AFCRL - 65-207

AD614044

ON FACTORING THE CORRELATIONS OF DISCRETE MULTIVARIABLE STOCHASTIC PROCESSES

Ralph A. Wiggins

Massachusetts Institute of Technology
Cambridge 39, Massachusetts

Scientific Report No. 9 of Contract
AF 19(604) 7378
S. M. Simpson, Director
February 28, 1965
Project 8652
Task 865203

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

WORK SPONSORED BY ADVANCED RESEARCH PROJECTS AGENCY

PROJECT VELA-UNIFORM
ARPA Order No. 180-61, Amendment 2

AFCRL - 65-207

ON FACTORING THE CORRELATIONS
OF DISCRETE MULTIVARIABLE STOCHASTIC PROCESSES

Ralph A. Wiggins

Massachusetts Institute of Technology
Cambridge 39, Massachusetts

Scientific Report No. 9 of Contract
AF 19(604) 7378
S. M. Simpson, Director
February 28, 1965
Project 8652
Task 865203

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

WORK SPONSORED BY ADVANCED RESEARCH PROJECTS AGENCY

PROJECT VELA-UNIFORM
ARPA Order No. 180-61, Amendment 2

Requests for additional copies by agencies of the Department of Defense, their contractors, or other government agencies should be directed to:

Defense Documentation Center(DDC)
Cameron Station
Alexandria, Virgina 22314

Department of Defense contractors must be established for DDC services or have their "need-to-know" certified by the cognizant military agency of their project or contract.

All other persons and organizations should apply to the:

Clearinghouse for Federal Scientific
and Technical Information (CFSTI)
Sills Building
5285 Port Royal Road
Springfield, Virginia 22151

ON FACTORING THE CORRELATIONS
OF DISCRETE MULTIVARIABLE STOCHASTIC PROCESSES

by Ralph Ambrose Wiggins

Submitted to the Department of Geology and Geophysics on March 2, 1965 in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

This thesis is an extension of the theory of discrete scalar time series analysis to multivariable processes. This extension is facilitated by expanding the algebra of polynomial matrices (matrices with polynomial elements).

Multivariable processes may have a multiplicity of either the independent or the dependent variable. Such processes are called multi-dimensional or multi-channel, respectively. All multi-dimensional processes may be formally mapped into matrix notation. Once this mapping is made the properties of all multivariable linear operators and autocorrelations can be studied in terms of the polynomial matrices that represent their z-transforms.

Polynomial matrices can be decomposed into three related forms: the spectral factorization, the Smith-McMillan canonical form, or the Robinson canonical form. Each of these representations leads to the concept of an invertible or minimum delay wavelet.

The algorithms for finding the spectral factorization and for finding the Smith-McMillan canonical form can be extended to provide an analytic factorization of a multi-channel autocorrelation in term of invertible wavelets. In addition the autocorrelation may be approximately factored by a recursive least-squares algorithm, or by a projection technique.

Of the factorization methods available, the recursive algorithm is the most efficient and is therefore extended to include the more general problem of signal shaping in the presence of noise.

Finally, as an illustration, the problem of designing a finite optimum two-dimensional band-pass, band-reject filter is solved and the characteristics of a few particular realizations of such filters are presented.

Thesis Supervisor: Stephen M. Simpson, Jr.
Title: Lecturer in Geology and Geophysics

BLANK PAGE

ACKNOWLEDGMENTS

I wish to express my sincere thanks and appreciation to Dr. Stephen M. Simpson, Jr., and to Dr. Enders A. Robinson for their help, ideas, and encouragement during my preparation of this thesis. Valuable aid and criticism were also given by Professor Theodore R. Madden and by Professor Henry P. McKean, Jr.

I am grateful to my fellow students, James N. Galbraith, Jr., and Donald J. Krotser, and especially to Jon F. Claerbout and Carl Wunsch, for their assistance, discussions and criticisms during the many phases of preparation of the thesis.

I am also very grateful to Joseph T. Procito, Jr., for his invaluable assistance with the preparation of the many computer programs that supported the research presented here, and to Mrs. Barbara Cullum for her excellent work in typing the thesis.

Acknowledgment is also extended to Geoscience Inc. for support and for use of certain computer programs and to the Advanced Research Projects Agency which sponsored this work as part of Contract AF 19(604)7378.

TABLE OF CONTENTS

LIST OF FIGURES

BLANK PAGE

1. INTRODUCTION

Geophysics may be viewed as the study of the properties of the earth by the interpretation of signals that are affected by the structure of the earth. These signals may be of almost any conceivable type -- seismic, tidal, electric current, electromagnetic, or light -- and may have a wide variety of sources. In each case the complexity of the media that modulates the signals will introduce noise into the system (we define noise as any portion of a signal which does not contain information that we desire). In addition, the information may be difficult to interpret because the signal shapes are difficult to recognize.

The idea of applying the concepts of statistical analysis to signal interpretation has become widely accepted during the last decade. A large portion of this analysis has taken the form of applying linear filters to incoming data to enhance and shape the desired information. Because of its versatility, the least-squares optimum (Wiener) filter was frequently applied. However, a problem arose in the computation of such filters for geophysical applications. Geophysical signals are usually multivariable; that is, the signals are characterized by having more than one independent variable (dimension) or by having more than one dependent variable (channel). In most cases, to adequately process such multivariable signals one should use multivariable filters. However, the solution of the discrete least-squares filter problem is a set of simultaneous equations, with one equation for each coefficient. Thus the magnitude of the problem quickly overloaded the capacity of even the largest computers available. This limitation on the size of possible filters greatly restricted their usefulness.

The computational problem was reduced by an order of magnitude in storage space and execution time by Robinson (1963a) when he was able to extend a recursive method introduced by Levinson (1947) to multivariable filter generation. This development has led to greatly renewed interest in the applications of optimum filtering.

The crucial step in optimum least-squares filter design is the factorization of the autocorrelation of a process. This factorization is the problem to which this thesis is addressed.

There are now four known techniques for factoring the autocorrelations of multivariable processes. As indicated above, the least-squares approximate factorization has been known for some time (for example, see Wadsworth, et al., 1953). However, before the discovery of the recursive computation algorithm, it was not considered to be useful. In fact it was this consideration that led Wiener and Masani (1957 and 1958) to develop a projection technique of approximate factorization. Experience now shows that this technique is not competitive with the recursive method. Quenouille (1957) presented an analytic factorization algorithm which, when placed upon a rigorous mathematical basis, has proven to be a very valuable tool for understanding and manipulating multivariable time series and autocorrelations. Another analytic factorization method is developed here based upon the Smith canonical form for polynomial matrices following a similar development by Youla (1961). Neither of these analytic methods are computationally competitive with the least-squares recursive algorithm although they are invaluable for instilling theoretical insight into the factorization problem.

All of the factorization schemes that are considered are stated for discrete processes with finite autocorrelations. Since these factors (which we call wavelets) are also finite they are members of the Hardy class (Wiener and Masani, 1957, p. 113). Because we are dealing with finite wavelets we are able to obtain specific results which are of a more constructive nature than those found in some more generalized approaches. As such this thesis may be considered as a complement to recent works on stochastic processes such as Helson and Lowdenslager (1958), Robinson (1962), and Wiener and Masani (1957 and 1958).

This thesis then is primarily an examination and evaluation of the methods of factoring multivariable autocorrelations. From another point of view, however, it may be thought of as a treatise on polynomial and rational matrices, that is, on matrices whose elements are either polynomial or rational. This is a subject that has received surprisingly little attention in the literature. For this reason it is given a rather thorough development here in the first three chapters.

The final chapters are devoted to an expansion of the least-squares approximate factorization to the calculation of filters with specified noise suppressing and signal shaping properties. Computational examples are included that illustrate some of the forms that such computations may take.

The presentation that follows assumes a basic knowledge of scalar, i.e. single-variable, time series analysis (see Lee, 1960; Robinson, 1962; Wiener, 1949; or Whittle, 1963). Most of the primary ideas, such as wavelets, all-pass systems, minimum phase, minimum delay, convolution, autocorrelation, and predictive decomposition are reviewed briefly when they are first encountered but are not developed rigorously. The material here is not intended to be a review of time series analysis, but is intended to be an extension of the concepts of scalar time series to multivariable time series. On the other hand, much of the detail considered is not necessary for an overall grasp of multivariable time series analysis. Thus the reader who is unfamiliar with the subject may profitably skip over several sections. These sections include 4.221 (the details of the factorization of an elementary autocorrelation matrix), 4.223 (the Smith-McMillan factorization technique), 4.232 (the Wiener-Masani approximate factorization by projections), and 5.12 (the details of the recursive algorithm for least-squares filters).

