# Array Processing for Composite Wavefront Decomposition

By

Owen Simeon Halpeny

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
1972

To Michael Owen Halpeny

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to Dr. D. G. Childers, chairman of his supervisory committee, for his invaluable and timely inspiration, suggestions and encouragement. He also wishes to thank Dr. T. S. George and Dr. N. W. Perry, Jr., the other members of his supervisory committee for their guidance.

The valuable discussions with Miss Julia McCoy and other staff members of the Visual Science Laboratory are gratefully acknowledged.

The author also wishes to especially thank Mrs. Carol Halpeny who assisted in the coding of the programs, typed the manuscript, prepared the drawings, and offered many valuable comments regarding the organization of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the
Graduate Council of the University of Florida in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy

ARRAY PROCESSING FOR COMPOSITE WAVEFRONT DECOMPOSITION

By

Owen Simeon Halpeny

August, 1972

Chairman: Dr. D. G. Childers
Major Department: Electrical Engineering

This study considers the problem of detecting and estimating multiple planewaves in a noise environment. The data are assumed to be available in array form, i.e., several spatially distinct time records of the same process. The solution which is pursued is computationally efficient, but generally suboptimal. However, under some conditions, the solution is optimum in the maximum-likelihood sense.

The determination of the number of planewaves and their respective vector velocities is accomplished by using the multidimensional frequency-wavenumber power spectral density function. This technique avoids the complexity of more elaborate schemes and yields usable results for sufficiently high signal-to-noise ratios. The multiwave problem is reduced to a sequence of single wave estimation problems by using frequency-wavenumber digital filters which pass a single planewave. The multidimensional fast Fourier transform is used for both the spectral estimation and digital filtering.

After filtering, maximum-likelihood estimates are made for each wave. The estimates are compared to the simpler delay and sum estimates.

All of the techniques advanced in this study are tested using simulated and actual data on a large scale digital computer. It is found that the spectral density estimation and filtering can be accomplished efficiently and effectively. By comparison, the general maximum-likelihood estimators are computationally cumbersome and offer little improvement over the delay and sum estimates.

# CHAPTER I

# INTRODUCTION

The analysis of spatio-temporal signals which are immersed in noise is undertaken here because of the current interest in many fields of both theoretical and practical considerations. Typical research includes the radar and sonar problem of locating and classifying targets that emit or reflect waves of energy. In radio astronomy the purpose is to study electromagnetic radiation from different regions of the universe. In seismology one is frequently interested in [1] differentiating between earthquakes and nuclear explosions.

The original motivation for this study is based upon the conjecture that these sophisticated techniques may be of value in the study of electroencephalographs (EEG) or brain waves. Accordingly, this study is limited to techniques which have definite practical merit. The signal model is a sum of planewaves which overlap in space and time. The noise consists of both ambient channel noise and interfering planewaves. It is generally assumed that the noise statistics are available, but the signals are completely unknown a priori. Thus, while this study is guided by a particular application, the formulation of the problem is sufficiently general to permit application to many other areas.

The basic quantity being studied is a wave. In general terms a wave is simply a phenomenon or disturbance which moves. The shock wave caused by an explosion or the tidal wave caused by an earthquake are obvious examples. A planewave is simply a wave which travels in a straight line and does not exhibit variations in any plane perpendicular to the path of propagation. In the examples cited the wave travels away from the source of disturbance. In complex biological systems this simple relationship may not exist.

Mathematically a planewave may be defined as

$$s(t - d/v)$$

where t is time, d is distance as measured from the spatial origin along the direction of propagation, v is the velocity of the wave, and s(t) is the waveshape or functional form of the wave at the spatial origin. Using the conventional Cartesian coordinate system, this wave becomes

$$s(t - \vec{\alpha}\cdot\vec{r})$$

where $\vec{r} = (r_x, r_y, r_z)$ is the position vector at which s is evaluated and $\vec{\alpha}$ is the inverse velocity vector

$$\vec{\alpha} = \vec{v}/|\vec{v}|^2 \quad .$$

A monochromatic planewave in complex exponential form is given by

$$e^{-i2\pi(ft-f\vec{\alpha}\cdot\vec{r})} = e^{-i2\pi(ft-\vec{k}\cdot\vec{r})}$$

where the vector wavenumber is $\vec{k} = f\vec{\alpha}$. Any spatio-temporal function which is transformable may be expressed as a sum of monochromatic planewaves.

$$s(t,\vec{r}) = \int\int_{-\infty}^{+\infty} S(f,\vec{k})e^{-i2\pi(ft-\vec{k}\cdot\vec{r})}\, df\, d\vec{k}$$

where $s(t,\vec{r})$ and $S(f,\vec{k})$ form a multidimensional Fourier transform pair. This provides an additional incentive for studying planewaves.

To determine the waveshape of a single wave of known vector velocity, in a noiseless environment, one simply observes the wave at any point in space and applies the appropriate delay (or advance) to produce s(t). If a single planewave is spatially sampled using a discrete array of sensors and each sensor adds white noise, an estimate of the wave is obtained by aligning the signals on each of the channels and summing the results. This is the simplest form of array processing. If, in addition to the channel noise, noise exists in the form of an interfering planewave, more complicated processing must be employed. This is a common problem encountered in radar. Allen [2] provides a particularly readable introduction to this subject and lists some of the many references. This estimation problem is usually solved by adding amplitude weighting to each channel before the delay and sum operations. Design engineering judgment is usually applied to obtain acceptable estimates under a variety of conditions.

To obtain the optimum estimates, it is necessary to apply the concepts of decision and estimation theory. For most cases of practical interest, the array processor then requires a different linear filter applied to each channel before the delay and sum operation. The filters generally depend upon the array geometry and noise statistics. The resulting estimates are referred to as maximum-likelihood estimates. This problem was first solved by Kelly and Levin [1]. A closed form solution was not obtained to the general problem of estimating the waveshape and vector velocity. An integral equation must be solved by iteration. The parameter being varied is the vector velocity. If the vector velocity is known, a closed form solution is possible. Capon [3] has successfully implemented this solution and has applied the results to seismological data. In a later paper Capon [4] also discusses the wave detection problem.

The next step is to consider the multiwave maximum-likelihood estimation problem. For the special case where the number of signal and noise planewaves is known, the solution is given by Schweppe [5]. Again, a closed form solution is not obtained. This time it is necessary to vary all of the vector velocities simultaneously in order to solve an integral equation by iteration. When compared to the one wave problem, this is referred to as a multidimensional search. If the vector velocities are known, closed form solutions are possible. An implementation is given by Kobayashi and Welch [6] for two planewaves of known vector velocities. The multiwave detection problem, i.e., determining the number of waves, has not received much attention in the literature.

The general problem considered in this dissertation is to determine the number, vector velocity, and waveshape of overlapping planewaves in the presence of additive noise. A general optimum solution is not found. Instead, a heuristic solution is presented along with a complete working implementation scheme for large scale computers. For the case where the number of waves and the vector velocities are known, the solution is optimum.

The detection of waves and the estimation of the vector velocities is accomplished heuristically by using the frequency-wavenumber power spectral density function. A formal solution of this part of the problem would require maximum-likelihood detection and an exhaustive search algorithm. The general subject of frequency-wavenumber power spectral density estimation is covered in Chapter II.

To reduce the multiwave problem to a succession of single wave estimation problems, the frequency-wavenumber filters are employed, which pass only a single planewave. This avoids the difficult multi-dimensional search required by Schweppe's solution. The general subject of frequency-wavenumber filtering is covered in Chapter III.

The last step is to form a series of single wave maximum-likelihood estimates. This is covered in Chapter IV along with a discussion of the important properties of the maximum-likelihood estimators.

The fifth chapter briefly outlines some of the considerations to be made in applying the estimation techniques to electroencephalography. A limited number of actual data are analyzed and the effectiveness of spectral estimation and filtering is demonstrated.

# CHAPTER II

# MULTIDIMENSIONAL POWER SPECTRAL DENSITY ESTIMATION

## Introduction

The multiwave maximum-likelihood estimator presented in this dissertation requires a planewave filter which effectively reduces the multiwave estimation problem to a single wave estimation problem. The planewave filter is realized as a digital filter which is specified in the frequency-wavenumber domain. The problems encountered for this type of filter are very similar to those encountered in performing multidimensional power spectral density estimates. It is thus convenient to study power spectral estimates before addressing the filtering problem since the concept of multidimensional space is more easily explained in terms of power spectra. The primary reason for developing a method for estimating multidimensional power spectra is to enable one to quickly search the frequency-wavenumber space for the presence of planewaves. This then reduces the number of times it is necessary to perform maximum-likelihood estimates for single waves.

The remainder of this chapter is divided into two parts: one-dimensional power spectral estimation and multidimensional power spectral estimation. The first section uses existing techniques for forming acceptable estimates. Procedures which are extended to the multidimensional case are particularly emphasized. The multidimensional

estimates are first extended to two dimensions and then to three dimensions. The actual computer implementation is carried out in three dimensions. An evaluation of its performance is also made. An extension to n dimensions is an obvious extension of the work contained in this chapter.

## One-dimensional Spectra

In this section common methods of estimating one-dimensional power spectral density (PSD) functions are discussed. The important estimation error considerations are also reviewed. One-dimensional estimates are used to form the cross-power spectral matrices required for the maximum-likelihood estimates of Chapter IV. The material gives the necessary background for the multidimensional case which follows immediately.

The Fourier transform pair for continuous signals is defined in the usual way

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft}\,dt \tag{2-1}$$

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{+i2\pi ft}\,df \tag{2-2}$$

for $-\infty < f,t < +\infty$ and $i=\sqrt{-1}$. The units of t are usually understood to be time (sec) and f to be frequency ($sec^{-1}$) or Hz. It will always be assumed that these integrals exist. The functions x(t) and X(f) may be either deterministic or random. The random processes which are to be transformed are assumed to be ergodic and, thus, at least wide-sense stationary.

The power spectral density function can be defined by

$$P(f) \triangleq \mathcal{F}\{R_x(\tau)\} \equiv \int_{-\infty}^{+\infty} R_x(\tau) e^{-i2\pi f\tau}\, d\tau \qquad (2\text{-}3)$$

where the autocorrelation $R_x(\tau)$ is given by

$$R_x(\tau) = E[x(t)\, x(t+\tau)] \qquad (2\text{-}4)$$

and E denotes expectation or ensemble average. Power spectral estimates which first require an estimate of the autocorrelation function and then an estimate of its transform, as indicated by (2-3), are called indirect estimates. These estimates are not considered here.

The power spectral density function can be equivalently defined as [7]

$$P(f) \triangleq \lim_{T\to\infty} \frac{E|X_T(f)|^2}{T} \qquad (2\text{-}5)$$

where $|\cdot|$ denotes absolute value and $X_T(f)$ represents a truncated transform

$$X_T(f) \triangleq \int_{-T/2}^{+T/2} x(t) e^{-i2\pi ft}\, dt \qquad (2\text{-}6)$$

If x(t) and hence X(f) are deterministic, then the operator E may be omitted and, if the time function is zero for $|t| > T$, (2-5) becomes

$$P(f) = 1/T\ |X_T(f)|^2 \qquad (2\text{-}7)$$

It is important to note that (2-7) is identical to (2-5) if the power spectral density of a truncated deterministic function is desired. If, on the other hand, an estimate of the true power spectral density as given in (2-5) is desired, (2-7) can be used as an estimator. The resultant estimate will depend on T due to Gibb's phenomenon and, as will be explained subsequently, techniques are available which modify (2-7) to reduce the effects of the record length T. Turning now to the case where x(t) and X(f) are sample functions of random processes, the same problem exists in truncating the data, but, in addition, it is now necessary to approximate the ensemble average over the infinite set $\{x_T(t)\}$ of sample functions.

How these problems are solved to form appropriate estimates of the power spectral density will now be discussed. Any estimate formed from the magnitude-squared of the transform, as opposed to the transform of the autocorrelation function, is referred to as a direct estimate.

If x(t) is a random variable, any estimate of its power spectral density will also be a random variable. With the estimate of the true power spectral density, P(f), denoted as $\hat{P}(f)$, the fractional error or normalized mean square error is given as [8]

$$\epsilon = E\{[\hat{P} - E[\hat{P}]]^2\}/E^2[\hat{P}] \quad . \tag{2-8}$$

For the case of White noise (P(f)=constant) $\epsilon=\sqrt{2}$, which says that the fractional error is greater than unity. In [9] the complete calculation of $\epsilon$ is carried through for any function with a Gaussian distribution. The results show that $\hat{P}(0)$ is the worst estimate with

respect to f. For values of f not too near zero, the fractional error $\epsilon$ is approximately unity regardless of the length of T. This result should not be too surprising for, in effect, the average required by (2-5) is approximated by using one sample function. Estimates formed in this manner are frequently referred to as raw estimates.

The next logical step is to examine the effects of using more sample functions to approximate the ensemble average. For comparison purposes the total record length T is maintained, but the record is divided into M contiguous segments. This operation is valid because we have assumed ergodicity and independence of (uncorrelated) segments provided the segments are sufficiently long. The raw power spectral density estimate is formed for each segment and the M results are averaged to form an estimate of P(f). In general one would expect $\epsilon$ to decrease with increasing M for a given frequency. For the Gaussian case the approximation derived in [10] is

$$\epsilon^2 = 1/M \quad . \tag{2-9}$$

For the non-Gaussian variates, similar behavior seems probable and this approximation will be accepted here. If other distributions are encountered, it should be possible to derive appropriate error functions.

The final step in considering power spectral density estimates of continuous functions is to realize that segmenting a given record of fixed length, while increasing the stability of the estimate, actually decreases the frequency resolution of the resultant power spectral

density estimate. As is shown in [9] , the averaging method described above is exactly analogous to premultiplying the data record of length T by a triangular window, Figure 2.1(a), which in turn is exactly equivalent to convolving (smoothing) the raw estimate by the Fourier transform of the window function, Figure 2.1(b). For comparison and later reference, a rectangular window and its transform are shown in Figure 2.2.

The question naturally arises at this point regarding the suitability of using alternate functions for increasing statistical stability. Numerous alternatives have been suggested in the literature [11,12,13]. One alternative (used in Chapter IV) is called the Hanning window which, in the sampled data case, results in smoothing with weights ($\frac{1}{4},\frac{1}{2},\frac{1}{4}$). The basic objective of any window is to effect a compromise between improving statistical reliability and maintaining acceptable resolution. To assist in explaining the concepts of error control and resolution, the power spectral density of an ideal sinusoid (extending to $\pm\infty$) will be estimated. There is no requirement for averaging since this function is deterministic, but the effect of truncation must be considered. The ideal sinusoid and its transform are shown in Figure 2.3. To limit the time function to T seconds, the ideal sinusoid is multiplied by the rectangular window of Figure 2.2. The spectrum of the result, shown in Figure 2.4(b), is the convolution of the delta functions of Figure 2.3(b) with the transform of the rectangle, Figure 2.2(b). The power density is then the square of the function of Figure 2.4(b) divided by the record length T. From this figure it should be clear that estimates at very low frequencies tend

to be poorer than at higher frequencies, for fixed T, because of overlapping sidelobes. This phenomenon is easily controlled by insuring that the record contains a sufficient number of cycles of the lowest frequency terms so that the tails of the spectral distributions are nearly zero at the origin. The second observation is that if a second time function were added to the first, such that the main lobes of the spectra of the individual signals overlapped, it would be practically impossible to resolve two waves from the spectrum of the sum. This is, of course, the resolution problem. The errors in spectral analysis arise from the presence of the sidelobes of the main lobes. They can be misinterpreted as low amplitude signals, or mask the presence of low energy sinusoids, in the spectrum. This is especially evident when the effects are shown on a log scale as shown in Figures 2.1(c) and 2.2(c). Ideally, one would alter the raw estimate to decrease the sidelobes and simultaneously increase the sharpness of the main lobe. Unfortunately, these demands tend to be conflicting. As a possible compromise, consider the use of the triangular truncating window of Figure 2.1 instead of the rectangular window. The new estimate is simply given as the convolution of Figure 2.1(b) with the delta functions of Figure 2.3(b). The results are given in Figure 2.5(b). Comparing the estimate shown in Figure 2.5(b) with the former reveals that the "leakage" of energy from the main lobe into the sidelobes has been significantly diminished, but the frequency resolution is only about half the former resolution due to a wider and smaller main lobe. The optimum choice is a subject of current research [14].

$$e = \sqrt{1/T\Delta f} \qquad (2\text{-}10)$$

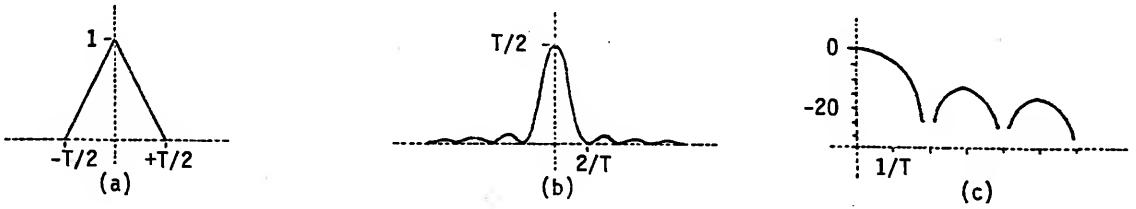



Figure 2.1 Fourier transform pair for a triangular window. (a) time function, (b) frequency function, and (c) positive frequency.



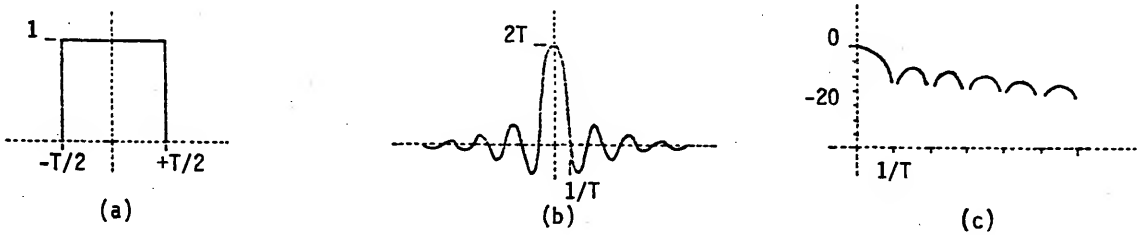



Figure 2.2 Fourier transform pair for a rectangular window. (a) time function, (b) frequency function, and (c) log (decibels) of absolute value of (b) for positive frequency.

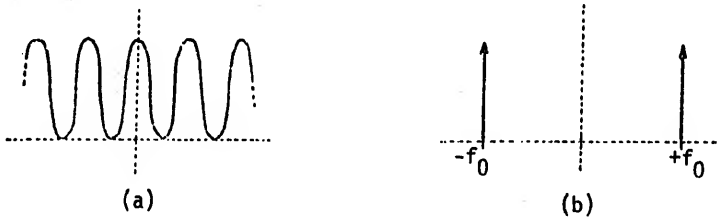



Figure 2.3 Fourier transform pair for ideal sinusoid. (a) time function and (b) frequency function.

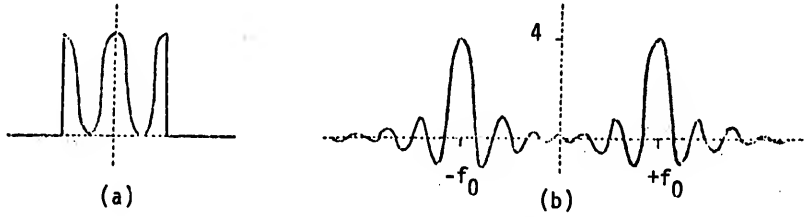



Figure 2.4 Fourier transform pair for rectangular-truncated sinusoid. (a) time function and (b) frequency function.

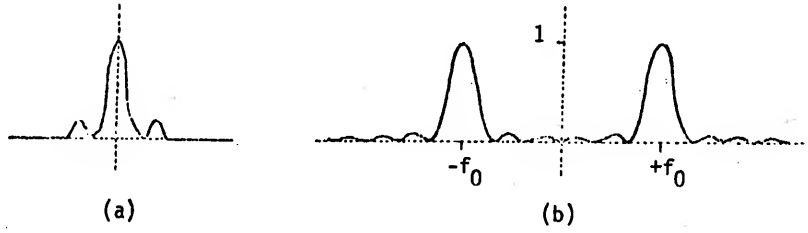



Figure 2.5 Fourier transform pair for triangle-truncated sinusoid. (a) time function and (b) frequency function.

where $\Delta f$ is the frequency resolution given as the effective width of the principal lobe of the frequency domain smoothing function. This is the equation relating resolution, record length, and error most generally used and is frequently referred to as the uncertainty relation [10].

If we now consider the basic time function to be either the sum of an unknown number of sinusoids or a random variable, it is not difficult to visualize why the triangular window is generally regarded as yielding better estimates of the power density than the rectangular window. In general, any smooth truncating function will tend to decrease errors due to sidelobes at the expense of diminished resolution. It is important to realize, however, that raw estimates due to the rectangular window are acceptable if the true spectrum is characterized by large, well-separated peaks. In this regard, it is of interest to note that it is possible to accentuate the main lobe by actually accentuating the discontinuity (e.g., by differentiating the rectangular window); however, if this is done, sidelobes can become as great as the main lobe and the resulting ambiguity seriously limits the usefulness of such estimates. This is the same problem faced in the design of weighted delay and sum point array antennas where the resolution is in bearing angle rather than in frequency [2,15].

The results presented thus far have been justified for continuous functions only. As this dissertation is concerned exclusively with sampled data, we shall now discuss the effects of sampling. The transition is made quite easily because it is well known that sampled data offer a complete representation of a function if the function is

sampled at at least twice the sampling frequency (i.e., at the Nyquist rate) and acceptable spectral estimates are based upon forming acceptable approximations of Fourier transforms by using truncated data. To extend the results to form estimates of power spectral density functions by using sampled data, the following steps are explained. The data are assumed to be sampled at at least the Nyquist rate yielding an N-point time series of length T seconds and time interval $\Delta T$. The truncated Fourier transform has now become the discrete Fourier transform (DFT) defined as

$$X(k\Delta F) = 1/N \sum_{n=0}^{N-1} x(\ell\Delta T)e^{-i2\pi k\ell/N} \quad k=0,1, \ldots, N-1 \qquad (2\text{-}11)$$

where the frequency interval is

$$\Delta f = 1/T = 1/N\Delta T$$

and conversely

$$\Delta T = 1/N\Delta F = 1/F$$

where F is the sampling rate. It should now be noted that the discrete frequency domain function is also sampled. Mathematically this is equivalent to performing a periodic extension of the original truncated time series so that the new series is periodic with period N. The DFT calculates only the first period. Having computed the transform, a discrete convolution operation may be employed to reduce sidelobes.

The frequency series is thus squared and scaled to obtain the discrete power spectral estimate.

Having shown that sampled data representation loses none of the necessary information, it remains only to explain the "rectangular integration" scheme implied by the DFT. It is recognized that more sophisticated integration (e.g., trapezoidal or Simpson's Rule) could be employed to reduce the integration estimation error, but it must be remembered that these techniques are significantly superior only when the integrand has relatively sharp peaks and valleys (i.e., contain high frequency terms). The minimum acceptable sampling, however, insures that the sampling interval will increase as the frequency components increase. Increasing the sampling, in turn, increases the accuracy of the estimated integral. We thus have some assurance that integration estimation errors are not too sensitive to ill-conditioned data. We make the usual assumptions that these errors are negligible in comparison with the truncation and averaging errors mentioned earlier.

The principal reason for using the DFT in approximating a truncated Fourier transform stems from the fact that it is possible to utilize the fast Fourier transform (FFT) algorithm to perform the computation. This numerical technique was first made popular by [16], is discussed thoroughly in [17], and a particularly readable account is given in [18]. If (2-7) is computed directly, it is seen that $N^2$ operations are required (multiply/add). The successive doubling algorithm [17] upon which the FFT is based permits one to take two Fourier analyses of N/2 points each and combine them in N/2 operations to obtain an analysis

of an N-point record of the same data. Successive application of this technique yields an N-point DFT in approximately N $\log_2$ N operations.

A comparison of the execution times to perform a DFT is shown in Figure 2.6 where it is assumed that execution time is directly proportional to the number of operations. Graphs for actual computer execution time can be found in [9]. This great difference in execution speed is widely recognized as being responsible for making discrete Fourier analysis feasible and far outweighs any disadvantage incurred in not employing other numerical integration techniques. The increased speed is also responsible for the present trend of forming direct, rather than indirect, estimates.

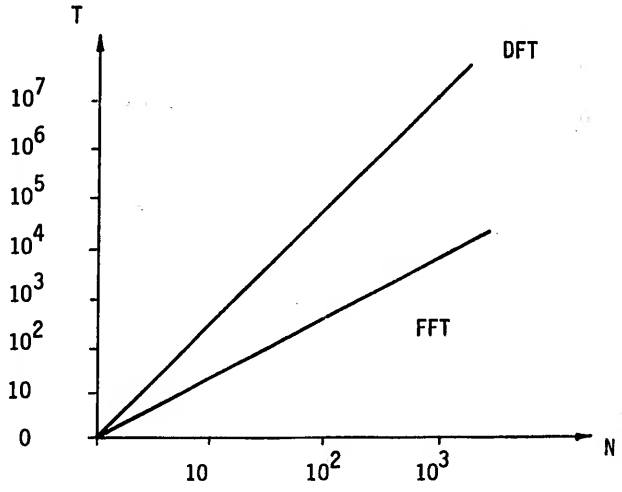



Figure 2.6 Relative execution times.

This section has described various methods of estimating the one-dimensional power spectral density function. The results of actual computer implementation are contained in Chapter IV. Details of the calculations can be found in [19]. The remainder of this chapter is concerned with extending the concepts to the multidimensional case and, in so doing, presents a new method of estimating frequency-wavenumber spectra.

## Frequency-wavenumber Power Spectral Density Estimation

The function $s(t,\vec{r})$ is a general spatio-temporal function of a single variable time t and the one-, two- or three-dimensional spatial vector $\vec{r} = (r_x, r_y, r_z)$. It gives the value of s at time t and vector position $\vec{r}$ with respect to the spatio-temporal origin. This function and its related functions are special cases of general multidimensional functions. The distinction between multidimensional and spatio-temporal functions only becomes necessary when $s(t,\vec{r})$ is a planewave; otherwise, $s(t,\vec{r})$ may be viewed as a function of an arbitrary number of variables.

The multidimensional autocorrelation function $R(\tau,\vec{p})$ of $s(t,\vec{r})$ is given as

$$R(\tau,\vec{p}) = E[s(t,\vec{r})s(t+\tau,\vec{r}+\vec{p})] \qquad (2\text{-}12)$$

where E denotes the average value, or expectation. Spatial and temporal stationarity and ergodicity are assumed.

The multidimensional power spectral density function $P(f,\vec{k})$ can be defined as

$$P(f,\vec{k}) \triangleq \mathcal{F}\{R(\tau,\vec{r})\} \equiv \int\int_{-\infty}^{+\infty} R(\tau,\vec{r})e^{-i2\pi(f\tau+\vec{k}\cdot\vec{r})}\, d\tau\, d\vec{r} \qquad (2\text{-}13)$$

where the multidimensional autocorrelation function is given by

$$R(\tau,\vec{r}) \triangleq E[s(t,\vec{r}_0)s(t+\tau,\vec{r}_0+\vec{r})] \quad . \qquad (2\text{-}14)$$

For physical systems the spatial vector $\vec{r}$ is one-, two- or three-dimensional, $\vec{r} = (r_x,r_y,r_z)$. The meaning of the wavenumber vector $\vec{k}$ will become clear subsequently.

The spatial Fourier transform of $P(f,\vec{k})$ yields

$$\int P(f,\vec{k})e^{-i2\pi(\vec{k}\cdot\vec{r})}\, d\vec{k} = P_x(f,\vec{r}) \qquad (2\text{-}15)$$

and its inverse is

$$\int P_x(f,\vec{r})e^{+i2\pi(\vec{k}\cdot\vec{r})}\, d\vec{r} = P(f,\vec{k}) \quad . \qquad (2\text{-}16)$$

For spatially discrete two-dimensional systems, $\vec{r}$ may be interpreted as the separation between the m-th and n-th sensors, $\vec{r} = (x_m-x_n,y_m-y_n)$, and (2-15) becomes

$$P(f,\vec{k}) = \sum_{m,n=1}^{K} P_x(f,x_m-x_n,y_m-y_n)e^{+i2\pi(\vec{k}\cdot\vec{r})} \qquad (2\text{-}17)$$

where f is the frequency, or, more formally, the temporal frequency, and $\vec{k}$ is the "spatial frequency" or the vector wavenumber. f has units of time$^{-1}$ and $\vec{k}$, length$^{-1}$. The interpretation of $\vec{k}$ is made subsequently.

The inverse space Fourier transform of $P(f,\vec{k})$ yields the cross-power spectrum, $P_x(f,\vec{p})$, between two points in space separated by $\vec{p}$.

$$P_x(f,\vec{p}) = \int_{-\infty}^{+\infty} P(f,\vec{k}) e^{+i2\pi(\vec{k}\cdot\vec{p})}\, d\vec{k} \quad . \tag{2-18}$$

The complementary transform may be used to define $P(f,\vec{k})$

$$P(f,\vec{k}) = \int_{-\infty}^{+\infty} P_x(f,\vec{p}) e^{-i2\pi(\vec{k}\cdot\vec{p})}\, d\vec{p} \quad . \tag{2-19}$$

The inverse temporal Fourier transform of $P_x(f,\vec{p})$ yields the multidimensional autocorrelation function

$$R(\tau,\vec{p}) = \int_{-\infty}^{+\infty} P_x(f,\vec{p}) e^{+i2\pi f\tau}\, df \tag{2-20}$$

which, when expressed in this manner, demonstrates that $R(\tau,\vec{p})$ may also be interpreted as cross-correlation, or covariance, between points spatially separated by $\vec{p}$.

The multidimensional spectrum may be defined directly in a manner similar to (2-5)

$$P(f,\vec{k}) \triangleq \lim_{T_t,T_r\to\infty} \left[ E|S_T(f,k)|^2/T_tT_r2^n \right] \tag{2-21}$$

where $S_T$ is the truncated n-dimensional Fourier transform of $s(t,\vec{r})$. In order to develop an appreciation of the meaning of the frequency-wavenumber space, and to introduce the spectral estimation problem, the two-dimensional case is considered before presenting the theory and implementation for three dimensions.

## Two-dimensional Spectra

Consider a spatio-temporal function of two independent variables, time and space. Further, let it be required that this function $s(t,x)$ represents an ideal traveling planewave, which propagates in the positive x direction. The wave is ideal in the sense that it is not attenuated or filtered or otherwise contaminated as it travels. The functional form is the same at each point in space except for the delay due to a finite propagation velocity. Thus if $s(t)$ is the functional form at the origin, the value at any arbitrary x is given by $s(t-x/v)$ where v is the velocity of propagation of the wave in the positive x direction. For convenience the propagation is usually expressed in terms of inverse velocity

$$\alpha = \vec{v}/|v|^2 \quad . \tag{2-22}$$

Note that $\alpha$ is oriented in the direction of propagation of the wave and represents the vector of delays incurred per unit distance as measured along each coordinate axis. It is sometimes referred to as the slowness vector. If there is a sinusoidal component of the signal of frequency f , then its vector wavenumber is given by $k \triangleq f\alpha$. The use of vectors instead of scalers is an obvious extension to the general three-dimensional space, i.e.,

$$\vec{v} \triangleq (v_x, v_y, v_z)$$
$$\vec{\alpha} \triangleq (\alpha_x, \alpha_y, \alpha_z)$$

and

$$\vec{k} \triangleq (k_x, k_y, k_z) \quad . \tag{2-23}$$

If $s(t,x)$ is a monochromatic planewave with inverse velocity $\alpha_0$ and frequency $f_0$

$$s(t,x) = A \cos [2\pi f_0(t-\alpha_0 x) + \theta] = A \cos [2\pi(f_0 t - k_0 x) + \theta] \quad (2\text{-}24)$$

where $\theta$ is uniformly distributed on $[0,2\pi]$. The autocorrelation of $s(t,x)$ is

$$R(\tau,p_x) = E[s(t,x)s(t+\tau,x+p_x)]$$

$$= A^2/2 \cos (2\pi f_0 \tau + k_0 x) \quad . \quad (2\text{-}25)$$

From [20] the frequency-wavenumber power spectral density function is the two-dimensional transform of the autocorrelation

$$P(f,k_x) = \int_{-\infty}^{+\infty}\int A^2/2 \cos (2\pi f_0 \tau + k_{x0} x)\, dt\, dx \quad (2\text{-}26)$$

$$P(f,k_x) = 1/2\pi^2 \, \delta(f-f_0, k_x - k_{x0}) \quad . \quad (2\text{-}27)$$

The spatio-temporal function is illustrated in Figure 2.7 and its frequency-wavenumber power spectral density is illustrated in Figure 2.8. Obviously the frequency-wavenumber power spectrum concisely summarizes the defining parameters of the given planewave. It is important to realize that if $s(t,x)$ were wideband, instead of monochromatic, the energy distribution in the frequency-wavenumber domain would be contained in a plane as suggested by Figure 2.9. It is also

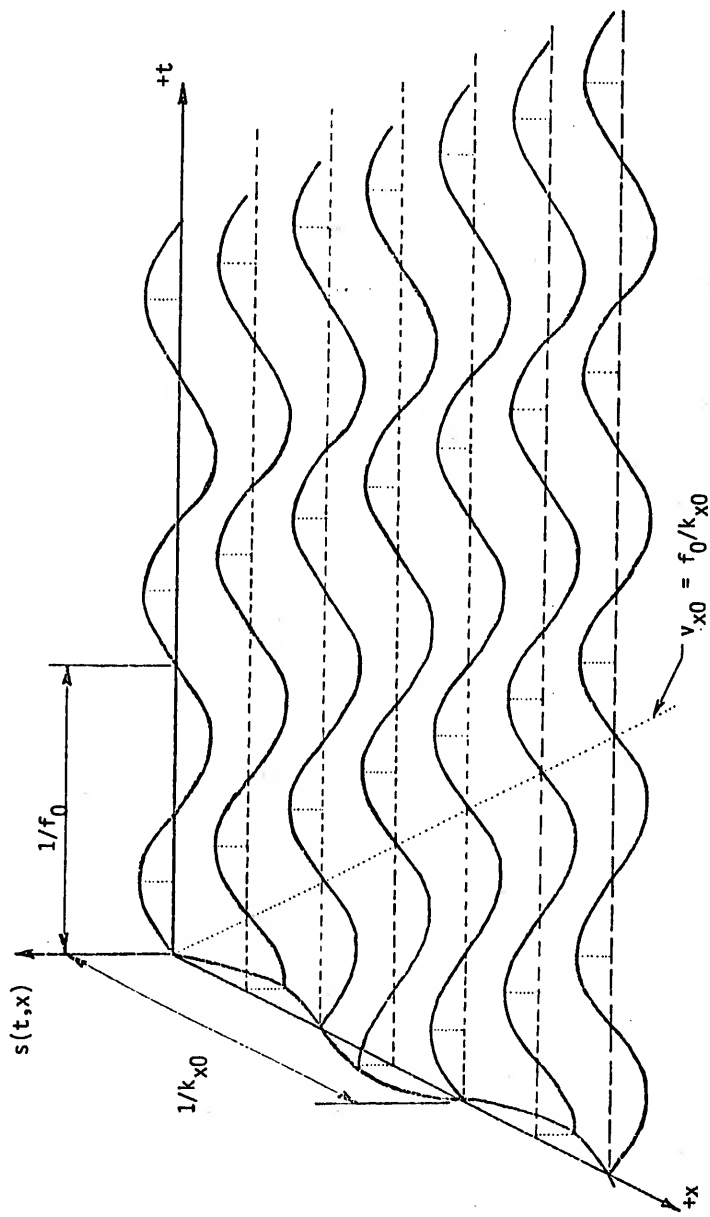



Figure 2.7 Two-dimensional monochromatic spatio-temporal function with frequency $f_0$, wavenumber $k_{x0}$, and velocity $v_{x0}$.

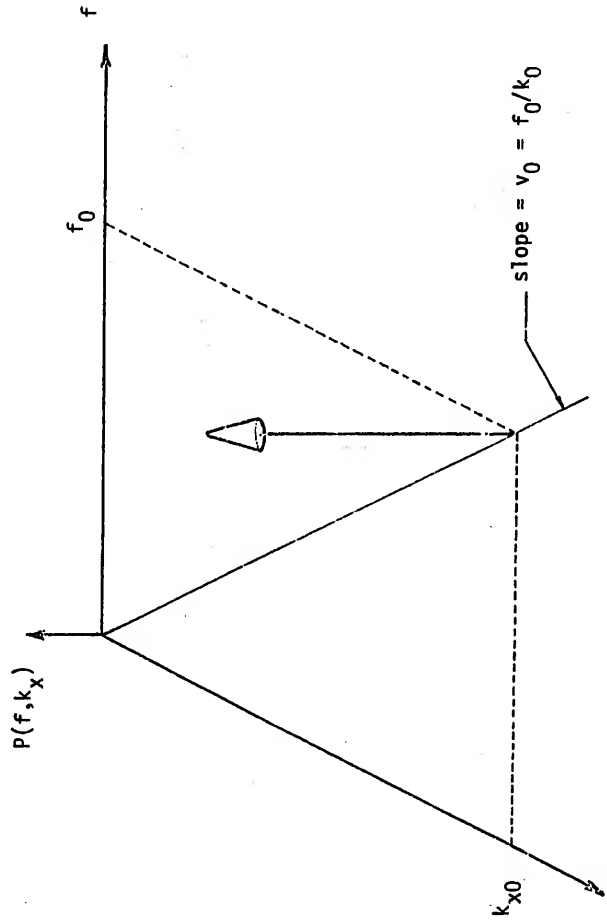



Figure 2.8 Frequency-wavenumber spectrum of two-dimensional monochromatic spatio-temporal function with frequency $f_0$, wavenumber $k_{x0}$, and velocity $v_{x0}$.

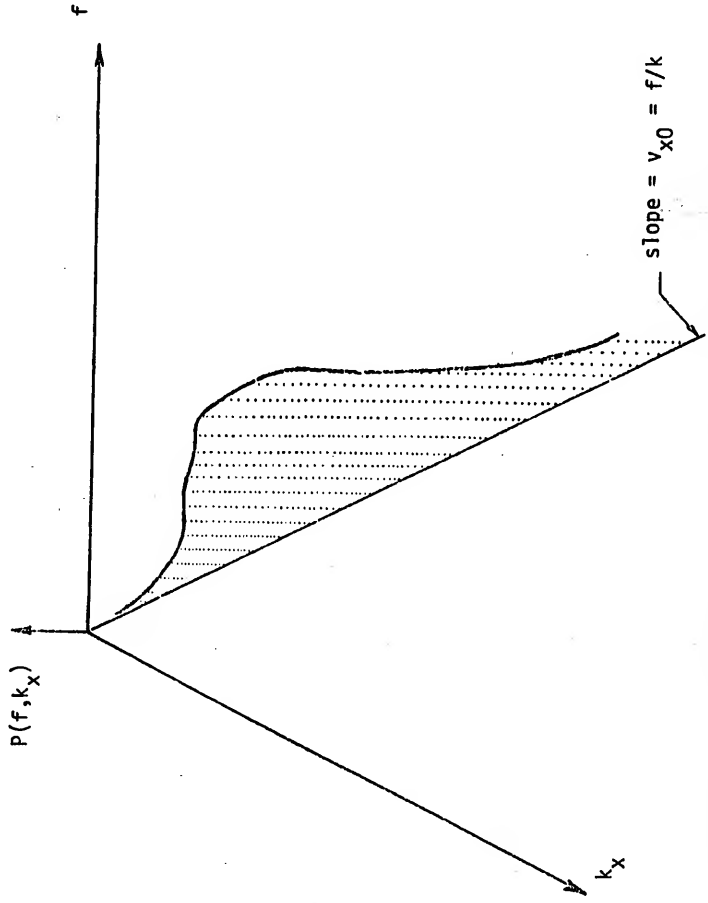



Figure 2.9 Frequency-wavenumber spectrum for two-dimensional wideband spatio-temporal function with velocity $v_{x0}$.

of interest to note that, if we take the limiting operation, $\lim_{v\to\infty} k_0=0$ and the familiar one-dimensional PSD is obtained. In practical terms this condition implies that the velocity is so large that the wave appears identical at every point in space and that the power spectrum analysis reduces to an analysis of a function of one variable, time.

If s(t,x) were a sum of wideband planewaves, each with a unique velocity, each planewave would appear in the spectrum as a power density function centered on the ray defined by the velocity of the wave as shown in Figure 2.9. In general the frequency-wavenumber power spectral density function accounts for the distribution of energy of an arbitrary spatio-temporal function in terms of frequency and velocity (or wavenumber).

In considering the spectral estimation problem we again turn to the direct definition of the frequency-wavenumber power spectral density function

$$P(f,k_x) = \lim_{T_t,T_r\to\infty} [E|S_T(f,k_x)|^2/4T_tT_r] \qquad (2\text{-}28)$$

where

$$S(f,k_x) = \int_{-T_t/2}^{+T_t/2}\int_{-T_r/2}^{+T_r/2} s(t,x)e^{-i2\pi(ft+k_xx)}\,dt\,dx \quad . \qquad (2\text{-}29)$$

The problems encountered in forming acceptable estimates for the two-dimensional case follow directly from the one-dimensional discussion. For deterministic data the E operator may be ignored; however, the effects of truncation are present in the form of

two-dimensional sidelobes, surrounding the main lobe. If the truncation window has circular symmetry, or otherwise has simple geometry, the two-dimensional transform study of Papoulis [21] or Goodman [22] may be consulted to determine the trade-offs between main lobe resolution and sidelobe level. These matters are studied in image processing, optical systems, and two-dimensional antenna theory. However, the lack of symmetry in spatio-temporal functions makes it difficult to apply these theoretical results to frequency-wavenumber power estimation. If $s(t,x)$ is a sample function of a random process, one will usually want to know the relationship between the number of spatio-temporal samples needed to approximate the ensemble average, the window one uses for truncation and, of course, the statistical error function. While these matters are quite important in determining the value of a general power density estimator, they are not investigated here for two reasons. First the power spectral density estimator required for this study is needed only to assist in the detection and estimation of strong signals. If spatio-temporal fields are encountered which have fairly flat frequency-wavenumber distribution, the power spectral density estimator simply is not used. The second reason for not surveying the statistical advantage of averaging and/or windowing stems from the fact that frequent applications are characterized by arrays of very limited spatial population. The array considered in Chapter V consists of a 4 x 4 array. Thus if an attempt is made to smooth using existing techniques, the reduced resolution renders the estimate virtually useless.

## Three-dimensional Spectra

The spectrum analysis of spatio-temporal functions of one spatial variable and time is conceptually easily extended to the general case of three spatial variables plus time. In the latter case the wavenumber k becomes the vector wavenumber $\vec{k}$ which has three coordinate components, i.e.,

$$PSD\{X(t,\vec{r})\} = P(f,\vec{k}) \tag{2-30}$$

where

$$\vec{r} = \{r_x, r_y, r_z\}$$

and

$$\vec{k} = \{k_x, k_y, k_z\} \quad . \tag{2-31}$$

From a practical viewpoint one spatial coordinate is too restrictive to be of much use in modeling physical systems and at the other extreme, with three spatial coordinates, an element in the frequency-wavenumber space is an element in a five-dimensional space whose coordinates are power density, frequency, $k_x$, $k_y$, and $k_z$. The data processing task for handling data of this magnitude is considerable as is the task of interpreting the results. As a compromise it is common in sonar and seismology to use two spatial coordinates. This compromise is also used in Chapter V. An examination of this estimation problem follows.

Consider a function similar to (2-24) but extended to two spatial coordinates

$$s(t,x,y) = A \cos (2\pi f_0(t-\alpha x_0 x + \alpha y_0 y) + \theta) \tag{2-32}$$

or using vector notation

$$s(t,\vec{r}) = A \cos\left(2\pi f_0(t-\vec{\alpha}_0\cdot\vec{r}) + \theta\right)$$

$$s(t,\vec{r}) = A \cos\left(2\pi f t - \vec{k}_0\cdot\vec{r} + \theta\right) \qquad (2\text{-}33)$$

where $\vec{k}_0 = f\vec{\alpha}$. The frequency-wavenumber spectrum of this becomes a delta function

$$K\,\delta(f-f_0,\vec{k}-\vec{k}_0) \quad .$$

The vector velocity and inverse phase velocity are related by

$$\vec{\alpha} = \vec{v}/|\vec{v}|^2$$

and

$$\vec{k} = f\vec{\alpha} \quad .$$

The term phase is added to emphasize the fact that we are dealing with a projection of the actual (group) velocity onto the x-y plane.

In attempting to sketch the frequency-wavenumber spectrum, we will use points to indicate three-dimensional delta functions, thus the example cited may appear as in Figure 2.10. Note that the direction of propagation of the wave is the same as $\vec{k}_0$ if the $k_x$ axis is relabled x and the $k_y$ axis is relabled y. The magnitude velocity of the wave is given by the slope of the line from the origin to the dot, i.e.,

$$|v_0| = 1/|\alpha_0| = f_0/k_0 \quad . \qquad (2\text{-}34)$$

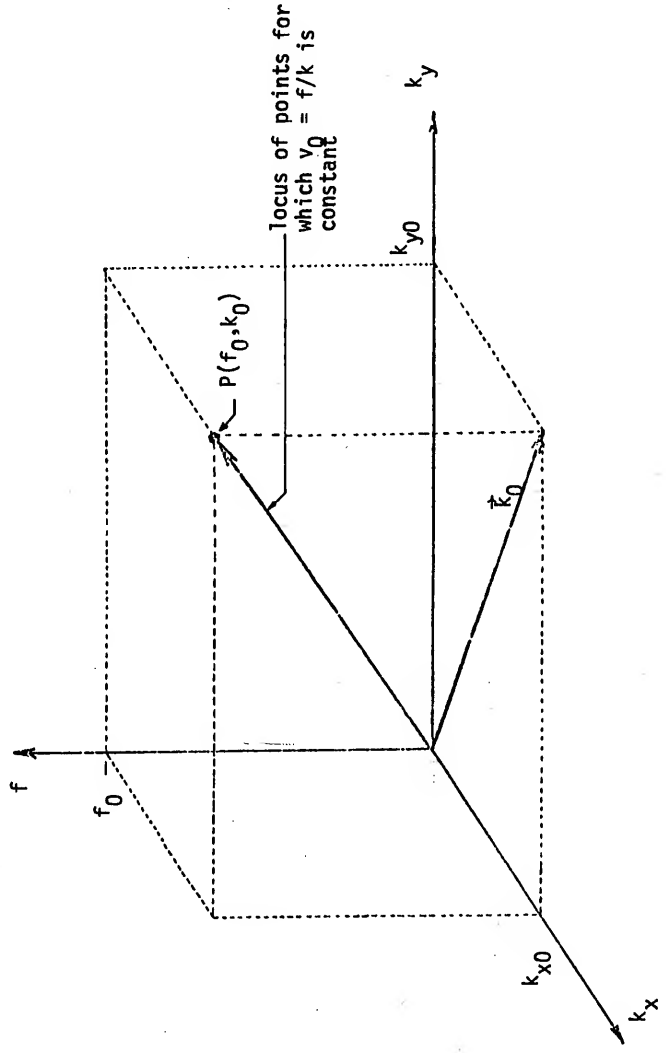



Figure 2.10 Frequency-wavenumber spectrum of three-dimensional monochromatic spatio-temporal function with frequency $f_0$, wavenumber $\vec{k}_0$, and velocity $v_0$.

Obviously then the power spectrum characterizes the vector velocity, frequency, and amplitude of the monochromatic wave. If wideband waves are considered, the points must spread along the line passing from the origin to the dot.

To enhance our knowledge of this case we compare our technique to that used by others. As noted in [23] beam-forming and one-dimensional spectral analysis can be combined to accomplish a study of the velocity and frequency structure of the wave without any direct suggestion that estimates of frequency-wavenumber spectra are being obtained. This is referred to as the time domain method and is illustrated in Figure 2.11. The spatio-temporal function is sampled as indicated by the inputs $s_i$. Each channel is delayed to align any incoming wave with a specified vector velocity. The channels are then summed to diminish the effects of noise. The resulting function is referred to as a beam and is then spectral analyzed in a convenient manner and the end result is power as a function of $(f_0,\vec{v}_0)$. While the method is conceptually simple, it is subject to severe limitations in velocity filtering or weighted delay and sum operations. Generally, the weighting is adjusted to suppress interfering events when the frequency composition and bearing of the unwanted events are known a priori. With no a priori statistics maximum-likelihood weights can be obtained but, in practice, this offers little advantage over straight delay and sum with unit weights [3].

The next method, referred to [23] as the beam-forming method, frequency domain, is termed the conventional method by Capon [3]. It consists of an extension of (2-18), i.e.,

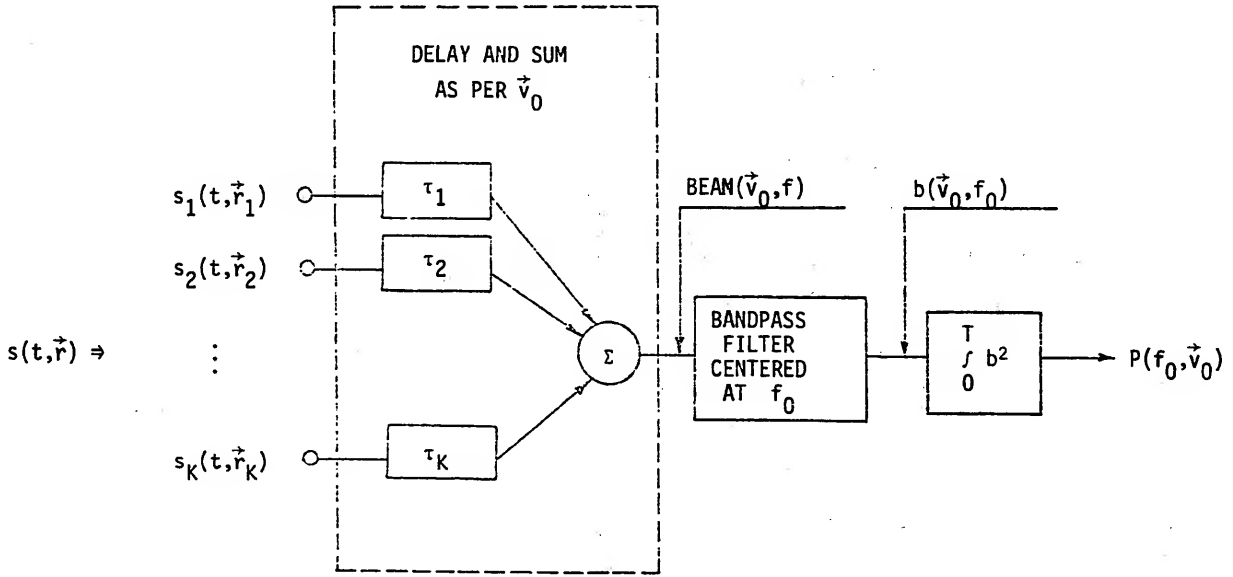



Figure 2.11 Frequency-wavenumber PSD estimation by Time Domain Method.

$$\hat{P}(f,\vec{k}) = \sum_{\vec{p}} W_p \hat{P}_x(f,\vec{p}) e^{-i2\pi \vec{k}_0 \vec{p}} \qquad (2\text{-}35)$$

where $\hat{P}_x$ is an estimate of the cross-power spectral density and $W_p$ is a window used to improve the estimate of the two-dimensional transform of $\hat{P}_x$. The advantage of this technique over the former is that it is not necessary to form a large number of beams. From an implementation viewpoint this estimate is appealing as standard techniques are available for estimating $\hat{P}_x$ and the two-dimensional transform can be estimated using windows which may be symmetrical as the units in both dimensions are identical. The estimation errors associated with this procedure are covered in [3,23].

