ARL-TR-88-19

Copy No. 62

BISPECTRAL INVERSION: THE CONSTRUCTION OF A TIME SERIES FROM ITS BISPECTRUM

Peter S. Allison

APPLIED RESEARCH LABORATORIES
THE UNIVERSITY OF TEXAS AT AUSTIN
POST OFFICE BOX 8029, AUSTIN, TEXAS 78713-8029

13 April 1988

Technical Report

Prepared for:

APPLIED RESEARCH LABORATORIES
THE UNIVERSITY OF TEXAS AT AUSTIN
AUSTIN, TX 78713-8029

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ARL-TR-88-19	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Applied Research Laboratories 6b. OFFICE SYMBOL (If applicable) ARL:UT	7a. NAME OF MONITORING ORGANIZATION Applied Research Laboratories
6c. ADDRESS (City, State, and ZIP Code) The University of Texas at Austin P.O. Box 8029 Austin, Texas 78713-8029	7b. ADDRESS (City, State, and ZIP Code) The University of Texas at Austin P.O. Box 8029 Austin, Texas 78713-8029
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER ARL:UT Independent Research and Development Program
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO
11 TITLE (Include Security Classification) Bispectral Inversion: The Construction of a Time Series from its Bispectrum	
12 PERSONAL AUTHOR(S) Allison, Peter S.	
13a TYPE OF REPORT technical 13b TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 88-4-13 15 PAGE COUNT
16 SUPPLEMENTARY NOTATION	
17 COSATI CODES FIELD / GROUP / SUB-GROUP	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) bispectrum, bispectral inversion, time series, bicovariance

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report presents a method of producing a stochastic time series that has any desired bispectral characteristic. This was done using a quadratically nonlinear, infinite order, moving average model whose bicovariance was obtainable in terms of a finite linear function of the kernels. This expression was inverted to produce an equation for the kernels in terms of the bicovariance. Knowing the kernels, the time series can easily be computed. Alternately the DFT of the time series can also be computed directly. This is actually easier than the time domain computation, as it only involves one summation as opposed to two.

Keywords: Computer programs. (KR)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Marshall E. Frazer	22b TELEPHONE (Include Area Code) 512-835-3450 22c OFFICE SYMBOL SPG

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

This material will be submitted in May 1988 as a thesis, in partial fulfillment of the requirements for the degree of Master of Science in Engineering, The University of Texas at Austin.

ACKNOWLEDGMENTS

I would first like to express my sincere appreciation to Dr. Gary R. Wilson for his guidance and support in the preparation of this thesis. His willingness to provide many helpful suggestions was invaluable. I would also like to thank Professor Edward J. Powers both for what he has taught me about time series analysis and for his insightful comments in regard to this thesis.

I would also like to thank Dr. Loyd Hampton and Dr. Thomas A. Griffy for their aid through the Ocean Acoustics Scholarship Program. In addition Dr. Griffy's willingness to read this manuscript is much appreciated. Finally, I would express appreciation to Dr. Melvin J. Hinich for his sincere interest and helpful conversations.

February 1988

CHAPTER 1

INTRODUCTION

The objective of this thesis is to present the results of an investigation into the problem of producing a discrete stochastic time series that has any desired bispectral characteristic. In essence this amounts to an *inversion* of the bispectrum; however, the resulting time series is not guaranteed to be unique in any way except that it has the specified bispectrum.

Typically most papers dealing with the bispectrum or its applications are interested in how one estimates the bispectrum from a sampled subset of the underlying random process [e.g., Brillinger and Rosenblatt, 1967b; Rao and Gabr, 1984; Raghuveer and Nikias, 1986; Nikias and Raghuveer, 1987]. This thesis is interested in exactly the opposite problem, i.e., how does one obtain or estimate the times series knowing only its bispectrum. While this is not a well studied problem, fortunately much can be done by a straightfoward generalization from the power spectrum, which is very well known. In addition there is one advantage to inverting the bispectrum, that is not present in the bispectral estimation problem. It is that one can assume to have the *true* bispectrum and not just an estimate of it, unlike the reverse problem where one cannot generally assume to have all the representations of a random process.

The inversion can be accomplished following a procedure first suggested by Wolinsky [1988]. The starting point is to assume a model for the time series. There are many models one could choose for this purpose. However, most involve very complex and often unsolvable relations between the model parameters and the bispectrum. But since the choice is open it only makes sense to choose one that will make the inversion as easy as possible. Accordingly the model

$$x(t) = \eta(t) + \varepsilon(t) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\,\varepsilon(t+r)\,\eta(t+s) \qquad (1.1)$$

has been chosen, for reasons which will be explained more fully in chapter 3. Here x(t) is the desired time series, $\varepsilon(t)$ and $\eta(t)$ are Gaussian independent identically distributed (i.i.d.) random variables with zero mean and variance σ^2, and g(r,s) is called the kernel function. This is a slightly different model than the one suggested by Wolinsky, but the resulting relation of the kernels to the bicovariance is essentially the same. They both possess a finite linear relation between the kernel function and the bicovariance. Accordingly we will define a loose class of models called the universal bispectral model (UBM) whose members have this characteristic. This term is due to Wolinsky, but he applied it to only one specific model. The main features belonging to this particular model, which are not found in Wolinsky's, are that its kernel function can be made to be symmetric in r and s, i.e., g(r,s) = g(s,r); the expression relating the bicovariance to the kernel function does not involve any shifting of the bicovariance; and, third, the power spectrum of the time series can be changed by more than an additive constant without changing the bispectrum. The

ability to alter the power spectrum is the result of an extra degree of freedom in the kernel function.

From the model equation in (1.1), expressions for the kernel function g(r,s) can be derived in terms of either the bispectrum or the bicovariance. These calculations, which are done in the case of discrete time/continuous frequency, constitute the heart of chapter 3. In chapter 4 the subtleties of approximating the discrete time/continuous frequency case with discrete time/discrete frequency Fourier transforms are discussed and these results are applied to some specific cases. The discrete time/discrete frequency is found to place some restrictions on the arbitrariness of the bispectrum.

There are two applications of bispectral inversion that the writer is aware of, but potentially there are many more. One application would be to test bispectral estimators. If one had a time series with a known bispectrum, one could then test an estimator to see how noise affected the convergence time, the amplitude, or the position of the peaks of the estimated bispectrum. Another application would be in the area of communications. A communications system using linear signals, for instance, requires that multiple transmissions be separated in either a time window or a frequency bandwidth. In other words, multiple signals of distinct frequency bandwidths can be sent at the same time and still be recoverable, and multiple signals occupying the same frequency bandwidth can be sent in different time windows and still be resolvable. But of course multiple linear signals cannot share both the same frequency band and the same time window and still transmit recoverable information. With nonlinear signals, however, this is not necessarily true. Multiple nonlinear

signals occupying identical frequency bands and occurring simultaneously can be recovered and separated provided there exists a nonlinear characteristic that *is* different. The bispectrum is such a characteristic. Thus by exploiting the bispectral characteristics of the signals, a communications channel could simultaneously transmit a second or a third signal in the same frequency band as the first and still be able to demultiplex each signal without ambiguity. The use of nonlinear multiplexing to encode signals is the topic of a paper by Hinich [1979]. However, in Hinich's paper it is the multiplexing operation that is nonlinear and not the signal itself. In either case, the end result is to increase the number of signals that can be transmitted simultaneously without increasing either the required bandwidth or the duration of the time window the signals occupy. The only price to be paid is the additional computation required to estimate the bespectrum.

Before proceeding to the derivation of the g-kernels it would be well, considering the relative newness of the bispectrum to engineering applications, to present a review of the bispectrum and its characteristics.

CHAPTER 2

BISPECTRAL TUTORIAL

Essentially the bispectrum is one component in the set of statistical quantities called higher order spectra. These higher order spectra are the frequency domain equivalent of higher order statistical quantities, called *cumulants*, in that they are Fourier transform pairs. The higher order spectra, or *polyspectra*, provide useful information about nonlinear systems. Just as linear systems can be analyzed by linear regression techniques using the power spectrum, so in an entirely analogous manner nonlinear systems can be analyzed by multiple regression techniques using polyspectra. In the event there is only one time series, the second order spectra reduces to the ordinary power spectrum, while the third order polyspectra is called the *bispectrum*. If there are two time series the above quantities become the cross spectrum and the cross bispectrum, respectively. Likewise other polyspectra can be defined, such as the fourth order spectra which is commonly called the *trispectrum*. However the higher orders are increasingly more complex and difficult to actually compute. Only with the advent of faster hardware will polyspectra of greater order be practical to compute. In addition polyspectra beyond the trispectrum involve more than three dimensions making them very difficult, if not impossible, to visualize.

The purpose of this chapter is, first, to provide some background on polyspectra culminating in a definition of the bispectrum, and secondly to introduce

some of its mathematical properties, paying particular attention to its symmetries. The important topic concerning estimation of the bispectrum will be passed by without much comment as it is not particularly germane to this thesis.

2.1 Definitions: Random Processes, Cumulants, Moments and Polyspectra

Loosely speaking, detailed knowledge of the topics in measure theory on which the theory of random variables and random processes rests, and from which the measure theoretic definitions of expected value and other statistical quantities come, is not necessary in the actual estimation of polyspectra. However it is good to have some idea where these concepts come from. Hence this section starts with a very brief definition of random processes. Further details can be found in works such as Priestly [1981] or, for the more ambitious, Loeve [1977].

2.1.1 Random Processes: The First Building Block

A random process is defined as a collection of Borel measurable functions (random variables) $\{ X_t, t \in T \}$ indexed by a parameter t belonging to a set T, and the measure is some probability measure defined on some probability space. If T is the set of real numbers $\{ t : -\infty < t < \infty \}$, denoted by $\Re$, then the process is called continuous. If T is the set of natural numbers { t : . . . -2,-1,0,1,2. . . }, denoted by N, then $\{X_t\}$ is a discrete parameter process. If t is associated with time, the process is called a time series. While this is the most common association, it is not necessarily the only one. This thesis will deal primarily with a discrete parameter process. It is worth noting that there are two parameters involved in a random

process. The obvious one is t, and the not so obvious one is the subset of the set $\Omega = \{\omega : \omega \in \Omega\}$ which X maps to $\Re$. Thus for any fixed t the random process reduces to a random variable, while for any fixed $\omega \in \Omega$ the random process becomes a simple function of t sometimes called a *sample path* or *trajectory*. The reason ω is not commonly written will be clear shortly.

The k dimensional probability distribution function is defined as a function mapping $\Re^k$ to $\Re$ such that

$$F_{X(t_1), X(t_2), \ldots X(t_k)}[x_1, \ldots x_k] = P\{ X(t_1) \leq x_1, \ldots, X(t_k) \leq x_k \} \qquad (2.1)$$

where P is the probability measure mentioned above and $k \in N$ [Wise, 1986]. In the foregoing definitions probability spaces and measures have been used, but have not been *explicitly defined*. The reason is due to Kolmogorov who proved that given a set of finite dimensional distribution functions satisfying the necessary requirements, there exists a probability space and a probability measure on that space whose family of finite dimensional distribution functions coincides with the given family of finite dimensional distributions [Wise, 1986]. Thus in practice it is not necessary to show the existence of probability spaces or measures. Having the distribution function guarantees their existence. Since all statistical properties are completely specified by the distribution function, neither is it necessary to find the probability measure or space in order to find moments or cumulants. It is for this same reason that ω is not written explicitly.

A strictly stationary random process is one where, for $k \in N$,

$$F[X(t_1+t), X(t_2+t), \ldots, X(t_k+t)] = F[X(t_1), X(t_2), \ldots X(t_k)]. \qquad (2.2)$$

This means that the distribution function does not change when all the index parameters t are shifted an equal amount up or down the axes in $\Re^k$. A less stringent restriction is stationarity to order n. A random process is considered stationary to order n, $n \in N$, if Eq. (2.2) holds for any $k \leq n$.

2.1.2 Cumulants and Moments

The final definition before reaching the bispectrum is that of cumulants. Following Rosenblatt [1981] we define the moment generating function (also called the characteristic function) of the vector random process $\{X_{a_1}(t_1)...X_{a_k}(t_k)\}$ as

$$\Phi(\psi)= \Phi(\psi_p...\psi_k) = E\{e^{j\sum_{i=1}^{k}\psi_i X_{a_i}}\} \quad . \tag{2.3}$$

The joint moments can be found as the coefficients of $(\psi_1,...\psi_k)$ in the Taylor series expansion of $\Phi(\psi)$ about the origin provided the moments are finite. As Brillinger [1975] points out, however, this is not really a problem since all time series available for processing are bounded. The cumulants of the same set of random variables are then defined as the coefficients of the Taylor series expansion of the log $\Phi(\psi)$ about zero. Explicitly this gives for the relation of the joint cumulant sequence of order r, where $r=n_1+n_2+ \ldots n_k$ [Nikias *et al.*,1987]:

$$C^{n_1 n_2 \ldots n_k}_{a_1 a_2 \ldots a_k}(t_1,...t_k) \equiv (-j)^r \frac{\partial^r \{\ln \Phi(\psi_1,\psi_2 \ldots \psi_k)\}}{\partial\psi_1^{n_1}\partial\psi_2^{n_2}...\partial\psi_k^{n_k}}\Bigg|_{\psi_1=\psi_2=...=\psi_k=0} \quad . \tag{2.4}$$

The t_j refers to the random variable specified by evaluating X_{a_j} at t_j. Generally cumulants (and moments) are regarded as constants. Random processes, however, are a collection of random variables of which each combination has a cumulant.

Hence the r^{th} order cumulant of the k-variate random process can be viewed as a sequence in $\Re^k$ indexed by the same k parameters as the random vector forming the cumulant or moment. The corresponding expression for the joint moment sequence of order r, which is defined as

$$m^{n_1 \ldots n_k}_{a_1 \ldots a_k}(t_1 \ldots t_k) \equiv E\{x_{a_1}(t_1)^{n_1} x_{a_2}(t_2)^{n_2} \ldots x_{a_k}(t_k)^{n_k}\} \quad , \tag{2.5}$$

is simply

$$m^{n_1 \ldots n_k}_{a_1 \ldots a_k}(t_1 \ldots t_k) = (-j)^r \frac{\partial^r \{\Phi(\psi_1, \psi_2 \ldots \psi_k)\}}{\partial \psi_1^{n_1} \partial \psi_2^{n_2} \ldots \partial \psi_k^{n_k}} \Bigg|_{\psi_1 = \psi_2 = \ldots = \psi_k = 0} \tag{2.6}$$

where a_j, $j \in N_k$ specifies the random process and n_j specifies the power to which the a_j^{th} process is raised. Further examination of these expressions can provide some intuitive explanation of the difference between moments and cumulants. Dr. Choi's Ph.D. dissertation provides good insight on this point [Choi, 1984]. If any of the random variables in (2.3) are independent, the expectation in (2.3) will factor into the product of at least two expectations. Each term will be a function of only those random variables that are independent of all the other random variables. This is to say that $\Phi(\psi)$ can be written as $\Phi(\psi)=\Phi_1(\psi_1)\Phi_2(\psi_2)$ where the arguments of Φ_1 and Φ_2 are two disjoint sets of variables. The partial derivative in (2.6) is not affected by this factoring. However, when the logarithm of this factored expression for $\Phi(\psi)$ is taken, a sum will result. The partial derivative in (2.5), to which the cumulants are related, *is* affected by the factoring. When (2.3) can be factored, (2.5) becomes identically zero. (Notice that the logarithm does not commute with the expected value operator; the definition of a cumulant is the log of an expected value, not the

expected value of a log.) Thus while the cumulant provides information about the dependence of the random variables [Choi, 1984], the moments provide information about the joint distribution functions. This information is the statistical analog to the information provided by the moment of inertia about the distribution of mass in a physical system. Some of the important properties of cumulants obtainable from (2.3) and (2.4) are repeated here [Rosenblatt, 1981]:

1) $C\{X_1(t_1),...,X_k(t_k)\}$ is symmetric in all of its arguments.

2) If any group of X's are independent of the others,
$C\{X_1(t_1),...,X_k(t_k)\} = 0.$

3) For any two independent groups $\{X_1, ..., X_n\}$ and $\{X_{n+1}, ...,X_k\}$
$C\{X_1(t_1) + X_{n+1}(t_{n+1}),...,X_n(t_n) + X_k(t_k)\} = C\{X_1(t_1),...,X_n(t_n)\}$
$+ C\{X_{n+1}(t_{n+1}),...,X_k(t_k)\}.$

4) For a set of constants $\{a_1...a_k\}$,
$C\{a_1X_1(t_1)...a_kX_k(t_k)\} = a_1...a_kC\{X_1(t_1)...X_k(t_k)\}.$

As alluded to earlier, the polyspectra are defined as Fourier transform pairs with the cumulants. However, since characteristic functions are in general not available, the above expressions are not useful in the actual computation of cumulants. Since they are also not directly computable from representations of a random process [Kendall *et al.*, 1958], it is of interest to obtain relations for cumulants in terms of the moments which are directly computable from ensemble

representations of a time series. These relations are obtained by taking the logarithm of the Taylor series expansion of (2.3) (whose coefficients are the moments) and expanding it again as a Taylor series about the origin. The coefficients of this expansion are the cumulants by definition. This is worked out in detail in Kendall *et al.* [1958]. The cumulants can then be found by suitable matching of terms. An expression for the moments in terms of the cumulants can be found in an analogous manner. This derivation was done for the general case by Leonov and Shriyaev in 1959. In spectral analysis applications, however, only the *simple* moments are of any interest. Simple moments are those where $n_1=n_2=...=n_k=1$. Hence we can use much simpler results, which are only valid in two special cases.

