5.1(b). S/N ratio = +5dB, 3x3 ACF, DFT size (NDFT) = 128x128 for (a) and (b)



Bartlett estimate; $\underline{\omega}=(0.1\pi,0.1\pi)$; CINC = 1.85dB

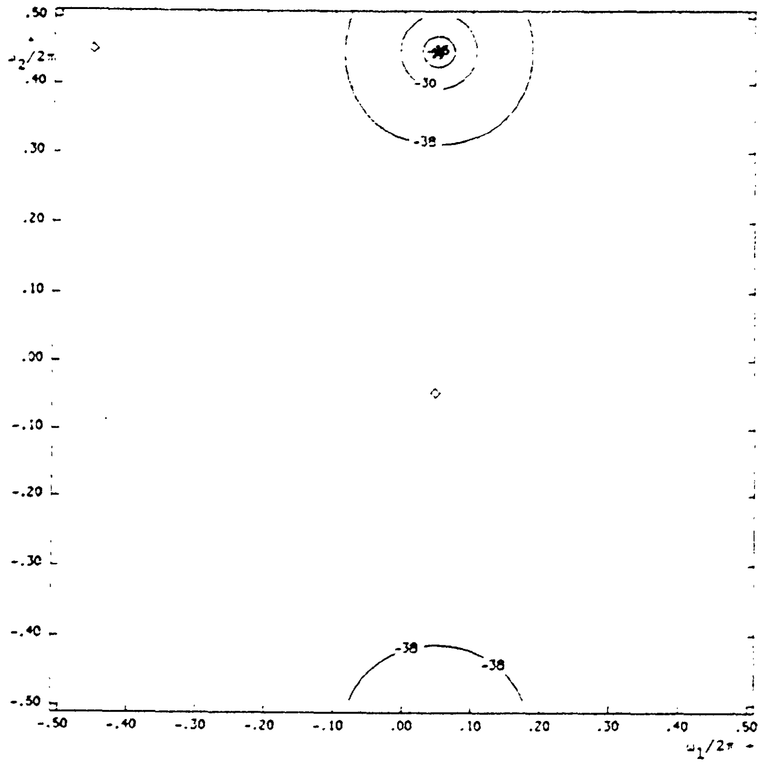
Fig. 5.2 (continued) (d)



ME estimate; $\underline{\omega}=(0.24\pi,0.24\pi)$; CINC = 5.19dB

Fig. 5.3: Improvement in the ME PS estimate as the peak location is changed. SNR = +5dB, 3x3 ACF, NDFT = 128x128

frequency plane. Fig. 5.4 illustrates the results for complex data using the same parameters as in Fig. 5.2(a). The peak is well resolved and the estimated peak location is excellent. Several other examples using complex data support this conclusion.



ME estimate; $\underline{\omega} = (0.1\pi, 0.9\pi)$;

Fig. 5.4: Uniform resolution for complex data regardless of peak location. The SNR and peak location are the same as Fig. 5.2(a). NDFT = 128x128, CINC = 7.6dB, (11 iterations, 308 seconds CPU time).

5.3 S/N Ratio Versus Resolution

To determine the effect of S/N ratio on the power spectrum estimates, several examples using the same ACF support and the same peak(s) location were considered. The signal to noise ratio (SNR) is defined as the sum of the powers of each peak divided by the total noise power. That is, for the case of M sinusoids with a_i^2 representing the power of the i'th sinusoid, the SNR is given by

$$\mathrm{SNR} \triangleq \frac{\sum_{i=1}^{M} a_i^2}{\sigma^2}$$

where σ^2 is the noise power. As was mentioned in section 5.2 above, the peak width in the spectral estimate decreases with increasing S/N ratio, and this leads to two distinct effects. First, for real data, the interference between mirror peaks decreases and this leads to an enlarged "region of resolution". Second, two peaks located close together in the 2-D frequency plane, which are not resolved at low S/N ratios, become resolvable at high S/N ratios.

Several examples were tried in order to verify these observations. For the set of experiments with one real sinusoid, a 3x3 ACF support region was used. The distance d_1 along the line $\omega_1 = \omega_2$ from the origin was used as a measure of the size of the region where good spectral peak locations can be obtained for real data. This distance indicates the closest that a sinusoid can be located to the origin to result in an "acceptable" ME PS estimate for a particular S/N ratio and the given ACF support. Table 5.1 summarizes the performance of the ME method versus the ML and Bartlett techniques for various S/N ratios. It is clear that the ME method affords the best performance of the three techniques, and has the largest "region of resolution" for the case of real sinusoids.

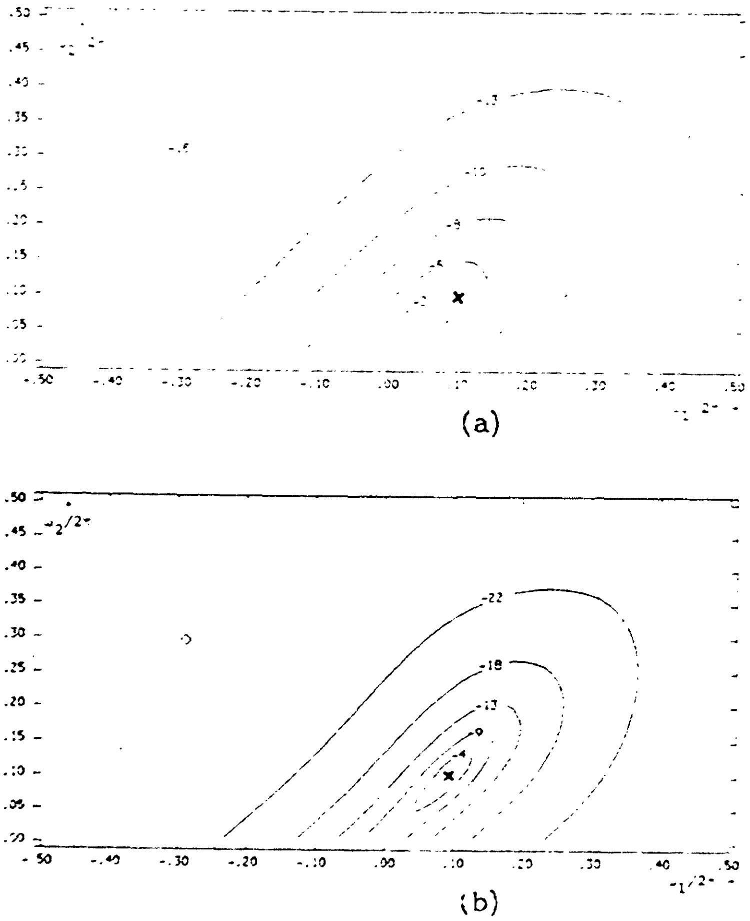
Figs. 5.5(a) and (b) are representative of the effect of increasing the S/N ratio for the case of a single real sinusoid. It is clear that the peak in the estimate is considerably sharper for the higher S/N ratio case, shown in Fig. 5.5(b). In fact, the peak location for the lower S/N ratio, Fig. 5.5(a), is quite erroneous. This is because the location lies in the special region (see Table 5.1) for this S/N ratio and the given ACF support.

Another set of examples were tried to determine the effect of changing the S/N ratio on the PS estimates of two sinusoids located close together in the 2-frequency plane. For this set of examples, it was decided that complex

TABLE 5.1

Minimum distance from the origin along $\omega_1 = \omega_2$ where good peak location estimates can be expected. ACF size 3x3.

3 x 3 Autocorrelation function	
S/N ratio	d_1
-5 dB	0.29π
0 dB	0.26π
+5 dB	0.20π



(a) ME estimate; $\underline{\omega} = (0.2\pi, 0.2\pi)$; SNR = 0db; NDFT = 64x64; CINC = 2.7dB; 61 iterations

(b) ME estimate; $\underline{\omega} = (0.2\pi, 0.2\pi)$; SNR = 5dB; NDFT = 256x256; CINC = 4.57dB; 52 iterations

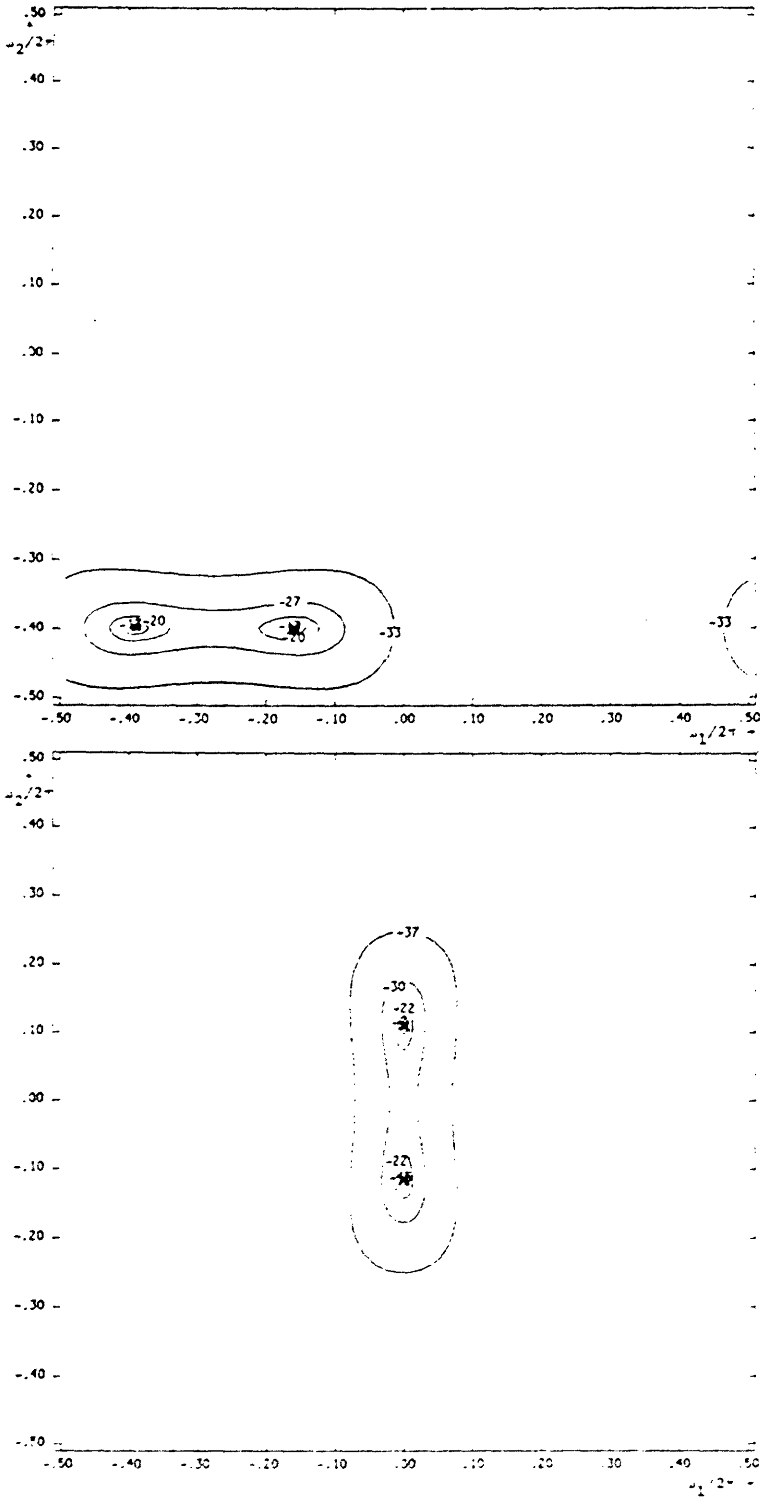
Fig. 5.5: Improvement in the ME PS estimate with increasing SNR. 3x3 ACF.

data would be used so as to separate the issue of resolution from that of the special regions for real data as discussed above. The size of the ACF support region was chosen to be 5x5 and both true and estimated ACF values were used.

The first observation that was made from this set of experiments was that the PS estimates did not depend on the absolute location of the peaks in the 2-D frequency plane. That is, the shape and size of the estimated spectral peaks remained the same regardless of where the complex sinusoids were located, if the same relative distance and orientation of the peaks was maintained. Figs. 5.6(a) and (b) illustrate this phenomenon. In these cases, the frequency separation between the peaks was held constant and the orientation of the peaks was kept either horizontal or vertical. The results clearly show the invariance of the spectral estimates under these conditions. Several other examples support this conclusion.

Using the above observations, examples were run to determine the minimum distance between two peaks such that they were resolved in the sense that the estimated power spectra displayed two distinct peaks. One peak location was held constant, while the location of the second peak was varied over a range such that initially the peaks were not resolved, and as the distance between the peaks was increased, the two peaks were resolved in the PS estimate. Since the accuracy of spectral peak location was not at issue here, d_2 the minimum distance between the two peaks where they became resolved was used as a measure of the resolution performance. It is clear that smaller values of d_2 imply higher resolution, while larger values of d_2 imply lower resolution for the various methods of estimating the power spectra.

Figs. 5.7 (a),(b),(c) and (d) are representative of the results obtained by



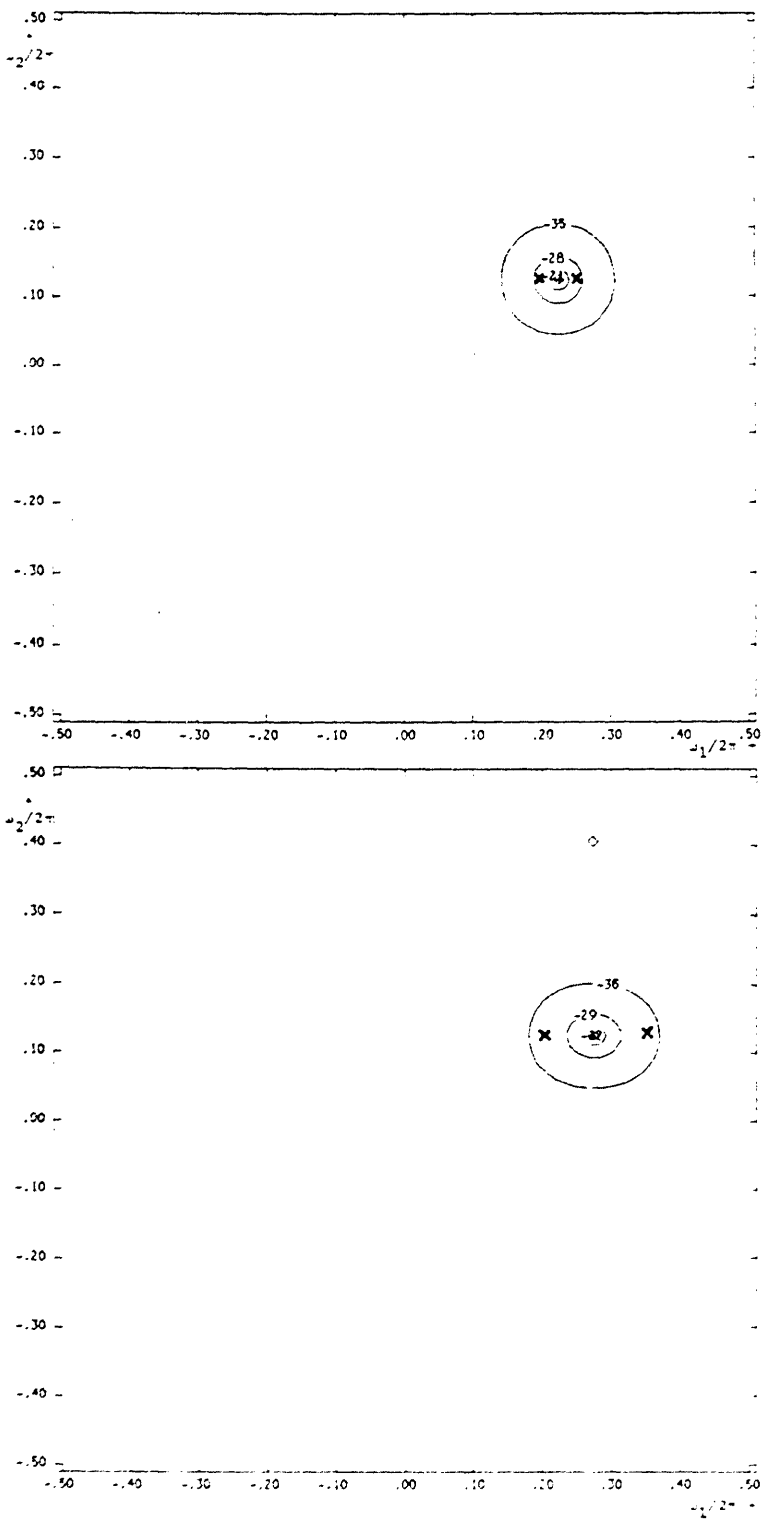
(a)

ME estimate; peaks: (−0.3π,−0.8π) and (−0.8π,−0.8π); CINC = 6.78dB

(b)

ME estimate; peaks: (0.0π,−0.25π) and (0.0π,0.25π); CINC = 7.54dB

Fig. 5.6: The ME PS estimates do not depend on absolute peak location for complex signals. SNR = −5dB, 5x5 ACF, NDFT = 512x512.



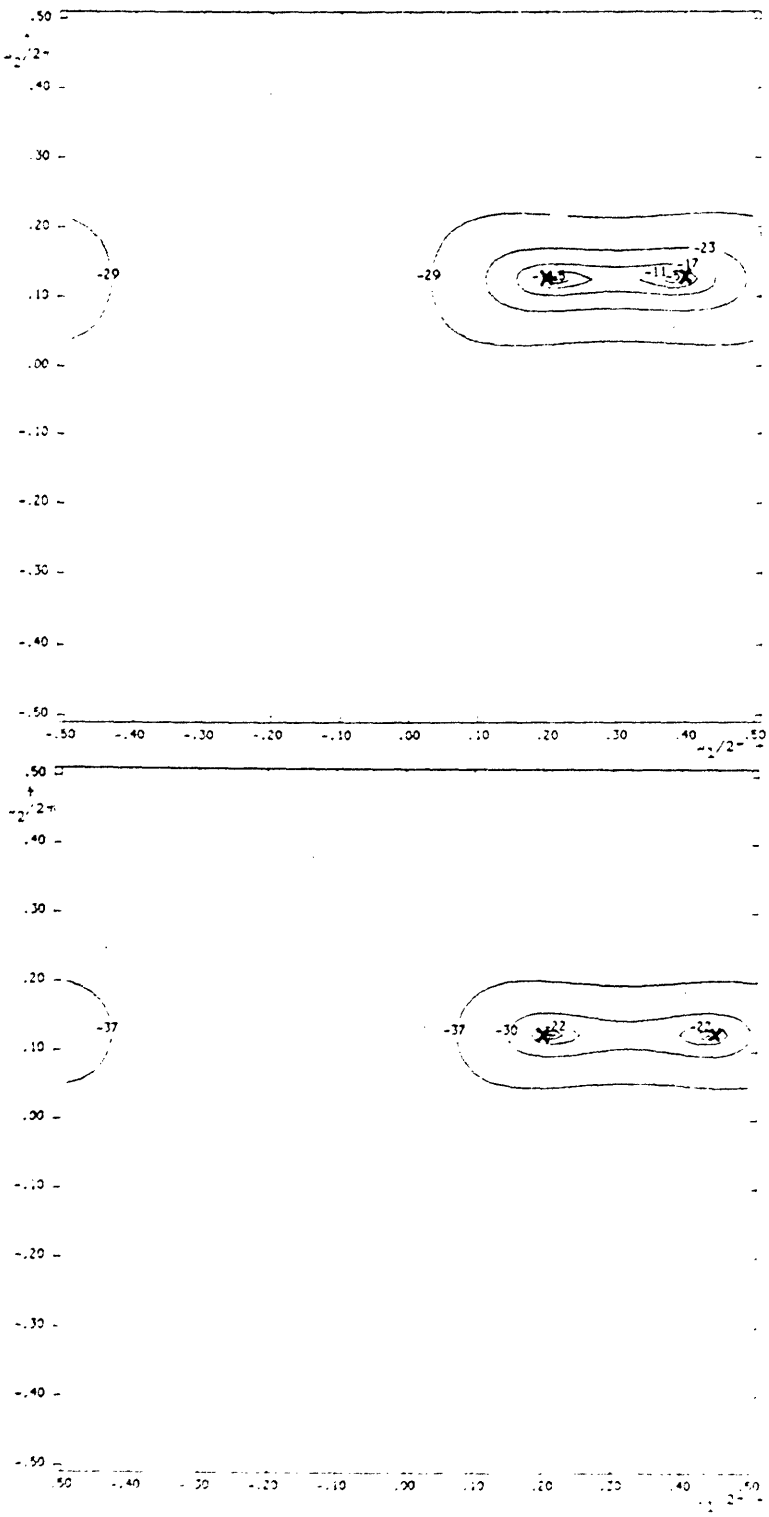
(a)

ME estimate; peaks: (0.4π,0.25π) ana (0.5π,0.25π); 7.03dB

(b)

ME estimate; peaks: (0.4π,0.25π) and (0.7π,0.25π); CINC = 7.34dB

Fig. 5.7: Change in the ME PS estimates as the separation between two peaks is increased. SNR = +5dB, 5x5 ACF, NDFT = 512x512.