Whenever estimates involve approximating Fourier transforms, one encounters the problem of imperfect (not ideal) resolution in the transform domain. The usual techniques involve choosing a truncation window in one domain which yields acceptable resolution in the transform domain. Capon [20] developed the "high resolution" estimate which utilizes decision theoretic ideas to form an optimum estimate. By definition the high resolution is

$$P'(f,k) \triangleq \left[ \sum_{j,\ell=1}^{K} \hat{q}_j (f) e^{-i2\, k(x_j - x_\ell)} \right]^{-1} \qquad (2\text{-}36)$$

where $\hat{q}$ is the inverse of the estimated cross-power matrix. The essence of this technique will be covered in Chapter IV along with maximum-likelihood estimation. It provides excellent estimates, in general, but is computationally inefficient.

The estimation presented in this study is based upon estimating directly from a form of the defining equation

$$P(f,\vec{k}) \triangleq \lim_{\vec{T}\to\infty} 1/16T_tT_xT_y\, E\left|\int_{-T}^{+T} s(t,r)e^{i2\pi(ft+\vec{k}\cdot\vec{r})}\, dt\, d\vec{r}\right|^2 \qquad (2\text{-}37)$$

where $\vec{T} = T_tT_xT_y$. As an estimate this becomes

$$\hat{P}(f,\vec{k}) = 1/16\vec{T}\ |DFT\{s(t,\vec{r})\}|^2 \quad . \qquad (2\text{-}38)$$

This method of estimating the frequency-wavenumber spectrum will be referred to as the direct FFT method. This procedure is similar to that advanced by Smart and Flinn [24]. The difference is mainly due to the method of calculation. The reason for choosing this estimate is to utilize the computational efficiency of the multidimensional fast Fourier transform [16,17]. While reference to the multidimensional FFT is contained in the landmark paper of FFT [9], the literature seems devoid of its application to frequency-wavenumber spectrum estimation. This may possibly be explained by the requirement of the FFT to utilize uniform sampling in each coordinate direction. It should be noted, however, that sampling intervals in each direction need not be equal.

It is also interesting to note that early attempts at one-dimensional power estimates were based on the indirect method of first estimating an intermediate function (covariance) and then transforming to obtain an estimate of power density [9]. With the introduction of the FFT, the indirect method was replaced by the direct method which essentially estimates the magnitude-squared of the Fourier transform.

Where the requirement for uniform sampling can be accepted, the use of the direct method of estimating frequency-wavenumber power spectral density via the multidimensional FFT will very probably become increasingly popular.

As mentioned previously, the question of finding suitable three-dimensional windows to improve the raw estimates is not investigated here. The Fortran programs written to implement the estimate (2-38), however, include provisions for multidimensional windows. This point will be covered in the next section. Whether or not special windows are included, the computations required by direct frequency-wavenumber PSD estimation are significantly less than those required by other estimates. The frequency domain method appears to offer the most efficient second choice. In order to compare these two estimates, assume we have $N_1 \times N_2 \times N_3$ elements of a spatio-temporal function to be analyzed, where the ordering is time by x by y. The direct FFT method can be accomplished in about $N_1N_2N_3\log_2N_1N_2N_3$ operations [17]. Assuming the one-dimensional FFT is used, the frequency domain method requires $N_1\log_2N_1$ operations to transform the $N_1$ time functions plus $(N_2N_3)^2N_1$ operations to form the cross-power matrix and finally $N_2^2N_3^2$ operations to perform the final two-dimensional transform. Table 1 compares the number of operations required for the two methods for four particular values of $N_1$, $N_2$, and $N_3$. The execution time for the 32 x 4 x 4 case on an IBM 360/65 computer is about 20 seconds using the direct FFT method.

Table 1

Number of operations

| $N_1,N_2,N_3$ | Frequency domain method | Direct FFT method |
|---|---|---|
| 32,4,4 | 8.6 K | 4.6 K |
| 32,8,8 | 135 K | 22 K |
| 1024,4,4 | 266 K | 65 K |
| 1024,8,8 | 4100 K | 130 K |

## Implementation

Sets of Fortran subroutines have been prepared for the purpose of estimating the three-dimensional frequency-wavenumber power spectral density functions via the direct FFT method. The dimension and format specifications are specifically written to handle input arrays of size 32 x 4 x 4 in the order of time by x direction by y direction. This corresponds to a typical array used in other studies to which this research has been applied. Straightforward programming modification could be incorporated to handle virtually any array size.

The explanation of the purpose of the principal programming units used to analyze simulated data follows. The analysis of actual data is given in Chapter V. Actual programs are included as Appendix D. The flow chart of Figure 2.12 should be noted in the following discussion.

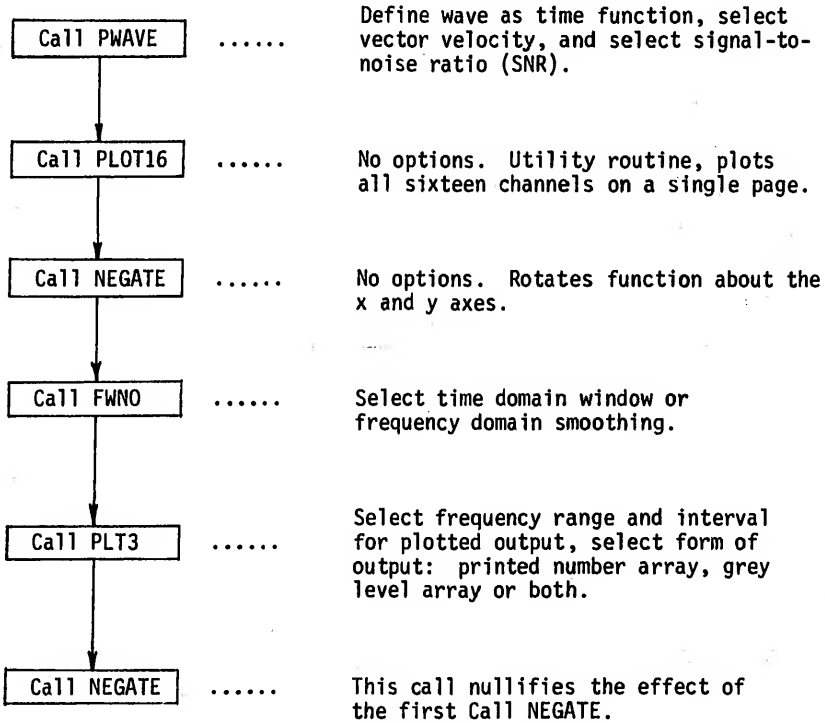



Figure 2.12 Simplified flow chart for simulated data.

Subroutine PWAVE accepts a given function of time which is taken to be the signal at the origin in space. Fifteen additional functions are then generated which correspond to the time functions which would be received by an array of sensors for the specified vector velocity. The array is assumed to be square with 2 cm sensor spacing. If ambient channel noise is desired the signal-to-noise ratio (SNR) is set to $0 \leq SNR \leq 990$. If this condition is set each channel receives additive

independent Gaussian noise [19] with a peak rms noise of SNR. If SNR > 990 no noise is added. If SNR ≤ 0 noise only is returned with unity rms value. PWAVE may be called repeatedly to generate a sum of several planewaves.

If we refer again to Figure 2.12, PLOT16 is next to be called. This routine provides a quick look at all sixteen channels with relatively low amplitude resolution.

Subroutine NEGATE is used to transfer s(t,x,y) to s(t,-x,-y). This is required whenever a standard FFT is used in which the exponential term of the forward transform is given as

$$e^{-i2\pi(ft+k_x x+k_y y)} \quad .$$

In order for the $\vec{k}$ and $\vec{v}$ to have the same direction, the exponential term must be

$$e^{-i2\pi(ft-k_x x-k_y y)}$$

which is effectively accomplished by NEGATE. If NEGATE is omitted the $k_x$ and $k_y$ directions will be reversed in the final output.

Subroutine FWNO estimates the three-dimensional spectrum basically by calculating the magnitude-squared of the FFT of the input spatio-temporal series. To enhance statistical reliability and reduce leakage several options are available. One may specify tapering (windowing) in the spatio-temporal domain or smoothing of the raw spectrum by either Hanning or Hamming [9] smoothing. The frequency

domain smoothing only smooths in the temporal frequency direction and not in the wavenumber coordinates. The tapering utilizes a 10 percent cosine window in the time coordinates and the spatial planes are bordered with 0.5 weighting with the center four elements weighted at 1.0. FWNO may be called repeatedly, with different samples of the same process and the results averaged.

To increase resolution in the wavenumber domain zeros are added in the spatial domain. Each x-y spatial plane is padded with zeros to double the resolution in the transform domain. As adequate resolution is available in the temporal frequency coordinate, zeros are not added to the time coordinate. The next step is to calculate raw power estimates for positive temporal frequencies. IBM's subroutine HARM is used for the actual FFT [25]. As the ordering of the spatial and temporal frequencies from any FFT is positive frequencies followed by negative frequencies in all coordinates, the last step reorders the frequencies to normal ascending order.

If the input series is a spatio-temporal impulse response, the above applies except that the zeros are added to the spatial tails of the impulse response. Instead of bordering the x-y plane with zeros, zeros are inserted at the positive and negative ends of the spatial records. This optional method is selected by calling FWNO5, instead of FWNO. FWNO is actually an entry into FWNO5.

Subroutine FWNO returns a power density array which may be used for any purpose. If a line printer display of the data is desired, subroutine PLT3 can be used. This routine outputs

power as a function of $k_x$ and $k_y$ for each temporal frequency desired. The output is normalized on a negative decibel scale with respect to the maximum of the input power array. At each temporal frequency the output consists of a matrix of numbers (db's) whose position corresponds to the proper position in the $k_x$-$k_y$ space. Contour plots, using grey levels as described in [26,27], can also be added. The contour plot more than doubles the cost of the output, but renders the results much easier to interpret. If normalization is desired with respect to a previous PLT3 call, PLT4 should be called. This is useful for observing the effects of filtering, comparing noise level to signal level, etc. If normalization is desired with respect to a particular point of the input array, PLT5 should be called. This is useful in examining the amplitude-squared response of a three-dimensional filter as it allows one to normalize with respect to a specific point in the passband. Ripples are then seen as positive or negative deviations from 0 db.

If it is required to return the spatio-temporal series to its original state, NEGATE must be called again. NEGATE was not included within FWNO as several calls may be made to FWNO without ever returning to the spatio-temporal domain.

## Results

The usefulness of frequency-wavenumber power spectral estimates is demonstrated by application throughout this study. In this section, an example of the application to the analysis of simulated data is given. A general assessment of

the performance of the estimate is also included.

To illustrate the performance of the Fortran subroutines, seven truncated, monochromatic planewaves were generated and summed using PWAVE. The frequency-wavenumber spectrum of the resulting spatio-temporal function was estimated by FWNO and plotted via PLT3. The input functions are listed in Table 2.

The temporal sampling rate was 50 Hz, which yields a folding frequency of 25 Hz and a time interval of 0.02 sec. The record length was 32 points or 640 msec. Spatial sampling took the form of a 4 x 4 square array with a spatial interval of 2 cm. The "spatial folding frequency" was then 0.25 $cm^{-1}$ and wavenumber interval, 0.125 $cm^{-1}$. The spatial dimensions of the array are 6 cm x 6 cm.

The raw spectral estimates were computed by FWNO and the result listed by PLT3. The actual output is reproduced in Figures 2.13 through 2.29, inclusive. Each figure corresponds to a segment of the frequency-wavenumber space for which the temporal frequency is constant. Thus Figure 2.13 gives power as a function of $k_x$ and $k_y$ for 0 Hz. The power level is recorded at each point in the space in negative db with respect to recorded peak power density. The minus signs are suppressed for simplicity. The peak power density is calculated for an input given in micro-volts or micro-amperes. The power density scale is limited at -90 db. Power levels below this point are recorded as -90 db. The coding of the power density regions into grey tones is given in Table 3. The coding scheme was chosen to emphasize the higher power level and, in particular,

Table 2

Listing of overlapping planewaves

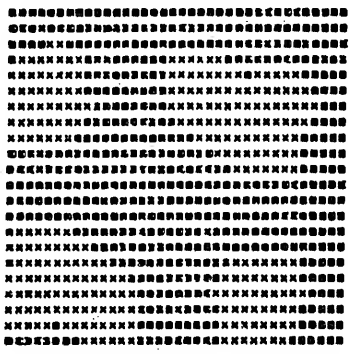
| Number | $f_0$ (Hz) | $v_0$ (cm/sec) | $\sphericalangle[v_0]$ (degrees) |
|---|---|---|---|
| 1 | 3.1 | 30.0 | 120.0 |
| 2 | 7.8 | 42.0 | 90.0 |
| 3 | 7.8 | 125.0 | 90.0 |
| 4 | 12.5 | 75.0 | 45.0 |
| 5 | 12.5 | 75.0 | -45.0 |
| 6 | 17.2 | 100.0 | 0.0 |
| 7 | 21.9 | 100.0 | 0.0 |

Table 3

Grey levels

| Level | db range | Overstruck symbols | |
|---|---|---|---|
| 1 | +∞, -3 | ( , , , ) | |
| 2 | -4, -9 | ( , , ,-) | - |
| 3 | -10,-15 | ( , , ,=) | = |
| 4 | -16,-21 | ( , ,+,+) | + |
| 5 | -22,-27 | ( , ,X,X) | X |
| 6 | -28,-33 | ( ,X,X,=) | ≠ |
| 7 | -34,-39 | ( ,*,X,O) | ▒ |
| 8 | -40,- ∞ | (#,M,W,O) | ▓ |

```
PW MATRIX FOR TEMPORAL FREQ = 0.0 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICOWATTSSECSCM**2

 38   39   41   36   41   39   38   45    4          0.250          0.0

 34   30   37   35   31   36   34   37    3          0.188          0.0

 32   29   36   35   29   29   30   37    2          0.125          0.0

 34   34   36   37   35   29   31   51    1          0.063          0.0

 37   38   43   38   43   38   37   45    0            0.0        *******

 31   29   35   37   36   34   34   51   -1         -0.063          0.0

 30   29   29   35   36   29   32   37   -2         -0.125          0.0

 34   36   31   35   37   30   34   37   -3         -0.188          0.0


 -3   -2   -1    0    1    2    3    4
```



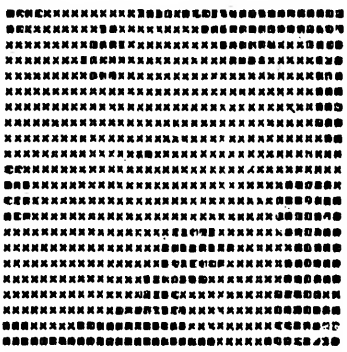
```
AT LOCAL MAX: VELOC =    0.0  CM/SEC.  BEARING =-135. DEGREES
```

Figure 2.13 FWNO output, f = 0.0 Hz.

PW MATRIX FOR TEMPORAL FREQ = 1.6 HZ
IN -08 W.R.T. PEAK POWER AT 7.8 HZ
OF 0.1961399E 04 PICOWATTSSEC*CM**2

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 39 | 32 | 32 | 34 | 35 | 38 | 36 | 42 | 4 | 0.250 | 6.25 |
| 30 | 27 | 38 | 29 | 28 | 37 | 30 | 36 | 3 | 0.188 | 8.33 |
| 29 | 30 | 29 | 28 | 29 | 26 | 28 | 39 | 2 | 0.125 | 12.50 |
| 33 | 30 | 25 | 34 | 27 | 24 | 32 | 32 | 1 | 0.063 | 25.00 |
| 37 | 32 | 32 | 32 | 27 | 29 | 36 | 34 | 0 | 0.0 | ******* |
| 30 | 32 | 30 | 29 | 41 | 32 | 33 | 42 | -1 | -0.063 | -25.00 |
| 31 | 28 | 27 | 36 | 32 | 28 | 35 | 33 | -2 | -0.125 | -12.50 |
| 36 | 35 | 38 | 42 | 34 | 33 | 34 | 37 | -3 | -0.188 | -8.33 |

-3 -2 -1 0 1 2 3 4



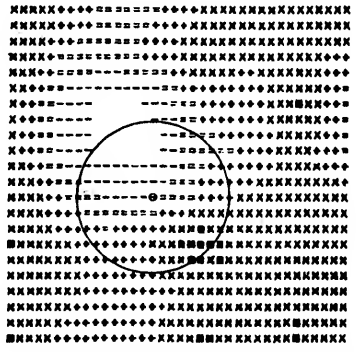
AT LOCAL MAX: VELOC = 11.18 CM/SEC. BEARING = 27. DEGREES

Figure 2.14 FWNO output, f = 1.6 Hz.

PW MATRIX FOR TEMPORAL FREQ = 3.1 HZ
IN -DB W.R.T. PEAK POWER AT 7.8 HZ
OF 0.1961399E 04 PICOWATT*SEC*CM**2

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 21 | 14 | 15 | 23 | 20 | 27 | 27 | 4 | 0.250 | 12.50 |
| 32 | 15 | 12 | 17 | 27 | 22 | 25 | 19 | 3 | 0.188 | 16.67 |
| 23 | 7 | 2 | 4 | 18 | 16 | 34 | 13 | 2 | 0.125 | 25.00 |
| 22 | 8 | 2 | 3 | 14 | 16 | 26 | 14 | 1 | 0.063 | 50.00 |
| 30 | 16 | 9 | 9 | 17 | 24 | 23 | 25 | 0 | 0.0 | ******* |
| 34 | 23 | 16 | 22 | 44 | 26 | 27 | 25 | -1 | -0.063 | -50.00 |
| 31 | 26 | 18 | 16 | 24 | 33 | 33 | 33 | -2 | -0.125 | -25.00 |
| 34 | 20 | 16 | 20 | 36 | 25 | 34 | 25 | -3 | -0.188 | -16.67 |

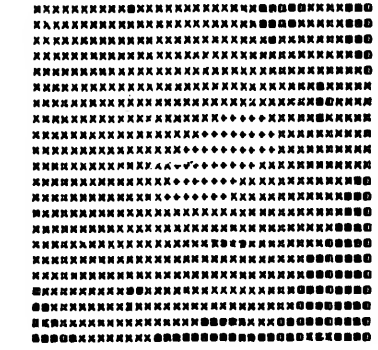
-3 -2 -1 0 1 2 3 4



AT LOCAL MAX: VELOC = 35.36 CM/SEC, BEARING = 135. DEGREES

Figure 2.15 FWNO output, f = 3.1 Hz.

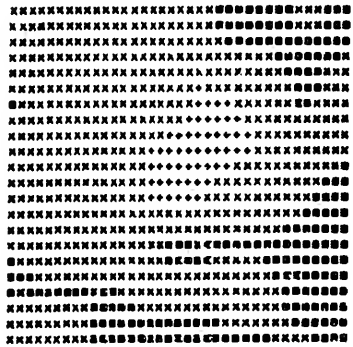
```
P# MATRIX FOR TEMPORAL FREQ = 4.7 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICO-WATT-SEC-CM**2

 28   25   35   25   25   39   30   38      4          0.250          18.75

 26   20   29   22   24   31   29   36      3          0.188          25.00

 33   25   24   32   22   22   36   29      2          0.125          37.50

 33   23   28   22   17   21   31   28      1          0.063          75.00

 28   29   29   19   20   29   29   38      0            0.0        ********

 30   28   26   27   37   31   33   36     -1         -0.063         -75.00

 34   29   34   32   25   28   38   34     -2         -0.125         -37.50

 36   33   33   34   38   36   36   45     -3         -0.188         -25.00


 -3   -2   -1    0    1    2    3    4
```



```
AT LOCAL MAX: VELOC =  53.03 CM/SEC.  BEARING =  45. DEGREES
```

Figure 2.16 FWNO output, f = 4.7 Hz.

```
PW MATRIX FOR TEMPURAL FREQ = 6.3 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICOWATT*SEC*CM**2

 26   28   31   22   24   63   30   37    4        0.250          25.00

 29   29   25   22   24   30   36   33    3        0.188          33.33

 35   22   25   29   20   22   35   30    2        0.125          50.00

 30   24   29   20   17   21   28   31    1        0.063         100.00

 29   32   28   20   21   27   30   40    0          0.0         *******

 33   30   30   34   37   35   36   44   -1       -0.063         -100.00

 38   34   34   32   26   28   37   39   -2       -0.125          -50.00

 28   27   36   35   38   35   32   39   -3       -0.188          -33.33


 -3   -2   -1    0    1    2    3    4
```



```
AT LOCAL MAX: VELOC =  70.71 CM/SEC.  BEARING =  45. DEGREES
```

Figure 2.17 FWNO output, f = 6.3 Hz.

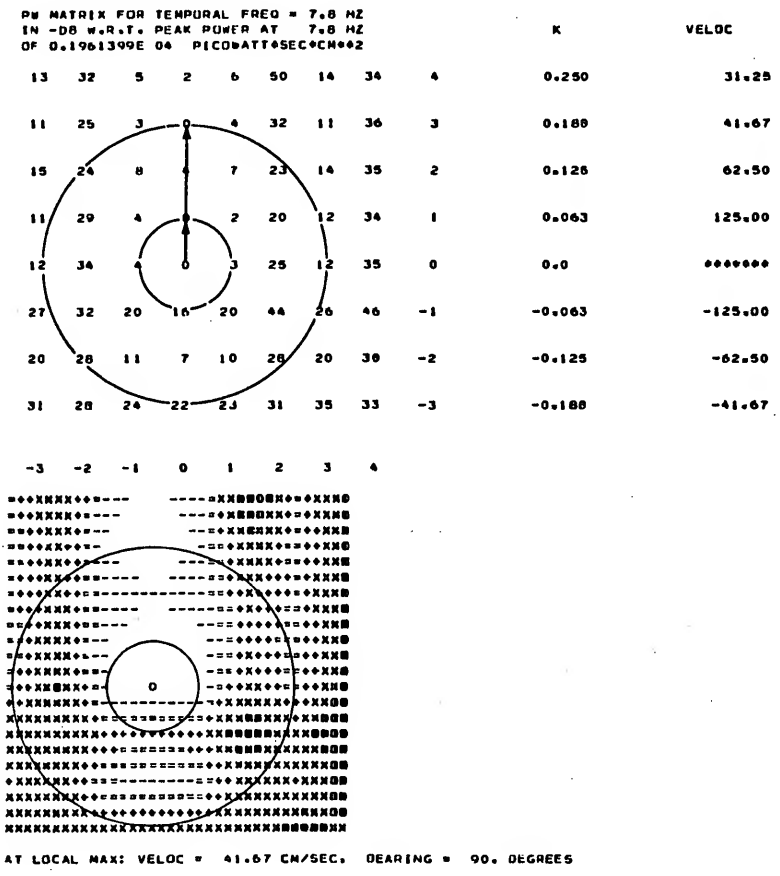
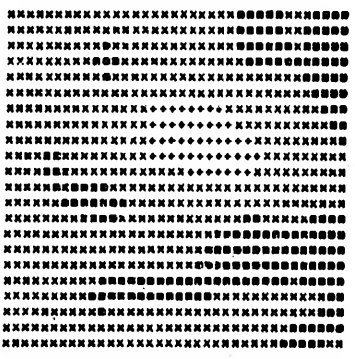



Figure 2.18 FWNO output, f = 7.8 Hz.

```
PW MATRIX FOR TEMPORAL FREQ = 9.4 HZ
IN -08 W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 00  PICOWATT*SEC*CM**2

 30   28   27   22   24   39   31   39    4          0.250          37.50

 27   25   37   28   26   34   34   61    3          0.188          50.00

 30   28   27   21   20   23   27   39    2          0.125          75.00

 30   35   27   21   19   20   24   32    1          0.063         150.00

 32   33   38   29   23   23   29   32    0          0.0           ********

 33   28   30   31   33   56   37   48   -1         -0.063         -150.00

 25   27   38   35   35   28   31   35   -2         -0.125          -75.00

 27   33   29   23   25   31   35   33   -3         -0.188          -50.00


 -3   -2   -1    0    1    2    3    4
```

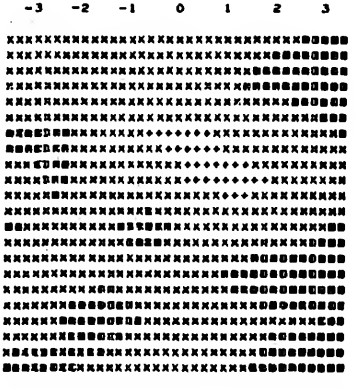


```
AT LOCAL MAX: VELOC = 106.07 CM/SEC.  BEARING =  45. DEGREES
```

Figure 2.19 FWNO output, f = 9.4 Hz.

```
P# MATRIX FOR TEMPORAL FREQ =10.9 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICO#ATT*SEC*CM**2
```

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 27 | 33 | 23 | 23 | 32 | 33 | 36 | 4 | 0.250 | 43.75 |
| 28 | 28 | 33 | 25 | 26 | 34 | 38 | 41 | 3 | 0.188 | 58.33 |
| 36 | 36 | 26 | 21 | 21 | 23 | 26 | 34 | 2 | 0.125 | 87.50 |
| 29 | 36 | 28 | 23 | 21 | 20 | 23 | 30 | 1 | 0.063 | 175.00 |
| 35 | 30 | 30 | 40 | 24 | 23 | 28 | 35 | 0 | 0.0 | ******* |
| 30 | 27 | 30 | 29 | 29 | 38 | 42 | 39 | -1 | -0.063 | -175.00 |
| 28 | 31 | 44 | 32 | 33 | 30 | 33 | 34 | -2 | -0.125 | -87.50 |
| 36 | 36 | 30 | 23 | 23 | 30 | 50 | 43 | -3 | -0.188 | -58.33 |




```
AT LOCAL MAX: VELOC =  78.26 CM/SEC,  BEARING =  27. DEGREES
```

Figure 2.20 FWNO output, f = 10.9 Hz.

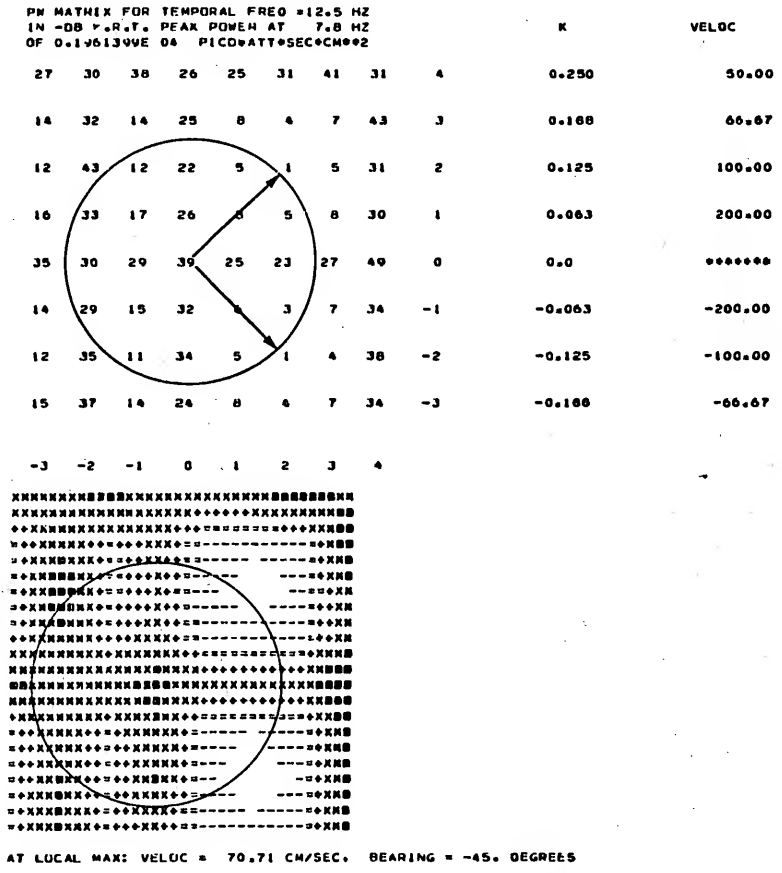
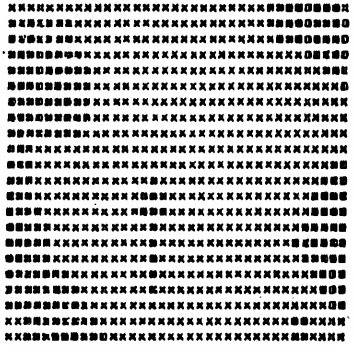



Figure 2.21 FWNO output, f = 12.5 Hz.

```
P# MATRIX FOR TEMPORAL FREQ =14.1 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 00  PICO*WATTS*SEC*CM**2

 31   33   33   31   28   30   43   33      4          0.250          56.25

 43   36   30   27   27   29   33   39      3          0.188          75.00

 36   45   30   25   23   24   26   31      2          0.125         112.50

 34   33   30   30   26   22   24   32      1          0.063         225.00

 37   31   28   36   26   23   27   48      0            0.0       ********

 47   31   29   34   29   28   34   38     -1         -0.063        -225.00

 36   36   31   34   30   27   32   35     -2         -0.125        -112.50

 29   41   34   28   24   25   35   31     -3         -0.188         -75.00


 -3   -2   -1    0    1    2    3    4
```



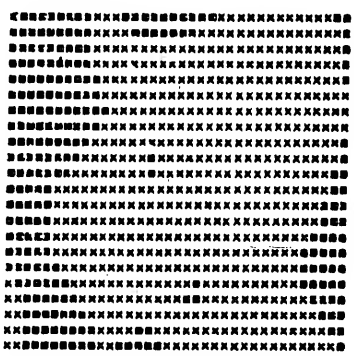
```
AT LOCAL MAX: VELOC = 100.62 CM/SEC.  BEARING =  27. DEGREES
```

Figure 2.22 FWNO output, f = 14.1 Hz.

```
Pw MATRIX FOR TEMPORAL FREQ =15.6 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICOWATT*SEC*CM**2
```

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 35 | 36 | 32 | 37 | 35 | 30 | 32 | 34 | 4 | 0.250 | 62.50 |
| 41 | 38 | 32 | 30 | 30 | 29 | 29 | 34 | 3 | 0.188 | 83.33 |
| 37 | 52 | 34 | 29 | 26 | 24 | 28 | 33 | 2 | 0.125 | 125.00 |
| 38 | 36 | 30 | 34 | 28 | 23 | 25 | 34 | 1 | 0.063 | 250.00 |
| 41 | 32 | 27 | 33 | 29 | 24 | 27 | 39 | 0 | 0.0 | ******* |
| 38 | 33 | 26 | 30 | 33 | 27 | 33 | 39 | -1 | -0.063 | -250.00 |
| 31 | 38 | 29 | 32 | 34 | 28 | 33 | 35 | -2 | -0.125 | -125.00 |
| 30 | 45 | 33 | 35 | 29 | 26 | 31 | 34 | -3 | -0.188 | -83.33 |

| -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|



AT LOCAL MAX: VELOC = 111.80 CM/SEC.  BEARING =  27. DEGREES

Figure 2.23 FWNO output, f = 15.6 Hz.

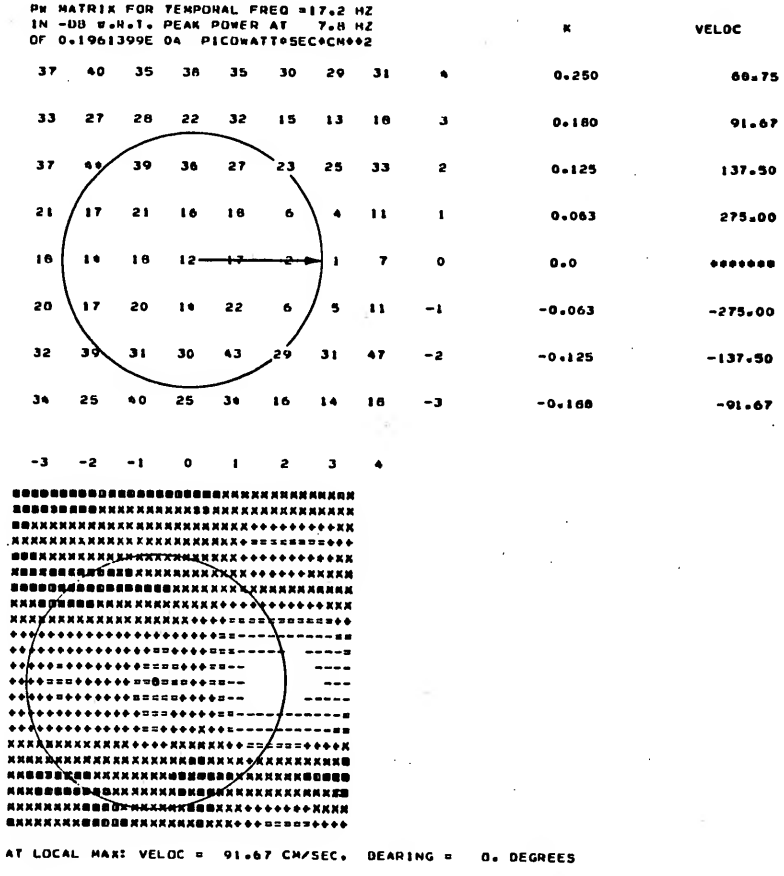
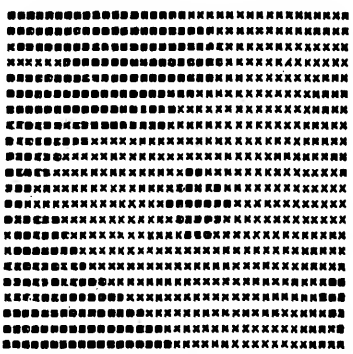



Figure 2.24 FWNO output, f = 17.2 Hz.

PW MATRIX FOR TEMPORAL FREQ =18.8 HZ
IN -DB W.R.T. PEAK POWER AT 7.8 HZ
OF 0.1961399E 04 PICOWATT*SEC*CM**2

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 38 | 41 | 38 | 31 | 33 | 30 | 29 | 4 | 0.250 | 75.00 |
| 33 | 33 | 38 | 50 | 41 | 28 | 25 | 27 | 3 | 0.188 | 100.00 |
| 36 | 42 | 42 | 38 | 28 | 24 | 26 | 33 | 2 | 0.125 | 150.00 |
| 35 | 34 | 28 | 31 | 29 | 27 | 28 | 29 | 1 | 0.063 | 300.00 |
| 34 | 32 | 25 | 27 | 46 | 27 | 24 | 25 | 0 | 0.0 | ******* |
| 33 | 40 | 27 | 27 | 31 | 27 | 27 | 30 | -1 | -0.063 | -300.00 |
| 38 | 36 | 37 | 30 | 33 | 29 | 28 | 39 | -2 | -0.125 | -150.00 |
| 38 | 40 | 46 | 36 | 31 | 26 | 26 | 32 | -3 | -0.188 | -100.00 |

-3 -2 -1 0 1 2 3 4



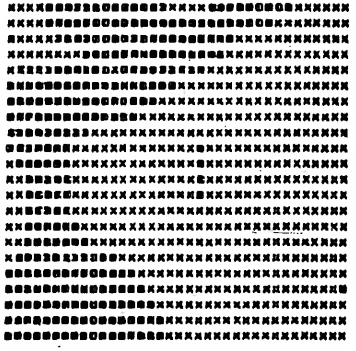
AT LOCAL MAX: VELOC = 106.07 CM/SEC, BEARING = 45. DEGREES

Figure 2.25 FWNO output, f = 18.8 Hz.

PW MATRIX FOR TEMPORAL FREQ =20.3 HZ
IN -DO W.R.T. PEAK POWER AT 7.8 HZ
OF 0.1961399E 04 PICOWATTSSECSCMSS2

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 30 | 36 | 34 | 34 | 32 | 40 | 33 | 28 | 4 | 0.250 | 81.25 |
| 31 | 33 | 34 | 40 | 37 | 29 | 26 | 27 | 3 | 0.188 | 108.33 |
| 38 | 45 | 40 | 34 | 29 | 26 | 28 | 33 | 2 | 0.125 | 162.50 |
| 34 | 36 | 27 | 29 | 34 | 29 | 26 | 29 | 1 | 0.063 | 325.00 |
| 32 | 36 | 25 | 26 | 34 | 28 | 26 | 27 | 0 | 0.0 | ******* |
| 31 | 41 | 30 | 27 | 29 | 29 | 29 | 30 | -1 | -0.063 | -325.00 |
| 36 | 35 | 47 | 33 | 33 | 30 | 29 | 34 | -2 | -0.125 | -162.50 |
| 39 | 44 | 41 | 35 | 30 | 28 | 29 | 32 | -3 | -0.188 | -108.33 |

-3 -2 -1 0 1 2 3 4



AT LOCAL MAX: VELOC = 325.00 CM/SEC, BEARING = 180. DEGREES

Figure 2.26 FWNO output, f = 20.3 Hz.

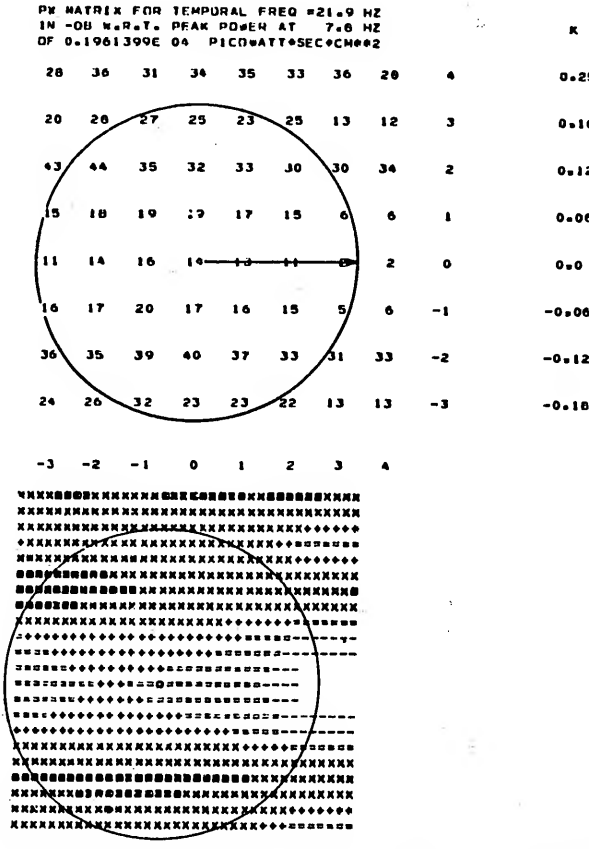
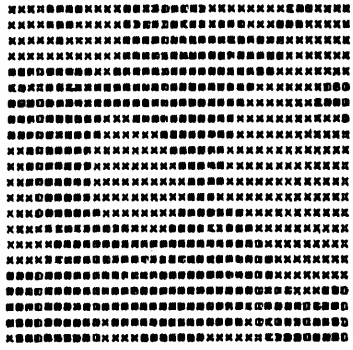



Figure 2.27 FWNO output, f = 21.9 Hz.

PN MATRIX FOR TEMPORAL FREQ =23.4 HZ
IN -DB W.R.T. PEAK POWER AT 7.8 HZ
OF 0.1961399E 04 PICOWATTSSECSCM**2

| | | | | | | | | | K | VELOC |
|---|---|---|---|---|---|---|---|---|---|---|
| 29 | 35 | 32 | 39 | 34 | 31 | 35 | 29 | 4 | 0.250 | 93.75 |
| 33 | 33 | 32 | 38 | 41 | 37 | 32 | 30 | 3 | 0.188 | 125.00 |
| 42 | 39 | 35 | 36 | 45 | 32 | 32 | 38 | 2 | 0.125 | 187.50 |
| 30 | 46 | 29 | 30 | 40 | 28 | 26 | 28 | 1 | 0.063 | 375.00 |
| 28 | 44 | 29 | 28 | 35 | 31 | 25 | 23 | 0 | 0.0 | ******* |
| 27 | 35 | 41 | 34 | 38 | 37 | 28 | 28 | -1 | -0.063 | -375.00 |
| 37 | 34 | 35 | 42 | 45 | 36 | 33 | 34 | -2 | -0.125 | -187.50 |
| 33 | 51 | 32 | 34 | 34 | 31 | 42 | 34 | -3 | -0.188 | -125.00 |

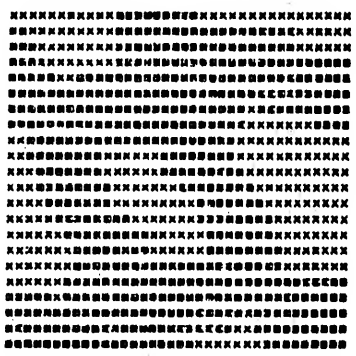
-3 -2 -1 0 1 2 3 4



AT LOCAL MAX: VELOC = 93.75 CM/SEC, BEARING = 0. DEGREES

Figure 2.28 FWNO output, f = 23.4 Hz.

```
PW MATRIX FOR TEMPORAL FREQ =25.0 HZ
IN -DB W.R.T. PEAK POWER AT   7.8 HZ
OF 0.1961399E 04  PICOWATT*SEC*CM**2

 32   32   33   49   33   32   32   30      4          0.250          100.00

 38   31   32   43   39   55   34   34      3          0.188          133.33

 36   37   40   48   36   33   36   39      2          0.125          200.00

 29   44   33   33   43   31   27   26      1          0.063          400.00

 26   36   34   30   34   36   26   24      0            0.0        ********

 27   31   43   33   33   44   29   26     -1         -0.063         -400.00

 36   33   36   48   40   37   36   39     -2         -0.125         -200.00

 34   55   39   43   32   31   30   34     -3         -0.188         -133.33


 -3   -2   -1    0    1    2    3    4
```



```
AT LOCAL MAX: VELOC = 100.00 CM/SEC.  BEARING =   0. DEGREES
```

Figure 2.29 FWNO output, f = 25.0 Hz.

the -3 db contour. Subroutine GREY (Appendix D) generates the printed output. The grey levels are produced by overprinting as shown in Table 3. Linear interpolation is used to determine the level between given data points.

The first two figures exhibit no power density greater than -24 db. In Figure 2.15, a peak power density of -2 db is recorded. This is obviously due to the first wave listed in Table 2. The db number array is rounded to 1 db before listing, but the determination of peak values is based on single precision floating point arithmetic. Thus the actual local maximum occurs at $\vec{v} = 35.36|\underline{135}°$ or $k_x = -0.063$ and $k_y = +0.063$. The circles of constant velocity = 30.0 cm/sec are superimposed on the actual computer output. From the contour plot, it is apparent that the "centroid" of the ±3 db region approximately corresponds to the actual vector velocity. The superimposed arrows correspond to the true velocity. The estimated velocity is given below the grey plot.

Figures 2.16 and 2.17 exhibit only small power levels, but the twin peaks of Figure 2.18 obviously correspond to the second and third entries of Table 2. The inner concentric circle represents the locus of points of 125.0 cm/sec and the outer circle 42.0 cm/sec. This figure illustrates that the wavenumber resolution is about 0.125 $cm^{-1}$. A comparison of Figure 2.15 with Figure 2.18 may lead to the erroneous conclusion that the first wave is 2 db less than the third and fourth. This is a resolution problem since at 3.1 Hz the power density at the actual maximum ($k_x = .087$, $k_y = 0.54$) was not computed as it was in the

latter case and, instead, we have the peak power density near the maximum.

In Figure 2.21, the circle corresponds to 75 cm/sec and the two peaks account for the fourth and fifth entries of Table 2.

The circles of Figures 2.24 and 2.27 both correspond to 100 cm/sec and the peak of these frequencies corresponds to the last two entries in Table 2. Figure 2.27 also illustrates the effect of spatial aliasing. The actual wave has a center wavenumber of k = 0.219, but, due to truncation, the actual wavenumber is spread beyond the aliasing point of k = 0.25. As a result the tails of high wavenumbers begin to fold back onto negative wavenumbers. This accounts for the -11 db level at the left side of Figure 2.27. The effects of sidelobe leakage can be observed in Figure 2.24 above, below and to the left of the main peak. As these sub-peaks are not much more than 10 db below the main peak, caution must be exercised so that they are not misinterpreted as additional planewaves.

If the k scale of Figures 2.13 through 2.29 is carefully observed, it may be noted that there is one more positive term in each of the k directions than there are negative terms. This is a characteristic of the FFT which has no counterpart in one-dimensional frequency series. To understand why this occurs, consider a band-limited function as shown in Figure 2.30(a). If the time function is sampled at exactly the Nyquist rate F, the frequency function of Figure 2.30(b) results. If the transform of the periodic extension of the truncated sampled function is performed, we have the DFT as shown in Figure 2.30(c).

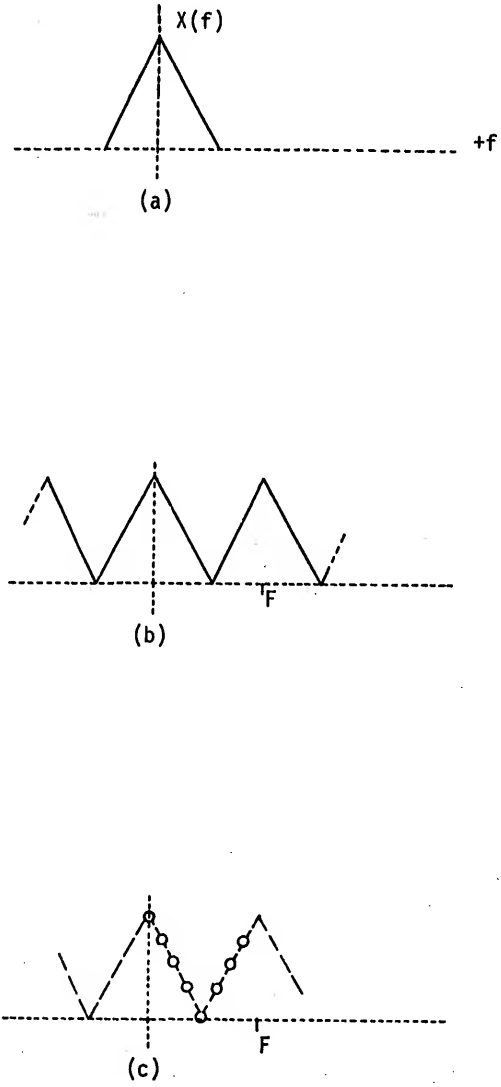



Figure 2.30 Example of positive and negative frequency terms.

If there are N points in the series the first (N/2) + 1 correspond to positive frequency and the remainder, (N/2) - 1, correspond to negative frequency. Since the DFT of real one-dimensional series is always symmetrical with respect to positive and negative frequency, the calculated negative frequency may be ignored. In the wavenumber space, however, this symmetry does not exist and there is a difference in the number of points on the positive and negative scales. This apparent deficiency is easily eliminated by performing a second transform with the spatial directions reversed. Now the negative frequencies will be computed to (N/2) + 1 terms. For two spatial dimensions the x and y axes are negated by not calling NEGATE before transforming to the frequency-wavenumber domain. If necessary, the x and y axes could easily be negated separately to obtain the full range in all coordinates.

## Summary

This chapter has presented a detailed account of the frequency-wavenumber space. This should be beneficial in studying the more complex subjects of multidimensional filtering, Chapter III, and maximum-likelihood estimation, Chapter IV.

The principal contribution of this chapter is the computationally efficient direct FFT method of estimating the frequency-wavenumber power spectral density. This, too, will be used in succeeding chapters.

# CHAPTER III

# MULTIDIMENSIONAL DIGITAL FILTERING

This chapter has two purposes. The first is to present a method of realizing a digital filter (one-dimensional) for an arbitrary response specification given in the frequency domain. This technique is needed in the implementation of the maximum-likelihood filter which is, in turn, a necessary constituent of the maximum-likelihood estimator. The second purpose is to present a multidimensional filter which can be used to stop, pass or modify a traveling planewave. This filter is necessary to reduce the multiwave maximum-likelihood estimation problem to that of a single wave estimation problem.

## One-dimensional Filtering

Recursive filters have an impulse response of infinite duration and the output of such a filter is a function of both the past output and the past input. The digital approximation to these is based on classical analog filter theory and leads to sophisticated design techniques such as Butterworth, Chebyshev, and elliptical filters [13].

Nonrecursive digital filters are approximations to filters which have a finite impulse response and the output of this type is only a function of the past input. The principal advantage of the nonrecursive filter is the much greater degree of design flexibility. Only nonrecursive digital filters are considered in this study.

The digital approximation of nonrecursive filters can be approached in one of two ways. The first method, the window method, truncates and samples the impulse response of an ideal filter by using a suitable truncation window. The actual frequency response is then the convolution of the ideal response with the transform of the truncation window. The problem of choosing a suitable window function is the same problem encountered in forming acceptable power spectral estimates. A drawback of this technique is that the optimality criteria must be expressed in terms of the transform of the window rather than explicitly in terms of the resultant actual frequency response. Helms [14] shows how the main-lobe width and sidelobe width can be adjusted in an optimum fashion using a Dolph-Chebyshev window.

The second approach to nonrecursive digital filtering overcomes this difficulty for special cases by choosing transition values of the frequency response so as to minimize the maximum out-of-band response. This method, called the sampling method, is given in [28]. The synthesis proceeds as follows:

1. the exact frequency response is specified at equidistant points in the frequency domain for the stopband and passband;

2. the transition points are selected by using computer optimization techniques which calculate the maximum out-of-band response as a function of the transition values. The frequency response is interpolated to any desired degree of resolution by using FFT techniques as explained later. Computer optimization is used because it is impossible to obtain general closed form solutions. Unfortunately, it is not possible to establish general filter design guidelines from [28].

Instead one must consult the tabulated results for optimum transition values. If a filter is required which was not considered in [28], the costly optimization must be repeated.

The window method and the sampling method are used in subsequent data analysis; therefore, an example of both for low-pass filtering is given. In Figure 3.1, a low-pass 16-point filter is given with the narrowest possible transition width. Only positive frequencies are shown. The imaginary terms are specified to be zero. The out-of-band terms are specified to be zero, but appear at -90 db due to plotting conventions. An inverse FFT is used to transform this function to the time (impulse) domain. The result is shown in Figure 3.2. The ordering, characteristic of discrete transforms, should be noted in this figure. Positive time is in ascending order (9 points), left to right, followed by negative time (7 points) also in ascending order. The impulse response is padded with zeros as shown in Figure 3.3. The interpolated frequency is then obtained by a forward FFT as shown in Figure 3.4.

For comparison purposes the window (Hanning) method is applied to the ideal response of Figure 3.1, yielding the specified response of Figure 3.5. For the specified Hanning window, smoothing (convolution) was performed in the frequency domain rather than by multiplication in the time domain in order to eliminate a redundant transformation. The impulse response was then obtained as before, yielding the response of Figure 3.6. After zeros are added and the series is forward transformed to the frequency domain, the interpolated amplitude response of Figure 3.7 is obtained.