Case 1:

$$n_1 = n_2 =...= n_k = 1 \quad \text{(simple moments)}$$

Case 2:

$$n_2 = ... = n_k = 0, \quad n_1 \text{ arbitrary.}$$

With these restrictions we have from their results:

$$E\{x_1(t_1)\ldots x_k(t_k)\}=\sum_{\upsilon} c(\upsilon_1)...c(\upsilon_k) \tag{2.7}$$

where $\upsilon = (\upsilon_1...\upsilon_k)$, υ_j is a partition of the integers $\{1,2,...,k\}$, and the summation is over all such partitions. Likewise the expression for the cumulants in terms of the moments is

$$c(x_1(t_1)\ldots x_k(t_k))=\sum_{\upsilon}(-1)^{p-1}(p-1)!m(\upsilon_1)\ldots m(\upsilon_p) \qquad (2.8)$$

where the υ and the summation are the same as before and m is a simple moment about the origin. Thus simple cumulants of order k can be expressed as polynomials of simple moments of order no greater than k and the moments can be expressed as polynomials of the cumulants of order no greater than k. The information contained in the excluded cases (i.e., non-simple moments) is not being thrown away. It is actually already included as a special case in a higher order (r^{th} order) simple moment. For instance, $m_{12}^{21}(t_1,t_2) = m_{112}^{111}(t_1,t_1,t_2)$. The second case, *i.e.*, ($n_2 = \ldots = n_k = 0$, n_1 arbitrary) is simply a special case of the first one, namely the univariate. In addition, for a stationary time series it turns out to always be a constant. Hence, without much danger of confusion the notation can be simplified by dropping the superscripts since they are always one in the cases of interest. By way of illustration consider the simple fourth order cumulant of a univariate random process (that is, $X_1= X_2= \ldots X_k$). According to (2.8) we have

$$\begin{aligned} c_{1111}(t_1,t_2,t_3,t_4) &= m_{1111}(t_1,t_2,t_3,t_4) - m_{11}(t_1,t_2)m_{11}(t_3,t_4) - m_{11}(t_1,t_3)m_{11}(t_2,t_4) \\ &\quad - m_{11}(t_1,t_4)m_{11}(t_2,t_3) - m_1(t_1)m_{111}(t_2,t_3,t_4) - m_1(t_2)m_{111}(t_1,t_3,t_4) \\ &\quad - m_1(t_3)m_{111}(t_1,t_2,t_4) - m_1(t_4)m_{111}(t_1,t_2,t_3) \\ &\quad + 2\{m_1(t_1)m_1(t_2)m_{11}(t_3,t_4) + m_1(t_1)m_1(t_3)m_{11}(t_2,t_4) \\ &\quad + m_1(t_1)m_1(t_4)m_{11}(t_2,t_3) + m_1(t_2)m_1(t_3)m_{11}(t_1,t_4) \\ &\quad + m_1(t_2)m_1(t_4)m_{11}(t_1,t_3) + m_1(t_3)m_1(t_4)m_{11}(t_1,t_2)\} \end{aligned}$$

$$- 6m_1(t_1)m_1(t_2)m_1(t_3)m_1(t_4) \quad . \tag{2.9}$$

If we now assume that the random process is zero mean, i.e., m(t)=0, and that it is stationary to at least fourth order, (2.9) becomes much simpler. A direct consequence of the application of the definition of stationarity to the expected value operator ($E[x] = \int_{\Omega} X \, dF$) is that moments of stationary random processes can be written in terms of time differences. If the time lag is chosen to be $\tau_{n-1} = t_n - t_1$ for $n \in \{1,2,...,k\}$, then the k^{th} order moment can be written in terms of k-1 lags because the lag corresponding to n=1 has become identically 0 and can be dropped. Again the notation can be simplified. By convention, when dealing with a univariate sequence, the moment we have written as $m_{11...1}$ is often written as $m_{1+1+...+1}$. Following this convention (2.9) simplifies to

$$c_4(\tau_1,\tau_2,\tau_3) = m_4(\tau_1,\tau_2,\tau_3)$$

$$- m_2(\tau_1)m_2(\tau_3-\tau_2) - m_2(\tau_2)m_2(\tau_3-\tau_1) - m_2(\tau_3)m_2(\tau_2-\tau_1) \quad . \tag{2.10}$$

Now suppose that the random variables defined by fixing τ_2 and τ_3 are independent of those at 0 and τ_1. Then the expectation of the first term on the right side of (2.10) will factor making $m_4(\tau_1,\tau_2,\tau_3)=m_2(\tau_1)m_2(\tau_3-\tau_2)$ and the first two terms in (2.10) would cancel out. The last two terms become zero, because the mean is zero, causing the whole cumulant to vanish. The nonstationary cumulant in (2.9) would vanish as well. If one random variable were independent of some, but not all of the others, then some but not all the terms in (2.9) or (2.10) would cancel. Thus the cumulant measures the *degree* of dependence between the random variables. Table 1 provides a summary of the relation between cumulants and moments up to order four. The table

is written in the univariate notation, but can be generalized to the multivariate case by simply replacing term for term the univariate moment (cumulant) with the desired moment (cumulant).

Table 1 Relation Between Cumulants and Moments

$$C_1 = m_1 = 0$$

$$C_2(\tau_1) = m_2(\tau_1)$$

$$C_3(\tau_1,\tau_2) = m_3(\tau_1,\tau_2)$$

$$C_4(\tau_1,\tau_2,\tau_3) = m_4(\tau_1,\tau_2,\tau_3) - m_2(\tau_1)m_2(\tau_3-\tau_2) - m_2(\tau_2)m_2(\tau_3-\tau_1) - m_2(\tau_3)m_2(\tau_2-\tau_1)$$

2.1.3 Polyspectra

Using the results of the last two sections the formal definition of polyspectra follows very easily. If a k-dimensional random process satisfies the stationarity requirement given in (2.2) and the moment $m_{a_1 \ldots a_k}(t_1 \ldots t_k)$ exists, then, following Brillinger [1965], the k^{th} order cumulant can be written:

$$C_{a_1 \cdots a_k}(t_1 \ldots t_k) = \ldots \iiint_{-\infty}^{\infty} g_{a_1 \cdots a_k}(f_1 \ldots f_k) e^{j2\pi((ft)_1 \ldots (ft)_k)} \delta(f_1 + \ldots + f_k)\, df_1 \ldots df_k \quad (2.11)$$

where $g_{a_1 \cdots a_k}(f_1 \ldots f_k)\delta(f_1 + \ldots + f_k)$ is the k^{th} order *cumulant spectral density* function. Since stationarity causes the spectral density to be zero except when $f_k = -f_1 - f_2 - \ldots - f_{k-1}$, the k^{th} order density can be written as a function of k-1 arguments with the omitted k^{th} argument understood to be the sum of the first k-1 frequencies. The expression in (2.11) can be inverted to give an expression for the spectral density in terms of the cumulant; or equivalently, the k^{th} order cumulant is the Fourier transform of the k^{th} order polyspectra. If we first integrate with respect to f_3 using the dirac function, we can rewrite (2.11) as

$$C_{a_1 \cdots a_k}(t_1 \ldots t_k) = \ldots \iint_{-\infty}^{\infty} g_{a_1 \cdots a_k}(f_1, \ldots, f_{k-1}) e^{j2\pi(f_1\tau_1 \ldots f_{k-1}\tau_{k-1})} df_1 \ldots df_{k-1} \quad (2.12)$$

where f_k is not written but is understood to be $f_k = -f_1 - f_2 - \ldots - f_{k-1}$ and $\tau_n = t_n - t_k$. Note that the right side is a function of τ only. This produces the same result obtained earlier for stationary time series; namely, the cumulant can be written in term of time differences. Thus the frequency domain equivalent of writing moments in term of time lags is that the spectra is zero except at points where all the frequencies sum to zero. Continuing the inversion of the Fourier transform gives for the spectral density function of order k-1 for a stationary time series:

$$g_{a_1 \cdots a_k}(f_1 \ldots f_{k-1}) = \ldots \iint_{-\infty}^{\infty} C_{a_1 \cdots a_k}(\tau_1 \ldots \tau_{k-1})\, e^{-j2\pi(f_1\tau_1 \ldots f_{k-1}\tau_{k-1})} d\tau_1 \ldots d\tau_{k-1} \ . \quad (2.13)$$

For the univariate case with k=2 we have from the table relating cumulants and moments that $C_{11}(\tau)=m_{11}(\tau)$. Since for a real time series $m_{11}(\tau)$ is just the autocorrelation, we see that the second order cumulant spectra reduces to the power spectrum as stated earlier. When k=3 the spectral density function is called the bispectrum. In the bivariate case, i.e., $C_{12}(\tau)$ or $C_{112}(\tau_1,\tau_2)$, the spectral density is called the cross-power or bispectrum. In this thesis we will be interested exclusively in the bispectrum. In particular we will be interested in the discrete frequency / discrete time bispectrum. The objectives of the next section will be to bridge the gap between the discrete and continuous versions of the bispectrum and to present some of the details of the bispectrum that will be important in the inversion process.

2.2 *The Bispectrum*

In concentrating on (2.13) for the specific case where k=3 we will make some simplifying assumptions. From this point onward we will restrict ourselves to real, stationary, zero mean time series. The assumption of reality is very easily justified. Any experimentally measured time series is, of course, real. They do not necessarily satisfy the other two assumptions though. The zero mean assumption is easily achieved by simply subtracting the mean. The stationarity assumption, on the other hand, is rarely, if ever, completely satisfied in experimental data. It is an approximation that is valid over "short" intervals. But in the case being investigated by this thesis the actual model generating the series is known exactly. Hence we can say without approximation that the time series is stationary. With these assumptions

let us first rewrite (2.13) in convenient, more common notation. Since X(t) is real we can equate

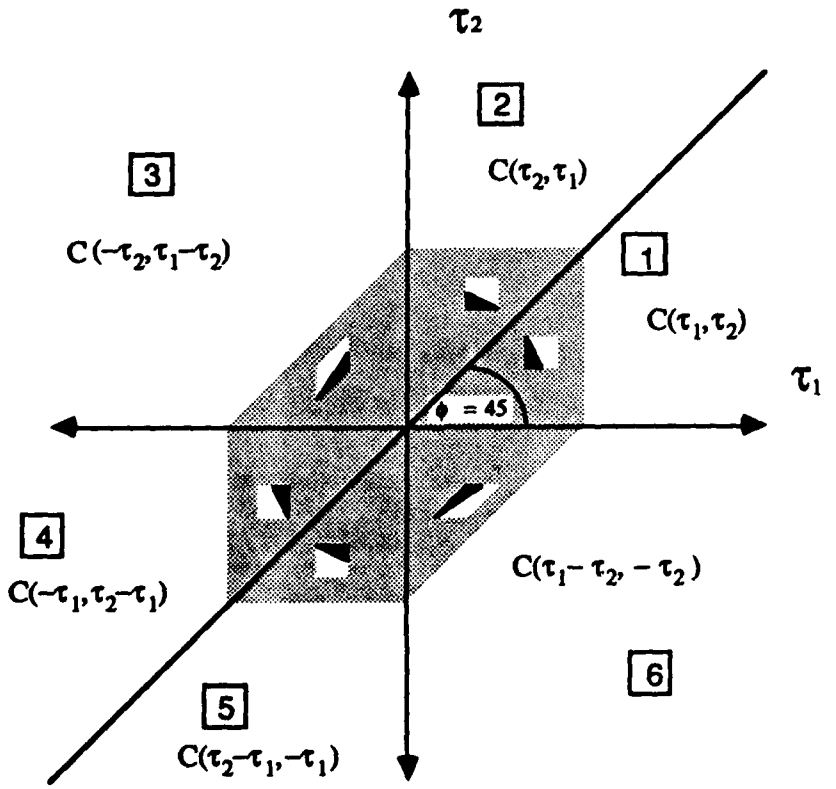
$$C_3(\tau_1,\tau_2) = m_3(\tau_1,\tau_2) = E[\, X(t)X(t+\tau_1)X(t+\tau_2)] \int R(\tau_1,\tau_2) \quad .$$

When this equivalence is applied to (2.13) in the specific case of the bispectrum, it gives:

$$B(f_1,f_2) = \iint_{-}^{\infty} E[\, X(t)X(t+\tau_1)X(t+\tau_2)]e^{-j2\pi(f_1\tau_1+f_2\tau_2)}d\tau_1 d\tau_2 \qquad (2.14)$$

where B is the bispectrum and the integral is the two-dimensional Fourier transform of the third order moment. Fig. 2.1 shows the symmetry regions of the third order moment sequence, $C_3(\tau_1,\tau_2)$.

Fig. 2.1 Symmetry of the Third Order Moment



a) Shaded area shows non zero region for finite time series
b) Boxes designate symmetries for a real time series

Equation (2.14) gives a pleasing analytical expression for the bispectrum, but it is not the most convenient expression to use to compute the bispectrum. A better one can be obtained by going back to (2.11) and inverting the k-dimensional Fourier transform directly to give

$$g_3(f_1,f_2,f_3)\delta(f_1+f_2+f_3) = \iiint_{-\infty}^{\infty} C_3(t_1,t_2,t_3)\, e^{-j2\pi(f_1t_1 + f_2t_2 + f_3t_3)} dt_1\, dt_2 dt_3 \,. \quad (2.15)$$

Making the two substitutions

1) $$C(t_1,t_2,t_3) = E\left[X(t_1)X(t_2)X(t_3)\right]$$

2) $$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi ft} df$$

and interchanging the order of integration allows (2.15) to be rewritten as:

$$g_3(f_1,f_2,f_3)\delta(f_1+f_2+f_3) = \iiint_{-\infty}^{\infty} d\hat{f}_1 d\hat{f}_2 d\hat{f}_3 \iiint_{-\infty}^{\infty} dt_1 dt_2 dt_3$$

$$\times \quad E\left[X(\hat{f}_1)X(\hat{f}_2)X(\hat{f}_3)\right] e^{-j2\pi\left\{(f_1-\hat{f}_1)t_1 + (f_2-\hat{f}_2)t_2 + (f_3-\hat{f}_3)t_3\right\}} \,. \quad (2.16)$$

Carrying out the the three integrations with respect to t produces the product of three Dirac functions $\delta(f_1-\hat{f}_1)\,\delta(f_2-\hat{f}_2)\,\delta(f_3-\hat{f}_3)$. This makes the integration with respect to f trivial, giving for (2.16)

$$g_3(f_1,f_2,f_3)\delta(f_1+f_2+f_3) = E\left[X(f_1)X(f_2)X(f_3)\right] \,. \quad (2.17)$$

Thus $E\left[X(f_1)X(f_2)X(f_3)\right] = 0$ unless all three frequencies sum to zero. Taking this into account, (2.17) can be rewritten to give the more revealing expression for the bispectrum

$$B(f_1,f_2)\delta_f(0) = E\left[X(f_1)X(f_2)X^*(f_1+f_2)\right] \,. \quad (2.18)$$

From this expression one can see that the bispectrum is nonzero only when the frequency pair (f_1, f_2) and its sum (f_1+f_2) are coupled so as not to be statistically independent. The bispectrum is a measure of quadratic nonlinearity because it is a quadratic interaction between f_1 and f_2 which will produce a third frequency (f_1+f_2) that is statistically dependent to the first two. From the expression in (2.18) it is a simple matter to determine the symmetries of the bispectrum, although they can also be found from (2.14) using the symmetry of the third order moment shown in fig 2.1.

2.2.1 Symmetries of the Continuous Frequency Bispectrum

From (2.18) it is clear that f_1 and f_2 are interchangeable.

Property 1:

$$B(f_1,f_2) = B(f_2, f_1).$$

This produces the line of symmetry labeled 1 in fig 2.2. Substituting $f_1 \rightarrow (-f_1)$ and $f_2 \rightarrow (-f_2)$ into (2.18) and using the fact that $X(f) = X^*(-f)$ gives a second symmetry.

Property 2:

$$B(f_1,f_2) = B^*(-f_1,-f_2).$$

This produces the line of symmetry labeled 2 in fig 2.2. Two additional relations can be found by first substituting in $f_1 \rightarrow (f_1+f_2)$ and either $f_2 \rightarrow (-f_2)$ or $f_2 \rightarrow (-f_1)$ and then applying the conjugate property of the Fourier transform.

Property 3:

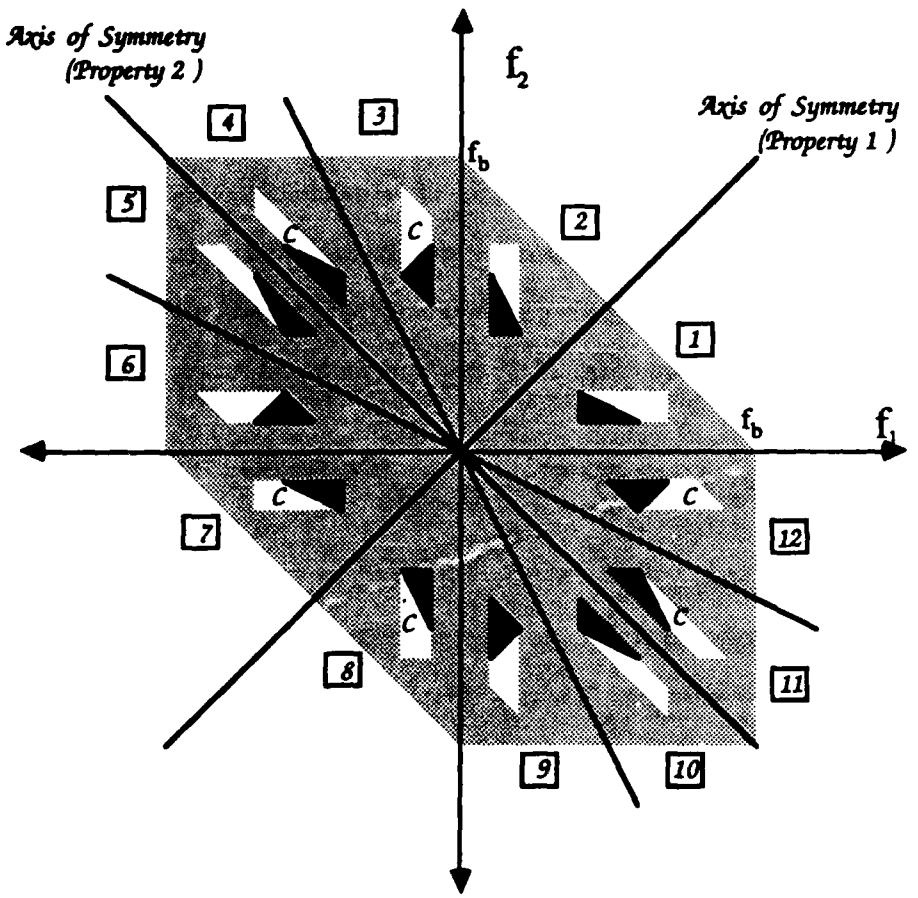
$$B(f_1,f_2) = B^*(f_1+ f_2,- f_1)$$
$$B(f_1,f_2) = B^*(f_1+ f_2,- f_2) \quad .$$

Application of property 2 to each of the terms in property 3 produces two more terms. This gives five terms in addition to property 1. Applying property 1 to these five gives five more for a total of ten terms. These symmetries divide the bispectral domain into 12 equivalent regions as shown in fig 2.2. Although the regions do not appear to be of equal area, each region does contribute equally to the region of support. The shaded region represents the region of support for the bispectrum of a bandlimited process. The bispectrum is uniquely specified by knowing the values in any one of these regions. If we take the region $\{(f_1,f_2)$: $f_1 \geq f_2;\ f_2 \geq 0\}$ to be the primary region (principal domain) then the 11 other equivalent regions are given explicitly in terms of this primary region by (2.19). Thus for a bandlimited process the region of support within the principal domain is the only part of the bispectrum that needs to be computed. When arbitrarily specifying the bispectrum it should be noted that the bispectrum must be purely real when either f_1 or f_2 is zero. This is evident from (2.18) or from the fact that this is the only way to satisfy (2.19) on the boundaries between regions of the bispectrum which are complex conjugates of each other.

If computers worked with continuous time variables we could stop here. However they do not. As will be seen shortly, this discretization produces some significant changes in the the frequency plane of the bispectrum.

$$B(f_1,f_2) = \begin{cases} B(f_1,f_2) & \boxed{1} \\ B(f_2,f_1) & \boxed{2} \\ B^*(-f_2,f_1+f_2) & \boxed{3} \\ B^*(-f_1,f_1+f_2) & \boxed{4} \\ B(-f_1-f_2,f_1) & \boxed{5} \\ B(-f_1-f_2,f_2) & \boxed{6} \\ B^*(-f_1,-f_2) & \boxed{7} \\ B^*(-f_2,-f_1) & \boxed{8} \\ B(f_2,-f_1-f_2) & \boxed{9} \\ B(f_1,-f_1-f_2) & \boxed{10} \\ B^*(f_1+f_2,-f_1) & \boxed{11} \\ B^*(f_1+f_2,-f_2) & \boxed{12} \end{cases} \qquad (2.19)$$

Fig. 2.2 Bispectral Symmetry for Continuous Frequency



a) Shaded area represents the region of support for a bandlimited bispectrum whose highest frequency is f_b

b) Boxed numbers reference expressions in Eq. (2.19)

c) The mapping of an arbitrary magnitude characteristic is shown by the twelve boxes. The 'C' denotes where the bispectrum is the conjugate of the principal domain, and the triangles in the boxes reference specific corners and sides.