2. DEFINITIONS AND NOTATION

Processes may have multiple independent or dependent variables. In sections 3.4 and 4.3 a technique is developed for mapping processes with several independent variables into a form with several dependent variables and only one independent variable. This mapping is given in order to simplify the analysis and notation of the factorization problem. However, there are important differences between these two representations that should be recalled when applications are made of factorization. This chapter is devoted primarily to an examination of these differences. In addition, a few general notational questions are examined.

2.1 Dimensionality of Processes

A dimension is defined as a measurable extent. In this thesis the number of dimensions of a process will indicate the number of orthogonal measurable directions, i.e., the number of independent variables. Most processes that have been considered in communication theory and in economic analysis are one-dimensional time series. However, in geophysics higher dimensioned processes are often encountered that may or may not have a time-like dimension. For example, the output of a single vertical seismometer is a one-dimensional time process. The output of a linear row of seismomenters is a two-dimensional process -- one time dimension and one space dimension. On the other hand (neglecting small, higher order effects) the acceleration of gravity at each of these seismometer locations represents a one-dimensional spatial process.

In nearly all of our analysis we shall assume that one of the dimensions, or directions, of a process is a preferred (time-like) direction. We do this for several reasons. (1) In many processes there actually exists a preferred direction. It is only natural to take advantage of the physical significance of this direction. (2) The use of vector notation greatly simplifies the representation of processes with a preferred direction. (3) Present digital computers have one-dimensional storage memories. Thus when a process is mapped into a computational scheme, we must necessarily choose a preferred direction.

Whittle (1954, p. 434) has pointed out that there is a basic difference between a preferred direction that has time, or time-like, physical significance and a direction that is chosen merely for notational purposes. A time-like direction is inherently one-sided. That is, the state of a process at any time can be dependent only upon past values of the process. However, purely spatial processes are usually not one-sided. This distinction is important when designing operators for processes.

The importance of the preferred direction is emphasized when we define the geometrical structure of the sampling of the independent variables of a process. We can think of this structure as an array of sample points in a multi-dimensional space. For nearly all applications we will restrict these points to being equally spaced along straight lines that are parallel to the preferred direction. This is equivalent to saying that the process will have equal digitization increments in the preferred direction and fixed sampling positions in the other directions. Although the digitization increment is fixed, the sampling instants for the various positions need not be in phase.

For most applications in this paper, we will require that the lines form regular patterns in the other dimensions. The simplest, and most useful, pattern is that of a rectangular grid. However, other patterns (triangles, parallelograms, hexagons, and combinations of these in higher dimensions) are frequently encountered.

2.2 Order of Processes

Processes may have multiple dependent variables as well as multiple independent variables. In general, the dependent variables need not have any dimensional relationship. For example, one variable may represent the electric field while another may represent the magnetic field. The <u>order</u> of a process is the number of dependent variables that represent a process at each point in space. Thus, a linear array of 3 component seismometers would be a 3rd order, 2-dimensional process.

Throughout this paper we will refer to processes of order greater than 1 as multi-channel or matrix-valued. The latter designation stems from the fact that we will use a matrix representation to group the variables of a process.

The one-dimensional, multi-channel process is of special interest since its configuration best reflects the importance of the preferred direction. This fact sometimes prompts us to view each of the space samples of a multi-dimensional process as one channel for a higher-ordered multi-channel process. Thus, a linear array of 12 three-component seismometers might be viewed as a 3rd order, 2-dimensional process, or, viewing each seismometer as providing a separate time series (channel), we may view this as a 36th order, one-dimensional process.

Even though a mapping from a multi-dimensional process to a higher-ordered, one-dimensional process is possible, the basic differences between these representations should be emphasized. First, we usually think of a discrete, multi-dimensional process as a manifestation of a continuous function. Thus, it is possible to approximate values between the digitization positions by some form of interpolation. Second, in a multi-dimensional process we can think of extending the space dimensions to infinity. The formal structure of a multi-channel process allows neither of these possibilities.

2.3 Subscript Notation

Subscripts will be used to indicate the variables of a process. In general, there will be two groups of subscripts. The first group will refer to indexing of the independent variables; the second group will refer to indexing of the dependent variables. We will adopt the convention that the first subscript in the first group will always stand for the preferred direction. Thus a component of a process X may be referred to as

$$(X_{i,i_1, \ldots, i_N})_{k_1,k_2}$$

or, if the preferred direction refers specifically to time, it will be written

$$(X_{t,i_1, \ldots, i_N})_{k_1,k_2} \quad .$$

Since matrices are at most 2-dimensional, the second group will have at most 2 indices. We will always consider that the first of these 2 indices will be the row index, and that the second will be the column index.

In order to simplify our writing we shall adopt a vector notation for the subscripts

$$\underline{i} = (i,i_1, \ldots, i_N)$$

$$\underline{k} = (k_1,k_2)$$

so that the process may also be referred to as

$$(X_{\underline{i}})_{\underline{k}}$$

Frequently it is desirable to order the spatial sampling positions (i.e. the sampling positions in the non preferred direction) sequentially. Thus we may use one subscript for all spatial variables:

$$(X_{t,i})_{\underline{k}}$$

This subscript takes on a different value for each sampling position. Finally, for much of our work we will be concerned only with the dimensional indices and will suppress the matrix indices and the parentheses.

2.4 Flow Diagram Notation

The important decomposition and factorization theorems in the following chapters are illustrated by flow diagrams. In general, these diagrams are self explanatory, however, a description of some of the conventions used will facilitate their interpretation.

1. Square boxes indicate operations.
2. Rounded boxes illustrate results of operations.
3. Sol'd lines between boxes indicate the primary lines of logical flow as well as transference of data between steps.
4. Dotted lines between boxes indicate only the transference of data between steps.
5. Boxes drawn with heavy lines indicate the beginning and the ending of the algorithm.

3. STRUCTURE OF DISCRETE LINEAR OPERATORS

The operators that we consider are finite moving average devices that may be represented by the diagram

$$\text{Input} \longrightarrow \boxed{\begin{array}{c}\text{Linear}\\ \text{Operator}\end{array}} \longrightarrow \text{Output} \quad .$$

If the input is a spike (a delta function appropriate to the geometry involved) then the output is a wavelet with real coefficients which completely describes the properties of the linear operator. In fact the output of the linear operator for a general input is just the convolution of the input with the wavelet.

In this chapter we will study those characteristic properties of a wavelet by which it may be classified. The approach used here is to factor a wavelet into simpler components and then to use the properties of these components to delineate the classification of the wavelet. The complexity or existence of the factorization is the key problem. In the scalar one-dimensional case there is a unique natural factorization from which the general properties are easily deducible. In the matrix-valued one-dimensional case there are a multiplicity of such factorizations. In the multi-dimensional case there is no natural factorization. Thus, our treatment for these cases will vary markedly.

3.1 z-Transform

The z-transform of a discrete finite wavelet is defined simply as the quasipolynomial

$$a(z) = \sum_{i=N}^{M} a_i z^i \quad , \qquad -\infty < N \leq M < \infty$$

whose coefficients a_i are the values of the wavelet at the $i^{\underline{th}}$ sample time. For the general multi-dimensional processes we have

$$a(z, z_1, \ldots, z_n) = \sum_{i_j = N_j}^{M_j} (a_{i, i_1, \ldots, i_n}) z^i z_1^{i_1} \ldots z_n^{i_n}$$

$$- \infty < N_j \leq M_j < \infty .$$

A quasipolynomial $a(z)$ may always be transformed into a polynomial by multiplying it by the proper power of z. The z-transform of a wavelet will be indicated specifically by writing the wavelet as a function of z as indicated above.

Two important properties of the z-transform will be exploited frequently:

1. Convolution in the time-space domain corresponds to multiplication in the z domain.

2. The z-transform evaluated on the unit circle, $z = e^{-i\omega}$, corresponds to the Fourier transform of the wavelet.

Much of the analysis in this chapter is based on the algebra of quasipolynomials that corresponds to the z-transforms of wavelets.

3.2 One-Dimensional Scalar Wavelets

One-dimensional scalar wavelets of the Hardy class have been treated extensively in the literature, (Wold, 1938; Wiener and Masani, 1957; Robinson, 1962; Whittle, 1963; Robinson and Treitel, 1964) and, therefore, the treatment here will be brief and heuristic.

3.21 Spectral Decomposition

Let us consider the one-sided wavelet

$$a_0, a_1, \ldots, a_n \quad .$$

The z-transform of this wavelet

$$a(z) = a_0 + a_1 z + \ldots + a_n z^n$$

can be factored, according to the fundamental theorem of algebra, into the form

$$a(z) = a_0(1 - \alpha_1 z) \ldots (1 - \alpha_n z)$$

where $1/\alpha_i$ $i = 1, \ldots, n$ are the zeros of the polynomial $a(z)$. These roots, $1/\alpha_i$, are generally complex but since the coefficients of $a(z)$ are real, the roots must occur in complex conjugate pairs.