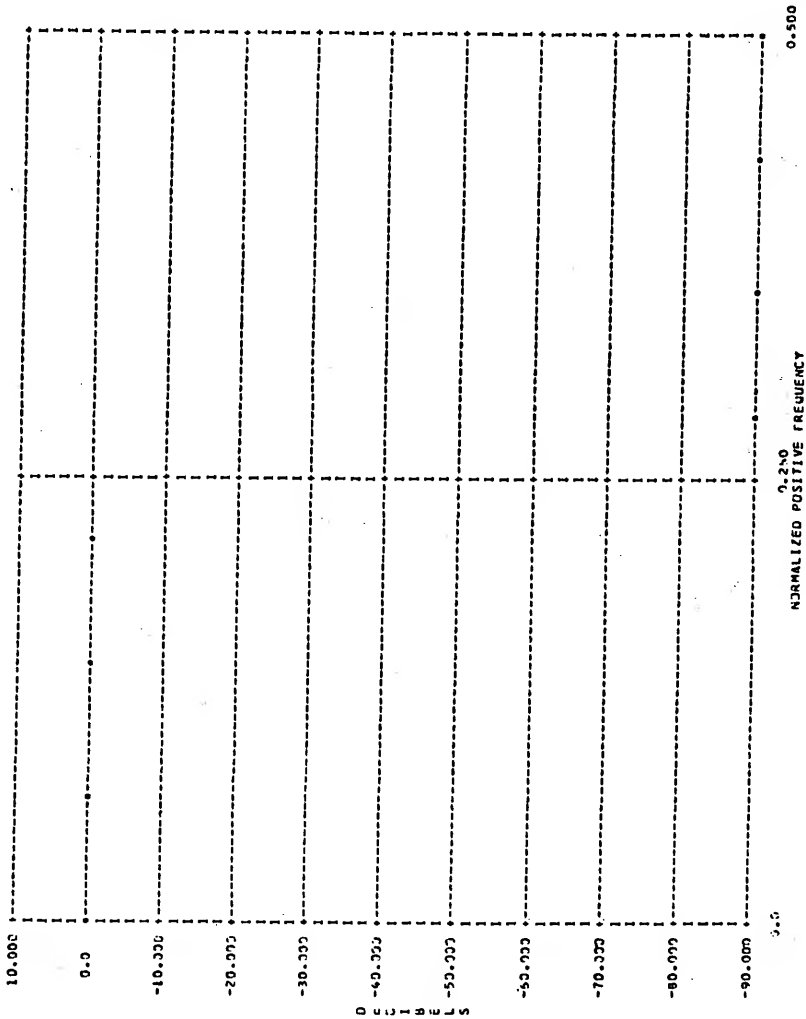



Figure 3.1 Minimum transition width low-pass filter amplitude response.

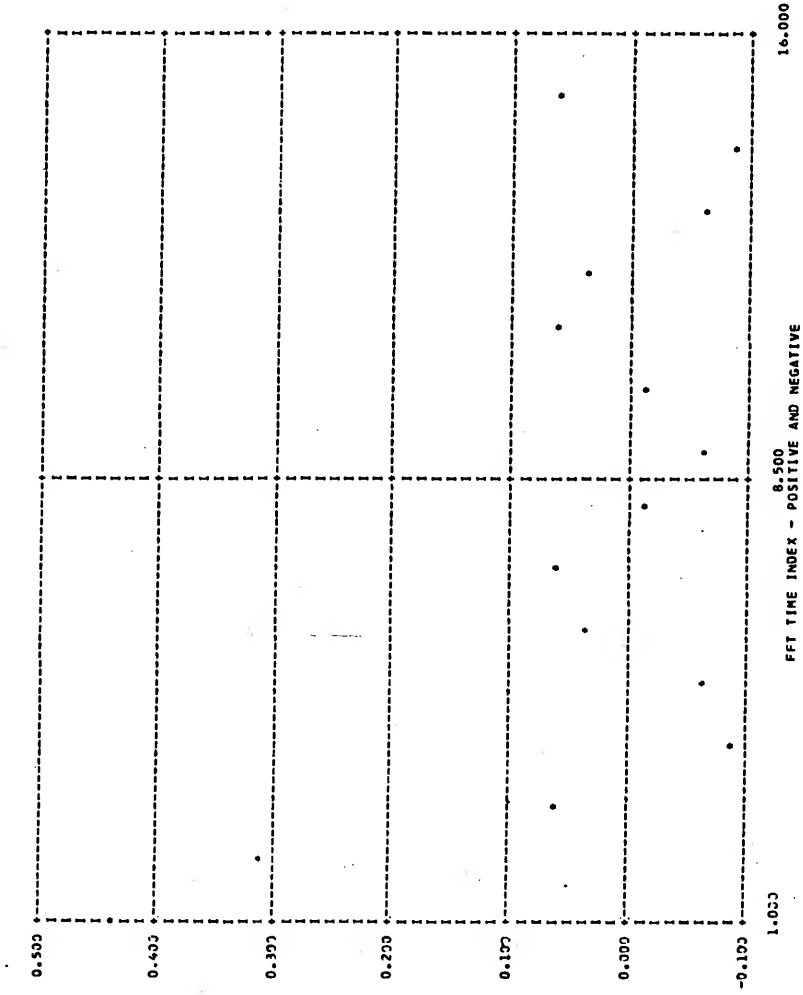



Figure 3.2 Minimum transition width low-pass filter impulse response.

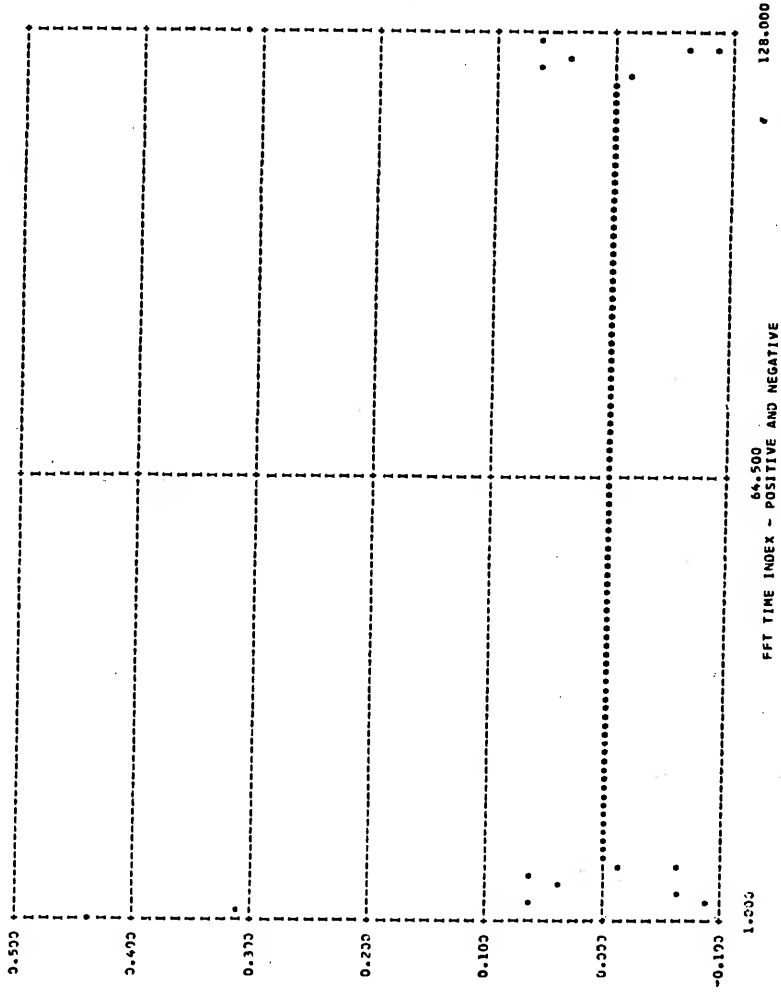



Figure 3.3 Impulse response with zeros added.

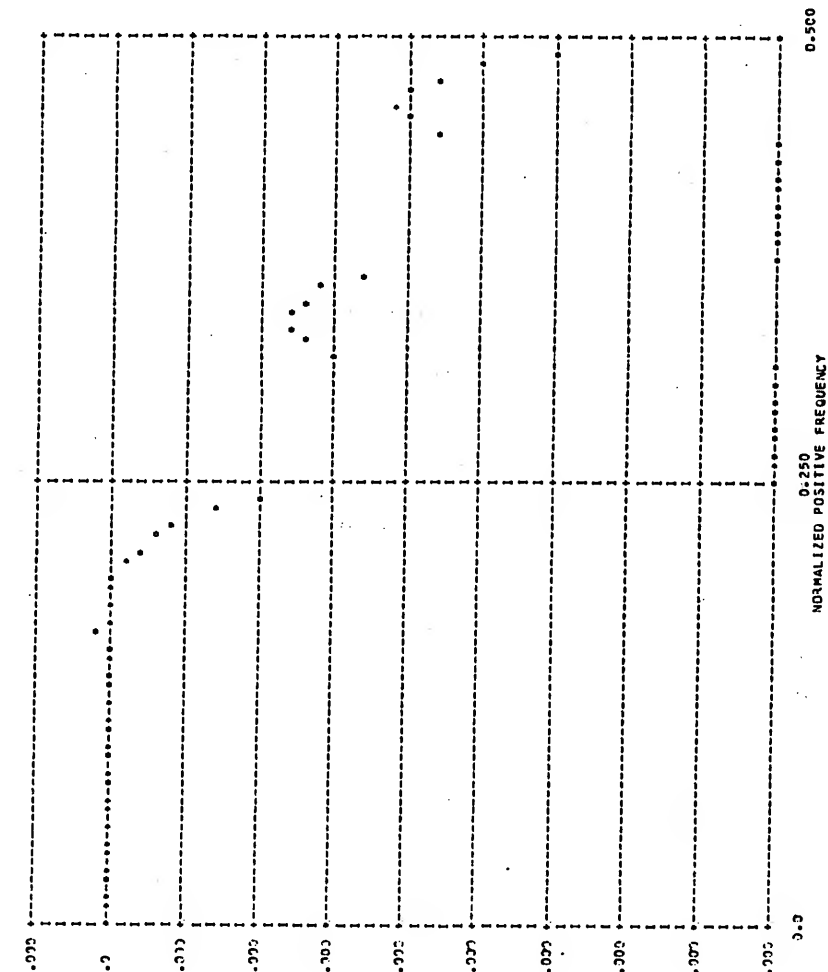



Figure 3.4 Minimum transition width interpolated amplitude response.

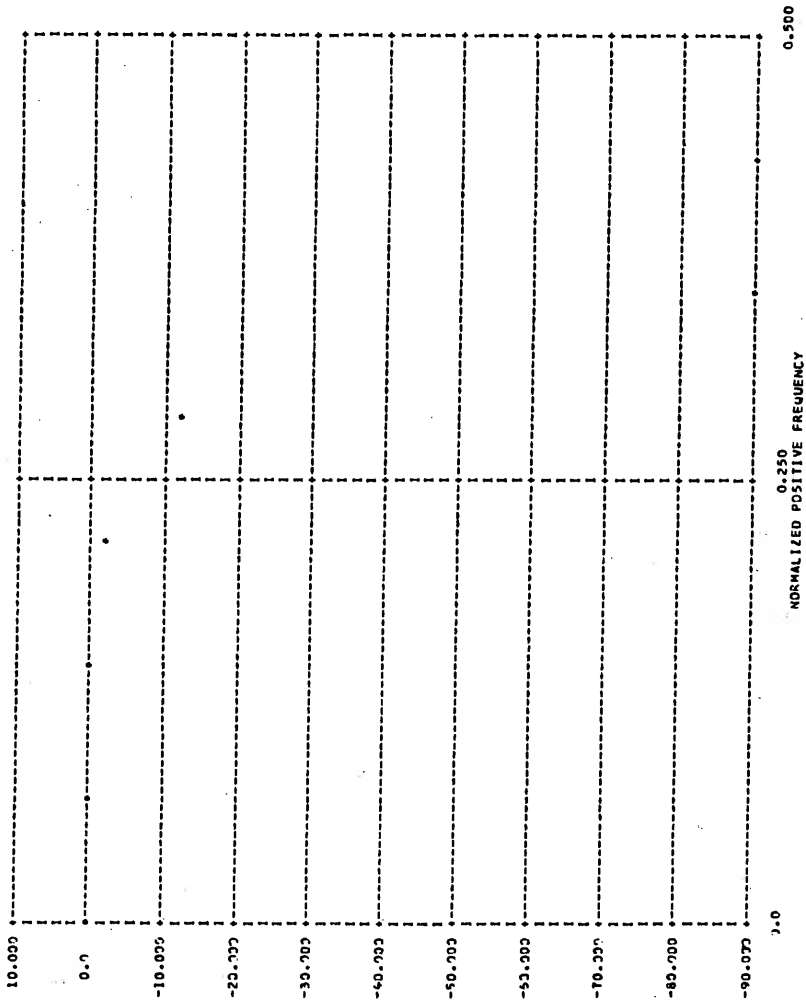



Figure 3.5 Hanning window transition low-pass filter amplitude response.

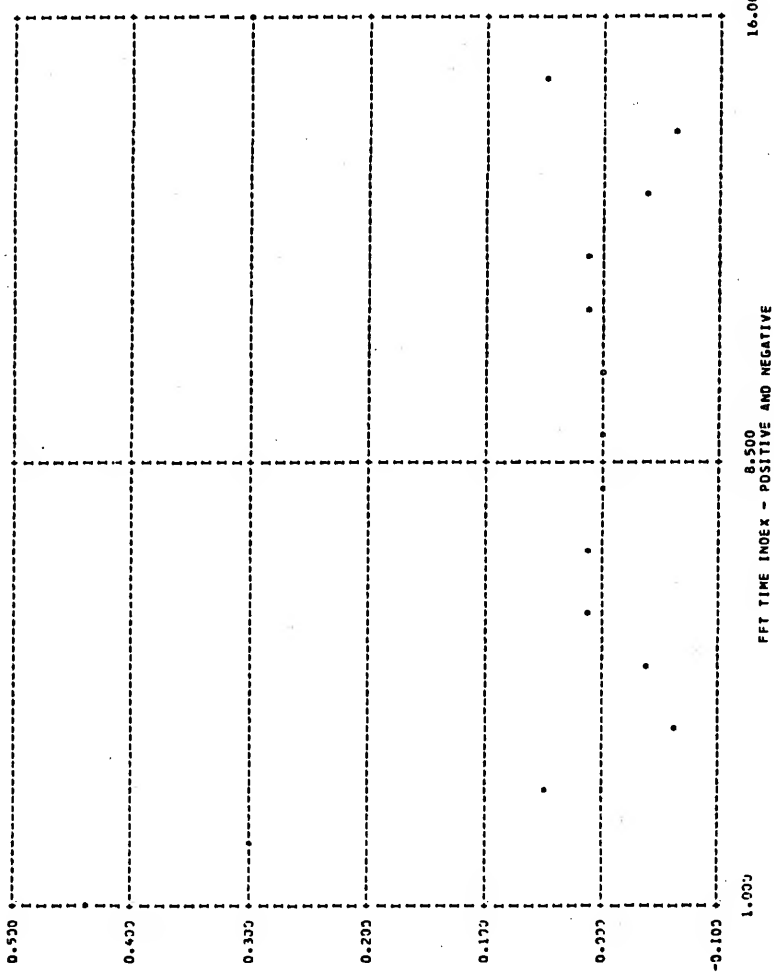



Figure 3.6 Hanning window transition low-pass filter impulse response.

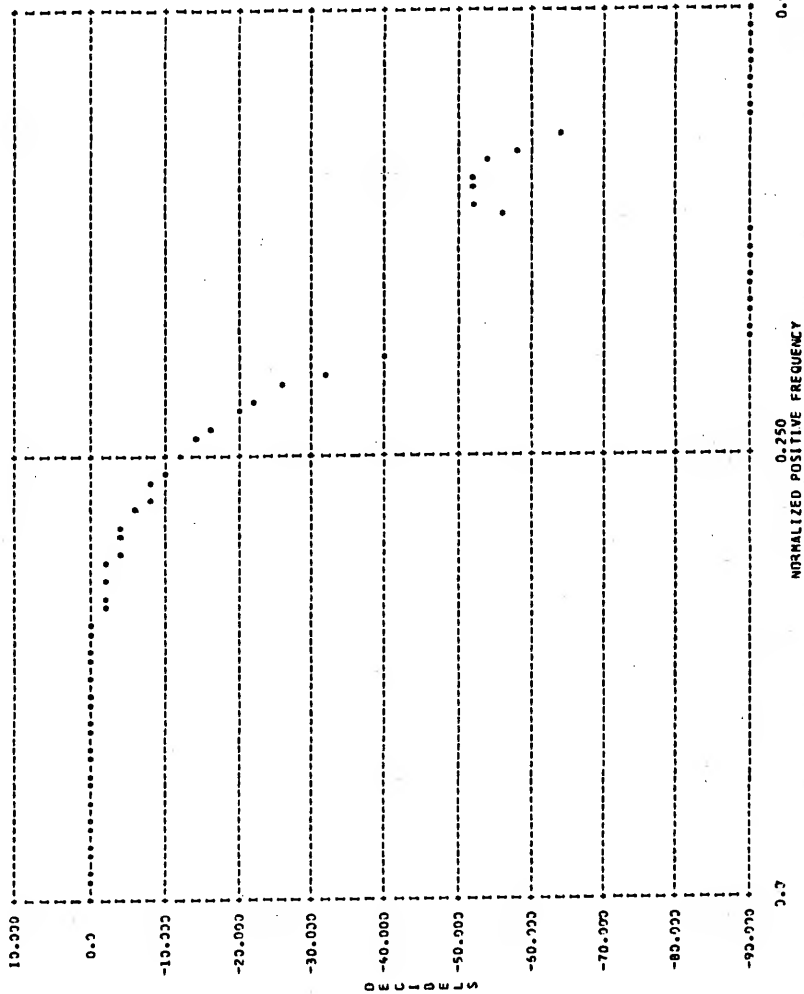



Figure 3.7 Hanning window transition interpolated amplitude response.

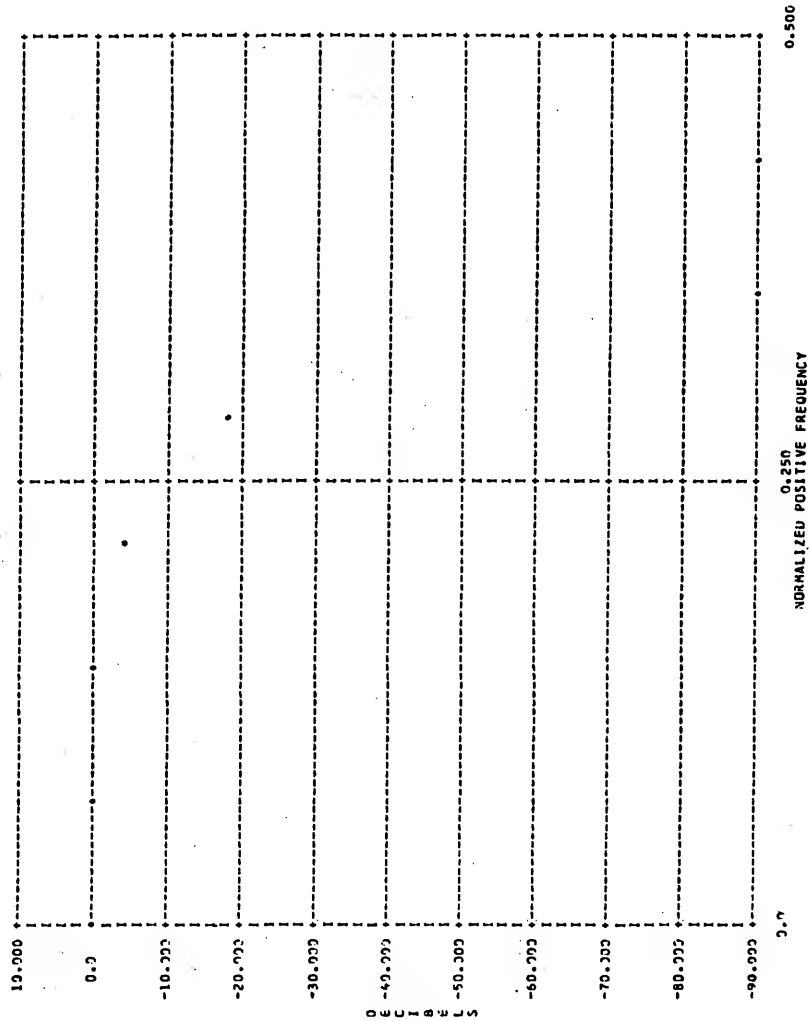



Figure 3.8 Optimum 2-point transition low-pass filter amplitude response.

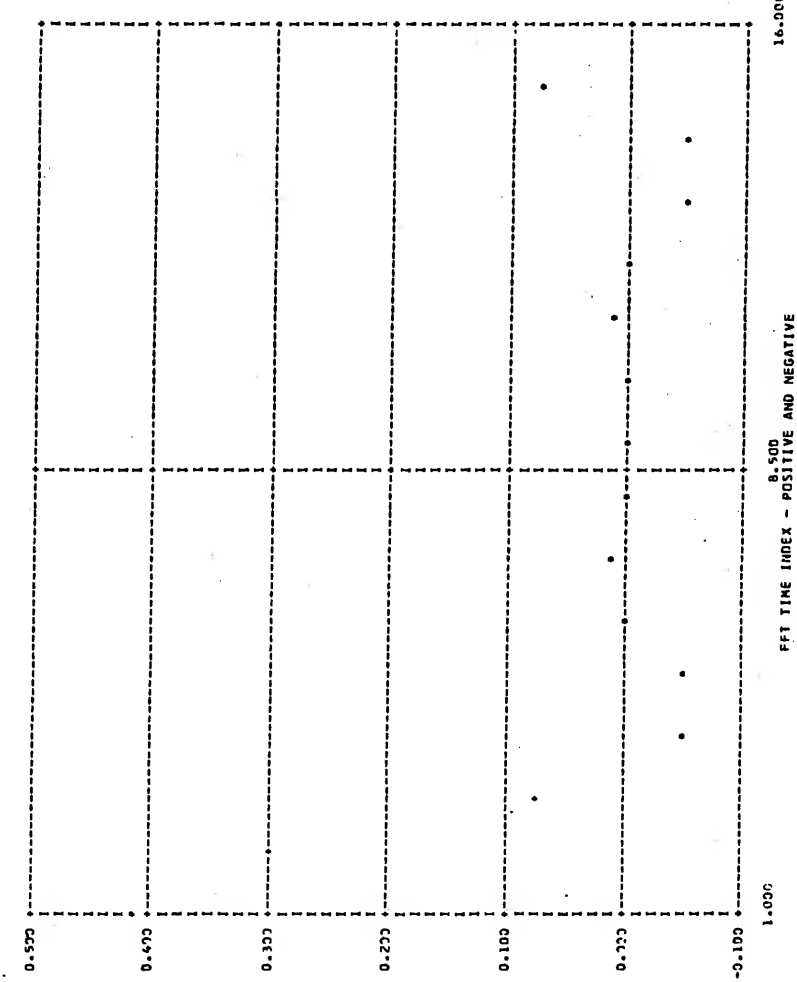



Figure 3.9 Optimum 2-point transition low-pass filter impulse response.

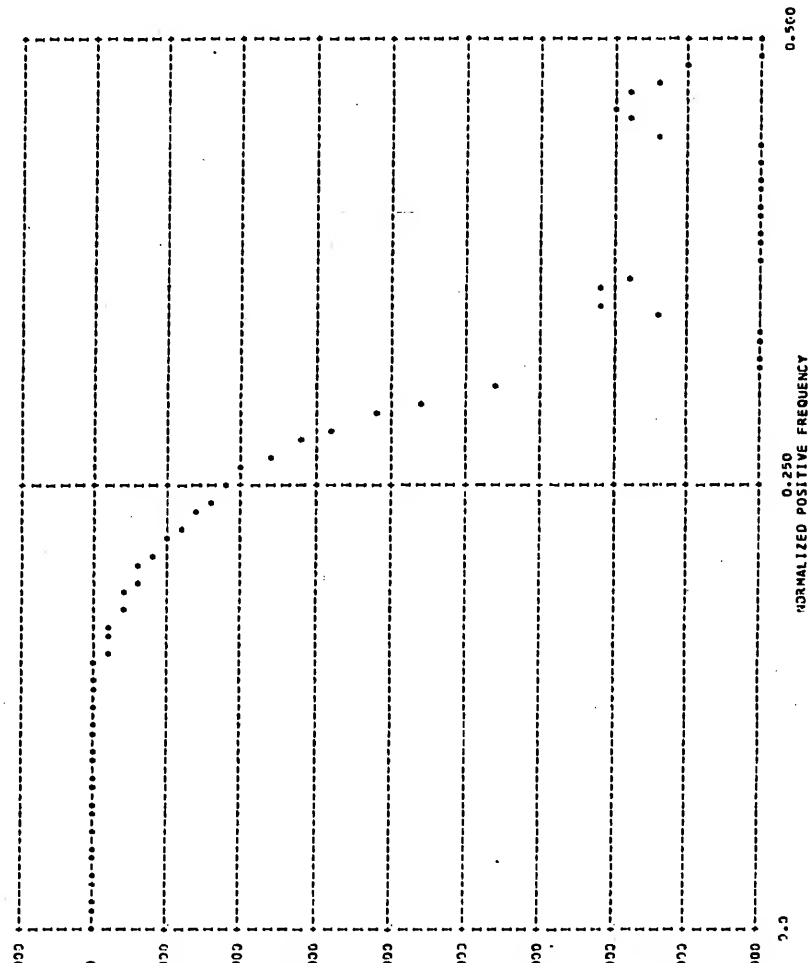



Figure 3.10 Optimum 2-point transition interpolated amplitude response.

The optimum 2-point transition values of the sampling method are illustrated in Figure 3.8. The impulse response is given in Figure 3.9 and the interpolated response is given in Figure 3.10.

The response of Figure 3.4 which is, in effect, due to a rectangular window is seen to have a sharp transition but a rather large out-of-band response. The Hanning window method lowers the out-of-band response, Figure 3.7, at the expense of a wider transition region. The sampling method provides an additional reduction in the out-of-band response with no further increase in transition bandwidth. These results obviously argue in favor of the sampling method, but it must be emphasized that this technique can only, at present, be applied to a limited class of filters. The window method, however, can be applied to any desired frequency response function. The maximum-likelihood filter of Chapter IV requires a filter which is a function of the power spectrum of the input data. Since sharp peaks in the spectrum result in very narrow transition in the filter amplitude response, it was found that the filter could be successfully approximated by first limiting the dynamic range of the spectrum and then applying a Hanning window.

Subroutine AHANN of Appendix D is used to Hanning smooth an arbitrary series in the frequency domain. For bandstop or bandpass filtering of a given time series, subroutine BPF may be employed. BPF automatically smooths the specified transition and returns the filtered time series. Subroutine SPBPF is an example of a special purpose (i.e., fixed transition and fixed number of filter points) bandpass filter

which utilizes the sampling method. This filter is used in Chapter V to pre-condition data for multidimensional analysis.

All of the digital filters developed for this study are specified in the frequency domain as real functions. In a simulation sense this is equivalent to zero phase shift. To determine how closely this condition is approximated, all transformations were done using complex arithmetic and the imaginary interpolated frequency response was monitored. It was found that for filters of all dimensionality the imaginary component was consistently at least 30 db lower than the real component. Consequently, it is reasonable to assume that negligible phase shift has occurred.

## Three-dimensional Filtering

In most practical instances sensor data will be available in a spatially discrete form. It may, as in Chapter V, be temporally discrete also, but, for the moment, this condition will be ignored. Under these conditions the most general linear filtering that may be applied to the data consists of separate linear filtering of each channel (sensor output). This is referred to as generalized multichannel linear filtering and is illustrated in Figure 3.11. Delay and sum, weighted delay and sum, and even maximum-likelihood filtering can be put into this form.

If the spatio-temporal data are not sampled and are treated as continuous in all dimensions, the most general linear filter is the multidimensional filter. The basic properties of the three-dimensional filter will be considered. The N-dimensional case is an obvious extension.

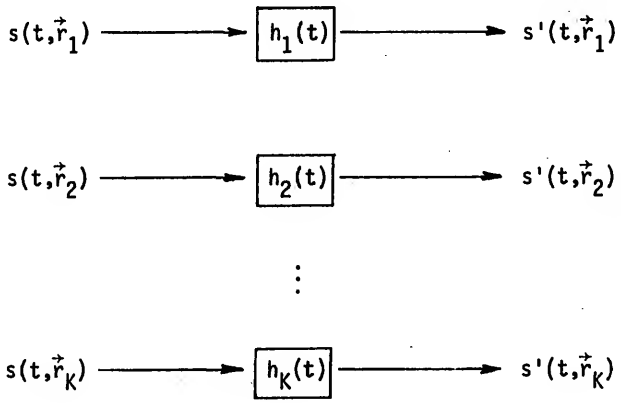



Figure 3.11 Generalized multichannel filter.

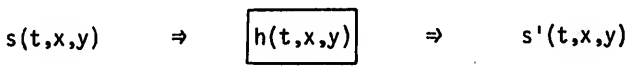



Figure 3.12 Generalized multidimensional filter.

The three-dimensional linear time and space invariant impulse response $h(t,x,y)$ is the response at time t and position $(x,y)$ to an impulse applied at the space-time origin. If $s(t,x,y)$ is the input to such a filter, note Figure 3.12, the output is given as the three-dimensional convolution of the input with the impulse response,

$$s'(t,x,y) = \int_{-\infty}^{+\infty}\int\int h(\tau,\alpha,\beta)\, s(t-\tau,x-\alpha,y-\beta)\, d\tau d\alpha d\beta. \qquad (3\text{-}1)$$

Furthermore, since $s(t,x,y)$ has a three-dimensional transform

$$S(f,k_x,k_y) = \int_{-\infty}^{+\infty}\int\int s(t,x,y)\, e^{-i2\pi(ft-k_x x-k_y y)}\, dt\, dx\, dy, \qquad (3\text{-}2)$$

the output can be expressed as

$$s'(t,x,y) = F_{3D}^{-1}\, S(f,k_x,k_y)\, H(f,k_x,k_y) \equiv$$

$$\int_{-\infty}^{+\infty}\int\int S(f,k_x,k_y)\, H(f,k_x,k_y)\, e^{+i2\pi(\cdot)}\, df\, dk_x\, dk_y, \qquad (3\text{-}3)$$

where $H(f,k_x,k_y)$ is the complex three-dimensional frequency-wavenumber response of the three-dimensional filter.

The spatial transform of the frequency-wavenumber response yields

$$H_o(f,\vec{r}) = \int H(f,\vec{k})\, e^{i2\pi(\vec{k}\cdot\vec{r})}\, d\vec{k}\ , \qquad (3\text{-}4)$$

where $\vec{r} = (x,y)$ and $\vec{k} = (k_x,k_y)$. The complementary transform is

$$H(f,\vec{k}) = \int H_o(f,\vec{r})\, e^{-2\pi(\vec{k}\cdot\vec{r})}\, d\vec{r}\ , \tag{3-5}$$

where $H_o(f,\vec{r})$ is an infinite set (with respect to space) of one-dimensional filters. If we approximate $H(f,\vec{k})$ in (3-4) by restricting $H_o$ to be a sample point operator and limit the number of samples to K, we have

$$\hat{H}(f,\vec{k}) = \sum_{j=1}^{K} H_o(f,\vec{r}_j)\, e^{-i2\ (\vec{k}\cdot\vec{r}_j)}\ . \tag{3-6}$$

The question of obtaining acceptable approximations is readily recognized as a problem in obtaining acceptable estimates of multidimensional Fourier transforms using truncated data. It has thus been shown that if spatial stationarity exists, we may transform a multidimensional filter to K one-dimensional filters. In surveying the literature on the subject of spatio-temporal filtering, it was found that this technique is commonly employed in seismology to effect velocity filtering of planewaves [29,30].

The filter requirement is usually developed in the multidimensional domain, transformed to one-dimensional filters, and implemented using discrete convolution. In this chapter an alternative solution is presented which consists of digital filtering directly in the multidimensional domain. This method offers substantial savings in computational effort if the multidimensional FFT is employed. Table 4 compares the number of operations required for three different, although equivalent, filters.

As stated earlier, the purpose of the filter, which precedes the single wave maximum-likelihood estimator, is to pass a single planewave

and stop all others. If we assume continuous and infinite data in all three dimensions and adapt Wiener's least mean square error as a criterion, it can be shown [31] that the multidimensional complex frequency-wavenumber filter response is given by

$$H(f,\vec{k}) = P_S(f,\vec{k})/[P_S(f,\vec{k})+P_N(f,\vec{k})] \tag{3-7}$$

where $P_S$ and $P_N$ are frequency-wavenumber power spectral density functions of the signal and noise, respectively. For the required filter the signal is a single wideband planewave with specified vector velocity

$$\vec{v}_0 = f\vec{k}/|k|^2 \quad . \tag{3-8}$$

The noise is the sum of all other planewaves. Thus, the numerator of (3-7) is zero except when this condition holds and, since the frequency-wavenumber distribution of signal and noise is unknown a priori, the maximum uncertainty condition must be assumed, that is, both the signal and noise are assumed to be flat, and the final filter is

$$H(f,\vec{k}) = \begin{cases} 1, & \vec{k} = \vec{v}f/|\vec{v}|^2 \\ 0, & \text{elsewhere} \end{cases} \tag{3-9}$$

for all f.

The response pattern is sketched in Figure 3.13. The ray emanating from the origin is the locus of points $H(f,\vec{k}) = 1$ and corresponds to points for which $\vec{v}_0 = f\vec{k}/|\vec{k}|^2$. For all other points in the frequency-wavenumber space, $H(f,\vec{k}) = 0$. The resultant filter is a vector velocity bandpass filter. It is optimum under the conditions described

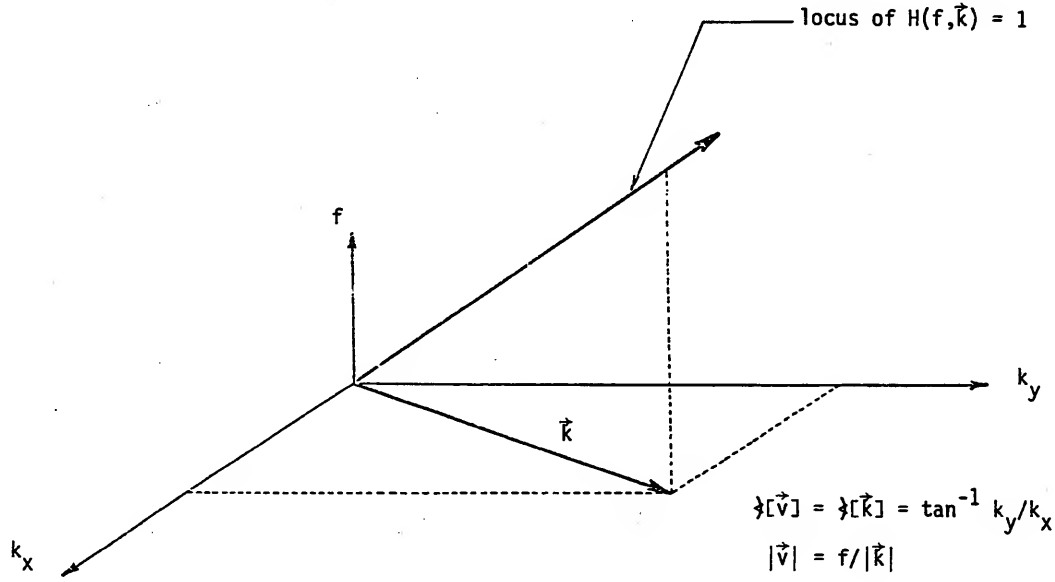



Figure 3.13 Frequency response for one wideband pass filter.

Table 4

Number of operations required for various data array sizes

| Data array size | Multichannel frequency domain filtering | Multidimensional frequency domain filtering | Multidimensional discrete convolution |
|---|---|---|---|
| 4x4x32 | $13 \times 10^3$ | $9 \times 10^3$ | $10^6$ |
| 8x8x32 | $151 \times 10^3$ | $45 \times 10^3$ | $10^7$ |
| 4x4x1024 | $3 \times 10^6$ | $0.4 \times 10^6$ | $10^8$ |
| 8x8x1024 | $4 \times 10^7$ | $0.2 \times 10^7$ | $10^{10}$ |

and its output contains the signal (with no filtering distortion) whose velocity is $\vec{v}_0$ and any interfering planewave (noise) whose velocity is $\vec{v}_0$. In practical terms this means that if two or more waves are within a beamwidth of each other, that is, their vector velocities are approximately equal, the filter is incapable of separating them and simply passes the sum as one composite wave. In all other cases, however, the ideal filter passes only one specified planewave. The remainder of this chapter is concerned with the development of digital filters which approximate the ideal planewave filters.

## Implementation

The digital implementation chosen for this study involves approximating the function of (3-9) by using multidimensional frequency domain specifications by the window method. Before proceeding, it is important to realize that we are approximating an idealized response based on

continuous data of unlimited extent in all dimensions. As an alternative one could reformulate the problem to be one of determining the optimum linear processor which operates on sampled data of fixed extent in all dimensions. The relative merit of these two approaches is an issue aside from the main purpose of maximum-likelihood estimation and is not investigated here. The frequency domain method was chosen because it is consistent with techniques previously discussed and appears efficient from a computational viewpoint.

The principal programming units are divided into four subroutines. The first three units generate the complex frequency vector wavenumber response in the frequency domain. The fourth unit utilizes the specified response and filters a given spatio-temporal series. In addition to providing the capability to pass a single planewave based on its vector velocity, the filters can pass or stop groups of waves based on their bearing, speed or temporal frequency. This generalization is provided to facilitate data analysis.

## Angle Filtering

The purpose of an angle filter is to pass (or stop) all waves coming from a particular direction and stop (or pass) all others. The filtering is to be accomplished independent of the velocity magnitude and bandwidth of the planewaves. The relative ease with which this may be carried out stems from the fact that the vector wavenumber $\vec{k}$ and the vector velocity have the same direction

$$\vec{v} = f\vec{k}/|\vec{k}|^2 \ . \qquad (3\text{-}10)$$

To specify the frequency response all points lying within the beam wedge, as shown in Figure 3.14, where $\theta$ is the specified bearing angle and $\phi$ is the specified beamwidth, are set to unity, and all other points are set to zero.

It was demonstrated in the section on one-dimensional filtering that it is usually advisable to smooth the transition from stop to pass. This same principle applies to any multidimensional filter. Two-dimensional smoothing need only be applied to each $k_x$-$k_y$ plane since the angle filter response is independent of frequency.

## Digital Angle Filter Implementation

Given the values of the complex frequency-wavenumber response only at discrete points of the ideal response, the complete performance of the digital angle filter can be determined in one of two ways. The sampling theorem, as applied to multidimensions, may be invoked to determine the response at every point in the frequency-wavenumber space. Alternatively, one may numerically interpolate the response utilizing multidimensional FFT techniques. The latter technique is applied here as the functions resulting from the mathematical technique do not generally provide a concise overview of the filtering operation. This is particularly true when one is interested in examining the amount of passband ripple and stopband attenuation. This approach is readily recognized as an extension of one-dimensional, nonrecursive digital filtering design.

Subroutine FA accepts as input the desired bearing (look angle) and beamwidth and assumes that a 4 x 4 point spatial array, with 2 cm

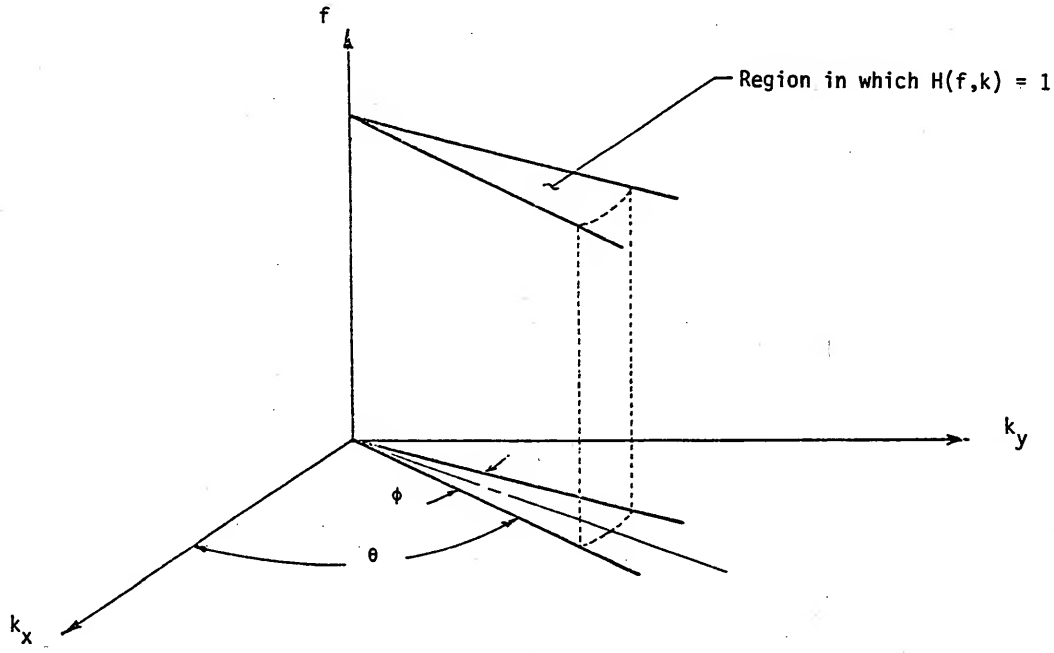



Figure 3.14 Angle filter frequency-wavenumber response.

spacing, is being employed. (Simple program modifications can be incorporated to effect any rectangular array with uniform spacing. The spacing interval in the x and y directions need not be equal.) Based on bearing and beamwidth, FA determines whether each point of the wavenumber response series should be set to zero or unity. If bandstop filtering is selected instead of bandpass, the ones and zeros are complemented. At this point, it is generally advisable to smooth the transition from pass to stop in order to decrease the rippling. The specific array considered in this study, however, is only 4 x 4 points in extent. If any of these points are spent in forming a wider transition band, the resulting filter has an angular resolution which is too coarse to be of much value. Consequently, the smoothing option was not incorporated in FA and, in effect, a rectangular window is employed. After the 4 x 4 array is generated and returned to the calling program an optional 4:1 interpolation of the wavenumber response may be instigated. With this option the user can easily assess the actual expected performance. The process begins by inverse transforming (via FFT techniques) the two-dimensional wavenumber response to the spatial impulse response domain. In a manner similar to the one-dimensional interpolation, refer to Figure 3.3, the array is separated so that zeros may be added to the positive and negative tails of the impulse response. Subroutine EXPAND performs the rather complicated operation for any array size. The expanded array is then forward transformed back to the wavenumber domain and returned to the calling program. In a manner similar to the power spectral estimates (FWNO/PLT3) a line printer output can be specified with a listing of the filter response on a db scale or as a grey tone

contour plot. To demonstrate the angle filter and the effects of angle filtering, and to present a basis for some general observations, the following example is presented.

If FA is called with the bearing and beamwidth set at $45^{o}$ and the complete printed output specified, Figures 3.15 and 3.16 result. Figure 3.15 lists the expanded (interpolated) response array in negative db with respect to a specified point in the passband. To avoid printing excess minus signs, the sign of each number is complemented before printing. Thus the -2's found on the column corresponding to $k_x = 0.21875\ cm^{-1}$ and those found on the row corresponding to $k_y = 0.18750\ cm^{-1}$ indicate a 2 db rise above the specified response (0 db). The region in the first quadrant, enclosed by the two lines superimposed on the printer output, is the beam pattern specified by the calling arguments of FA. If perfect response were possible, all numbers in the region would be zero and those outside would be 90. The lowest level printed is -90 db. Lower levels are limited accordingly. The sixteen circled numbers of Figure 3.15 are the numbers generated initially by FA and returned for actual filtering operation. The remaining numbers are all part of the interpolated response. Figure 3.16 presents the same data in grey tone format. The use of the seven circled numbers will be explained subsequently. In Figure 3.16 the overall response of the chosen angle is concisely summarized. The grey code is explained in Table 3.

From an examination of the interpolated frequency response patterns such as the one presented above, some general observations can be made. The poorest response, in terms of maximum deviation from the ideal

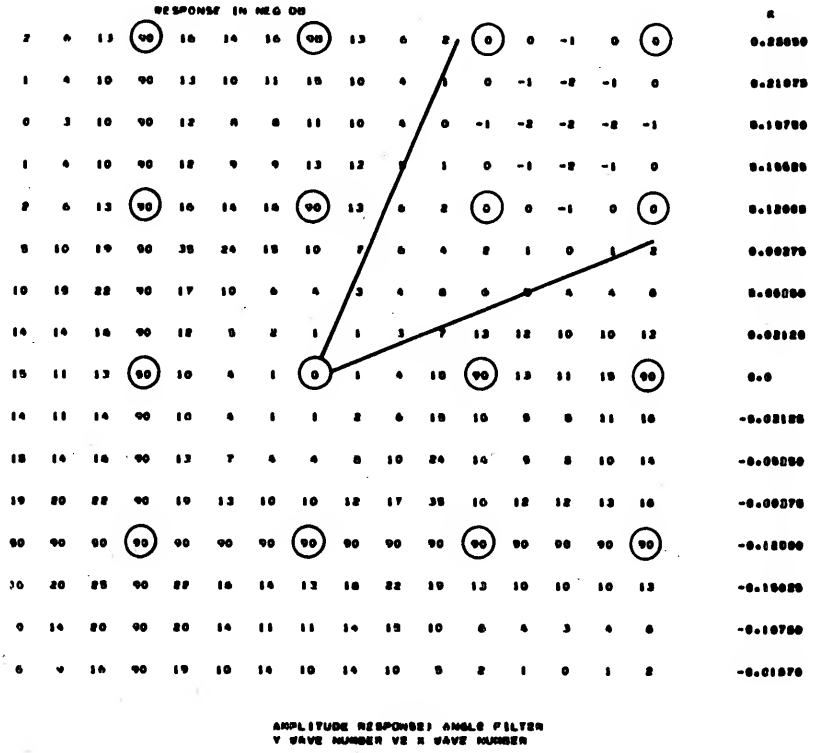



Figure 3.15 Number array output of subroutine FA.
Bearing = 45.0 degrees and beamwidth = 45.0 degrees.

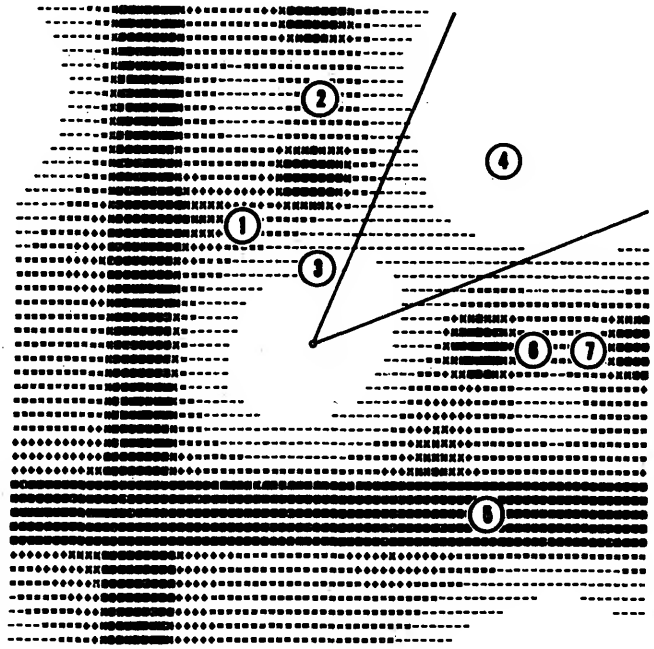



Figure 3.16 Grey tone contour output of subroutine FA.
Bearing = 45.0 degrees and beamwidth = 45.0 degrees.

response, occurs at or near the points farthest removed from the specified response. If the spatial array were doubled in size, the additional points which can specify the response exactly approximately correspond to those points which formerly gave the poorest performance. Significantly improved performance can clearly be expected as the spatial data are allowed to increase. However, as the number of data points increase, so do the required computations. The ideal situation would probably be a combination of more data points with transition bands. Another measure of performance is the angular resolution. For this study the resolution is defined to be $Y^{o}$ at a bearing of $X^{o}$ if the digital angle filter yields zero discrimination between two or more waves whose bearings are in the range [X-Y/2,X+Y/2]. Since we consider only rectangular arrays, which do not exhibit circular symmetry, the resolution is a function of bearing. The approximate resolution for given bearings is given for 4 x 4 and 8 x 8 arrays in Table 5.

As expected, the larger array yields significantly better resolution. The table also demonstrates the degraded performance in the third quadrant when small arrays are used. The latter effect can be completely eliminated by performing sign reversal before transformation as explained in Chapter II.

To demonstrate the use of the angle filter, the seven planewaves which are listed in Table 2 were added and an effort was made to eliminate all but the wave whose direction of propagation is +45.0 degrees (wave number 4). The simplified flow chart of Figure 3.17 illustrates the basic steps to be followed in a typical angle filtering exercise.

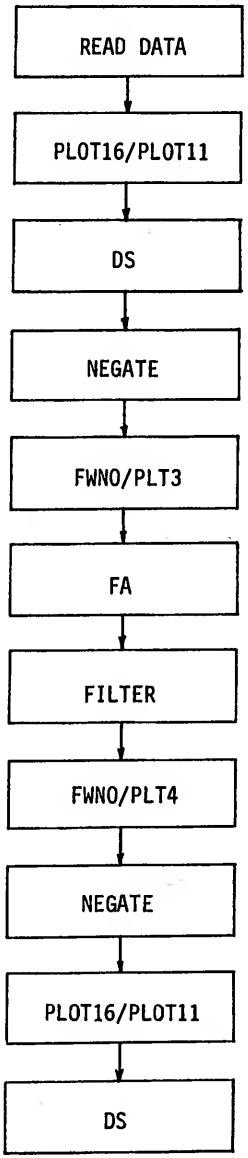



Figure 3.17 Simplified flow chart for angle filtering exercise.