2.2.2 Symmetries of the Discrete Time/Continuous Frequency Bispectrum

The discrete time series causes the bispectrum to be periodic in f_1 and f_2 with a period of f_s or 1, where f_s is the sampling frequency and 1 is in terms of normalized frequency. This is a direct consequence of the periodicity of the Fourier transform of a discrete time series as can be seen from (2.18). Thus the frequencies that had to sum to zero in the continuous case now only have to sum to 0 (mod 1). The two-dimensional Fourier transform of the third order moment (bispectrum) now becomes:

$$\tilde{B}(f_1,f_2) = \sum_{\tau_1,\tau_2=-\infty}^{\infty}\sum C(\tau_1,\tau_2)e^{-j2\pi(f_1\tau_1+f_2\tau_2)} \tag{2.20}$$

where the domain of B extends over the whole real plane. Although B is defined over the whole real plane, it is now doubly periodic with respect to both f_1 and f_2 with period 1. This implies $B(f_1,f_2) = B(f_1,f_2+1) = B(f_1+1,f_2+1)$. The fundamental domain is defined as the region $\{(f_1,f_2): -\frac{1}{2} < f_1 < \frac{1}{2},\ -\frac{1}{2} < f_2 < \frac{1}{2}\}$; the principal domain continues to be that portion of the fundamental that uniquely specifies the bispectrum. The tilda is to distinguish the periodic and non-periodic versions of B. Equation (2.20) is the discrete equivalent of (2.14). The discrete equivalent to (2.15) is similar except that now because of periodicity $(f_1+f_2+f_3)$ may sum to 0 (mod 1). The dirac on the right side of (2.15) becomes $\delta(n-[f_1+f_2+f_3])$ where $n = \{0, \pm1, \pm2, \ldots\}$. The rest of the development of the discrete equivalent to (2.18) follows in a completely analogous manner allowing (2.18) to be written in the discrete case as:

$$\tilde{B}(f_1,f_2) = E[\, X(f_1)X(f_2)\overline{X(n-\{f_1+f_2\})}] \qquad (2.21)$$

where $\quad X(f) = \sum_{t=-\infty}^{\infty} x(t)e^{-j2\pi ft}$

where the overlined portion represents the modified selection rule for the third frequency. When $\{f_1,f_2\}$ lie inside the fundamental domain, the only two values of n that yield a third frequency that is also between $-\frac{1}{2}$ and $\frac{1}{2}$ are $n = 0,1$. The other values of n, which put the third frequency outside the fundamental domain, yield only redundant information since $X(f+n) = X(f)$. When n=0, (2.21) reduces to (2.18); the resulting symmetries are the same as before. When n=1, however, some new symmetries result. Substituting $f_1 \rightarrow (1- f_1- f_2)$ and $f_2 \rightarrow f_1$ or f_2 gives two additional properties.

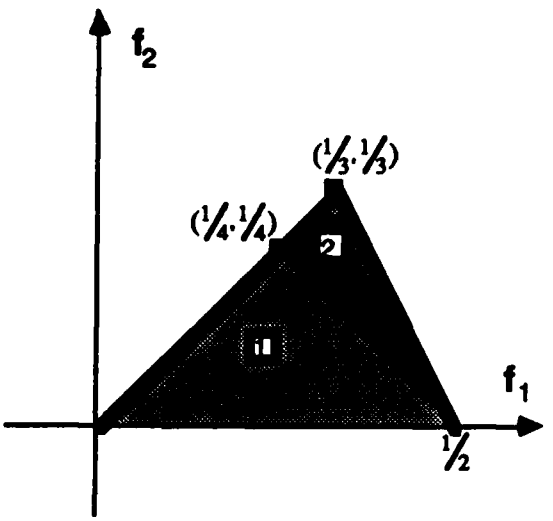
Additional Symmetries for Discrete Time:

a) $B(f_1,f_2) = B(1-f_1-f_2, f_1)$

b) $B(f_1,f_2) = B(1-f_1-f_2, f_2)$.

If f_1 and f_2 are chosen from region 1 in fig. 2.3, these additional relations simply repeat the previous ones in (2.19). If, however, they are chosen from region 2 there is no repetition; they yield new symmetries, as evidenced by the fact that both of the resulting terms still lie within region 1 on fig 2.2. It now appears that there are

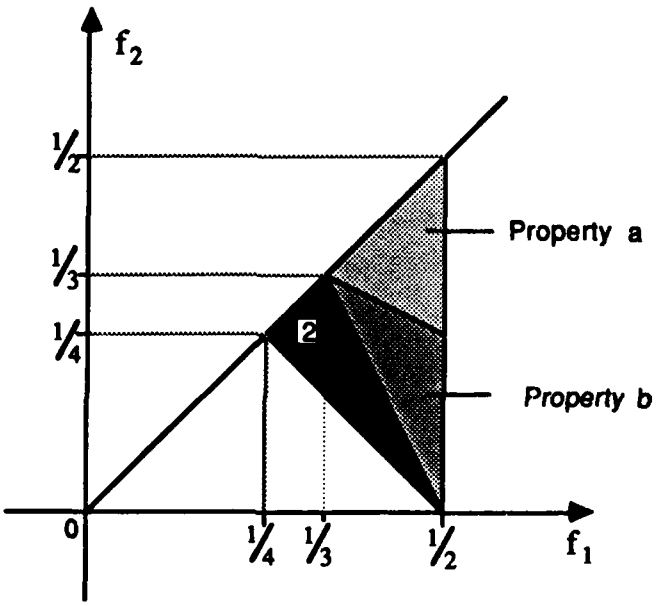
Fig. 2.3 Principal Domain of Discrete Time Bispectrum



additional symmetries within region 1 on fig. 2.2, and indeed this is the case as is shown in fig. 2.3a. Applying property 1 to the area shown in fig. 2.3a produces the three equivalent regions shown in fig. 2.3b. The use of property 2 will give six more equivalent regions in the fourth quadrant for a total of 12 additional symmetrical regions as shown in fig. 2.3b. These "extra" regions arise because of the double periodicity of the bispectrum. Using this periodicity these 12 "extra" regions can be paired with the original 12 regions shown in fig. 2.2 to produce a different

fundamental domain. This alternate fundamental domain is shown in fig. 2.4. This is the fundamental domain that would result from the application of (2.19) to frequency pairs (f_1,f_2) in both regions of the principal domain shown in fig. 2.3. This is the more usual definition of the fundamental domain. It is simpler in that it does not require the use of additional symmetry relations; however, the two are otherwise entirely equivalent. In this thesis we will work with the square domain for the simple reason that it is much easier to perform the inverse discrete Fourier transform over a square region than over a hexagonal one.

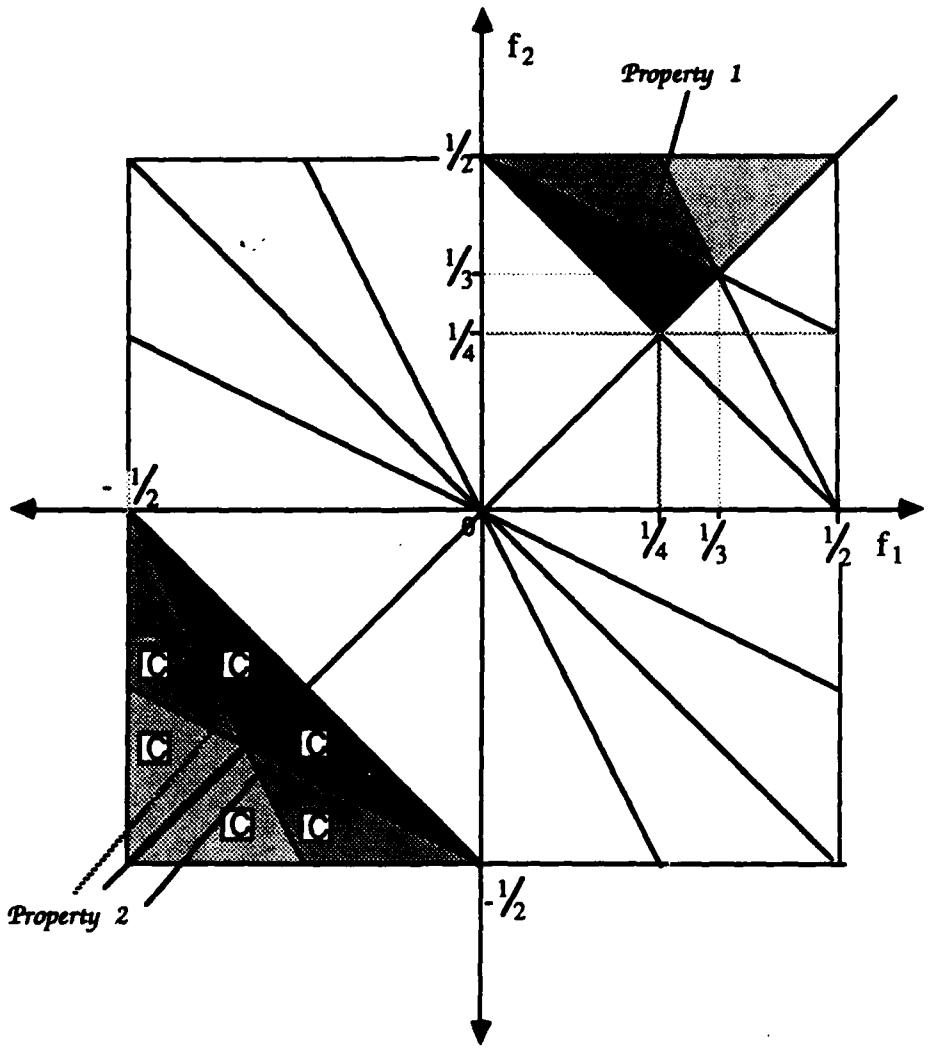
Fig. 2.3a Additional Bispectral Symmetry for Discrete Time



Notice that the additional symmetry in fig. 2.3 falls outside the region of support for a bandlimited system. This means that for a properly sampled, stationary time series region 2 in fig. 2.3 will be zero. Recently it has been suggested that a nonzero result in this region implies either a nonstationary or aliased time series [Hinich *et al.*,1988].

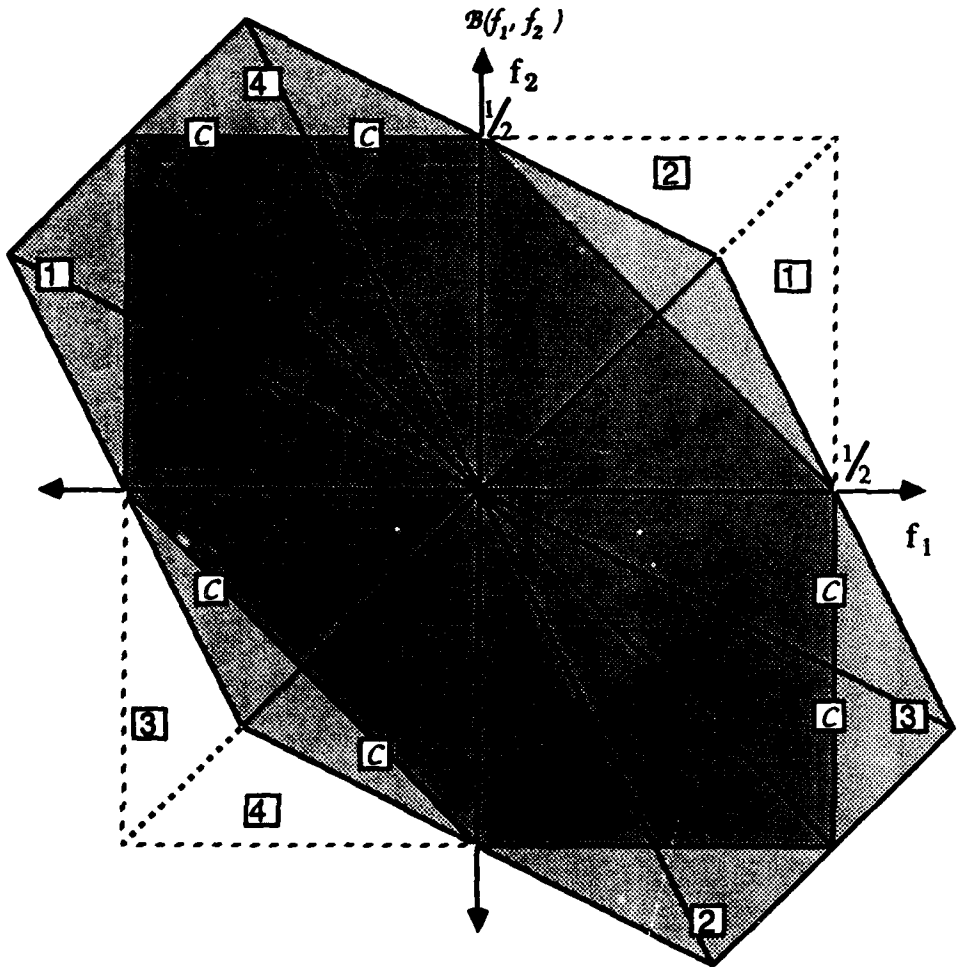
The relations defined in (2.18) and (2.14) coupled with a knowledge of the symmetries given in section 2.2.2 provide the mathematical framework for the inversion of the bispectrum. Much detail has been accorded to these symmetries because they are crucial in performing the inversion. If they are not satisfied, the third order moment obtained by transforming the bispectrum will not be real or will not have the proper symmetries for a third order moment.

Fig. 2.3b Additional Bispectral Symmetry for Discrete Time



a) The large outer box shows the fundamental domain

Fig. 2.4 Fundamental Domain of Discrete Time Bispectrum



Boxed numbers indicate equivalent regions of bispectrum

. . . Zero part of bispectrum for bandlimited process

. . . Non-zero part of bispectrum

CHAPTER 3

INVERSION OF THE BISPECTRUM

The *inversion* of the bispectrum essentially involves finding a stochastic time series whose bispectrum satisfies some predetermined constraints. For instance it might be desired to produce a time series whose bispectrum is a constant over all points or, at the other extreme, the requirement might be to have a bispectrum that is zero at every point but one. With the exception of two philosophical differences, the method used to invert the bispectrum is almost the exact reverse of that used in the more usual problem of estimating the bispectrum of a time series. The first difference is that one can assume to start with the *true* bispectrum and not just an estimate of it. When attempting to estimate the bispectrum from a finite number of ensemble representations, one generally assumes to have only part of the time series. The goal is to obtain a large enough subset of this time series so that one can obtain an estimate of the bispectrum to within some acceptable confidence level. In the present problem, however, we assume to know the bispectrum exactly. But, in general, the time series represented by this bispectrum (i.e., the one we wish to compute) is infinite. In a finite time span we will only be able to generate a finite number of ensemble representations of this underlying time series. Hence we are in essence only *estimating* the time series. The quality of the estimate could be stated in terms of the number of ensembles that are needed to produce a bispectral estimate that is arbitrarily close to the true bispectrum. Since this number could vary depending on

the estimator, this provides a means to perform comparison tests between bispectral estimators. The second difference is that while every time series has only one bispectrum, there are, in general, many different time series that all have the same bispectrum. This is completely analogous to the fact that different time series (and in general an infinity of time series) have the same autocorrelation function. This second difference is not of great concern here since all that is needed is *one* time series. The fact that there may be more is of no consequence.

It should also be noted that the results of this chapter are valid for continuous time without any further modification. The problems arise when one attempts to handle discrete frequencies, as will be seen in chapter four.

3.1 Necessary and Sufficient Conditions for Invertibility

To invert the bispectrum we will use the simplest class of models that are capable of reproducing the bispectrum. To be capable of reproducing an arbitrary bispectrum the model must satisfy three general requirements.

1) It must have a non-zero bicovariance.

2) The resulting expression for the bicovariance of the model in terms of the g–kernels must be a *function*, which we will call the *bicovariance function* . (The domain of this function is the kernel sequence, and the range is the bicovariance or third order cumulant sequence.)

3) The domain of the kernel sequence must be the same as the domain of the bicovariance.

The first one is obvious and included only for completeness. The second condition guarantees that the kernels will be mapped to only one bicovariance. Essentially it eliminates the possibility that two different bicovariances will both have the same g kernels, and ensures that it will be possible to express the model kernels in terms of a unique bicovariance. This requirement might eliminate some AR processes. Note, however, that the inverse relation need not be a function, because it is permissible to have different kernel sequences produce the same bicovariance; the significance of this will be apparent later when we attempt to invert the bicovariance function. The third condition guarantees the ability to reproduce *any* arbitrary bispectrum. It removes any constraints on the bispectrum or bicovariance. Without this condition it would be possible to change the bicovariance without changing the g kernels. There are many models that could be constructed which fit these general requirements. The problem is to find a model which has a bicovariance function which is easily invertible. The quadratically nonlinear, infinite order moving average (MA) model of the universal bispectral class is ideally suited to this problem. Since the bispectrum can be interpreted as a measure of quadratic coupling, it is no surprise that the model best suited to the inversion should itself be quadratically nonlinear. The model, introduced in chapter one as the universal bispectrum model, is repeated here in (3.1) for convenience.

$$x(t) = \eta(t) + \varepsilon(t) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+r)\eta(t+s) \tag{3.1}$$

where $\varepsilon(t)$ and $\eta(t)$ are Gaussian independent identically distributed (i.i.d.) random variables with zero mean and variance σ^2 or $N(0,\sigma^2)$. This model is unique in that

the bicovariance function is a finite linear relation consisting of the the sum of six kernel terms. Models outside the universal bispectral class possess more complicated bicovariance functions which are very difficult, if not impossible, to invert.

3.2 Inversion in the Time Domain

There are two equivalent methods of inverting the bispectrum. The first is in the time domain. The three main steps in this method are as follows.

1) Solve for the bicovariance of the model using the definition of the bicovariance.

2) Invert the resulting expression to obtain a new expression for the kernels in terms of the bicovariance.

3) Compute the g-kernels from the relation found in 2) thus specifying the time series by means of (3.1).