3.22 Invertibility

<u>Definition 3.2-1</u>. A one-sided wavelet $a(z)$ is said to be <u>invertible</u> if there exists a one-sided wavelet $a^{-1}(z)$ such that $a(z)\, a^{-1}(z) = 1$.

The condition that the Taylor expansion of $1/a(z)$ will converge is that $a(z)$ has no zeros inside the unit circle. Thus if $|1/\alpha_i| > 1 \quad i = 1, \ldots, n$ then $a(z)$ is invertible.

<u>Jury</u> (1964) reviews several simple techniques for testing for the invertibility of a wavelet. One of the simpler conditions involves polynomial divisions to find the number of roots inside the unit circle. The procedure begins by performing the division

$$\frac{a(z)}{z^n\, a(1/z)} = q_0 + \frac{a_1(z)}{z^n\, a(1/z)}$$

where $a_1(z)$ is the remainder. Then we find the other $q_i \quad i = 1, \ldots, n-2$ according to

$$q_i = \frac{a_i(z)}{z^{n-i}\, a_i(1/z)} - \frac{a_{i+1}(z)}{z^{n-i}\, a_i(1/z)}$$

Now the number of roots inside the unit circle is equal to the number of products P_k which are negative, where P_k is defined as

$$P_k = \Big[|q_0| - 1\Big]\left[\frac{1}{|q_1|} - 1\right] \cdots \left[\frac{1}{|q_{k-1}|} - 1\right] .$$

3.23 Robinson Canonical Form and All-Pass Systems

Theorem 3.2-1 (Robinson Canonical Form). Any wavelet $a(z)$ can be uniquely represented in the Robinson Canonical Form

$$a(z) = p(z)\, a_0(z)$$

where $a_0(z)$ is invertible and $p(z)$ is an all-pass system.

Let us review a few properties of all-pass systems.

Theorem 3.2-2. An all-pass discrete system has unit gain at all frequencies, i.e. $\left|p(e^{-i\omega})\right| = 1$ for all real ω.

Theorem 3.2-3. An all-pass system is trivial if $\left|p(z)\right| = 1$ for all z; that is, if $p(z)$ is constant.

Theorem 3.2-4. The inverse system to a non-trivial all-pass system is not one-sided.

The invertible factor $a_0(z)$ is completely determined by the amplitude spectrum of $a(z)$ (see 4.122).

3.24 Delay

The delay of a one-sided wavelet a_i is a measure of how the operator redistributes the energy of an input process in forming the output. It may be defined in terms of the partial energy

$$\varepsilon_i = \sum_{j=0}^{i} a_j^2$$

Robinson (1962) has proven the following Minimum Delay Theorem.

Theorem 3.2-5 (Minimum Delay). The delays of the set of wavelets $a_i(z)$ which have the same invertible Robinson canonical form $a_0(z)$ are greater than or equal to the delay of $a_0(z)$. Equality holds if and only if the all-pass system $p(z)$ is trivial. That is, the partial energies obey the relation

$$\sum_{j=0}^{k} a_{i,j}^2 \leq \sum_{j=0}^{k} a_{0,j}^2 \qquad \text{for all } k$$

where i is the wavelet index and j is the time index.

3.25 Phase

The Fourier transform of a wavelet yields frequency information about the outputs of an operator with respect to the inputs. This information is presented in the form of an amplitude change and a phase lag.

If we examine the Fourier transform of a wavelet

$$\begin{aligned} a(e^{-i\omega}) &= a_0 + a_1 e^{-i\omega} + a_2 e^{-i2\omega} + \ldots + a_n e^{-in\omega} \\ &= \alpha(\omega)\, e^{i\varpi(\omega)} \end{aligned}$$

we see that the polar representation leads to the concept of a phase lag characteristic $-\varpi(\omega)$.

We are now in a position to formulate the minimum-phase theorem:

<u>Theorem 3.2-6</u> (<u>Minimum phase</u>). The phase-lags of the set of wavelets $a_i(z)$ which have the invertible Robinson canonical form $a_0(z)$ are greater than or equal to the phase-lag of $a_0(z)$. Equality holds if and only if the all-pass system $p(z)$ is trivial. Furthermore, the phase-lag difference is

$$\varphi_i(0) - \varphi_i(\pi) = m_i\pi$$

where m_i is the number of zeros of $a_i(z)$ that are inside the unit circle.

An interesting result that follows directly from the theorem above is

<u>Corollary 3.2-6</u>. The cosine transform $a(\cos \omega)$ of a wavelet is non-negative if the wavelet $a(z)$ has no zeros inside the unit circle (i.e. is minimum phase or minimum delay) and if the wavelet is normalized so that $a(1) > 1$. The number of zeros of $a(z)$ inside the unit circle is equal to the number of zero <u>crossings</u> of the cosine-transform $a(\cos \omega)$.

The proofs to both the Minimum Phase Theorem and its Corollary follow directly from examining the nature of the definition of phase (<u>Robinson and Treitel</u>, 1964).

Figure 3.2-1 illustrates the behavior of the

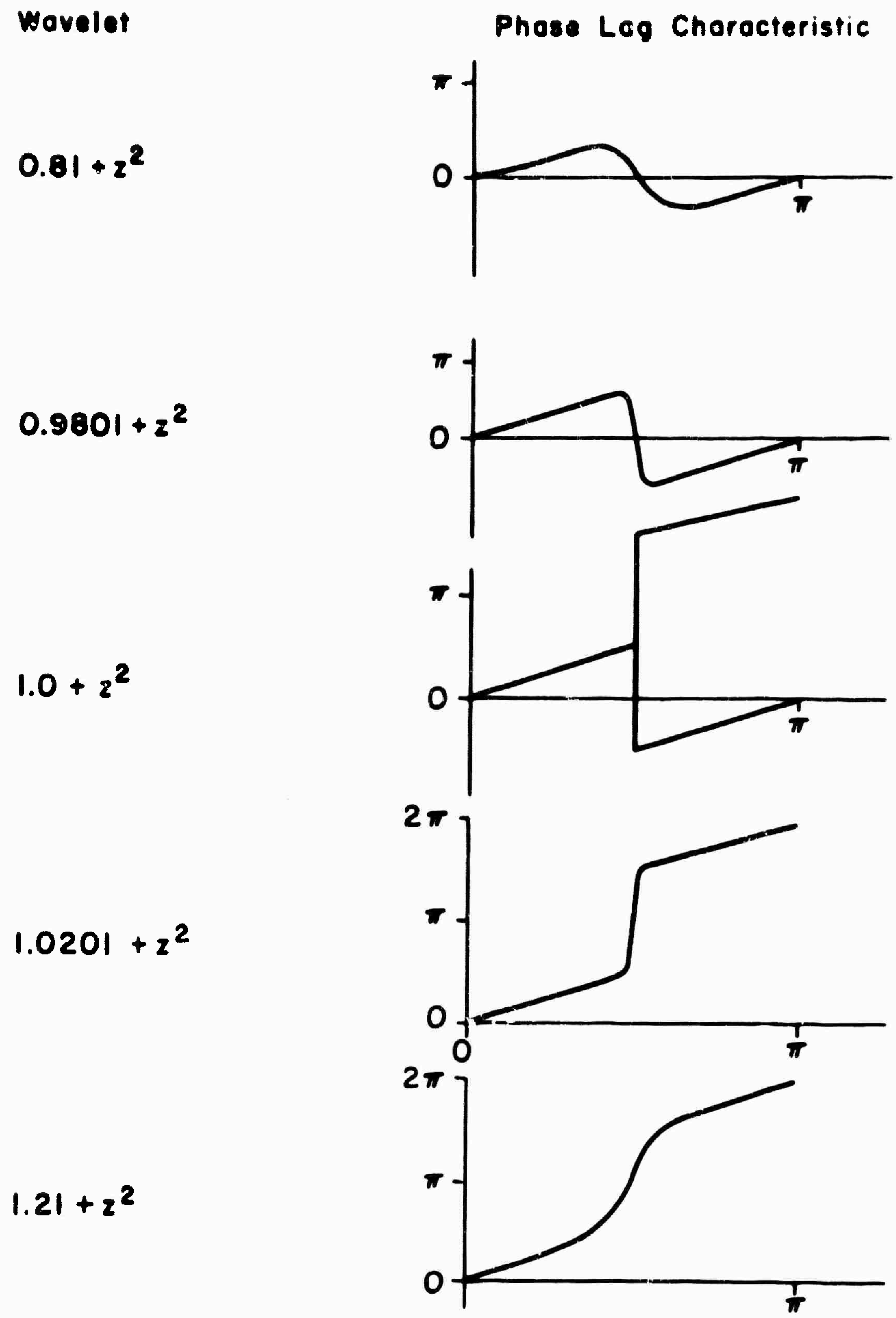


Figure 3.2 - 1: The phase-lag characteristics of 5 three term wavelets for various zero positions along the imaginary axis.

phase-lag curve for a 3-term wavelet for various positions of the zeros near the unit circle. In this case the zeros were placed on the imaginary axis so that the discontinuity for the middle curve lies at $\omega = \pi/2$. Because of this discontinuity we may interpret the wavelet either as minimum phase or maximum phase.