Table 5

Approximate resolution for various bearings

| Bearing (degrees) | Resolution (degrees) | |
|---|---|---|
| | 4 x 4 | 8 x 8 |
| 0 | 54 | 28 |
| 27 | 36 | 14 |
| 45 | 36 | 16 |
| 63 | 36 | 14 |
| 90 | 54 | 28 |
| 117 | 40 | 16 |
| 135 | 40 | 18 |
| 180 | 90 | 36 |
| 225 | 90 | 22 |
| 270 | 90 | 36 |
| 315 | 36 | 18 |
| 333 | 36 | 14 |

The data are obtained from a preliminary processing routine or, as in this example, by seven calls to subroutine PWAVE. PLOT16 may be called to obtain a quick look at the data. For data containing more than one wave, or a significant noise level, PLOT16 does little more than display the dynamic range of the data. PLOT11 plots the waveform received at the origin. Figure 3.18(a) is a reproduction of PLOT11 before angle filtering. If only one channel of data is available,

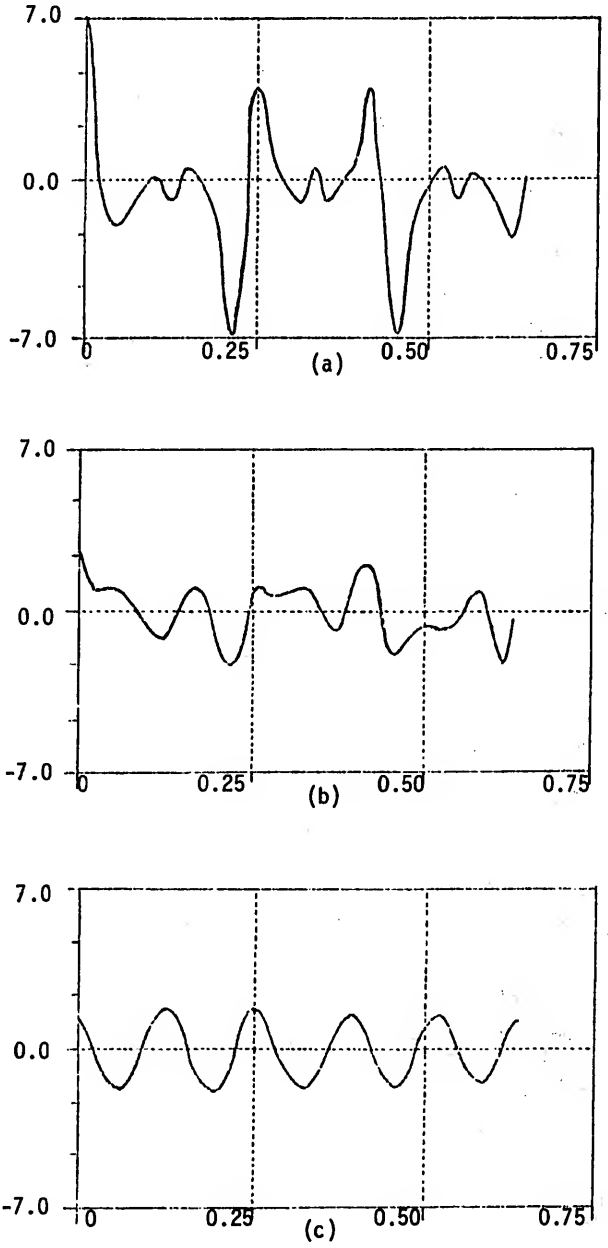



Figure 3.18 Estimated waveform using only one channel. (a) prefilter estimate, (b) post angle filter estimate, and (c) post velocity filter estimate.

this output is the maximum-likelihood estimate. This point is covered in Chapter IV.

As explained in Chapter IV, the delay and sum estimate is optimum (maximum-likelihood, or minimum variance and unbiased) if white noise is present. If interfering planewaves are present, as they are in this example, the spatial white noise condition is not met, but the estimate is then the nonparametric least mean square error (LMSE) estimate [1]. If angle filtering is to be of any value, the desired wave must pass through the filter with little distortion and the unwanted waves must be greatly attenuated. If this condition is met, the postfilter delay and sum estimates should exhibit an improved estimate of wave number 4. The delay and sum estimate of each wave before addition with the other waves is shown as the (a) part of Figures 3.19 to 3.25. This estimate is essentially perfect; no noise or interference is present. After addition, the delay and sum estimates (now least mean square error estimates) are shown in the (b) part of Figures 3.19 to 3.25.

To prepare the spatio-temporal series for discrete transformation, subroutine NEGATE is called. The estimated frequency-wavenumber spectrum was given in the last chapter, Figures 2.13 to 2.29.

For this example, subroutine FA was called with a bearing and beamwidth of +45.0 degrees. The response plots are given in Figures 3.15 and 3.16. The position of the circled numbers of Figure 3.16 corresponds to the positions of the numbered planewaves in the frequency-wavenumber space. From response patterns the effects of the filter can be determined. Wave number 4 should pass as desired with

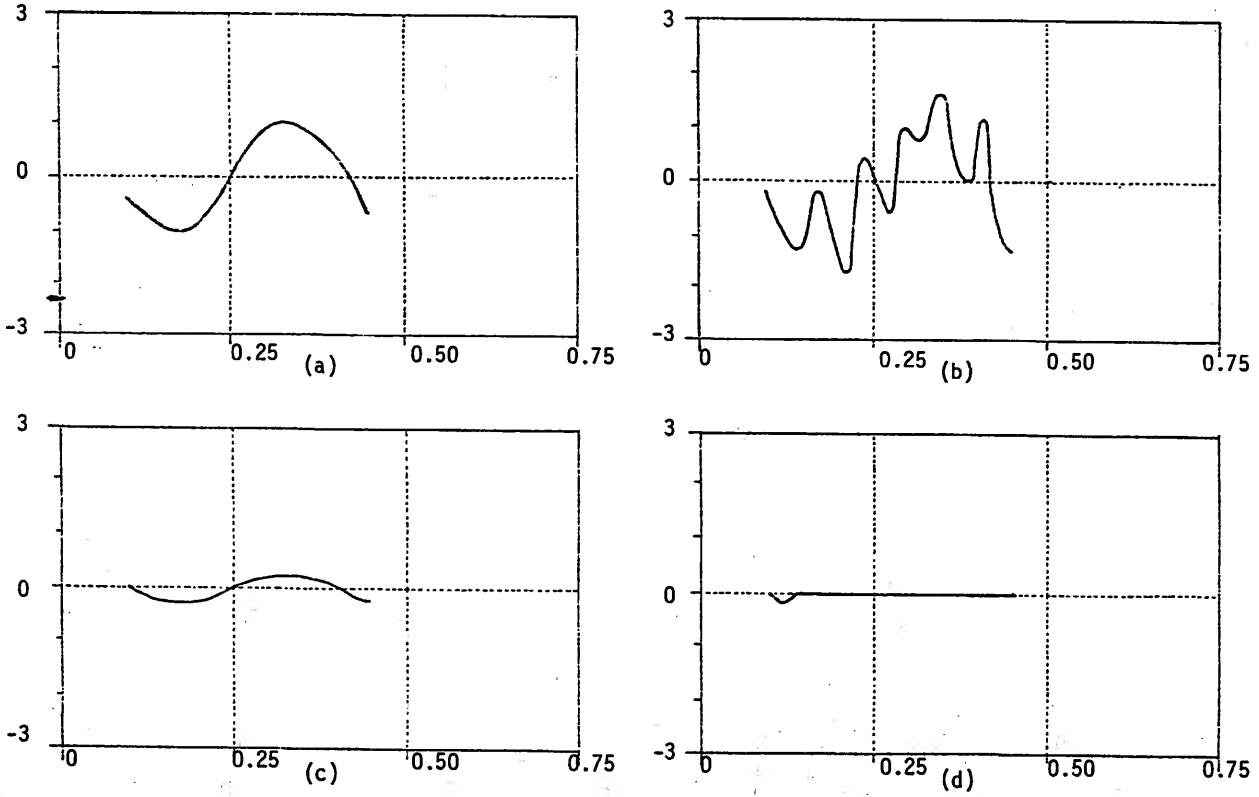



Figure 3.19 Estimates of planewave number 1. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

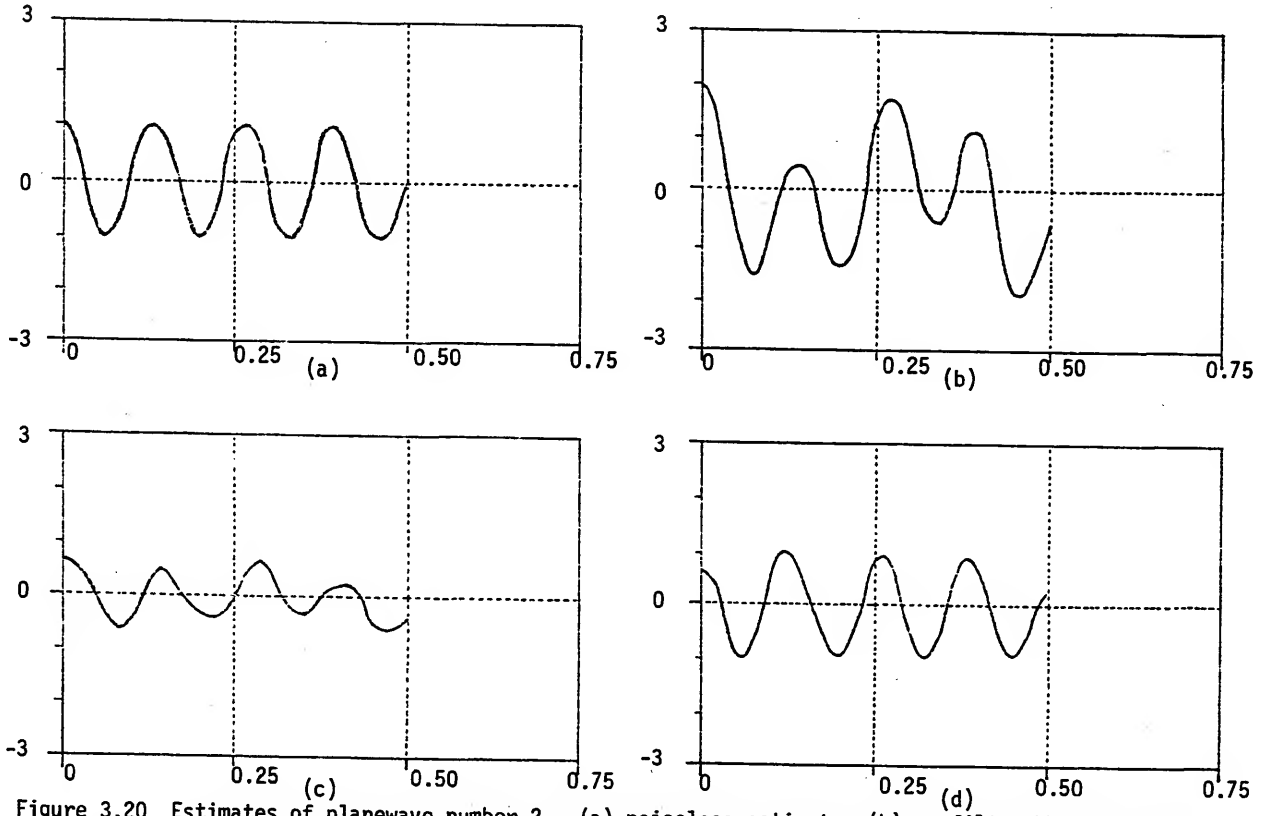



Figure 3.20 Estimates of planewave number 2. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

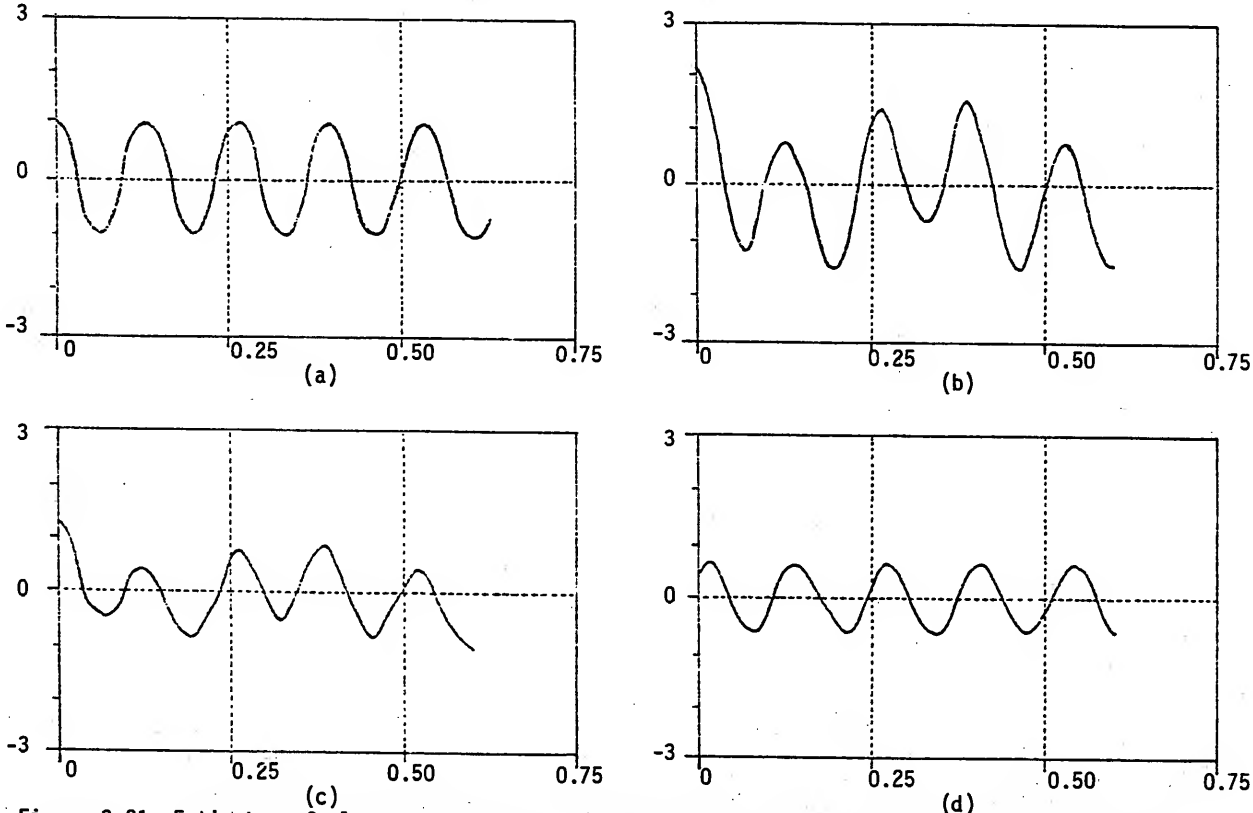



Figure 3.21 Estimates of planewave number 3. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

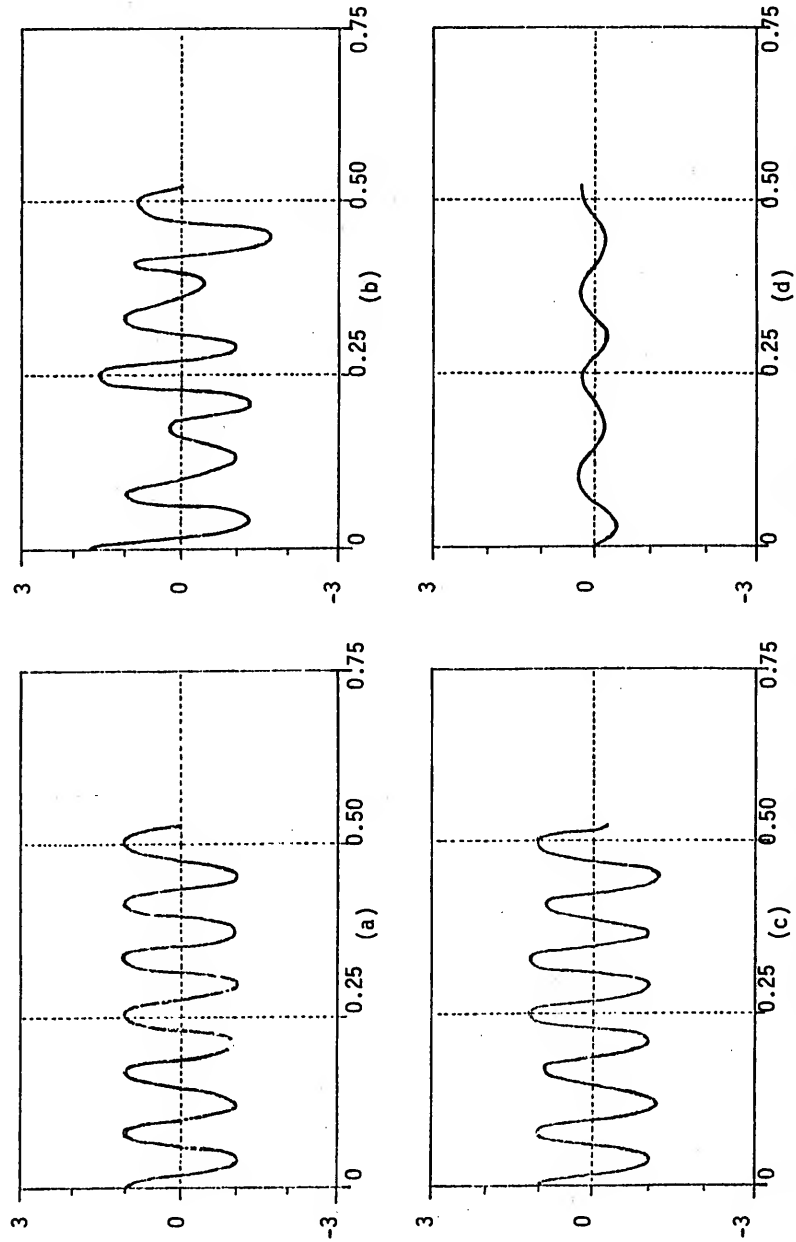



Figure 3.22 Estimates of planewave number 4. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

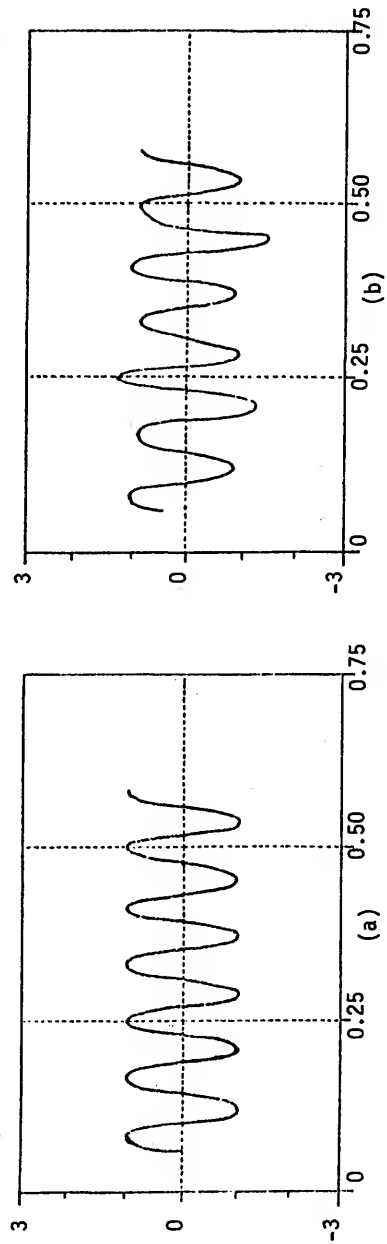



Figure 3.23 Estimates of planewave number 5. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

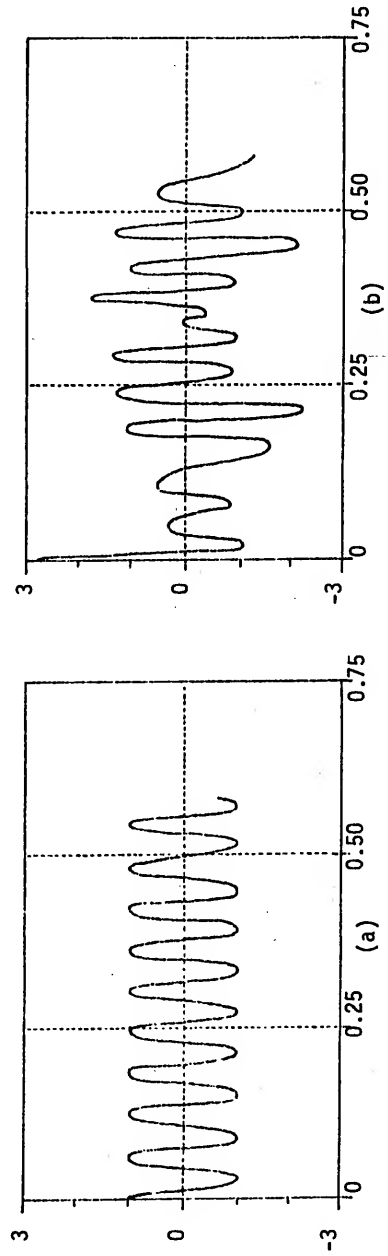



Figure 3.24 Estimates of planewave number 6. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

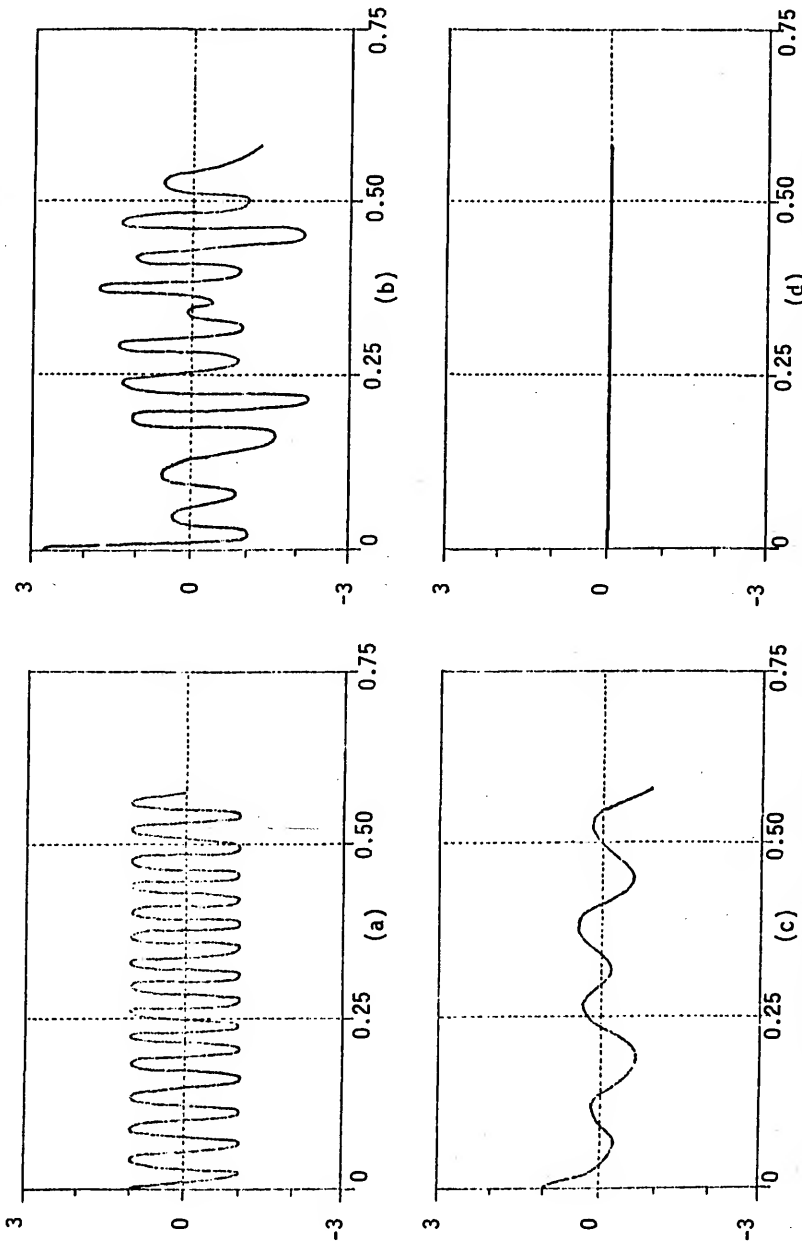



Figure 3.25 Estimates of planewave number 7. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) post angle filter LMSE estimate, and (d) post velocity filter LMSE estimate.

little or no distortion, wave number 5 is practically eliminated, wave number 3 is only slightly reduced, and the remaining four are attenuated by about 10 db.

In order to return to the spatio-temporal domain, NEGATE is called to reorder the axes of the filtered series. If perfect filtering were accomplished, i.e., only one undistorted planewave remained, PLOT16 or PLOT11 would suffice to display the result. The postfilter PLOT11 output after angle filtering is shown in Figure 3.18(b). The prefilter and postfilter estimates should be compared to the noiseless estimate of Figure 3.22(a). As expected, the prefiltered output bears no resemblance to the actual wave and, due to imperfect filtering, the postfilter output only slightly resembles the actual wave.

To obtain the postfilter LMSE estimates, DS is repetitively called for each of the seven waves. The results are given as the (c) part of Figures 3.19 to 3.25. Figure 3.22 demonstrates the advantage of performing the angle filtering before the delay and sum operations. The postfilter LMSE is clearly a superior estimate to the prefilter LMSE estimate. This is a significant result that will generally hold whenever the filter passes the desired signal with negligible distortion and reduces the energy in the composite wave due to the interfering events. Since any measure of improvement must depend on the particular waves being analyzed, and this study is only concerned with the analysis of signals which are unknown a priori, the quantification of improvement factors is not pursued. The postfilter estimates of the remaining six waves do not in general appear to be filtered versions of the corresponding input waves. This is, of course, due to the fact

that the estimates are significantly affected by the presence of the wave which was intentionally passed in addition to any wave which is imperfectly attenuated. The only condition under which the delay and sum estimate is perfect occurs when only one wave is present and it is the wave being estimated. If only one wave was present, and FA was used to stop the wave, the postfilter DS output would be zero on the plotting scale which we have presented. The stopband postfilter estimates are included only to indicate, by comparison with the pre-filter estimates, that the angle filter eliminates, or attenuates, the corresponding input waves.

## Velocity Filtering

Multidimensional digital filtering is commonly used in pattern recognition, but filtering of spatio-temporal functions by this technique has not been reported in the literature. Instead, the problem is formulated directly in terms of multichannel filtering or the problem is formulated in the multidimensional domain and then transformed to the multichannel specifications [30,31]. The reason for this apparent lack of interest stems from the fact that in many applications,such as radar, sonar, and radio astronomy, the velocity of propagation is approximately constant and simple filtering operations such as delay and sum or weighted delay and sum are usually adequate [2]. The only area where the velocity of propagation is important and not constant is seismology. [28] provides an interesting introduction to this subject. The velocity filter introduced in this study differs from others in that the filtering is done directly in the frequency-wavenumber domain.

The purpose of a velocity filter is to pass or stop all planewaves which have a magnitude velocity lying within a specified band. The ideal velocity discrimination is made independent of the bearing and frequency composition of planewaves. The ease with which the filtering operation may be specified in the frequency-wavenumber space is due to the colinearity of $\vec{k}$ and $\vec{v}$.

If it is desired to pass all planewaves with velocity v, such that $v_1 \leq v \leq v_2$, the complex frequency-wavenumber is set to unity in the region between the two right circular cones as shown in Figure 3.26. All other points in the space are set to zero. The fact that all points on the surface of a cone correspond to waves whose velocity is given by the slope is a direct consequence of the definition of wavenumber,

$$\vec{k} = f\vec{v}/|\vec{v}|^2 \quad ; \tag{3-11}$$

therefore, $k = f/v$ and $v = f/k$ in magnitude.

## Digital Velocity Filter Implementation

Given the values of the frequency-wavenumber response at only discrete points of the ideal response, the complete performance of the filter can be determined by the zero-adding process which has been previously discussed. The axial symmetry of the velocity filter permits expansion in a plane passing through the f axis and perpendicular to the $k_x$-$k_y$ plane.

The basic steps to be followed in a typical velocity filtering exercise are the same as those shown for the angle filter in Figure 3.17.

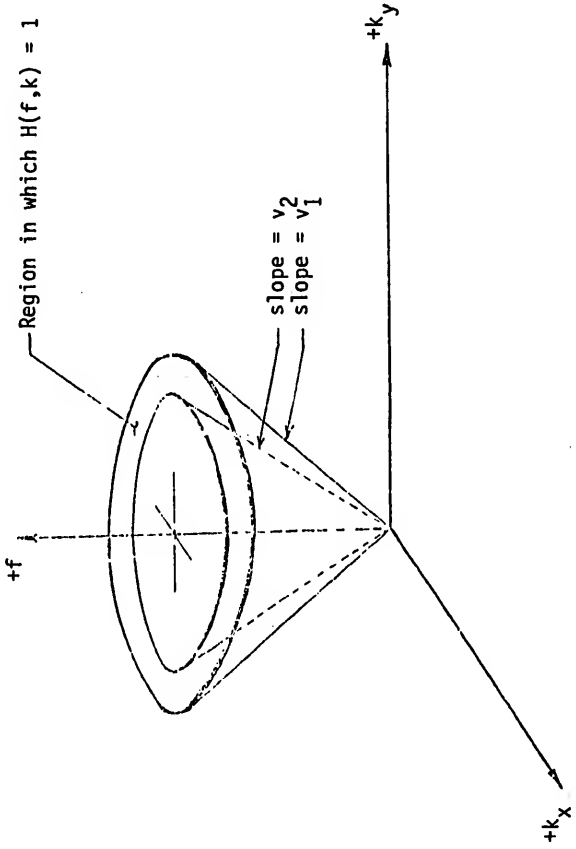



Figure 3.26 Velocity filter frequency-wavenumber response.

The only difference is that FV is called instead of FA. For the interpolated response a 2:1 interpolation is done in the frequency coordinate and an 8:1 interpolation is given for k. With the high velocity transition set to 61.5 cm/sec, the interpolated response is given in Figure 3.27. The circled numbers represent the uninterpolated array which is returned to the calling program for use by subroutine FILTER. The triangular region enclosed by the two lines is a cross-section of the two cones which define the passband. Refer to Figure 3.26. If perfect filtering were possible all numbers inside this region would be zero and those outside would be -90 db. Figure 3.28 provides the grey tone contour output for the same filter specification. The complete performance of the digital angle filter can be obtained by generating this output for filtering specifications of interest.

For the particular 4 x 4 sensor array being considered, the velocity resolution is about 15 cm/sec from approximately 10 cm/sec to 200 cm/sec. Beyond 200 cm/sec discrimination is impossible. With an 8 x 8 sensor array the resolution is near 5 cm/sec from 5 cm/sec to 400 cm/sec.

An option of FV allows the smoothing (Hanning or Hamming) of the transition along the frequency coordinates. Smoothing options in the $\vec{k}$ direction are not provided. These options permit the user to exchange resolution for improved in-band and out-of-band performance. The usefulness of these options depends upon the particular application. For the application considered herein, smoothing is not used so that the best possible resolution would be obtained.

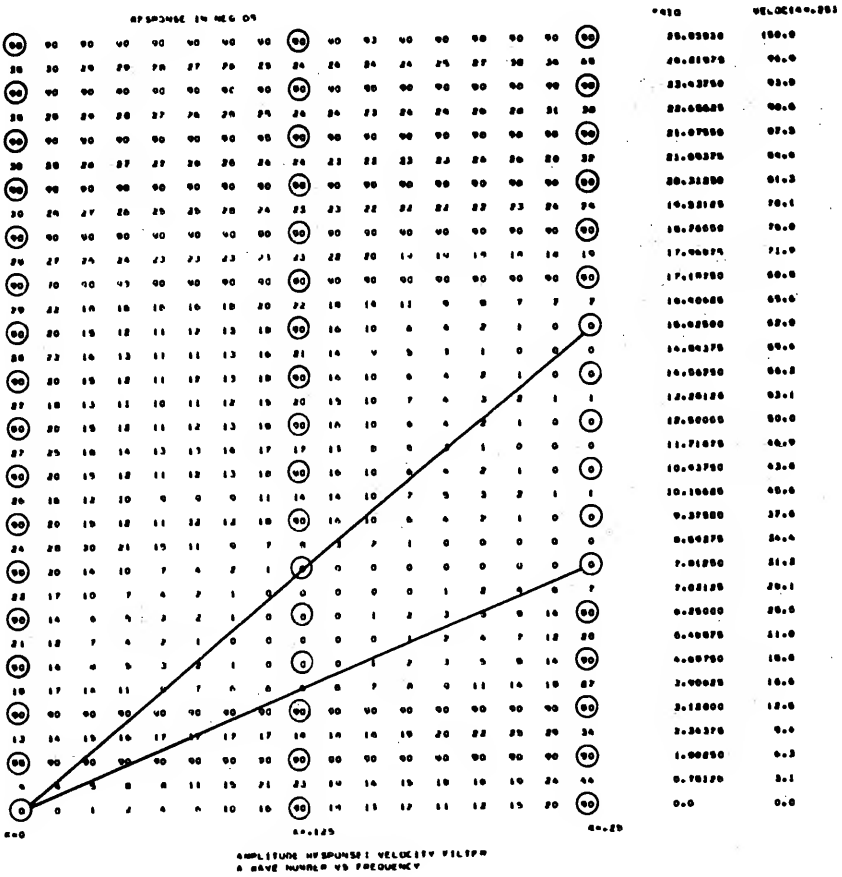


Figure 3.27 Number array output of subroutine FV.
High velocity = 61.5 and low velocity = 31.25.

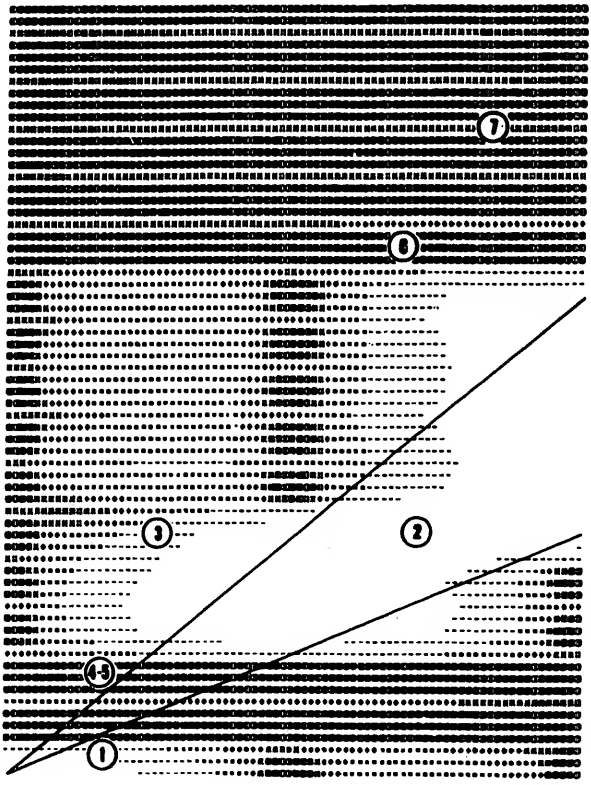



Figure 3.28 Grey tone output of subroutine FV.
High velocity = 61.5 and low velocity = 31.25.

To demonstrate the velocity filter, the seven planewaves of Table 2 were added and an effort was made to eliminate all but wave number 2. The filtering specifications depicted by Figures 3.27 and 3.28 were used. The numbers superimposed on Figure 3.28 correspond to the numbered waves. Waves 4 and 5 are in the same position because they have the same velocity. Waves 2 and 3 obviously have the same center frequency.

The waves of Figures 3.18(a) and 3.18(c) are the one channel prefilter and postfilter estimates of wave number 2. The effectiveness of the velocity filter is quite evident when the estimate of the latter figure is compared with the actual wave as shown in Figure 3.20(a).

The prefilter and postfilter estimates of all seven waves are displayed in the (b) and (d) parts of Figures 3.19 to 3.25. As expected wave numbers 6 and 7 are heavily attenuated. Wave number 3 is only attenuated by approximately 3 db. The final estimate of wave number 2 is almost indistinguishable from the original wave.

## Frequency Filter

A frequency filter is a frequency-wavenumber filter which has the same effect as identical bandpass filters applied to each channel of the sensor data. Strictly speaking, this filter is not required for pre maximum-likelihood velocity filtering. It is provided for preliminary data reduction where one may be able to take advantage of temporal frequency differences between the signal and noise processes. Subroutine FF provides the filtering of the array in a manner similar to that for velocity filtering and angle filtering. The user specifies the high

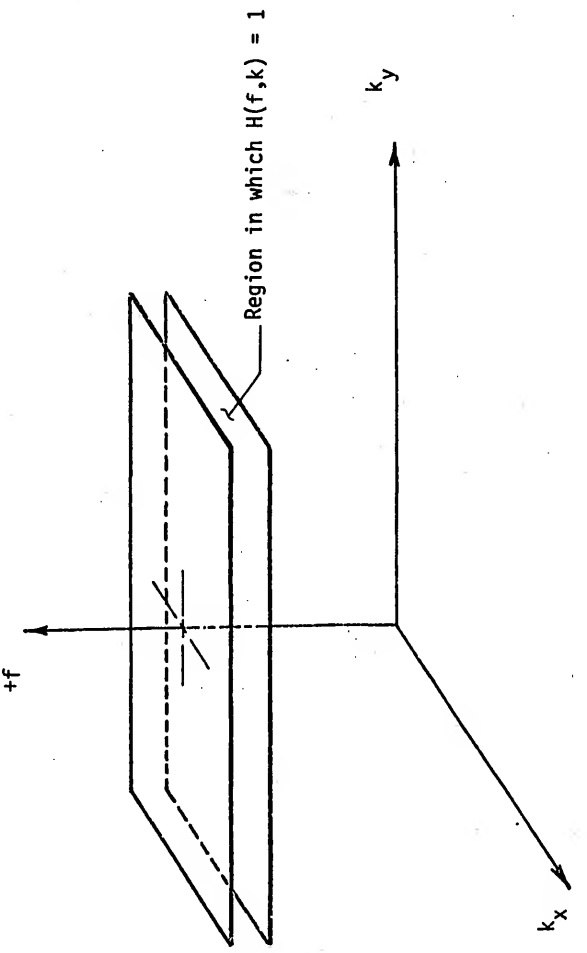



Figure 3.29 Frequency filter frequency-wavenumber response.

frequency transition and the low frequency transition and selects either bandpass or bandstop filtering. All points within the desired band, illustrated in Figure 3.29, are set to unity and all others, zero, for bandpass filtering. For bandstop filtering the ones and zeros are complemented. It is also possible to select Hanning or Hamming smoothing of the transitions.

## Frequency Filter Implementation

To illustrate the use of subroutine FF, wave number 1 of Table 2 was extracted from the sum of the seven waves. The low frequency transition was set at 0.0 Hz and the high frequency transition was set to 6.3 Hz. The interpolated frequency response yields sidelobes no greater than -30 db. Smoothing was not utilized. The results of the filtering are contained in Figure 3.30. The (a) part is the original wave, (b) is the prefilter LMSE estimate, (c) is the one channel postfilter estimate, and (d) is the postfilter LMSE estimate. If we compare (c) and (d) to (a), we see a very effective filtering action. This occurs because this wave is well separated from the other six in the frequency-wavenumber space and we are independently able to specify seventeen filtering points along the filtering axis.

## Summary

If an angle filter is cascaded with a velocity filter, the result is obviously a vector velocity filter. These two filtering specifications are kept distinct in order to maintain a greater degree of flexibility. It should be noted that both filtering operations can be accomplished with only one call to subroutine FILTER. In fact,

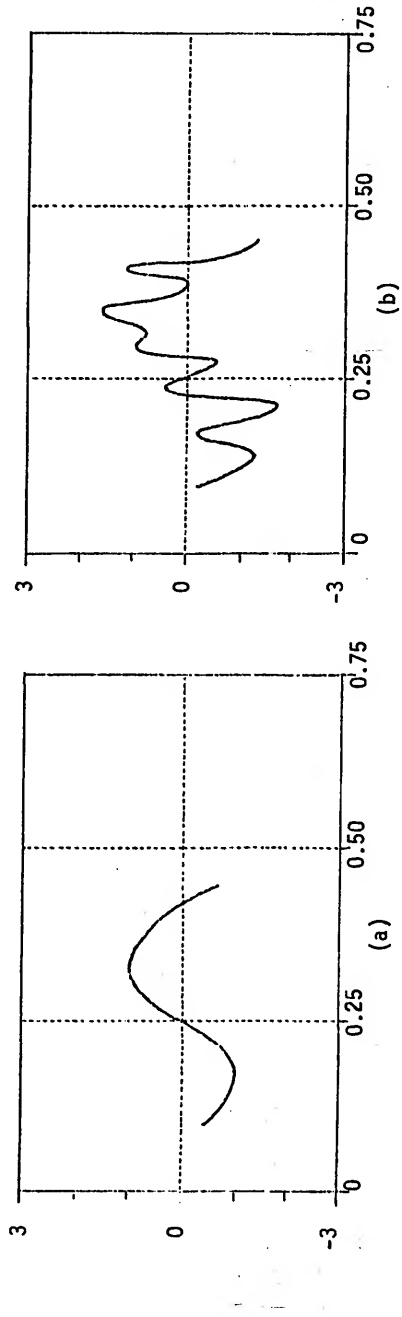



Figure 3.30 Frequency filter example, planewave number 1. (a) noiseless estimate, (b) prefilter LMSE estimate, (c) one channel postfilter estimate, and (d) postfilter LMSE estimate.

if angle filtering is desired without velocity filtering, the Fortran matrix which specifies velocity filtering must be initialized to unity. For details see program FWNO7 of Appendix D.

The complete frequency-wavenumber response of any number of filtering operations can be easily obtained by passing an impulse through the composite filter and examining the spectrum of the result. The subroutines were designed with this function as a specification. Refer to program RESPONSE (Appendix D) for a detailed example.

This chapter has presented a discussion of methods of filtering spatio-temporal signals. This has direct application in the following chapter on maximum-likelihood estimation. It is also felt that this may be of some utilitarian value in the study of spatio-temporal functions.

The principal contributions of this chapter are the very flexible, and computationally efficient, digital spatio-temporal filters.

# CHAPTER IV

# MULTIWAVE MAXIMUM-LIKELIHOOD ESTIMATION

The principal subject of this study is multiwave maximum-likelihood estimation. It will be shown in this chapter how the multiwave problem can be reduced to single wave estimation. The single wave estimation will be thoroughly discussed before proceeding to the general case.

## Single Wave Maximum-likelihood Estimation

This study is motivated by a need to obtain estimates of spatio-temporal signals which are completely unknown a priori. The investigation is limited to planewaves because any stationary spatio-temporal function can be decomposed into a sum (infinite) of Fourier components which are complex planewaves. The maximum-likelihood estimation was chosen because it is the optimum solution in the decision theoretic sense. A very brief explanation of this concept is given in Appendix B. Helstrom [32] and Van Trees [33] are two of many references providing detailed explanation of decision theory from an engineering point of view.

Loosely, the maximum-likelihood estimate of a set of parameters $\underline{\theta} = \{\theta_j\}, j=1,2,\ldots,N$, are those parameters $\hat{\underline{\theta}}$ that maximize the probability that $\underline{\theta}$ has been received with noise. The maximum-likelihood estimator is formally defined in Appendix C. The maximum-likelihood

estimate of single planewaves with known vector velocity can be approached by one of two methods. The first method, the time domain method, is based on the assumption that the data are limited and sampled both spatially and temporally. It is further assumed that the noise is additive and multidimensional Gaussian. The noise is temporally stationary but not necessarily spatially stationary. The samples may be correlated in time and space in accordance with a known covariance matrix. The solution,which is given in [3] and derived in detail in Appendix C, follows directly from the definition of the maximum-likelihood estimate. The solution consists of linear time-varying filters applied to each of the K sensor channels, followed by a summing operation. While this solution is the actual maximum-likelihood solution whenever actual data are considered, it is not pursued in this study because it has been observed [3] that the computer implementation of the solution is quite sensitive to the assumption of noise stationarity and it is computationally inefficient with respect to the frequency domain method which is discussed next.

The frequency domain solution makes the same assumptions as those stated above except that it is assumed that the K sensor time functions are continuous. Transform methods are employed to obtain the solution, and the requirement for realizable filters is relaxed. As a result, one must approximate ideal filters to implement the frequency domain maximum-likelihood estimator. A detailed derivation of this estimator is given by Kelly and Levin [1]. The derivation is quite lengthy, but common to decision theoretic problems. For brevity the derivation is not repeated herein. The remainder of this section is concerned,

instead, with examining the maximum-likelihood estimator and its errors under various conditions.

The maximum-likelihood estimate of a single planewave in the frequency domain is [1]

$$\hat{S}(f) = \frac{\sum_{k,\ell=1}^{K} \epsilon_{k\ell}(f)\, e^{i2\pi\vec{\alpha}\cdot\vec{r}_\ell}\, X_\ell(f)}{\sum_{k,\ell=1}^{K} \epsilon_{k\ell}(f)} \tag{4-1}$$

where $\hat{S}(f)$ is the Fourier transform of the planewave $\hat{s}(t)$ propagating with inverse phase velocity $\vec{\alpha}$.

$$\hat{S}(f) = F\{\hat{s}(t)\}$$

and

$$\epsilon_{k\ell}(f) = Q_{k\ell}(f)\, e^{i2\pi\vec{\alpha}\cdot(\vec{r}_k - \vec{r}_\ell)} \tag{4-2}$$

where $Q_{k\ell}(f)$ is an element of the inverse matrix of the cross-power spectral matrix

$$[Q_{k\ell}(f)] = [P_{k\ell}(f)]^{-1}$$

and $X_\ell(f)$ is the Fourier transform of the waveform received at the $\ell$-th of the K sensors whose vector position is $\vec{r}_\ell$.

$$X_\ell(f) = F\{X_\ell(t)\} \quad .$$

If we define

$$\Delta(f) \triangleq \sum_{k,\ell=1}^{K} \epsilon_{k\ell}(f)$$

and expand (4-1)

$$\hat{S}(f) = 1/\Delta(f) \sum_{k=1}^{K} \sum_{\ell=1}^{K} Q_{k\ell}(f)\, e^{i2\pi\vec{\alpha}\cdot\vec{r}_k}\, e^{-i2\pi\vec{\alpha}\cdot\vec{r}_\ell}\, e^{i2\pi\vec{\alpha}\cdot\vec{r}_\ell}\, X_\ell(f)$$

which can be rearranged and written as

$$\hat{S}(f) = \{ \sum_{k=1}^{K} [X_k(f)] [\sum_{\ell=1}^{K} Q_{\ell k}(f)\, e^{i2\pi\alpha\cdot r_\ell}] \} \{1/\Delta(f)\} \quad .$$

With the delay element $e^{i2\pi\vec{\alpha}\cdot\vec{r}_\ell}$ written as $\tau_\ell$ the block diagram of the general maximum-likelihood estimator is given in Figure 4.1. The term array prewhitener is adapted because these elements are not necessary if the noise is white. The term beam-former is borrowed from radar terminology. It is recognized as a delay and sum processor which is a least mean square error estimator. This point is covered in Appendix A. The last stage of the estimator is called an inverse filter because it has the effect of correcting or inverting the distortion caused by the prewhitening filters. It is interesting to note that if the inverse filter is suitably redefined the maximum-likelihood estimator becomes the optimum spatio-temporal detector examined by Childers and Reed [34].

If there is only one sensor, the array prewhitener reduces to a single element

$$G(f) = 1/P(f)$$

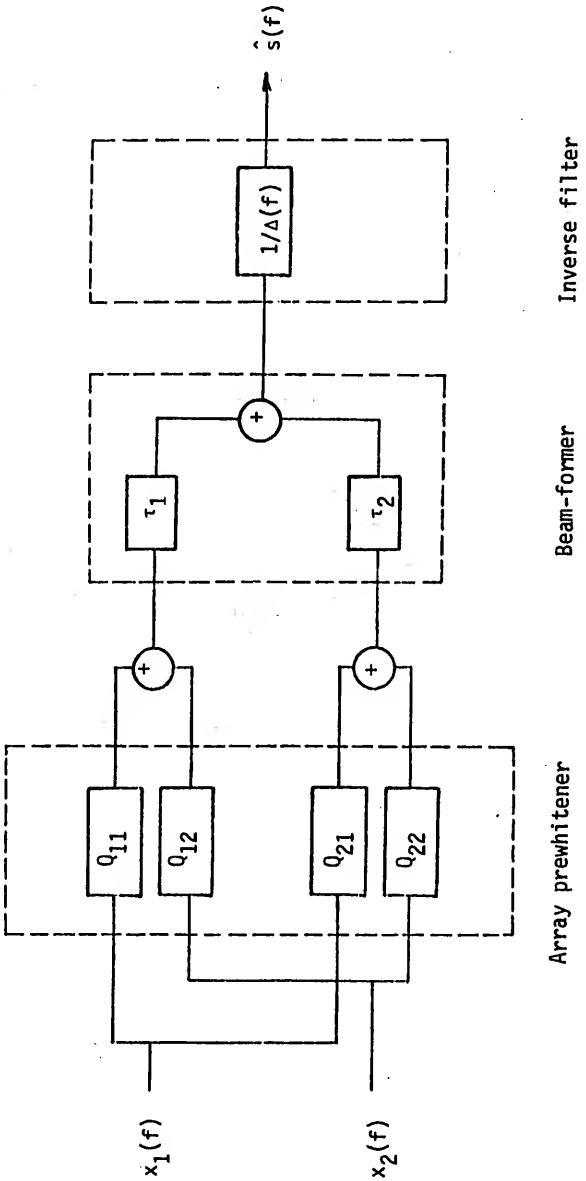



Figure 4.1 General maximum-likelihood estimator, K=2.

where P(f) is the power spectrum of the noise on the given channel. The inverse filter is then given by

$$1/Q(f) = P(f)$$

and the maximum-likelihood estimate is simply the received signal suitably delayed. It is shown later that the general maximum-likelihood estimator does not distort the signal. Thus, with only one received waveform, additional filtering is not possible.

Returning to the general case of K sensors, we now assume that the off-diagonal terms of the cross-power spectral matrix are zero. In this case the block diagram of Figure 4.2 results. If the channel noise is white the filtering functions become weighting constants and an optimum weighted delay and sum procedure results. If the off-diagonal terms are zero and each channel has a common spectral density P(f), the pre-whitening and inverse operations cancel. The result is diagrammed in Figure 4.3 which is identical to Figure A.1. Thus, the conditions under which the least mean square error estimate is the maximum-likelihood estimate are established.

To reduce the general solution to a form amiable to numerical calculations (4-1) is rewritten in the form

$$\hat{S}(f) = \sum_{\ell=1}^{K} \{ \left[ \frac{\sum_{k=1}^{K} \epsilon_{k\ell}(f)}{\Delta(f)} \right] \left[ X_\ell(f) e^{i2\pi \vec{\alpha} \cdot \vec{r}_\ell} \right] \} \quad .$$

Let

$$A_\ell \triangleq \sum_{k=1}^{K} \epsilon_{k\ell}(f) \; / \; \Delta(f)$$

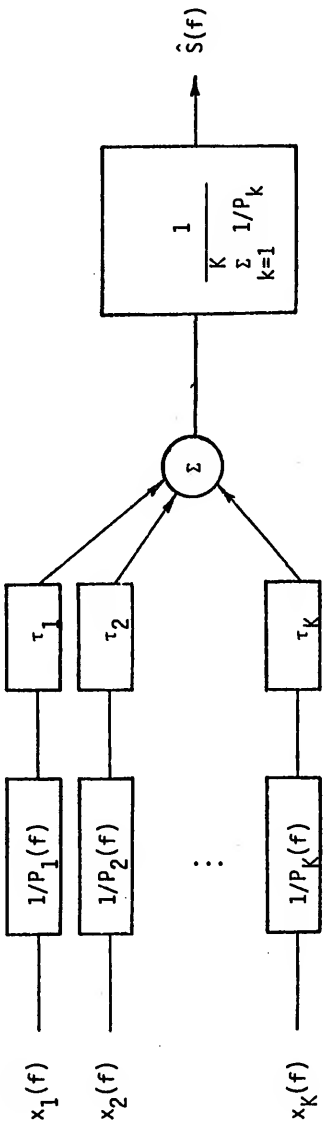



Figure 4.2 Maximum-likelihood estimator with zero cross-power.