3.2.1 The Bicovariance Function of the Universal Bispectral Model

The first step in obtaining the bicovariance is to substitute (3.1) into the expression for the third order cumulant sequence (or the moment since they are equal):

$$c_3(\tau_1,\tau_2) = \mathrm{E}[\, X(t)X(t+\tau_1)X(t+\tau_2)] \quad . \tag{3.2}$$

Doing this we obtain:

$$c_3(\tau_1,\tau_2) = E\{\varepsilon(t) + \eta(t) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+r)\eta(t+s)$$

$$\times \quad \varepsilon(t+\tau_1) + \eta(t+\tau_1) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_1+r)\eta(t+\tau_1+s)$$

$$\times \quad \varepsilon(t+\tau_2) + \eta(t+\tau_2) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_2+r)\eta(t+\tau_2+s)\ \} \cdot \quad (3.3)$$

Expanding this product yields a sum of 27 terms. Invoking the fact that $\varepsilon(t)$ and $\eta(t)$ are independent and zero mean, and the fact that odd order moments of Gaussian random variables are zero, reduces this expansion to only six non-zero terms. For example the first term of the expansion would be $E\left[\varepsilon(t)\ \varepsilon(t+\tau_1)\ \varepsilon(t+\tau_2)\right]$. This is a third order moment of a Gaussian random variable and therefore zero. The term containing the expression

$$E\left[\varepsilon(t+r)\eta(t+s)\varepsilon(t+\tau_1+r)\eta(t+\tau_1+s)\varepsilon(t+\tau_2+r)\eta(t+\tau_2+s)\right]$$

is also zero since the expectation can be factored into the product of two terms each of which is a third order moment of a Gaussian process. Writing only the six remaining terms gives for the bicovariance

$$c_3(\tau_1,\tau_2) = E[\ \varepsilon(t)\ \eta(t+\tau_1) \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_2+r)\eta(t+\tau_2+s)$$

$$+ \ \varepsilon(t)\ \eta(t+\tau_2) \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_1+r)\eta(t+\tau_1+s)$$

$$+ \eta(t)\,\varepsilon(t+\tau_1)\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_2+r)\eta(t+\tau_2+s)$$

$$+ \eta(t)\,\varepsilon(t+\tau_2)\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+\tau_1+r)\eta(t+\tau_1+s)$$

$$+ \eta(t+\tau_1)\,\varepsilon(t+\tau_2)\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+r)\eta(t+s)$$

$$+ \varepsilon(t+\tau_1)\eta(t+\tau_2)\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\varepsilon(t+r)\eta(t+s)] \quad . \tag{3.3}$$

Taking the expected value of this expression allows us to write (3.3) as

$$\begin{aligned} c_3(\tau_1,\tau_2) = \sigma^4 \Bigg\{ & \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(s+\tau_2-\tau_1)\delta(r+\tau_2) \\ & + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(s+\tau_1-\tau_2)\delta(r+\tau_1) \\ & + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(r+\tau_1-\tau_2)\delta(s+\tau_1) \\ & + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(r+\tau_2-\tau_1)\delta(s+\tau_2) \\ & + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(r-\tau_1)\delta(s-\tau_2) \\ & + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s)\delta(r-\tau_2)\delta(s-\tau_1) \Bigg\} \end{aligned} \tag{3.4}$$

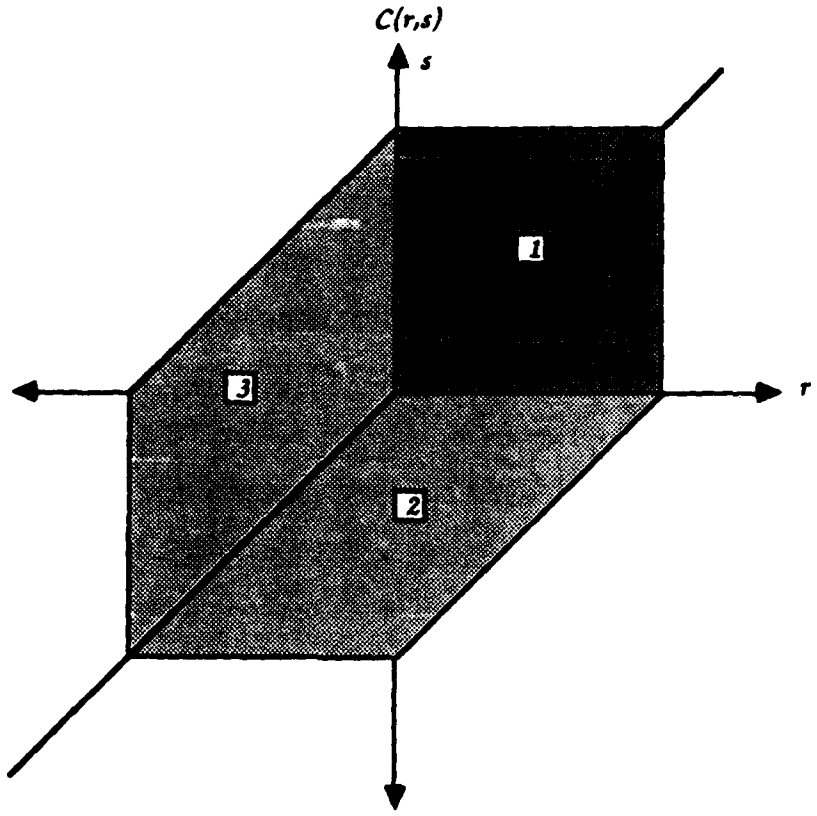
where $\delta(\cdot)$ is the Dirac delta function. The delta function collapses the double summation to just a single term allowing one to write (3.4) as the linear finite relation

$$c_3(\tau_1,\tau_2) = g(\tau_1,\tau_2) + g(\tau_2,\tau_1) + g(-\tau_1,\tau_2-\tau_1) + g(\tau_2-\tau_1,-\tau_1)$$

$$+ g(-\tau_2,\tau_1-\tau_2) + g(\tau_1-\tau_2,-\tau_2) \quad . \tag{3.5}$$

This is a very simple, pleasing result, but it can be simplified even more. Notice that this expression (but not the kernel function itself) automatically possesses the symmetry necessary for a third order moment of a real time series since the terms on the right side are exactly the same as the symmetries of the third order moment shown in fig 2.1. Thus substituting into (3.5) the symmetry relations for a third order moment will produce a system of six equations each containing six kernel terms, but in each case the right side will remain unchanged as expected. But the symmetry also has the less desirable effect of making the system indeterminate; another method of inversion must be used. One simplification comes from noting that for certain values of τ_1 and τ_2 not all six g terms are present. Inside any of the three regions in fig. 3.1 only two terms from the right side of (3.5) exist. On the boundaries all the terms from the adjacent regions exist. Thus at the origin $\{(r,s): r=s=0\}$ all six terms contribute, but in this case (3.3) reduces to $c_3(0,0) = 6g(0,0)$. On the other boundaries only four terms contribute. Two terms are excluded because the values, for which the delta function in (3.2) is non-zero, are not in the range of summation causing the whole summation to vanish. Exactly which terms contribute depends on which boundaries are being considered. Off the origin and boundaries only two terms contribute to the bicovariance. The other four do not contribute for the reason given above. The exact relation between the bicovariance and the kernel function is given in (3.6).

Fig. 3.1 The Contribution of the Kernel Function to the Bicovariance Plane



1 .... *g(r,s) and g(s,r) contribute*

2 .... *g(r-s,-s) and g(-s,r-s) contribute*

3 .... *g(s-r,-r) and g(-r,s-r) contribute*

On the boundaries all terms from adjacent regions contribute.
Note that the kernel function itself is only defined in region 1.

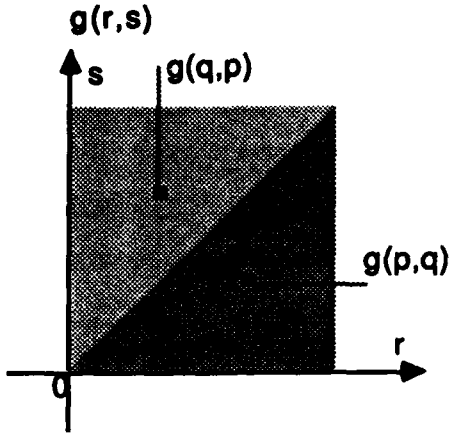
$$
\begin{aligned}
c_3(\tau_1,\tau_2) = {} & g(\tau_1,\tau_2) + g(\tau_2,\tau_1) && \tau_1 > 0,\ \tau_2 > 0 \\
& 2\,g(\tau_1,0) + g(0,\tau_1) && \tau_1 > 0,\ \tau_2 = 0 \\
& g(\tau_1-\tau_2,-\tau_2) + g(-\tau_2,\tau_1-\tau_2) && \tau_1 > \tau_2,\ \tau_2 < 0 \\
& 2\,g(-\tau_1,0) + g(0,-\tau_1) && \tau_1 < 0,\ \tau_2 = \tau_1 \\
& g(\tau_2-\tau_1,-\tau_1) + g(-\tau_1,\tau_2-\tau_1) && \tau_2 > \tau_1,\ \tau_1 < 0 \\
& 2\,g(\tau_2,0) + g(0,\tau_2) && \tau_2 > 0,\ \tau_1 = 0 \\
& 6\,g(0,0) && \tau_1 = 0,\ \tau_2 = 0 \quad . \qquad (3.6)
\end{aligned}
$$

This simplification is the result of causality and therefore could also have been obtained from (3.5) directly by noting that $g(r,s) = 0$ when $r < 0$, or $s < 0$.

3.2.2 Inversion of the Bicovariance Function

The next step in the inversion of the bispectrum is to invert (3.6) and express the kernels in terms of the bicovariance. However the inverse of this equation is not a function. There are an infinity of kernel functions that will satisfy (3.6). This arises from the fact that in order to uniquely specify the bicovariance we only need to know the values of $C_3(r,s)$ in the region $\{(r,s): s > 0, r > s\}$). For any point in this region, however, there are two kernel terms that contribute as can be seen in fig. 3.1. This amounts to an extra degree of freedom. The extra degree of freedom is actually not a problem, but rather it is an advantage. We are free to specify any relation between g(r,s) and g(s,r) that we desire. (Note that we are not talking about the specific term g(r,s), but rather how the weights in the region $\{(r,s): s > 0, r > s\}$ are related to the weights in the region $\{(r,s): r > 0, s > r\}$ as shown in fig. 3.2).

Fig. 3.2 Relation Between the Kernels



Since the sum of g(p,q) and g(q,p) contributes to c(p,q), the values of g(p,q) and g(q,p) are arbitrary as long as the sum is the same.

The bicovariance in any region is dependant on only the sum of two kernel terms. This means the value of the individual terms can vary without affecting the bicovariance as long as the sum remains constant. The extra degree of freedom is one reason this particular model was used instead of the one suggested by Wolinsky [1988]. These two degrees of freedom give one the ability to generate multiple time series which have identical bicovariances, but whose other moments (and hence polyspectra) are in general different. This is because the other moments will in general be dependent on the value of individual kernel terms and not just on the sum of two terms. This will be explored briefly when we look at the autocorrelation of the model. For the purpose of reproducing the bispectrum, let $g(r,s) = g(s,r)$. This provides the simplest possible relation for (3.6). We may now write:

$$g(\tau_1,\tau_2) = \begin{cases} \frac{1}{2\sigma^4} c_3(\tau_1,\tau_2) & \tau_1 > 0,\ \tau_2 > 0 \\ \frac{1}{4\sigma^4} c_3(\tau_1,0) & \tau_1 > 0,\ \tau_2 = 0 \\ \frac{1}{6\sigma^4} c_3(0,0) & \tau_1 = 0,\ \tau_2 = 0 \end{cases} . \qquad (3.7)$$

The rest of (3.6) does not need to be inverted. The other expressions only duplicate the information given by (3.7). This should be no surprise since the weights are zero outside of the domain defined in (3.7).

Given a bicovariance one could at this point compute the kernels from (3.7) and then compute a time series using (3.1) that had the given bicovariance. As it turns out, this is a rather slow way to compute the time series. It is much faster to compute the Fourier transform of the time series and then take the inverse transform. Since the goal is to compute a time series given its bispectrum, it would also be nice to stay entirely in the frequency domain and be able to go directly from the bispectrum to the Fourier transform of the time series without the need to inverse transform continuous frequency quantities into the time domain. Having to relate the model parameters to the bicovariance in the time domain causes approximations to enter the picture. The approximations arise from representing the bicovariance, which is the inverse transform of a continuous function, by the inverse discrete Fourier transform. Thus staying in the frequency domain would in effect allow one to know the Fourier transform of the time series exactly. Thus one would only have to resort to an approximation when computing the time series from its transform. But it turns out that this trouble spot is not evaded so easily. The reason is that one cannot exploit

the effect of causality in the frequency domain. Thus there is no way to invert what would be the Fourier transform of the bicovariance function except in the non-causal case where the kernels equal the bicovariance scaled by a factor of six. However, it is still informative to look at the inversion process in the frequency domain because, having found the kernels in the time domain, they can then be transformed to the frequency domain where the computation of the time series is more easily accomplished. But before doing this we will look briefly at the autocorrelation of the model.

3.2.3 Auto-correlation of the Universal Bispectral Model

The procedure for computing the autocorrelation of the model is entirely analogous to what was done earlier. Using the definition $C_2(\tau) \equiv E[\, x(t)x(t+\tau)]$ gives for the second order cumulant sequence:

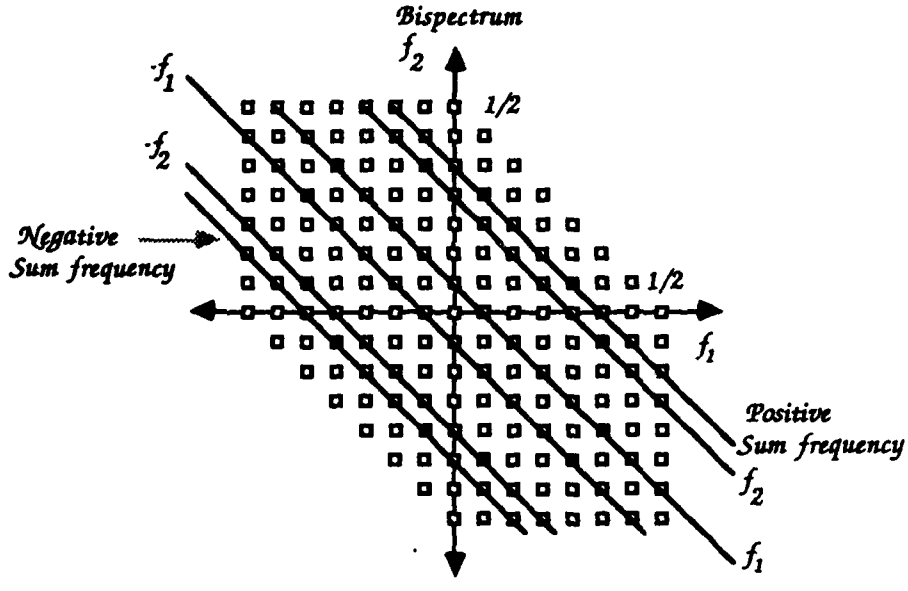
$$c_2(\tau) = 2\sigma^2 \delta(\tau) + \sigma^4 \sum_{r,s=\max\{\tau,0\}}^{\infty} \sum^{\infty} g(r,s)g(r-\tau,s-\tau) \quad . \tag{3.8}$$

It is not obvious that (3.8) has the necessary symmetry, but it does. The important thing is that this equation is sensitive to the relation between g(r,s) and g(s,r). Thus, as claimed, it is possible to produce different autocorrelation functions without changing the bicovariance simply by changing the relation between g(r,s) and g(s,r). However, it is certainly not a trivial problem to determine the relation necessary to produce a desired autocorrelation function. For completeness the power spectrum is given below.

$$P(f) = 2\sigma^2 + \sigma^4 \int_{-1/2}^{1/2} \left| G(f_1, f-f_1) \right|^2 df_1$$

$G(f_1, f_2)$ is the two-dimensional Fourier transform of the kernels. The limitation on arbitrarily specifying the power spectrum stems from the fact that once the bispectrum is picked to be nonzero at a certain point, say (f_1,f_2), the power spectrum cannot be set to zero at f_1, f_2 or $f_1+ f_2$. In the non-causal model the transform of the weights is simply the bispectrum scaled by a factor of six. This will be shown in the next section. But, as will be shown, even in the causal case the modulus of $G(f_1, f_2)$ bears some resemblance to the modulus of the bispectrum. Thus it is an interesting sidelight to note how the power spectrum is related to the bispectrum in the *non-causal case.* This is shown in fig. 3.3. In the figure the diagonal lines represent the path of the integral for specific values of f. The bispectral symmetry forces the power spectrum to be symmetric about the origin as it should be.

Fig. 3.3 Contribution of Bispectrum to the Power Spectrum



3.3 Inversion in the Frequency Domain

The first item on the agenda is to find the Fourier transform of the model. We first substitute into (3.1) the Fourier integral for ε(t+r) and η(t+r). This gives for x(t)

$$x(t) = \eta(t) + \varepsilon(t) + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s) \int_{-1/2}^{1/2} \varepsilon(f_1) e^{j2\pi f_1(r+t)}\, df_1 \int_{-1/2}^{1/2} \eta(f_2) e^{j2\pi f_2(s+t)}\, df_2 \quad . (3.9)$$

Interchanging the order of integration and summation and rearranging the terms gives

$$x(t) = \eta(t) + \varepsilon(t) + \int\int_{-1/2}^{1/2} df_1 df_2\, \varepsilon(f_1)\eta(f_2) e^{j2\pi(f_1+f_2)t} \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} g(r,s) e^{j2\pi(f_1 r + f_2 s)}. \quad (3.10)$$

Since the kernels are zero for r or s < 0, the summation on the right constitutes the conjugate of the Fourier transform of g(r,s) or $G^*(f_1,f_1)$. Taking the Fourier transform of both sides of (3.10), and bringing the Fourier transform operator through the integrals gives

$$X(f) = \eta(f) + \varepsilon(f) + \int\int_{-1/2}^{1/2} df_1 df_2\, \varepsilon(f_1)\eta(f_2)\, G^*(f_1 f_2)\, \Im\left\{e^{j2\pi(f_1+f_2)t}\right\} \quad (3.11)$$

$$\text{where} \quad G(f_1 f_2) = \sum_{r=-\infty}^{\infty}\sum_{s=-\infty}^{\infty} g(r,s) e^{-j2\pi(f_1 r + f_2 s)} \quad .$$

The transform of the exponential is simply the delta function. The delta function allows evaluation of one integral, reducing the double integral to a single integral and allowing us to write (3.11) as

$$X(f) = \eta(f) + \varepsilon(f) + \int_{-1/2}^{1/2} df_1\, \varepsilon(f_1)\eta(f - f_1)\, G^*(f_1 f - f_1) \quad . \quad (3.12)$$

At this point we can substitute (3.12) into (2.18). This gives:

$$B(f_1,f_2) = E\{\,[\eta(f_1)+\varepsilon(f_1)+\int_{-\frac{1}{2}}^{\frac{1}{2}} df\ \varepsilon(f)\eta(f_1-f)\,G^*(f,f_1-f)]$$

$$\times\ [\eta(f_2)+\varepsilon(f_2)+\int_{-\frac{1}{2}}^{\frac{1}{2}} df\ \varepsilon(f)\eta(f_2-f)\,G^*(f,f_2-f)]$$

$$\times\ [\eta(f_1+f_2)+\varepsilon(f_1+f_2)+\int_{-\frac{1}{2}}^{\frac{1}{2}} df\ \varepsilon(f)\eta(f_1+f_2-f)\,G^*(f,f_1+f_2-f)]\,\}\ . \quad (3.13)$$