3.3 One-Dimensional Matrix-Valued Wavelets

Various aspects of matrix-valued wavelets, or polynomial matrices, have been treated by a number of authors. This section will review in some detail many of their important results as well as extend the theory in certain areas.

3.31 Polynomial Matrix Notation

Let us begin by reviewing the basic notation and terminology used in describing polynomial matrices.

Let A be an arbitrary matrix. Then:

A'	denotes transpose
$\bar{A}$	denotes complex conjugate
A^*	denotes complex conjugate transpose
A^{-1}	denotes inverse
Det A or $\lvert A \rvert$	denotes determinant of A
Adj A	denotes adjugate of A. (The adjugate of A is the transposed matrix of cofactors of A.

Note that Adj A/Det A $= A^{-1}$ if Det A $\neq 0$.)

A diagonal matrix A with diagonal terms $a_1, a_2, \ldots, a_n$ is written as $A = \text{diag}\left[a_1, a_2, \ldots, a_n\right]$. Column vectors are represented by $\underline{x}$, $\underline{y}$, etc., or in the alternative fashion $\underline{x} = (x_1, x_2, \ldots, x_n)'$ whenever it is desirable to indicate the components explicitly. The symbols 1_n or I, $\underline{0}_n$, and $0_{n,m}$ represent the $n \times n$ identity matrix, the n-component zero vector and the $n \times m$ zero matrix.

A matrix $A(z)$ is <u>polynomial</u> or <u>quasipolynomial</u> if each of its elements is a polynomial or quasipolynomial in z . $A(z)$ is <u>rational</u> if each of its elements is the ratio of two polynomials or quasipolynomials in z .

$A(z)$ is said to be <u>real</u> if $\bar{A}(z) = A(\bar{z})$. Unless stated otherwise, all matrices considered here will be real.

The non-negative integer $r(A)$ is the <u>rank</u> of the rational matrix $A(z)$ for a given value of z if

(1) there exists at least one subminor of order r which does not vanish identically, and
(2) all minors of order $> r$ vanish identically.

The rank of an $n \times n$ matrix $A(z)$ is the same for all z except for a finite set of points z_i, $i = 1, \ldots, p$ in the z plane at which the rank may decrease. These points are known as the <u>latent zeros</u> of the matrix $A(z)$

(see section 3.32). The maximum number of latent zeros for an $n \times n$ matrix $A(z)$ is $p = n\,m$ where m is the maximum number of zeros in any quasipolynomial element of $A(z)$. If $p < n\,m$ the matrix $A(z)$ is called degenerate.

A nonsquare matrix does not possess an inverse in the ordinary sense. However, it may have either a right or left inverse. Thus, if A is $m \times n$, A possesses a right inverse A^{-1}, such that $A\,A^{-1} = 1_m$ if and only if $m \leq n$ and $r(A) = m$.

An elementary polynomial matrix is a polynomial matrix possessing either a right or left polynomial inverse. A square matrix $A(z)$ is elementary if and only if its determinant is independent of z and non-zero.

$A(z)$ is analytic in a region of the z plane if all of its elements are analytic in this region.

The point z_0 is a pole of $A(z)$ if some element of $A(z)$ has a pole at $z = z_0$.

If z_0 is a pole of the rational matrix $A(z)$, each element of A may be expanded in partial fractions and after collecting all those terms having poles at z_0 there is obtained for $z_0 \neq \infty$

$$A(z) = (z - z_0)^{-k} A_k + (z - z_0)^{-k+1} A_{k-1} + \dots + (z - z_0)^{-1} A_1 + A_0(z) \qquad (3.3\text{-}1)$$

where $A_0(z_0)$ is finite, $A_k \neq 0$, and A_i, $1 \leq i \leq k$ are constant matrices. If $z_0 = \infty$, $(z - z_0)^{-i}$ is replaced by z^i, $1 \leq i \leq k$. All of $A_0(z)$, A_1, ..., A_k are uniquely defined by their construction from $A(z)$.

Definition 3.3-1. If $A(z)$ is given by equation 3.3-1, then k is the order of the pole of $A(z)$ at $z = z_0$.

Definition 3.3-2. A complex rational matrix is said to be reverse-hermitian if $A^*(z) = A(1/\bar{z})$ (the function A is symmetric with respect to the unit circle). Hence, on the unit circle, $z = e^{i\omega}$, $A^*(e^{i\omega}) = A(e^{i\omega})$ and $A(e^{i\omega})$ is hermitian in the ordinary sense. For real $A(z)$, this condition simplifies to $A'(1/z) = A(z)$ and will be called reverse-symmetrical. A real scalar function $f(z)$ satisfying $f(1/z) = f(z)$ is also called reverse-symmetrical.

It is most convenient for typographical reasons to let

$$A_*(z) \equiv A^*(1/\bar{z}).$$

This notation is used throughout the remainder of this paper. Notice that $A_{**}(z) = A(z)$, $(A\ B)_* = B_*\ A_*$.

Definition 3.3-3. A rational $m \times n$ matrix $A(z)$ is said to be reverse-unitary if

$$A(z)\ A_*(z) = 1_m .$$

A reverse-unitary matrix is also called all-pass.

Definition 3.3-4. A matrix $A(z)$ is said to be regular if it is analytic inside the unit circle $|z| < 1$. A matrix $A(z)$ is said to be Hurwitzian if it is analytic inside and on the unit circle $|z| \leq 1$.

3.32 Spectral Decomposition

The decomposition of polynomial matrices that is discussed in this section is very closely related to that of the Spectral Theorem of Linear Algebra (Hoffman and Kunze, 1961, pp. 275-6) which is stated for normal operators. Thus we will call the decomposition theorem the Spectral Theorem.

Before stating this theorem we shall investigate the properties of the latent zeros and vectors of a polynomial matrix. These properties will account for the principle restrictions placed upon the theorem.

3.321 Latent zeros and vectors

Let us consider the $n \times n$ square polynomial matrix

$$A(z) = A_0 + A_1 z + \ldots + A_m z^m$$

The latent zeros z_i of $A(z)$ are those values of $z = z_i$ $i = 1, \ldots, p$ ($p = nm$ if $A(z)$ is non-degenerate) for which $\text{Det } A(z) = 0$. Since the determinant has real coefficients, complex roots may only occur in conjugate pairs.

Frazer, et al., (pp. 61-65, 1947) prove the following properties concerning polynomial matrices at the zero positions z_ℓ :

(a) The matrix $A(z_\ell)$ is necessarily singular. When z_ℓ is an unrepeated root, $A(z_\ell)$ has rank $r(\ A(z_\ell)\ ) = n - 1$.

(b) When $A(z_\ell)$ has rank $r(\ A(z_\ell)\ ) = n - q$, at least q of the roots $z_1, z_2, \ldots, z_p$ are equal to z_ℓ .

(c) The matrix $A(z_\ell)$ does not necessarily have rank $n - q$ when z_ℓ is a root of multiplicity q .

(d) When $A(z_\ell)$ has rank $r(\ A(z_\ell)\ ) = n - 1$ the adjugate $\mathrm{Adj}\ A(z_\ell)$ has unit rank, $r(\mathrm{Adj}\ A(z_\ell)\ ) = 1$. Hence it is expressible as a product of the form

$$\mathrm{Adj}\ A(z_\ell) = \underline{u}_\ell\ \underline{v}_\ell'$$

where $\underline{u}_\ell$ and $\underline{v}_\ell$ are column vectors (called latent vectors) of length n and are constants appropriate to the selected zero z_ℓ . At least one element of each vector is non-zero.

Finally, if we let

$$D^m A(z_\ell) = \frac{d^m}{dz^m} A(z)\bigg|_{z = z_\ell}$$

we have

(e) When $A(z_\ell)$ has rank $r(\,A(z_\ell)\,) = n - q$, where $q > 1$, the adjugate matrix $\text{Adj}\ A(z)$ and its derivatives up to and including $D^{q-2}\ \text{Adj}\ A(z_\ell)$ are all null. However, the matrix $D^{q-1}\ \text{Adj}\ A(z_\ell)$ has rank q and is expressible as a product of the form

$$D^{q-1}\ \text{Adj}\ A(z_\ell) \;=\; \underline{u}_\ell\ \underline{b}_\ell$$

where $\underline{u}_\ell$ and $\underline{b}_\ell$ are $n \times q$ matrices. The columns of these matrices can then be used to form q pairs of latent vectors $\underline{u}_i$ and $\underline{v}_i$.