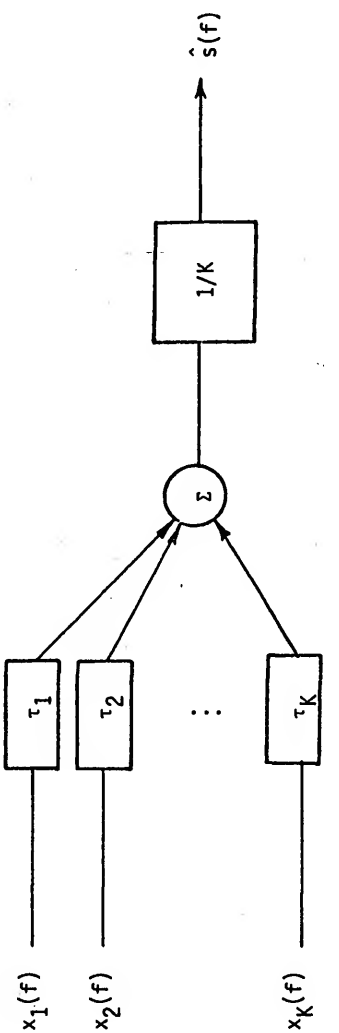



Figure 4.3 Optimum maximum-likelihood estimator for noise independence between sensors with common spectral density.

then

$$\hat{S}(f) = \sum_{\ell=1}^{K} X_\ell(f) e^{i2\pi \vec{\alpha}\cdot\vec{r}_\ell} A_\ell(f) \quad . \tag{4-3}$$

This solution is diagrammed in Figure 4.4. This is the form used by Capon [3] except that $\vec{\alpha}$ is negated to conform with the definition used in [1]. Equation (4-3) is the equation used later in this chapter for the implementation of the maximum-likelihood estimator.

The basic purpose of the maximum-likelihood estimator is to estimate a single planewave immersed in noise in an optimum manner. The planewave in question may be monochromatic or wideband with respect to temporal frequency. If the noise is spatially stationary and the wave being estimated is monochromatic, the estimate is Capon's [20] high resolution estimate. Thus, the high resolution spectral estimates are optimum in the maximum-likelihood sense. If the field is not spatially stationary, the frequency-wavenumber spectrum is not defined.

In the remainder of this section the errors in maximum-likelihood estimates will be briefly explained. Complete analysis can be found in the references. In discussing the errors in the maximum-likelihood estimate, it is important to realize that we are referring to the errors in the ideal estimates. The additional error incurred when the ideal filter is approximated is discussed in the next section.

To determine the bias of the estimate, substitute

$$X_\ell(f) e^{i2\pi \vec{\alpha}\cdot\vec{r}_\ell} = S(f) + N_\ell(f)$$

into (4-1) where $N_\ell(f)$ is the noise sample received at the $\ell$-th sensor.

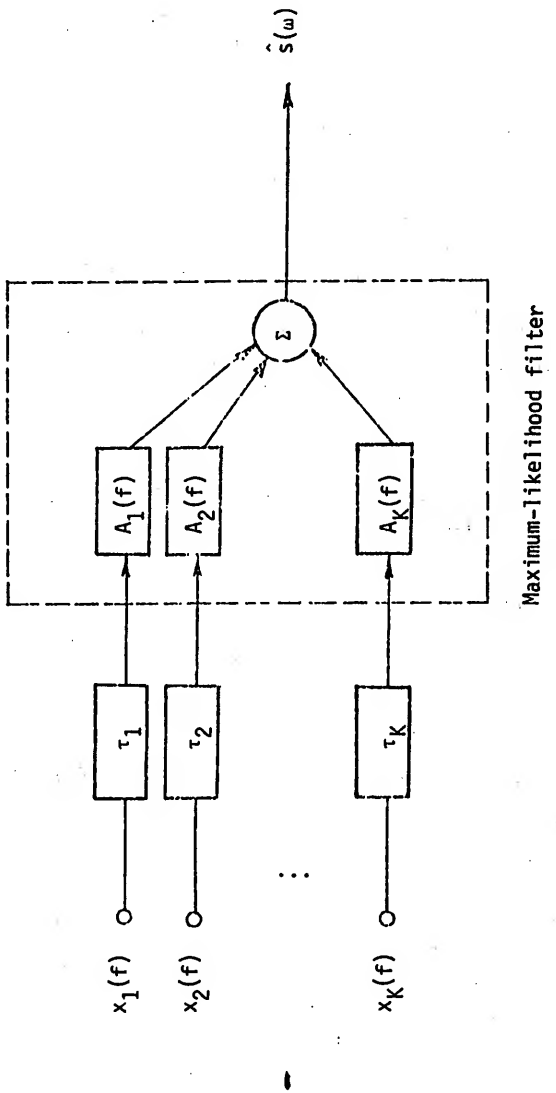



Figure 4.4 Maximum-likelihood estimator form suitable for numerical calculations.

After taking the expectation the result is

$$E[\hat{S}(f)] = E[S(f)] + \frac{\sum_{k,\ell=1}^{K} \epsilon \, E[N_\ell]}{\sum_{k,\ell=1}^{K} \epsilon_{k\ell}} \quad .$$

If the noise has zero mean

$$E[\hat{S}(f)] = E[S(f)]$$

and the estimate is unbiased, the transformed estimate $\hat{s}(t)$ will also be unbiased.

If the estimate is made in the presence of zero noise we have

$$\hat{S}(f) = \frac{\sum_{k,\ell} \epsilon_{k\ell} S(f)}{\sum_{k,\ell} \epsilon_{k\ell}}$$

$$\hat{S}(f) = S(f) \quad .$$

This proves that the maximum-likelihood estimator is distortionless. It should be noted that a vector velocity bandstop filter followed by a least mean square error estimator also has this property.

For the general case where the noise is correlated and spatially nonstationary, the spectral density of the added noise form is [1]

$$1/\Delta(f) = 1 \Big/ \sum_{k=1}^{K} \sum_{\ell=1}^{K} Q_k \, e^{i2\pi\vec{\alpha}(\vec{r}_k - \vec{r}_\ell)} \quad .$$

This can be derived in a manner similar to the derivation of the error in the least mean square error estimate (Appendix A). If the noise is spatially independent with common spectral density P(f) at each sensor

$$1/\Delta(f) = 1/[\sum_{k=1} 1/P(f)] = [1/K][P(f)] \quad .$$

The ratio of total output to input power is

$$\frac{P_{out}}{P_{in}} = \frac{1/K\ P(f)}{K\ P(f)} = 1/K^2 \quad .$$

This result is, as expected, identical to the improvement given by the least mean square error estimator. If the noise is due entirely to interfering planewaves, the noise in the output of the ideal maximum-likelihood estimate is reduced to zero as long as the noise waves have different vector velocity. This property is frequently referred to as superdirectivity [35,36] or infinite sidelobe suppression, singular detection, etc. [1,3,4,5,29]. The basic operation consists of using the given noise statistics to estimate each noise source and then subtract these estimates from the estimate of the signal planewave. Neither the least mean square error nor the weighted delay and sum processors possess this property. Unquestionably, this property is the most significant property of the maximum-likelihood estimator.

The foregoing discussion strictly applies only if the noise can be modeled as a multidimensional Gaussian process. However, it has been shown by Kelly [37] and Capon [3] that if the estimator is designed

under the Gaussian assumption, but, in fact, the noise process is not Gaussian, the resultant estimator still provides an unbiased and minimum variance estimate.

Before proceeding to the multiwave problem it is important to realize that we have only discussed, thus far, the problem of estimating the waveshape s(t). In any practical application one must also solve the detection problem, i.e., determine if a wave is present. If a wave is present its vector velocity can be formulated (estimated) in either a least mean square error or maximum-likelihood sense [1]. Neither of solutions is in closed form, unfortunately, and it is necessary to search the expected domain of the parameter values for the maximum value of integral equations. The dctection problem may be solved by a strategy known as maximum-likelihood detection [38]. In essence, this method first finds the maximum-likelihood estimate of a signal and then compares the likelihood ratio to the preset decision level. These formal solutions are not pursued in this study in order to limit the scope of the study and also because their implementation does not appear to be particularly computationally efficient. Instead, the frequency-wavenumber power spectrum will be used to detect planewaves and to estimate their vector velocity. This technique is demonstrated in Chapters II and V.

## Multiwave Maximum-likelihood Estimation

If the signal consists of a finite sum of J planewaves which overlap in time, added to noise which consists of interfering planewaves plus channel noise which does not exhibit spatial stationarity, we have

the general multiwave estimation problem. If the noise is limited to be white both spatially and temporally, and the term signal is modified to include all planewaves whose total number J is known, the solution to the problem is given by Schweppe [5]. An implementation of his results is given in [6] for the case of two planewaves. As one might expect, the solution requires a determination of the vector velocities of each of the J planewaves. This requires a J-dimensional search similar to the one-dimensional search required for the one wave estimation problem.

This study is concerned with the solution of the general problem with an arbitrary background noise. In any practical case one must be able to detect the number of planewaves J, their vector velocities $\vec{v}$, and their functional form s(t). The theoretical solution is valid for the general case, but for implementation, a modified solution is utilized which is only acceptable when the SNR is fairly high. The general maximum-likelihood estimator is given in Figure 4.5. In simplest terms it is a cascade of a frequency-wavenumber filter which reduces the multiwave problem to a single wave problem followed by a single wave maximum-likelihood estimator. The frequency-wavenumber filters which can accomplish this task are discussed in Chapter III. A single wave maximum-likelihood estimator for a given vector velocity has been developed [3]. To determine the number of waves and their vector velocities, one must scan the frequency-wavenumber space and repetitively determine if a wave is present or not. Since the wave-shape is unknown a priori a maximum-likelihood detector is used. The search logic could be a computer algorithm which is designed to find

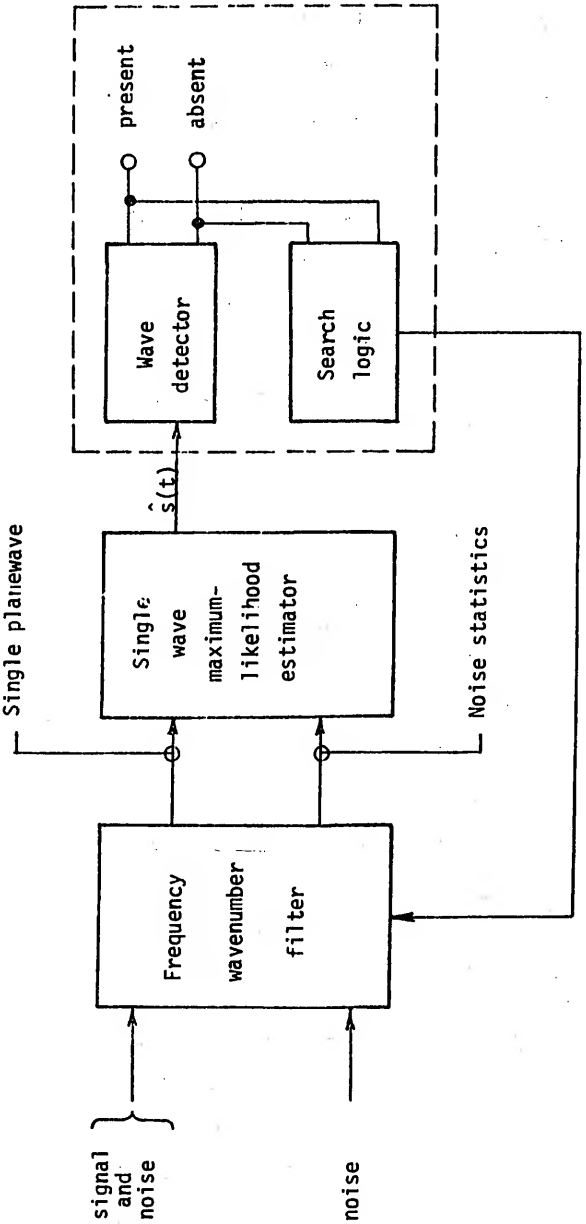



Figure 4.5 General multiwave maximum-likelihood estimator.

the extrema of a function of several variables. Director [39] has given a survey of the techniques which have proven to be of practical value. If the noise level is high, one may alternatively utilize stochastic approximation methods [40,41,42].

The elements enclosed by the broken lines in Figure 4.5, i.e., those elements that are, in effect, responsible for the detection and determination of the vector velocities, will require very sophisticated and elaborate designs which are not attempted in this study. Instead, the use of the frequency-wavenumber power spectral density function is advanced. If the SNR is sufficiently high the local maxima of the PSD can be attributed to the presence of signal planewaves. The vector velocities are readily determined from the wavenumber and frequency. The usefulness of this technique is obviously limited, but realization is very efficient from a computational viewpoint.

The principal contribution offered by this solution is that we have reduced the rather complex J-dimensional problem to a one-dimensional problem. After the spatio-temporal function is passed through the frequency-wavenumber filter, there can be no more than one wave present regardless of whether it is signal or noise. If it is of noise origin the maximum-likelihood estimator effectively eliminates the planewave. If the planewave is of signal origin the ideal maximum-likelihood estimator will pass the wave without distortion. Another important aspect of this system stems from the fact that if the background noise is essentially dominated by planewaves as opposed to channel noise, the maximum-likelihood estimator becomes a simple delay and sum device. Under this condition approximately a 1000:1 increase in computational speed can be realized by substituting the simpler operation.

## Implementation

To implement the general maximum-likelihood estimator, the only additional programming required is the single wave maximum-likelihood estimator. The other components have been presented in the previous chapters. Subroutine ML approximates all of the required operations of a maximum-likelihood estimator as expressed by (4-3). In addition an inverse transform is performed to estimate $\hat{s}(t)$. Capon [3] is the principal reference. Figure 4.6 is a simplified flow chart for ML. Using Figure 4.6 as a guide, the operation of ML will now be examined. The input to ML consists of spatio-temporal functions to represent the signal plus noise and the noise alone. The vector velocity of the wave to be estimated is also required.

The first (left) column of operations aligns the set of received signals in the time domain and transforms the results to the frequency domain. The alignment is accomplished by calling subroutine DELAY with the velocity magnitude argument negated. Each individual time series is tapered, via subroutine TAPER, in order to improve the discrete approximation of the Fourier transforms. Each time series is then padded with an equal number of zeros. This step avoids circular convolution effects when the final filtering is accomplished. Lastly, subroutine HARM is used to one-dimensional transform from the time domain to the frequency domain.

The operations of the second and third columns form an estimate of the set of complex filters $\{A_{\ell}(f)\}$. The transform of each of the noise time series is taken in a manner identical to the signal plus noise. Before the cross-power spectral matrix is formed, the dynamic

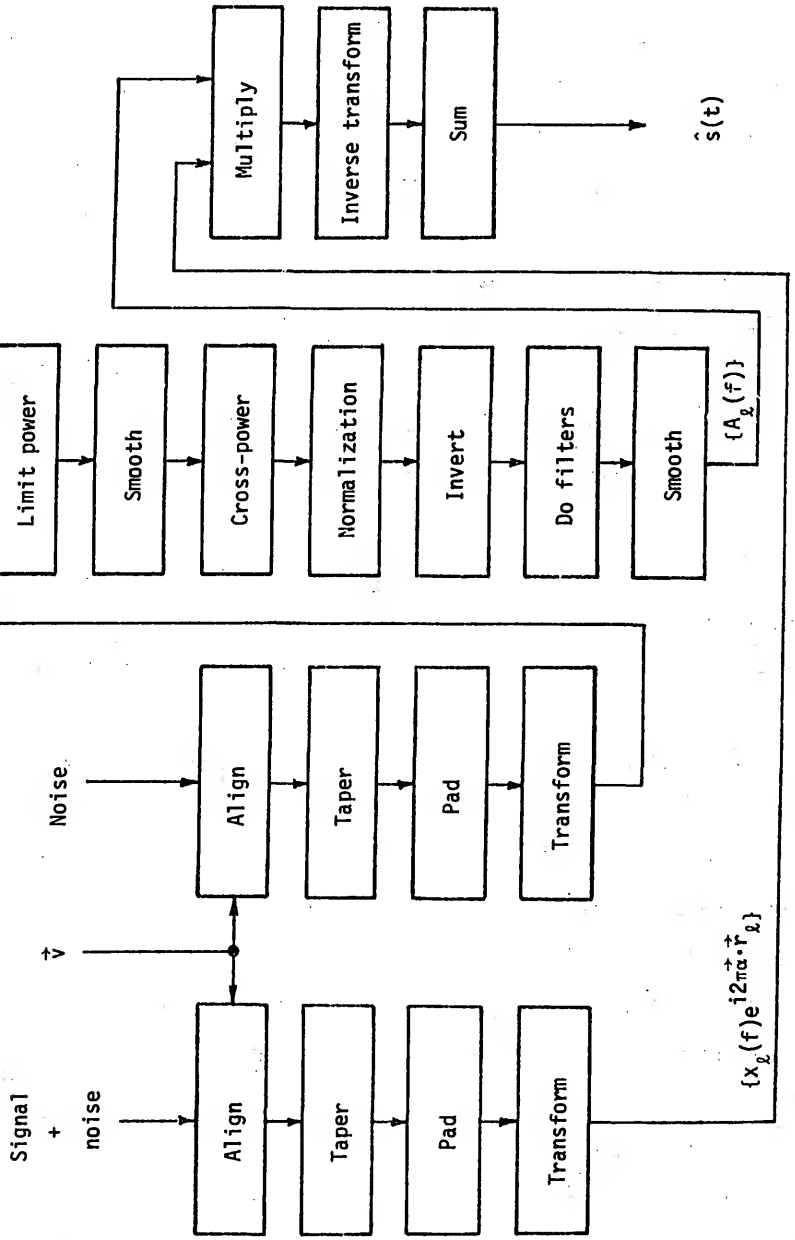



Figure 4.6 Simplified flow chart for maximum-likelihood operation.

range of the transform is limited to a magnitude of 30 db. This step is necessary to avoid excessive error in the subsequent matrix inversion and it appears to produce better approximations of the ideal filters.

Smoothing is accomplished via subroutine HAMN to enhance the statistical reliability. The cross-power spectral density matrix is formed by multiplying the transform at one location by the conjugate of the transform at another. Since the matrix is symmetrical, the number of multiplications is reduced accordingly. The normalization is performed by dividing $P_{jk}$ by $(P_{jj}P_{kk})^{\frac{1}{2}}$. This process compensates for any unequal channel gain settings and provides additional conditioning to insure that the matrix inversion is successful.

The matrix inversion is done using the standard Gauss-Jordan method. For the examples which follow, the order of the matrices is sixteen. This necessitates the use of special precautions in order to insure that the accumulated round-off error does not seriously affect the results. The dynamic range limitation and normalization are the two processes which were found to condition the data so that the matrices could be successfully inverted. Without preliminary operations, the matrix could not be inverted, even when calculations were done in quadruple precision (35 decimal digits).

The last two steps of the third column form the $\{A_{\ell}(f)\}$ from the inverted cross-power spectral density matrix. The smoothing is added in an effort to improve the approximation to the ideal filters.

The remaining steps consist of multiplying the frequency response with the transformed input, inverse transforming, summing, and returning the maximum-likelihood estimate $\hat{s}(t)$. The summing is done

in the time domain, rather than the frequency domain, for programming convenience. It is accomplished by calling DS with a velocity greater than 990 cm/sec.

At this point difficulties in forming acceptable maximum-likelihood estimates should be apparent. One must first estimate the cross-power spectral density matrix. In general, these estimates tend to be unreliable if they are based upon very limited data or if the data tend to be nonstationary. Unfortunately, for some applications, this condition does exist. If the power matrix estimate, and its inverse, are found to be acceptable we still have the problem of approximating ideal frequency response functions. Unlike bandpass filters the ideal response is entirely dependent upon the data of the particular application. For this reason it is essentially impossible to optimize the approximation of the ideal response. Instead, we simply perform a smoothing operation which would decrease any ringing caused by abrupt discontinuities. By contrast, the least mean square error estimator simply changes the indexing of a set of time series and performs one summing operation. Obviously, greater discrepancies must be expected between the theoretical and actual performance for the maximum-likelihood estimator as compared to the least mean square error estimator.

## Examples

In this section we consider three examples. The first two illustrate the use of the single maximum-likelihood estimator. The third is a multiwave estimator.

The first example demonstrates the effectiveness of the maximum-likelihood estimator in reducing the effect of an interfering planewave. The standard of comparison is the LMSE estimate. Two planewaves are used. The signal planewave is a positive 5 Hz sine wave with vector velocity of 25 cm/sec at 30 degrees. The interfering planewave is a negative 5 Hz sine wave with vector velocity of 25 cm/sec at 0 degrees. The spatio-temporal dimensions are the same as those used in previous chapters. To each of the sixteen time series that represent the interfering planewave, independent Gaussian noise is added with an rms value of unity. This is done to simulate channel noise. The amplitude of the signal wave is fixed at 10.0; the amplitude of the interfering wave varies. The signal-to-interference ratio (SIR) is the ratio of the absolute value of the amplitudes. Figure 4.7 depicts the condition for SIR = 2. In Figure 4.7(a) the one channel maximum-likelihood estimate, or simply the waveform received at the origin, is shown. Figure 4.7(b) is the LMSE estimate, and (c) is the maximum-likelihood estimate. Figure 4.8 presents the same waveforms for SIR = 1. At this SIR a very slight improvement is observed in the maximum-likelihood estimate over the LMSE estimate. Figures 4.9 and 4.10 display the results for SIR of 0.5 and 0.1, respectively. As the interfering signal becomes greater, the improvement realized by using the maximum-likelihood estimate, as opposed to the LMSE estimate, becomes more apparent. Several runs were made for lower SIR's and it was obvious that both estimates were greatly influenced by the interfering signal.

In the second example the effect of the independent channel noise is examined. The same two planewaves that were used for the first

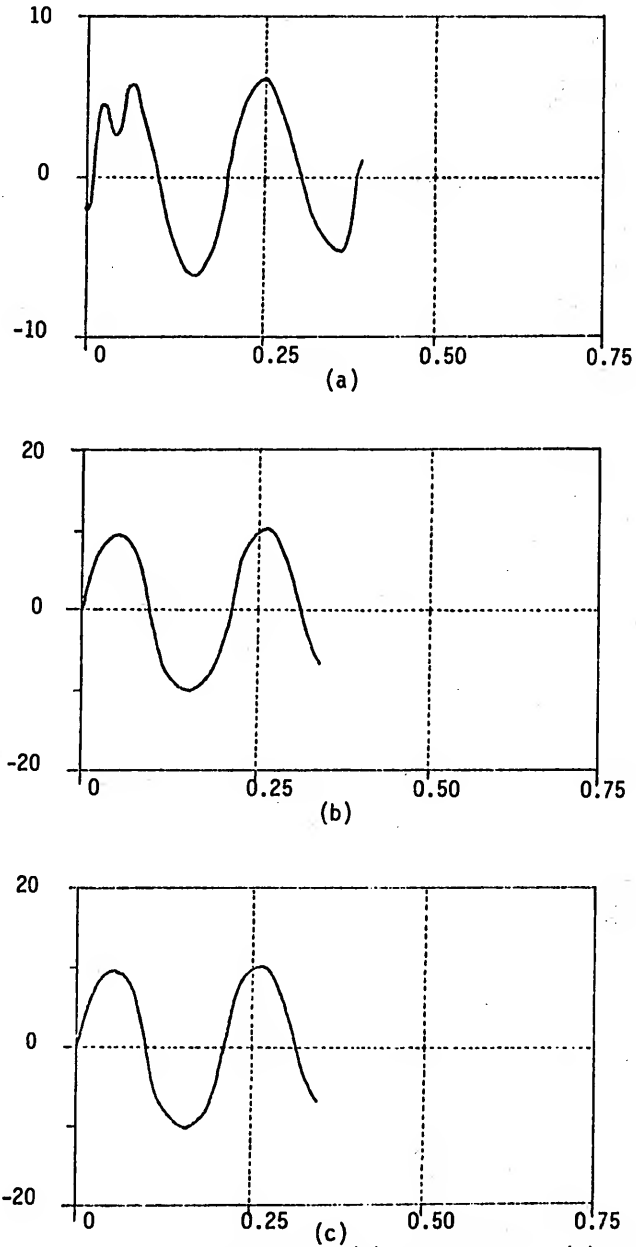



Figure 4.7 Estimation comparison, SIR=2. (a) K=1 estimate, (b) LMSE estimate, and (c) maximum-likelihood estimate.

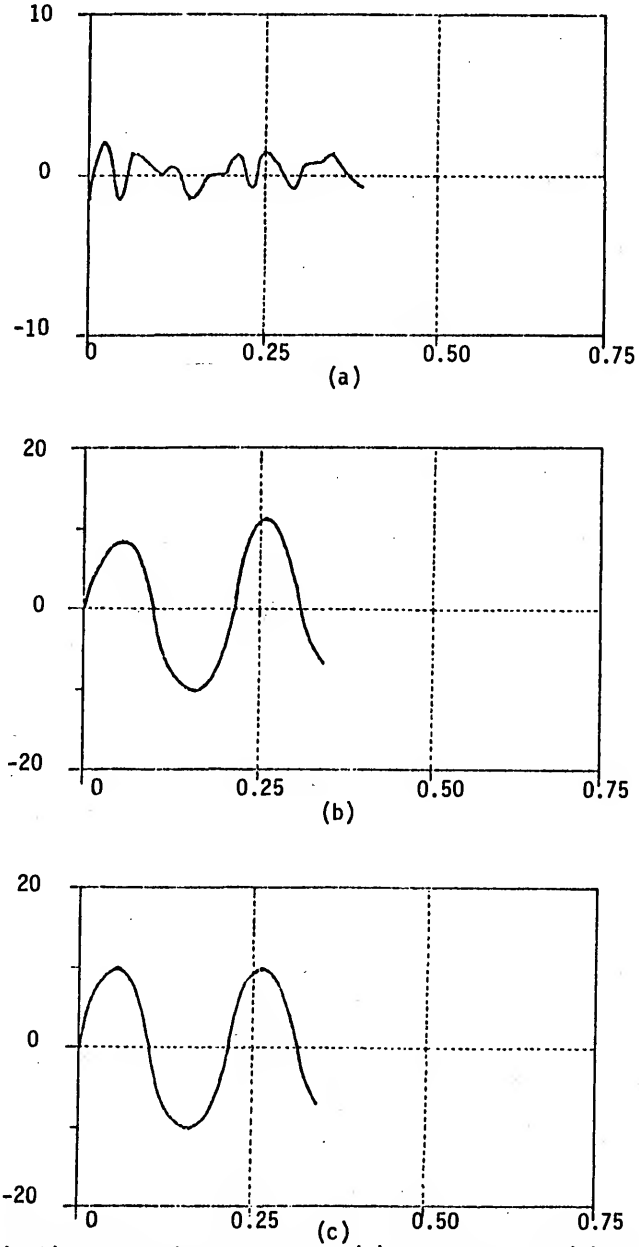
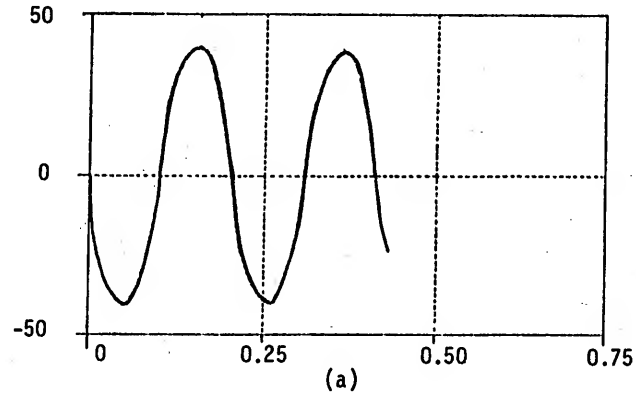
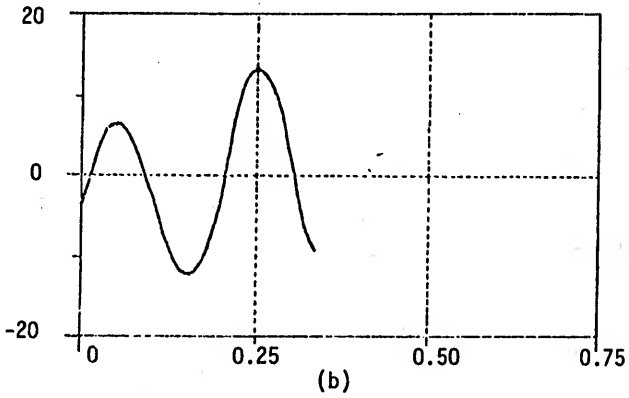



Figure 4.8 Estimation comparison, SIR=1. (a) K=1 estimate, (b) LMSE estimate, and (c) maximum-likelihood estimate.



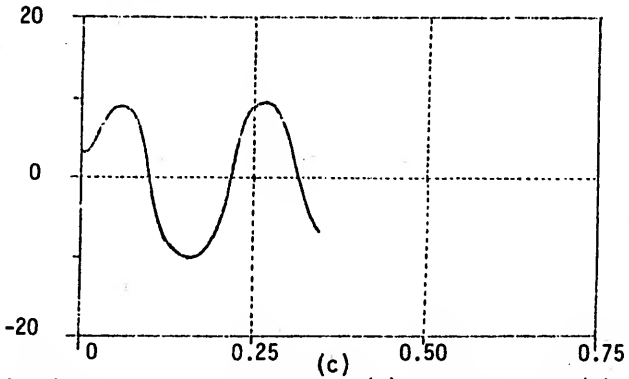




Figure 4.9 Estimation comparison, SIR=0.5. (a) K=1 estimate, (b) LMSE estimate, and (c) maximum-likelihood estimate.

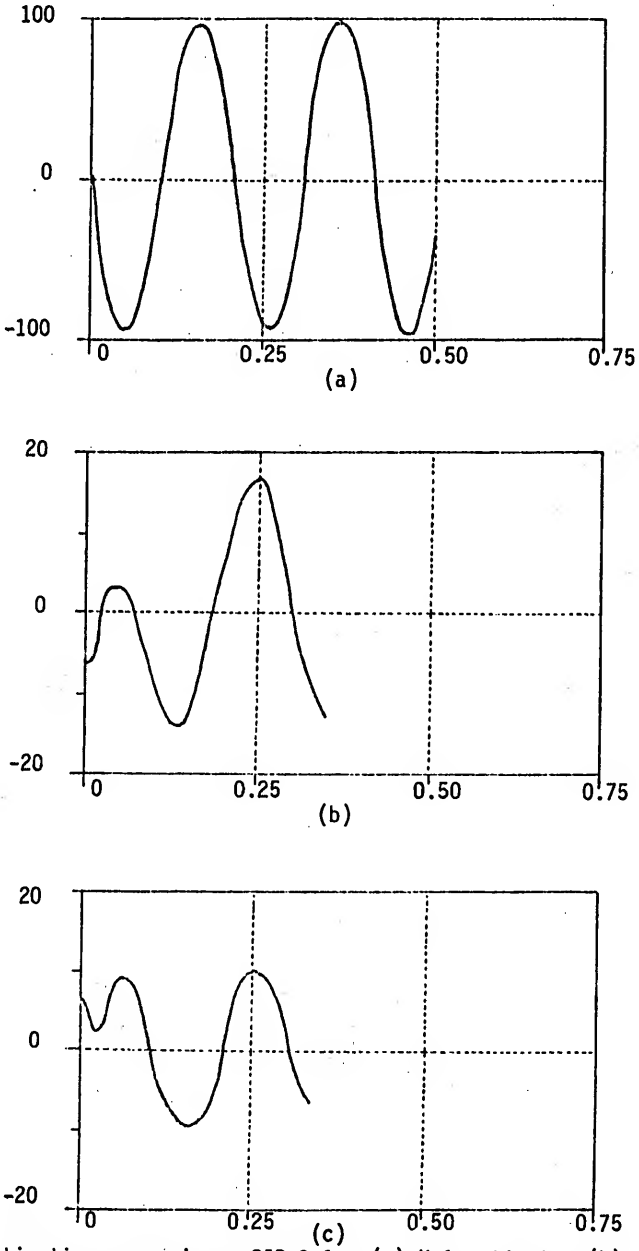
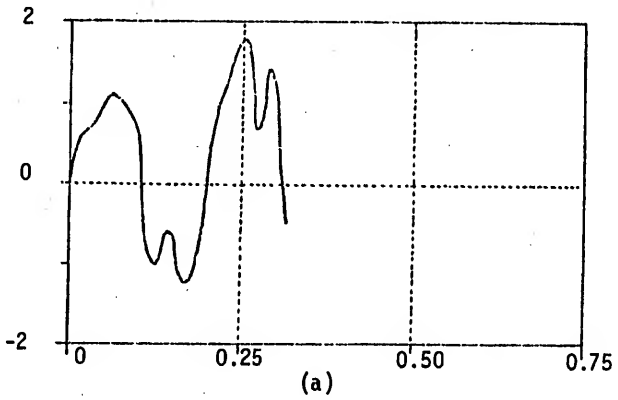



Figure 4.10 Estimation comparison, SIR=0.1. (a) K=1 estimate, (b) LMSE estimate, and (c) maximum-likelihood estimate.

example are used again. The amplitudes of the two waves are fixed at unity and the rms value of the additive Gaussian noise is varied. The SNR is the amplitude of the signal divided by the rms value of the noise. Many computer runs were made for various SNR's. Figures 4.11, 4.12, and 4.13 present the results for SNR = 0.5, 1.0, and 5.0, respectively. The (a) part of these figures is the LMSE estimate; the (b) part is the maximum-likelihood estimate. From these figures it appears that the two estimates are of approximately the same quality. This was found to be true for many other SNR's as well as other plane-wave combinations. This is, of course, an expected result as long as the interfering planewave is small with respect to the signal planewave.

For the final example, five of the seven planewaves of Table 2 (wave numbers 1,3,5,6, and 7) were summed to form the noise. Channel noise was not added. The remaining two planewaves were taken to be the signals to be estimated. The frequency-wavenumber filters presented in Chapter III were used to limit the number of planewaves. The results are summarized in Figures 4.14 and 4.15. The (a) part is the actual wave being estimated. The (b) part is the prefilter LMSE estimate. The (c) part is the postfilter LMSE estimate. The (d) part is the maximum-likelihood estimate. From this example it appears that the maximum-likelihood estimate is better than the prefilter LMSE estimate. The postfilter LMSE estimate appears superior, however, to the maximum-likelihood estimate. Since the background noise consists entirely of planewaves, the postfilter estimate and the maximum-likelihood estimate are identical in the ideal case. In the actual realization, however, the added complexity of the general maximum-likelihood estimator



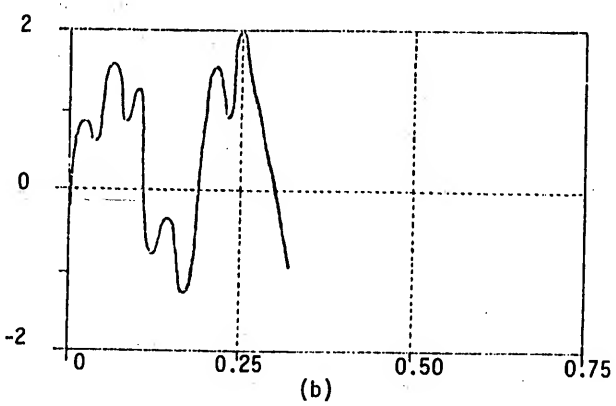




Figure 4.11 Estimation comparison, SNR=0.5. (a) LMSE estimate and (b) maximum-likelihood estimate.

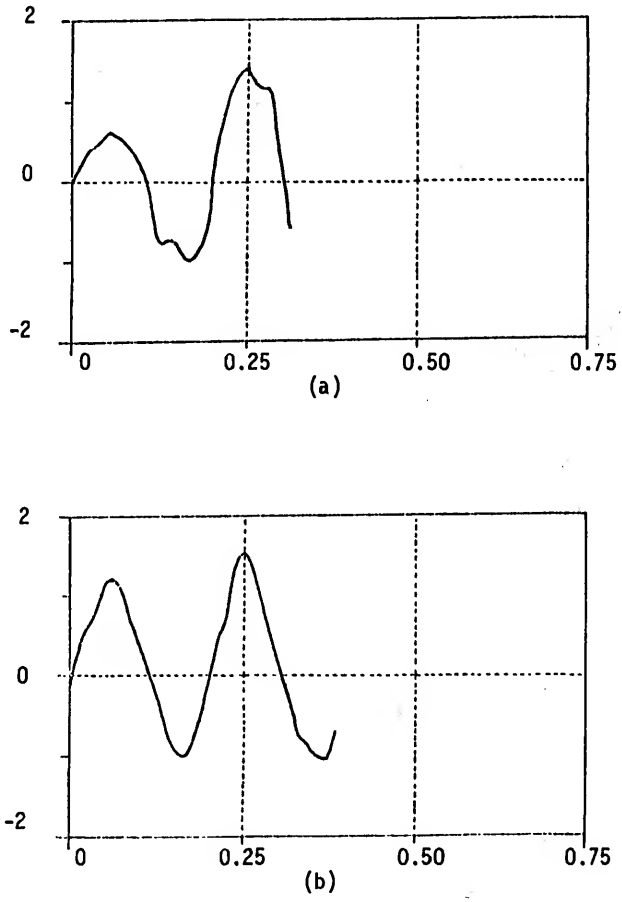



Figure 4.12 Estimation comparison, SNR=1.0. (a) LMSE estimate and (b) maximum-likelihood estimate.

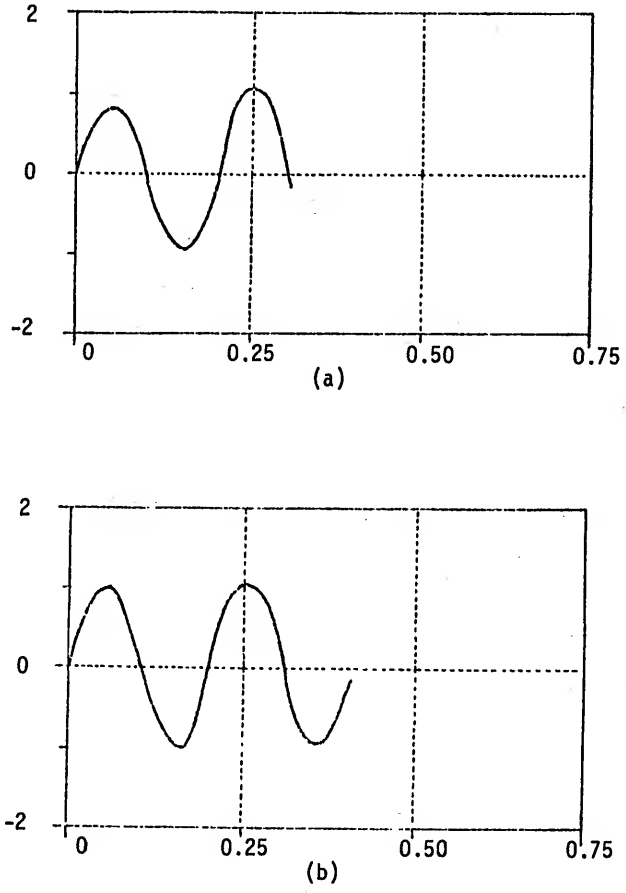



Figure 4.13 Estimation comparison, SNR=5.0. (a) LMSE estimate and (b) maximum-likelihood estimate.

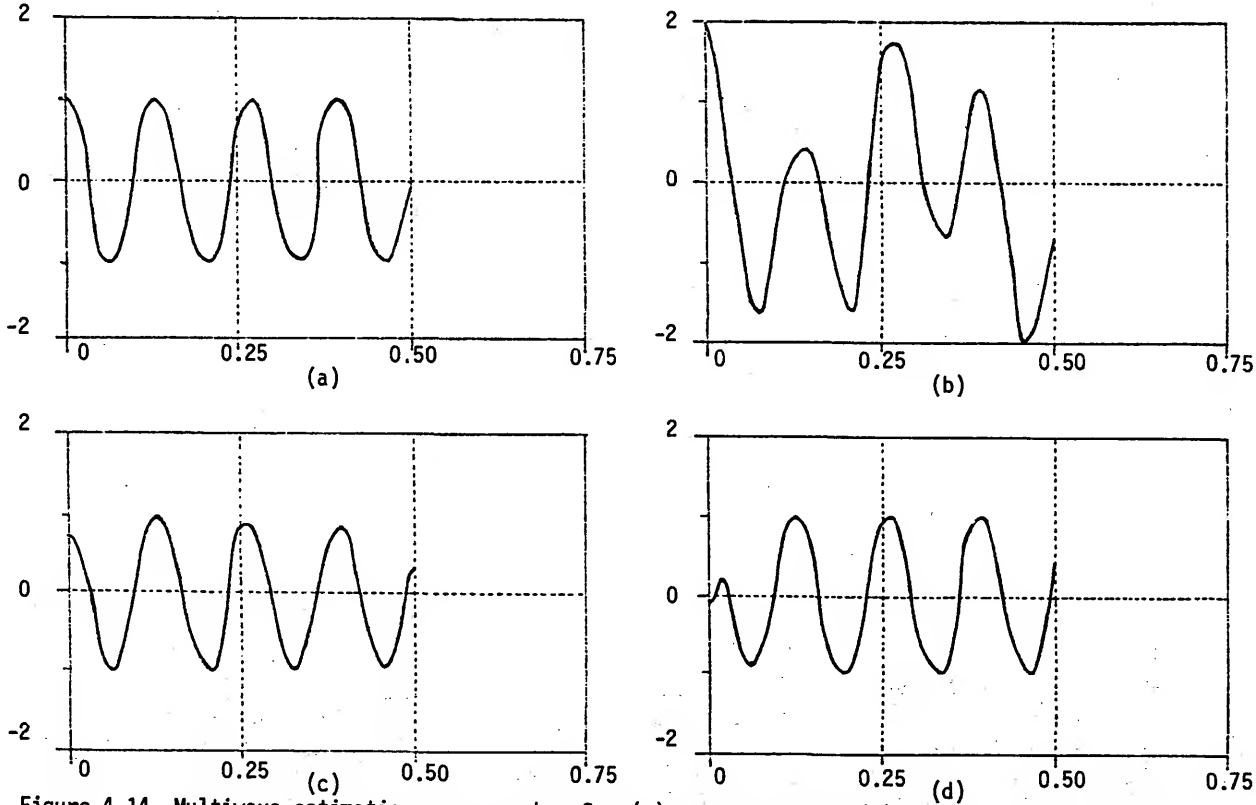



Figure 4.14 Multiwave estimation, wave number 2. (a) wave number 2, (b) prefilter LMSE estimate, (c) postfilter LMSE estimate, and (d) maximum-likelihood estimate.

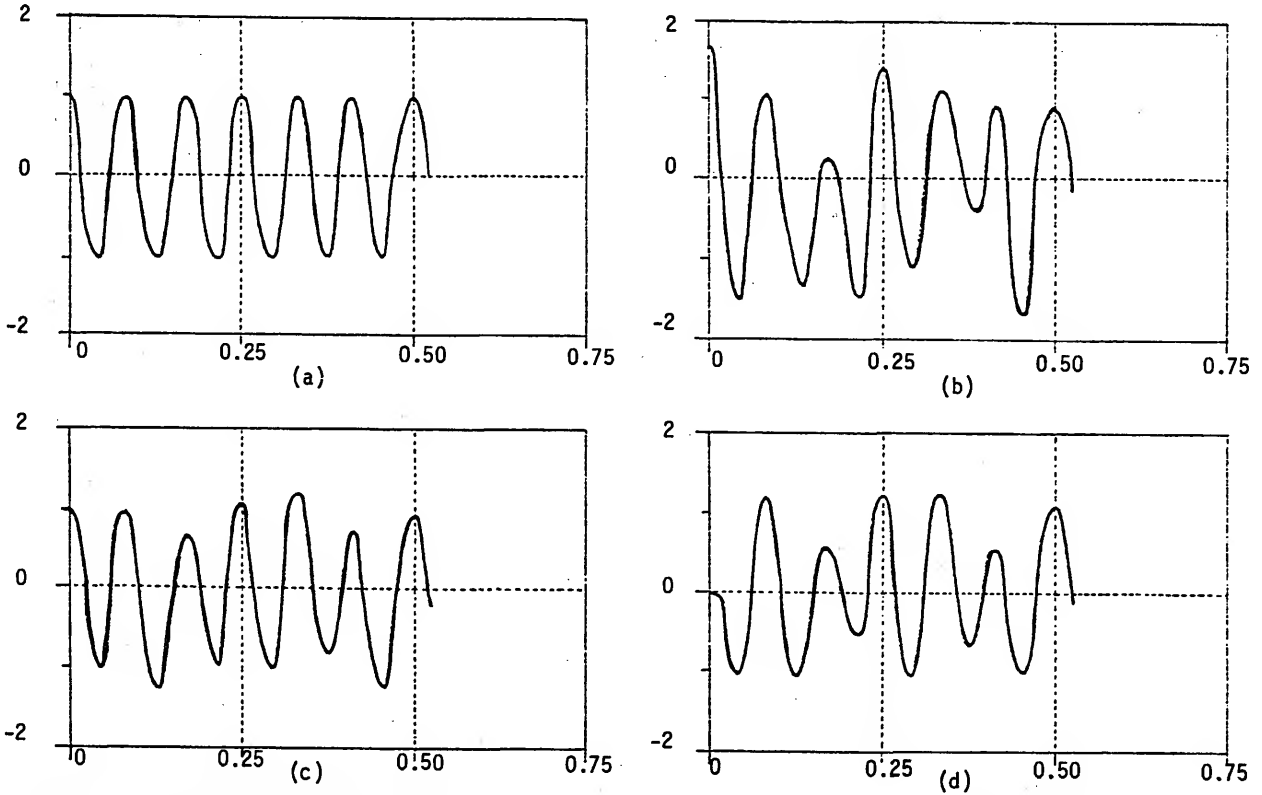



Figure 4.15 Multiwave estimation, wave number 4. (a) wave number 4, (b) prefilter LMSE estimate, (c) postfilter LMSE estimate, and (d) maximum-likelihood estimate.

introduces errors which can be avoided by using the simpler delay and sum operation.

## Summary

The principal contributions of this chapter are as follows.

1. A general multiwave maximum-likelihood estimate is developed. It differs from previous estimators in that the multiwave problem is reduced to a succession of single wave problems.

2. A complete data processing system is presented which accomplishes the multiwave estimation for sufficiently large signal-to-noise ratios.

# CHAPTER V

## COMPUTER PROCESSING OF SPATIO-TEMPORAL DATA

The techniques presented in the preceding three chapters are well suited for the analysis of data in fields such as sonar and seismology. In this chapter it will be shown that the techniques are particularly well suited for the analysis of electroencephalograms (recorded brain waves). The principals of data collection and processing will be covered.

A particular EEG, the visual evoked response (VER), is, by definition, the bioelectrical activity resulting from the presentation of a visual stimulus. Records can be made from the surface of the scalp of animals or man or the electrodes can be surgically implanted for recording purposes. Our interest lies exclusively in scalp electrodes on human subjects. The basic idea is to vary the stimulus and the recording location using subjects with normal vision and visual dysfunctions in an effort to form correlations between the recorded response and visual and/or cerebral defects as well as to study the basic visual processing system. A recent survey of this field is contained in [43]. The University of Florida Visual Science Laboratory has been engaged in this research for about ten years.

As previously mentioned one of the parameters in studying the VER is the location of the recording electrodes. Frequently many electrodes

are simultaneously attached to the subject in an effort to monitor the distribution of potential as a function of time [44]. Many studies are also concerned with correlations between spatially distinct electrode pairs [45]. To date no attempt to perform true array processing, in our context, has been made.

The recorded spatio-temporal VER is a "signature" which represents the net effect of the source of bioelectrical activity, the transmission of the wave to the scalp electrodes, and the recording instrumentation. Through standard engineering practices, instrumentation errors can be separated and practically eliminated. The separation of the effect of the medium from the effect of the origin of the electrical activity cannot be accomplished without specific assumptions regarding the physical and electrical structure of the medium. Analysis techniques which are presented in this study should prove to be useful tools in attempting to separate the medium from the sources through subsequent research.

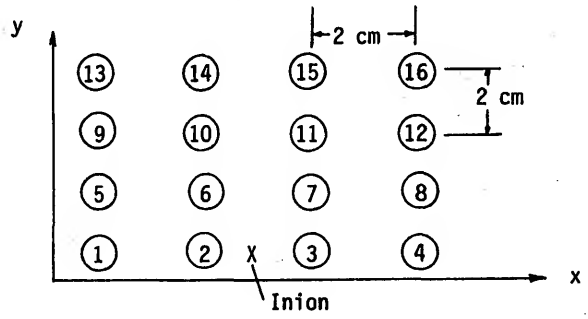



Figure 5.1 Scalp electrode layout (rear view).

The data are collected using standard techniques [46]. Electrode layout is given in Figure 5.1. The stimulus is presented to the subject repetitively. The signals are amplified (approximately 50 db), digitized, and recorded. At the end of the recording the individual responses are summated to obtain an average for a given stimulus condition. The summated data are then recorded on IBM compatible tape. All subsequent operations are done on a large scale digital computer (IBM 360/65). The data acquisition is done with the aid of a special purpose mini-computer (PDP-8/I).

The processing of the data can be divided into two steps. The first step utilizes program PRELIM to preprocess the data for subsequent analysis. The basic steps are outlined in Figure 5.2. The signals on each channel are adjusted to compensate for any unequal gain from the data acquisition system. Calibration signals are used to establish the correcting matrix. The signals are tapered and padded in preparation for filtering. Using subroutine SPBPF, each channel is low-pass filtered at 25 Hz and any dc is also blocked. The data are digitized at 125 Hz. All frequencies above 65.5 Hz will appear in the spectrum as aliased frequencies. The aliasing chart of Figure 5.3 is presented to summarize the effects of filtering and aliasing. All frequencies above the first row are aliased onto the first row as shown. All frequencies between 25 Hz and 65.5 Hz, and their aliases, are removed by filtering. After filtering, the redundant samples are removed by sampling the data at 50 Hz. For convenience the data are truncated at 640 msec. The channels are reordered so that they appear as shown in Figure 5.1. Finally, the data are stored on cards or tape.

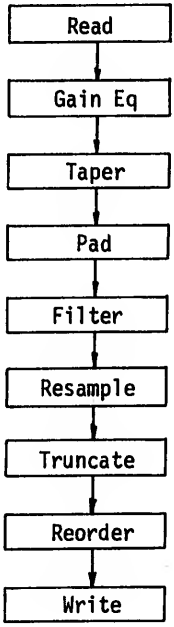



Figure 5.2 Flow chart for PRELIM.

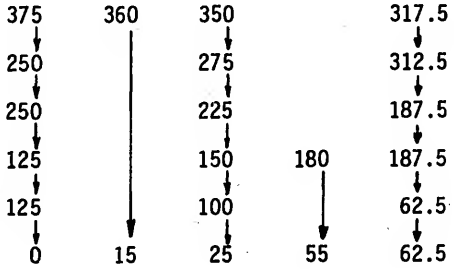



Figure 5.3 Aliasing chart, 125 Hz sampling.

The next phase is the actual analysis of the data. Many possible areas of interest could be investigated. For illustration the effect of low-pass velocity filtering will be investigated.