What follows now is similar to what was done in the time domain except that here we are computing the bispectrum directly in terms of the two-dimensional Fourier transform of the kernels instead of the bicovariance in terms of the kernels. There are 27 terms in the expansion, and again only six remain after taking the expected value. Note that $E[\varepsilon(f_1)\varepsilon(f_2)] = \sigma^2\delta(f_1+f_2)$ allows us to write for the bispectrum

$$B(f_1,f_2) = \sigma^4\int_{-\frac{1}{2}}^{\frac{1}{2}} df\,\Big[G(f,f_1+f_2-f)\delta(f-f_1) + G(f,f_1+f_2-f)\delta(f-f_2)$$

$$+\ G^*(f,f_2-f)\delta(f+f_1) + G^*(f,f_1-f)\delta(f+f_2)$$

$$+\ G^*(f,f_2-f)\delta(f+f_1) + G^*(f,f_1-f)\delta(f+f_2)\,\Big]\ . \quad (3.14)$$

The Dirac functions allow us to evaluate the integral; thus we can write the bispectrum in terms of the transform of the kernels as

$$B(f_1,f_2) = \sigma^4\Big\{G(f_1,f_2) + G(f_2,f_1) + G^*(-f_2,f_1+f_2) + G^*(-f_1,f_1+f_2)$$

$$+ G^*(f_1+f_2,-f_2)+ G^*(f_1+f_2,-f_1) \Big\} . \qquad (3.15)$$

This relation is very similar to (3.5) in form, and in fact is simply the term by term two-dimensional Fourier transform of (3.5). This expression cannot be inverted as easily as (3.5) was, because the causality of the kernels cannot be exploited as it was in the time domain expression. However, using the knowledge gained by the inversion in the time domain, one can obtain an expression for $G(f_1,f_2)$ in terms of $B(f_1,f_2)$. While this result will not be very useful from a computational aspect, it is helpful in providing an intuitive understanding about the relation between the kernels and the bispectrum. We can rewrite (3.7) as

$$g(r,s) = \frac{1}{4} c(r,s)\delta(r)U(s) + \frac{1}{4} c(r,s)\delta(s)U(r) + \frac{1}{2} c(r,s)U(s)U(r)$$

$$- \frac{5}{6} c(r,s)\delta(s)\delta(r) \qquad (3.16)$$

where $\delta(.)$ is the Kronecker delta function and $U(.)$ is the discrete equivalent of the Heavyside step function. Taking the two-dimensional Fourier transform of both sides gives $G(f_1,f_2)$ on the left side and some function of the bispectrum on the right side. The step and delta functions by which the bicovariance is multiplied can be regarded as separable two-dimensional functions. Thus their two-dimensional Fourier transform is simply the product of each of the one-dimensional Fourier transforms. $G(f_1,f_2)$ can now be easily written as :

$$G(f_1,f_2)=\frac{1}{4}B(f_1,f_2)*\left\{\delta(f_1)+\frac{1}{j2\pi f_1}\right\}+\frac{1}{4}B(f_1,f_2)*\left\{\delta(f_1)+\frac{1}{j2\pi f_1}\right\}$$

$$+\frac{1}{2}B(f_1,f_2)*\left\{\delta(f_1)\delta(f_2)+\frac{\delta(f_2)}{j2\pi f_1}+\frac{\delta(f_1)}{j2\pi f_2}-\frac{1}{(2\pi)^2 f_1 f_2}\right\}-\frac{5}{6}B(f_1,f_2)*1 \quad (3.17)$$

where the asterisk denotes a two-dimensional convolution. One can see why this would be a difficult relation to use for computation, but it does tell us that the transform of the kernels are a "smeared" version of the bispectrum. They bear some resemblance to the bispectrum. It is worth mentioning that in the non-causal case alluded to earlier, the transform of the kernels is simply the bispectrum scaled by a constant.

This chapter has described the mathematics involved in the inversion of the bispectrum in the case of discrete time domain quantities that are infinite in duration and frequency quantities that are continuous in nature. Of course, in order to actually implement the methods set forth in this chapter one must deal with finite time sequences and discrete frequency functions. This means that the bispectrum will be approximated by samples from a finite number of points, and the bicovariance will be found by applying the inverse discrete Fourier transform to this approximation. The implications and restrictions inherent in this approximation, as well as ways to overcome them, will be the subject of the next chapter.

CHAPTER 4

IMPLEMENTATION: METHODS AND RESULTS

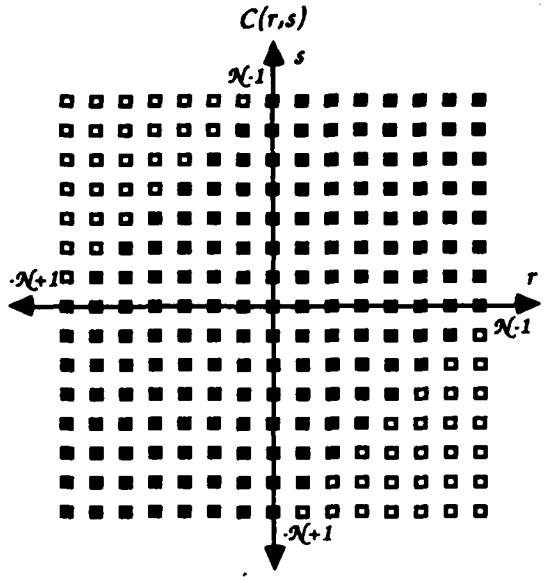
The purpose of this chapter is twofold. The first is to discuss the subtleties involved in the implementation of the inversion methods derived in the previous chapter. These subtleties revolve around the fact that we are discretizing a continuous quantity (frequency) in order to compute transforms via the DFT. While to some such things are mere trivialities, they nevertheless are necessary if one is to obtain correct results. The second purpose for this chapter is to give the step by step results of an inversion of the bispectrum.

4.1 Obtaining an Unaliased Bicovariance

Up to this point we have assumed the bispectrum to be a continuous, periodic function. Its inverse transform is found by an integration over the fundamental domain. We now want to sample this continuous function and use the inverse discrete Fourier transform (DFT) to compute the bicovariance. Since we presume to know the continuous bispectrum we are free to sample with any frequency resolution we desire. The resolution that is used will place a constraint on the minimum length of the sequence, but otherwise does not present any fundamental problems. The fundamental difficulty is in the sampling itself. It causes the bicovariance to be represented as a finite two-dimensional sequence. This finiteness raises an immediate problem. Since the bicovariance is in general defined in two dimensions over the

interval $(-\infty, \infty)$, the finite sequence obtained from the inverse DFT of the sampled bispectrum is at best an aliased version of the bicovariance and not the quantity with which the kernels are defined. For definiteness, suppose the bispectrum is sampled to form an [2N-1 x 2N-1] sequence, thereby constraining the bicovariance to be an [2N-1 x 2N-1] sequence defined over lags in the interval [-N+1, N-1]. For any arbitrarily specified bispectrum there is no *a priori* method to determine in advance if the bicovariance at lags greater than $\pm(N-1)$ is relatively small or not. If it is small, the error due to aliasing may be negligible. But generally one cannot assume this. The only way to determine aliasing is to simply compute the inverse transform of the bispectrum. By looking at the bicovariance outside the region of support one can determine whether it is aliased or not. The values outside the region of support for a finite record but within the [2N-1 x 2N-1] square will be zero if the bicovariance is unaliased. These values are represented by the empty squares in fig. 4.1. This is because the bicovariance in these regions is equivalent to the bicovariance in regions where at least one of the lags is greater than $\pm(N-1)$. Thus a nonzero value in this region implies that the bicovariance extends farther than the space allotted to it by the DFT. One can easily verify this by choosing a point in the region containing empty squares in fig. 4.1 and applying the symmetry relations of the bicovariance to it.

Fig. 4.1 Zero and Non-Zero Regions of a Finite Bicovariance



■ . . . *Point where bicovariance is nonzero*

□ . . . *Point where unaliased bicovariance is zero; but an aliased bicovariance is nonzero*

When estimating in the forward direction, *i.e.*, time series to spectra, a time domain window is used to truncate the bicovariance in a non-abrupt fashion. The assumption is that the bicovariance is negligible for lags beyond a certain value. This is essentially what is needed here. But, while applying such a window is a relatively simple process in the time domain, it is quite the opposite in the frequency domain. There it involves a convolution with the transform of the window. But that is the least of the problems. A bigger concern is that one does not know the duration or shape of the bicovariance. Thus it is impossible to apply any sensible criteria in

picking a suitable truncation boundary. One cannot determine the lags beyond which the bicovariance is negligible. The result is that although a window is needed, one does not have any way of knowing or sensibly choosing what window parameters to use; specifically, how long should the window be to include all the significant points in the bicovariance. However, there is another way to look at this problem. Let us leave the windowing approach for the moment and pursue the problem from a different perspective.

Another method of obtaining the unaliased bicovariance is to find a bispectrum which has a finite bicovariance. This is not as hopeless as it may initially appear. Since the bicovariance can be arbitrarily defined over the interval $(-\infty, \infty)$, there is no reason why it could not be zero for lags greater than some L.

$$C(\tau_1,\tau_2) = \begin{cases} C(\tau_1,\tau_2) & -L \le \tau_1,\tau_2 \le L \\ 0 & \textit{elsewhere} \end{cases} \tag{4.1}$$

This places a restriction on the arbitrariness of the bispectra that can be reproduced. One can now consider only those bispectra which possess finite bicovariances. This may seem like a very severe restriction, but this exact assumption is commonly made when estimating the bispectrum from a time series. Nevertheless it is the price that must be paid for the use of the DFT. Fortunately this does not place severe practical limitations on what sorts of bispectra can be reproduced. It turns out that the bigger problem is not only finding such a bispectra, but also finding one whose magnitude characteristics are close to those of the desired bispectrum.

A word should be inserted here to avoid possible confusion when attempting to apply what has been said here to the analogous case of the power spectrum. Unless one maintains a clear distinction between the estimate of a quantity and that quantity itself, confusion may easily result. For instance, it is well known that the inverse Fourier transform of the power spectrum produces an aliased autocorrelation function. This statement is referring to the situation where one has an M point random time series belonging to a random process and possessing a coherence time much less than M (or at least half of M). Thus one assumes that the true correlation only possesses significant values for time lags in the interval $\{\tau: -M/2 \leq \tau \leq M/2\}$ thereby allowing the power spectrum to be estimated from an M point DFT. However the estimate of the autocorrelation sequence obtained from one M point ensemble is 2M-1 points long. This covariance estimate, unlike the true covariance, is of course not obtainable from an M point estimate of the power spectrum. On the other hand, if one assumes to know the *true* power spectrum representing the *true*, finite M point autocorrelation function, then one can obtain it from the M point inverse transform of the power spectrum. In the problem that is the subject of this thesis we assume to know the true bispectra; the problem is finding one which actually has a finite bicovariance.

Finding a bispectrum which has a finite bicovariance is conceptually simple, but not as easily implemented. In order to produce a finite bicovariance, one merely has to satisfy certain additional constraints in the bispectrum. These additional relations are found by simply setting the DFT expression for a particular value of the bicovariance equal to zero. For example, if N=32 then C(20,-20) is outside the non-zero region of an unaliased bicovariance. Setting this equal to zero gives

$$\sum_{f_1=0}^{N-1} \sum_{f_2=0}^{N-1} B(f_1,f_2)\, e^{j2\pi(2if_1 - 2if_2)} = 0 \quad .$$

In general there are $1/4(N^2-1)$ such points in the bicovariance. This gives the following linear system of equations:

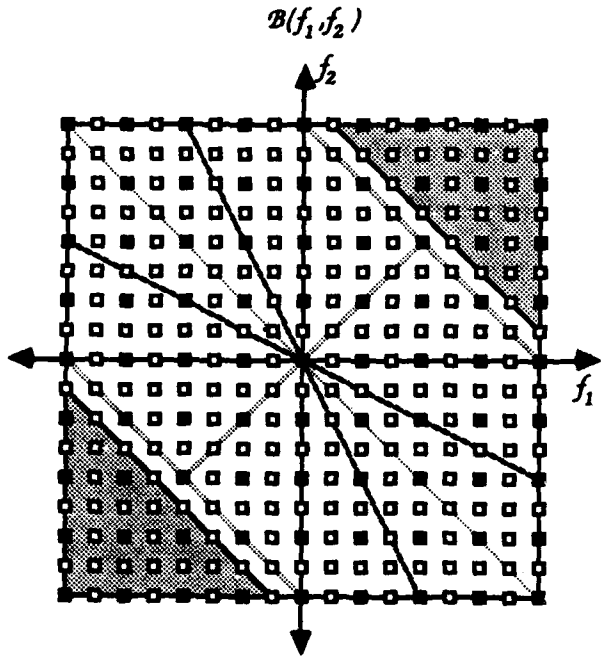
$$C(j,k)_i \equiv \sum_{f_1=0}^{N-1} \sum_{f_2=0}^{N-1} B(f_1,f_2)\, e^{j2\pi(if_1 + kf_2)} = 0 \qquad (4.2)$$

$$\text{where} \quad i \in \{1,\ldots 1/4(N^2-1)\} \quad .$$

Even for small N, solving this system of equations is no small task. It is further complicated by the fact that the system is over determined, forcing one to pick the other $3/4(N^2-1)$ bispectral points before solving the system. These have to be chosen in such a way that the resulting bispectrum has magnitude characteristics similar to the desired bispectrum. Since the equations bear close resemblance to the DFT and in fact differ only by a constant, one can solve them numerically by an iterative method using the DFT. This is accomplished by specifying the bispectrum over every other point in a two-dimensional fashion, taking the inverse DFT, and then setting to zero the proper points in the bicovariance. The modified bicovariance is then transformed back to the bispectrum where the initially specified points are reset and the other points are left alone. This process is continued until the desired degree of accuracy is reached. The iteration will converge as long as there are at least $1/4(N^2 - 1)$ points left unspecified in the bispectrum. In other words, all that is being done is to trade the choice of setting values in the bispectrum for the choice of setting an equal number of values in the bicovariance. In this way one can modify a

bispectrum to force it to possess a finite bicovariance. By spreading the specified points out in the fashion indicated above we are enabled to satisfy both the symmetry requirements of the bispectrum and the goal that the modulus of the modified

Fig. 4.2 Initial Specification of Bispectrum to Produce an Unaliased Bicovariance



□ ... *Bispectrum point initially left unspecified and allowed to assume any value.*

■ ... *Point set to a desired value and maintained at that value.*
Shaded regions indicate areas where the bispectrum is zero for bandlimited processes .

bispectrum have close agreement with the desired bispectrum. The end result of this iteration is simply the equivalent of windowing in the time domain. If one does not window, then there will be nonzero values in the part of the bicovariance shown by empty squares in fig. 4.1. Blindly proceeding with the inversion is equivalent to setting these values to zero. This of course changes the bispectrum in an unknown and hence uncontrollable way. Use of the iteration algorithm also changes the bispectrum, but it does so in a way that allows one to control what gets changed and what does not. Finally, it is worth noting that the surface integral of the modified bispectrum over the fundamental domain is unchanged. In other words the points added to the modified bispectrum contribute in such a way so as not to change the skewness of the time series.

4.2 The DFT of the Model

The purpose of this section is simply to modify the frequency domain equations derived in the last chapter so that they are written in terms of the DFT. Since the model is essentially a linear two dimensional convolution a certain amount of zero padding is necessary to break the circularity of the DFT.

The method used to find the DFT is identical to that used in section 3.3. Accordingly, in this section, we will only present the equations that are different. In order to obtain a sensible result, the DFT of eta, epsilon, x(t), and the kernel sequence must be evaluated at the same points in the frequency domain. Since the random variates, the time series, and the kernel sequence are all of different length, they must each be padded with zeroes to achieve a uniform length over which to compute the DFT. Let the kernel sequence be N x N. The time series must then be

at least of length 2N making the length of epsilon and eta 3N-1 points. Further zero padding may be necessary in order to efficiently use an FFT algorithm. Letting the longest length be n, the DFT of the model can be written as

$$\underline{X}(k) = \underline{\varepsilon}(k) + \underline{\eta}(k) + \frac{1}{n}\sum_{k_1=0}^{n} \underline{\varepsilon}(k)\,\underline{\eta}(k-k_1)\,\underline{G}^{*}(k,k-k_1) \quad . \tag{4.3}$$

The underline signifies a zero padded sequence. This will produce a time series of length n. Only the first 2N values are retained; the remainder are either zero or unwanted numbers.

4.3 A Demonstration of the Inversion of a Bispectrum with the UBM

The software to implement the inversion was written in Fortran 77 on the CYBER 830 at ARL:UT. It allows one to specify the desired bispectrum and the desired resolution. Current limitations of the CYBER restrict the resolution of the specified bispectrum to a complex array of 128 x 128. For convenience, let the frequency interval between points be 1 Hz . This choice is entirely arbitrary. Any other number could just as easily be used. This restriction implies that the maximum possible resolution in the bispectrum is obtained when the isosceles triangle in the principal domain has a base of 65 points and a height of 33. This in turn will produce 64^2 nonzero weights.

Another factor is the computation time. Each iteration of the specified bispectrum requires two 128-point two-dimensional Fourier transforms plus additional computations on a significant portion of the array. In the light of these

considerations, 50 iterations were performed on the specified bispectrum in the results presented below. This resulted in an error of less than 1 percent at the specified points of the bispectrum.

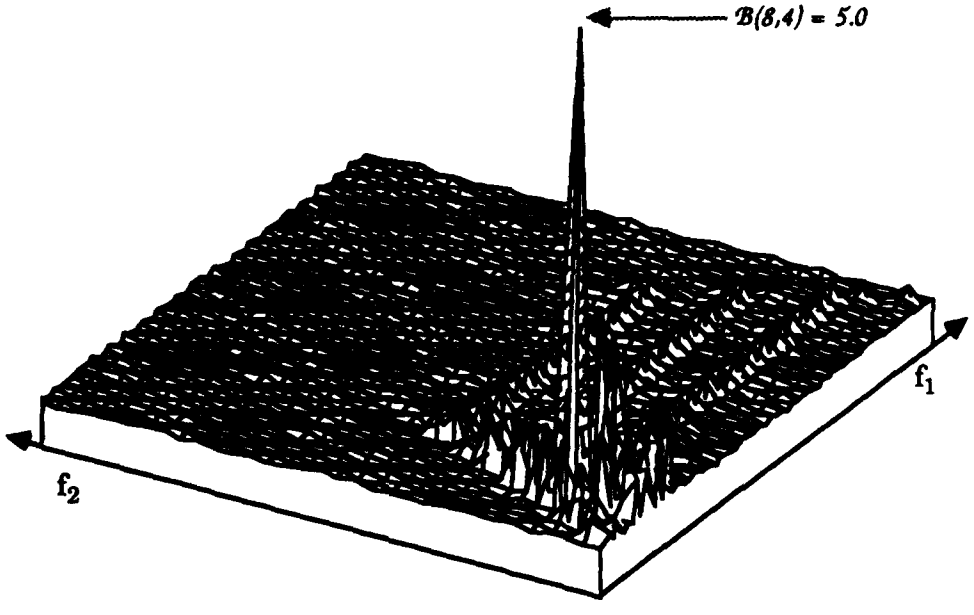
4.3.1 The Results

For this example the initial bispectrum was specified to be:

$$B(f_1, f_2) = 5\delta(f_1 - 8)\,\delta(f_2 - 4) \quad . \tag{4.4}$$

The maximum resolution was used. A plot of the resulting modified bispectrum is shown in fig. 4.3. This is the bispectrum that has a bicovariance which is zero for lags greater than 63. The kernel sequence was found from this bispectrum. Using the relation in (4.3), 60 128-point ensembles were computed. The first eight records of the resulting time series are shown in fig. 4.4. The estimated power spectrum and the actual power spectrum are shown in fig. 4.5 and fig. 4.6. The actual power spectrum was computed from the relation given in section 3.2.3. As the overlay plot in fig. 4.7 shows, there is very good agreement between the estimated and the actual power spectrum. Since the power spectrum is much easier to compute, it was used as an indicator of how many time series were necessary. Poor agreement here would be indicative of even worse agreement in the bispectrum. Fig. 4.8 shows an estimate of the bispectrum obtained using the Hinich estimator [Hinich, 1982]. While it is not helpful to overlay the initial bispectrum and this estimate, one can observe that they are in good general agreement. By raising the threshold of the plot one can verify the location of the big peak. Such a plot is shown in fig. 4.9. From this plot it is easily seen that the peak is in the proper location.