3.322 Spectral Theorem

It is frequently convenient to introduce the concept of 2-term operators which correspond to polynomial matrices of degree 1. If we examine one of these 2-term operators

$$I - \mathfrak{U}z\ ,$$

we see that it is closely related to the characteristic value problem that is usually formulated in terms of λ:

$$\mathfrak{U} - I\lambda .$$

Thus we may apply our existing knowledge of the characteristic zeros and vectors of constant matrices to the more general case of polynomial matrices. This approach is used in the spectral theorem.

Theorem 3.3-1 (Spectral). Let $A(z)$ be an $n \times n$ real polynomial matrix of rank n and degree m

$$A(z) = A_0 + A_1 z + \ldots + A_m z^m .$$

Then $A(z)$ may be represented as

$$\begin{aligned} A(z) &= G_0(z)\,(I - U_1 z)\ \ldots\ (I - U_\ell z) \\ &= G_0(z)\, G_1(z)\ \ldots\ G_\ell(z) \end{aligned} \qquad (3.3\text{-}2)$$

or as

$$\begin{aligned} A(z) &= (I - \mathfrak{B}_\ell z)\ \ldots\ (I - \mathfrak{B}_1 z)\, \tilde{G}_0(z) \\ &= \tilde{G}_\ell(z)\ \ldots\ \tilde{G}_1(z)\, \tilde{G}_0(z) \end{aligned} \qquad (3.3\text{-}3)$$

where $G_0(z)$ and $\tilde{G}_0(z)$ are elementary, if, for every zero z_i of multiplicity q, $r(\ A(z_i)\ ) = n - q_1$ where $q/q_1 \le \ell$.

Proof. (Claerbout (personal communication) has developed a similar factorization.)

First, consider equation 3.3-2. Since $|A\ B| = |A|\,|B|$, the latent roots z_i $i = 1, \ldots, p$ of $A(z)$ must be the union of the latent roots of $G_1, G_2, \ldots, G_\ell$. The n latent vectors of $A(z)$ and $G_\ell(z)$ are given by

$$\begin{aligned} \text{Adj } A(z_j) &= \text{Adj } (G_0\ G_1\ \ldots\ G_\ell) \\ &= \text{Adj } G_\ell(z_j)\ \text{Adj } (G_0\ G_1\ \ldots\ G_{\ell-1}) \\ &= \underline{u}_j\ \underline{b}'_j\ \text{Adj } (G_0\ G_1\ \ldots\ G_{\ell-1}) \quad j = 1, \ldots, n . \end{aligned} \qquad (3.3\text{-}4)$$

Thus the n latent vectors $\underline{u}_j$ are the same for $A(z_j)$ and $G_\ell(z_j)$. Therefore, if we determine a set of n zeros of $A(z)$ that have n independent latent vectors, we may recombine them by the well-known formula (Frazer, et al., pp. 66-68)

$$U_\ell = \left[(\underline{u}_1) \cdot\cdot (\underline{u}_n) \right] \begin{bmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{bmatrix}^{-1} \left[(\underline{u}_1) \cdot\cdot (\underline{u}_n) \right]^{-1}$$

$$= U_\ell Z_\ell^{-1} U_\ell^{-1} \qquad (3.3\text{-}5)$$

and

$$G_\ell = I - U_\ell z \ .$$

The 2-term polynomial $G_\ell(z)$ is a factor of $A(z)$. For, if we substitute the matrix U_ℓ into the polynomial $A(z)$

$$A(U_\ell) = A_0 + A_1 U_\ell + \ldots + A_m U_\ell^m \qquad (3.3\text{-}6)$$

$$= A_0 + A_1 U_\ell Z_\ell^{-1} U_\ell^{-1} + \ldots + A_m U_\ell (Z_\ell^{-1})^m U_\ell^{-1}$$

we see that $A(U_\ell) \equiv 0$ identically. Thus $G_\ell(z) = I - U_\ell z$ will right-divide $A(z)$ with a remainder of zero (Frazer, et al., 1947, p. 60) and therefore is a factor of $A(z)$.

(Q.E.D.)

The factorization is continued then by removing $G_\ell(z)$ by right division, determining n more independent latent vectors and constructing a second 2-term wavelet

$\mathfrak{G}_{\ell-1}$. This process is repeated until $\mathfrak{G}_0(z)$, an elementary matrix, remains.

Alternately, we may factor on the basis of the $\underline{\mathfrak{v}}_i$ latent vectors. Thus, let us consider equation 3.3-3. Here again the latent zeros of $A(z)$ and the factors $\tilde{\mathfrak{G}}_i(z)$ are the same. The latent vectors of $A'(z)$ are given by

$$\begin{aligned} \text{Adj}\ A'(z_j) &= \text{Adj}\ (\tilde{\mathfrak{G}}_\ell \ldots \tilde{\mathfrak{G}}_1\ \tilde{\mathfrak{G}}_0)' \\ &= \text{Adj}\ \tilde{\mathfrak{G}}_\ell'(z_j)\ \text{Adj}\ (\tilde{\mathfrak{G}}_{\ell-1} \ldots \tilde{\mathfrak{G}}_1\ \tilde{\mathfrak{G}}_0)' \\ &= \underline{\mathfrak{v}}_j\ \underline{\mathfrak{u}}_j'\ \ \text{Adj}\ (\tilde{\mathfrak{G}}_{\ell-1} \ldots \tilde{\mathfrak{G}}_1\ \tilde{\mathfrak{G}}_0)' \end{aligned} \qquad (3.3\text{-}7)$$

Thus the n latent vectors of $A'(z)$ corresponding to z_i, $i = 1, 2, \ldots, n$ are the same as the n latent vectors of $\tilde{\mathfrak{G}}_\ell(z)$. As before, if we choose n zeros such that the associated latent vectors are independent then they may be recombined as

$$\mathfrak{B}_\ell' = \begin{bmatrix} (\underline{\mathfrak{v}}_1) \cdots (\underline{\mathfrak{v}}_n) \end{bmatrix} \begin{bmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{bmatrix}^{1} \begin{bmatrix} (\underline{\mathfrak{v}}_1) \cdots (\underline{\mathfrak{v}}_n) \end{bmatrix}^{-1} \qquad (3.3\text{-}8)$$

and used to remove $\tilde{\mathfrak{G}}_\ell(z)$ from $A(z)$ by left division.

If the zeros and vectors are independent then the choice of which n zeros to associate with each 2-term