(c)

ME estimate; peaks: $(0.4\pi, 0.25\pi)$ and $(0.8\pi, 0.25\pi)$; CINC = 5.95dB

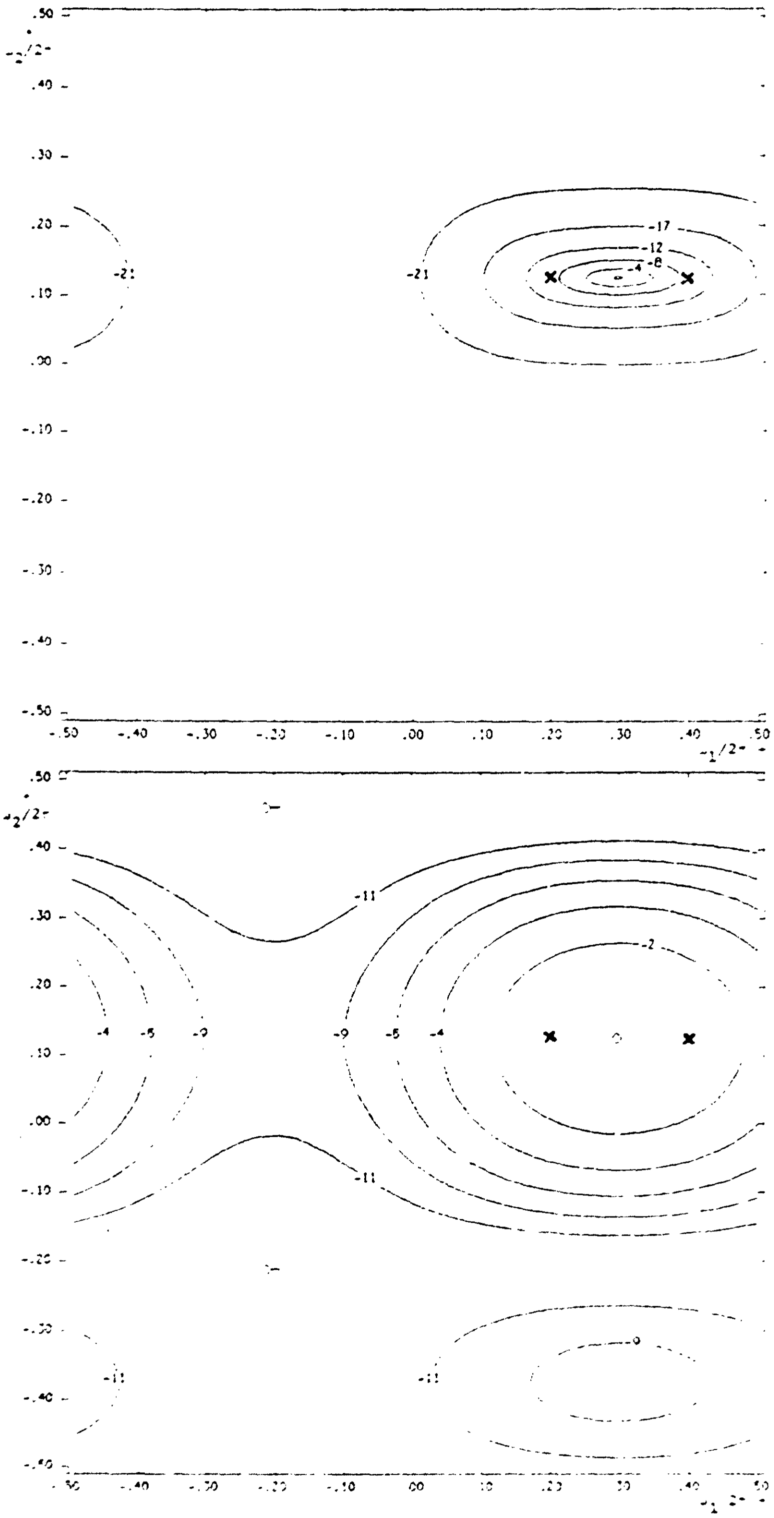
(d)

ME estimate; peaks: $(0.4\pi, 0.25\pi)$ and $(0.9\pi, 0.25\pi)$; CINC = 7.53dB

Fig. 5.7: Continued.

the ME method for a particular S/N ratio as the separation between the peaks is increased. Initially the two peaks are not resolved, and the spectral estimate consists of a single spectral peak, located approximately at the mid point of the line joining the true peak locations. As the distance between the peaks increases, the spectral estimate shows a distortion or stretching in the direction of the peaks, and eventually, the two peaks are resolved. The Maximum Likelihood and Bartlett estimates for the same data set as in Fig. 5.7 (c) are shown in Figs. 5.8(a) and (b). All the examples illustrated for the three methods used a 5x5 support region for the ACF. The poorer resolution of the ML and Bartlett techniques as compared to the Maximum Entropy method is apparent. Fig. 5.9 summarizes the resolution performance of the three techniques. It is clear that as in the 1-D case, the ME method affords higher resolution than the other two methods. It may be noted here that the resolution performance of the Bartlett estimates is determined only by the size of the ACF array available for analysis, and is independent of the S/N ratio, as far as the resolution measure d_2 is concerned.[7] The minimum distance d_2 for the peaks to be resolved in the ME and ML estimates decreases with increasing S/N ratio, with the ME method consistently outperforming the ML method.

At this point, it is necessary to point out that the measure adopted for the resolution performance evaluation is fairly arbitrary, and is used only to gauge the relative performance of the different techniques under the same set of conditions. It is obvious that the minimum resolution distance between two peaks also depends on their orientation in the 2-D frequency plane, as well as on the shape of 'A'. For example, when the shape of the ACF support region 'A' is a rectangle, one would expect a higher resolution in the direction of the longer dimension. This effect is shown in Figs. 5.10(a) and (b) for the ME method. The



(a) ML estimate; peaks: (0.4π,0.25π) and (0.8π,0.25π); CINC = 4.29dB

(b) Bartlett estimate; peaks: (0.4π,0.25π) and (0.8π,0.25π); CINC = 2.25dB

Fig. 5.8: Maximum Likelihood and Bartlett estimates for the data of Fig. 5.7(c).

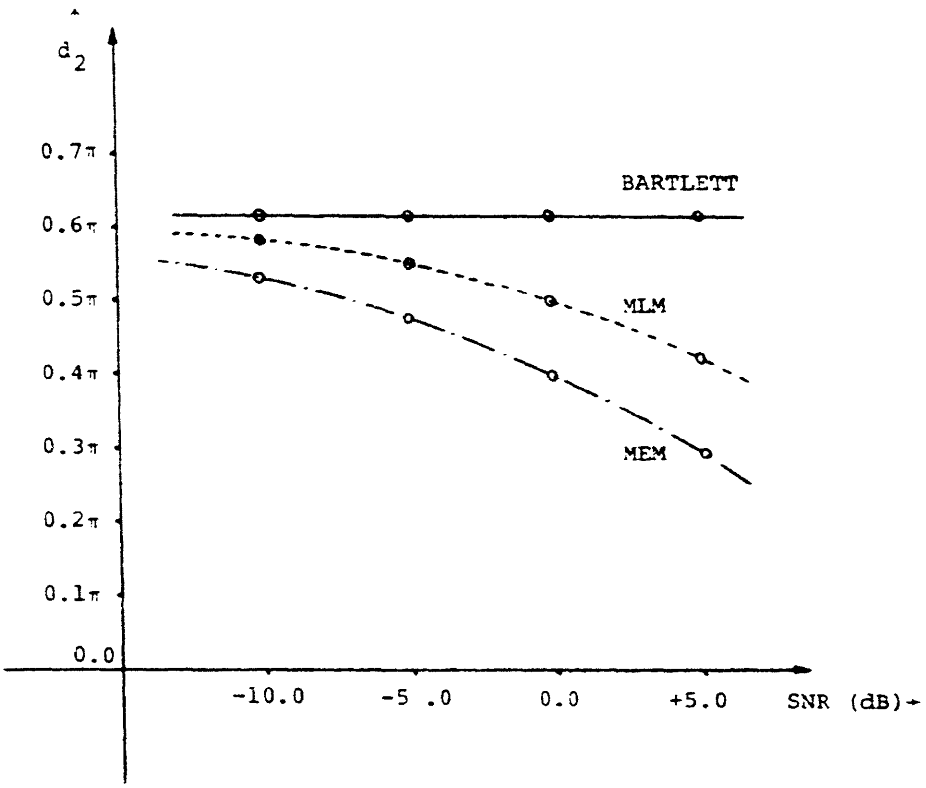
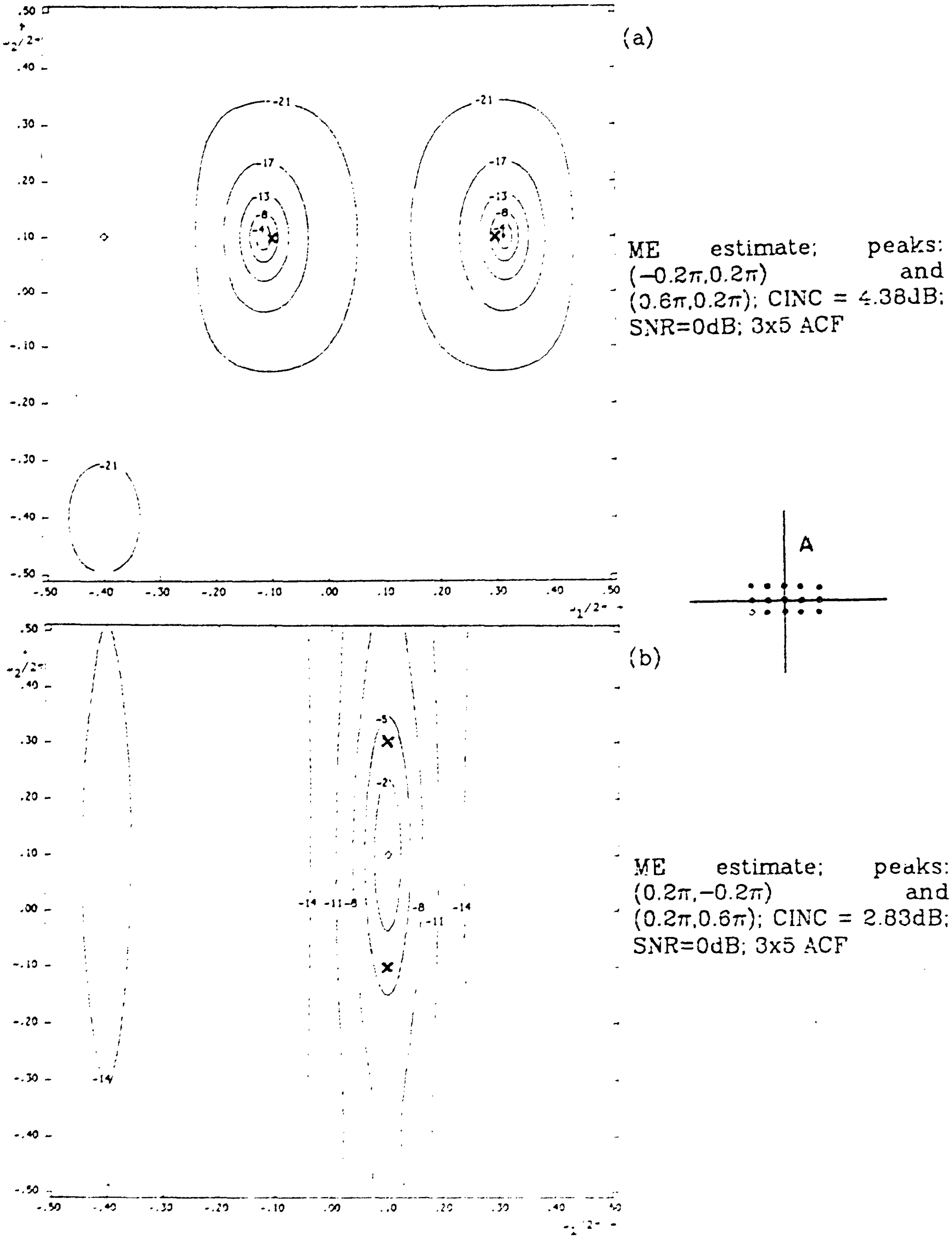


Fig. 5.9: Resolution properties of the ME, ML and Bartlett estimates.

shape of the region 'A' in these examples is a rectangle of size 3x5. Fig. 5.10(a) shows the ME PS estimate when the orientation of the peaks is along the longer dimension, and the peaks are seen to be resolved. In Fig. 5.10(b), the orientation of the peaks is in the direction of the shorter dimension of the region 'A', and the resulting PS estimate shows only a single peak. These examples clearly show that the resolution performance depends not only on the S/N ratio, but also on the specific shape of the ACF support region employed. The resolution measure adopted in this section is, therefore, only an indicator of the performance of the various techniques, and should not be taken as an absolute measure.



ME estimate; peaks: $(-0.2\pi, 0.2\pi)$ and $(0.6\pi, 0.2\pi)$; CINC = 4.38dB; SNR=0dB; 3x5 ACF

ME estimate; peaks: $(0.2\pi, -0.2\pi)$ and $(0.2\pi, 0.6\pi)$; CINC = 2.83dB; SNR=0dB; 3x5 ACF

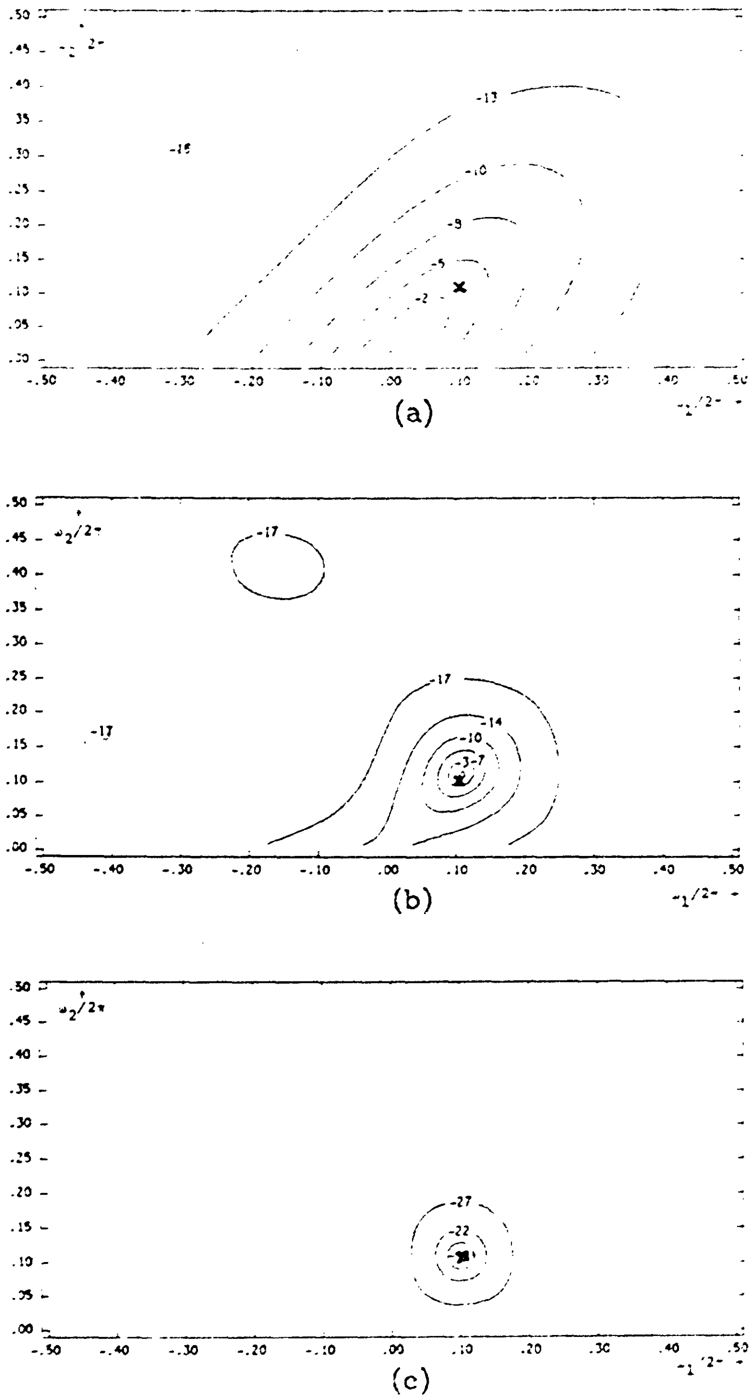
Fig. 5.10: Resolution of the ME PS estimates depends on the shape of the ACF support region 'A'.

5.4 ACF Support Size Versus Spectral Estimates

Another experiment that was performed in order to quantify the performance of the ME technique for 2-D PSE was to determine the effect of increasing the ACF support region 'A'. This is similar to increasing the model order[32] for 1-D signals. Again, exact ACF values were used and for a fixed S/N ratio, the effect of changing the ACF support size on the power spectrum estimates was noted.

The results obtained were similar to those obtained by increasing the S/N ratio. That is, the peaks in the spectral estimates grow much sharper and the size of the region of resolution for real data increases as the size of the ACF support region is increased. Figs. 5.11(a), (b) and (c) show the results for the case of a single real sinusoid in white noise. The ACF support region was fixed to be a square, and its size was varied from a 3x3 to 7x7 region about the origin. The change in the resulting PS estimates is fairly dramatic, but it should be kept in mind that with a square shape fixed for the region 'A', changing the ACF support from a 3x3 region to a 5x5 region increases the known information about the signal by a large amount. Whereas the 3x3 ACF has only 5 independent values (due to the Hermetian symmetry of the ACF), the 5x5 ACF has 13 independent values and for the 7x7 case the number of independent ACF values available for obtaining the PS estimates jumps to 25. A more gradual change in the power spectrum estimates would be expected if fewer new points were added to the ACF.