Fig. 4.3 The Desired Bispectrum



Resolution is 1 Hz per division.
The bispectrum computed
from Eq. 3.15

Fig. 4.4 The Generated Time Series

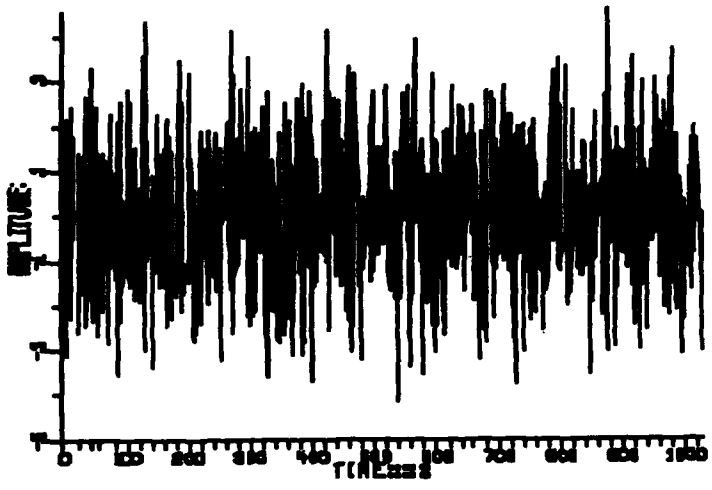


Fig. 4.5 The Estimated Power Spectrum

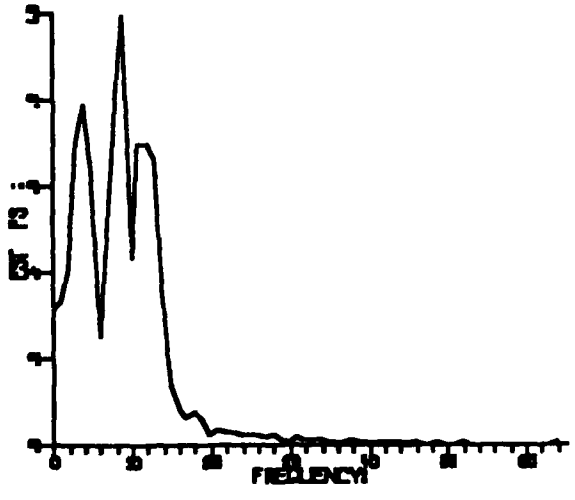


Fig. 4.6 The Predicted Power Spectrum

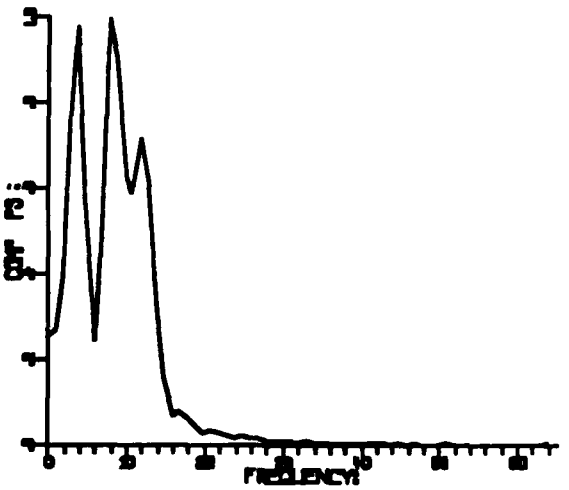


Fig. 4.7 Predicted & Estimated Power Spectrum

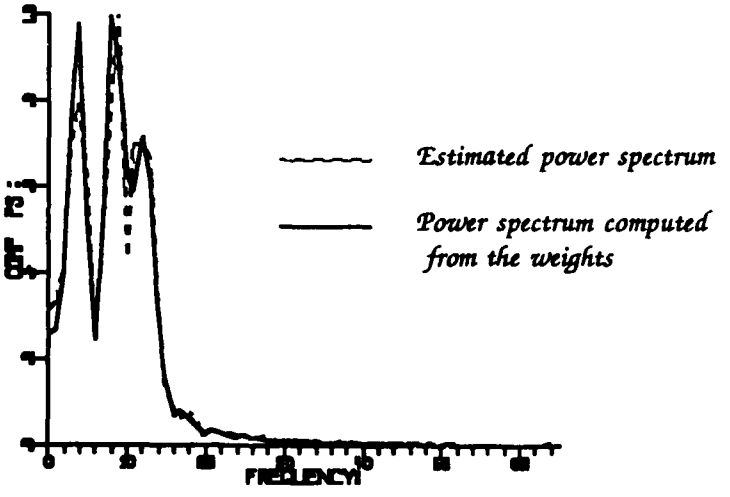
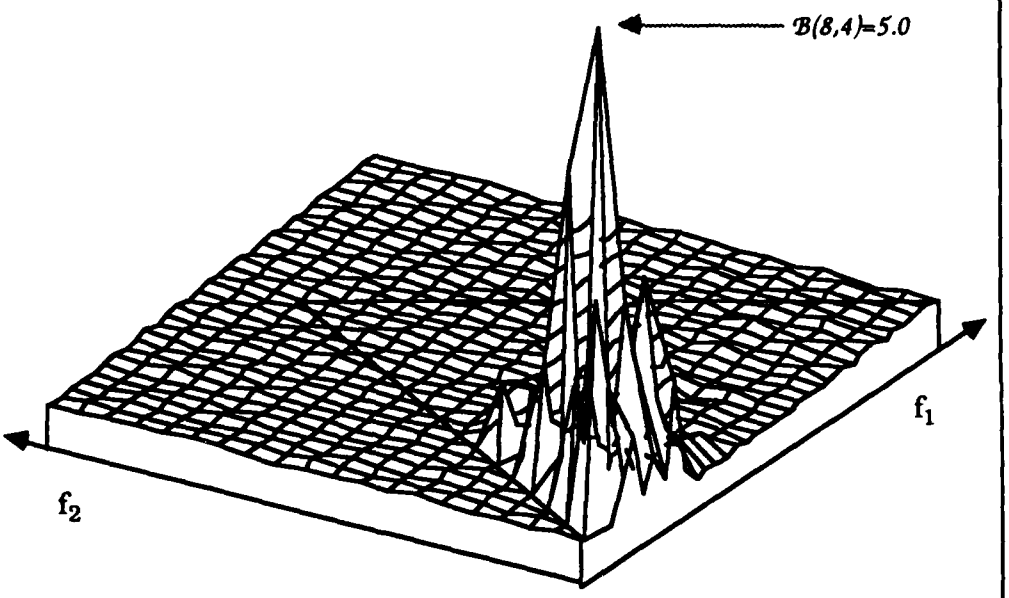
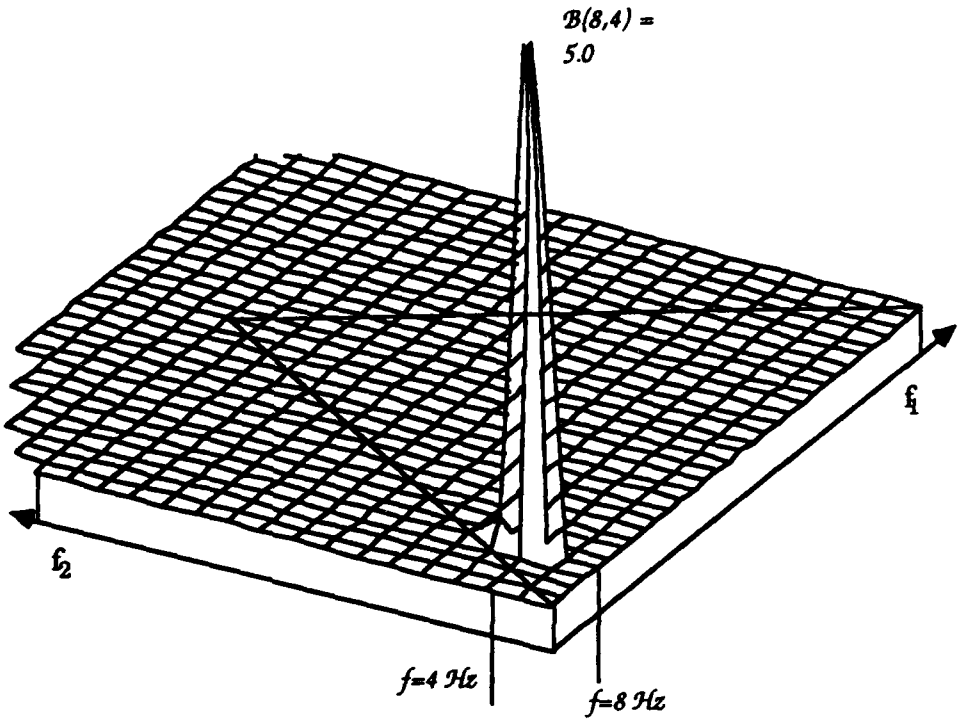


Fig. 4.8 Estimated Bispectrum



- *Resolution is 2 Hz per division.*
- *This estimate was found with the Hinich estimator.*

Fig. 4.9 Raised Threshold Plot of the Estimated Bispectrum



Resolution is 2 Hz per division.

Fig. 4.10 Power Spectrum with Asymmetric Kernel Relation

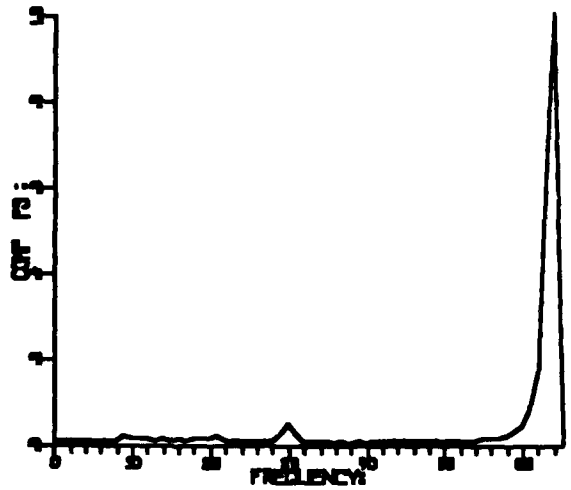


Fig. 4.11 Power Spectrum of Time Series with Symmetric Kernels

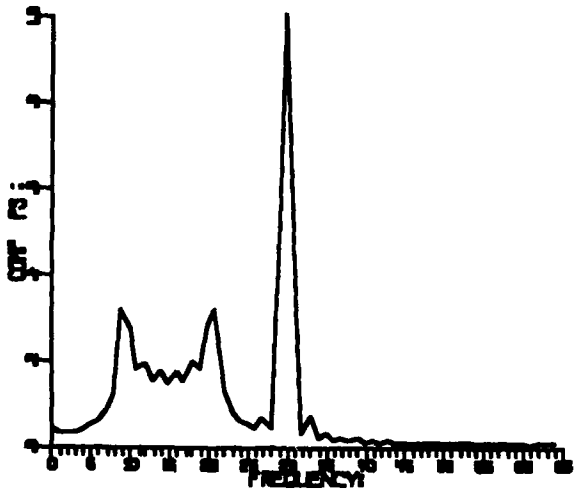
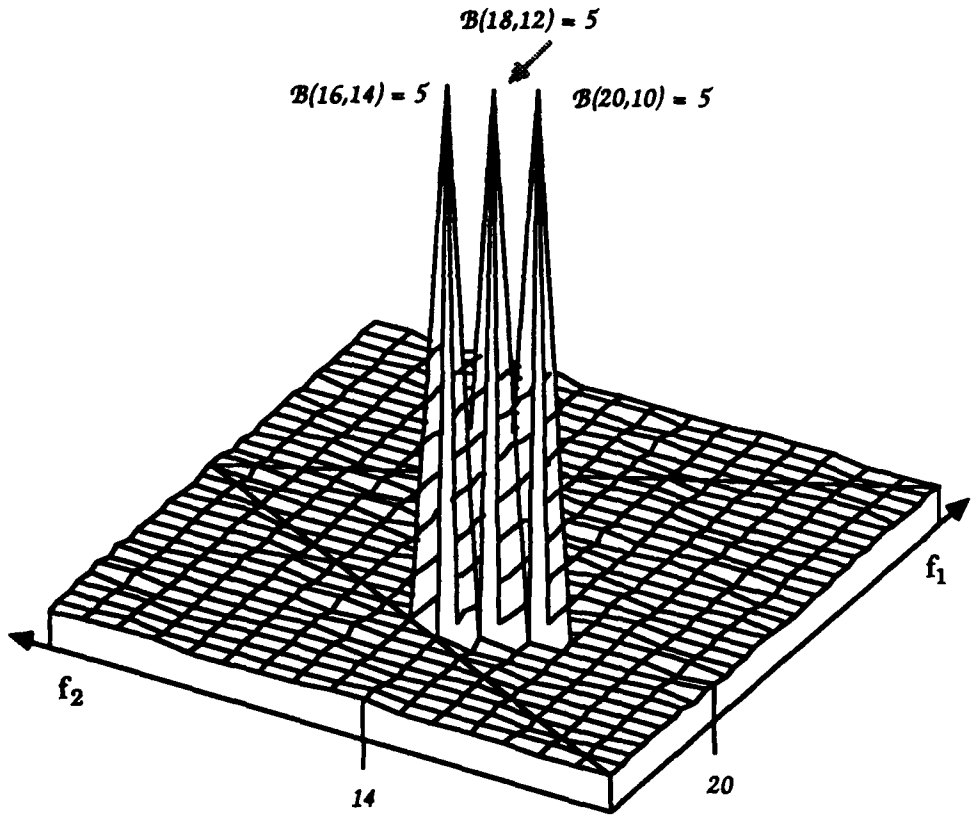


Fig. 4.12 Bispectrum of the Two Sample Time Series



Resolution is 2 Hz per division.

In section 3.2.3 it was shown that by altering the relation between the kernel functions (see fig. 3.2) different time series could be produced which had identical bispectra, but different power spectra. An example of this is shown in figs. 4.10 and 4.11. These power spectra were computed from two time series which have

identical bispectra. This bispectrum is shown in fig. 4.12. Because it is not possible to arbitrarily specify both the bispectrum and the power spectrum at the same time, it is not possible to significantly change the power spectrum. One cannot remove power at any of the three frequencies associated with a point in the bispectrum. Thus in essence all that has been done in the above example is to "bury" the peaks seen in the spectrum of fig. 4.11 by adding a very large peak to the spectrum of fig. 4.10.

CHAPTER 5

SUMMARY

This thesis has presented a method of producing or estimating a stochastic time series that has any desired bispectral characteristic. This involved the use of an infinite order moving average model whose bicovariance was obtainable as a finite linear relation of the kernels. This expression was then inverted to produce an expression giving the kernels in terms of the bicovariance. However, this inversion is not unique. In order to identify a unique kernel function, an additional constraint had to be invoked. This constraint was imposed in the form of a relation between the kernel function g(r,s) and g(s,r). The particular relation used was g(r,s) = g(s,r). This was an entirely arbitrary choice used only because of convenience. Any other choice could have been used. As was demonstrated, a different choice will not change the bicovariance, but it will in general change the other polyspectra. The effect on the power spectrum of a different kernel relation is shown at the conclusion of chapter 4.

Once the kernels have been found the problem is theoretically solved. Knowing the kernels allows one to compute the time series. However, when one attempts to implement this solution some other problems arise. These problems are the direct consequence of approximating infinite sequences with finite ones. In general the inverse transform of the bispectrum is an infinite sequence. Thus blindly inverse transforming the bispectrum will in general give one an aliased version of the bicovariance. This problem was circumvented by finding a bispectrum which has a

finite bicovariance. This simply amounted to giving up control over the value of some bispectral points in return for control over an equal number of points in the bicovariance. This was done in such a way so as to have minimal effect on the modulus of the desired bispectrum. It should be emphasized that if the bicovariance is finite it can be obtained from the inverse transform of the bispectrum, and in an analogous manner the unaliased autocorrelation function can be found from the power spectrum provided the number of points used to represent the power spectrum is equal to or greater than the length of the autocorrelation, i.e., $F \leq 1/T$ where F is the resolution of the power spectrum and T is the length of the autocorrelation. This may come as a surprise to one thinking in terms of estimating spectra from a time series. This is the result of confusing an estimate of the bicovariance (autocorrelation) with the true bicovariance (autocorrelation).

Having found the kernels it turned out to be easier to compute the transform of the weights and then compute the DFT of the time series. This is because the DFT involves a single integral (summation) over a product of the transform of the weights and the Gaussian variates as opposed to the double integral (summation) that is required in the time domain. Results of a sample inversion were shown in the case where the bispectrum consisted of a single peak. Other inversions were also done yielding equally good results. Generally speaking, increasing the variance increases the number of time series needed to reproduce the bispectrum. This is simply due to the fact that epsilon and eta are produced by pseudo-random number generators. The resulting time series has the proper bispectrum to the extent that the random variates approach the Gaussian, zero mean assumption. A higher variance simply means that

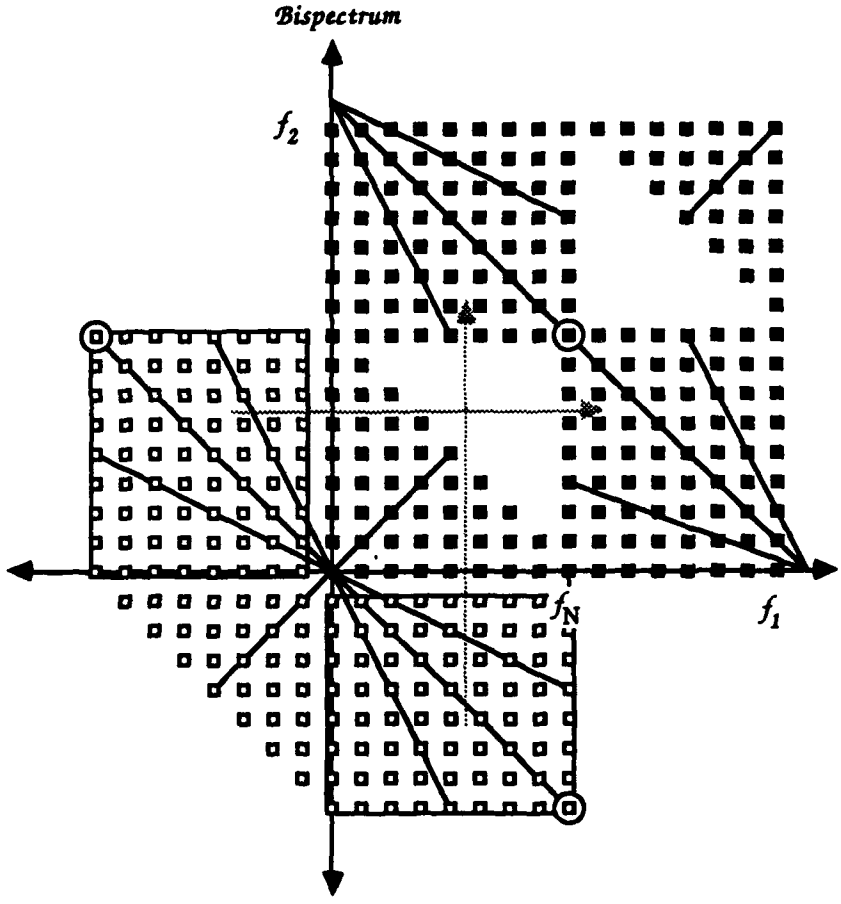
more random variate records are needed before the estimated third, fifth, and higher order moments are actually zero.