The first step is to determine where in the frequency-wavenumber space the energy is concentrated. Subroutine FWNO accepts the output of PRELIM without further processing. A review of the frequency-wavenumber spectrum of a particular subject (JGM) reveals that most of the energy is concentrated along the frequency axis of the frequency-wavenumber spectrum. This indicates a very high velocity wave or,in the limiting case, the major components of the time function are identical at each sensor. The effect can be observed in either the spatio-temporal domain or the frequency-wavenumber domain. Figures 5.4 and 5.5 display the spatio-temporal functions via subroutine PLOT16 for two different stimulus conditions. The similarity of the time function on each channel is obvious. Figure 5.6 is the output of a noise trial (light occluded). It is included for comparison purposes. Figure 5.7(a) and (b) contain a section of the output of FWNO (1.6 Hz to 7.8 Hz) for the same stimulus conditions shown in Figures 5.4 and 5.5, respectively. The high energy region in the center of each frame again demonstrates the high velocity effect.

To determine if there are slow planewaves present, the high velocity component must be removed. The data were low-pass velocity filtered at 200 cm/sec using FA and FILTER (reference VERFWNO). The results are displayed in Figure 5.8, (a) and (b). This process has obviously revealed components which were completely masked before filtering. To obtain an estimate of a wave of particular interest,

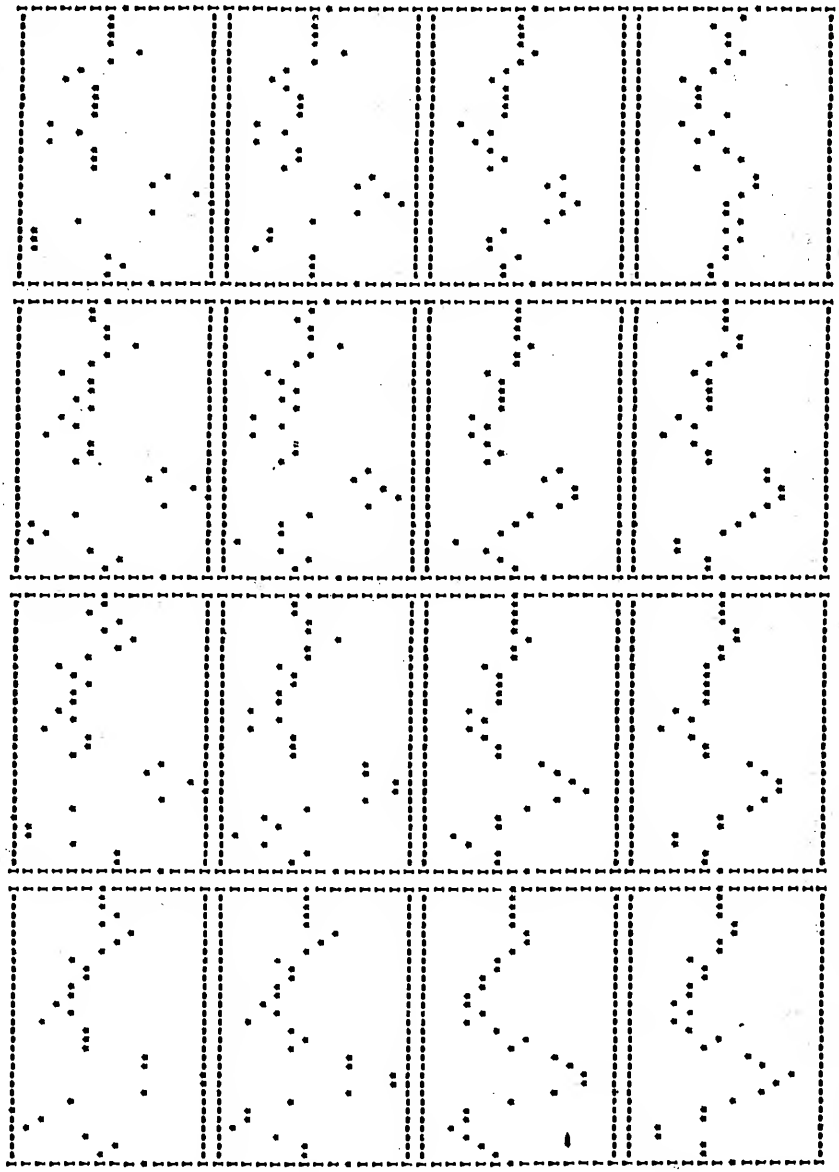



Figure 5.4 VER, white stimulus.

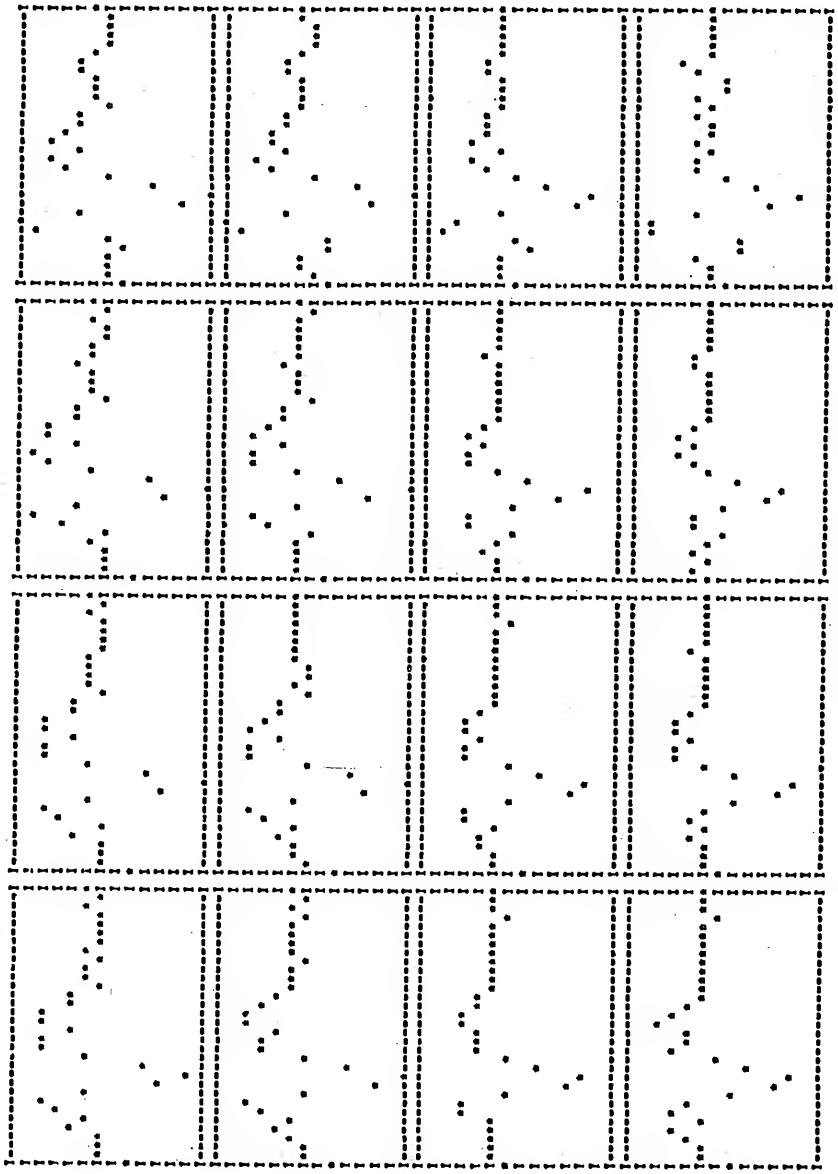



Figure 5.5 VER, red-green stimulus.

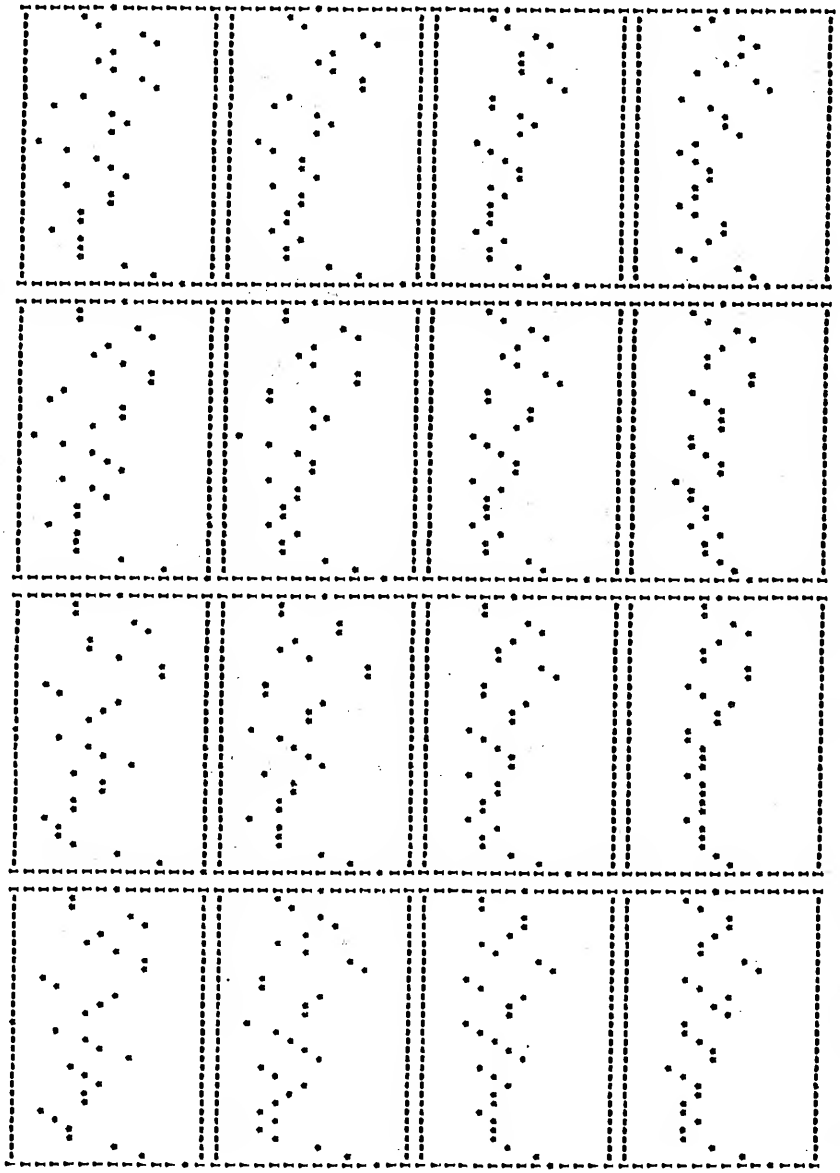



Figure 5.6 VER, noise.

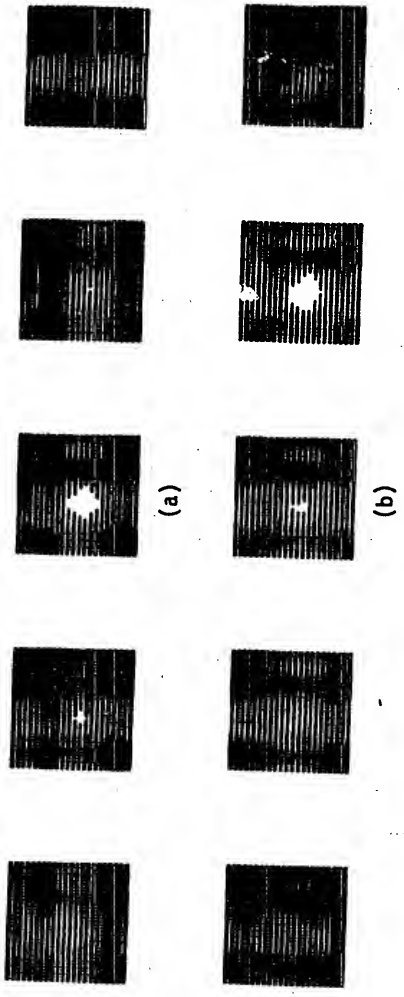



Figure 5.7 FWNO output, 1.6 Hz to 7.8 Hz, before filtering. (a) binocular white stimulus and (b) right eye red and left eye green stimulus.

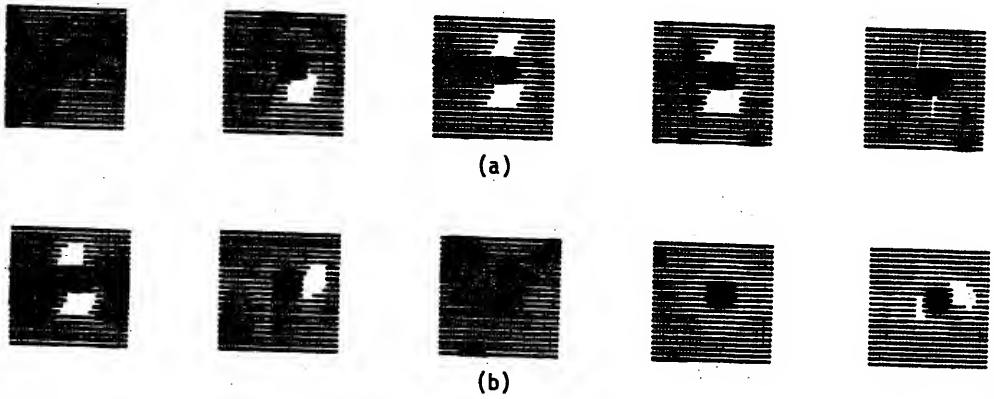



Figure 5.8 FWNO output, 1.6 Hz to 7.8 Hz, after filtering. (a) binocular white stimulus and (b) right eye red and left eye green stimulus.

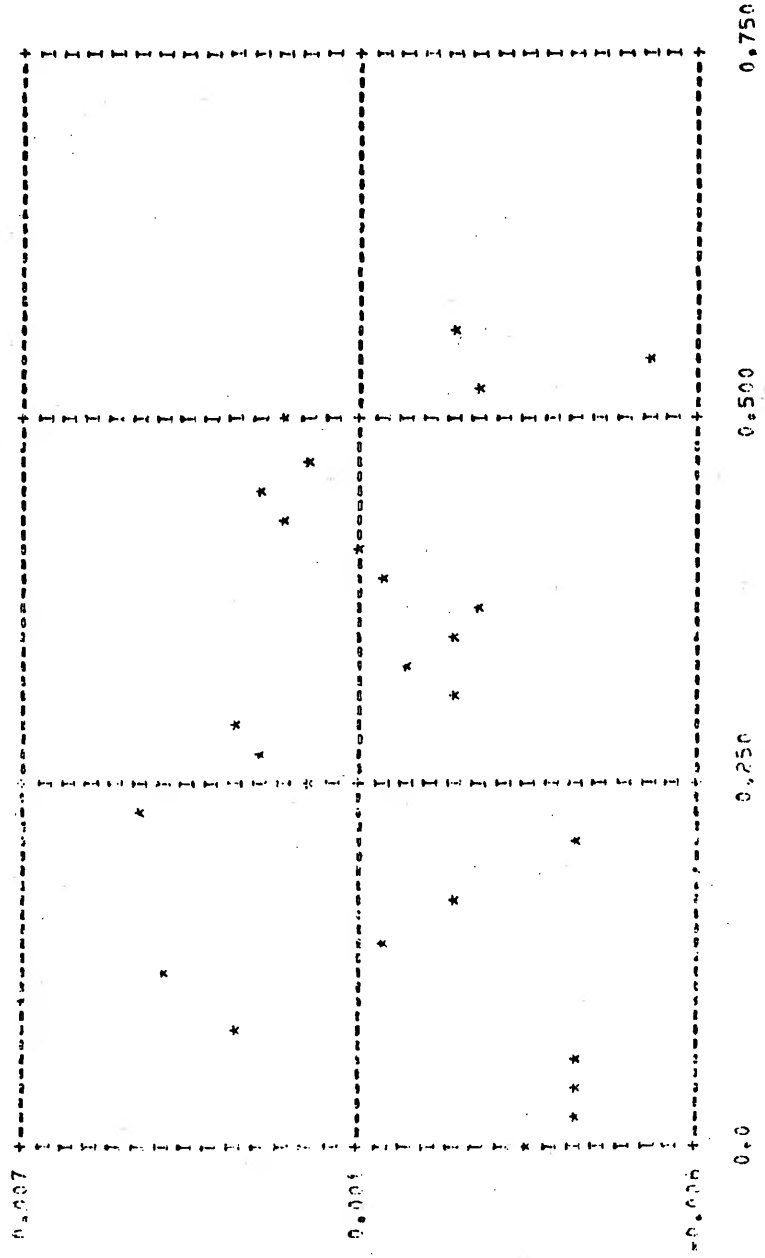
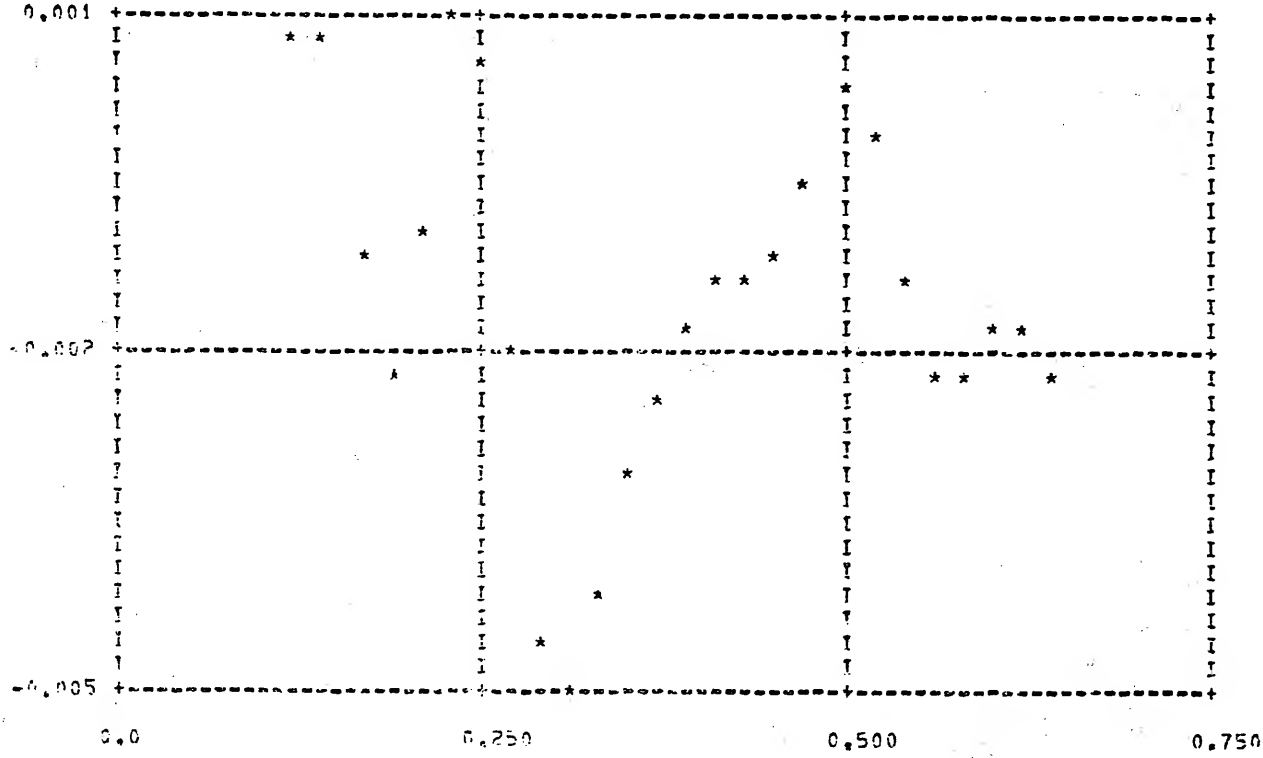



Figure 5.9 Delay and sum output, white stimulus.




DELAYED AND SUMMED OUTPUT W.R.T. ORIGIN
VELOCITY = 50.0 BEARING = 180.0 DEGREES
LEFT SHIFT = 0 NPTS, RIGHT SHIFT = 6 NPTS

Figure 5.10 Delay and sum output, red-green stimulus.

one can call subroutine DS or subroutine ML. Two examples of the output of DS are given in Figures 5.9 and 5.10.

This chapter has shown how the tools developed in the last three chapters may be used in the analysis of actual data. For actual spatio-temporal VER data, the frequency-wavenumber filter revealed features of the data which were otherwise obscured. Further investigations will undoubtedly reveal many additional features.

# CHAPTER VI

## SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

In this study array processing techniques have been developed for the decomposition of overlapping planewaves. The investigation was limited to estimating spatio-temporal functions which are completely unknown a priori. Each technique was investigated both theoretically and computationally. Techniques which appeared to be computationally inefficient were avoided. The general problem of determining the number, vector velocity, and waveshape of an arbitrary sum of planewaves combined with arbitrary noise is solved in a suboptimum manner. The optimum maximum-likelihood estimates are obtained in several important cases.

The principal results of each main chapter will now be reviewed. Recommendations for further work will also be given.

### Multidimensional Power Spectral Density Estimation

The principal result of this chapter is the development of a computationally efficient spatio-temporal power density estimator. The efficiency is achieved by the use of the multidimensional fast Fourier transform. The results of this chapter have been implemented in Fortran programs and can be used to search for multidimensional signals immersed in noise.

As a follow-on effort it is recommended that the stability of the estimator be investigated as a function of record length, averaging technique, and smoothing function. If this could be accomplished the programs could be used to reliably estimate noise spectra.

In addition, it may be of interest to investigate the possibility of redefining the multidimensional FFT with the objective being to eliminate the unequal scaling in positive and negative frequencies. This would be advisable only if work continues with arrays with as few as four points in one dimension.

## Multidimensional Digital Filtering

Generalized nonrecursive frequency domain digital filters are developed which can stop or pass planewaves on the basis of their bearing or velocity or frequency composition or any combination thereof. The filters are fully tested using Fortran subroutines. This chapter represents an extension of a one-dimensional digital filtering technique to multiple dimensions. Computational efficiency is achieved by utilizing multidimensional fast Fourier transforms. Considerable flexibility is achieved by specifying the filters in the frequency-wavenumber domain.

The logical continuation of this work is the development of optimal filters. The optimality criterion itself will first have to be researched. What is needed is a multidimensional measure of transition band and rippling. Transition values should be chosen to obtain optimum trade-off between these two effects. Computer iteration techniques will undoubtedly have to be used in this study. It could be

accomplished on a large scale computer using existing programs. If a small dedicated computer with multidimensional FFT software were available, this study could probably be accomplished in a more efficient and effective manner due to the close interaction between the designer and machine. As a separate study other filtering techniques may be investigated. As an example, the optimum sampled and limited data frequency-wavenumber bandpass filter could be investigated.

## Multiwave Maximum-likelihood Estimation

The general problem of estimating the number, bearing, velocity, and waveshape of overlapping planewaves in noise is approached as follows. The number and vector velocity of the wave is determined using the spectral estimator of Chapter II. The multiwave problem is reduced to a succession of single wave problems using the frequency-wavenumber filters developed in Chapter III. In the fourth chapter the required single wave maximum-likelihood estimator is presented, examined, and implemented in Fortran. This solution could be made optimum by using a maximum-likelihood detector and a wave searching algorithm instead of the simpler multidimensional spectral analysis. This concept was not pursued because of the high cost of the maximum-likelihood estimates and the added complexity of the additional stages.

The principal contribution of this chapter is the reduction of the multiwave problem to a succession of single wave estimation problems. This avoids the difficult problem of finding a suitable algorithm for a multidimensional search.

The relatively high cost of executing the general maximum-likelihood estimator makes its use unattractive. However, this estimator, in the ideal case, is the standard by which others should be judged. It is recommended that further research be directed towards comparing the cascade of a frequency-wavenumber filter and the simple delay and sum operator with the cascade of a frequency-wavenumber filter and the general maximum-likelihood filter. It is expected that the difference will be insignificant in most cases of practical interest.

## Computer Processing of Spatio-temporal Data

In Chapter V the use of the results of the previous three chapters is introduced to the analysis of spatio-temporal VER data. It is shown that filtering is capable of revealing waves that are otherwise obscured.

Two items which need to be investigated are the possibility of spatial aliasing and the error introduced by assuming that the human scalp is flat over the dimensions of the electrode array. To assess the aliasing problem, different electrode spacing, e.g., random spacing, should be tried to determine the effect on the resulting transform. The nonplanar electrode array problem can be solved by using four-dimensional transformations to include the third spatial dimension. Alternatively, it may be possible to simply apply approximate scale correction factors in the existing programs.

The relatively low cost of processing data of dimension 32 x 4 x 4 makes it attractive to expand the array in all dimensions in an attempt

to glean more information from the data. Expanding the spatial dimensions with either real data or zeros provides an additional velocity (or $\vec{k}$) resolution. This activity should probably be coupled with a search for optimum filters such as the low-pass velocity filters.

# Appendices

## APPENDIX A

## NONPARAMETRIC LEAST MEAN SQUARE ERROR ESTIMATION

In this section a nonparametric least mean square error (LMSE) estimation procedure for a single planewave immersed in noise is presented. The original work is due to Kelly and Levin [1]. This procedure is incorporated into this study because it is an extremely important special case of the maximum-likelihood estimator.

Let the sensors be enumerated by k=1,2, ..., K and let the location of the k-th sensor be specified by its coordinates

$$\vec{r}_k = (r_{kx}, r_{ky}) \tag{A-1}$$

in the conventional two-dimensional Cartesian system. Let s(t) be the signal that would be observed at the origin in the absence of noise. The output of the k-th sensor is then

$$x_k(t) = s(t-\vec{\alpha}\cdot\vec{r}_k) + n_k(t) \tag{A-2}$$

where $n_k(t)$ is the noise at the k-th sensor and $\vec{\alpha}$ is the vector of delays per unit distance suffered by the wave as measured along the coordinate axes. It is assumed that the observation interval is of sufficient length so that the signal is observed at each sensor. The vector velocity of the wave is related to the delay vector by

$$\vec{v} = \vec{\alpha}/|\alpha|^2 \quad . \tag{A-3}$$

This estimation procedure is a nonparametric means of estimating both the vector velocity $\vec{v}$ and the waveform s(t) of a single planewave. Only the estimate of s(t) will be considered. The criterion is to minimize the sum over all the sensors of the integrated squared difference.

$$D(\vec{\alpha},s(t)) = \sum_{k=1}^{K} \int_0^T [x_k(t) - s(t-\vec{\alpha}\cdot r_k)]^2 \, dt \quad . \tag{A-4}$$

By setting $x_k$=0 for t outside {0,T}, allowing the limits of integration to extend to $\pm\infty$ , changing variables, and expanding, we have

$$D(\vec{\alpha},s(t)) = \sum_k \int x_k^2(t)dt + K\int [s(t) - \frac{1}{K} \sum_k x_k(t+\vec{\alpha}\cdot\vec{r}_k)]^2 \, dt$$

$$- \frac{1}{K} \int [\sum_k x_k(t+\vec{\alpha}\cdot\vec{r}_k)]^2 \, dt \quad . \tag{A-5}$$

The expression is minimized with respect to s(t) by choosing

$$s(t) = \frac{1}{K} \sum_k x_k(t+\vec{\alpha}\cdot\vec{r}_k) \quad . \tag{A-6}$$

The estimator is diagramed in Figure A.1 with $\vec{\alpha}\cdot\vec{r}_j=\tau_j$. This estimator is by definition the delay and sum processor.

If the received waveforms $\{x_k\}$ contain only the signal without noise, the output is exactly equal to the signal with respect to the origin in space. If the input consists of noise only, the output consists

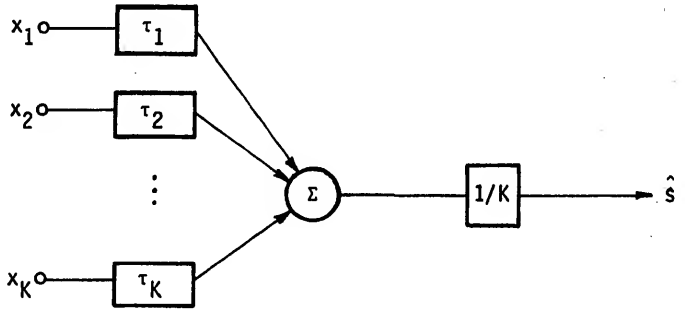



Figure A.1 General least mean square error estimator.

of less noise. To prove this, consider the output of the summer

$$b(t) = x_1(t+\tau_1) + x_2(t+\tau_2) + \ldots + x_K(t+\tau_K) \qquad \text{(A-7)}$$

where

$$\tau_i = \vec{\alpha} \cdot \vec{r}_i \quad , \qquad i=1,2, \ldots, K \quad .$$

The autocorrelation of b(t) is

$$\begin{aligned} R_{bb}(t,t+\tau) &= E[b(t)\, b(t+\tau)] \\ &= E\,[x_1(t+\tau_1)x_1(t+\tau_1+\tau) + \ldots + x_1(t+\tau_1)x_K(t+\tau_K+\tau) + \\ &\quad x_2(t+\tau_2)x_1(t+\tau_1+\tau) + \ldots + x_2(t+\tau_2)x_K(t+\tau_K+\tau) + \\ &\quad \vdots \\ &\quad x_K(t+\tau_K)x_1(t+\tau_1+\tau) + \ldots + x_K(t+\tau_K)x_K(t+\tau_K+\tau)] \quad . \end{aligned} \qquad \text{(A-8)}$$

Assuming wide-sense stationarity

$$R_{bb}(\tau) = R_{11}(\tau) + R_{12}[\tau-(\tau_1-\tau_2)] + \ldots + R_{1K}[\tau-(\tau_1-\tau_K)] +$$

$$R_{21}[\tau-(\tau_2-\tau_1)] + R_{22}(\tau) + \ldots + R_{2K}[\tau-(\tau_2-\tau_K)] +$$

$$\vdots$$

$$R_{K1}[\tau-(\tau_K-\tau_1)] + \ldots + R_{KK}(\tau) \quad . \tag{A-9}$$

The power spectral density function of b(t) is

$$P_{bb}(f) = \mathcal{F}\{R_{bb}(t)\}$$

$$= P_{11}(f) + P_{12}(f)e^{-i2\pi(\tau_1-\tau_2)} + \ldots + P_{1K}(f)e^{-i2\pi(\tau_1-\tau_K)} +$$

$$P_{21}(f)e^{-i2\pi(\tau_1-\tau_2)} + P_{22}(f) + \ldots +$$

$$\vdots$$

$$P_{K1}(f)e^{-i2\pi(\tau_1-\tau_K)} + \ldots + P_{KK}(f) \quad . \tag{A-10}$$

$$P_{bb}(f) = \sum_{k,\ell}^{K} P_{k\ell}(f)e^{-i2\pi\vec{\alpha}\cdot(\vec{r}_k-\vec{r}_\ell)} \quad . \tag{A-11}$$

The spectral density of the total output noise is

$$P_{out}(f) = P_{bb}(f)\ |1/K|^2$$

$$P_{out}(f) = 1/K^2 \sum_{k,\ell}^{K} P_{k\ell}e^{-i2\pi\vec{\alpha}\cdot(\vec{r}_k-\vec{r}_\ell)} \quad . \tag{A-12}$$

The ratio of noise power out to noise power in is

$$\frac{P_{out}(f)}{P_{in}(f)} = \frac{1/K^2 \sum_{k,\ell=1}^{K} P_{\ell k}(f) e^{-i2\pi \vec{\alpha} \cdot (\vec{r}_\ell - \vec{r}_k)}}{\sum_{k,\ell=1}^{K} P_{\ell k}(f)} \quad . \tag{A-13}$$

If the input noise is uncorrelated between sensors, that is, the off-diagonal terms of the cross-power spectral matrix are zero, the ratio becomes

$$\frac{P_{out}}{P_{in}} = \frac{1/K^2 \sum_{k=1}^{K} P_{kk}(f)}{\sum_{k=1}^{K} P_{kk}} = 1/K \quad . \tag{A-14}$$

If the noise is spatially correlated in the form of a planewave (or sum of planewaves) and if the velocity of propagation is large with respect to the separation between sensors, i.e., $\vec{\alpha}_o \cdot (\vec{r}_\ell - \vec{r}_k)$ is nearly zero, the exponential term of the numerator of (A-14) will be approximately unity and the off-diagonal terms will be approximately real and positive. Under these conditions the power ratio will be greater than $1/K^2$.

It is shown in Chapter IV that if an ideal maximum-likelihood estimator is used instead of least mean square error estimation procedures, and only spatially correlated noise is present, the output noise power is zero. In Chapter III it is shown that the same result can be accomplished using ideal vector velocity filters. This comparison demonstrates the advantage of the filter delay and sum operation over the simpler delay and sum estimator. If, however, the

noise is uncorrelated between sensors, the filter-delay operation reduces to the simpler delay operation.

The foregoing comparison applies to ideal processes. In practice the general filter delay and sum operation is not realizable and can only be approximated. In Chapters III and IV the difference between ideal filters and the digital approximations are discussed. The ideal delay and sum operation, however, is realizable and approximations are not necessary. The only errors in the digital delay and sum operations are due to the limited precision of the calculations. For large scale computers operating on the type of data considered in this dissertation, these errors are negligible in comparison with the errors committed in employing digital approximation of ideal filters.

## APPENDIX B

## A BRIEF REVIEW OF MAXIMUM-LIKELIHOOD ESTIMATION

In this appendix we will qualitatively explain what a maximum-likelihood (ML) estimator is with respect to other optimum estimators, quantitatively define the ML estimate, and review its properties.

Generally, the problem of optimum detection is conceptually simpler than that of estimation. In this section we will present a simplified qualitative description of the detection problem before addressing the subject of estimation.

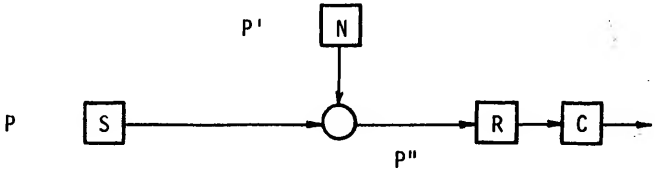



Figure B.1 General statistical model.

Consider the system described in Figure B.1. S, the signal source, is a random variable (random number generator) described in terms of the statistics P (a priori statistics). N, the noise source, is a random variable with statistics P'. S and N are combined in a

known manner. The statistics of the result are denoted by P" (a posteriori statistics). R must decide with the least probability of error which of the M possible numbers was generated by S (M'ary hypothesis test). C is a cost matrix which may be thought of as weighting decisions so as to maximize one's winnings as in a game of chance. More conveniently, C minimizes the average risk in making decisions.

## Bayes Criterion

If we know P, P', P", and C and use the output of C to make the decision we have adopted the Bayes criterion of minimizing the average risk.

## Minimax Criterion

If P is unknown, but all other factors are known, we may select the value of P which maximizes the minimum average (Bayes) risk. The purpose of adopting this procedure is to operate with a known bound on the average risk.

## Neyman-Pearson Criterion

This criterion is usually only applied to binary hypothesis tests [33]. For example [32], consider the missile detection problem in radar. One of four conditions must exist:

1. declare target present (S=1) when it is,
2. declare target present (S=1) when it is not,
3. declare target absent (S=0) when it is, or
4. declare target absent (S=0) when it is not.

Both Bayes and Minimax procedures assign relative costs to all possible outcomes and the decisions are weighted accordingly. In the Neyman-Pearson case, the cost function is unknown and the criterion requires one to maximize the probability of target detection, condition 1, subject to the constraint of an upper bound on failing to detect targets, condition 4.

## Maximum a Posteriori

The maximum a posteriori (MAP) criterion is a special case of the Bayes criterion where the costs of deciding correctly are set to unity and the cost of deciding incorrectly are set to zero. In other words, we want to maximize the probability of correct decision, with equal weight given to each decision, without regard to cost (consequences) of incorrect decision. This is a criterion commonly explored in communication theory [47].

## Maximum-likelihood

This is a special case of the MAP detector. The same cost matrix (one-zero) is used but the maximum-likelihood (ML) detection does not have available the a priori statistics P.

In most detection problems of practical interest, one is concerned with the presence or absence of complete waveforms. If some, or all, of the parameters of the waveform are unknown, we can still pose the optimum detection problem (if a priori information is available) but we may also be interested in determining, as well as we are able, the unknowns. This is the estimation problem. The theory of estimation

is concerned with minimizing the errors due to noise and analyzing the average irreducible residual error. The theory of estimation presupposes that we have a complete statistical description of the measured values (or received waveform) which represent the signal of interest after being contaminated by noise. Specifically, we must have the joint probability density function for the received vector $p(\vec{x}/\vec{\theta})$ as a function of the vector of unknown parameters $\vec{\theta}$. This means that the noise must be completely characterized, i.e., its joint probability density function must be given, and the method of combining the signal and noise must be known.

The particular estimation strategy to be used depends upon the availability of a priori statistics and the same sort of consideration that we enumerated for the detection problem. The problem to be considered in this dissertation assumes complete ignorance of the unknowns and ignores cost functions. This leads directly to maximum-likelihood estimates. By definition, the maximum-likelihood estimates of a set of parameters $\vec{\theta}$ are those values of $\vec{\theta}=\hat{\vec{\theta}}$ which maximize $p(\vec{x}/\vec{\theta})$. The important properties of maximum-likelihood estimates are

1. the estimate is asymptotically efficient, i.e., in the limit as the number of independent measures approaches infinity, $N\to\infty$, no other estimate renders a smaller variance,

2. the maximum-likelihood estimate is consistant, i.e., as $N\to\infty$, $\hat{\vec{\theta}}\to\vec{\theta}$,

3. the maximum-likelihood estimate is unbiased, i.e., $E\{\hat{\vec{\theta}}\} = \vec{\theta}$,

4. as $N\to\infty$, $p(\hat{\vec{\theta}})$ is multivariate Gaussian.

## APPENDIX C

## DERIVATION OF THE TIME DOMAIN SOLUTION

In this appendix the time domain solution of the single planewave maximum-likelihood estimation problem is given. Consider any set of N Gaussian random variables, $\underline{x} \triangleq [x_1, x_2, \ldots, x_N]$. In matrix form the N-variate normal density function is given by

$$p(\underline{x}) = \frac{1}{(2\pi)^{N/2} |\underline{\rho}|^{1/2}} \exp\{-1/2\ (\underline{x}-\overline{\underline{x}})\ \underline{\rho}^{-1}\ (\underline{x}-\overline{\underline{x}})^T\} \qquad (C\text{-}1)$$

where $\overline{\underline{x}}$ is the vector of means,

$$\overline{\underline{x}} = \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_N ,$$

$\underline{\rho}$ is the covariance matrix,

$$\underline{\rho} \triangleq \begin{bmatrix} \rho_{11} & \cdots & \rho_{1N} \\ \vdots & & \\ \rho_{N1} & \cdots & \rho_{NN} \end{bmatrix} = \{\rho_{ij}\}$$

and the general element of this matrix is

$$\rho_{ij} \triangleq E[(x_i-\overline{x}_i)(x_j-\overline{x}_j)], \qquad 1 \le i,j \le N \quad .$$

Also $|\underline{\rho}|$ is the determinant of $\underline{\rho}$, $\underline{\rho}^{-1}$ is the inverse of $\underline{\rho}$, i.e.,

$$\underline{\rho}^{-1} \triangleq \{\rho_{ij}^{-1}\}$$

and

$$\rho_{ij}^{-1} \triangleq (-1)^{i+j} \, |\underline{\rho}_{ij}|/|\underline{\rho}|$$

where $\underline{\rho}_{ij}$ is obtained from $\underline{\rho}$ by deleting the i-th row and the j-th column.

The expression (C-1) for $p(\underline{x})$ can represent the joint probability density function (PDF) of a Gaussian random process $x(t)$ if $x(t)$ is sampled in time such that

$$\underline{x}(\underline{t}) \triangleq [x(t_1), x(t_2), \ldots, x(t_N)] = [x_1, x_2, \ldots, x_N]$$

constitutes a set of jointly Gaussian random variables for any set of observations $\{t_j\}$, $1 \leq j \leq N$.

This concept is easily extended to multidimensional sampling. In array processing the sampling is accomplished spatially and temporally. Suppose that at each of N time instances $\{t\}$, $1 \leq j \leq N$, K array sensors are simultaneously sampled at the points $\{s_i\}$, $1 \leq i \leq K$, such that M=NK is the total number of samples and

$$\begin{aligned} \underline{x}(s,t) \triangleq & [x(s_1,t_1), x(s_1,t_2), \ldots, x(s_1,t_N), \\ & x(s_2,t_1), x(s_2,t_2), \ldots, x(s_2,t_N), \\ & \vdots \\ & x(s_K,t_1), x(s_K,t_2), \ldots, x(s_K,t_N)] \\ \triangleq & [x_1, x_2, \ldots, x_M] = \underline{x} \quad . \end{aligned}$$

Noting that $\underline{x}$ is still a vector of random variables (C-1) still applies with N replaced by M. It now remains to discuss the covariance matrix $\underline{\rho}$.

Expanding $\underline{x}$ using two subscripts instead of one,

$$\underline{x} = [x_{11}, x_{12}, \ldots, x_{1N},$$
$$x_{21}, x_{22}, \ldots, x_{2N},$$
$$\vdots$$
$$x_{K1}, x_{K2}, \ldots, x_{KN}]$$

or

$$\underline{x} = \{x_{ij}\}, \quad 1 \le i \le K, \quad 1 \le j \le N \quad .$$

Now partition $\underline{x}$ into K submatrices

$$\underline{x} = [\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_K]$$

or

$$\underline{x} = \{\underline{x}_i\}, \quad 1 \le i \le K$$

where

$$\underline{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]$$

or

$$\underline{x}_i = \{x_{ij}\}, \quad 1 \le j \le N \quad .$$

In this context vector $\underline{x}_i$ is a temporal subvector of the principal spatial vector $\underline{x}$. The covariance matrix is similarly partitioned with all means set to zero.

$$\underline{\rho} \triangleq \begin{bmatrix} \underline{\rho}_{11} & \underline{\rho}_{12} & \cdots & \underline{\rho}_{1K} \\ & \vdots & & \\ \underline{\rho}_{K1} & \underline{\rho}_{K2} & \cdots & \underline{\rho}_{KK} \end{bmatrix}$$

or

$$\underline{\rho} = \{\underline{\rho}_{rs}\}, \quad 1 \leq r,s \leq K$$

where

$$\underline{\rho}_{rs} \triangleq \begin{bmatrix} \rho_{rs}(1,1), & \rho_{rs}(1,2) & \cdots & \rho_{rs}(1,N) \\ \rho_{rs}(2,1) & & & \\ & & \vdots & \\ \rho_{rs}(N,1) & & \cdots & \rho_{rs}(N,N) \end{bmatrix}$$

or

$$\underline{\rho}_{rs} = \{\rho_{rs}(m,n)\}, \quad 1 \leq m,n \leq N$$

where

$$\rho_{rs}(m,n) = E[x_{r,m}x_{s,n}] \quad .$$

At this point it may be observed that covariances can be specified by the cross-correlation function

$$R_{x_r x_s}[t,t'] \triangleq E[x_r(t),x_s(t')]$$

for two random processes $x_r(t)$ and $x_s(t')$ or by the autocorrelation function

$$R_x[(r,t),(s,t')] \triangleq E[x(r,t)x(s,t')]$$

of the single random process $x(t,r)$. The latter approach will be followed in this discussion.

For most practical applications we assume stationarity. Assuming spatial stationarity

$$\rho_{rs}(m,n) = \rho_{rs}(n-m)$$

and, if $n' \triangleq n-m$ ,

$$\rho_{rs}(m,n) = \rho_{rs}(n') = E[x_{r,m'} x_{r,m'+n'}] \quad .$$

For spatial stationarity, in addition to temporal stationarity,

$$\rho_{rs}(m,n) = \rho_{r'}(n')$$

where $r' = s-r$. Thus, our covariance matrix consists of $K^2$ submatrices. Each submatrix is itself a covariance matrix.

If the noise is a zero-mean multidimensional discrete random process, the PDF is

$$p_0(\underline{x}) = \frac{1}{(2\pi)^{M/2} |\underline{\rho}|^{1/2}} \exp\{-1/2\ \underline{x}\ \underline{\rho}^{-1}\ \underline{x}^T\} \quad .$$

Using the submatrices as previously defined

$$\exp[-1/2\ \underline{x}\rho^{-1}\underline{x}^T] = \exp\{-1/2[\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_K][\begin{matrix} \underline{\rho}_{11}^{-1} & \underline{\rho}_{12}^{-1} & \cdots & \underline{\rho}_{1K}^{-1} \\ \underline{\rho}_{21}^{-1} & & & \\ \vdots & & \ddots & \vdots \\ \underline{\rho}_{K1}^{-1} & & \cdots & \underline{\rho}_{KK}^{-1} \end{matrix}][\begin{matrix} \underline{x}_1 \\ \vdots \\ \underline{x}_K \end{matrix}]\} \quad .$$

If instead of receiving noise alone we receive, in addition, a deterministic signal which is the same at each sensor (or made so by using delay elements) the statistical description of the received random process is changed in mean value by an amount equal to the signal, i.e.,

$$p_1(\underline{x}) = \frac{1}{(2\pi)^{M/2}|\underline{\rho}|^{1/2}} \exp\{-1/2\ (\underline{x}-\underline{s})\ \underline{\rho}^{-1}\ (\underline{x}-\underline{s})^T\}$$

$$\underline{s} \triangleq [\underline{s}_1, \underline{s}_2, \ldots, \underline{s}_K] = \{\underline{s}_i\}$$

$$\underline{s}_i = [\, s_{i1}, s_{i2}, \ldots, s_{iN}\,] \quad .$$

$\underline{s}_i$ is the vector of the samples of the signal and is the same for all sensors, $1 \le i \le K$. Eliminating the vector notation

$$p_1(\underline{x}) = C \exp\{-1/2 \sum_{\ell=1}^{M} (x_\ell - s_\ell)\rho_\ell(m)(x_m - s_m)\}$$

with $C = 1/(2\pi)^{M/2}|\rho|^{1/2}$ . Using the double subscript notation

$$p_1(\underline{x}) = C \exp\{-1/2 \sum_{i=1}^{K}\sum_{m=1}^{N}\sum_{j=1}^{K}\sum_{n=1}^{N} (x_{im} - s_{im})\rho_{ij}^{-1}(m,n)(x_{jn} - s_{jn})\} \quad .$$

When written in this form the first subscript on $s_{im}$ is redundant since

$$s_{in} = s_{jm}, \qquad 1 \le i,j \le K \quad .$$

Accordingly, we define the N-dimensional vector

$$\underline{s}_N \triangleq [s_1, s_2, \ldots, s_N] \quad .$$

The maximum-likelihood estimates are the values of the parameters being estimated which maximize the joint PDF of the received data. In this case the estimates can be found by maximizing $p_1(x)$ with respect to the received signal. Due to the monotonic character of this function, we may first take the logarithm and then equate the derivative with respect to $s_n$ to zero. We have N parameters which constitute the sampled signal s

$$\phi = \frac{d}{ds_n} \ln p_1(x), \qquad 1 \le n \le N$$

$$\phi = \sum_{i=1}^{K} \sum_{m=1}^{N} \sum_{j=1}^{K} \{s_m \rho_{ij}^{-1}(m,n) - x_{im}\rho_{ij}^{-1}(m,n)\}, \quad 1 \le n \le N \quad .$$

The values of $\{s_m\}$, $1 \le m \le N$, which satisfy the above relationship, constitute the maximum-likelihood estimates of the signal. These estimates are denoted as $\hat{\underline{S}}_N$. Thus,

$$\Sigma\,\Sigma\,\Sigma\, \hat{S}_m \rho_{ij}^{-1}(m,n) = \Sigma\,\Sigma\,\Sigma\, x_{jm}\rho_{ij}^{-1}(m,n), \qquad 1 \le n \le N$$

$$\sum_{m=1}^{N} \hat{S}_m \sum_{j,k} \rho_{jk}^{-1}(m,n) = \sum_{m=1}^{N} \sum_{j,k=1}^{K} \rho_{jk}^{-1}(m,n) x_{jm} \ , \quad 1 \le n \le N. \quad \text{(C-2)}$$

The last expression represents a set of N linear equations which can be solved by introducing the following N x N matrix.

$$\underline{A}^{-1}(m,n) \triangleq \{\alpha^{-1}(m,n)\}, \qquad 1 \le m,n \le N$$

where

$$\alpha^{-1}(m,n) \triangleq \sum_{j,k}^{K} \rho_{jk}^{-1}(m,n).$$

The right side of (C-2) must be defined as an N column vector, i.e.,

$$\underline{B} \triangleq \{\beta_n\}, \quad 1 \le n \le N$$

$$\beta_n \triangleq \sum_{m=1}^{N} \sum_{j,k=1}^{K} \rho_{jk}^{-1}(m,n) x_{jm} \quad .$$

Thus, (C-2) becomes

$$\underline{\hat{S}}_N \, A^{-1}(m,n) = \underline{B}$$

then

$$\underline{\hat{S}}_N = \underline{B} \, \underline{A}(m,n)$$

and the general n-th solution is

$$\hat{S}_{N_n} = \sum_{m'=1}^{N} \{ [ \sum_{m=1}^{N} \sum_{j,k=1}^{K} \rho_{jk}^{-1}(m,m') x_{jm} ] [\alpha(m',n)] \}, 1 \le n \le N.$$

Since $(m',n)$ may be taken inside the left pair, we may rearrange the order of summation

$$\hat{S}_{N_n} = \sum_{m=1}^{N} \sum_{j=1}^{K} [ \sum_{k=1}^{K} \sum_{m'=1}^{N} \rho_{jk}^{-1}(m,m') \alpha(m',n) ] \, x_{jm}$$

or

$$\hat{S}_{N_n} = \sum_{j=1}^{K} [ \sum_{m=1}^{N} h_{j,m,n} x_{jm} ] \qquad \text{(C-3)}$$

where

$$h_{j,m,n} \triangleq \sum_{k=1}^{K} \sum_{m'=1}^{N} \rho_{jk}^{-1}(m,m')\alpha(m',n) \quad . \tag{C-4}$$

$\hat{S}_{N_n}$ is thus a sum over K filters of a linear operation (weighted sum) on the inputs. Except for minor notational changes, this is the solution given by Capon [3].

Equation (C-4) demonstrates that the solution is dependent upon both the spatial and temporal sampling. The fact that (C-4) depends upon (m,n) rather than (m-n) implies that (C-3) is a set of K linear, but time varying, operations.