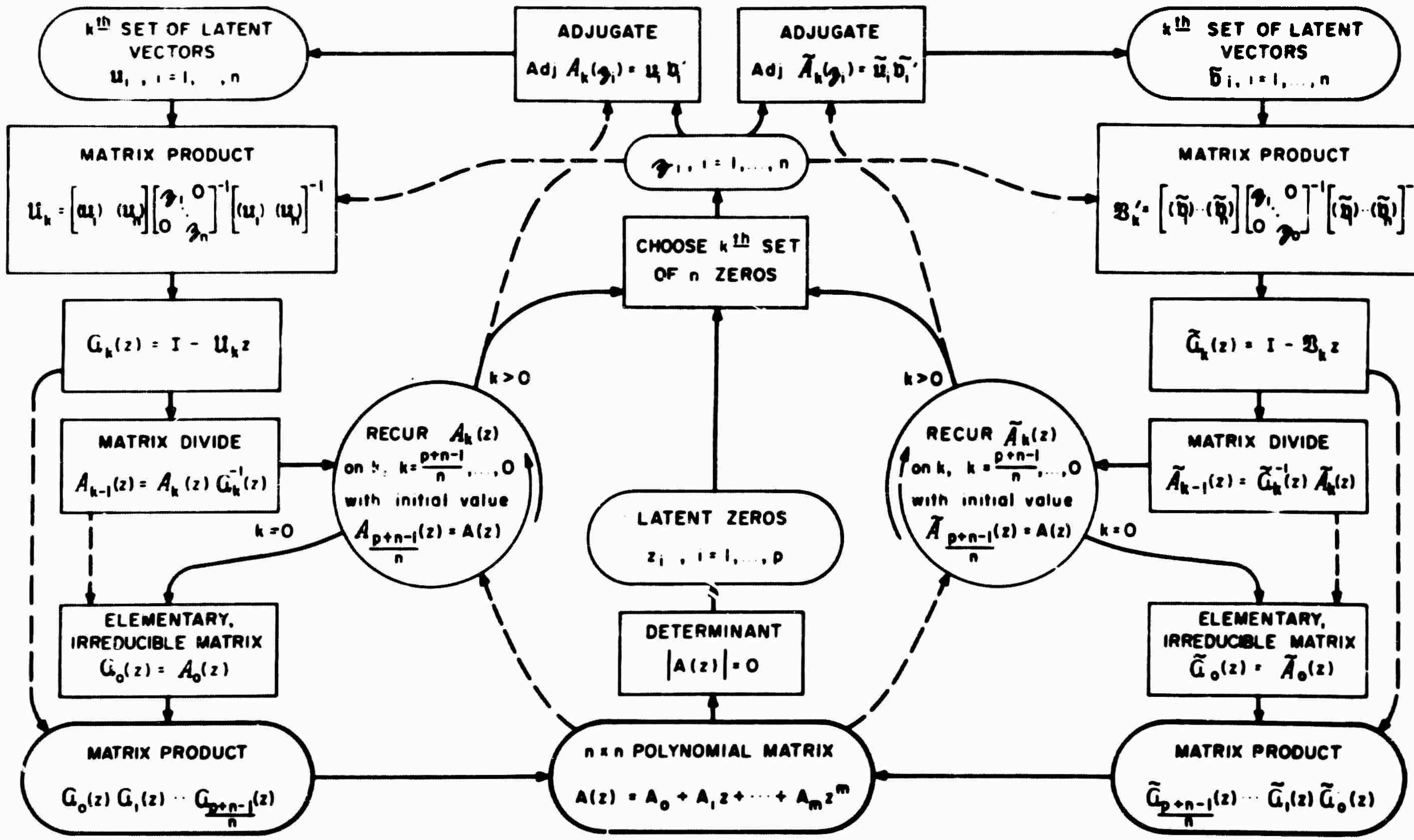


Figure 3.3 - 1: Spectral decomposition of a full rank polynomial matrix according to Theorem 3.3-1.

factor is arbitrary. Altogether, there may be $(nm)!/(n!)^m$ different factorizations. Once the choice is made, of course, the order of factorization must be preserved since multiplication is not commutative, in general. In some instances the choice of zeros must be made under certain restrictions so that the full factorization may be realized. As indicated above, this restriction consists of choosing the zeros so that the latent vectors $\underline{u}_i$ or $\underline{b}_i$ are independent for each set of n vectors. Such a choice may always be made if for every zero z_i of multiplicity q_i, $r(\ A(z_i)\ ) = n - q_i$ where $q/q_i \leq \ell$. Q.E.D.

The details of this factorization are illustrated in Figure 3.3-1. The right half shows the decomposition in terms of m_i and the left half shows the decomposition in terms of u_i .

In general it is not necessary to go through the intermediate steps of forming 2-term factors to construct a polynomial matrix from its latent roots and vectors. This more direct approach is the subject of the Spectral Corollary.

<u>Corollary 3.3-1</u> (<u>Spectral</u>). Let $A(z)$ be an $n \times n$ real polynomial matrix of rank n and degree m

$$A(z) = A_0 + A_1 z + \dots + A_m z^m .$$

Then $A(z)$ is completely described by

a) the elementary matrix $G_0(z)$
b) the latent zeros z_i $i = 1, 2, \ldots, p$, and
c) the p $(p \leq m\,n)$ corresponding latent vectors $\underline{u}_i$ or $\underline{v}_i$ if for every root z_ℓ of multiplicity q, $r(\,A(z_\ell)\,) = n - q$.

Notice that this corollary is not so general as the spectral theorem in its treatment of multiple zeros with identical latent vectors.

Proof (Suggested by Quenouille, 1957, pp. 5-25) Let us first consider the case for which $A(z)$ is non-degenerate, i.e. that $G_0(z)$ is a constant non-singular matrix and the degree of the determinant, Det $A(z)$, is $p = m\,n$.

Consider the factored form of $A(z)$

$$\begin{aligned} A(z) &= A_0\,(I - U_1\,z - \ldots - U_m\,z^m) \\ &= A_0\,G(z) \end{aligned} \tag{3.3-9}$$

where $U_i = -A_0^{-1}\,A_i$. Then, if we inquire about the solutions to the equation

$$\underline{u} - U_1\,\underline{u}\,z - \ldots - U_m\,\underline{u}\,z^m = 0, \tag{3.3-10}$$

we see that solutions are possible only if the determinant

$$I - U_1\,z - \ldots - U_m\,z^m = 0 \tag{3.3-11}$$

is zero. It is zero at the p locations z_i $i = 1, \ldots, p$ which are the latent zeros of $G(z)$ and consequently of

$A(z)$. Therefore the solution vectors of equation 3.3-10 are the latent vectors of $G(z)$. Since

$$\begin{aligned} \text{Adj } A(z_\ell) &= \text{Adj } (A_0 \, G(z_\ell)) \\ &= \text{Adj } G(z_\ell) \text{ Adj } A_0 \\ &= \underline{u}_\ell \, \underline{v}_\ell' \text{ Adj } A_0 \, , \end{aligned} \tag{3.3-12}$$

the solution vectors are also latent vectors of $A(z)$.
Now join the latent vectors and latent zeros into the modal matrix

$$\begin{array}{cc} \xleftarrow{\quad p = nm \quad} & \\ u = \begin{bmatrix} (u_1) & (u_2) & \dots & (u_p) \end{bmatrix} & \updownarrow n \end{array} \, , \tag{3.3-13}$$

and the zero matrix

$$Z = \text{diag} \begin{bmatrix} z_1, & z_2, & \dots & z_p \end{bmatrix}$$

and substitute these matrices into equation 3.3-10 for $\underline{u}$ and z:

$$U_1 \, u \, Z + \dots + U_m \, u \, Z^m = u \tag{3.3-14}$$

Clearly we can solve this set of simultaneous equations for $U_1, U_2, \dots U_m$ if the columns of $u \, Z$ are independent. This does not occur when a zero of multiplicity q has fewer than q independent latent vectors, that is, if a multiple root has identical latent vectors.

Alternately, we may choose to use the vectors $\underline{v}_\ell$ for the reconstruction. For this case we would factor $A(z)$ as

$$A(z) = (I - V_1 z - \ldots - V_m z^m) A_0$$

$$= \mathcal{G}(z) A_0 \qquad (3.3\text{-}15)$$

Where $V_i = - A_i A_0^{-1}$. Since the latent zeros of $\mathcal{G}(z)$ are the same as for $A(z)$, and since the latent vectors of A' are

$$\text{Adj } A'(z_\ell) = \text{Adj } (\mathcal{G}(z_\ell) A_0)'$$

$$= \text{Adj } \mathcal{G}'(z_\ell) \text{ Adj } A_0'$$

$$= \underline{v}\ \underline{u}' \text{ Adj } A_0' \qquad (3.3\text{-}16)$$

we may reconstruct the matrix $\mathcal{G}(z)$ by the simultaneous equations

$$V_1' \mathcal{V} Z + \ldots + V_m' \mathcal{V} Z^m = \mathcal{V} \qquad (3.3\text{-}17)$$

where

$$\mathcal{V} = \begin{bmatrix} (v_1) & (v_2) & \ldots & (v_p) \end{bmatrix} . \qquad (3.3\text{-}18)$$