As a direct consequence of the sharper peaks in the estimate, the resolving power of the ME estimate increases with increasing size of 'A', as does the "region of resolution" for real data. This has been verified by various examples, and one such example is shown in Fig. 5.12 which shows the ME PS



ME estimate; $\underline{\omega}$ = (0.2π,0.2π); 3x3 ACF; CINC = 2.7dB; SNR = 0dB

ME estimate; $\underline{\omega}$ = (0.2π,0.2π); 5x5 ACF; CINC = 3.54dB; SNR = 0dB

ME estimate; $\underline{\omega}$ = (0.2π,0.2π); 7x7 ACF; CINC = 5.53dB; SNR = 0dB

Fig. 5.11: ACF support size versus ME PS estimates.

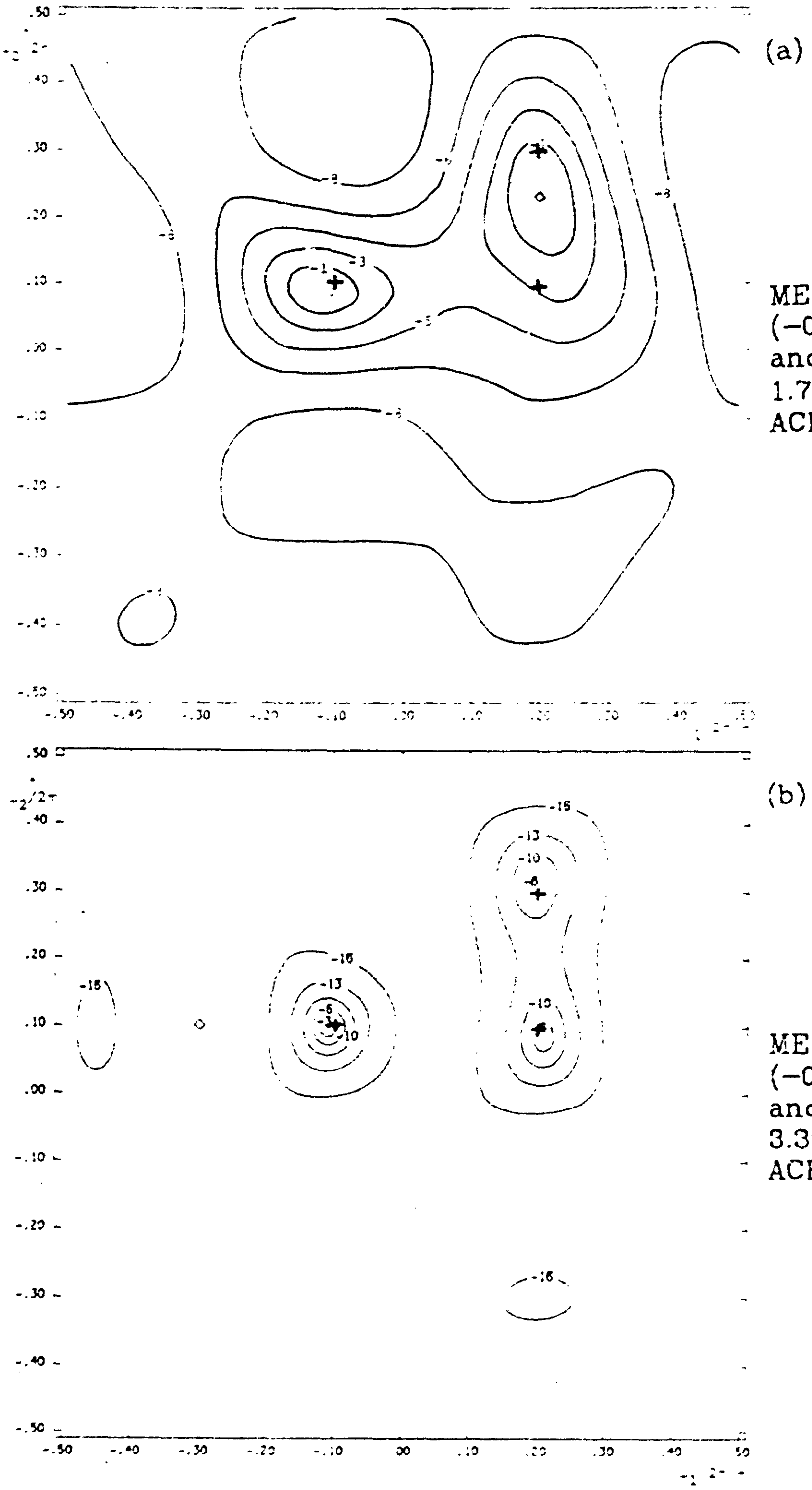
estimate for 3 sinusoids in white noise at a S/N ratio of -5dB. When the ACF support region has size 5x5,(Fig. 5.12(a)) two of the peaks merge into a single peak and the resulting estimate only shows two peaks. Fig. 5.12(b) shows the result of using a 7x7 ACF support size. The peak estimates are seen to be sharper and all three peaks are resolved.

Another effect which is common to increasing the S/N ratio or increasing the size of the ACF support region 'A' is the accuracy of the resulting peak location in the estimates. With a single sinusoid, the location of the spectral peak (LOSP) is fairly accurate even for low S/N ratios or for small sizes of the region 'A'. However, when multiple peaks are present in the spectrum, the interference between the peaks can lead to erroneous estimates for the spectral peak locations, especially for very low S/N ratios, or for small sizes of the region 'A'. Referring back to Fig. 5.11, it is seen that the LOSP is totally incorrect for a 3x3 ACF support region, becomes better when the size of the region 'A' is increased to 5x5 and is very accurate when the ACF support is of size 7x7. A similar conclusion is derived when one considers Fig. 5.5, where the accuracy of the LOSP changes with increasing S/N ratio.

The sum of the errors in peak locations is used as a performance measure in the accuracy of LOSP determination. The error in the LOSP is defined as:

$$\text{error(LOSP)} = \sum_{i=1}^{M} \sqrt{(\omega_{i1e} - \omega_{i1t})^2 + (\omega_{i2e} - \omega_{i2t})^2}$$

where the number of sinusoids is M, ω_{i1e} and ω_{i2e} represent the estimated x and y frequency locations of the i'th peak and ω_{i1t} and ω_{i2t} represent the true x and y frequencies of the peak location. Table 5.2 shows a comparison of the



ME estimate; peaks: $(-0.2\pi, 0.2\pi)$, $(0.4\pi, 0.2\pi)$ and $(0.4\pi, 0.6\pi)$; CINC = 1.74dB; SNR = -5dB; 5x5 ACF

ME estimate; peaks: $(-0.2\pi, 0.2\pi)$, $(0.4\pi, 0.2\pi)$ and $(0.4\pi, 0.6\pi)$; CINC = 3.35dB; SNR = -5dB; 7x7 ACF

Fig. 5.12· Increased resolution for multiple sinusoids with increasing ACF support size 'A'.

TABLE 5.2

Comparison of ME, ML and Bartlett estimates for peak location accuracy using exact autocorrelation values. The peak locations are listed as the pairs $(\omega_1/2\pi, \omega_2/2\pi)$, and the peak location error, (LOSP ERROR) is also in units of 2π. SNR = +5dB, one sinusoid cases: 3x3 ACF, two sinusoid cases: 5x5 ACF.

	TRUE LOCATION	MAXIMUM ENTROPY		MAXIMUM LIKELIHOOD		BARTLETT ESTIMATE	
		ESTIMATED LOCATION	LOSP ERROR	ESTIMATED LOCATION	LOSP ERROR	ESTIMATED LOCATION	LOSP ERROR
ONE SINUSOID	-.4000,0.4000	-.4000,0.4000	0.0000	-.4000,0.4000	0.0000	-.4000,0.4000	0.0000
	0.0745,-.4456	0.0745,-.4456	0.0000	0.0745,-.4456	0.0000	0.0745,-.4456	0.0000
	-.3000,-.3000	-.3000,-.3000	0.0000	-.3000,-.3000	0.0000	-.3000,-.3000	0.0000
	-.0500,-.0500	-.0500,-.0500	0.0000	-.0500,-.0500	0.0000	-.0500,-.0500	0.0000
	-.3125,0.3000	-.3125,0.3000	0.0000	-.3125,0.3000	0.0000	-.3125,0.3000	0.0000
TWO SINUSOIDS	-.4000,0.0000 0.0745,-.4456	-.4010,0.0040 0.0755,-.4496	0.0082	-.4010,0.0000 0.0755,-.4456	0.0020	-.4010,0.0020 0.0755,-.4476	0.0045
	0.3000,-.3000 -.3000,-.3000	0.2760,-.3000 -.2760,-.3000	0.0480	0.2970,-.3000 -.2970,-.3000	0.0059	0.2810,-.3000 -.2810,-.3000	0.0379
	0.3000,0.4120 -.0500,-.0500	0.3050,0.4060 -.0550,-.0440	0.0156	0.2990,0.4110 -.0490,-.0490	0.0028	0.3010,0.4120 -.0510,-.0500	0.0019
	0.1234,0.3456 -.3125,0.3000	0.1374,0.3396 -.3265,0.3060	0.0304	0.1304,0.3476 -.3195,0.2980	0.0146	0.1374,0.3416 -.3265,0.3040	0.0291
	0.2000,0.3125 -.1125,0.0330	0.1950,0.3135 -.1075,0.0320	0.0102	0.1990,0.3115 -.1115,0.0340	0.0028	0.1990,0.3125 -.1115,0.0330	0.0019
	0.3300,0.0000 0.0000,0.3333	0.3230,0.0070 0.0070,0.3263	0.0197	0.3300,0.0000 0.0000,0.3333	0.0000	0.3300,0.0000 0.0000,0.3333	0.0000
	-.3000,-.2000 0.1000,0.4430	-.2900,-.2040 0.0900,0.4470	0.0215	-.3000,-.2000 0.1000,0.4430	0.0000	-.3000,-.2000 0.1000,0.4430	0.0000
	-.1000,-.1000 0.3900,0.4000	-.1010,-.1000 0.3910,0.4000	0.0019	-.1000,-.1000 0.3900,0.4000	0.0000	-.1000,-.1000 0.3900,0.4000	0.0000

ME, ML, and Bartlett estimates as regards accuracy of estimated peak location for a few representative examples. The results are based on a 3x3 ACF for the single sinusoid case and exact ACF values, whereas the two sinusoid cases are based on a 5x5 region of support for the ACF. The frequency locations for the two sinusoid cases were chosen to be such that the peaks were resolved by all methods.

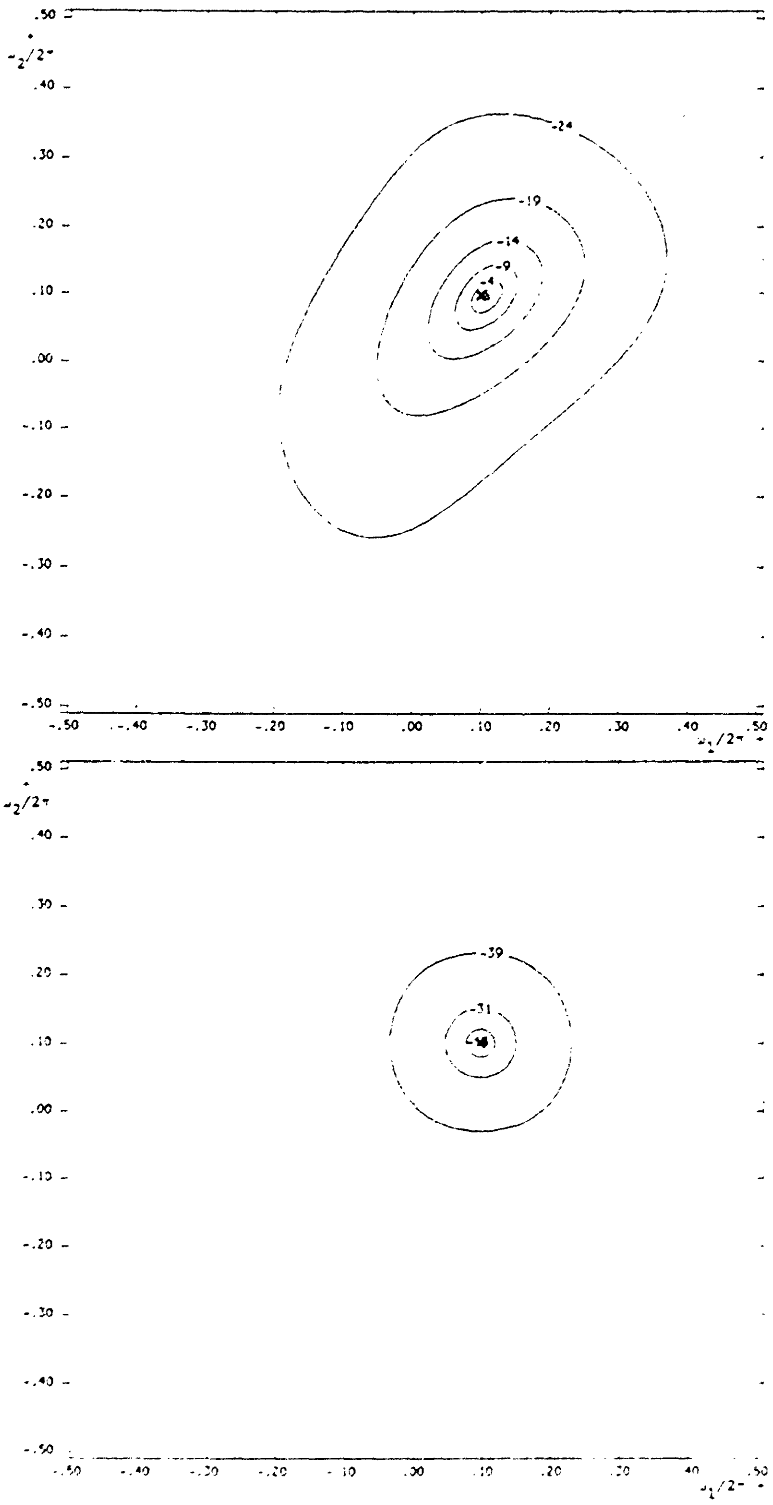
From Table 5.2, it is clear that all three techniques give excellent results in estimating the location of a single complex sinusoid. In fact, the LOSP error is precisely zero. For the case of two complex sinusoids, the situation is different. In that case, all methods show some errors in the estimation of the peak locations. Although the ME estimates exhibit much sharper peaks than the other two, it is seen that when the peaks are located such that all three methods would resolve them, then the ME method in fact, gives the worst results in terms of spectral peak location estimation. The ML and Bartlett estimates track each other quite closely, and give LOSP errors of approximately the same magnitude. For the last three examples in Table 5.2, both the ML and Bartlett methods were able to locate the peaks correctly. The ME method was never able to do so. The situation is similar for lower S/N ratios, although it may be expected that the 2-D ME estimates would give much better results at higher S/N ratios, like their 1-D counterparts.

5.5 Data Length Versus Spectral Estimates

In most applications of power spectrum estimation, it is the actual data rather than its ACF that is available for analysis. In such cases, the ACF has to be estimated from the data and then used to obtain the ME PS estimates. As was mentioned earlier, the biased estimator for the ACF is used in all cases here, since the unbiased estimator can lead to non- positive definite autocorrelation estimates. However, it is also not certain that the biased estimator will always give an extendible ACF.[22]

One important issue that arises is the effect of the size of the data segment on the PS estimates. It is clear that if one has a large amount of data, the ACF estimates will be very good, and hence the ME PS estimates will be better also. Similarly, the smaller the amount of data, the poorer the ACF estimate and hence the PS estimate can be expected to suffer. In order to quantify these observations, several examples were run where synthetic data was generated and the autocorrelation values were estimated from the data. The shape of the 2-D data segment was always taken to be square, and the ACF support region was also taken to be square. Fig. 5.13 shows the effect of changing the size of the known data set on the ME PS estimates, for the case of one sinusoid and a 3x3 region of support for the ACF. Data lengths ranging from 4x4 upto 60x60 were tried. As is clear from the figures, the shorter data length gives a spectral estimate that seems distorted (stretched) and the LOSP is not very accurate. As the data length is increased the shape of the spectral peak becomes more symmetric and the accuracy of the LOSP improves. The improvement of the spectral estimates is very rapid and the difference between the estimates obtained from a 12x12 data segment and the 20x20 or 36x36 data segment are negligible. Several other examples support this conclusion.

Fig. 5.14 illustrates the results of similar experiments for the case of two sinusoids with zero degrees relative phase, using a 5x5 region of support for the ACF. The location of the peaks was such that they could be resolved with true ACF values. The results are similar to the case of one sinusoid in that the



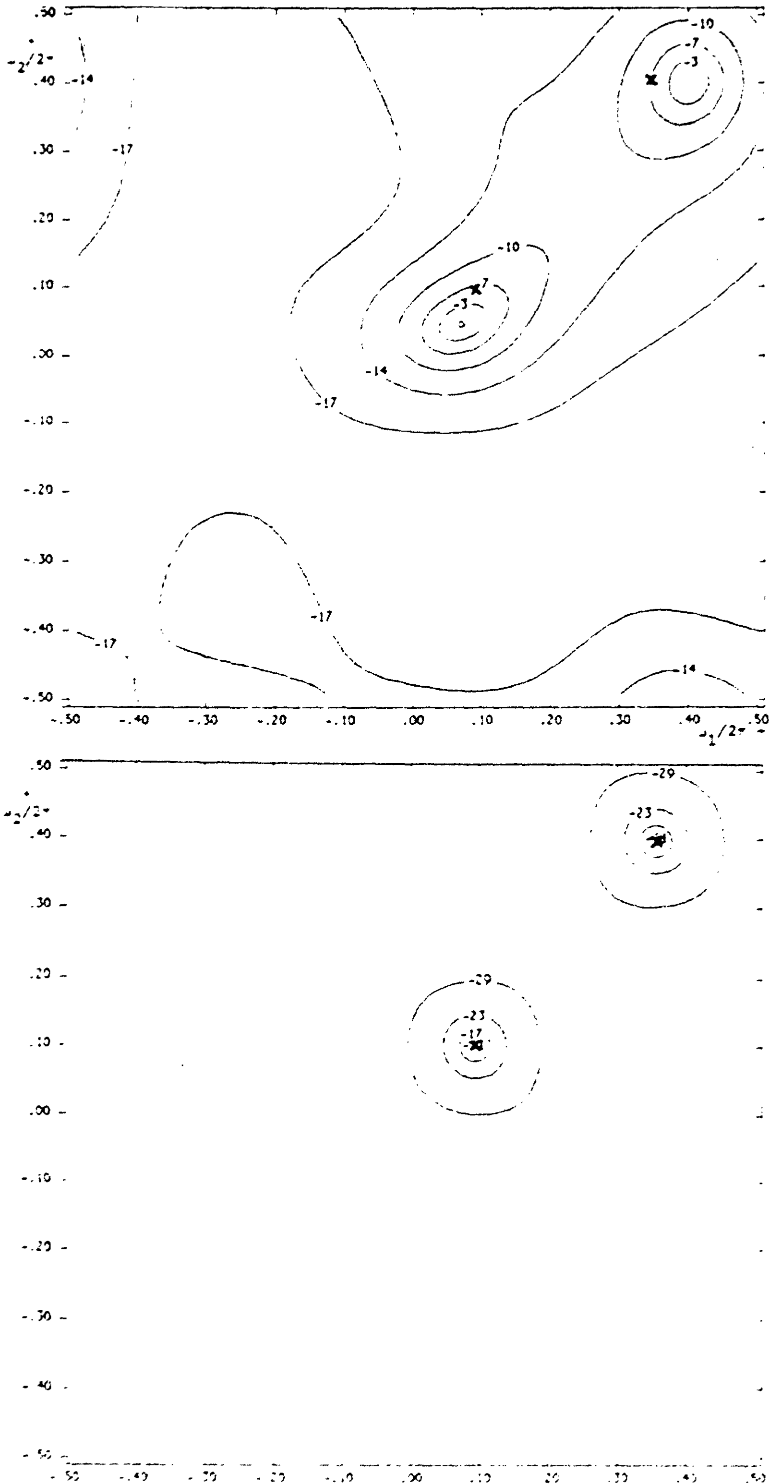
(a) Data length = 4x4.