# APPENDIX D

# PROGRAM LISTINGS

```
C           COMPLEX*16 VERSION OF IBM'S MINV
C     ..................................................................COMINV
C                                                                      COMINV
C        SUBROUTINE COMINV                                               MINV
C                                                                      COMINV
C        PURPOSE                                                       COMINV
C           INVERT A MATRIX                                            COMINV
C                                                                      COMINV
C                                                                      COMINV
C        USAGE                                                         COMINV
C           CALL COMINV(A,N,D,L,M)                                       MINV 1
C                                                                      COMINV 1
C        DESCRIPTION OF PARAMETERS                                     COMINV 1
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY COMINV 1
C               RESULTANT INVERSE.                                     COMINV 1
C           N - ORDER OF MATRIX A                                      COMINV 1
C           D - RESULTANT DETERMINANT                                  COMINV 1
C           L - WORK VECTOR OF LENGTH N                                COMINV 1
C           M - WORK VECTOR OF LENGTH N                                COMINV 1
C                                                                      COMINV 1
C        REMARKS                                                       COMINV 2
C           MATRIX A MUST BE A GENERAL MATRIX                          COMINV 2
C                                                                       MINV 220
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MINV 230
C           NONE                                                        MINV 240
C                                                                       MINV 250
C        METHOD                                                         MINV 260
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   MINV 270
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    MINV 280
C           THE MATRIX IS SINGULAR.                                     MINV 290
C                                                                       MINV 300
C     ..................................................................MINV 310
C                                                                       MINV 320
      SUBROUTINE COMINV(A,N,D,L,M)
      IMPLICIT COMPLEX*16 (A-H,O-Z),INTEGER*4(I-N)
      DIMENSION A(1),L(1),M(1)
                                                                        MINV 350
C     ..................................................................MINV 360
C        SEARCH FOR LARGEST ELEMENT                                     MINV 540
C                                                                       MINV 550
      D=(1.00,0.00)
      NK=-N                                                             MINV 570
      DO 80 K=1,N                                                       MINV 580
      NK=NK+N                                                           MINV 590
      L(K)=K                                                            MINV 600
      M(K)=K                                                            MINV 610
      KK=NK+K                                                           MINV 620
      BIGA=A(KK)                                                        MINV 630
      DO 20 J=K,N                                                       MINV 640
      IZ=N*(J-1)                                                        MINV 650
      DO 20 I=K,N                                                       MINV 660
      IJ=IZ+I                                                           MINV 670
10    IF(CDABS(BIGA)- CDABS(A(IJ))) 15,20,20                            MINV 690
   15 BIGA=A(IJ)                                                        MINV 700
      L(K)=I                                                            MINV 710
      M(K)=J                                                            MINV 720
   20 CONTINUE                                                          MINV 730
C                                                                       MINV 740
C        INTERCHANGE ROWS                                               MINV 750
C                                                                       MINV 760
      J=L(K)                                                            MINV 770
      IF(J-K) 35,35,25                                                  MINV 780
   25 KI=K-N
```

```
      DO 30 I=1,N                                                       MINV 790
      KI=KI+N                                                           MINV 800
      HOLD=-A(KI)                                                       MINV 810
      JI=KI-K+J                                                         MINV 820
      A(KI)=A(JI)                                                       MINV 830
   30 A(JI) =HOLD                                                       MINV 840
C                                                                       MINV 850
C        INTERCHANGE COLUMNS                                            MINV 860
C                                                                       MINV 870
   35 I=M(K)                                                            MINV 880
      IF(I-K) 45,45,38                                                  MINV 890
   38 JP=N*(I-1)                                                        MINV 900
      DO 40 J=1,N                                                       MINV 910
      JK=NK+J                                                           MINV 920
      JI=JP+J                                                           MINV 930
      HOLD=-A(JK)                                                       MINV 940
      A(JK)=A(JI)                                                       MINV 950
   40 A(JI) =HOLD                                                       MINV 960
C                                                                       MINV 970
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS        MINV 980
C        CONTAINED IN BIGA)                                             MINV 990
C                                                                       MINV1000
   45 IF(DABS(BIGA)) 48,46,48                                           MINV1010
   46 D=(0.00,0.00)                                                     MINV1020
      RETURN                                                            MINV1030
   48 DO 55 I=1,N                                                       MINV1040
      IF(I-K) 50,55,50                                                  MINV1050
   50 IK=NK+I                                                           MINV1060
      A(IK)=A(IK)/(-BIGA)                                               MINV1070
   55 CONTINUE                                                          MINV1080
C                                                                       MINV1090
C        REDUCE MATRIX                                                  MINV1100
C                                                                       MINV1110
      DO 65 I=1,N                                                       MINV1120
      IK=NK+I                                                           MINV1130
      HOLD=A(IK)                                                        MINV1140
      IJ=I-N                                                            MINV1150
      DO 65 J=1,N                                                       MINV1160
      IJ=IJ+N                                                           MINV1170
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
C     MOD TO IBM'S ALGORITHM ---                                        
C     IF(DABS(A(IJ))-.0001*DABS(HOLD*A(KJ)).LT.0.0) A(IJ)=(0.000,0.000)
   65 CONTINUE                                                          MINV1220
C                                                                       MINV1230
C        DIVIDE ROW BY PIVOT                                            MINV1240
C                                                                       MINV1250
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
C                                                                       MINV1320
C        PRODUCT OF PIVOTS                                              MINV1330
C                                                                       MINV1340
      D=D*BIGA                                                          MINV1350
C                                                                       MINV1360
C        REPLACE PIVOT BY RECIPROCAL                                    MINV1370
C                                                                       MINV1380
```

```
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
C                                                                       MINV1410
C        FINAL ROW AND COLUMN INTERCHANGE                               MINV1420
C                                                                       MINV1430
      K=N                                                               MINV1440
  100 K=(K-1)                                                           MINV1450
      IF(K) 150,150,105                                                 MINV1460
  105 I=L(K)                                                            MINV1470
      IF(I-K) 120,120,108                                               MINV1480
  108 JQ=N*(K-1)                                                        MINV1490
      JR=N*(I-1)                                                        MINV1500
      DO 110 J=1,N                                                      MINV1510
      JK=JQ+J                                                           MINV1520
      HOLD=A(JK)                                                        MINV1530
      JI=JR+J                                                           MINV1540
      A(JK)=-A(JI)                                                      MINV1550
  110 A(JI) =HOLD                                                       MINV1560
  120 J=M(K)                                                            MINV1570
      IF(J-K) 100,100,125                                               MINV1580
  125 KI=K-N                                                            MINV1590
      DO 130 I=1,N                                                      MINV1600
      KI=KI+N                                                           MINV1610
      HOLD=A(KI)                                                        MINV1620
      JI=KI-K+J                                                         MINV1630
      A(KI)=-A(JI)                                                      MINV1640
  130 A(JI) =HOLD                                                       MINV1650
      GO TO 100                                                         MINV1660
  150 RETURN                                                            MINV1680
      END                                                               MINV1690
      SUBROUTINE DS (VELOC,ANGLE,SS,SDS)
C     IF (ABS(VELOC).GT.999) SUM ONLY
C     VELOC IS NEGATED TO ALIGN VICE DELAY AND IS RETURNED UNCHANGED
C     FOR A GIVEN VELOCITY (CM/SEC) AND ANGLE (IN DEGREES FROM X AXIS
C     AND INPUT SS, THE DELAYED AND SUMMED OUTPUT IS GIVEN AS SDS.
      DIMENSION SS(32,4,4),SDS(32),S(32,4,4),GRAPHU(2000)
      DIMENSION GRAPH(2000),T(32)
      EQUIVALENCE (GRAPH(1),GRAPHU(1))
      COMMON/GRAPH/GRAPHU
      COMMON/DDS/JR,JS
      LOGICAL SD
      SD=.FALSE.
      IF(ABS(VELOC).LT.999.) SD=.TRUE.
      DEM=16.
      IF(.NOT.SD)DEM=1.
C
C     S WILL BE AVAILABLE AS 'DELAYED' VERSION OF SS AS PER VELOC AND ANGLE
2     DO 11 K=1,4
      DO 11 J=1,4
      DO 11 I=1,32
11    S(I,J,K)=SS(I,J,K)
C
 1    NT=32
      DT=0.02
      NX=4
      NY=4
C
      IF(SD)CALL DELAY (-VELOC,ANGLE,S)
      DO 5 I=1,NT
5     SDS(I)=0.0
      DO 6 I=1,NT
      DO 6 J=1,NX
```

```
      DO 6 K=1,NY
    6 SDS(I)=SDS(I)+S(I,J,K)/DEN
      DO 7 I=1,NT
    7 T(I)=(I-1.)/31.)*0.64
      YMIN=+1.E+10
      YMAX=-1.E+10
      IF(.NOT.SD) GO TO 46
      NPLT=NT-JB+JS
      NST=1+JB
      NED=NST+NPLT-1
      GO TO 47
   46 NST=1
      NED=NT
      NPLT=NT
   47 DO 8 I=NST,NED
      IF(SDS(I).GT.YMAX)YMAX=SDS(I)
    8 IF(SDS(I).LT.YMIN)YMIN=SDS(I)
      CALL PLOT(1,29,14,76,25)
      CALL PLOT2(GRAPH,J,.75,0.0,YMAX,YMIN)
      CALL PLOT3 (1H*,T(NST),SDS(NST),NPLT)
      PRINT 77
      CALL PLOT4(1,1H )
   77 FORMAT(1H1)
      IF(.NOT.SD) GO TO 60
      JS=-JS
      PRINT 78, VELOC,ANGLE,JS,JB
   78 FORMAT('0',20X,'DELAYED AND SUMMED OUTPUT W.R.T. ORIGIN',/,
     2 2X,'VELOCITY = ',F6.1,5X,'BEARING = ',F6.1,1X,' DEGREES',
     /23X,'LEFT SHIFT = ',I2,' NPTS.  RIGHT SHIFT = ',I2,' NPTS')
      CALL PLOT(1,51,10,101,10)
      RETURN
   60 PRINT 61
   61 FORMAT('0',32X,'SUMMED OUTPUT W.R.T. ORIGIN')
      CALL PLOT(1,51,10,101,10)
      RETURN
      END
      SUBROUTINE SHIFT (S,J,K,N)
      DIMENSION S(32,4,4)
      INTEGER KXS(16)/1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4/
      INTEGER KYS(16)/4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1/
C     RE SUBROUTINE DS
      KX=KXS(K)
      KY=KYS(K)
      IF(J.EQ.0) RETURN
      IF(IABS(J).GE.N) GO TO 1
      IF(J.LT.0) GO TO 3
C
C     J.GT.0, SHIFT RIGHT
      LF=1
      LL=N-J
      I=N+1
      L=N-J+1
      DO 4 L1=LF,LL
      L=L-1
      I=I-1
    4 S(I,KX,KY)=S(L,KX,KY)
      DO 5 I=1,J
    5 S(I,KX,KY)=0.0
      RETURN
C
C     J.LT.0, SHIFT LEFT
    3 J=-J
```

```
      LF=J+1
      LL=N
      I=0
      DO 6 L=LF,LL
      I=I+1
6     S(I,KK,KY)=S(L,KX,KY)
      LF=N-J+1
      DO 7 I=LF,N
7     S(I,KK,KY)=0.0
      RETURN
1     DO 2 I=1,N
2     S(I,KX,KY)=0.0
      RETURN
      END
      SUBROUTINE DELAY (VELOC,ANGLE,S)
      COMPLEX C(16)/
     (0.,6.),(2.,6.),(4.,6.),(6.,6.)
     ,(0.,4.),(2.,4.),(4.,4.),(6.,4.)
     ,(0.,2.),(2.,2.),(4.,2.),(6.,2.)
     ,(0.,0.),(2.,0.),(4.,0.),(6.,0.)/
      DIMENSION S(32,4,4)
      COMMON/ODS/JB,JS
1     NT=32
      DT=0.02
      NX=4
      NY=4
C
      PI=4.*ATAN(1.)
      ARG=ANGLE*PI/180.
      VX=VELOC*COS(ARG)
      VY=VELOC*SIN(ARG)
      ALPHAX=VX/(VELOC*VELOC)
      ALPHAY=VY/(VELOC*VELOC)
C
C     OUTPUT IS OF FORM SIT-SHIFT)
      JB=0
      JS=0
      DO 4 K=1,16
C     Q=TIME
      Q=ALPHAX*REAL(C(K))+ ALPHAY*AIMAG(C(K))
      J=IFIX(ABS(Q/DT)+0.501)
      IF(Q/DT.LT.0.0) J=-J
      IF(J.GT.JB)JB=J
      IF(J.LT.JS)JS=J
      CALL SHIFT(S,J,K,NT)
4     CONTINUE
      RETURN
      END
      SUBROUTINE PWAVE (SIGNAL,NPTS,DT,VELOC,ANGLE,SNR,ISEED,S3)
C     PWAVE MOD II
C     ISEED MUST BE AN ODD INTEGER
      DIMENSION SIGNAL(32),X(32),A(16,32)
      DIMENSION S3(32,4,4)
      COMPLEX C(16)/                                                    1/5
     (0.,6.),(2.,6.),(4.,6.),(6.,6.),                                   2/5
     (0.,4.),(2.,4.),(4.,4.),(6.,4.),                                   3/5
     (0.,2.),(2.,2.),(4.,2.),(6.,2.),                                   4/5
     (0.,0.),(2.,0.),(4.,0.),(6.,0.)/                                   5/5
C
C     NPTS OF SIGNAL SAMPLED AT HZ ARE RECEIVED AT THE ORIGIN WITH VELOCITY
C     (VELOC) AT ANGLE DEGREES FROM THE POS X AXIS.
C                         ON RETURN A(K,I) IS THE SIGNAL AT SENSOR K AT TIME I
```

```
C     WITH ADDITIVE GAUSSIAN NOISE OF SPECIFIED SNR (PEAK SIGNAL TO RMS NOISE).
C     ALL DISTANCES IN CM
C     ON RETURN S3(I,J,K) IS THE VALUE AT TIME=I*DT, X=J*DX, Y=K*DY
C
C     INITILZE ALL A TO ZERO IN CASE SENSORS MISS THE SIGNAL
      DO 1 K=1,16
      DO 1 I=1,NPTS
1     A(K,I)=0.0
      IF(SNR.LE.0.0) GO TO 7
C
C     T=TIME SHIFT IN SEC.  J=TIME INDEX SHIFT.
C     A'S ARE OF FORM SIGNAL(T-SHIFT)
      ARAD=ANGLE*4.*ATAN(1.0)/180.
      ALPHAX=COS(ARAD)/VELOC
      ALPHAY=SIN(ARAD)/VELOC
      HZ=1./DT
      DO 4 K=1,16
      T=ALPHAX*REAL(C(K)) +ALPHAY*AIMAG(C(K))
      J=IFIX(ABS(HZ*T)+0.501)
      IF(HZ*T.LE.0.0) J=-J
      IF (J.LE.0) GO TO 2
C     SHIFT SIGNAL RIGHT - DELAY
      LF=1
      LL=NPTS-J
      IF(LL.LE.2) GO TO 4
      I=J
      GO TO 3
C     SHIFT SIGNAL LEFT - ADVANCE
2     LF=-J+1
      IF(LF.GE.NPTS) GO TO 4
      LL=NPTS
      I=0
3     DO 4 L=LF,LL
      I=I+1
      A(K,I)=SIGNAL(L)
4     CONTINUE
C
C     ADD NOISE  NPTS=32 ONLY--
      IF(SNR.GT.98) GO TO 6
      SIGMAX=BIG(SIGNAL,NPTS)
      RMSN=SIGMAX/SNR
7     K=0
      ISTART=ISEED
      IEND=ISTART*32-1
      IF(SNR.LE.0.0) RMSN=1.0
      DO 5 IODD=ISTART,IEND,2
      K=K+1
C     CALL WHITE(S,RMSN,IODD,XN)
      CALL GAUSS(NPTS,0.0,RMSN,IODD,XN)
      DO 5 I=1,NPTS
5     A(K,I)=A(K,I)+XN(I)
C     CONVERT 2-3
6     DO 10 K=1,4
      L=12-(K-1)*4
      DO 10 J=1,4
      L=L+1
      DO 10 I=1,NPTS
10    S3(I,J,K)=A(L,I)
      RETURN
      END
      SUBROUTINE TAPER (DATA,NDATA)
      DIMENSION DATA(1)
```

```
      M=NDATA/10
      DATA(1)=0.0
      PI=4.*ATAN(1.)
      F=PI/(2*M)
      DO 3 I=2,M
3     DATA(I)=DATA(I)*SIN(F*I)
      N=NDATA-M
      DO 4 I=N,NDATA
      K=NDATA-I
4     DATA(I)=DATA(I)*SIN(F*K)
      RETURN
      END
      SUBROUTINE AHANN(ARRAY,NPTS)
C  NPTS= NUMBER OF COMPLEX POINTS, MAX = 128.
      DIMENSION ARRAY(1),X(128),Y(128)
      N=NPTS-1
      X(1)=ARRAY(1)/2.0+ARRAY(3)/2.0
      Y(1)=ARRAY(2)/2.0+ARRAY(4)/2.0
      DO 1 I=2,N
      A=ARRAY(2*I-3)/4.0
      B=ARRAY(2*I-1)/2.0
      C=ARRAY(2*I+1)/4.0
      X(I)=A+B+C
      A=ARRAY(2*I-2)/4.0
      B=ARRAY(2*I)/2.0
      C=ARRAY(2*I+2)/4.0
    1 Y(I)=A+B+C
      X(NPTS)=ARRAY(2*NPTS-3)/2.0+ARRAY(2*NPTS-1)/2.0
      Y(NPTS)=ARRAY(2*NPTS-2)/2.0+ARRAY(2*NPTS)/2.0
      DO 2 I=1,NPTS
      ARRAY(2*I-1)=X(I)
    2 ARRAY(2*I)=Y(I)
      RETURN
      END
      SUBROUTINE APLOT2(A,NPTS)
C
C     NPTS =   OF PTS OF A
C
      REAL IMAG(129),IMAX,IMIN
      COMMON/GRAPH/GRAPH
      DIMENSION A(1)
      DIMENSION       REAL(129),T(128),GRAPH(2000)
      N=NPTS/2+1
      N=N/2
      GO TO 100
      ENTRY APLOT(A,NPTS)
      N=NPTS
      N=N/2
  100 DO 101 I=1,N
      REAL(I)=A(2*I-1)
      IMAG(I)=A(2*I)
  101 T(I)=I
      RMAX=BIG(REAL,N)
      RMIN=SMALL(REAL,N)
      IMAX=BIG(IMAG,N)
      IMIN=SMALL(IMAG,N)
      TMAX=T(N)
      IF(RMAX.EQ.RMIN) GO TO 10
      WRITE(5,200)
      CALL PLOT1(2,21,10,101,10)
      CALL PLOT2(GRAPH,TMAX+1,1.0,RMAX,RMIN)
      CALL PLOT3('*',T,REAL,N)
```

```
      CALL PLOT4(1,1H )
      WRITE(5,201)
 10   IF(IMAX.EQ.IMIN) GO TO 11
      CALL PLOT2(GRAPH,TMAX+1,1.0,IMAX,IMIN)
      CALL PLOT3(1H*,T,IMAG,N)
      CALL PLOT4(1,1H )
      WRITE(5,202)
  200 FORMAT('1')
 201  FORMAT('0',50X,'REAL',//)
  202 FORMAT('0',45X,'IMAGINARY')
 11   CALL PLOT1(0,51,10,101,10)
      RETURN
      END
      SUBROUTINE PLOT11 (S)
      DIMENSION S(32,4,4),SDS(32),T(32),GRAPH(2000)
      COMMON/GRAPH/GRAPH
      DO 1 I=1,32
      T(I)=(I-1.)/31.)*0.64
 1    SDS(I)=S(I,1,1)
      YMIN=+1.E+20
      YMAX=-1.E+20
      DO 2 I=1,32
      IF(SDS(I).GT.YMAX)YMAX=SDS(I)
 2    IF(SDS(I).LT.YMIN)YMIN=SDS(I)
      PRINT 77
      CALL PLOT1(1,29,14,76,25)
      CALL PLOT2(GRAPH,0.79,3.0,YMAX,YMIN)
      CALL PLOT3(1H*,T,SDS,32)
      CALL PLOT4(1,1H )
      CALL PLOT1(1,51,10,101,10)
      PRINT 78
      RETURN
 77   FORMAT(1H1)
 78   FORMAT('0',30X,'RECEIVED WAVEFORM AT ORIGIN')
      END
      SUBROUTINE PLOT16 (A,NT)
C  SUBROUTINE PLOT16 PLOTS A 4X4 DISPLAY  WITH NPTS X 4 X 4 POINTS
C  IN THE INPUT ARRAY, E.G. 4X4X32=512 POINTS
C     WHERE NT = NO OF POINTS IN TIME E.G. NTX4X4
C
C
      IMPLICIT LOGICAL*1 (L)
      DIMENSION A(NT,4,4)
      DIMENSION AA(512),X(131),Y(131),LGRAPH(131,20)
      DATA LBK,LSOLID,LDASH,LDOT/' ','*','-','I' /
      COMMON/GRAPH/AA,X,Y,LGRAPH
C
C THIS SECTION SETS UP NECESSARY CONSTANTS
      NPTS=NT*4*4
      NHL=14
      NSHL=NHL-1
      NVL=131
      NSVL=32
      N=NPTS/16
      NX=NPTS/4 + 3
C
C  THIS SECTION GENERATES A 1-DIMENSIONAL ARRAY FROM THE INPUT ARRAY AND
C   FINDS MAXIMUMS AND MINIMUMS
      IA=1
      K=5
      DO 99 IY=1,4
      K=K-1
```

```
      DO 99 IX=1,4
      DO 99 IT=1,N
      IA=IA+1
   99 AA(IA)=A(IT,IX,K)
      AMAX=0.0
      AMIN=0.0
      DO 80 I=1,NPTS
      IF(AA(I).LT.AMIN) AMIN=AA(I)
      IF(AA(I).GT.AMAX) AMAX=AA(I)
   80 CONTINUE
      C=AMAX-AMIN
      DO 100 I=1,NX
      Y(I)=0.0
  100 X(I)=I
      XMAX=NX
      XMIN=1.0
      SCALEX=130.0/(XMAX-XMIN)
C
      WRITE(6,200)
  200 FORMAT('1')
C  OUTER LOOP TO PLOT EACH OF THE 4 HORIZ ROWS OF 4 GRAPHSO
      INT=0
      DO 107 IOUT=1,4
C
C  THIS SECTION SETS UP GRID LINES FOR 1 ROW OF 4 PLOTSO
      DO 102 IY=1,NHL
      DO 102 IX=1,131
  102 LGRAPH(IX,IY)=LBK
      IB=-32
      DO 104 I=1,4
      IB=IB+33
      IE=IB+31
      DO 104 IX=IB,IE
      LGRAPH(IX,1)=LDASH
  104 LGRAPH(IX,NHL)=LDASH
      IB=-32
      DO 103 I=1,4
      IB=IB+33
      IE=IB+31
      DO 103 IY=1,NHL
      LGRAPH(IB,IY)=LBAR
  103 LGRAPH(IE,IY)=LBAR
C
      DO 101 I=1,N
      Y(I)=(AA(I+INT)-AMIN)/C
      K=I+N+1
      Y(K)=(AA(K-1+INT)-AMIN)/C
      K=I+2*N+2
      Y(K)=(AA(K-2+INT)-AMIN)/C
      M=I+3*N+3
  101 Y(M)=(AA(M-3+INT)-AMIN)/C
C
      DO 105 IO=1,131
      IF(X(IO).GT.XMAX.OR.X(IO).LT.1.0)  GO TO 107
      IF (Y(IO).GT.1.0.OR.Y(IO).LT.0.0)  GO TO 107
      IX=(X(IO)-1.0)*SCALEX+1.5
      IY=Y(IO)*(NHL-1)+1.5
  105 LGRAPH(IX,IY)=LCHAR
      DO 106 IY=1,NHL
      DO 106 IX=33,99,33
  106 LGRAPH(IX,IY)=LBK
C
```

```
      J=NHL+1
      DO 106 IY=I,NHL
      J=J-1
      WRITE(6,201) (LGRAPH(IX,J), IX=1,131)
  106 CONTINUE
  201 FORMAT (1X,131A1)
C
  107 INT=INT+NPTS/4
      WRITE(6,300) AMIN,AMAX
  300 FORMAT(' ',/,50X,'MIN =',E14.7,'  MAX =',E14.7)
      RETURN
      END
      SUBROUTINE FWN05 (SS,ITAP,IHAMN,PW)
C     ENTER HERE TO EXAMINE FREQ DOMAIN AMLP RESP FOR GIVEN IMPULSE SS.
      DIMENSION A(32),AOUT(128)
      LOGICAL*1 LFLAG
      LFLAG=.TRUE.
      GO TO 4
      ENTRY FWNO (SS,ITAP,IHAMN,PW)
C     ENTER HERE TO EXAMINE POWER SPECT. OF GIVEN TIME SERIES SS.
      LFLAG=.FALSE.
4     CONTINUE
C     RE SUBROUTINE PLT3
      DIMENSION SS(32,4,4),TEMP(32),H(3),A(4096),PW(17,8,8)
      DIMENSION S3(32,8,8)
        DIMENSION S(512),INV(512)
      COMMON/DIM/NT,DT,DX,DY
      DIMENSION AU(4096),SU(512),INVU(512)
      EQUIVALENCE (A,AU),(S,SU),(INV,INVU)
      COMMON/U/TEMP,S3,AU,SU,INVU
      NT=32
      NX=8
      NY=8
C
C     CALL NEGATE(SS)          DO THIS IN CALLING PGM - BEFORE FWNO
C
C     TIME TAPER - COSINE 10 PERCENT
      IF(ITAP.NE.1) GO TO 100
      DO 1 J=1,4
      DO 1 K=1,4
      DO 111 I=1,NT
  111 TEMP(I)=SS(I,J,K)
      CALL TAPER (TEMP,NT)
      DO 1 I=1,NT
1     SS(I,J,K)=TEMP(I)
C
C     SPACE TAPER - .5,1,1,.5
      DO 2 I=1,NT
      DO 21 J=1,4
      DO 21 K=1,4,3
21    SS(I,J,K)=0.5*SS(I,J,K)
      DO 2 J=1,4,3
      DO 2 K=2,3
2     SS(I,J,K)=0.5*SS(I,J,K)
100   CONTINUE
      IF(LFLAG) GO TO 5
C
C     PAD WITH ZEROS IN X-Y
      DO 3 I=1,32
      DO 31 K=1,4
      DO 31 J=1,4
31    SS(I,J,K)=SS(I,J,K)
```

```
      DO 32 K=1,8
      DO 32 J=5,8
32    S3(I,J,K)=0.0
      DO 3 K=5,8
      DO 3 J=1,4
3     S3(I,J,K)=0.0
      GO TO 5
C     ADD ZEROS TO TAILS OF IMPULSE RESP.
5     CONTINUE
      COMMON/NS/NX4,NXF,NX4F,NXF1,NZX,
     *          NY4,NYF,NY4F,NYF1,NZY
      NX4=8
      NY4=8
      NXF=3
      NYF=3
      NX4F=5
      NY4F=5
      NXF1=4
      NYF1=4
      NZX=4
      NZY=4
      DO 9 I=1,17
      N=-1
      DO 91 K=1,4
      DO 91 J=1,4
      N=N+2
      AIN(N+1)=0.0
91    AIN(N)=S3(I,J,K)
      CALL EXPAND(AIN,4,4,AOUT)
      N=-1
      DO 9 K=1,8
      DO 9 J=1,8
      N=N+2
      NN=2*((K-1)*NT*8+(J-1)*NT+(I-1))+1
      A(NN)=AOUT(N)
9     A(NN+1)=0.0
C     FORM 2ND TIME
C     ROUTINE VALID ONLY FOR (NT,8,8) ARRAYS
      NTF=(NT/2)+1
      NTF1=NTF+1
      NTF2=NTF-1
      INTEGER ICON(8)/1,6,7,8,5,2,3,4/
      DO 7 KC=1,8
      K=ICON(KC)
      DO 7 JC=1,8
      J=ICON(JC)
      IRO=2*((K-1)*NT*8+(JC-1)*NT+(NTF1-1))+1
      IRO=IRO-2
      IIO=IRO+1
      IRO=2*((K-1)*NT*8+(J-1)*NT+(NTFM-1))+1
      IRO=IRO+2
      IIO=IRO+1
      DO 7 IC=NTF1,NT
      IRO=IRO+2
      IIO=IIO+2
      IRO=IRO-2
      IIO=IIO-2
      A(IRO)=A(IRO)
7     A(IIO)=-A(IIO)
6     CONTINUE
C
C     XFM
```

```
      IFSET=+1
      M(1)=5
      M(2)=3
      M(3)=3
      NX=8
      NY=8
      IF(ILFLAG) GO TO 10
      N=-1
      DO 11 K=1,NY
      DO 11 J=1,NX
      DO 11 I=1,NT
      N=N+2
      A(N)=S3(I,J,K)
11    A(N+1)=0.0
10    CALL HARMIA,M,INV,S,IFSET,IFERR)
C
C     FORM PW ARRAY
C     CALCULATES RAW POWER FROM COMPLEX 3-D FREQ WAVENUMBER SPECTRUM
C     ORDER IN PW ARRAY ISO MIN(NEGI-ZERO-MAX(POS)
C     PW(IF,KX,KX) (17,8,9)
      IF=(NT/2)+1
      DO 12 K=1,NY
      DO 12 J=1,NX
      N=2*((K-1)*NT*NX+(J-1)*NT+(1-1))+1
      N=N-2
      DO 12 I=1,IF
      N=N+2
      PW(I,J,K)=A( N )*A( N )+A(N+1)*A(N+1)
      PW(I,J,K)=PW(I,J,K)/(NT*DT*4*DX*4*DY)
12      CONTINUE
C     PUT NEG PW FIRST W.R.T. K  SQUARE ARRAYS ONLY
      NF=(NX/2)+1
      N=(NT/2)+1
      JS=NX-NF
      DO 40 IF=1,N
      DO 30 J=1,NX
      DO 30 JC=1,JS
      T=PW(IF,J,NY)
      KY=NX+1
      DO 20 IC=2,NX
      KY=KY-1
20    PW(IF,J,KY)=PW(IF,J,KY-1)
30    PW(IF,J,1)=T
      DO 40 J=1,NX
      DO 40 JC=1,JS
      T=PW(IF,NY,J)
      KX=NX+1
      DO 50 IC=2,NY
      KX=KX-1
50    PW(IF,KX,J)=PW(IF,KX-1,J)
40    PW(IF,1,J)=T
C
C     SMOOTH
      CALL HAMN3(PW,N,NX,NY,IHAMN)
C
C     CALL PLT3 (PW,RLF,HF,ISKP,IFPLT)      DO IN CALL PGM - AFTER FWND
C     FOR IMPULSE WORK CALL PLTS INSTEAD
C
      RETURN
      END
      SUBROUTINE PLTS (PW,RLF,HF,ISKP,IFPLT)
C
```

```
      COMMON/P5/IM,JM,KM
C     ENTER HERE TO NORMALIZE W. R. T. PW(IM,2*JM,2*KM)
C     WHICH IS A SPECIFIED PASSBAND POINT
C     INDECES IN 'NOT-HARM'*(-,+) ORDER
      LOGICAL*1 LFLAG
      LFLAG=.TRUE.
      GO TO 4
      ENTRY PLT3 (PW,NLF,NF,ISKP,IFPLT)
C     ENTER HERE TO NORMALIZE W.R.T. MAX OF PW
      LFLAG=.FALSE.
4     CONTINUE
C     OUTPUT EVERY ISKP-TH FREQ MATRIX FROM FREQ = NLF TO NF
      DIMENSION IRO(8)
      DIMENSION PW(17,8,8),A(64)
      DIMENSION IPW(9,8)
      DIMENSION FREQ(17),WK(8),V(17,8)
      DIMENSION IGW(64)
      COMMON/DIM/NT,DT,DX,DY
      COMMON/DU/IRJ,A
      EQUIVALENCE (IPW,IGW)
      NX=8
      NY=9
      CON=180./(4.*ATAN(1.0))
      FF=.5/DT
      NFF=(NT/2)+1
      NHF=((NF/FF)*FLOAT(NFF)+0.501)
      IF(NHF.GT.NFF) NHF=NFF
      NLF=((NLF/FF)*FLOAT(IFF)+0.501)
      IF(NLF.LT.1) NLF=1
C     GET MAX-PW IN PASSBAND   DON'T SKIP
      IF(LFLAG)  GO TO 5
      PWM=0.0
      DO 55 J=1,NX
      DO 55 K=1,NY
      DO 55 I=NLF,NHF
55    IF(PWM.LT.PW(I,J,K))PWM=PW(I,J,K)
      GO TO 6
5     PWM=PW(IM,2*JM,2*KM)
6     CONTINUE
      CALL SCALES(FREQ,WK,V)
C
C     ENTER HERE TO USE PW-MAX OF LAST PLT3 CALL
      ENTRY PLT4(PW)
      IF(ISKP.LE.0)ISKP=1
      DO 1 I=NLF,NHF,ISKP
      DO 1 J=1,NX
      DO 1 K=1,NY
      PW(I,J,K)=PW(I,J,K)/PWM
      IF(PW(I,J,K).LE.1.E-9) PW(I,J,K)=1.E-9
      PW(I,J,K)=-10.*ALOG10(PW(I,J,K))
C     GET MAX-PW W.R.T. FREQ
      IF(PW(I,J,K).EQ.0.0) PMAX=FREQ(I)
1     CONTINUE
C
      NXY=NX*NY
      DO 35 I=NLF,NHF,ISKP
      SM=100.
      DO 3 J=1,NX
      DO 3 K=1,NY
      IF(PW(I,J,K).GE.SM) GO TO 3
      SM=PW(I,J,K)
      KK=J
```

```
      KY=K
3     CONTINUE
      IF(WK(KX).EQ.0.0.AND.WK(KY).EQ.0.0) GO TO 10
      TH=ATAN2(WK(KY),WK(KX))*CON
      GO TO 11
10    TH=100000.
11    CONTINUE
      ABSK=SQRT(WK(KX)*WK(KX)+WK(KY)*WK(KY))
      IF(ABSK.NE.0.0) GO TO 112
      VE=1.E+10
      GO TO 12
112   VE=FREQ(I)/ABSK
12    CONTINUE
      DO 61 M=1,NX
      DO 61 N=1,NY
61    IPW(M,N)=PW(I,M,N)
      PRINT 2
2     FORMAT('1')
      PRINT 60,FREQ(I),FMAX,PWM
   60 FORMAT(2X,'PW MATRIX FOR TEMPORAL FREQ =',F4.1,' HZ',/,2X,'IN -DB
     W.R.T. PEAK POWER AT  ',F4.1,' HZ',20X,'K',13X,'VELOC',/,2X,'OF',
     E14.7,'  PICOWATT*SEC*CM**2',/)
      IRN=9
      IRA=5
      DO 25 IR=1,8
      IRA=IRA-1
      IRN=IRN-1
   25 WRITE(6,59) (IPW(IC,IRN),IC=1,8),IRA,WK(IRN),V(I,IRN)
   59 FORMAT(8(2X,I3),4X,I2,8X,F8.3,12X,F7.2,//)
      IRA=-4
      DO 690 IC=1,8
      IRA=IRA+1
690   IR(IC)=IRA
      PRINT 6900,(IR(IW),IW=1,NX)
 6900 FORMAT(/,8(2X,I3)//)
      IF(IPPLT.LT.2)  GO TO 6901
      CALL GRAY(IW,4,4,8,2)
 6901 PRINT 31,VE,TH
31    FORMAT('  ',10X,'AT LOCAL MAXD VELOC =',F7.2,1X,'CM/SEC,  ','BEARIN
     G =',F5.0,' DEGREES')
35    CONTINUE
      RETURN
      END
      SUBROUTINE FILTER(SS,FANG,FVEL,FFQ)
C     ALWAYS NEGATE SS BEFORE FILTER
C     RE SUBROUTINES FF,FA,FV
C     I=TEMPORAL FREQ
C     J=KX
C     K=KY
      INTEGER ICON(4)/1,4,3,2/
      DIMENSION SS(32,4,4),A(1024),M(3 ),S(256),INV(256)
      DIMENSION FANG(4,4),FVEL(17,4,4),FFQ(17)
      DIMENSION AU(1024),SU(256),INVU(256)
      EQUIVALENCE (A,AU),(S,SU),(INV,INVU)
      COMMON/DATA/SU,IU/0
      COMMON/DIM/NT,DT,DX,DY
      NT=32
      NX=4
      NY=4
      N=-1
      DO 15 K=1,NY
      DO 15 J=1,NX
```

```
      DO 15 I=1,NT
      N=N+2
      A(N)=SS(I,J,K)
 15   A(N+1)=0.0
C
C     FWD XFM
      M(1)=5
      M(2)=2
      M(3)=2
      IFSET=1
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP2                                        STOP2
C
C     MULTIPLY IN FREQ-WAVE DOMAIN
      IF=(NT/2)+1
C     FLTR IRE AND IIM POS TEMPORAL FREQ WITH FLTR ARRAY
      DO I K=1,NY
      DO I J=1,NX
      N=2*((K-1)*NT*NX+(J-1)*NT+(1-1))+1
      N=N-1
      DO 1 I=1,IF
      DO 1 L=1,2
      N=N+1
      A(N)=A(N)*FANG(J,K)*FVEL(I,J,K)*FFQ(I)
 1    CONTINUE
C
C     DO NEG TEMPORAL FREQ
      IF1=IF+1
      DO 2 KC=1,NY
      K=ICON(KC)
      DO 2 JC=1,NX
      J=ICON(JC)
      IRU=2*((KC-1)*NT*NX+(JC-1)*NT+(1B-1))+1
      IRU=IRU-2
      IIU=IRU+1
      IRD=2*((K-1)*NT*NX+(J-1)*NT+(16-1))+1
      IRD=IRD+2
      IID=IRD+1
      DO 2 IC=IF1,NT
      IRU=IRU+2
      IIU=IIU+2
      IRD=IRD-2
      IID=IID-2
      A(IRU)=A(IRD)
      A(IIU)=-A(IID)
 2    CONTINUE
102   CONTINUE
C
C     INV XFM
      IFSET=-2
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP3                                        STOP3
103   CONTINUE
      IRE=-1
      DO 25 K=1,NY
      DO 25 J=1,NX
      DO 25 I=1,NT
      IRE=IRE+2
101   CONTINUE
25    SS(I,J,K)=A(IRE)
      RETURN
      END
```

```
      SUBROUTINE NEGATE (S3)
C     NEGATE ABOUT X AND Y AXES
      DIMENSION S3(32,4,4),T(32)
      COMMON/DIM/NT,DT,DX,DY
      COMMON/U/T
      NX=4
      NY=4
      DO 10 I=1,NT
      DO 11 J=1,NX
      KD=NY+1
      DO 12 K=1,NY
      KD=KD-1
12    T(KD)=S3(I,J,K)
      DO 13 K=1,NY
13    S3(I,J,K)=T(K)
11    CONTINUE
      DO 10 K=1,NY
      JD=NX+1
      DO 14 J=1,NX
      JD=JD-1
14    T(JD)=S3(I,J,K)
      DO 15 J=1,NX
15    S3(I,J,K)=T(J)
10    CONTINUE
      RETURN
      END
      SUBROUTINE PWAVE (SIGNAL,NPTS,DT,VELOC,ANGLE,SNR,ISEED,S3)
C     PWAVE MOD II
C     ISEED MUST BE AN ODD INTEGER
      DIMENSION SIGNAL(32),X(32),A(16,32)
      DIMENSION S3(32,4,4)
      COMMON/U/A
      COMPLEX C(16)/                                                    1/5
     (0.,6.),(2.,6.),(4.,6.),(6.,6.),                                   2/5
     (0.,4.),(2.,4.),(4.,4.),(6.,4.),                                   3/5
     (0.,2.),(2.,2.),(4.,2.),(6.,2.),                                   4/5
     (0.,0.),(2.,0.),(4.,0.),(6.,0.)/                                   5/5
C
C     NPTS OF SIGNAL SAMPLED AT HZ ARE RECEIVED AT THE ORIGIN WITH VELOCITY
C     (VELOC) AT ANGLE DEGREES FROM THE POS X AXIS.
C                          ON RETURN A(K,I) IS THE SIGNAL AT SENSOR K AT TIME I
C     WITH ADDITIVE GAUSSIAN NOISE OF SPECIFIED SNR (PEAK SIGNAL TO RMS NOISE).
C     ALL DISTANCES IN CM
C     ON RETURN S3(I,J,K) IS THE VALUE AT TIME=I*DT, X=J*DX, Y=K*DY
C
C     INITIALZE ALL A TO ZERO IN CASE SENSORS MISS THE SIGNAL
      DO 1 K=1,16
      DO 1 I=1,NPTS
1     A(K,I)=0.0
      IF(SNR.LE.0.0) GO TO 7
C
C     T=TIME SHIFT IN SEC.  J=TIME INDEX SHIFT.
C     A'S ARE OF FORM SIGNAL(T-SHIFT)
      ARAD=ANGLE*4.*ATAN(1.0)/180.
      ALPHAX=COS(ARAD)/VELOC
      ALPHAY=SIN(ARAD)/VELOC
      HZ=1./DT
      DO 4 K=1,16
      T=ALPHAX*REAL(C(K)) +ALPHAY*AIMAG(C(K))
      J=IFIX(ABS(HZ*T)+0.501)
      IF(HZ*T.LE.0.0) J=-J
      IF (J.LE.0) GO TO 2
```

```
C     SHIFT SIGNAL RIGHT - DELAY
      LF=1
      LL=NPTS-J
      IF(LL.LE.2) GO TO 4
      I=J
      GO TO 3
C     SHIFT SIGNAL LEFT - ADVANCE
2     LF=-J+1
      IF(LF.GE.NPTS) GO TO 4
      LL=NPTS
      I=0
3     DO 4 L=LF,LL
      I=I+1
      A(K,I)=SIGNAL(L)
4     CONTINUE
C
C     ADD NOISE  NPTS=32 ONLY--
      IF(SNR.GT.98) GO TO 6
      SIGMAX=BIG(SIGNAL,NPTS)
      RMSN=SIGMAX/SNR
7     K=0
      ISTART=ISEED
      IEND=ISTART+32
      IF(SNR.LE.0.0) RMSN=1.0
      DO 5 IDUD=ISTART,IEND,2
      K=K+1
      CALL GAUSS(NPTS,0.0,RMSN,IDUD,XN)
      DO 5 I=1,NPTS
5     A(K,I)=A(K,I)+XN(I)
C     CONVERT 2-3
6     DO 10 K=1,4
      L=12-(K-1)*4
      DO 10 J=1,4
      L=L+1
      DO 10 I=1,NPTS
10    S3(I,J,K)=A(L,I)
      RETURN
      END
      SUBROUTINE FF (RLF,HF,IBP,FFO,IHAMN,IFPLT)
C     RE SUBROUTINE FILTER
      DIMENSION A(256),S(32),INV(32),H(65),FREQ(65)
      DIMENSION M(3)
      DIMENSION FFQ(17),GRAPH(2000)
      DIMENSION AU(256),SU(32),INVU(32),GRAPHU(2000)
      EQUIVALENCE (A,AU),(S,SU),(INV,INVU),(GRAPH,GRAPHU)
      COMMON/UU/AU,SU,INVU,GRAPHU
L
C     ACTUAL
      NT=32
      NF=(NT/2)+1
      NF1=NF+1
      DT=0.02
      FF=.5/DT
C
C     SET IBP=1 FOR BANDPASS FILTER  IBP=0 YIELDS BANDSTOP FILTER
      IBP=1
      NHF=((HF/FF)*FLOAT(NF-1)+0.501)
      NLF=((RLF/FF)+FLOAT(NF-1)+0.501)
      IF(NLF.GE.NF-1 STOP72                                          STUP72
C
C     ZERO LOWER STOPBAND
      IF(NLF.LE.0) GO TO 2
```

```
      N=-1
      DO 1 I=1,NLF
      N=N+2
      A(N)=0.0
1     A(N+1)=0.0
C
C     SET PASSBAND TO UNITY
2     IF(NHF.GE.NF ) NHF=NF
      NLF1=NLF+1
      N=2*(NLF1-1)+1
      N=N-2
      DO 3 I=NLF1,NHF
      N=N+2
      A(N)=1.0
3     A(N+1)=0.0
C
C     ZERO UPPER STOPBAND
      IF(NHF.EQ.NF ) GO TO 5
      NHF1=NHF+1
      N=2*(NHF1-1)+1
      N=N-2
      DO 4 I=NHF1,NF
      N=N+2
      A(N)=0.0
4     A(N+1)=0.0
C
C     REVERSE STOP AND PASS IF IBP=0
5     IF(IBP.EQ.1) GO TO 6
      N=-1
      DO 7 I=1,NF
      N=N+2
      IF(A(N).EQ.0.0) GO TO 8
      A(N)=0.0
      GO  TO 7
8     A(N)=1.0
7     CONTINUE
6     CONTINUE
C
C     SMOOTH
      IF(IHAMN.EQ.0) GO TO 65
      N=-1
      DO 225 I=1,NF
      N=N+2
225   H(I)=A(N)
      CALL HAMN(H,NF,IHAMN)
      N=-1
      DO 226 I=1,NF
      N=N+2
226   A(N)=H(I)
      DO 227 I=1,17
227   FFG(I)=H(I)
      IF(IFPLT.EQ.0) RETURN
65    CONTINUE
C
C
C     GEN NEG FREQ
      NF1=NF+1
      N=2*(NF1-1)+1
      N=N-2
      L=2*(I (F-1)-1)+1
      L=L+2
      DO 13 I=NF1,NT
```

```
      N=N+2
      L=L-2
      A(N)=A(L)
      A(N+1)=-A(L+1)
13      CONTINUE
C
C     EXPANDED
      NT4=4*NT
      DT4=DT/4.0
      NF4=(NT4/2)+1
C
C     INV XFM
      M(1)=5
      M(2)=0
      M(3)=0
      IFSET=-1
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP73                                    STOP73
C
C     APPLY WINDOW HERE INSTEAD OF CALL HAMM IF DESIRED
C     IW=1
C     CALL WINCON(A,NT,0.32,IW)
C
C     ADD 3X ZEROS
      NZ=3*NT-1
      NB=(NF+1)+NZ
      N=2*(NT-1)+1
      N=N+2
      K=2*(NT4-1)+1
      K=K+2
      DO 14 I=NB,NT4
      K=K-2
      N=N-2
      A(K)=A(N)
14      A(K+1)=A(N+1)
      N=2*(NF1-1)+1
      N=N-2
      NE=NF1+NZ-1
      DO 40 I=NF1,NE
      N=N+2
      A(N)=0.0
40    A(N+1)=0.0
C
C     FWD XFM
      IFSET=+1
      M(1)=7
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP74                                    STOP74
C
C     PLOT ACTUAL RESPONSE
      N=-1
      HMAX=0.
      DO 15 I=1,NF4
      FREQ(I)=((I-1.)/(FLOAT(NF4)-1.))*FF
      N=N+2
      H(I)=SQRT(A(N)*A(N)+A(N+1)*A(N+1))
15    IF(H(I).GT.HMAX) HMAX=H(I)
      DO 16 I=1,NF4
      H(I)=H(I)/HMAX
      IF(H(I).LE.10.E-4) H(I)=10.E-4
      H(I)=20.*ALOG10(H(I))
16    IF(H(I).GE.20.) H(I)=20.
```

```
      NPTS=NF4
      PRINT 77
 77   FORMAT(1H1)
           CALL PLOT2(GRAPH,25.,0.0,20.,-80.)
      CALL PLOT3(1H*,FREQ,H ,NPTS)
      CALL PLOT4 (30,30HRECIBELS                                    )
      PRINT 155
 155  FORMAT(/,35X,'AMPLITUDE RESPONSED FREQUENCY FILTER')
      RETURN
      END
      SUBROUTINE SCALES (FREQ,WK,V)
C     RC SUBROUTINE PLT3
C     VALID ONLY WHEN SENSORS FORM SQUARE ARRAY
C     FREQ IS ORDERED ZERO TO PJS FOLDING FREQ
C     WK AND VELOCITY (V) ARE ORDERED NEG-ZERO-POS
      DIMENSION FREQ(17),WK(8),V(17,8)
      COMMON/DIM/NT,DT,DX,DY
      NX=8
      NY=8
      RT=DT*NT
      RX=DX*NX
      DF=1./RT
      DWK=1./RX
C     FREQ SCALE
      N=(NT/2)+1
      DO 1 I=1,N
1     FREQ(I)=(I-1.)*DF
C     GEN POS K
      N=(NX/2)+1
      DO 2 J=1,4
2     WK(J)=(J-1)*DWK
C     GEN NEG K
      K=N
      N=N+1
      DO 5 I=N,NX
      K=K-1
5     WK(I)=-WK(K)
C     PUT NEG K FIRST
      N=(NX/2)+1
      JS=NX-1
      DO 4 JC=1,JS
      T=WK(NX)
      IK=NX+1
      DO 3 I=2,IK
      IK=IK-1
3     WK(IK)=WK(IK-1)
4     WK(1)=T
C     VELOCITY SCALE
      NF=(N1/2)+1
      DO 6 IF=1,NF
      DO 6 IK=1,NX
      IF(WK(IK).EQ.0.0) V(IF,IK)=1.E+5
      IF(WK(IK).EQ.0.0) GO TO 6
      V(IF,IK)=FREQ(IF)/WK(IK)
6     CONTINUE
      RETURN
      END
      SUBROUTINE HAMN (DATA,NDATA,IHAMN)
      DIMENSION DATA(1)
      DIMENSION DT(1024)
      COMMON/D/DT
C     IHAMN=1 IS HAMMING WINDOW .22,.56,.22
```

```
C       IHAMN=2 0 HANNING WINDOW .25,.50,.25
        REAL HA(2,3)/.22,.25,.56,.50,.22,.25/
        N=NDATA-1
        DT(1)=DATA(1)/2.+DATA(2)/2.
        DT(NDATA)=DATA(N)/2.0+DATA(NDATA)/2.0
        DO 1 I=2,N
        A=(DATA(I-1))*HA(IHAMN,1)
        B=(DATA(I))*HA(IHAMN,2)
        C=(DATA(I+1))*HA(IHAMN,3)
1       DT(I)=A+B+C
        DO 2 I=1,NDATA
2       DATA(I)=DT(I)
        RETURN
        END
        SUBROUTINE HAMN3 (S,NT,NX,NY,IHAMN)
        DIMENSION S(NT,NX,NY),T(32)
        IF(IHAMN.EQ.0) RETURN
C
C       SMOOTH I FOR EACH (J,K)
        DO 1 K=1,NY
        DO 1 J=1,NX
        DO 2 I=1,NT
2       T(I)=S(I,J,K)
        CALL HAMN (T,NT,IHAMN)
        DO 1 I=1,NT
1       S(I,J,K)=T(I)
        RETURN
        END
        SUBROUTINE FV (RLV,NV,IBP,BEAM,IHAMN,IFPLT)
        DIMENSION IPW(33,17),PW(33,17)
        DIMENSION BEAM(17,4,4),A(4096),INV(512),S(512),M(3),AS(256)
        DIMENSION AU(4096),INVU(512),SU(512),ASU(256)
        DIMENSION IGW(561)
        EQUIVALENCE (A,AU),(INV,INVU),(S,SU),(AS,ASU)
        COMMON/UAU,INVU,SU,ASU,PW
        DIMENSION U2(4,4)
        NY=4
        DX=2.0
        NX=4
C       N.B. EXPANDING X BY 8
        NX4=NX*8
        NXF=(NX/2)+1
        NX4F=(NX4/2)+1
        NXF1=NXF+1
        NZX=NX4-NX
        NT=32
C       N.B. EXPANDING F BY 2
        NT4=NT*2
        NTF=(NT/2)+1
        NT4F=(NT4/2)+1
        NTF1=NTF+1
        NZT=NT4-NT
        DT=0.02
        DF=1./(NT*DT)
        NF=NT
C
C       DEFINE BEAM IN 'HARM' ORDER
        IF(RLV.GE.250.) GO TO 60
        REAL X(4)/0.0,0.125,0.25,-0.125/
        REAL Y(4)/0.0,0.125,0.25,-0.125/
        U=0.5*DF
        DO 1 K=1,NX
```

```
      DO 1 J=1,NY
      DO 1 I=1,NTF
      F=(I-1)*DF
      FH=Q*SQRT(X(K)**2+Y(J)**2)*HV
      FL=-Q*SQRT(X(K)**2+Y(J)**2)*RLV
      IF(F.LT.FH.AND.F.GT.FL) GO TO 2
      BEAM(I,J,K)=0.0
      GO TO 1
    2 BEAM(I,J,K)=1.0
    1 CONTINUE
      GO TO 61
60    DO 62 I=1,NTF
      DO 62 J=1,NY
      DO 62 K=1,NX
62    BEAM(I,J,K)=0.0
      DO 63 I=1,NTF
63    BEAM(I,1,1)=1.0
61    CONTINUE
C
C     INVERT STOP AND PASS
      IF(IOP.EQ.1) GO TO 50
      DO 55 K=1,NY
      DO 55 J=1,NX
      DO 55 I=1,NTF
      IF(BEAM(I,J,K).EQ.0.0)  GO TO 550
      BEAM(I,J,K)=0.0
      GO TO 55
  550 BEAM(I,J,K)=1.0
50    CONTINUE
C
C     SMOOTH
      IF(IHAMM.EQ.0) GO TO 40
C     PUT REG 1ST
      DO 30 I=1,NTF
      DO 31 K=1,NY
      DO 31 J=1,NX
31    D2(J,K)=BEAM(I,J,K)
      CALL RFT(D2,NX,NY)
      DO 32 K=1,NY
      DO 32 J=1,NX
32    BEAM(I,J,K)=D2(J,K)
30    CALL HAMMB(BEAM,NX,NY,NTF,IHAMM)
C     PUT PHS 1ST
      DO 40 I=1,NTF
      DO 41 K=1,NY
      DO 41 J=1,NX
41    D2(J,K)=BEAM(I,J,K)
      CALL REVREV(D2,NX,NY)
      DO 42 K=1,NY
      DO 42 J=1,NX
42    BEAM(I,J,K)=D2(J,K)
40    CONTINUE
      IF(IFPLT.EQ.0) RETURN
C
C     NOTE MAX IN SPECIFIED PASS BAND
      PWM=0.0
      K=1
      DO 56 J=1,NX
      DO 56 I=1,NTF
      IF(PWM.LT.BEAM(I,J,K)) GO TO 57
      GO TO 56
57    PWM=BEAM(I,J,K)
```

```
      JM=J
      IM=I
   56 CONTINUE
C
C     PUT BEAM INTO 'AS' (1ST QUAD)
      K=1
      DO 3 J=1,NXF
      DO 3 I=1,NTF
      N=2*((J-1)*NT+(I-1))+1
      AS(N)=BEAM(I,J,K)
    3 AS(N+1)=0.0
C
C     ROTATE ABOUT X AXIS
      DO 4 J=1,NXF
      ID=NTF-1
      ID=ID+1
      DO 4 I=NTF1,NT
      ID=ID-1
      N=2*((J-1)*NT+(I-1))+1
      L=2*((J-1)*NT+(ID-1))+1
      AS(N)=AS(L)
    4 AS(N+1)=0.0
C
C     ROTATE ABOUT F AXIS
      DO 5 I=1,NT
      JD=NXF-1
      JD=JD+1
      DO 5 J=NXF1,NX
      JD=JD-1
      N=2*((J-1)*NT+(I-1))+1
      L=2*((JD-1)*NT+(I-1))+1
      AS(N)=AS(L)
    5 AS(N+1)=0.0
C
C     INV XFM
      M(1)=5
      M(2)=2
      M(3)=0
      IFSET=-1
      CALL HARMIAS,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP62                                          STOP62
C
C     EXPAND 9X,2F
      COMMON/AS/NT4,NTF,NT4F,NTF1,NZ1,NX4,NXF,NX4F,NXF1,NZX
      CALL EXPAND (AS,NT,NX,A)
C
C
C     FWD XFM
      M(1)=0
      M(2)=5
      IFSET =+1
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      IF(IFERR.NE.0) STOP63                                          STOP63
C     FORM PW ARRAY = ABS(H) WHEN OUTPUT IN IO=ALOGIO MANNER
      DO 15 J=1,NX4F
      DO 15 I=1,NT4F
      N=2*((J-1)*NT4+(I-1))+1
      PW(I,J)=A(N)*A(N)+A(N+1)*A(N+1)
   15 PW(I,J)=PW(I,J)/(NT*DT*NX*DX)
C
C     NORMALIZE IN NEG DB
      PWM=PW(2*IM-1,3*JM-7)
```

```
      DO 17 J=1,NX4F
      DO 17 I=1,NT4F
      PW(I,J)=PW(I,J)/PWM
      IF(PW(I,J).LT.1.E-9) PW(I,J)=1.E-9
17     PW(I,J)=-10.*ALOG10(PW(I,J))
C
C     PLOT
      DO 25 J=1,NX4F
      DO 25 I=1,NT4F
25    IPW(I,J)=PW(I,J)+.501
      PRINT 80
   80 FORMAT('1')
      PRINT 77
   77 FORMAT(20X,'RESPONSE IN NEG DB',55X,'FREQ',10X,'VELOC(K=.25)',/)
      D=25.*DF/2.
      I=.T4F+1
      DO 27 KC=1,NT4F
      I=I-1
      D=D-(DF/2.)
      V=D/9.25
27    PRINT 26,(IPW(I,J),J=1,NX4F),D,V)
26    FORMAT(1?(2X,I3),10X,F9.5,5X,F5.1,/ )
      PRINT 79
   79 FORMAT(2X,'K=0',30X,'K=.125',35X,'K=.25')
      PRINT 78
   78 FORMAT(/,35X,'AMPLITUDE RESPONSEG VELOCITY FILTER',/35X,'K WAVE NU
     MBER VS FREQUENCY')
      IF(IPLT.LT.2)  RETURN
      PRINT 10
      N=0
      DO 90 I=1,NT4F
      DO 86 J=1,NX4F
      N=N+1
89    I=(N)=IPW(I,J)
      CALL xMAY(IGH,NX4F,4,NT4F,1)
      RETURN
      END
      SUBROUTINE GRAY(IDT,NH,NSH,NV,NSV)
C     SUBROUTINE GRAY DISPLAYS AN ARRAY OF NV X NH ELEMENTS
C     NSH=NUMBER OF SPACES BETWEEN THE NH HORIZ ELEMENTS OF IDT.  DITTO V VICS &
C     ORDER IN IS FORTRAN TYPE IE (1,1),(2,1)....
      INTEGER*2 JJ(132)
      DIMENSION IG(11),A(12,17),T(132)
      DIMENSION PIC(4,132),GR(4,8)
      COMMON/JJ/JJ,IX,A,T,PIC
      DATA GR/' ','*',',','O',' ','+','X','J',' ','=',2H*X,2H  ,2H*X
     1,2H  ,2H*I,3H  ,'+','=',3H  ,'-',4H  /
      IDTOT=NV*NH+1
      NSV1=NSV+1
      NSV2=NSV+2
      NSH1=NSH+1
      DX=1./(NSH+1)
      DY=1./(NSV+1)
      MSHORT=(NH-1)*NSH+NH
      DO 1 I=1,MSHORT
    1 JJ(I)=0
C
      DO 15 IOUT=1,NV
      IF (IOUT.EQ.NV)  NSV1=1
C     SET UP COLUMNS FOR 2 ROWS OF IDT.
      DO 2 IX=1,NSV2,NSV1
      IC=NH+1
```

```
      DO 7 ICT=1,NH
      IDTCT=IDTCT-1
      IC=IC-1
    2 A(IR,IC)=IDT(IDTCT)
      IDTCT=IDTCT+NH
C
      DO 3 IC=1,NH
      D=(A(NSV2,IC)-A(1,IC))*DY
      DO 3 IR=2,NSV1
    3 A(IR,IC)=A(IR-1,IC)+D
C
      DO 15 IR=1,NSV1
      IT=0
      DO 4 IE=2,NH
      IT=IT+1
      I0=IE-1
      D=(A(IR,IE)-A(IR,I0))*DX
      DO 4  I=1,NSH
      IT=IT+1
    4 T(IT)=A(IR,I0)+D*I
C
      IC=0
      DO 5 IT=1,MSHORT,NSH1
      IC=IC+1
    5 T(IT)=A(IR,IC)
C
      DO 6 I=1,MSHORT
      JJ(I)=T(I)+0.5
      ITEST=JJ(I)
      IF(ITEST.LE.00)  JJ(I)=8
      IF(ITEST.GT.00)  JJ(I)=8
      IF(ITEST.GE. 4)  JJ(I)=7
      IF(ITEST.GE.10)  JJ(I)=6
      IF(ITEST.GE.16)  JJ(I)=5
      IF(ITEST.GE.22)  JJ(I)=4
      IF(ITEST.GE.28)  JJ(I)=3
      IF(ITEST.GE.34)  JJ(I)=2
      IF(ITEST.GE.40)  JJ(I)=1
    6 CONTINUE
C
C
      DO 10 J=1,MSHORT
      KK=JJ(J)
      DO 10  K=1,4
   10 PIC(K,J)=GR(K,KK)
      DO 11 K=1,4
   11 WRITE(6,202) (PIC(K,II), II=1,MSHORT)
      WRITE(6,201)
   15 CONTINUE
C
  201 FORMAT(1X)
  202 FORMAT(1H+,132A1)
      RETURN
      END
      SUBROUTINE FA (BEAR,DANGLE,IBP,BEAM       ,IFPLT)
C     RE SUBROUTINE FILTER
      REAL AREF(4,4)/                                                  1/5
      0.0,0.0,0.0,180.0,                                               2/5
      90.0,45.0,26.6,135.0,                                            3/5
      90.0,63.4,45.0,116.6,                                            4/5
      -90.0,-45.0,-26.6,-135.0/                                        5/5
      DIMENSION IPW(16,16) ,PW(16,16)
```

```
      DIMENSION A2(4,4), BEAM(4,4),AI5I2),INV(64 ), SI64 ),M(3),AS(32)
      DIMENSION IPG(250)
      EQUIVALENCE (IPW,IPG)
      DIMENSION AU(512),SU(64),INVU(64)
      EQUIVALENCE(A,AU),(S,SU),(INV,INVU)
      COMMON/U/AU,SU,INVU
      NX=4
      DX=2.
      NY=4
      DY=2.0
      DAMIN=.0
      IF(DANGLE.GT.180.0.OR.DANGLE.LT.DAMIN) STOP5D                    STOP5D
C
C     SET -180.LT.BEAR.LT.+180
      IF(ABS(BEAR).GT.360.0) STOP51                                    STOP51
      IF(BEAR.LT.-180.0) BEAR=BEAR+360.0
      IF(BEAR.GT.+180.0) BEAR=BEAR-360.0
C
C     RESCALE AREF
      D2=0.5*DANGLE
      DO 1 K=1,NY
      DO 1 J=1,NX
      A2(J,K)=AREF(J,K)-BEAR
      IF(ABS(A2(J,K)).LE.180.) GO TO 1
      IF(A2(J,K).LT.-180.0) GO TO 2
      A2(J,K)=A2(J,K)-360.0
      GO TO 1
2     A2(J,K)=A2(J,K)+360.0
1     CONTINUE
C
C     SET SPATIAL FREQ RESPONSE IN 'HARM' ORDER
      DL=-D2
      DH=+D2
      Q=1.5
      DH=DH+Q
      DL=DL-Q
      DO 3 K=1,NY
      DO 3 J=1,NX
      IF(A2(J,K).GT.DH. OR.A2(J,K).LT.DL) GO TO 4
      BEAM(J,K)=1.0
      GO TO 3
4     BEAM(J,K)=0.0
3     CONTINUE
      BEAM(1,1)=1.0
C
C     INVERT STOP AND PASS
      IF(IBP.EQ.1) GO TO 5
      DO 5 K=1,NY
      DO 5 J=1,NX
      IF(BEAM(J,K).EQ.0.0) GO TO 55
      BEAM(J,K)=0.0
      GO TO 5
55    BEAM(J,K)=1.0
5     CONTINUE
      IF(IPPLT.EQ.0) RETURN
C
C     NOTE MAX IN SPECIFIED PASSBAND
      PWR=0.0
      DO 56 K=1,NY
      DO 56 J=1,NX
      IF(PWR.LT.BEAM(J,K)) GO TO 57
      GO TO 56
```

```
 57     PWM=BEAM(J,K)
        JMAX=J
        KMAX=K
 56     CONTINUE
C
C       INV XFM
        N=-1
        DO 6 K=1,NY
        DO 6 J=1,NX
        N=N+2
        AS(N)=BEAM(J,K)
 6      AS(N+1)=0.0
        M(1)=2
        M(2)=2
        M(3)=0
        IFSET=-1
        CALL HARM(AS,M,INV,S,IFSET,IFERR)
        IF(IFERR.NE.0) STOP52                                    STOP52
C
C       EXPAND 4X
        COMMON/WS/NX4,NXF,NX4F ,NXF1,NZX,NY4,NYF,NY4F,NYF1,NZY
        NX4=NX*4
        NXF=(NX/2)+1
        NX4F=(NX4/2)+1
        NXF1=NXF+1
        NZX=3*NX
        NY4=NY*4
        NYF=(NY/2)+1
        NY4F=(NY4/2)+1
        NYF1=NYF+1
        NZY=3*NY
        CALL EXPAND(AS,NX,NY,A)
C
C       END XFM
        IFSET=+1
        M(1)=4
        M(2)=4
        CALL HARM(A,M,INV,S,IFSET,IFERR)
        IF(IFERR.NE.0) STOP53                                    STOP53
C
C       FORM PW ARRAY = ABS(H) WHEN OUTPUT IN 10*ALOG10 MANNER
        N=-1
        DO 15 K=1,NY4
        DO 15 J=1,NX4
        N=N+2
        PW(J,K)=A(N)*A(N)+A(N+1)*A(N+1)
 15     PW(J,K)=PW(J,K)/(NX*DX*NY*DY)
C
C       NORMALIZE IN NEG DB W.R.T. MAX SPEC
        PWM=PW(4*JMAX-3,4*KMAX-3)
        DO 17 K=1,NY4
        DO 17 J=1,NY4
        PW(J,K)=PW(J,K)/PWM
        IF(PW(J,K).LT.1.E-9) PW(J,K)=1.E-9
 17     PW(J,K)=-10.*ALOG10(PW(J,K))
C
C       PUT NEG WAVEND FIRST
        CALL REV(PW,NX4,NY4)
C
C       PLOT
        DO 25 K=1,NX4
        DO 25 J=1,NY4
```

```
   25    IPW(J,K)=PW(J,K)+0.501
         PRINT 77
   77    FORMAT('1')
         PRINT 78
      78 FORMAT(20X,'RESPONSE IN NEG DB',55X,'K',/)
         D=0.25+0.03125
         K=17
         DO 27 KC=1,16
         K=K-1
         D=D-0.03125
   27    PRINT 26 ,(IPW(J,K),J=1,16),D
   26    FORMAT(16(2X,I3),10X,F8.5//)
         PRINT 777
  777 FORMAT(/,35X,'AMPLITUDE RESPONSED ANGLE FILTER',/35X,'Y WAVE NUMBE
        R VS X WAVE NUMBER')
         IF(IPPLT.EQ.1) RETURN
         PRINT 77
         CALL GRAY(IPG,16,4,16,2)
         RETURN
         END
         SUBROUTINE DOALL (AS,A,NX,NX4)
         DIMENSION A(1),AS(1)
         COMMON/DO/JS,KS,J,K
         NS=2*((KS-1)*NX+(JS-1))+1
         N=2*((K-1)*NX4+(J-1))+1
  100    CONTINUE
         A(N)=AS(NS)
         A(N+1)=AS(NS+1)
         RETURN
         END
         SUBROUTINE EXPAND(AS,NX,NY,A)
         COMMON/AS/NX4,NXF, NX4F ,NXF1,NZX,NY4,NYF,NY4F,NYF1,NZY
         DIMENSION A(1),AS(1)
C  EXPAND 'AS' BY 4X BY APPROPRIATELY ADDING ZEROS TO FORM 'A'
C     X REFERS TO THE 1ST DIMENSION  Y TO THE 2ND
C     INITILIZE A ARRAY
         N=0
         DO 8 K=1,NY4
         DO 8 J=1,NX4
         DO 9 L=1,2
         N=N+1
    8    A(N)=0.0
C     FILL I. SAVED DATA
         NX=2*(NXF-1)
         COMMON/DO/JS,KS,J,K
         DO 11 KS=1,NYF
         DO 11 JS=1,NXF
         K=KS
         J=JS
   11    CALL DOALL(AS,A,NX,NX4)
         DO 12 KS=1,NYF
         DO 12 JS=NXF1,NX
         K=KS
         J=JS+NZX
   12    CALL DOALL(AS,A,NX,NX4)
         DO 13 KS=NYF1,NY
         DO 13 JS=NXF1,NX
         K=KS+NZY
         J=JS+NZX
   13    CALL DOALL(AS,A,NX,NX4)
         DO 14 KS=NYF1,NY
         DO 14 JS=1,NXF
```

```
      K=KS*NZY
      J=JS
   14 CALL OJALL(AS,A,NX,NX4)
      RETURN
      END
      SUBROUTINE REV (PW,NX,NY)
C     PUT NEG WAVENO FIRST
      DIMENSION PW(NX,NY)
      NXF=(NX/2)+1
      NYF=(NY/2)+1
C     ROTATE W.R.T. X
      JS=NY-NYF
      DO 18 J=1,NX
      DO 18 JC=1,JS
      T=PW(J,NY)
      KY=NY+1
      DO 19 IC=2,NY
      KY=KY-1
19    PW(J,KY)=PW(J,KY-1)
18    PW(J,1)=T
C     ROTATE W.R.T Y
      JS=NX-NXF
      DO 23 K=1,NY
      DO 23 KC=1,JS
      T=PW(NX,K)
      KX=NX+1
      DO 24 IC=2,NX
      KX=KX-1
24    PW(KX,K)=PW(KX-1,K)
23    PW(1,K)=T
      RETURN
      END
      SUBROUTINE REVREV (PW,NX,NY)
C     PUT POS WAVENO FIRST.  INVERSE OF REV
      DIMENSION PW(NX,NY)
      NXF=(NX/2)+1
      NYF=(NY/2)+1
C     ROTATE W.R.T. X
      JS=NY-NYF
      DO 18 J=1,NX
      DO 18 JC=1,JS
      T=PW(J,1)
      DO 19 IC=2,NY
19    PW(J,IC-1)=PW(J,IC)
18    PW(J,NY)=T
C     ROTATE W.R.T Y
      JS=NX-NXF
      DO 23 K=1,NY
      DO 23 KC=1,JS
      T=PW(1,K)
      DO 24 IC=2,NX
24    PW(IC-1,K)=PW(IC,K)
23    PW(NX,K)=T
      RETURN
      END
      SUBROUTINE ML (S,VELOC,ANGLE,XN,SDS)
      COMMON/U/U
      COMPLEX*8 CT,SC(64,16),A(64,16),SX(64)
      DIMENSION S(32,4,4),AH(128),SH(16),INVH(16),M(16)
      DIMENSION XN(32,4,4),L(16)
      DIMENSION SDS(1)
      COMPLEX*16 DM,D,FNU,Q(33,16,16),AC(256),XPW(33,16,16)
```

```
      EQUIVALENCE(Q(1,1,1),XPH(1,1,1)),(SC(1,1),A(1,1))
      NAMELIST/INVED/IFREQ,FREQ,D
      REAL*8 ANOR,BNOR
C
C     ALIGN NOISE AS PER SIG VELOCITY VECTOR
      CALL DELAY (-VELOC,ANGLE,XN)
      CALL PLOT16(XN,32)
      KC=0
      DO 1 K=1,4
      DO 1 J=1,4
      KC=KC+1
      DO 1 I=1,32
    1 SC(I,KC)=CMPLX(XN(I,J,K),0.0)
C
C  TAPER AND ADD ZEROS TO INPUT...
      DO 33 KC=1,16
      DO 31 I=1,32
   31 AH(I)=REAL(SC(I,KC))
      CALL TAPER (AH,32)
      DO 32 I=1,32
   32 SC(I,KC)=CMPLX(AH(I),0.0)
   33 CONTINUE
      DO 34 KC=1,16
      DO 34 I=33,64
   34 SC(I,KC)= (0.0,0.0)
C
C  TRANSFORM INPUT NOISE
      CALL NEGATE (XN)
      M(1)=6
      M(2)=0
      M(3)=0
      CALL HARM(AH,M,INVH,SH,0,IFERR)
      IFSET=+2
      PWB=0.0
      DO 3 KC=1,16
      N=-1
      DO 2 I=1,64
      N=N+2
      AH(N)=REAL(SC(I,KC))
2     AH(N+1)=0.0
      CALL HARM(AH,M,INVH,SH,IFSET,IFERR)
      IF(IFERR.NE.0) STOP40                                    STOP40
      IF(KC.EQ.1) CALL APLOT(AH,128)
      N=-1
      DO 3 I=1,64
      N=N+2
      CT=CMPLX(AH(N),AH(N+1))
      SC(I,KC)=CT
      T=CABS(CT)
      IF (T.GT.PWB) PWB=T
3     CONTINUE
C
C     WRITE (6,67) PWB
C  67 FORMAT(20X,/,1X,'PWB= ',E14.7)
671   CONTINUE
C
C     LIMIT DYNAMIC POWER RANGE
      DO 40 KC=1,16
      DO 40 I=1,64
      SC(I,KC)=SC(I,KC)/PWB
      T=CABS(SC(I,KC))
      IF (T.LT.0.001) SC(I,KC)=(SC(I,KC)/T)*0.001
```

```
   40 CONTINUE
C
C  SMOOTH SC
      DO 43 KC=1,16
      N=-1
      DO 41 I=1,64
      N=N+2
      AH(N)=REAL(SC(I,KC))
   41 AH(N+1)=AIMAG(SC(I,KC))
      CALL AHANN(AH,64)
      IF (KC.LT.3) CALL APLOT(AH,128)
      N=-1
      DO 42 I=1,64
      N=N+2
   42 SC(I,KC)=CMPLX(AH(N),AH(N+1))
   43 CONTINUE
C
C  GET CROSS POWER
      DO 6 J=1,16
      DO 6 K=J,16
      DO 4 I=1,33
4     XPW(I,J,K)=SC(I,J)*CONJG(SC(I,K))
      IF(J.EQ.K)  GO TO 6
      DO 5 I=1,33
    5 XPW(I,K,J)=CONJG(XPW(I,J,K))
    6 CONTINUE
C
C     NORMALIZE FOR GAIN EQUILIZATION
      DO 30 I=1,33
      DO 30 K=1,16
      ANOR=CABS(XPW(I,K,K))
      DO 30 J=1,16
      BNOR=CABS(XPW(I,J,J))
30    XPW(I,J,K)=XPW(I,J,K)/DSQRT(ANOR*BNOR)
C
C  INVERT XPW
      DO 8 I=1,33
      N=0
      DO 7 K=I,16
      DO 7 J=1,16
      N=N+1
    7 AC(N)=XPW(I,J,K)
75    CALL COMINV(AC,16,D,L,M)
      IF(D.EQ.0) I=I-1
      FR=0=(I-1)*1./1.28
C     WRITE(6,1)FREQ
      IF(CABS(D).EQ.0.0) STOP54
      N=0
      DO 8 K=1,16
      DO 8 J=1,16
      N=N+1
    8 Q(I,J,K)=AC(N)
C
C  DO FINAL FILTER
      DO 10 I=2,33
      DN=(0.000,0.000)
      DO 11 K=1,16
      DO 11 J=1,16
   11 DN=DN+Q(I,J,K)
      DO 10 K=1,16
      END=(0.000,0.000)
      DO 13 J=1,16
```

```
13        FNU=FNU+Q(I,J,K)
          A(I,K)=FNU/DM
10      CONTINUE
C   BLOCK INPUT DC
          DO 12 KC=1,16
       12 A(1,KC)=CMPLX(0.0,0.0)
C
C   NOTE THAT XN IS DESTROYED IN PLOTTING...
C         KC=0
C         DO 50 K=1,4
C         DO 50 J=1,4
C         KC=KC+1
C         DO 50 I=1,32
C      50 XN(I,J,K)=CABS(A(I,KC))
C         CALL PLOT16(XN,32)
C
C
C         DO NEG FREQ
          N=33
          DO 9 I=34,64
          N=N-1
          DO 9 K=1,16
        9 A(I,K)=CONJG(A(N,K))
C         ISTEP=16
C         DO 1075 KC=1,16,ISTEP
C         DO 107 I=1,64
C         X=REAL(A(I,KC))
C         Y=AIMAG(A(I,KC))
C  107 AH(I)=(X*X+Y*Y)
C 1075 CALL UPLOT(AH,64)
C
C         ALIGN RCVD SIGNAL
          CALL DELAY (-VELOC,ANGLE,S)
C
C   TAPER RCVD SIGNAL..
          DO 36 K=1,4
          DO 36 J=1,4
          DO 35 I=1,32
       35 AH(I)=S(I,J,K)
          CALL TAPER (AH,32)
          DO 36 I=1,32
       36 S(I,J,K)=AH(I)
C
C         XFM RCVD SIGNAL
          CALL NEGATE (S)
          M(1)=6
          M(2)=0
          M(3)=0
          KC=0
          DO 1000 K=1,4
          DO 1000 J=1,4
          IFSET=+1
          N=-1
          KC=KC+1
          DO 200 I=1,32
          N=N+2
          AH(N)=S(I,J,K)
  200     AH(N+1)=0.0
C   ADD ZEROS
          DO 37 I=65,128
       37 AH(I)=0.0
          CALL HARM(AH,M,INV,SH,IFSET,IFERR)
```

```
      N=-1
      DO 300 I=1,64
      N=N+2
300   SX(I)=CMPLX(AH(N),AH(N+1))
C
C  SMOOTH FLT AND MLTPLY IN FREQ DOMAIN
      N=-1
      DO 400 I=1,64
      N=N+2
      AH(N)=REAL(A(I,KC))
400   AH(N+1)=AIMAG(A(I,KC))
      CALL AHAVN(AH,64)
      N=-1
      DO 600 I=1,64
      N=N+2
600   SX(I)=SX(I)*CMPLX(AH(N),AH(N+1))
C
C     INV XFM
      IFSET=-1
      N=-1
      DO 500 I=1,64
      N=N+2
      AH(N)=REAL(SX(I))
500     AH(N+1)=AIMAG(SX(I))
      CALL HARM(AH,M,INV,SH,IFSET,IFERR)
      N=-1
      DO 100 I=1,32
      N=N+2
100     S(I,J,K)=AH(N)
1000    CONTINUE
      CALL NEGATE (S)
C
C     PLOT A AD SJM
C     CALL PLOT16(S,32)
      CALL DS (10000.,0.,S,SDS)
      RETURN
C     DEBUG SUBCHK,SUBTRACE
66    FORMAT(' ',16(1X,D7.1))
      END
      SUBROUTINE SPBPF (DATA)
C  SPECIAL PURPOSE BPF TO FILTER 3-D VER DATA, BLOCK DC AND LPF AT 25 HZ.
      DIMENSION DATA(128),M(3),INV(32),S(32)
      COMMON/D/A(256)
      NPTS=128
      SR=100.0
      FF=SR/2.
      N2=NPTS/2
      NFF=N2+1
C     FF= FOLDING FREQ
      M(1)=7
      M(2)=0
      M(3)=0
      IFSET=1
      DO 1 I=1,NPTS
      IRE=2*I-1
      IIM=2*I
      A(IRE)=DATA(I)
1     A(IIM)=0.0
15    CALL HARM(A,M,INV,S,IFSET,IFERR)
C
C     HIGH FREQ
      TI=0.439
```

```
      T2=0.049
      NHF=30
      IRE=2*NHF-1
      IIM=2*NHF
      A(IRE)=T1*A(IRE)
      A(IIM)=T1*A(IIM)
      IRE=2*(NHF+1)-1
      IIM=2*(NHF+1)
      A(IRE)=T2*A(IRE)
      A(IIM)=T2*A(IIM)
C
C     ZERO STOP BAND
      NHZ=NHF+2
      DO 2 I=NHZ,NFF
      IRE=2*I-1
      IIM=2*I
      A(IRE)=0.0
2     A(IIM)=0.0
C
C     LOW FREQ
      NLF=3
      T1=.401
      T2=.046
C     HAND OTHER TRANSITION.
      IRE=2*NLF-1
      IIM=2*NLF
      A(IRE)=T1*A(IRE)
      A(IIM)=T1*A(IIM)
      IRE=2*(NLF-1)-1
      IIM=2*(NLF-1)
      A(IRE)=T2*A(IRE)
      A(IIM)=T2*A(IIM)
C
C     ZERO STOP BAND
      NLZ=NLF-2
      DO 3 I=1,NLZ
      IRE=2*I-1
      IIM=2*I
      A(IRE)=0.0
3     A(IIM)=0.0
C
C     ABOUT FF RE'S ARE EVEN,IM'S ARE ODD.
20    NFF1=NFF+1
      J=NFF
      DO 4 I=NFF1,NPTS
      J=J-1
      JRE=2*J-1
      JIM=2*J
      IRE=2*I-1
      IIM=2*I
      A(IRE)=A(JRE)
4     A(IIM)=-A(JIM)
C
C     INVERSE FFT
      IFSET=-2
      CALL HARM(A,M,INV,S,IFSET,IFERR)
      DO 6 I=1,NPTS
      IRE=2*I-1
6     DATA(I)=A(IRE)
      RETURN
      END
C     NL2
```

```
C       GEN 2 WAVES AND TEST ML ESTIMATION
        DIMENSION S(32,4,4),AA(64),SS(32),XN(32,4,4),SDS(32)
        REAL TT(32,4,4)
        COMMON/DIM/NT,DT,DX,DY
        NT=32
        COMMON/U/TT,AA
        DT=0.02
        DX=2.
        DY=2.
        THE=0.*3.14159/180.0
        AMP=-10.0
        DO 300 II=2,5
        IF(II.EQ.2) AMP=-5.
        IF(II.EQ.3) AMP=-10.
        IF(II.EQ.4) AMP=-50.
        IF(II.EQ.5) AMP=-500.
        HZ=50.
        FREQ=5.0
        PF2=8.)*ATAN(1.)*FREQ
        DO 1 I=1,32
        T=(I-1)*(1./HZ)
1       SS(I)=AMP*SIN(PF2*T+THE)
        VELOC=25.
        CALL RMEAN(SS,32,0.01)
        ANGLE=30.
        SNR=10000.
4       CONTINUE
C       GEN NOISE
        CALL PWAVE(SS,32,0.02,VELOC,ANGLE,SNR,301,XN)
        CALL PLOT16(XN,32)
C
        SNR=-SNR
        CALL PWAVE (SS,32,0.02,VELOC,ANGLE,SNR,301,TT)
        CALL PLOT16(TT,32)
        DO 6 I=1,32
        DO 6 J=1,4
        DO 6 K=1,4
6       XN(I,J,K)=XN(I,J,K)+TT(I,J,K)
C
C
C       GEN SIG
100     CONTINUE
        FREQ=5.
        PF2=8.*ATAN(1.)*FREQ
        AMPL=10.
        DO 2 I=1,32
        T=(I-1.)*(1./HZ)
2       SS(I)=AMPL*SIN(PF2*T)
        DO 21 I=22,32
21      SS(I)=0.0
101     CONTINUE
        SANGLE=30.0
        CALL PWAVE (SS,32,0.02,VELOC,SANGLE,1000.,903,S)
        CALL PLOT16(S,32)
        CALL DS (25.,SANGLE,S,SDS)
  200 CONTINUE
        DO 5 K=1,4
        DO 5 J=1,4
        DO 5 I=1,32
5       S(I,J,K)=S(I,J,K)+XN(I,J,K)
        CALL PLOT16(S,32)
        CALL DS (25.,SANGLE,S,SDS)
```

```
      CALL PLOT11(S)
      CALL ML (S,VELOC,SANGLE,XN,SDS)
300   CONTINUE
      STOP
      DEBUG SUBCHK
      END
C     ML7
C     GEN 7 WAVES AND DOES FILTERING AND ML ESTIMATION
      DIMENSION SDS(32)
      REAL SANDN(32,4,4)
      DIMENSION A(3),A(1024)
      DIMENSION PW(17,8,9)
      DIMENSION S(32),          SS(32,4,4),SS2(32,4,4)
      REAL FANG(4,4)/16*1.0/
      REAL EVEL(17,4,4)/272*1.0/
      REAL FREQ(17)/17*1.0/
      COMMON/DIM/NT,DT,DX,DY
      NT=32
      NX=4
      NY=4
      DT=0.02
      DX=2.
      DY=2.
      HZ=50.
      ISEED=101
C
C  1
      FREQ=12.5
      PF2=2.*4.*ATAN(1.)*FREQ
      DO 1 I=1,32
      T=(I-1)*(1./HZ)
1     S(I)=COS(PF2*T)
      VELOC=75.
      ANGLE=-45
      SNR=1000.
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,SNR,ISEED,SS)
99    CONTINUE
      DO 2 I=1,NT
      DO 2 J=1,NX
      DO 2 K=1,NY
2     SS2(I,J,K)=SS(I,J,K)
      CALL DS (VELOC,ANGLE,SS2,SDS)
C
C  2
100   CONTINUE
C
      ANGLE=+45.
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,SNR,ISEED,SANDN)
      CALL DS (VELOC,ANGLE,SANDN,SDS)
      GO TO 56
      DO 3 I=1,NT
      DO 3  J=1,NX
      DO 3 K=1,NX
3     SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
56    CONTINUE
C
C  3
      FREQ=7.8
      VELOC=125.
      ANGLE=+30.
      PF2=4.*ATAN(1.)*FREQ
      DO 4 I=1,32
```

```
      T=(I-1)*(1./HZ)
 4    S(I)=COS(PF2*T)
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 5 I=1,32
      DO 5 J=1,NX
      DO 5 K=1,NY
 5    SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  4
      VELOC=42.
      CALL PWAVE (S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 6 I=1,32
      DO 6 J=1,4
      DO 6 K=1,4
 6    SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
 55   CONTINUE
C
C  5
      FREQ=17.2
      VELOC=100.
      ANGLE=0.0
      PF2=8.*ATAN(1.)*FREQ
      DO 7 I=1,32
      T=(I-1)*(1./HZ)
 7    S(I)=COS(PF2*T)
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 8 I=1,32
      DO 8 J=1,4
      DO 8 K=1,4
 8    SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  6
      FREQ=21.9
      PF2=8.*ATAN(1.)*FREQ
      DO 9 I=1,32
      T=(I-1)*(1./HZ)
 9    S(I)=COS(PF2*T)
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 10 I=1,NT
      DO 10 J=1,4
      DO 10 K=1,4
10     SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  7
      FREQ=3.1
      VELOC=30.
      ANGLE=120.
      PF2=8.*ATAN(1.)*FREQ
      DO 11 I=1,32
      T=(I-1)*(1./HZ)
11    S(I)=COS(PF2*T)
      CALL PWAVE (S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 12 I=1,32
      DO 12 J=1,4
      DO 12 K=1,4
12    SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
      DO 13 I=1,32
```

```
      DO 13 J=1,4
      DO 13 K=1,4
13    SANDN(I,J,K)=SANDN(I,J,K)+SS2(I,J,K)
C
C     SANDN = SIGNAL AND NOISE
C     SS2 = NOISE
C
120   CONTINUE
      CALL PLOT11(SANDN)
      CALL DS (75.,45.,SANDN,SDS)
      CALL DS (42.,90.,SANDN,SDS)
C
C
C
      CALL NEGATE(SS2)
C      CALL FWND(SS2,0,0,PW)
C      CALL PLT3 (PW,-9.,45.,1,2)
      CALL FA (+45.,+45.,1,FANG,2)
C     CALL FV (31.25,62.5,1,FVEL,0,2)
      CALL FILTER (SS2,FANG,FVEL,FFQ)
C     CALL FWND(SS2,0,0,PW)
C     CALL PLT4(PW)
      CALL NEGATE (SS2)
C
      CALL NEGATE (SANDN)
      CALL FILTER (SANDN,FANG,FVEL,FFQ)
      CALL NEGATE (SANDN)
      CALL DS (42.,90.,SANDN,SDS)
      CALL DS (75.,45.,SANDN,SDS)
      CALL PL (SANDN,75.,45.,SS2,SDS)
      STOP
      DEBUG SUBCHK
      END
C     VERSION
C     THIS PGM ACCEPTS THE OUTPUT OF PRELIM
      DIMENSION S(32,4,41),PW(17,8,8)
      COMMON/LIMIT/T,DT,DX,DY
      REAL FANG(4,41/16*1.0/,FVEL(17,4,41/272*1.0/,FFQ(17)/17*1.0/
      NT=32
      DT=0.02
      DX=2.
      DY=2.
      ICOUNT=0
99    ICOUNT=ICOUNT+1
C     READY LOW PASS VELOCITY FILTER
      CALL F/(500.,500.,0,FVEL,0,2)
C
C     READ INPUT
      READ(5,102,END=103)(((S(I,J,K),I=1,32),J=1,4),K=1,4)
C
      CALL FILTER (S,FANG,FVEL,FFQ)
      CALL PLOT16(S,32)
      CALL NEGATE (S)
      CALL FWND (S,0,0,PW)
      CALL PLT3(PW,-9.0,24.,1,2)
      GO TO 99
103   STOP
102   FORMAT(11F7.4)
      END
C     RESPONSE
C     FREQ RESP TEST 3-D
      DIMENSION SUS(32)
```

```
      DIMENSION M(3),A(1024)
      DIMENSION PW(17,8,8)
      DIMENSION S(32),          SS(32,4,4)
      REAL SS2(32,4,4)/512*0.0/
      REAL FANG(4,4)/16*1.0/
      REAL FVEL(17,4,4)/272*1.0/
      REAL FF0(17)/17*1.0/
      COMMON/DIM/NT,DT,DX,DY
      COMMON/P5/IM,JM,KM
      IM=2
      JM=2
      KM=2
      NT=32
      NX=4
      NY=4
      DT=0.02
      DX=2.
      DY=2.
      SS2(1,1,1)=1.
      CALL PLOT10(SS2,32)
      CALL NEGATE(SS2)
      IFPLT=2
      IFSMO=0
      IMAMN=0
      BEAR=45.
      BW=45.
      CALL FA (BEAR,BW,1,FANG,IFPLT)
1     CONTINUE
      CALL FILTER(SS2,FANG,FVEL,FF0)
      CALL FMOD(SS2,IFSMO,IMAMN,PW)
      CALL PLT5(PW,0.0,26.,1,1)
      STOP
      DEBUG SUBCHK
      AT 1
      DO 3 J=1,4
3     WRITE(6,2) (    FANG(J,K),K=1,4)
2     FORMAT(2X     ,4F6.2,/)
      STOP44
      END
C     FMOD
C     GEN T WAVES AND DOES FILTERING AND LMSE ESTIMATION
      DIMENSION SDS(32)
      DIMENSION M(3),A(1024)
      DIMENSION PW(17,8,8)
      DIMENSION S(32),          SS(32,4,4),SS2(32,4,4)
      REAL FANG(4,4)/16*1.0/
      REAL FVEL(17,4,4)/272*1.0/
      REAL FF0(17)/17*1.0/
      COMMON/DIM/NT,DT,DX,DY
      NT=32
      NX=4
      NY=4
      DT=0.02
      DX=2.
      DY=2.
      MZ=50.
      ISEED=101
C
C  1
      FREQ=12.5
      PF2=2.*4.*ATAN(1.)*FREQ
      DO 1 I=1,32
```

```
      T=(I-1)*(1./HZ)
1     S(I)=COS(PF2*T)
      VELOC=75.
      ANGLE=-45
      SNR=1000.
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,SNR,ISEED,SS)
99    CONTINUE
      DO 2 I=1,NT
      DO 2 J=1,NX
      DO 2 K=1,NY
2     SS2(I,J,K)=SS(I,J,K)
      CALL DS (VELOC,ANGLE,SS2,SDS)
C
C  2
103   CONTINUE
C
      ANGLE=+45.
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,SNR,ISEED,SS)
      CALL DS (VELOC,ANGLE,SS,SDS)
      DO 3 I=1,NT
      DO 3  J=1,NX
      DO 3 K=1,NY
3     SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  3
      FREQ=7.8
      VELOC=125.
      ANGLE=+90.
      PF2=8.*ATAN(1.)*FREQ
      DO 4 I=1,32
      T=(I-1)*(1./HZ)
4     S(I)=COS(PF2*T)
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 5 I=1,32
      DO 5 J=1,NX
      DO 5 K=1,NY
5     SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  4
      VELOC=42.
      CALL PWAVE (S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 6 I=1,32
      DO 6 J=1,4
      DO 6 K=1,4
6     SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
55    CONTINUE
C
C  5
      FREQ=17.2
      VELOC=100.
      ANGLE=0.0
      PF2=8.*ATAN(1.)*FREQ
      DO 7 I=1,32
      T=(I-1)*(1./HZ)
7     S(I)=COS(PF2*T)
      CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
      CALL DS(VELOC,ANGLE,SS,SDS)
      DO 8 I=1,32
      DO 8 J=1,4
      DO 8 K=1,4
```

```
8       SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  6
        FREQ=21.9
        PF2=8.*ATAN(1.)*FREQ
        DO 9 I=1,32
        T=(I-1)*(1./HZ)
9       S(I)=COS(PF2*T)
        CALL PWAVE(S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
        CALL DS(VELOC,ANGLE,SS,SDS)
        DO 10 I=1,NT
        DO 10 J=1,4
        DO 10 K=1,4
10       SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
C
C  7
        FREQ=3.1
        VELOC=10.
        ANGLE=120.
        PF2=8.*ATAN(1.)*FREQ
        DO 11 I=1,32
        T=(I-1)*(1./HZ)
11      S(I)=COS(PF2*T)
        CALL PWAVE (S,NT,DT,VELOC,ANGLE,1000.,ISEED,SS)
        CALL DS(VELOC,ANGLE,SS,SDS)
        DO 12 I=1,32
        DO 12 J=1,4
        DO 12 K=1,4
12      SS2(I,J,K)=SS2(I,J,K)+SS(I,J,K)
        CALL LOOK (SS2)
C
C
C
        CALL NEGATE(SS2)
         CALL FANDI(SS2,0,0,PW)
         CALL PLT3 (PW,-90.,45.,1,2)
        CALL FS (+45.,+45.,1,FANG,2)
        CALL FV (31.25,62.5,1,FVEL,0,2)
1234    CONTINUE
C       CALL FF(0.,6.3,1,FFQ,2,2)
        CALL FILTER (SS2,FANG,FVEL,FFQ)
        CALL FANDI(SS2,0,0,PW)
        CALL PLT4(PW)
        CALL NEGATE (SS2)
        CALL LOOK(SS2)
        STOP
        DEBUG SUBCHK
        AT 1234
        STOP1234
        END
        SUBROUTINE LOOK (SS2)
        REAL SS2(32,4,4),SDS(32)
        CALL DS (75.,-45.,SS2,SDS)
        CALL DS (75.,45.,SS2,SDS)
        CALL DS (125.,90.,SS2,SDS)
        CALL DS (42.,90.,SS2,SDS)
        CALL DS (100.0,0.0,SS2,SDS)
        CALL DS (100.,0.,SS2,SDS)
        CALL DS (30.,120.,SS2,SDS)
        CALL PLOT11 (SS2)
        CALL PLOT16 (SS2,32)
        RETURN
```