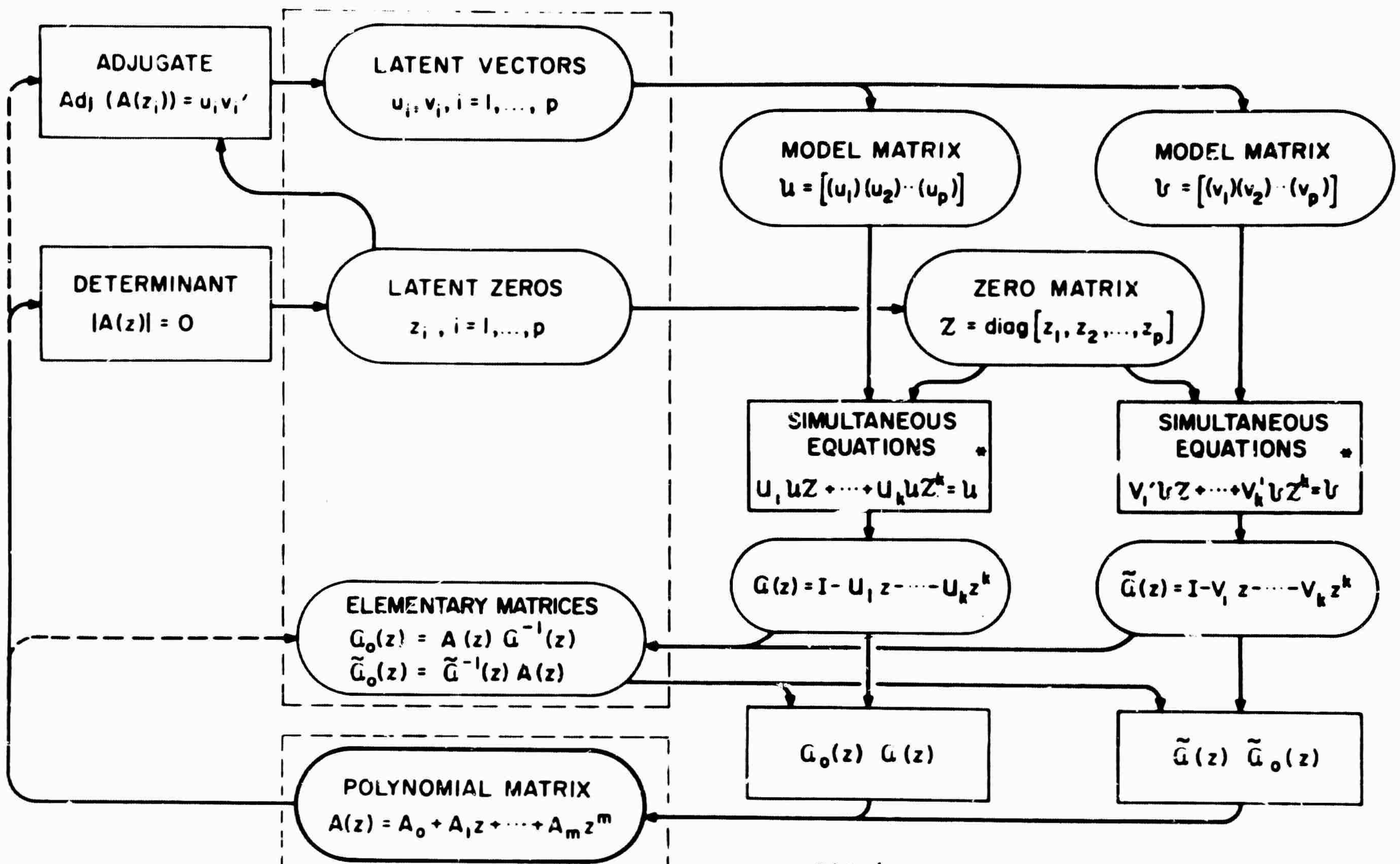
The same restraints hold here as held for the $\underline{u}$ vectors.

If $G_0(z)$, the elementary matrix multiplier, is not constant, then the number of zeros is $p < mn$. We can, however, proceed as above to find $G(z)$ and then determine $G_0(z)$ by the formula

$$G_0(z) = A(z) G^{-1}(z)$$

for the factorization in terms of the $\underline{u}$ vectors, or by the formula

$$\mathcal{G}_0(z) = \mathcal{G}^{-1}(z) A(z)$$



* $k = \frac{p+n-1}{n}$ where $A(z)$ is $n \times n$.

Figure 3.3 - 2: Spectral decomposition of a full-rank polynomial matrix according to Corollary 3.3-1.

for the factorization in terms of the $\underline{v}$ vectors.

Q.E.D.

These factorizations are illustrated in Figure 3.3-2. The boxes enclosed in dotted lines represent completely equivalent representations of the matrix.

<u>Example 3.3-1</u>. (after Claerbout, personal communication)

Consider the polynomial matrix

$$A(zC) = \begin{bmatrix} 2 - 20z + 50z^2 & -1 + 9z - 20z^2 \\ 14z - 58z^2 & 1 - 11z + 28z^2 \end{bmatrix} .$$

The determinant is

$$\begin{aligned} A(z) &= 2 - 28z + 142z^2 - 308z^3 + 240z^4 \\ &= 2\,(1 - 2z)\,(1 - 3z)\,(1 - 4z)\,(1 - 5z) \end{aligned} \tag{3.3-19}$$

The adjugate matrix is

$$\text{Adj } a(z) = \begin{bmatrix} 1 - 11z + 28z^2 & 1 - 9z + 20z^2 \\ -14z + 58z^2 & 2 - 20z + 50z^2 \end{bmatrix} \tag{3.3-20}$$

Substituting $z_1 = 0.5$ into Adj $A(z)$ gives the latent vectors $\underline{u}_1$ and $\underline{v}_1$

$$\text{Adj } A(0.5) = \begin{bmatrix} 2.5 & 1.5 \\ 7.5 & 4.5 \end{bmatrix} = 0.5 \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 5 & 2 \end{bmatrix}$$

Likewise, if we substitute for the other roots we will find all the latent vectors:

Zero	Latent vector $\underline{u}$	Latent vector $\underline{v}$
1/2	(1, 3)'	(5, 3)'
1/3	(1, 4)'	(2, 1)'
1/4	(0, 1)'	(1, 1)'
1/5	(1, 6)'	(1, 0)'

Now let us follow the reconstructions considered in the Spectral Corollary.

First, we may pre-divide by A_0 to find

$$A(z) = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \left\langle I - \begin{bmatrix} 3 & 1 \\ -14 & 11 \end{bmatrix} z - \begin{bmatrix} 4 & -4 \\ 58 & -28 \end{bmatrix} z^2 \right\rangle$$

$$= A_0 \left\langle I - U_1 z - U_2 z^2 \right\rangle \qquad (3.3\text{-}21)$$

Set up the matrices U and Z and substitute into the the transpose of equation 3.3-14

$$\begin{bmatrix} 1/2 & 3/2 & 1/4 & 3/4 \\ 1/3 & 4/3 & 1/9 & 4/9 \\ 0 & 1/4 & 0 & 1/16 \\ 1/5 & 6/5 & 1/25 & 6/25 \end{bmatrix} \begin{bmatrix} U_1' \\ \hline U_2' \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 4 \\ 0 & 1 \\ 1 & 6 \end{bmatrix} \qquad (3.3\text{-}22)$$

and obtain the values of U_1 and U_2 given above in equation 3.3-21.

Similarly, we may post-divide $A(z)$ by A_0 to find

$$A(z) = \left\langle I - \begin{bmatrix} 10 & 1 \\ -7 & 4 \end{bmatrix} z - \begin{bmatrix} -25 & -5 \\ 29 & 1 \end{bmatrix} z^2 \right\rangle \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix}$$

$$= \left\langle I - V_1 z - V_2 z^2 \right\rangle \; A_0 \qquad (3.3\text{-}23)$$

We set up the matrices U and Z and substitute into equation 3.3-17

$$\begin{bmatrix} 5/2 & 3/2 & 5/4 & 3/4 \\ 2/3 & 1/3 & 2/9 & 1/9 \\ 1/9 & 1/4 & 1/16 & 1/16 \\ 1/5 & 0 & 1/25 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ \hline V_2 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad (3.3\text{-}24)$$

to obtain the values of V_1 and V_2 given above in equation 3.3-23.

Equations 3.3-21 or 3.3-23 may also be reconstructed using the algorithm of the Spectral Theorem. We will illustrate the process for only the u latent vectors. Recall that this factorization is in the form

$$A(z) = A_0 \,(I - U_1 z)\,(I - U_2 z)$$

$$= A_0 \, G_1(z) \, G_2(z) \quad .$$

We arbitrarily choose the zeros $z = 1/4,\ 1/5$ to obtain

$$U_2 = \begin{bmatrix} 0 & 1 \\ 1 & 6 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} -6 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 0 \\ 6 & 4 \end{bmatrix}$$

and

$$G_2 = \left\langle I - \begin{bmatrix} 5 & 0 \\ 6 & 4 \end{bmatrix} z \right\rangle$$

If we now post divide $G(z)$ (from equation 3.3-21) we find

$$G_1 = \left\langle I - \begin{bmatrix} -2 & 1 \\ -20 & 7 \end{bmatrix} z \right\rangle$$

which has latent vectors

zero	latent vector $\underline{u}$
1/2	(1, -4)'
1/3	(1, -5)'

Of course, this is not the only possible factorization. Altogether, there are

$$\frac{(nm)!}{(n!)^m} = \frac{24}{4} = 6$$

different representations. Using the method illustrated above we find that

$$\left\langle \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 1 \\ -14 & 11 \end{bmatrix} z - \begin{bmatrix} 4 & -4 \\ 53 & -28 \end{bmatrix} z^2 \right\rangle =$$

$$= \left\langle I - \begin{bmatrix} -2 & 1 \\ -20 & 7 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} 5 & 0 \\ 6 & 4 \end{bmatrix} z \right\rangle$$

$$= \left\langle I - \begin{bmatrix} 4 & 0 \\ 10 & 2 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} -1 & 1 \\ -24 & 9 \end{bmatrix} z \right\rangle$$

$$= \left\langle I - \begin{bmatrix} 0 & 1 \\ -10 & 7 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} 3 & 0 \\ -4 & 4 \end{bmatrix} z \right\rangle$$

$$= \left\langle I - \begin{bmatrix} 4 & 0 \\ 4 & 3 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} -1 & 1 \\ -18 & 8 \end{bmatrix} z \right\rangle$$

$$= \left\langle I - \begin{bmatrix} 1 & 1 \\ -8 & 7 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} 2 & 0 \\ -6 & 4 \end{bmatrix} z \right\rangle$$

$$= \left\langle I - \begin{bmatrix} 4 & 0 \\ -2 & 5 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} -1 & 1 \\ -12 & 6 \end{bmatrix} z \right\rangle .$$