ME estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 4.82dB;

(b) Data length = 20x20.

ME estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 7.93dB;

Fig. 5.13: The effect of changing the size of the data set on the ME PS estimate for one sinusoid. SNR = −5dB, 3x3 ACF, initial phase = 0°, true peak location = (0.2π,0.2π).



(a) Data length = 4x4.

ME estimate; peaks: (0.2π,0.2π) and (0.7π,0.8π); CINC = 3.50dB;

(b) Data length = 60x60.

ME estimate; peaks: (0.2π,0.2π) and (0.7π,0.8π); CINC = 5.99dB;

Fig. 5.14: The effect of changing the size of the data set on the ME PS estimate for two sinusoids. SNR = +5dB, 5x5 ACF, relative phase = 0°, true peak locations are (0.2π,0.2π) and (0.7π,0.8π).

spectrum initially shows some distortion and the spectral estimates improve rapidly with increasing data length. The accuracy of the LOSP also improves with increasing data length.

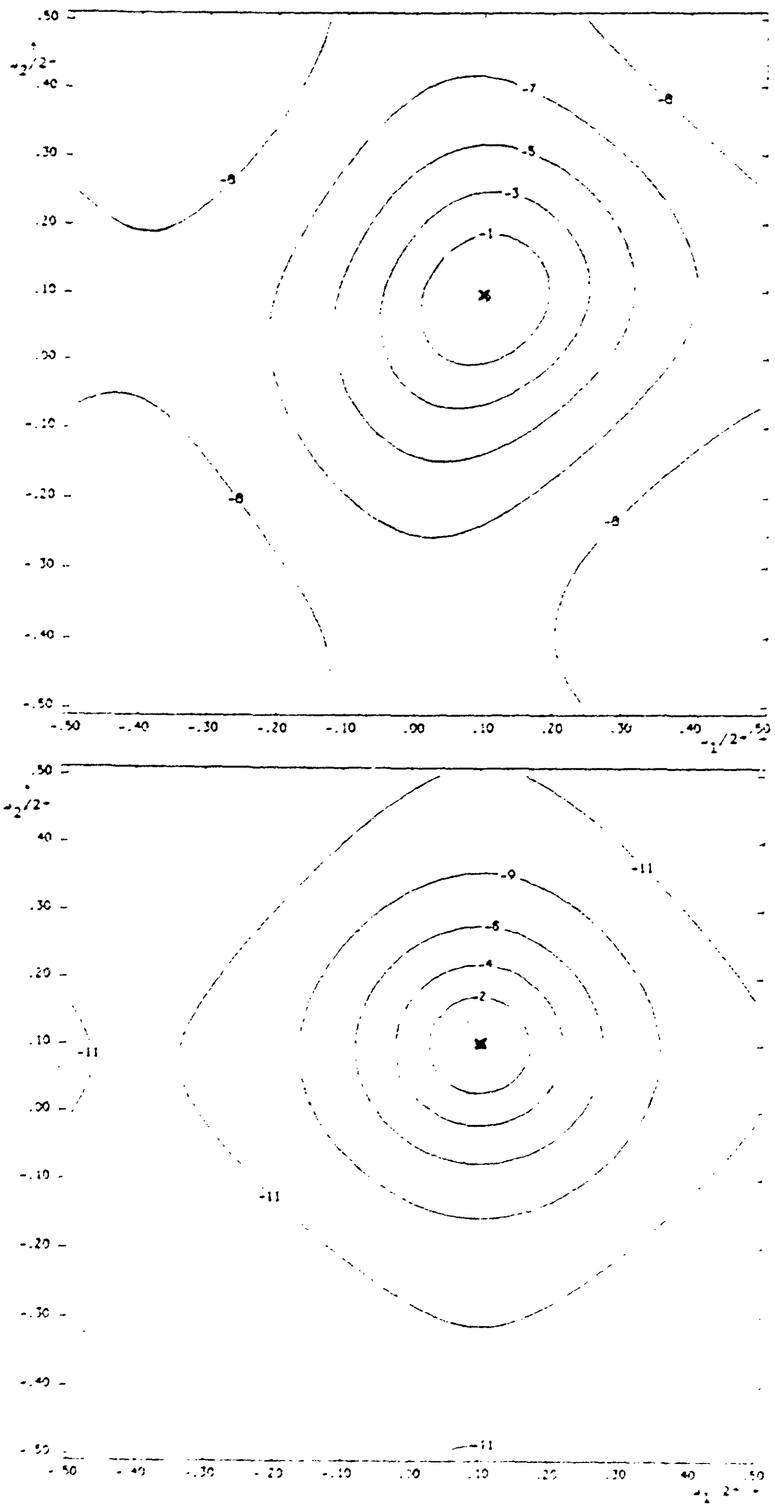
The results of the ME technique are again compared to those obtained via the ML and the Bartlett estimates as regards the accuracy of LOSP determination. Figs. 5.15 and 5.16 show the results obtained by the other two methods for the same data sets used in Figs. 5.13 and 5.14. The performance vis-a-vis peak location error is summarized in Table 5.3.

From Table 5.3, it is clear that the three methods again perform very well for the case of a single complex sinusoid. However, the ME method again gives the sharpest estimate, and although it is not apparent from the table, the improvement in the ME estimates is the most rapid, with very little improvement being visible in the spectrum after a data length of 12x12. The ML and Bartlett estimates do not stablize for data lengths upto 44x44, and show continuous improvement although the spectral contours donot achieve the same symmetry as the ME estimates till a data length of about 60x60. For the two-sinusoid case, the situation is similar except that now all techniques show larger LOSP errors. Again, the ME estimates have been observed to stabilize most rapidly for very short data lengths.

TABLE 5.3

Comparison of ME, ML and Bartlett estimates for peak location accuracy using estimated autocorrelation values. SNR = +5dB. One sinusoid case: 3x3 ACF, initial phase = 0°, peak location = $(0.2\pi, 0.2\pi)$. Two sinusoid case: 5x5 ACF, relative phase = 0°, peaks located at $(0.2\pi.0\ 2\pi)$ and $(0.7\pi, 0.8\pi)$.

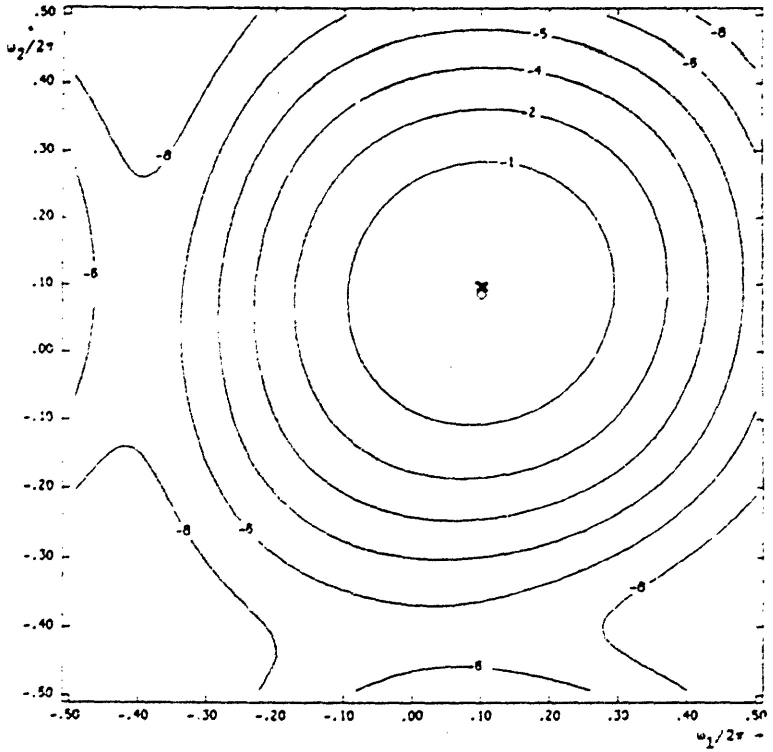
		MAXIMUM ENTROPY		MAXIMUM LIKELIHOOD		BARTLETT ESTIMATE	
	DATA SIZE	ESTIMATED LOCATION	LOSP ERROR	ESTIMATED LOCATION	LOSP ERROR	ESTIMATED LOCATION	LOSP ERROR
ONE SINUSOID	4x4	0.109,0.097	0.0094	0.091,0.101	0.009	0.101,0.090	0.0100
	12x12	0.100,0.101	0.0010	0.099,0.101	0.0014	0.099,0.101	0.0014
	20x20	0.100,0.101	0.0010	0.101,0.101	0.0014	0.101,0.101	0.0014
	28x28	0.100,0.100	0.0000	0.100,0.100	0.0000	0.100,0.100	0.0000
	36x36	0.100,0.100	0.0000	0.100,0.100	0.0000	0.100,0.100	0.0000
	44x44	0.09-,0.100	0.0009	0.099,0.100	0.0009	0.099,0.100	0.0009
	52x52	0.100,0.100	0.0000	0.100,0.100	0.0000	0.100,0.100	0.0000
	60x60	0.100,0.100	0.0000	0.100,0.100	0.0000	0.100,0.100	0.0000
TWO SINUSOIDS	4x4	0.086,0.075 0.375,0.399	0.0536	0.099,0.075 0.375,0.375	0.0589	0.109,0.093 0.366,0.375	0.0411
	12x12	0.094,0.103 0.357,0.397	0.0143	0.121,0.125 0.326,0.375	0.0673	0.102,0.105 0.345,0.392	0.0148
	20x20	0.095,0.104 0.357,0.396	0.0144	0.103,0.104 0.345,0.397	0.0108	0.103,0.105 0.346,0.397	0.0108
	28x28	0.095,0.102 0.358,0.397	0.0135	0.119,0.125 0.330,0.375	0.0634	0.101,0.103 0.348,0.397	0.0068
	36x36	0.097,0.103 0.361,0.395	0.0163	0.103,0.102 0.347,0.357	0.0072	0.102,0.103 0.348,0.397	0.0072
	44x44	0.094,0.102 0.356,0.398	0.0126	0.116,0.122 0.334,0.378	0.0544	0.101,0.103 0.348,0.397	0.0067
	52x52	0.094,0.102 0.356,0.398	0.0126	0.116,0.121 0.334,0.378	0.0536	0.101,0.102 0.348,0.397	0.0058
	60x60	0.094,0.102 0.356,0.398	0.0126	0.103,0.103 0.348,0.398	0.0071	0.102,0.103 0.349,0.398	0.0058



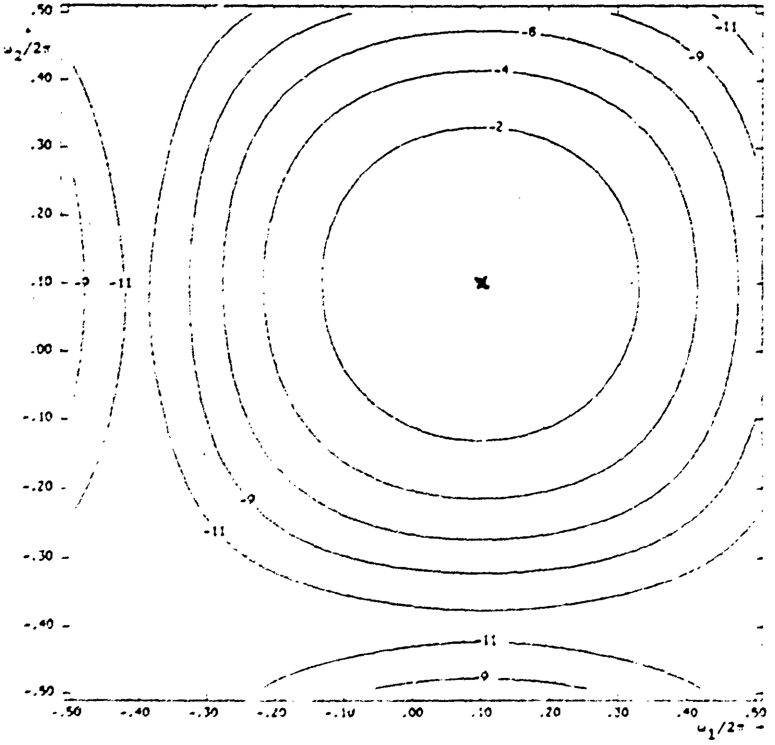
(a) ML estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 1.79dB

(b) ML estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 2.27dB

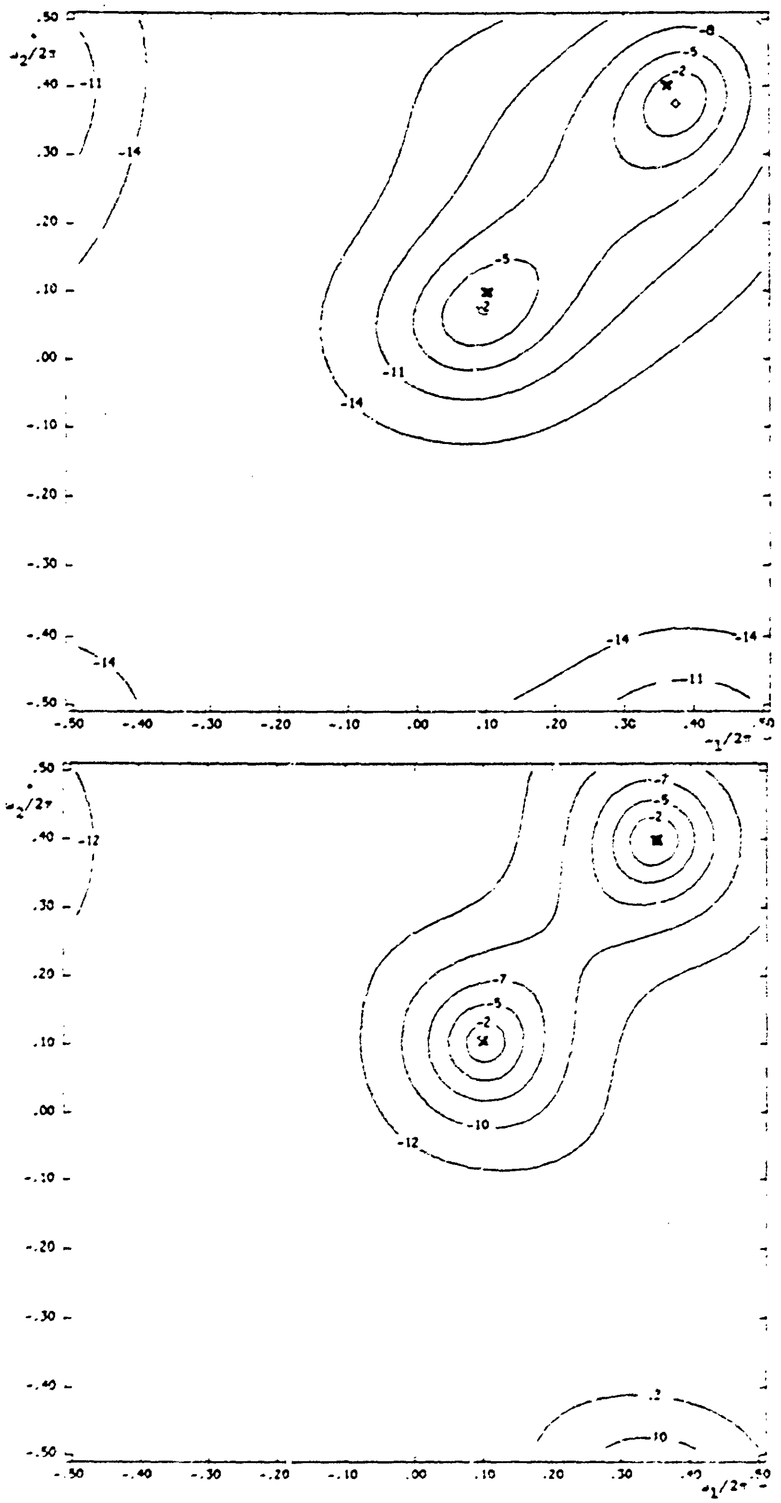
Fig. 5.15: The effect of data length on ML and Bartlett estimates for one sinusoid. Same data parameters as Fig. 5.13.



(c) Bartlett estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 1.37dB



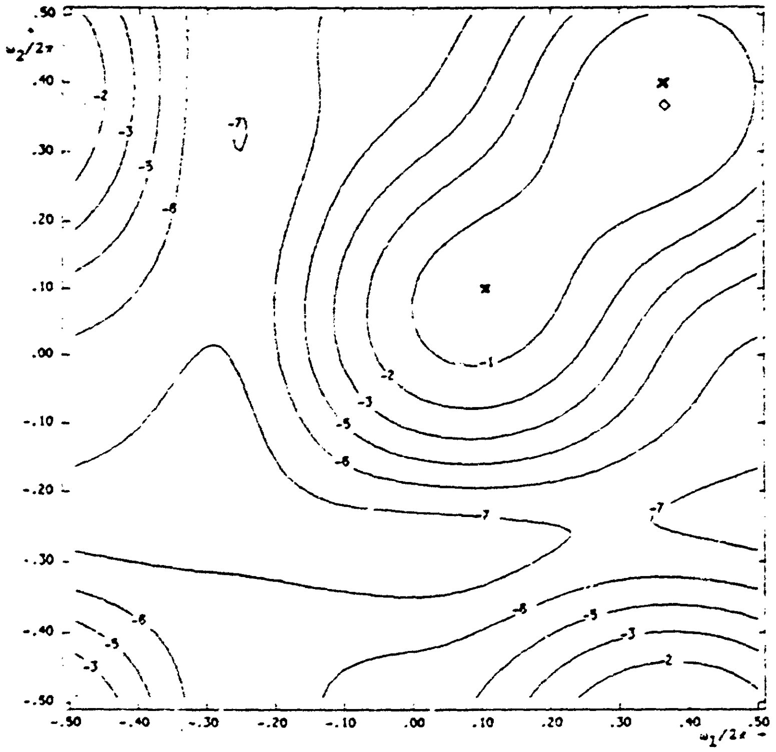
(d) Bartlett estimate; $\underline{\omega}$ = (0.2π,0.2π); CINC = 2.28dB



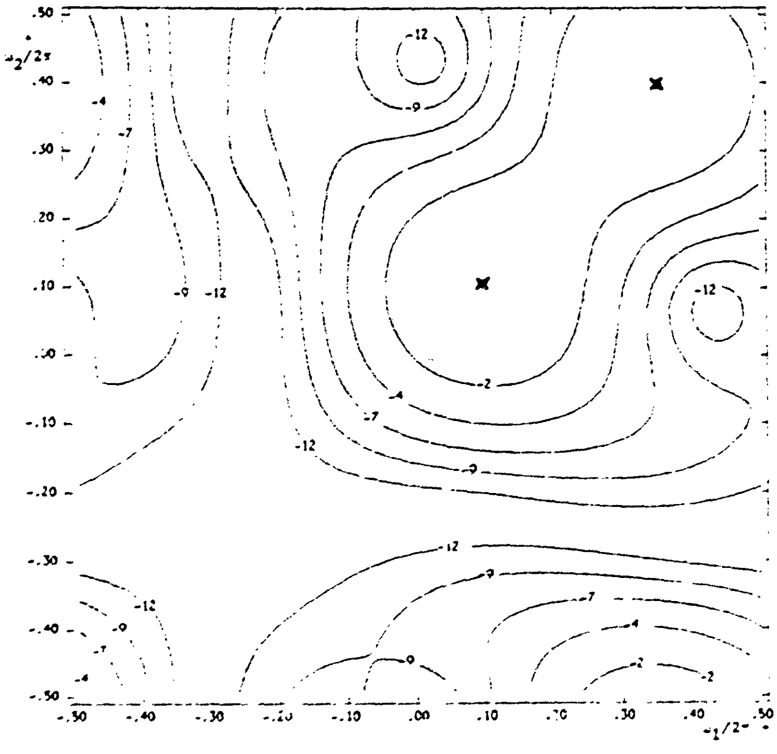
(a) ML estimate; peaks: $(0.2\pi, 0.2\pi)$ and $(0.7\pi, 0.8\pi)$; CINC = 2.90dB;

(b) ML estimate; peaks: $(0.2\pi, 0.2\pi)$ and $(0.7\pi, 0.8\pi)$; CINC = 2.53dB;

Fig. 5.16: The effect of data length on ML and Bartlett estimates for two sinusoids. Same data parameters as Fig. 5.14.