APPENDIX A

COMPUTING THE BICOVARIANCE WITH THE TWO-DIMENSIONAL FFT

Since the bicovariance is defined over the interval (-N, N) and the bispectrum is defined over frequencies [-1/2,1/2], they must both be appropriately shifted before being transformed with the DFT so that B(0,0) or C(0,0) sum without being phase shifted and not B(-1/2,-1/2) or C(-N,-N). The mechanics of this translation are shown in fig. A-1. By translating the bispectrum in this way, the negative frequency term $\{\exp(-j2\pi(k_1l_1+k_2l_2)/N)\}$ of the DFT is being replaced by the equivalent term $\{\exp(j2\pi((N-k_1)l_1+(N-k_2)l_2)/N)\}$. The inverse transform of this array is an analogously shifted version of the bicovariance. The mechanics of the translation of the bicovariance is nearly identical. The only difference is that the bicovariance must be padded two-dimensionally with a zero in order to make its dimensions even so that the FFT may be used.

Fig. A-1 Translation of Fundamental Domain of the Bispectrum



- *Areas with empty square dots translate to areas with black squares according to the direction of the arrows.*
- *The circled points become the same point in the translated bispectrum.*
- *The blank regions indicate places where the bandlimited bispectrum is zero.*

APPENDIX B

TIME SERIES GENERATION SOFTWARE

```
C   WRITTEN BY PETER ALLISON.

C   JANUARY 1988

C-------------------------------------------------------------------
C       THIS PROGRAM COMPUTES A TIME SERIES WITH A GIVEN BISPECTRAL
C       CHARACTERISTIC BY
C       1) COMPUTING THE WEIGHTS IN THE FREQ DOMAIN FROM THE GIVEN
C          BISPECTRUM
C       2) COMPUTING THE FT OF THE DESIRED SIGNAL IN THE FD (FROM THE
C          WEIGHTS)
C       3) INVERSE DF TRANSFORMING THE RESULT TO PRODUCE THE DESIRED
C          SIGNAL
C-------------------------------------------------------------------

    PROGRAM BTSFTN4

    INTEGER IRC,NREC,IWK(918),TEST,N2,M,K,L,LTS,NV
    INTEGER NRC,ERR

   REAL XT(128),XS(1024),XP(1024),ETA(200),GNU(200)
    REAL SIG,SCALE,MAX,MAXS,MINS,PSE(128),PST(512),RWK(918)
    REAL TIC

    DOUBLE PRECISION DSEED

    COMPLEX CX(128),CE(128),CN(128),SUM,C(128,128),CG(128,128)
    COMPLEX CXTS(128),CWK(128)

   OPEN(UNIT=9,FILE='WEIGHTC')

    DATA PSE/128*0./
    DATA MINS,MAXS,MAX/3*0./
    DATA C,CG/32768*(0.,0.)/
    DATA CXTS/128*(0.,0.)/
    DATA ERR/1/

    REWIND(9)

C-------------------------------------------------------------------
C       THERE IS AN OPTION TO EITHER READ THE BICOVARIANCE IN  FROM
C         A FILE  (WEIGHTC)  OR TO READ THE BISPECTRUM IN FROM
C         FILE DATA.
C-------------------------------------------------------------------

4   PRINT*,'INPUT 1 TO GENERATE THE MODEL WEIGHTS, INPUT 2 '
    PRINT*,'     TO READ THEM IN FROM FILE WEIGHTC'
    READ*,TEST
```

```
      PRINT*,' '
      LTS=128
      IF(TEST.EQ.1) THEN
2         PRINT*,'INPUT THE NUMBER OF WEIGHTS TO USE'
          PRINT*,'    (MAX 64)'
          READ*,N
          IF(N.GT.64) THEN
          PRINT*,' NUMBER IS GREATER THAN 64'
          GOTO 2
          ENDIF
          PRINT*,' '
          N2=2*N
          XCALL REWD
      ENDIF

      PRINT*,'INPUT THE NUMBER OF RECORDS TO GENERATE'
      READ*,NREC
      PRINT*,' '

      PRINT*,'INPUT SIGMA'
      READ*,SIG
      PRINT*,' '

      IF(TEST.EQ.2) THEN
          READ(9,*) N
          N2=2*N
          PRINT*,'ENTER 0 TO DISCARD EXISTING T.S.'
          READ*,NRC
          IF(NRC.EQ.0) CALL REWD
          CALL GENERG(C,CG,N2,SIG,RWK,IWK,CWK,TEST)
      ELSE
1         CALL BISPIN(C,CG,N2,SIG,IWK,RWK,CWK,TEST,ERR)
          IF(ERR.EQ.1) THEN
              PRINT*,'ENTER A BETTER VALUE FOR N'
              READ*,N
              PRINT*,'  '
              N2=2*N
              GOTO 1
          ENDIF
      ENDIF

      CLOSE(9)
      OPEN(UNIT=4,FILE='BTSOUT')
      OPEN(UNIT=5,FILE='TIMSER')
      OPEN(UNIT=6,FILE='TIMSER1')
      REWIND(4)
      WRITE(4,*) ' NUMBER OF WEIGHTS USED IS ',N
      WRITE(4,*) ' NUMBER OF RECORDS COMPUTED IS ',NREC
      WRITE(4,*) ' VARIANCE OF ETA AND EPSILON IS ',SIG**2

C-----------------------------------------------------------------
C  BISPECTRUM HAS BEEN LOADED INTO ARRAY C.
C  THE UNALIASED BICOVARIANCE HAS BEEN COMPUTED.
C  THE MODIFIED BISPECTRUM IS NOW IN ARRAY C.
```

```
C  FOURIER TRANSFORM OF WEIGHTS IS IN ARRAY CG
C  NOW GENERATE THE TIME SERIES IN THE FREQUENCY DOMAIN
C-------------------------------------------------------------

    PRINT*,'GENERATE RANDOM VARIATES'
    OPEN(UNIT=3,FILE='DSEED')
    REWIND(UNIT=3)
    READ(3,3) DSEED
3   FORMAT(D17.10)

    IF(N.LT.42) THEN
        FPAD=0
        NV=3*N-1
    ELSE
        FPAD=1
        NV=LTS-1
    ENDIF

C       LOOP 110 IS DONE ONCE FOR EACH RECORD GENERATED

    DO 110 IRC=1,NREC
        CALL GGNML(DSEED,NV,GNU)
        CALL GGNML(DSEED,NV,ETA)
        DO 120 I=1,NV
            GNU(I)=GNU(I)*SIG
            ETA(I)=ETA(I)*SIG
            CE(I)=CMPLX(ETA(I))
            CN(I)=CMPLX(GNU(I))
120     CONTINUE
        DO 130 I=NV+1,LTS
            CE(I)=CMPLX(0.)
            CN(I)=CMPLX(0.)
130     CONTINUE
        CALL FFT2C(CE,7,IWK)
        CALL FFT2C(CN,7,IWK)

C       THE DFT OF THE PADDED VARIATES IS COMPUTED IN 120 AND 130

        DO 140 K=0,LTS-1
            CX(K+1)=CMPLX(0)
            SUM=CMPLX(0)
            DO 150 L=0,LTS-1
                M=K-L
                IF(M.LT.0) M=LTS+M
                SUM=SUM+CE(L+1)*CN(M+1)*CONJG(CG(L+1,M+1))
150         CONTINUE
            CX(K+1)=SUM/LTS + CE(K+1) + CN(K+1)
            CXTS(K+1)=CXTS(K+1)+CX(K+1)
            CX(K+1)=CONJG(CX(K+1))
            CXTS(K+1)=CONJG(CXTS(K+1))
            PSE(K+1)=REAL(CX(K+1)*CONJG(CX(K+1)))/LTS+PSE(K+1)
140     CONTINUE
13      FORMAT('SUM(',I2,')=',4(E16.7))
```

```
C       TAKE THE INVERSE TRANSFORM OF THE TIME SERIES

        CALL FFT2C(CX,7,IWK)

        WRITE(4,*) '*** XT(I) **** CX(I) **** CX(I+N) *****'
        DO 160 I=1,LTS
            XT(I)=REAL(CX(I))/LTS
            IF(IRC.LT.9) XS((IRC-1)*LTS+I)=XT(I)
            IF(MAXS.LT.XT(I)) MAXS=XT(I)
            IF(MINS.GT.XT(I)) MINS=XT(I)
            WRITE(4,11) I,XT(I),CX(I),K
            IF(FPAD.EQ.0) THEN
                WRITE(6,9) XT(I)
                IF(I.GT.2*N) XT(I)=0.
            ELSE
                WRITE(6,9) XT(I)
                IF(I.GT.LTS-N) XT(I)=0.
            ENDIF
            WRITE(5,9) XT(I)
160     CONTINUE

        PRINT*,'FINISHED THE ',IRC ,'TH TIME SERIES'

110 CONTINUE
11   FORMAT(' XT(',I3,')=',F8.3,' CX(I)=',2(E12.4),' K = ',I3)
     CLOSE(5)
     CLOSE(6)
     REWIND(UNIT=3)
     WRITE(3,3) DSEED
     CLOSE(3)

C       COMPUTE THE BISPECTRUM

     CALL BISPGEN(CG,C,CXTS,LTS,SIG,NREC)
     CALL FFT2C(CXTS,7,IWK)
     PRINT*,'FINISHED COMPUTING THE BISPECTRUM'

     OPEN(UNIT=7,FILE='TIMSER2')
     DO 170 I=1,LTS
         CXTS(I)=CXTS(I)/(LTS*NREC)
         WRITE(7,9) REAL(CXTS(I))
170  CONTINUE
     CLOSE(7)


     PRINT*,' BEGIN POWER SPECTRUM COMPUTATION'
     WRITE(4,*) '********** POWER SPECTRUM ********************'
     DO 200 I=1,LTS/2+1
         PSE(I)=PSE(I)/NREC
         WRITE(4,8) I,PSE(I)
         IF(PSE(I).GT.MAX) MAX=PSE(I)
200  CONTINUE

     DO 210 I=1,1024
```

```
      XP(I)=I
210  CONTINUE

C       PLOT THE TIME SERIES AND THE BISPECTRUM

     CALL PLTLFN(L"PLOT1",L"1234567890",10000)
     CALL PLTORG(1.,1.)
      CALL PLTAXIS(0.,0.,5.,0.,0.,65.,2.,L"FREQUENCY",-10,5)
      CALL PLTAXIS(0.,0.,5.,90.,0.,1.,.2,L"EST  PS  ",10,1)
      CALL PLTDATA(XP,PSE,65,0,0,0.,65./5.,0.,MAX/5.,.08)
     CALL PLTEND(8.5,11.0)

     CALL POWSPEC(CG,SIG,PST,MAX)

     CALL PLTLFN(L"PLOT1",L"1234567890",10000)
     CALL PLTORG(1.,1.)
      CALL PLTAXIS(0.,0.,5.,0.,0.,65.,2.,L"FREQUENCY",-10,5)
      CALL PLTAXIS(0.,0.,5.,90.,0.,1.,.2,L"COMP  PS ",10,1)
      CALL PLTDATA(XP,PST,65,0,0,0.,65./5.,0.,MAX/5.,.08)
     CALL PLTEND(8.5,11.0)

     IF (NREC.GT.7) THEN
        PRINT*,'MAX IS ',MAXS,'  MIN IS ',MINS
        PRINT*,'INPUT STARTING VALUE FOR PLOT'
        READ*,MINS
        PRINT*,'INPUT THE TIC INTERVAL'
        READ*,TIC
        IF(MINS.GT.0)MINS=0.
        SCALE=(MAXS-MINS)/5
        CALL PLTLFN(L"PLOT1",L"1234567890",10000)
        CALL PLTORG(1.,1.)
        CALL PLTAXIS(0.,0.,5.,0.,0.,1024.,20.,L"TIME",-10,5)
        CALL PLTAXIS(0.,0.,5.,90.,MINS,MAXS,TIC,L"AMPLITUDE",10,2)
        CALL PLTDATA(XP,XS,1024,0,0,0.,1024/5.,MINS,SCALE,.08)
        CALL PLTEND(8.5,11.0)
     ENDIF

8    FORMAT(I3,2(E17.8))
9    FORMAT(E17.8)
    OPEN(UNIT=18,FILE='BISPDAT')
    REWIND(18)
     READ(18,*) NRC
     NREC=NRC+NREC
    REWIND(18)
     WRITE(18,*) NREC
     CLOSE(18)
     CALL BSDATFL(LTS,NREC)
     END
```

```
C  ******* SUBROUTINES *********************

      SUBROUTINE BISPIN(C,CG,N,SIG,IWK,RWK,CWK,TEST,ERR)

C-----------------------------------------------------------------------
C        THE BISPECTRUM IS INPUT FROM FILE DATA  THEN THE UNALIASED
C            BICOVARIANCE IS FOUND BY ITERATION.   THE BISPECTRUM,
C            THE TRANSFORM OF THE WEIGHTS, AND THE BICOVARIANCE
C            ALL ARE WRITTEN OUT TO FILES.
C-----------------------------------------------------------------------

      REAL CST(64),SIG,RWK(918)

      INTEGER L1(64),L2(64),F,N2,ERR,TEST,IWK(918)

      COMPLEX C(128,128),CG(128,128),B,CWK(128)

      OPEN(UNIT=1,FILE='FTOUT')
      OPEN(UNIT=2,FILE='DATA')
      OPEN(UNIT=15,FILE='PBISP')

      REWIND(UNIT=2)
      REWIND(UNIT=1)
      REWIND(15)

      WRITE(15,*) N/2+1
      WRITE(15,*) N/4+1
      N2=N/2

      PRINT*,'INPUT NUMBER OF NONZERO BISPECTRUM POINTS'
      READ(2,4) NP
4     FORMAT(I3)
      PRINT*,'NUMBER OF NONZERO BISP. POINTS IS ',NP
      F=N+1
      PRINT*,'INPUT BISPECTRUM FROM FILE  DATA '
      DO 5 I=1,NP
          READ(2,7)L1(I),L2(I),CST(I)
          PRINT*,'B(',L1(I),L2(I),') = ',CST(I)
          IF(L1(I).GT.N/2.OR.L2(I).GT.N/4.OR.L1(I)+L2(I).GT.N/2) THEN
              PRINT*,'POINT OUTSIDE THE PRIN. DOMAIN'
              ERR=1
              GOTO 99
          ENDIF
5     CONTINUE
7     FORMAT(I3,I2,F4.1)
      PRINT*,'INPUT # OF ITERATIONS'
      READ*,K
      WRITE(1,*) '************ K = ',K,'***************'

C         EACH LOOP 30  REPRESENTS ON ITERATION OF THE BISPECTRUM

      DO 30 M=1,K
```

```
      CALL NEXISTB(C,N)
      DO 15 I=1,NP
      B=CMPLX(CST(I))
      C(L1(I)+1,L2(I)+1)=B
      C(L2(I)+1,L1(I)+1)=B
      C(L1(I)+L2(I)+1,-L1(I)+F)=B
      IF (L2(I).NE.0) C(L1(I)+L2(I)+1,-L2(I)+F)=B
      IF (L2(I).NE.0) C(-L2(I)+F,-L1(I)+F)=B
      IF(L2(I).NE.0) C(-L1(I)+F,-L2(I)+F)=B
      C(-L1(I)-L2(I)+F,L2(I)+1)=B
      C(L2(I)+1,F-L1(I)-L2(I))=B
      C(L1(I)+1,F-L1(I)-L2(I))=B
      C(F-L1(I)-L2(I),L1(I)+1)=B
      IF(L2(I).NE.0) C(F-L2(I),L2(I)+L1(I)+1)=B
      C(F-L1(I),L1(I)+L2(I)+1)=B
15  CONTINUE
    CALL FFT3D(C,128,128,N,N,1,-1,IWK,RWK,CWK)
    IF(M.EQ.K) THEN
    DO 10 I=1,N2
    DO 10 J=1,N2
10      WRITE(1,1) I-1,J-1,C(I,J),M
    ENDIF

C       BEFORE TRANSFORMING SET TO 0 VALUES OF BICOV. THAT DONT EXIST.

    IF(M.NE.K) CALL NEXISTC(C,N)

C       IF IT IS THE LAST LOOP THEN USE THIS BICOVARIANCE TO COMPUTE G

    IF(M.EQ.K) CALL GENERG(C,CG,N,SIG,RWK,IWK,CWK,TEST)

    IF(M.EQ.K) THEN
        DO 50 I=1,128
        DO 50 J=1,128
            WRITE(1,3) I-1,J-1,CG(I,J)
50      CONTINUE
    ENDIF
    CALL FFT3D(C,128,128,N,N,1,1,IWK,RWK,CWK)
    IF(M.EQ.K) THEN
        DO 20 J=1,N2/2+1
        DO 20 I=1,N2+1
            IF(J.LE.I.AND.J+I.LE.N2) THEN
                WRITE(15,*) SQRT(REAL(C(I,J)*CONJG(C(I,J))))
            ELSE
                WRITE(15,*) 0.
            ENDIF
20      CONTINUE
        WRITE(1,2) I-1,J-1,C(I,J),M
    ENDIF
1   FORMAT(' C(',I2,',',I2,') = ',2(E16.6),'    K= ',I2)
2   FORMAT(' B(',I2,',',I2,') = ',2(E16.6),'    K= ',I2)
3   FORMAT(' G(',I3,',',I3,') = ',2(E16.6))
    PRINT*,'FINISHED THE ',M,'TH ITERATION OF THE BISPECTRUM'
30  CONTINUE
```

```
      ERR=0
99    CONTINUE
      CLOSE(1)
      CLOSE(2)
      CLOSE(15)
      RETURN
      END
```

```
      SUBROUTINE GENERG(C,CG,N,SIG,RWK,IWK,CWK,TEST)

C-----------------------------------------------------------------
C       THIS SUB COMPUTES THE KERNELS FROM THE BICOVARIANCE
C           IT USES A SYMMETRICAL RELATION BETWEEN THE WEIGHTS,
C           HOWEVER ANY RELATION COULD BE USED.
C-----------------------------------------------------------------

      COMPLEX CG(128,128),C(128,128),CWK(128)

      REAL RWK(918),SIG

      INTEGER N,IWK(918),TEST

      OPEN(UNIT=8,FILE='WEIGHTS')
      REWIND(UNIT=8)

C       EITHER THE WEIGHTS ARE READ IN OR THE BICOVARIANCE
C           COMPUTED IN SUB. BISPIN IS WRITTEN OUT TO A FILE

      IF(TEST.EQ.1)  THEN
          WRITE(8,*) N/2
          WRITE(8,*) N/2
          DO 30 I=1,N/2
          DO 30 J=1,N/2
              WRITE(8,*) REAL(C(I,J))
30        CONTINUE
      ENDIF

      IF(TEST.EQ.2) THEN
          PRINT*,'INPUTING G'
          DO 5 I=1,N/2
          DO 5 J=1,N/2
              READ(9,*) VALUE
              C(I,J)=CMPLX(VALUE)
5         CONTINUE
      ENDIF
C       FIND THE WEIGHTS
      DO 10 I=2,N/2
          CG(1,I)=.25*C(1,I)*(1./SIG**4)
          CG(I,1)=CG(1,I)
10    CONTINUE
      CG(1,1)=(1./6.)*C(1,1)*(1./SIG**4)
      DO 20 I=2,N/2
      DO 20 J=2,N/2
          CG(I,J)=.5*C(I,J)*(1./SIG**4)
20    CONTINUE

      PRINT*,'COMPUTING THE TRANSFORM OF THE WEIGHTS'
      CALL FFT3D(CG,128,128,128,128,1,1,IWK,RWK,CWK)
3     FORMAT(' G(',I3,',',I3,') = ',2(E16.6))
      DO 50 I=1,128
      DO 50 J=1,128
50        WRITE(1,3) I-1,J-1,CG(I,J)
```

```
CLOSE(8)
RETURN
END
```

```
      SUBROUTINE BISPGEN(CG,C,CXTS,LTS,SIG,NREC)

C-----------------------------------------------------------------
C       THIS SUB COMPUTES THE BISPECTRUM FROM THE WEIGHTS AND IT
C           COMPUTES THE BISPECTRUM OF THE AVERAGED TIME SERIES.
C           THIS IS A HIGH VARIANCE ESTIMATE OF THE BISPECTRUM.
C-----------------------------------------------------------------

      COMPLEX CXTS(128),CG(128,128),PMET,C(128,128)

      INTEGER LTS,L1,L2,L12,IJ,K,M,I,J,NREC

      REAL TEMP,SIG

      PRINT*,'COMPUTING BISPECTRUM'
      OPEN(UNIT=16,FILE='PBISPT')
      OPEN(UNIT=17,FILE='PBISPE')
      REWIND(16)
      REWIND(17)
      CALL ZERO2(C,128)
      K=LTS/4

      DO 10 I=0,LTS/2
          IF(I.LE.K) THEN
              M=I
          ELSE
              M=LTS/2-I
          ENDIF
          DO 20 J=0,M
              IF(J.NE.0)THEN
                  L2=128-J+1
              ELSE
                  L2=1
              ENDIF

              IF(I.NE.0) THEN
                  L1=128-I+1
              ELSE
                  L1=1
              ENDIF
              IJ=I+J+1
              PMET=CG(I+1,J+1)+CONJG(CG(L1,IJ))+CONJG(CG(IJ,L1))
              PMET=PMET+CONJG(CG(L2,IJ))+CONJG(CG(IJ,L2))+CG(J+1,I+1)
              WRITE(16,*) PMET
              PMET=CXTS(I+1)*CXTS(J+1)*CONJG(CXTS(IJ))
              WRITE(17,*) PMET
20        CONTINUE
          DO 30 J=M+1,K
              PMET=CMPLX(0.)
              WRITE(17,*) PMET
              WRITE(16,*) PMET
30        CONTINUE
10    CONTINUE
```

```
C       THE REMAINING CODE SIMPLY WRITES THE DATA OUT IN A FORM THAT
C         CAN BE PLOTTED.

      REWIND(16)
      REWIND(17)
      DO 40 I=0,LTS/2
      DO 40 J=0,K
          READ(16,*) C(I+1,J+1)
40    CONTINUE
      REWIND(16)
      WRITE(16,*) K+1
      WRITE(16,*) LTS/2+1
      DO 50 I=0,LTS/2
      DO 50 J=K,0,-1
          PMET=C(I+1,J+1)
          TEMP=(SIG**4)*SQRT(REAL(PMET*CONJG(PMET)))
          WRITE(16,*) TEMP
50    CONTINUE
      DO 60 I=0,LTS/2
      DO 60 J=0,K
          READ(17,*) C(I+1,J+1)
60    CONTINUE
      REWIND(17)
      WRITE(17,*) K+1
      WRITE(17,*) LTS/2+1
      DO 70 I=0,LTS/2
      DO 70 J=K,0,-1
          PMET=C(I+1,J+1)
          TEMP=REAL(PMET*CONJG(PMET))/NREC**3
          WRITE(17,*) TEMP
70    CONTINUE
      CLOSE(16)
      CLOSE(17)
      RETURN
      END
```

```
      SUBROUTINE REWD
      OPEN(UNIT=18,FILE='BISPDAT')
      OPEN(UNIT=7,FILE='TIMSER2')
      REWIND(5)
      REWIND(6)
      REWIND(7)
      REWIND(18)
      WRITE(18,*) 0
      CLOSE(18)
      CLOSE(7)
      RETURN
      END


      SUBROUTINE POWSPEC(CG,SIG,PST,MAX)
      COMPLEX CG(128,128)
      REAL PST(128),SIG,SUM,MAX
      MAX=0
      DO 10 I=0,127
          SUM=0
          DO 20 J=0,127
              M=I-J
              IF(M.LT.0) M=128+M
              SUM=SUM+REAL(CG(J+1,M+1)*CONJG(CG(J+1,M+1)))
20        CONTINUE
          PST(I+1)=2*SIG**2+SUM*(SIG**4)
          IF(PST(I+1).GT.MAX) MAX=PST(I+1)
10    CONTINUE
      RETURN
      END


      SUBROUTINE ZERO(A,ND)
      COMPLEX A(128)
      DO 10 I=1,ND
          A(I)=CMPLX(0)
10    CONTINUE
      RETURN
      END

      SUBROUTINE BSDATFL(N,NREC)

C         THIS SUB CREATES THE NECESSARY FILES WHICH CAN BE READ BY THE
C           BISPECTRUM ESTIMATION ALGORITHM

      INTEGER NUM,N,NREC
      NUM=N*NREC
      OPEN(UNIT=10,FILE='REFINF')
      OPEN(UNIT=11,FILE='FFTINF')
      OPEN(UNIT=12,FILE='PWRGINF')
      OPEN(UNIT=13,FILE='PWRAINF')
      OPEN(UNIT=14,FILE='BISPINF')
```

```
      REWIND(10)
      REWIND(11)
      REWIND(12)
      REWIND(13)
      REWIND(14)
       WRITE(10,*) '"TIMSER"'
       WRITE(10,*) '"TSPIO"'
       WRITE(10,*) '1,',NUM,',1'
       WRITE(10,*) '1,1,1'
      WRITE(10,*) N
       WRITE(10,*) '"REAL"'
       WRITE(10,*) '"REAL"'
       WRITE(10,*) '1'
       WRITE(10,*) '"(E17.8)"'
       WRITE(10,*) '17'
       WRITE(11,*) '"TSPIO"'
       WRITE(11,*) '"FFT"'
       WRITE(11,*) '1,',NREC,',1'
      WRITE(11,*) N
       WRITE(12,*) '"FFT"'
       WRITE(12,*) '"POWSPC"'
       WRITE(12,*) '1,',NREC,',1'
      WRITE(12,*) 2*N
       WRITE(13,*) '"POWSPC"'
      WRITE(13,*)'"NEWPWR"'
       WRITE(13,*) '1,',NREC,',1'
      WRITE(13,*) N/2+1
      WRITE(13,*) NREC
      CLOSE(13)
      OPEN(UNIT=14,FILE='BISPINF')
       WRITE(14,*)'"SPIO"'
       WRITE(14,*)'"FFT"'
       WRITE(14,*) '1,',NREC,',1'
       WRITE(14,*) '1,',2*N,',1'
       WRITE(14,*)'"Y"'
       WRITE(14,*)'"SPIO"'
       WRITE(14,*)'"NEWPWR"'
       WRITE(14,*) '1,1,1'
       WRITE(14,*) '1,',N/2+1,',1'
      WRITE(14,*) N
      WRITE(14,*) '2'
      WRITE(14,*)'"N"'
      WRITE(14,*) NREC
       WRITE(14,*)'"BISPRT"'
      WRITE(14,*)'"PWAVG"'
      WRITE(14,*)'"BIRAW"'
       WRITE(14,*)'"CHI"'
       WRITE(14,*)'"N"'
       WRITE(14,*) '"BISUM"'
       CLOSE(10)
       CLOSE(11)
       CLOSE(12)
       CLOSE(14)
       RETURN
```

```
      END


      SUBROUTINE NEXISTC(C,N)
      COMPLEX C(128,128),Z
      INTEGER N
      Z=CMPLX(0)
      DO 5 I=1,N
      DO 5 J=1,N
          C(I,J)=CMPLX(REAL(C(I,J)))
5     CONTINUE
      DO 10 I=2,N/2
      DO 10 J=I,N/2+1
          C(N/2+I,J)=Z
          C(J,N/2+I)=Z
10    CONTINUE
      DO 20 I=1,N
          C(N/2+1,I)=Z
          C(I,N/2+1)=Z
20    CONTINUE
      RETURN
      END


      SUBROUTINE NEXISTB(C,N)
      INTEGER NCT
      COMPLEX C(128,128)
      DO 10 I=1,N,2
      DO 10 J=1,N,2
          C(I,J)=CMPLX(0)
10    CONTINUE

C     ZERO OUT THE BISP THAT DOESNT EXIST- IF NYQUIST IS TO BE SATISFIED

      NCT=2
      DO 20 J=N/2-1,N/4+1,-1
          DO 30 I=NCT,J
              C(I+1,J+1)=CMPLX(0)
              C(J+1,I+1)=CMPLX(0)
30        CONTINUE
          NCT=NCT+1
20    CONTINUE
      NCT=N-2
      DO 40 J=N/2+1,N-2
          DO 50 I=N/2+1,NCT
              C(I+1,J+1)=CMPLX(0)
              C(J+1,I+1)=CMPLX(0)
50        CONTINUE
          NCT=NCT-1
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZERO2(B,M)
      INTEGER M
      COMPLEX B(128,128)
      DO 10 I=1,M
      DO 10 J=1,M
          B(I,J)=CMPLX(0)
10    CONTINUE
      RETURN
      END
```