## REFERENCES


[1] E. J. Kelly and M. J. Levin, "Signal Parameter Estimation for Seismic Arrays," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Report 339, January, 1964.

[2] J. L. Allen, "Array Antennas: New Applications for an Old Technique," IEEE Spectrum, November, 1964.

[3] J. Capon, R. J. Greenfield, and R. J. Kolker, "Multidimensional Maximum-likelihood Processing of a Large Seismic Array," Proceedings of the IEEE, Vol. 55, No. 2, February, 1967.

[4] J. Capon, "Applications of Detection and Estimation Theory to Large Array Seismology," Proceedings of the IEEE, Vol. 58, No. 5, May, 1970.

[5] F. C. Schweppe, "Sensor-array Data Processing for Multiple-Signal Sources," IEEE Transactions on Information Theory, Vol. IT-14, No. 2, March, 1968.

[6] H. Kobayashi and P. D. Welch, "The Detection and Estimation of Two Simultaneous Seismic Events," Symposium on Computer Processing in Communications, Polytechnic Institute of Brooklyn, April, 1969.

[7] J. B. Thomas, Statistical Communication Theory, John Wiley and Sons, Inc., New York, 1969.

[8] J. S. Bendat and A. G. Piersol, Measurement and Analysis of Random Data, Wiley and Sons, New York, 1966.

[9] R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra, Dover, New York, 1958.

[10] P. I. Richards, "Computing Reliable Power Spectra," IEEE Spectrum, January, 1967.

[11] P. Welch, "The Use of the Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, June, 1967.

[12] C. Bingham, M. D. Godfrey, and J. W. Tukey, "Modern Techniques of Power Spectrum Estimation," IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, June, 1967.

[13] B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, New York, 1969.

[14] H. D. Helms, "Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response," IEEE Transactions on Audio and Electroacoustics, Vol. AU-16, September, 1968.

[15] J. Kraus, Antennas, McGraw-Hill, New York, 1950.

[16] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, Vol. 19, No. 90, April, 1965.

[17] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The Fast Fourier Transform Algorithm and Its Applications," IBM Watson Research Center, Yorktown Heights, N.Y., Research Paper RC-1743, February, 1967.

[18] G. D. Bergland, "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum, July, 1969.

[19] O. S. Halpeny, "Epoch Detection by Digital Cepstrum Analysis," Master's Thesis, University of Florida, 1970.

[20] J. Capon, "High-Resolution Frequency-Wavenumber Spectrum Analysis," Proceedings of the IEEE, Vol. 57, No.8, August, 1969.

[21] A. Papoulis, Systems and Transforms With Applications in Optics, McGraw-Hill, New York, 1968.

[22] J. W. Goodman, Introduction to Fourier Optics, McGraw-Hill, New York, 1968.

[23] R. T. Lacoss, E. J. Kelly, and M. N. Toksoz, "Estimation of Seismic Noise Structure Using Arrays," Geophysics, Vol. 34, No. 1, February, 1969.

[24] E. Smart and E. A. Flinn, "Fast Frequency-Wavenumber Analysis and Fisher Signal Detection in Real-Time Infrasonic Array Data Processing," Geophys. J.R. Astr. Soc., Vol. 26, 1971.

[25] System/360 Scientific Subroutine Package, Version III, Fifth Edition, IBM Corporation, White Plains, N.Y., 1970.

[26] D. G. Childers, W. Mesa, O. S. Halpeny, and N. W. Perry, Jr., "Simulation of Spatio-Temporal Visual Evoked EEGs," Association for Computing Machinery, Annual Conference, Boston, Mass., August, 1972.

[27] D. G. Childers, N. W. Perry, Jr., O. S. Halpeny, and J. R. Bourne, "Spatio-temporal Measures of Cortical Functioning in Normal and Abnormal Vision," Computers and Biomedical Research, Vol. 5, 1972.

[28] L. R. Rabiner, B. Gold, and C. A. McGonegal, "An Approach to the Approximation Problem for Nonrecursive Digital Filters," IEEE Transactions on Audio and Electroacoustics, Vol. AU-18, June, 1970.

[29] E. A. Robinson, Statistical Communication and Detection with Special Reference to Digital Data Processing of Radar and Seismic Signals, Hafner Publishing Company, New York, 1967.

[30] M. Backus, J. Burg, D. Baldwin, and E. Bryan, "Wide-Band Extraction of Mantle P waves from Ambient Noise," Geophysics, Vol. 29, No. 5, October, 1964.

[31] J. P. Burg, "Three-Dimensional Filtering with an Array of Seismometers," Geophysics, Vol. 29, No. 5, October, 1964.

[32] C. W. Helstrom, Statistical Theory of Signal Detection, Pergamon Press, New York, 1968.

[33] H. L. Van Trees, Detection, Estimation, and Modulation Theory, Part 1, Wiley and Sons, New York, 1968.

[34] D. G. Childers and I. S. Reed, "Optimum Signal Processing in the Presence of Noise," Interim Technical Documentary Report No. SSD-TDR-64-127, University of Southern California, Los Angeles, Calif., March, 1964.

[35] F. Bryn, "Optimum Signal Processing of Three-Dimensional Arrays Operating on Gaussian Signals and Noise," Journal of the Acoustical Society of America, Vol. 34, No. 3, March, 1962.

[36] W. Vanderkulk, "Optimum Processing for Acoustic Arrays," Journal Brit. I.R.E., October, 1963.

[37] E. J. Kelly, "A Comparison of Seismic Array Processing Schemes," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Note 1965-21, June, 1965.

[38] E. J. Kelly, I. S. Reed, and W. L. Root, "The Detection of Radar Echoes in Noise," Journal Soc. Ind. Appl. Math., Vol. 8, June, 1960.

[39] S. W. Director, "Survey of Circuit-Oriented Optimization Techniques," IEEE Transactions on Circuit Theory, Vol. CT-18, No. 1, January, 1971.

[40] R. T. Lacoss, "Adaptive Combining of Wideband Array Data for Optimal Reception," IEEE Transactions on Geoscience Elect., Vol. GE-6, May, 1968.

[41] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," Proceedings of the IEEE, Vol. 57, No. 10, October, 1969.

[42] H. Kobayashi, "Iterative Synthesis Methods for a Seismic Array Processor," IEEE Transactions on Geoscience Elect., Vol. GE-8, July, 1970.

[43] D. G. Childers, N. W. Perry, Jr., and J. R. Bourne, "Brain Waves and Cerebral Information Processing in Vision," International Congress on Cybernetics, Namur, Belgium, September, 1970.

[44] J. R. Bourne, "Spatio-Temporal Characteristics of the Visual Evoked Response in Man," Ph.D. Dissertation, University of Florida, 1969.

[45] J. S. Barlow and T. Estrin, "Comparative Phase Characteristics of Induced and Intrinsic Alpha Activity," Electroencephalography and Clinical Neurophysiology, Vol. 30, June, 1971.

[46] N. W. Perry, Jr., and D. G. Childers, The Human Visual Evoked Response, Method and Theory, Charles C. Thomas Publisher, Springfield, Ill., 1969.

[47] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, John Wiley and Sons, Inc., New York, 1965.

## BIOGRAPHICAL SKETCH

Owen Simeon Halpeny was born October 3, 1940, at Covington, Kentucky. In June, 1958, he was graduated from Covington Catholic High School. In January, 1963, he received the degree of Bachelor of Science in Electrical Engineering from the University of Kentucky. Following his graduation he served in the United States Air Force, with assignments at Patrick Air Force Base, Florida, and London, England. In September, 1969, he began work toward the degree of Master of Science in Engineering at the University of Florida. In December, 1970, he received the degree of Master of Science in Engineering from the University of Florida and remained at the University of Florida to work toward the degree of Doctor of Philosophy. He has held a Research Assistantship from April, 1970, until the present time.

Owen Simeon Halpeny is married to the former Carol Ann Pace and has one son, Michael Owen. He is a member of Eta Kappa Nu, IEEE, and Sigma Xi.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D. G. Childers

Donald G. Childers, Chairman
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

T. S. George (proxy)

Theodore S. George
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

N. W. Perry, Ps

Nathan W. Perry
Professor of Clinical Psychology

This dissertation was submitted to the Dean of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August, 1972

Robert E. Uhrig

Dean, College of Engineering

A. G. Smith

Dean, Graduate School