(c) Bartlett estimate; peaks: $(0.2\pi, 0.2\pi)$ and $(0.7\pi, 0.8\pi)$; CINC = 1.27dB;

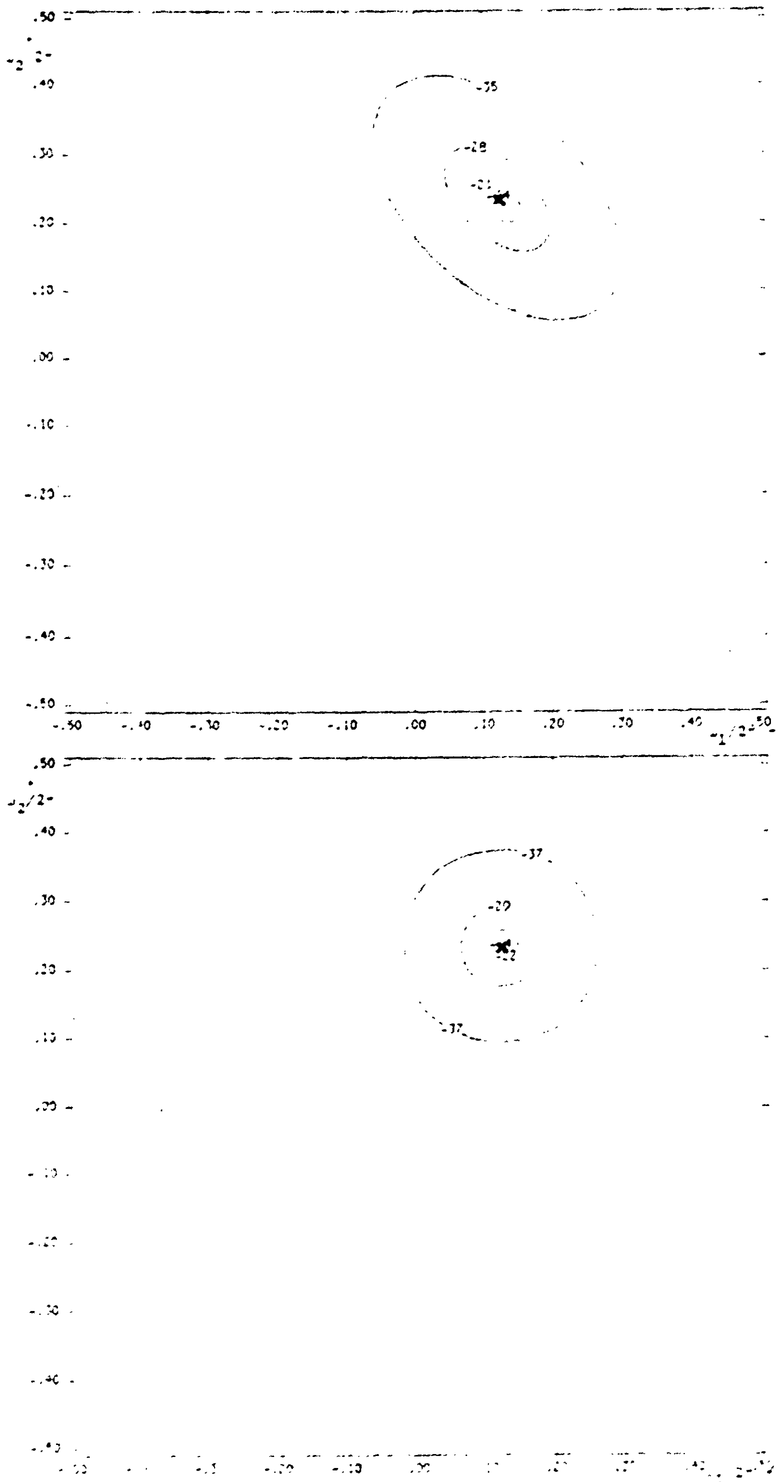


(d) Bartlett estimate; peaks: $(0.2\pi, 0.2\pi)$ and $(0.7\pi, 0.8\pi)$; CINC = 2.40dB;

5.6 Missing Correlation Values

It is clear that if the data were not available over a square segment the spectral estimates would show a corresponding distortion. However, the above results indicate that because of the rapid improvement of the spectral estimates, the shape of the data segment would have a considerable effect only if one of the dimensions was extremely small as compared to the other. Of course, the data need not be available over a rectangular grid, or over a rectangular region. It is also possible that the ACF may not be available or estimated over the full rectangular grid. When the data are sampled randomly, it is still possible to use the FFT implementation of the algorithm, as the randomly spaced samples (ACF values) can be placed on an underlying rectangular grid. Of course, depending on the randomness of the sample spacing, the grid may become extremely fine (very long FFT lengths). The ME spectral estimates could still be obtained by the algorithm as long as the estimated ACF values satisfied Woods' theorem,[2] that is, the ACF was extendible. Although Woods' theorem does not cover cases where the gaps in the ACF fall inside the outer boundary of the region 'A', Lang[22] has provided the necessary conditions when the ME estimates could be obtained in these situations, and the algorithm has been used to obtain ME PS estimates for such cases.

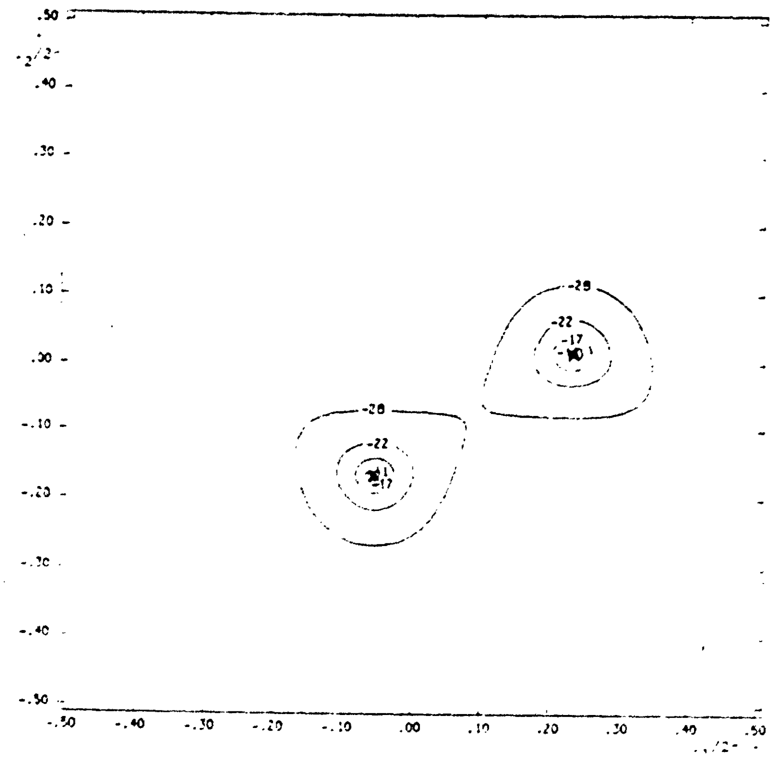
Fig. 5.17 illustrates the ME PS estimates for one and two sinusoids where the shape of the region 'A' is arbitrary. In fact, for the two sinusoid case, Fig. 5.17(c) the gaps in the ACF lie inside the boundary of the region 'A'. Figs. 5.17(b) and (d) show the corresponding ME estimates when there are no gaps in the ACF.



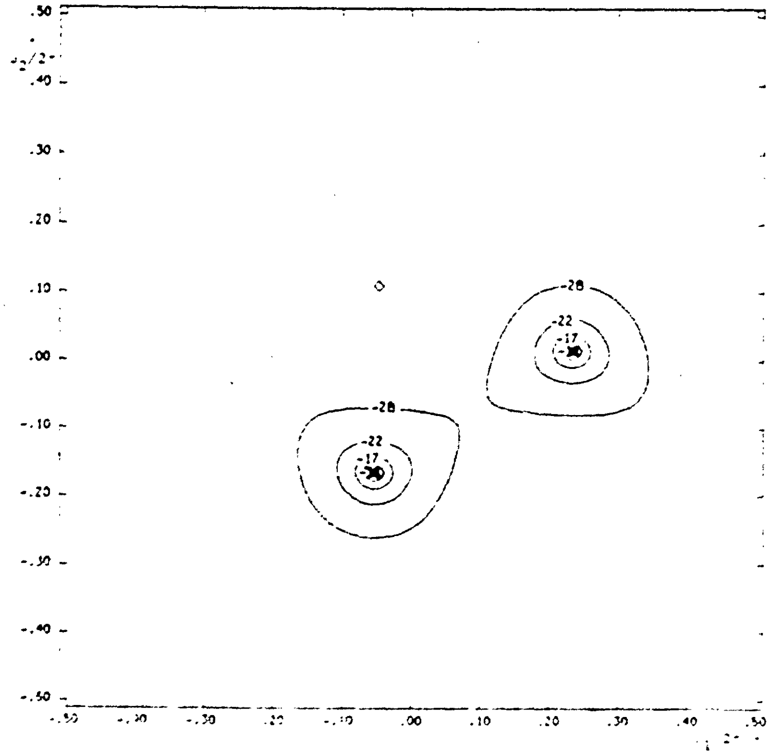
(a) ME estimate; $\underline{\omega} = (0.2468\pi, 0.469\pi)$; ACF lags (1,1) and (-1,-1) missing; 3x3 ACF; SNR = 5dB; CINC = 7.07dB

(b) ME estimate; $\underline{\omega} = (0.2468\pi, 0.469\pi)$; 3x3 ACF; SNR = 5dB; CINC = 7.42dB

Fig. 5.17: ME PS estimates when the ACF has missing points. SNR = +5dB.



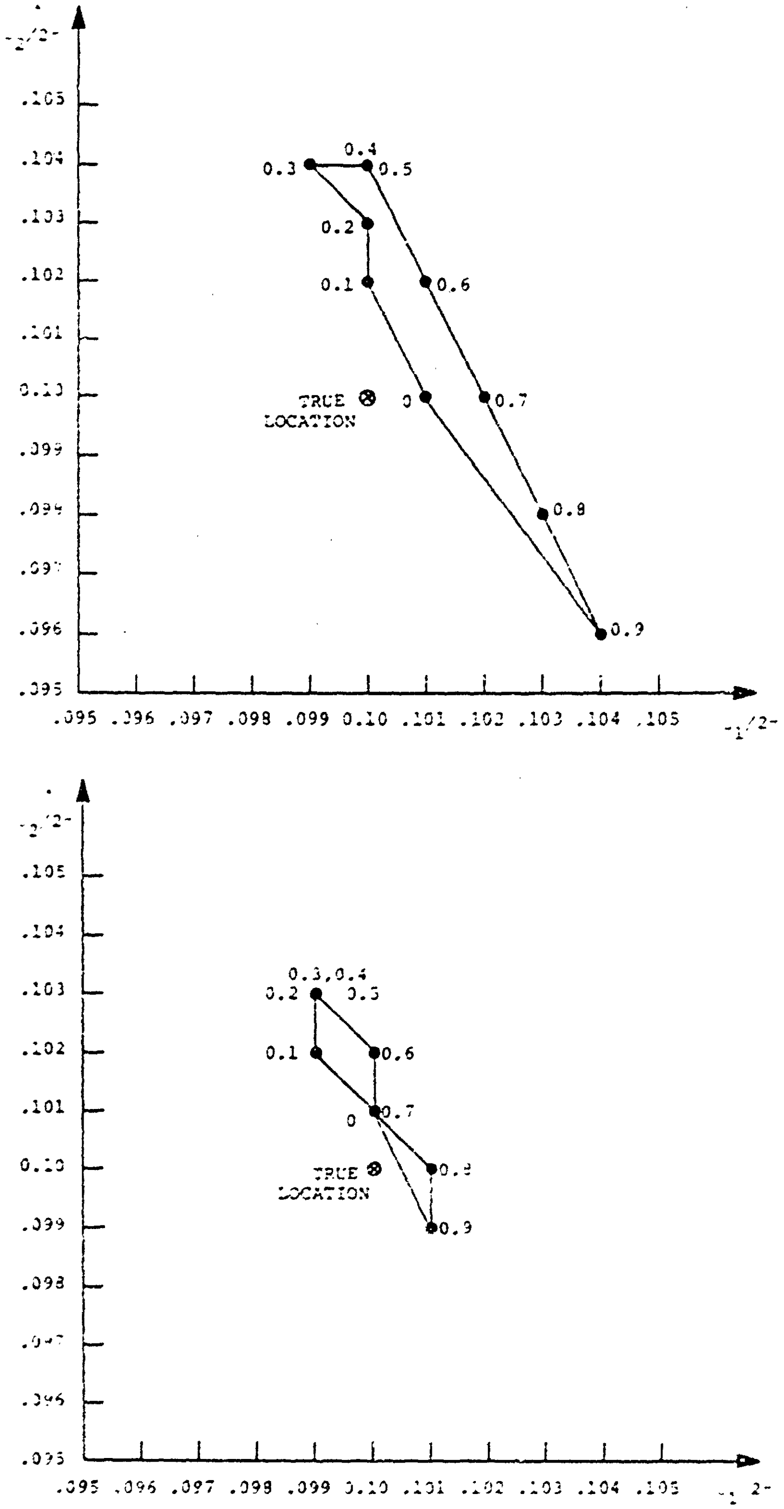
(c) ME estimate; peaks: $(-0.1\pi,-0.3\pi)$ and $(0.46\pi,0.0\pi)$; ACF lags (1,1) and (-1,-1) missing; 5x5 ACF; SNR = 5dB; CINC = 5.73dB



(d) ME estimate; peaks: $(-0.1\pi,-0.3\pi)$ and $(0.46\pi,0.0\pi)$; 5x5 ACF; SNR = 5dB; CINC = 5.72dB

5.7 Effect of Initial Phase on Spectral Estimates

When the data consists of a sinusoid in noise, and the ACF has to be estimated from the data itself, it becomes important to know the effect of the initial phase of the sinusoid on the spectral estimate. In the 1-D case, it has been noted that the phase causes a shift in the LOSP in the periodogram and the direct data Burg methods of PSE.[30,31] Although the algorithm used here is not a direct data method, the ACF must be estimated from the data, and therefore the starting phase will have an effect on the ACF values, and hence on the PS estimates. Several examples have been tried with the data consisting of a single sinusoid in white noise, using a 3x3 region of support for the ACF. The results are similar to the 1-D case in that the LOSP shows an oscillation about the true peak location, the amplitude of the oscillation decreasing with increasing data length, as could be expected. Fig. 5.18(a) shows the oscillation in the LOSP for one example where the size of the data segment used is 12x12, and the initial phase of the sinusoid is varied from zero to 2π. The size of the region 'A' is 3x3 and the S/N ratio is 0dB. The effect of the phase on the LOSP changes with S/N ratio, with the amplitude of the oscillation decreasing with increasing S/N ratio. Fig. 5.18(b) shows the LOSP oscillation for the same experiment as shown in Fig. 5.18(a) except that the S/N ratio is +5dB instead of 0dB. Figs. 5.18(c) and (d) show the corresponding results for the ML method and Figs. 5.18(e) and (f) give the results of the Bartlett technique. The oscillations in the LOSP for the three techniques are seen to be very similar in amplitude. Again, the behaviour of the ME method is different from the ML and Bartlett techniques which show almost identical errors in the LOSP.



(a) ME method SNR = 0dB

3x3 ACF

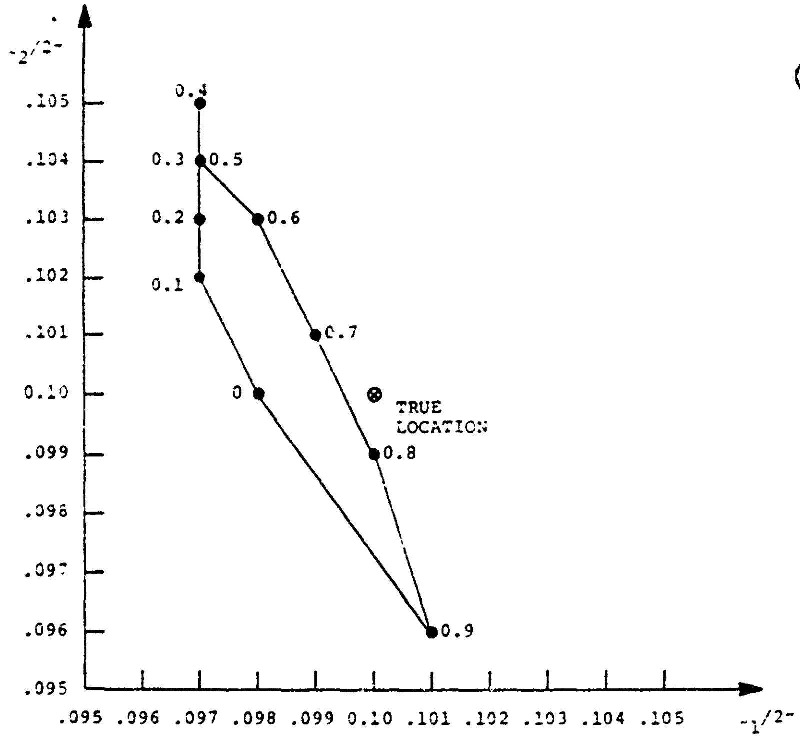
peak location = (0.2π,0.2π)

(b) ME method SNR = 5dB

3x3 ACF

peak location = (0.2π,0.2π)

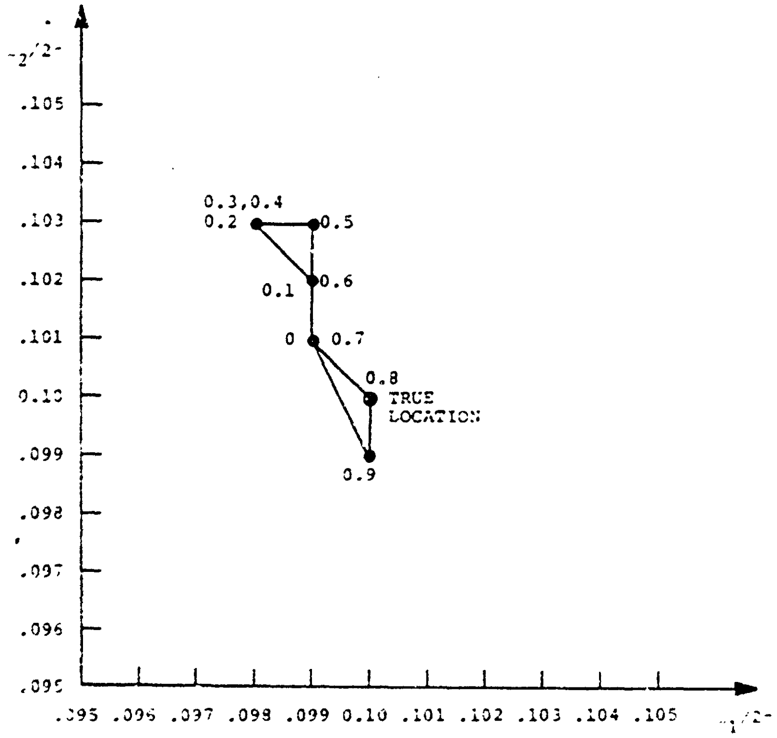
Fig. 5.18: The variation of estimated peak location with initial phase.