REFERENCES

Brillinger, David R. (1975) *Time Series Data Analysis and Theory,* Holt, Rinehart and Winston Inc., New York.

Brillinger, David R. (1965) "An Introduction to Polyspectra." *Ann. of Math. Statistics,* Vol. 36, pp. 1351-1374.

Brillinger, David R. and Rosenblatt, Murray (1967a) "Asymptotic Theory of Estimates of k-th Order Spectra." In *Spectral Analysis of Time Series* (ed. by Bernard Harris) pp. 153-188, Wiley, New York.

Brillinger, David R. and Rosenblatt, Murray (1967b) "Computation and Interpretation of k-th Order Spectra." In *Spectral Analysis of Time Series* (ed. by Bernard Harris) pp. 189-232, Wiley, New York.

Choi, Doowhan (1984) "Engineering Applications of Higher Order Spectra." Unpublished Ph.D. Dissertation presented to The University of Texas.

Dudgeon, Dan E. and Mersereau, Russell M. (1984) *Multidimensional Digital Signal Processing,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Hinich, Melvin J. (1979) "Quadratic Multiplexing- A New Method to Secure Communication." *IEEE Transactions on Information Theory,* Vol. 25, No. 3, pp. 354-356.

Hinich, Melvin J. (1982) "Testing for Gaussianity and Linearity of a Stationary Time Series." *J. Time Series Analysis* , Vol. 3, No. 3, pp. 169-176.

Hinich, Melvin J. and Wolinsky, Murray (1988) "A Test for Aliasing Using Bispectral Analysis." Accepted for publication in *J. Stat. Soc. Am.*

Kendall, Maurice G. and Stuart, Alan (1958) *Advanced Theory Of Statistics, Vol. 1.* pp. 55-93, Hafner Publishing Co. New York.

Kim,Young C. and Powers, Edward J. (1979) "Digital Bispectral Analysis and Its Applications to Nonlinear Wave Interactions," *IEEE Transactions on Plasma Science*, Vol. 7, No. 2, pp. 120-131.

Leonov, V. P. and Shiryaev, A. N. (1959) " On a Method of Calculation of Semi-invariants." *Theory of Prob. Appl.*, Vol. 4, No. 3, pp. 319-329.

Loeve, Michel (1977) *Probability Theory Vol. 1 and 2*, Springer Verlag, New York.

Nikias, Chrysostomos C. and Raghuveer, Mysore R. (1987) " Bispectrum Estimation: A Digital Signal Processing Framework." *Proceedings of the IEEE*, Vol. 75, No. 7, pp. 869-891.

Priestly, M. B. (1981) *Spectral Analysis and Time Series*, Vol. 1 and 2, Academic Press, Inc.

Raghuveer, Mysore R. and Nikias, Chrysostomos C. (1986) "Bispectrum Estimation via AR Modeling." *Signal Processing*, Vol. 9, No. 1, pp.35-48.

Rao, T. Subba and Gabr, M. M. (1984) *An Introduction to Bispectral Analysis and Bilinear Time Series Models*, pp. 1-115, Springer Verlag, New York.

Rosenblatt, Murray (1981) "Cumulants and Cumulant Spectra." In *Handbook of Statistics*, Vol. 3 (eds. P. R. Krishnaiah and D. R. Brillinger), pp. 369-382, Elsevier Science Publishers, Amsterdam.

Rosenblatt, Murray and Van Ness, J. W. (1965) "Estimation of the Bispectrum." *Ann. of Math. Statistics,* Vol. 36, pp. 1120-1135.

Wise, Gary L. Unpublished Lecture Notes.

Wolinsky, Murray (1988) "Invitation to the Bispectrum" Applied Research Laboratories Technical Report No. 88-7 (ARL-TR-88-7), Applied Research Laboratories, The University of Texas at Austin.

13 April 1988

DISTRIBUTION LIST FOR
ARL-TR-88-19
UNDER ARL:UT INDEPENDENT RESEARCH AND
DEVELOPMENT PROGRAM

Copy No.	
	Commander Naval Air Systems Command Department of the Navy Washington, D.C. 20361-9330
1	Attn: W. Parigian (Code 933A)
2	CDR T. Santry (PMA 264-D)
	Commander Naval Sea Systems Command Department of the Navy Washington, D.C. 20362-5101
3	Attn: Y. Yam (Code 63D4)
4	R. Lynch (Code 63D41)
5	M. Schultz (Code 63D22)
6	M. Montroll (PMS394A)
7	C. Walston (Code 61Z137)
8	S. Wieczorek (Code 55NB)
9	E. Plummer (Code 63D1)
	Commander Space and Naval Warfare Systems Command Department of the Navy Washington, D.C. 20363-5100
10	Attn: W. Richter (Code 80-B)
11	R. Cockerill (Code 180-521)
12	J. Bertrand (Code 183-1)
13	L. Parish (Code 180-4)
14	R. Mitnick (Code 180-5)
15	CAPT J. Jensen (Code 141)
	Office of the Chief of Naval Research Department of the Navy Arlington, VA 22217-5000
16	Attn: J. Smith (Code 1211)
17	N. Gerr (Code 1111SP)
18	R. Fitzgerald (Code 1125AO)
19	K. Lackie (Code 132)
20	G. Remmers (Code 233)
21	J. Warfield (Code 234)

Distribution List for ARL-TR-88-19
(cont'd)

Copy No.

Office of the Chief of Naval Operations
Department of the Navy
Washington, D.C. 20350-2000
22 Attn: LT D. David (OP 092V)
23 T. Long (0P 092V)

Commanding Officer
Naval Ocean Systems Center
San Diego, CA 92152
24 Attn: C. Persons (Code 732)
25 T. Albert (Code 632)
26 J. Silva
27 D. Hanna

Commanding Officer
David Taylor Research Center
Carderock Laboratory
Bethesda, MD 20084-5000
28 Attn: B. Douglas (Code 1961)
29 J. Valentine (Code 1932)
30 W. Hart (Code 1936)
31 N. Keech (Code 1926)
32 W. Reader (Code 1905.2)
33 J. Niemiec (Code 1965)

Commanding Officer
David Taylor Research Center
Annapolis Laboratory
Annapolis, MD 21402-5067
34 Attn: L. Riddle (Code 1927)

Commanding Officer
Naval Intelligence Support Center
4301 Suitland Rd.
Washington, DC 20390
35 Attn: J. Conrad (Code 20)
36 C. Clark (Code 20)

Commanding Officer
Naval Air Development Center
Warminster, PA 18974
37 Attn: B. Steinberg (Code 5031)
38 L. Allen (Code 5033)
39 T. Posimonti

Distribution List for ARL-TR-88-19
(cont'd)

Copy No.

Director
Naval Research Laboratory
Washington, DC 20375

40 Attn: D. Bradley (Code 200)
41 W. Kuperman (Code 5160)
42 O. Diachok (Code 5120)
43 P. Mignerey

Commanding Officer
Naval Underwater Systems Center
New London Laboratory
New London, CT 06320

44 Attn: B. Cole (Code 33A)
45 P. Herstein (Code 33A3)
46 G. Mayer (Code 33501)
47 R. Tozier (Code 2111)
48 J. Martins (Code 2111)
49 M. Maguire (Code 2111)
50 W. Carey
51 R. Tompkins (Code 33A2)
52 R. Radlinski (Code 311)
53 R. Dwyer

Commanding Officer
Naval Surface Warfare Center
White Oak Laboratory
Silver Spring, MD 20910

54 Attn: E. Hein
55 M. Williams

Commanding Officer
Naval Oceanographic Office
Stennis Space Center, MS 39522-5001

56 Attn: W. Jobst (Code 7300)

Commanding Officer
Naval Ocean Research and Development Activity
Stennis Space Center, MS 39529-5004

57 Attn: T. Goldsberry (Code 240)

Distribution List for ARL-TR-88-19
(cont'd)

Copy No.

	Defense Advanced Research Projects Agency Department of the Navy Arlington, VA 22209
58	Attn: C. Stuart
59	K. Hawker
	United States Naval Academy Annapolis, MD 21402
60	Attn: Library
61-72	Commanding Officer and Director Defense Technical Information Center Cameron Station, Building 5 5010 Duke Street Alexandria, VA 22314
	Director Johns Hopkins University Applied Physics Laboratory Johns Hopkins Rd. Laurel, MD 20707
73	Attn: F. Vaughan
74	Garland R. Barnard, ARL:UT
75	Kevin Baugh, ARL:UT
76	Eugene W. Brown, ARL:UT
77	George C. Coble, ARL:UT
78	Marshall E. Frazer, ARL:UT
79	Thomas A. Griffy, ARL:UT
80	Melvin J. Hinich, ARL:UT
81	John M. Huckabay, ARL:UT
82	Clark S. Penrod, ARL:UT

Distribution List for ARL-TR-88-19
(cont'd)

Copy No.	
83	Jack Sheehan, ARL:UT
84	Arnold J. Tucker, ARL:UT
85	Gary R. Wilson, ARL:UT
86	Library, ARL:UT
87-96	Reserve, ARL:UT