(c) ML method SNR = 0dB

3x3 ACF

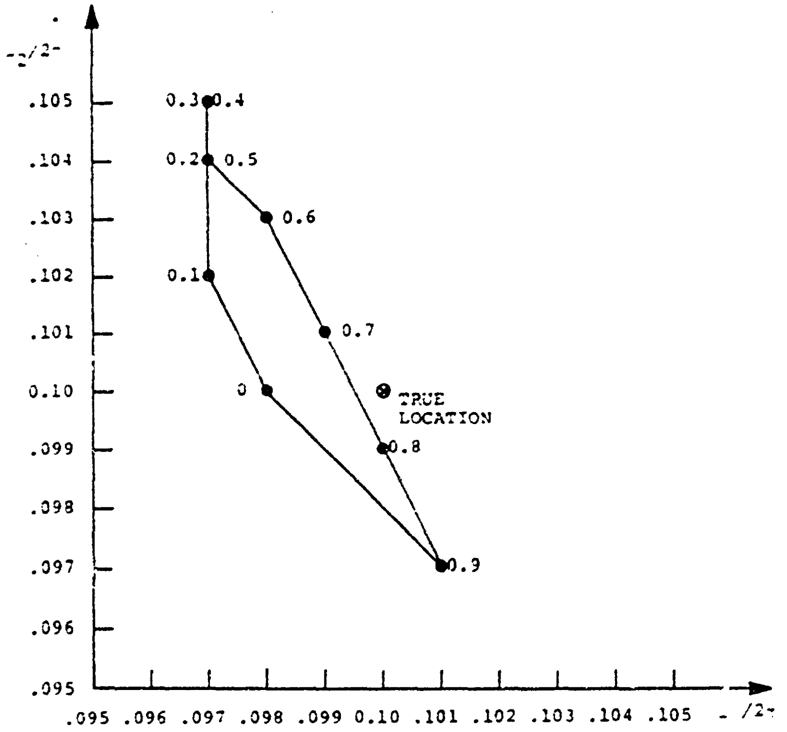
peak location = $(0.2\pi, 0.2\pi)$



(d) ML method SNR = 5dB

3x3 ACF

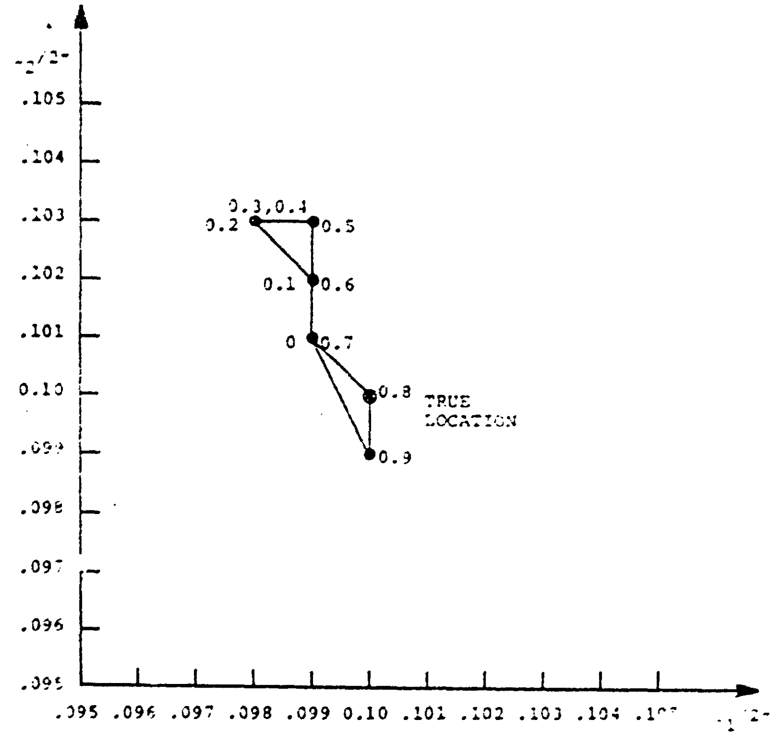
peak location = $(0.2\pi, 0.2\pi)$



(e) Bartlett; SNR = 0dB

. 3x3 ACF.

peak location = (0.2π,0.2π).



(f) Bartlett; SNR = 5dB

. 3x3 ACF.

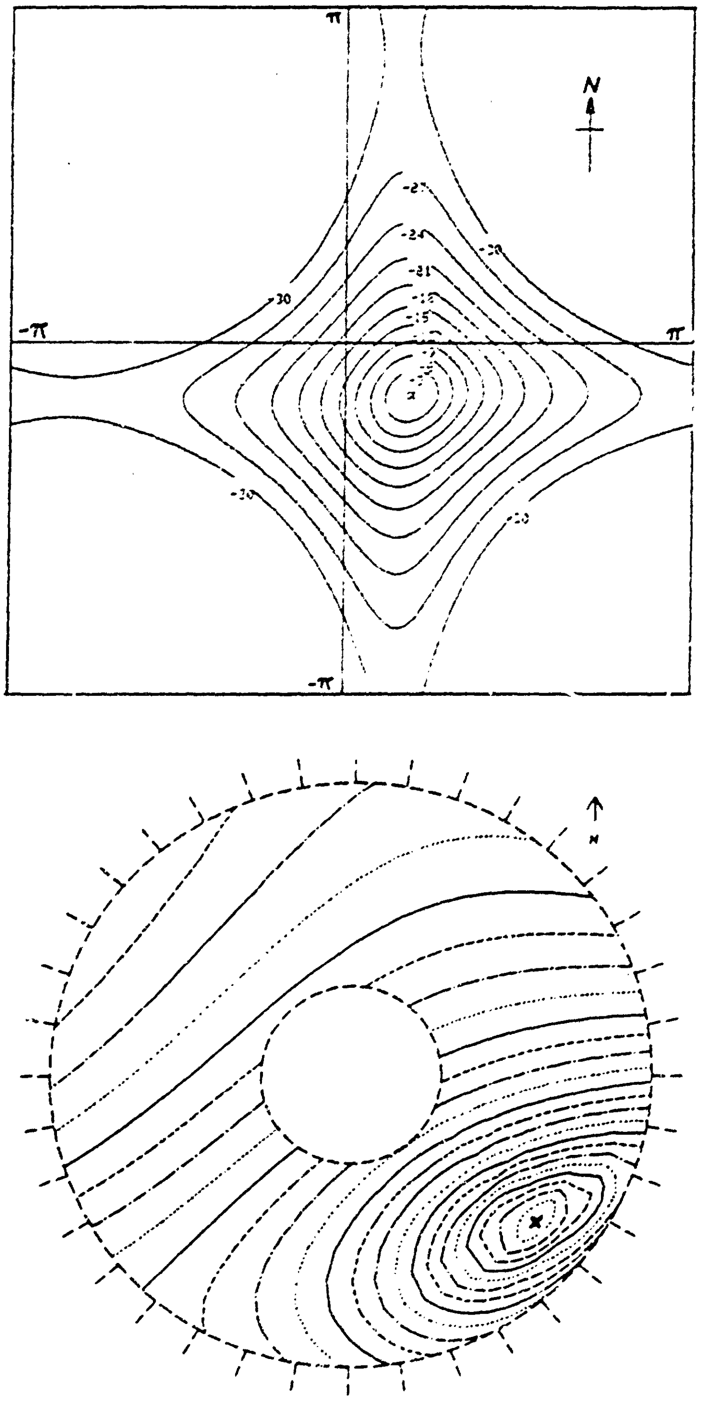
peak location = (0.2π,0.2π)

5.8 Results Using Real Data

To conclude the characterization of the new algorithm for 2-D ME PSE, the algorithm was used to obtain the ME PS estimate for real data gathered by a 2-D array of sensors. The sensor array consisted of nine microphones equally spaced in a 3x3 format on a square grid. The data gathered was the sound of a helicopter flying past the array. Each of the nine sensors was used to record time series. Thus, the data was actually three dimensional, with two spatial and one temporal dimension. However, since a large amount of data was available in the time dimension, it was decided to do a simple periodogram analysis, and then obtain high resolution estimates for the spatial power spectrum at a particular temporal frequency. The objective of the experiment was to determine the location of the helicopter in terms of azimuth and elevation with respect to the sensor array.

Given the time series at each of the microphones, the data was divided into 512 point sections. Periodograms of each section were obtained via the FFT and nine sections were then averaged to reduce the bias. The resulting temporal spectra were then analyzed for spectral peaks which indicated the presence of a strong signal (depending on the temporal frequencies being generated by the helicopter engine and rotors). These temporal frequencies were then chosen for spatial analysis. The nine channels were correlated at the chosen temporal frequency, and the resulting ACF values used to obtain both Maximum Likelihood and Maximum Entropy estimates of the spatial power spectrum.

The results of the experiment are shown in Fig. 5.19(a) and (b). Fig. 5.19(a) shows the ME estimate while Fig. 5.19(b) shows the ML estimate. The location of the helicopter is well determined by both of the methods, with the estimated location lying well within the error tolerance of the experiment. The spectral



(a) Maximum

Entropy estimate.

Horizon is located at $|\omega| = \pi$. The centre point is the location of the sensors. 3x3 ACF; CINC = 3dB

(b) Maximum Likelihood

estimate.

Polar plot. Outer circle is the horizon Highest contour is 0dB; CINC = 1dB

Fig. 5.19: 2-D ME and ML spectral estimates for real data. Spatial power spectra showing the location of a flying helicopter.

peak, however, is much sharper for the ME case, as was to be expected, and this could prove to be the deciding factor when high resolution estimates are desired in the presence of multiple targets.

5.9 Summary

This chapter has been concerned with obtaining a characterization of the ME method of PSE for two dimensional signals The experiments run were for data sets consisting of sinusoids in white Gaussian noise, using both exact ACF values as well as estimated ACF values. It was found that like the one dimensional case, the estimation of power spectra for real data gives erroneous peak location estimates when the peaks are located in certain regions of the 2-D frequency plane. The special regions are caused by the periodicity and symmetry of the spectrum and examples were run to demarcate the size of these regions. It was found that the errors in spectral peak location were most pronounced for small sizes of the given ACF segment (corresponding to low model orders in the 1-D case), and for low S/N ratios. The size of these special regions also depended on these two factors. Complex data caused no problems, and in fact, it was found that the ME PS estimates for complex data did not depend on the absolute frequency location of the peaks.

Increasing the S/N ratio increased the resolution of the ME PS estimates For the case of real data, the special regions mentioned above, were found to decrease in size with increasing S/N ratio, while for both real and complex data, the spectral peaks grew sharper in the PS estimates and the peak location accuracy improved. It was found that two peaks located close together that could not be resolved at low S/N ratios could be resolved at high S/N ratios

The resolution performance of the ME method was compared to that of the ML method and the Bartlett estimates It was found that like the 1-D case, the ME method afforded the highest resolution of the three techniques

The effect of changing the size of the known ACF gave results similar to those obtained by increasing the S/N ratio. Increasing the size of the ACF support resulted in sharper spectral peaks, more accurate peak location estimates and, for real data, smaller regions where the peak location accuracy may be expected to suffer. Changing the shape of the ACF support region results in distorted spectral estimates, with the distortion being proportional to the deviation away from a square support shape. Similarly, missing correlation values cause a distortion in the shape of the estimates, with the amount of distortion being proportional to the number of missing values.

For the case of estimated ACF values, it was found that the ME spectral estimates improved very rapidly with increasing data length. The effect of the starting phase of sinusoidal data was investigated and it was found that the location of the spectral peak oscillated about the true position for different values of the initial phase, similar to the 1-D case Finally, an example using real data was shown and the results were compared to those obtained via the ML method.

Chapter 6

The 1-D ME Problem With Missing Correlation Points

6.1 Introduction

The Maximum Entropy power spectrum estimation problem for 1-D signals, when the ACF values are known over a uniform grid, is linear, and analytically tractable. In fact, as was mentioned in chapter II, the solution to this problem is identical to that obtained by the mean-square error minimization of the prediction filter based on autoregressive signal modeling, which involves solving a set of linear equations for the filter coefficients. However, in many important applications of power spectrum estimation, the sensors used to gather the data are not placed on a uniform grid. In these applications, it is possible that the ACF values may not be estimated over a connected region (that is, the ACF support region 'A' may have missing points). In this case, the 1-D ME problem is also non-linear and no closed form solution has yet been proposed.

There are two basic alternatives in this situation. One is to interpolate the missing ACF values by some algorithm and then obtain an autoregressive filter via the linear problem formulation. The other is to obtain the true ME solution by using the new algorithm developed in chapter III to solve the non-linear problem. Although the interpolation approach may impose a lighter computational load, it presupposes extra information about the signal which may not be available, or which may impose unreasonable constraints on the data which is the very problem that the ME formulation tries to avoid. A recent study by Dowla[30] has shown that the ME spectral estimates give better spectral peak locations than the interpolation approach, especially when the ACF values are spaced very non-uniformly.

6.2 Application of the New Algorithm to 1-D Data

The algorithm developed in chapter III is completely general and is not restricted by the dimensionality of the signal to be analyzed. As such, it is immediately applicable to the 1-D problem. Further, a careful study of the algorithm will indicate that it is not restricted to any particular shape for the ACF support region 'A'. Thus, in the 1-D case, the region 'A' need not be a connected segment of the uniform grid which is required to obtain the closed form solution. The ME method can be viewed as a technique to extrapolate the known segment of the ACF. In the case of missing correlation values, the ME technique not only extrapolates the ACF beyond the known limits, but also interpolates the missing values during the course of obtaining the solution.

6.2.1 Comparison With The Closed Form Solution

First, consider the case of 1-D signals when the ACF is given over a uniform grid, for $-N \leq n \leq N$. Due to the existence of a closed form solution, this problem is ideal in illustrating that the solution obtained from the iterative algorithm developed in chapter III is indeed the ME PS estimate. Fig. 6.1(a) shows the ME PS estimate obtained via the iterative algorithm for the data parameters shown in the figure. Fig. 6.1(b) shows the result obtained from the closed form solution. Fig. 6.1(b) also shows the PS estimate of Fig. 6.1(a) (dotted line). It is clear from Fig. 6.1(b) that the two results are identical, and that the iterative algorithm does indeed lead to the ME power spectrum estimate. Many other examples have been tried to verify this conclusion.

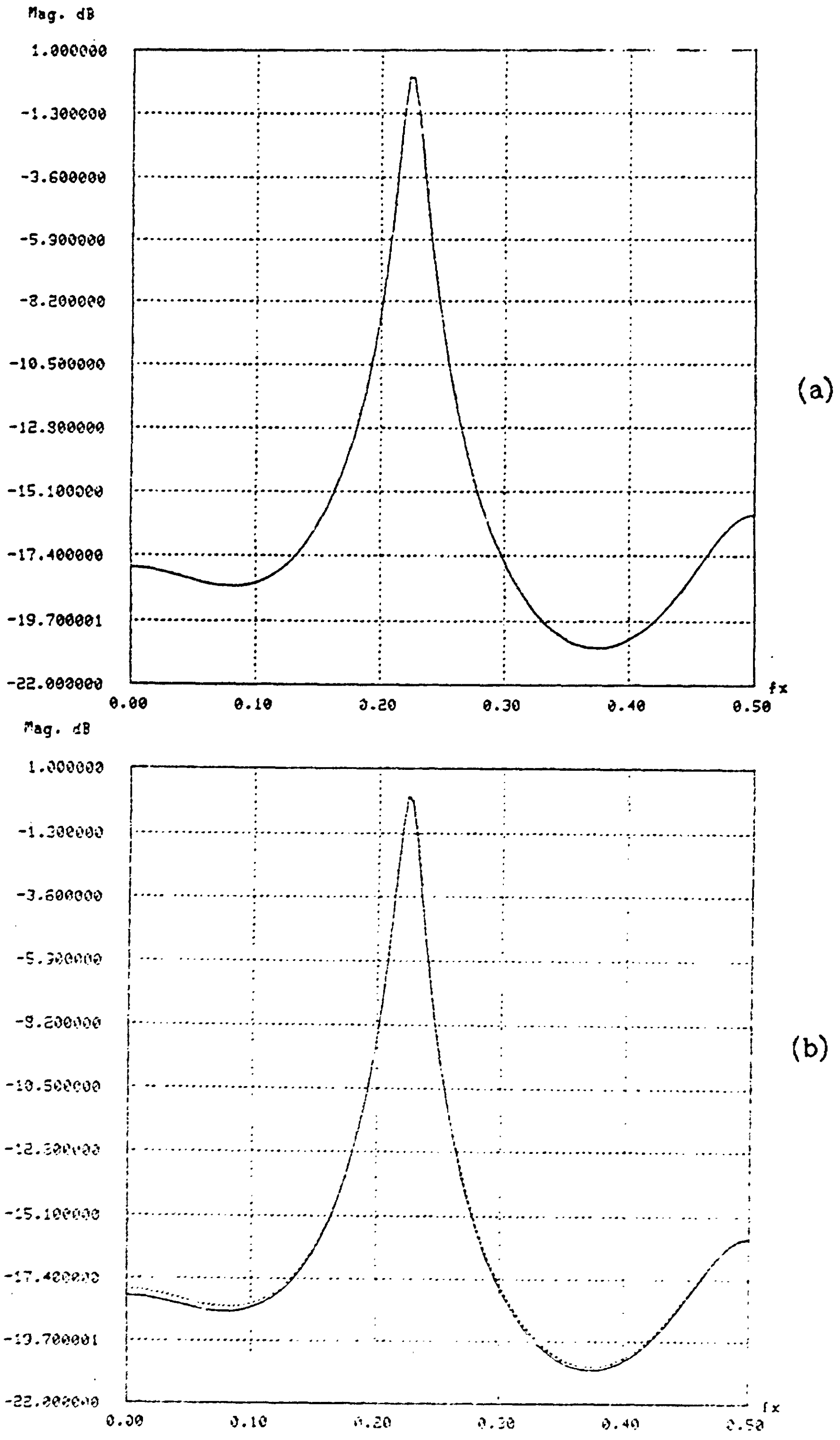


Fig. 6.1: The closed form solution compared to the iterative solution for 1-D signals. (a) Iterative solution. Peak location = 0.44π, fourth order model; SNR = 5dB; 11 iterations (5 secs. CPU time). (b) Closed form solution (solid line), solution of (a) is shown as a dotted line.

6.2.2 Missing Correlation Points

When the shape of the ACF support region 'A' is arbitrary, a closed form solution is no longer available, and the new algorithm assumes practical significance. Figure 6.2 shows the ME PS estimate obtained for the same data as was used for the example shown in figure 6.1 except that now the correlation values at lags $n = \pm 3$ are missing. The figure also shows the results of figure 6.1 (dotted line) superimposed on the estimate. The peak locations are seen to be shifted but the estimate retains its high resolution nature. Figure 6.3 shows another example of ME PS estimation for data with missing correlation values.

6.3 Conclusions

In addition to the above examples, a variety of other 1-D examples have been considered, and in all cases it has been observed that the iterative algorithm developed in Chap. 3 leads to the ME PS estimates. In the cases where the region 'A' is connected (no missing points on the uniform grid), the results are indistinguishable from the closed form solutions. Further, we have observed that the iterative algorithm leads to the ME solution for the PS estimation problem even when there are missing points in the ACF. In these cases, the ME method has been seen to generally preserve its high resolution characteristics, as long as the number of missing points is small as compared to size of the region 'A'.

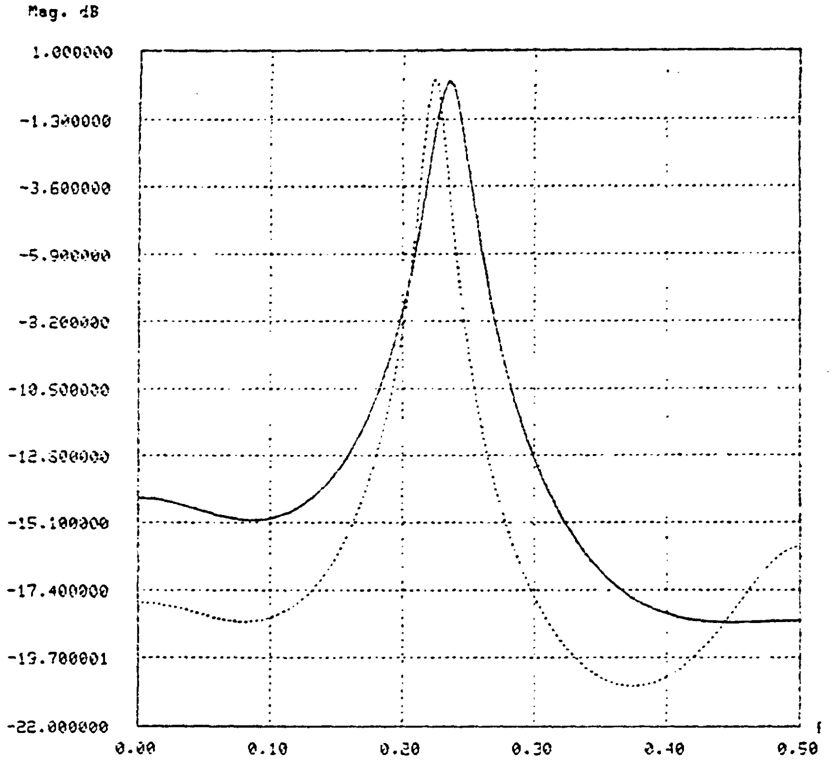


Fig. 6.2: The 1-D ME solution with missing correlation values. Peak location = 0.44π, fourth order model, correlation lags ±3 missing; the solution of Fig. 6.1 is shown as a dotted line. SNR = 5dB; 19 iterations (7.5 secs. CPU time).

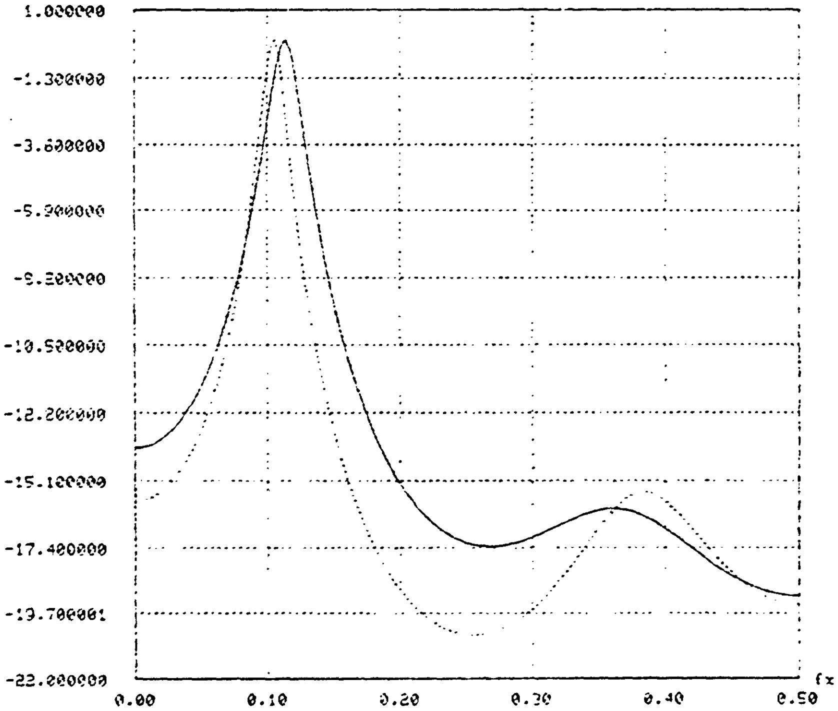


Fig. 6.3: 1-D ME PSE example with 4 missing correlation lags. Peak location = 0.2π, fourth order model, correlation lags ±2 and ±3 missing. SNR = 5dB. Dotted line shows the solution with no missing lags.

Chapter 7

Summary and Conclusions

7.1 Summary

This thesis tackled the problem of two dimensional Maximum Entropy power spectrum estimation. The problem was formulated and it was shown that the ME PS estimate is always autoregressive in nature, regardless of the dimensionality of the problem.The dependence of the solution on the signal dimensionality was pointed out. In particular, it was noted that whereas the 1-D ME PSE problem could be formulated as a linear one, with an analytic solution, the 2-D problem was highly non-linear, and no closed form solution existed.

A new algorithm for solving the highly non-linear 2-D problem was then developed. The algorithm is iterative in nature and iterates between the known correlation values and the AR coefficients that form the model for the spectrum. The requisite constraints are imposed in each of the two domains to arrive at the desired solution. The algorithm uses the Fast Fourier transform algorithm to achieve computational efficiency.

The implementation issues were then explored and different implementations for main-frame and mini-computers were developed. The mini-computer implementation uses direct computation of the DFT and therefore requires a minimal amount of storage. This implementation possesses viability for parallel processing hardware, in which case the slight loss of computational efficiency would be more than accounted for.

The properties of the 2-D ME estimates were then investigated and compared to the ML and Bartlett estimates. It was shown that the symmetry and periodicity of the power spectrum for real signals causes problems in the determination of spectral peak locations when the peaks are located in certain regions of the 2-D frequency plane. These special regions were demarcated and it was shown that their size depends on the S/N ratio (SNR) as well as on the size of the ACF support region 'A'. It was seen that complex data did not suffer from similar problems. The effect of changing the SNR on the spectral estimates was investigated. It was seen that closely spaced spectral peaks that could not be resolved at lower S/N ratios could be resolved at higher S/N ratios. The ME estimates were compared to the ML and Bartlett estimates and it was shown that the ME afforded the highest resolution of the three.

The effect of the size of the ACF support region on the ME PS estimates was also investigated and it was seen that the results were similar to increasing the SNR and higher resolution was achieved when the size of the ACF support region was increased. It was shown that the resolution properties of the ME estimates depended on the specific shape of the ACF support. The accuracy of peak location estimation for the ME ML and Bartlett methods were compared for the case of exact and estimated ACF values. It was found that although the accuracy for the three methods was comparable and excellent for the case of a single sinusoid, the ML and Bartlett techniques performed better for the two-sinusoid cases. For the case of estimated ACF values, the effects of the size of the data and the initial phase of the sinusoid were also investigated. It was seen that the ME estimates improved most rapidly as compared to the other methods when the data length was increased. The location of the spectral peak was found to oscillate about the true location when the initial phase of the sinusoid was changed. The amplitude of the oscillations was seen to decrease with increasing

SNR.

The algorithm was also shown to be applicable to the case when there were missing points in the ACF. Examples illustrating the effect cf missing ACF values on the ME estimates were shown. Finally, the algorithm was used on real field data and the results compared to the ML estimate.

It was seen that the development of the algorithm was completely general and did not restrict its application to signals of a particular dimensionality. Thus, the algorithm was also seen to be applicable to the one dimensional problem and proved useful in obtaining the ME PS estimates when there were missing correlation values.

7.2 Conclusions

This thesis has been concerned with developing a practical algorithm for 2-D ME PS estimation and investigating the properties of the estimates. The algorithm has been found to be a viable technique for obtaining the ME estimates, and has even proved to be extremely useful for the case of 1-D signals with missing correlation points. The extendibility problem of the ACF and similar theoretical issues have not been explicitly addressed in this thesis. The works of Woods[2] and Lang[22] would form excellent complementary reading.

It has been found that the 2-D ME estimates, like their 1-D counterparts, afford higher resolution than the Bartlett and ML methods. Although as estimators of the spectral peak location, the ME estimates show a larger deviation, excellent power spectral estimates can be formed even with very short data lengths at moderate S/N ratios, which has proved to be the strength of the ME method. It is not yet known how to form estimates of the power in the spectral peaks from the ME estimates. In the 1-D case, the area under the peaks is proportional to the power. It is conjectured that for the 2-D case, the volume under the spectral peaks would be proportional to the power. No proof of convergence for the algorithm has been forthcoming. If such a proof were to become available, it would place the new algorithm on a firm theoretical footing, and possibly point out means by which the convergence rates could be speeded up.

APPENDIX A

COMPUTER PROGRAM FOR MINI-COMPUTER

```
/*      MINCOMP.C                                               */
/*      PROGRAM TO OBTAIN THE 2-D MEM POWER SPECTRUM            */
/*      ESTIMATES BY DIRECT DFT COMPUTATION. THE                */
/*      STORAGE REQUIREMENTS ARE MINIMIZED TO ALLOW             */
/*      IMPLEMENTATION ON A MINI-COMPUTER                       */
/*      HANDLES COMPLEX VALUED AUTOCORRELATIONS.                */
/*      FOR REAL ACF, COMPUTATION CAN BE FURTHER MINIMIZED.     */
/*                                                              */
/*      NAVEED A. MALIK                                         */
/*      AUGUST, 1980                                            */

#include <stdio.h>
#include <math.h>
#include <complex.h>

/* useful constants                                                     */
#       define PI       3.14159265
#       define TWOPI    6.28318531

#define MN1     25      /* max ACF size. for C, use MN and Fortran indices */
#define MN      26      /*                                      */
#define M2N     52      /* 2*MN for complex data                */
#define MN2     13      /* Centre point of ACF array etc.       */
#define MN3     14      /*                                      */
#define MN4     26      /*                                      */
#define ZTST    1.0e-4  /* Error level to be achieved           */
#define ZTST1   2.5e-4  /* Alternate error level                */
#define ALPTST  0.99999 /* Maximum permissible alpha value      */

int n,n1,n2,n21,m1,m2,gap[MN][MN],nsin,ngap,n12;
float x1,x2,n4,pin,p[20], xfreq[20], yfreq[20], noise;

/*      gap:    gap array specifying missing ACF values */
/*      ngap:   no. of gaps                             */
/*      nsin:   no. of sinusoids                        */
/*      noise:  noise power                             */


main(argc,argv)
int argc ;
char ** argv;
{
FILE *fp, *fid, *fopen();
complex xlam[MN][MN],xold[MN][MN],r[MN][MN],px[MN][MN];

/*      xlam:   MEM filter values
        xold:   filter values from previous iteration
        r:      the given ACF values                    */

float zold,sclf,alpha,beta,den,z,xx,bmin,x3,x4,cft2();
float pxdum[M2N],fdx;
int nitr,i,j,k,is,js,gdum[MN],js2,js21;

/*      RESTARTING FROM SAVED DATA      */
/*      VALUABLE IF YOUR SYSTEM IS PRONE TO CRASHES!!!  */

if(argc >= 2){
        fp = fopen("save","r");
        fread((char *)&n, 2, 1, fp);
        fread((char *)&n1, 2, 1, fp);
        fread((char *)&n2, 2, 1, fp);
        fread((char *)&n21, 2, 1, fp);
        fread((char *)&m1, 2, 1, fp);
fread((char *)&m2, 2, 1, fp);
for(is=0; is<MN; is++){
        fread((char *)gdum, 2, MN, fp);
        for(js=0; js<MN; js++)
                gap[is][js] = gdum[js];
}
for(is=0; is<MN; is++){
        fread((char *)pxdum, 4, M2N, fp);
        for(js=0; js<MN; js++){
                js2 = 2 * js;
                js21 = js2 + 1;
                px[is][js] = zcplx(pxdum[js2],pxdum[js21]);
        }
}
for(is=0; is<MN; is++){
        fread((char *)pxdum, 2, M2N, fp);
        for(js=0; js<MN; js++){
                js2 = 2 * js;
                js21 = js2 + 1;
                xlam[is][js] = zcplx(pxdum[js2],pxdum[js21]);
        }
}
for(is=0; is<MN; is++){
        fread((char *)pxdum, 4, M2N, fp);
        for(js=0; js<MN; js++){
                js2 = 2 * js;
                js21 = js2 + 1;
                xold[is][js] = zcplx(pxdum[js2],pxdum[js21]);
        }
}
for(is=0; is<MN; is++){
        fread((char *)pxdum, 4, M2N, fp);
        for(js=0; js<MN; js++){
                js2 = 2 * js;
                js21 = js2 + 1;
                r[is][js] = zcplx(pxdum[js2],pxdum[js21]);
        }
}
fread((char *)&x1, 4, 1, fp);
fread((char *)&x2, 4, 1, fp);
fread((char *)&n4, 4, 1, fp);
fread((char *)&pin, 4, 1, fp);
fread((char *)&zold, 4, 1, fp);
fread((char *)&sclf, 4, 1, fp);
fread((char *)&alpha, 4, 1, fp);
fread((char *)&beta, 4, 1, fp);
fread((char *)&den, 4, 1, fp);
fread((char *)&z, 4, 1, fp);
fread((char *)&nitr, 4, 1, fp);
fclose(fp);
}
else {

/*      compute ACF, specify gaps etc */
cacf2d(r);

/*      initialize all arrays */
zold=1.0e30;
sclf=0.5;
alpha=0.0;
beta=0.0;
z=1.0e30;

/*      */
```

```
for ( i = 0; i < MN; i++ ){
        for( j = 0; j < MN; j++ ){
                xold[i][j] = zcplx(0.0,0.0);
                xlam[i][j] = zcplx(0.0,0.0);
                px[i][j] = zcplx(0.0,0.0);
        }
}
xlam[MN2][MN2] = zcplx(1.0,0.0);
xold[MN2][MN2] = zcplx(1.0,0.0);
px[MN2][MN2] = zcplx(1.0,0.0);
x1 =1.0;
x2 = 1.0;

/*      error normalizing factor        */
den = 0.0;
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++)
                den=den + zmag2(r[i][j]);
}
den=den-1.0;
nitr=0;

nitr++;

/*      obtain correction ACF and error         */
z=0.0;
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                px[i][j] = zsub(r[i][j] ,px[i][j]);
                z = z + zmag2(px[i][j]);
        }
}
z = z/den;
}               /* end if statement     */

/*      main loop for it rations        */
while(z > ZTST ){
if((nitr % 1) == 0){
        printf("NITR = %d ALPHA = %e ERROR = %e \n",nitr,alpha,z);
        fp = fopen("save","w");
        fwrite((char *)&n, 2, 1, fp);
        fwrite((char *)&n1, 2, 1, fp);
        fwrite((char *)&n2, 2, 1, fp);
        fwrite((char *)&n21, 2, 1, fp);
        fwrite((char *)&m1, 2, 1, fp);
        fwrite((char *)&m2, 2, 1, fp);
        for(is=0; is<MN; is++){
                for(js=0; js<MN; js++){
                        gdum[js] = gap[is][js];
                }
                fwrite((char *)gdum, 2, MN, fp);
        }
        for(is=0; is<MN; is++){
                for(js=0; js<MN; js++){
                        js2 = 2*js;
                        js21 = js2 + 1;
                        pxdum[js2] = (px[is][js]).r;
                        pxdum[js21] = (px[is][js]).i;
                }
                fwrite((char *)pxdum, 4, M2N, fp);
        }
        for(is=0; is<MN; is++){
                for(js=0; js<MN; js++){
                        js2 = 2*js;
                        js21 = js2 + 1;
                        pxdum[js2] = (xlam[is][js]).r;
                        pxdum[js21] = (xlam[is][js]).i;
                }
                fwrite((char *)pxdum, 4, M2N, fp);
        }
        for(is=0; is<MN; is++){
                for(js=0; js<MN; js++){
                        js2 = 2*js;
                        js21 = js2 + 1;
                        pxdum[js2] = (xold[is][js]).r;
                        pxdum[js21] = (xold[is][js]).i;
                }
                fwrite((char *)pxdum, 4, M2N, fp);
        }
        for(is=0; is<MN; is++){
                for(js=0; js<MN; js++){
                        js2 = 2*js;
                        js21 = js2 + 1;
                        pxdum[js2] = (r[is][js]).r;
                        pxdum[js21] = (r[is][js]).i;
                }
                fwrite((char *)pxdum, 4, M2N, fp);
        }
        fwrite((char *)&x1, 4, 1, fp);
        fwrite((char *)&x2, 4, 1, fp);
        fwrite((char *)&n4, 4, 1, fp);
        fwrite((char *)&pin, 4, 1, fp);
        fwrite((char *)&zold, 4, 1, fp);
        fwrite((char *)&sclf, 4, 1, fp);
        fwrite((char *)&alpha, 4, 1, fp);
        fwrite((char *)&beta, 4, 1, fp);
        fwrite((char *)&den, 4, 1, fp);
        fwrite((char *)&z, 4, 1, fp);
        fwrite((char *)&nitr, 4, 1, fp);
        fclose(fp);
}               /*      end if statement        */

if(z > zold || beta != 0.0){
        if((beta == 0.0) && (zold <= ZTST1)){

                /*      accept higher error level as convergence.
  Then, better estimate is the previous lambda set.                     */

                for(i=m1; i<=m2; i++){
                        for(j=m1; j<=m2; j++)
                                xlam[i][j] = xold[i][j];
                }
                goto lbl888;
        }
        alpha = (1.0+alpha)/2.0;
        sclf = sclf/2.0;
}
zold = z;

/*      compute minimum value of correction spectrum and update alpha */
x3 = cft2(px);
xx = 1.0 + sclf * x2/x3;
alpha = (xx > alpha)? xx: alpha ;
if(alpha >= ALPTST)goto lab999;

/*      compute lambda values for region A only          */
cft3(xold,px,alpha,xlam);
```

```
/*      Minimum value of F[lambda], compute beta        */
x4 = cft2(xlam);
if(x4 > 0.0){
        beta =0.0;
}
else {
        bmin = -x4/( x1-x4 );
        beta = (1.0+ (1.0-sclf)*(1.0/bmin-1.0))*bmin ;
        alpha = (1.0+alpha)/2.0;
        printf(" bete = %e \n",beta);
}

/*      obtain new lambda set   */
for ( i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                xlam[i][j] = zadd(zscal(beta,xold[i][j]),
                zscal((1.0-beta),xlam[i][j]));
                xold[i][j] = xlam[i][j];
        }
}
fid = fopen("filter","w"):
i = 0;
fwrite((char *)&i,2,1,fid);
fwrite((char *)&nsin,2,1,fid);
for(i=1; i<=nsin; i++){                 /* FORTRAN indices here!!!! */
        fwrite((char *)&p[i],4,1,fid);
        fwrite((char *)&xfreq[i],4,1,fid);
        fwrite((char *)&yfreq[i],4,1,fid);
}
fwrite((char *)&noise,4,1,fid);
fwrite((char *)&n12,2,1,fid);
fwrite((char *)&ngap,2,1,fid);
if(ngap != 0){
        for(i=0; i<ngap; i++){
                for(j=m1; j<=m2; j++){
                        for(k=m1; k<=m2; k++){
                                if(gap[j][k] == 0){
                                        is = j-MN2;
                                        js = k-MN2;
                                        fwrite((char *)&is, 2, 1, fid);
                                        fwrite((char *)&js, 2, 1, fid);
                                }
                        }
                }
        }
}
fwrite((char *)&n, 2, 1, fid);
fwrite((char *)&nitr, 2, 1, fid);
fwrite((char *)&z, 4, 1, fid);
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                fdx = (xlam[i][j]).r;
                fwrite((char *)&fdx, 4, 1, fid);
                fdx = (xlam[i][j]).i;
                fwrite((char *)&fdx, 4, 1, fid);
        }
}
fclose(fid);
/*      obtain new ACF over the region A only   */
cft(px,xlam);

nitr++;

/*      obtain correction ACF and new error     */
```

```
z=0.0;
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                px[i][j] = zsub(r[i][j], px[i][j]);
                z = z+ zmag2(px[i][j]);
        }
}
z=z/den;
}               /*      end while loop  */

/*      check solution    */
printf("Converged at iteration no. %d \n",nitr);
printf("TEST IN PROGRESS: PATIENCE..... \n");
lb1888:;
if(z > zold)printf(" 2.0e-4 ERROR LEVEL, NITR = %d",nitr);
n=4*n;
n2=n/2;
n21=n2+1;
n4=n;
n4=n4*n4;
pin=2.0*PI/n;
cft(px,xlam);
/*
        If non positive definite solution, error message
        issued by cft.c. If O.K., then error rechecked below:     */
z=0.0;
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++)
                z = z + zmag2(zsub(r[i][j],px[i][j]));
}
z=z/den;
if(z <= ZLT)
        printf("Good solution. Error is %e \n",z);
else
        printf("Positive but high error %e \n",z);
lab999:;
if(alpha >= ALPTST)
        printf(" ALPHA = 0.99999; use longer dft length!!!! \n");
}       /* ******END MAIN PROGRAM******* */

cacf2d(r)
complex r[MN][MN];
{
float xx, wx, wy;
int i,j,i1,i2,ia,ja;
complex zw;

/*      request input data      */
printf("Enter no. of complex exponentials \n ");
scanf("%d", &nsin);
printf("Number of complex exponentials is %d \n",nsin);
for(i=1; i<=nsin; i++){
        printf("Enter power, fx, fy for exponential no. %d \n",i);
        scanf("%e %e %e",&p[i],&xfreq[i],&yfreq[i]);
        printf("power= %e fx= %e fy= %e \n",p[i],xfreq[i],yfreq[i]);
}
printf("enter noise power \n");
scanf("%e", &noise );
printf(" noise power is %e \n", noise);
printf(" enter dimension of smallest square containing 'A' \n");
scanf("%d", &n12);
printf("ACF without gaps is %d x %d \n",n12,n12);
printf(" enter DFT length \n");
scanf("%d",&n);
```

```
printf("DFT length is %d \n",n);
n2=n/2;
n21=n2+1;
n4=n;
n4=n4*n4;
n1=(n12-1)/2;
m1=MN2-n1;
m2=MN2+n1;
pin=2.0*PI/n;
printf("ACF array is from ( %d %d ) to ( %d %d ) \n",
                                                    m1,m1,m2,m2);

/*      fill gap array  */
for(i=0; i<MN; i++){
        for(j=0; j<MN; j++)
                gap[i][j] = 1;
}

/*      find where the gaps are          */
printf(" no. of gaps in upper half plane ? \n");
scanf("%d",&ngap);
if(ngap != 0){
for(i=1; i<=ngap; i++){
        printf("enter coords for gap no. %d \n",i);
        scanf("%d %d",&i1,&i2);
        gap[i1][i2] = 0:
}
}

/*      calculate ACF   */
for(i1=1; i1<=nsin; i1++){
wx = 2.0*PI*xfreq[i1];
wy = 2.0*PI*yfreq[i1];
for(i=m1; i<=m2; i++){
        ia=i-MN2;
        for(j=m1; j<=m2; j++){
                ja=j-MN2:
                zw = zcplx(0.0,(wx * ia + wy * ja));
                /*      zero out acf array before computing ACF */
                if(i1==1)r[i][j] = zcplx(0.0,0.0);
                if(gap[i][j]==1)
                        r[i][j]=zadd(r[i][j],zscal(p[i1],zexp(zw)));
                }
        }
}

/*      add noise power and normalize   */
r[MN2][MN2] = zadd(r[MN2][MN2],zcplx(noise,0.0));
xx=r[MN2][MN2].r;
xx = 1.0/xx;
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                r[i][j] = zscal(xx,r[i][j]);
        }
}
return(MN2);
}
cft(px,xlam)
complex px[MN][MN],xlam[MN][MN];
{
complex zx00,zx22,wz;
double npow();
float amin1(),amax1(),x00,x11,x22,xk1,wx,wy;
int i,j,k,l,m,nn,nn1;
```

```
/*      compute the (0,0) and (n/2,n/2) values first     */
zx00 = zcplx(0.0,0.0);
zx22 = zcplx(0.0,0.0);
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
        zx00 = zadd(zx00,xlam[i][j]);
zx22 = zadd(zx22,zscal(npow(-1.0,(i+j-MN4)),xlam[i][j]));
        px[i][j] = zcplx(0.0,0.0);
        }
}
x00 = zx00.r;
x22 = zx22.r;
if(x00 <= 0.0 || x22 <= 0.0){
printf("BAD SOLUTION: F(LAMBDA) < 0 \n");
return(MN2);
}
x1=amin1(x00,x22);
x2=amax1(x00,x22);
x00=1.0/x00;
x22=1.0/x22;

/*      place the idft contribution of x00 and
        x22 in the px array      */
for(i=m1; i<=m2; i++){
if(i<MN2)
        nn=MN3;
else
        nn=MN2;
for(j=nn; j<=m2; j++)
        px[i][j] = zcplx(x00+x22*npow(-1.0,(i+j-MN4)),0.0);

}

/*      compute all other independent x(k,l) values in the
        plane using direct complex exp function calls    */

zx00 = xlam[MN2][MN2];

/*      specify (k,l) values in the plane      */
for(k=0; k<n; k++){
for(l=0; l<n; l++){
        if((k==0)&&(l==0))continue;
        if((k==n)&&(l==n))continue;
        zx22 = zcplx(0.0,0.0);
        for(m=m1; m<=m2; m++){
                if(m < MN2)
                        nn1 = MN3;
                else
                        nn1 = MN2;
                for(nn=nn1; nn<=m2; nn++){
                        if((m==MN2)&&(nn==MN2))continue;
                        wx = pim * (k*(m-MN2) + l*(nn-MN2));
                        wz = zcplx(0.0,-wx);
                        wz = zexp(wz);
                    zx22 = zadd(zx22,zmul(xlam[m][nn],wz));
                }
        }
        /*              */
        xkl = zx00.r + 2.0*zx22.r;
        if(xkl<=0.0){
                printf("BAD SOLUTION: F(LAMBDA) < 0 \n");
                return(MN2);
        }
        x1=amin1(x1,xkl);
```

```
         x2=amax1(x2,xkl);
         xkl=1.0/xkl;
         /*
       Given the x(k,l) value, update the running sums for the
                 ACF R'(m,n)                                       */
         for(m=m1; m<=m2; m++){
                 wx = pin * k *(m- MN2);
                 if(m<MN2)
                         nn1=MN3;
                 else
                         nn1=MN2;
                 for(nn=nn1; nn<=m2; nn++){
                         if(gap[m][nn]==0)continue;
                         wy = pin * l *(nn-MN2);
                         wz = zcplx(0.0,(wx+wy));
                         wz = zscal(xkl,zexp(wz));
                         px[m][nn] = zadd(px[m][nn],wz);
                 }
         }
}
}

/*       half array of ACF values computed.
         rearrange to full array
         and normalize   */
for(m=m1; m<=m2; m++){
         for(nn=m1; nn<=MN2; nn++)
                 px[m][nn]=zcnjg(px[m1+m2-m][m1+m2-nn]);
}
x00=(px[MN2][MN2]).r;
x11=x00/n4;
x00 = 1.0/x00;
for(i=m1; i<=m2; i++){
for(j=m1; j<=m2; j++){
         px[i][j] = zscal(x00,px[i][j]);
         xlam[i][j] = zscal(x11,xlam[i][j]);
}
}
x2=1.0/(x2*x11);
x1=x1*x11;
return(m1);
}

float cft2(px)
complex px[MN][MN];
{
complex zx00,zx22,wz;
double npow();
float amin1(),amax1(),x00,x22,xkl,wx,xmin;
int i,j,k,l,m,nn,nn1;

/*       compute the (0,0) and (n/2,n/2) values first    */
zx00 = zcplx(0.0,0.0);
zx22 = zcplx(0.0,0.0);
for(i=m1; i<=m2; i+){
         for(j=m1; j<=m2; j++){
                 zx00 = zadd(zx00,px[i][j]);
       zx22 = zadd(zx22,zscal(npow(-1.0,(i+j-MN4)),px[i][j]));
         }
}
x00 = zx00.r;
x22 = zx22.r;
xmin = amin1(x00,x22);
```

```
/*      compute all other x(k,l) values in the
        plane using direct complex exp function calls   */
zx00 = px[MN2][MN2];

/*      specify (k,l) values     */
for(k=0; k<n; k++){
        for(l=0; l<n; l++){
                if((k==0)&&(l==0))continue;
                if((k==n)&&(l==n))continue;
                zx22 = zcplx(0.0,0.0);
                for(m=m1; m<=m2; m++){
                if(m < MN2)
                        nn1 = MN3;
                else
                        nn1 = MN2;
                for(nn=nn1; nn<=m2; nn++){
                        if((m==MN2)&&(nn==MN2))continue;
                        wx = pin * (k*(m-MN2) + l*(nn-MN2));
                        wz = zcplx(0.0,-wx);
                        wz = zexp(wz);
                        zx22 = zadd(zx22,zmul(px[m][nn],wz));
                }
        }
        /*                */
        xkl = zx00.r + 2.0*zx22.r;
        xmin = amin1(xkl,xmin);
}
}
return(xmin);
}

cft3(xold,px,alpha,xlam)
complex xold[MN][MN],px[MN][MN],xlam[MN][MN];
float alpha;
{

complex zx00,zx22,wz,zy00,zy22;
double npow();
float amin1(),amax1(),x00,x22,xkl,ykl,wx,wy;
int i,j,k,l,m,nn,nn1;

/*      compute the (0,0) and (n/2,n/2) values first     */
zx00 = zcplx(0.0,0.0);
zx22 = zcplx(0.0,0.0);
for(i=m1; i<=m2; i++){
        for(j=m1; j<=m2; j++){
                wx = npow(-1.0,(i+j-MN4));
                zx00 = zadd(zx00,px[i][j]);
                zx22 = zadd(zx22,zscal(wx, px[i][j]));
                zy00 = zadd(zy00,xold[i][j]);
                zy22 = zadd(zy22,zscal(wx, xold[i][j]));
                xlam[i][j] = zcplx(0.0,0.0);
        }
}
wx = zx00.r;
wy = zy00.r;
x00 = 1.0/wy +(1.0 - alpha)* wx;
x00 = 1.0/x00;
wx = zx22.r;
wy = zy22.r;
x22 = 1.0/wy + (1.0 - alpha)* wx;
x22 = 1.0/x22;
```

```
/*      place the idft contributions of
        x00 and x22 in the lambda array */
for(m=m1; m<=m2; m++){
        if(m<MN2)
                nn1=MN3;
        else
                nn1=MN2;
        for(nn=nn1; nn<=m2; nn++)
     xlam[m][nn] = zcplx(x00+x22*npow(-1.0,(m+nn-MN4)),0.0);
}

/*      compute all other values in the plane      */

zx00 = px[MN2][MN2];
zy00 = xold[MN2][MN2];

/*      specify the k,l values                             */
for(k=0; k<n; k++){
        for( l=0; l<n; l++){
                if((k==0)&&(l==0))continue;
                if((k==n)&&(l==n))continue;
        zx22 = zcplx(0.0,9.0);
        zy22 = zx22;
        for(m=m1; m<=m2; m++){
                if(m<MN2)
                        nn1 = MN3;
                else
                        nn1 = MN2;
                for(nn=nn1; nn<=m2; nn++){
                        if((m==MN2)&&(nn==MN2))continue;
                        wx = pin * (k*(m-MN2) + l*(nn-MN2));
                        wz = zcplx(0.0,-wx);
                        wz = zexp(wz);
                        zx22 = zadd(zx22,zmul(px[m][nn],wz));
                 zy22 = zadd(zy22,zmul(xold[m][nn],wz));
                }
        }
        wx = zx00.r;
        wy = zx22.r;
        xkl = wx + 2.0*wy;
        wx = zy00.r;
        wy = zy22.r;
        ykl = wx + 2.0*wy;
        xkl = 1.0/ykl+(1.0-alpha)*xkl;
        xkl = 1.0/xkl;

 /*     update the running sums for the lambda values    */

        for(m=m1; m<=m2; m++){
                wx = pin * k * (m - MN2);
                if(m<MN2)
                        nn1=MN3;
                else
                        nn1=MN2;
                for(nn=nn1; nn<=m2; nn++){
                        if(gap[m][nn]==0)continue;
                        wy = pin * l * (nn - MN2);
                        wz = zcplx(0.0.(wx+wy));
                        wz = zscal(xkl,zexp(wz));
                        xlam[m][nn] = zadd(xlam[m][nn],wz);
                }
        }
 }
 }
```

```
/* half lambda array computed. rearrange and normalize     */

for(m=ml; m<=m2; m++){
        for(nn=ml; nn<=MN2; nn++)
                xlam[m][nn]=zcnjg(xlam[ml+m2-m][ml+m2-nn]);
}
wx = 1.0/n4;
for(m=ml; m<=m2; m++){
        for(nn=ml; nn<=m2; nn++)
                xlam[m][nn] = zscal(wx,xlam[m][nn]);
}
return(m2);
}
/*      function to obtain the minimum of two numbers   */
float aminl(x,y)
float x,y;
{
        return((x<y)?x:y);
}

/*      function to obtain the maximum of two numbers   */
float amaxl(x,y)
float x,y;
{
        return((x>y)?x:y);
}
```

References

1. J. Makhoul, "Spectral Analysis of Speech by Linear Prediction," *IEEE Trans. Audio Electroacoust.* **AU-21**, pp.140-148 (June 1973).

2. J. W. Woods, "Two-Dimensional Markov Spectral Estimation," *IEEE Trans. Inform. Theory* **IT-22**, pp.552-559 (Sept. 1976).

3. H. C. Andrews and B. R. Hunt, pp. 90-112 in *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, N.J. (1977).

4. J. H. McClellan and R. J. Purdy, "Application of Digital Signal Processing to Radar," pp. 239-326 in *Applications of Digital Signal Processing*, ed. A. V. Oppenheim, Prentice-Hall, Englewood Cliffs, N.J. (1978).

5. A. B. Baggeroer, "Sonar Signal Processing," pp. 331-428 in *Applications of Digital Signal Processing*, ed. A. V. Oppenheim, Prentice-Hall, Englewood Cliffs, N.J. (1978).

6. G. M. Jenkins and D. G. Watts, pp. 140-202 in *Spectral Analysis and its Applications*, Holden-Day, San Francisco (1968).

7. A. V. Oppenheim and R. W. Schafer, pp. 532-571 in *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J. (1975).

8. P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time-Averaging Over Short, Modified Periodograms," *IEEE Trans. Audio Electroacoust.* **AU-15**, pp.70-73 (June, 1967).

9. G. E. P. Box and G. M. Jenkins, in *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, Cal. (1976).

10. A. Papoulis, "A New Algorithm in Spectral Analysis and Bandlimited Extrapolation," *IEEE Trans. Circuits & Sys.* **CAS-22**(9), pp.735-742 (Sept. 1975).

11. J. A. Cadzow, "An Extrapolation Procedure for Bandlimited Signals," *IEEE Trans. Acoust. Speech & Sig. Proc.* **ASSP-27**(1) (Feb. 1979).

12. J. Capon, "High resolution frequency-wavenumber analysis," *Proc. IEEE* **57**, p.1408 (Aug. 1969).

13. J. P. Burg, "Maximum Entropy Spectral Analysis," *37th Meeting Soc. Exploration Geophysicists* (1967).

14. J. V. Pendrell, *The Maximum Entropy Principle in Two-Dimensional Spectral Analysis*, York University, Toronto, Ontario, Canada (Nov. 1979). Ph.D. Dissertation

15. R. W. Gerchberg, "Super Resolution Through Error Energy Reduction," *Optica Acta* **21**(9), pp.709-720 (1974).

16. J. P. Burg, "The Relationship between Maximum Entropy Spectra and Maximum Likelihood Spectra," *Geophysics* **37**, p.375 (1972).

17. J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE* **63**, pp.561-580 (Apr. 1975).

18. D. E. Smylie, G. K. C. Clarke, and T. J. Ulrych, "Analysis of Irregularities in the Earth's Rotation," pp. 412-416 in *Methods in Computational Physics* (1973).

19. C. E. Shannon and W. Weaver, in *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL (1949).

20. G. Szego, *Math. Z.* **6**, p.167.

21. A. VanDenBos, "Alternative Interpretation of Maximum Entropy Spectral Analysis," *IEEE Trans. Inform. Theory (Corresp.)* **IT-17**, pp.493-494 (July 1971).

22. B. W. Dickinson, "Two-Dimensional Markov Spectral Estimates need not exist," *IEEE Trans. Inform. Theory* **IT-20**, pp.120-121 (Jan. 1980).

23. S. W. Lang, *Spectral Estimation for Sensor Arrays*, M.I.T., Cambridge, MA (Aug. 1981). Ph.D. Dissertation

24. J. P. Burg, *Maximum Entropy Spectral Analysis*, Stanford University, Stanford, California (1975). Ph.D. Thesis

25. S. J. Wernecke and L. R. D'Addario, "Maximum Entropy Image Reconstruction," *IEEE Trans. Comput.* **C-26**, pp.351-364 (April, 1977).

26. A. K. Jain and S. Ranganath, "Two Dimensional Spectral Estimation," *Proc. RADC Spectrum Estimation Workshop*, pp.151-157 (May, 1978).

27. C. Ong, "An Investigation of two new High-Resolution Two-Dimensional Spectral Estimation Techniques," Special Report, No. 1, Long Period Array Processing Development, Texas Instruments Inc. (D).

28. S. E. Roucos and D. G. Childers, "A Two-Dimensional Maximum Entropy Spectral Estimator," *IEEE Trans. Inform. Theory* **IT-26**(5) (Sept. 1980).

29. W. I. Newman, "A New Method of Multidimensional Power Spectral Analysis," *Astron. Astrophys.* **54**, p.369 (1977).

30. J. O. Eklundh, "A Fast Computer Method for Matrix Transposing," *IEEE Trans. Computers (Corresp.)* (July 1972).

31 F. U. Dowla, *MEM Spectral Analysis for Nonuniformly Sampled Signals*, Massachusetts Institute of Technology, Cambridge, MA (April, 1981). M.S. thesis