Example 3.3-2. (Multiple roots)

Consider the polynomial matrix

$$A(z) = \begin{bmatrix} 3 - 6z + 3z^2 & 1 - 4z + z^2 \\ 1 - 2z + z^2 & 2 - 8z + 7z^2 \end{bmatrix}$$

Using the standard factorization techniques we find that this has latent zeros and vectors

zero	latent vector $\underline{u}$
1/2	(1, 1)
1/2	(1, 1)
1	(1, 0)
1	(1, 0)

If we set up the reconstruction equation 3.3-14

$$\begin{bmatrix} 1/2 & 1/2 & 1/4 & 1/4 \\ 1/2 & 1/2 & 1/4 & 1/4 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} U_1' \\ \hline U_2' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

we find that the left hand side is singular. However, if we use the algorithm outlined for the Spectral Theorem and use the zeros 1/2 and 1 for each of the 2-term factors, we find

$$A(z) = \left\langle I - \begin{bmatrix} -1 & -1 \\ 0 & -2 \end{bmatrix} z \right\rangle \left\langle I - \begin{bmatrix} -1 & -1 \\ 0 & -2 \end{bmatrix} z \right\rangle .$$

Thus, this approach is slightly more general.

3.33 Invertibility

<u>Definition 3.3-5</u>. A one-sided matrix-valued wavelet $A(z)$ is said to be <u>invertible</u> if there exists a <u>one-sided</u> left- or right-inverse wavelet $A^{-1}(z)$.

Let us consider only square matrix-valued wavelets. The inverse of such a wavelet is given by

$$A^{-1}(z) = \frac{\text{Adj } A(z)}{\text{Det } A(z)} .$$

The condition for invertibility is that the determinant of $A(z)$ has a stable inverse. This condition when applied to the determinant of a finite wavelet is exactly the same as that applied to the scalar wavelet. That is, the zeros of the determinant of $A(z)$ must be outside the unit circle in the z plane (see section 3.22).

3.34 Smith-McMillan Canonical Form

This canonical form for rational matrices involves the terms contained in the determinant and the rank of the matrix. It is the subject of the classical Smith-McMillan Theorem (Gantmacher, p. 134, 1959 and McMillan, p. 581, 1952).

<u>Theorem 3.3-2</u> (<u>Smith-McMillan</u>). Let $A(z)$ be an $m \times n$ complex rational matrix of normal rank r. Then there exist two elementary polynomial matrices $C(z)$ and $F(z)$ of orders $m \times r$ and $r \times n$, respectively, such that

$$A(z) = C(z)\,\mathrm{diag}\left[\frac{\varepsilon_1(z)}{\psi_1(z)}, \frac{\varepsilon_2(z)}{\psi_2(z)}, \ldots, \frac{\varepsilon_r(z)}{\psi_r(z)}\right] F(z)$$

$$= C\,D\,F \qquad (3.3\text{-}25)$$

where

a) $\varepsilon_k(z)$ and $\psi_k(z)$ are relatively prime polynomials with unit leading coefficients, $1 \leq k \leq r$;

b) Each $\varphi_k(z)$ divides $\varphi_{k+1}(z)$, $1 \leq k \leq r - 1$, and each $\psi_\ell(z)$ divides $\psi_{\ell-1}(z)$, $2 \leq \ell \leq r$;

c) The diagonal matrix $D(z)$ appearing in equation 3.3-25 is, subject to a) and b), uniquely determined by $A(z)$. It is, in fact canonic;

d) If $A(z)$ is real, the φ's, ψ's, $C(z)$, and $F(z)$ may also be chosen real;

e) The finite point $z = z_0$ is a pole of $A(z)$ of order k if and only if it is a zero of $\psi_1(z)$ of order k.

f) The order of $z = \infty$ as a pole of $A(z)$ is the same as the order of $1/z = 0$ as a pole of $A(1/z)$.

A rational matrix is said to be Smith-McMillan <u>canonic</u> if it is square, non-singular and diagonal with properties a) and b) listed above. The rational functions φ_1/ψ_1, φ_2/ψ_2, ..., φ_r/ψ_r are generalized <u>invariant factors</u> of $A(z)$. Clearly, since C and D are elementary, $\text{Det } A(z) = \text{Det } D(z)$. A set of polynomials are said to be relatively prime if their largest common denominator is 1.

<u>Frazer</u>, <u>et al</u>., (pp. 87-92, 1947) or <u>Gantmacher</u> (pp. 134-139, 1959) show in detail the technique for the reduction of a matrix to canonical form. The method employed is reminiscent of the elimination methods for in-

verting a matrix. We will illustrate it with a polynomial matrix example.

<u>Example 3.3-3</u>. Reduction of a polynomial matrix to canonical form.

Let us consider the matrix

$$A(z) = \begin{bmatrix} 2+z & z \\ 1 & 6+z \end{bmatrix}$$

a) Put a one in the first diagonal position. This is accomplished by multiplying A on the left by

$$S_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

to obtain

$$A_1 = \begin{bmatrix} 1 & 6+z \\ 2+z & z \end{bmatrix} .$$

b) Reduce the other terms in the first row and column to zero. This accomplished by multiplying A_1 on the left by

$$S_2 = \begin{bmatrix} 1 & 0 \\ -(2+z) & 1 \end{bmatrix}$$

to obtain

$$A_2 = \begin{bmatrix} 1 & 6+z \\ 0 & -z^2 - 7z - 12 \end{bmatrix}$$

and multiplying A_2 on the right by

BLANK PAGE

$$O_1 = \begin{bmatrix} 1 & -(6+z) \\ 0 & -1 \end{bmatrix}$$

to obtain

$$A_3 = \begin{bmatrix} 1 & 0 \\ 0 & z^2 + 7z + 12 \end{bmatrix}$$

c) Now, if we let

$$C\ (z) = (S_2\ S_1)^{-1} = \begin{bmatrix} 2+z & 1 \\ 1 & 0 \end{bmatrix}$$

$$F\ (z) = O_1^{-1} = \begin{bmatrix} 1 & -6-z \\ 0 & -1 \end{bmatrix}$$

we obtain the Smith-McMillan canonical form

$$A(z) = C\ D\ F$$

where

$$D = A_3 = \begin{bmatrix} 1 & 0 \\ 0 & (z+3)(z+4) \end{bmatrix}$$

Clearly C and F are elementary matrices and D is canonic. That is, 1 divides $(z + 3)(z + 4)$ and $\mathrm{Det}\ D(z) = \mathrm{Det}\ A(z)$.

3.35 Robinson Canonical Form and All-Pass Systems

Theorem 3.3-3 (Robinson Canonical Form). Any full-rank wavelet $A(z)$ can always be uniquely represented by the Robinson canonical form

$$A(z) = A_0(z)\, P(z)$$

where $A_0(z)$ is invertible and $P(z)$ is regular reverse-unitary (i.e. all-pass). More generally, if $A(z)$ is an $n \times m$ matrix and has rank $r \leq n, m$, then its canonical form becomes

$$A(z) = A_0(z) \left[1_r \;\middle|\; 0_{r,m-r} \right] P(z)$$

where $P(z)$ is a regular reverse-unitary $m \times m$ matrix.

Matrix reverse-unitary (all-pass) systems have similar properties to scalar all pass systems. We shall state several theorems concerning them now (Robinson, 1962, and Youla, 1962).

Theorem 3.3-4. An $n \times m$ matrix $P(z)$ of rank r is analytic in the entire z plane together with its inverse (either left, right, or both) if and only if it is an elementary polynomial matrix.

Proof: The "if" part is obvious. According to the Smith-McMillan Theorem (3.3-2), the analyticity of $P(z)$ for all z implies that all of the denominator terms, ψ_i, of the canonical form be constant. Now the existence of a left or right inverse implies that either $n = r$ or $m = r$, respectively. The canonic form for $P^{-1}(z)$ is

$$\operatorname{diag}\left[\frac{\psi_r(z)}{\varphi_r(z)}, \frac{\psi_r(z)}{\varphi_{r-1}(z)}, \dots \frac{\psi_1(z)}{\varphi_1(z)}\right] .$$

The analyticity of $P^{-1}(z)$ in the entire plane implies that φ_i, $i = 1, \dots, r$ is constant. Therefore $P(z)$ is the product of three elementary polynomial matrices, of rank r . Q.E.D.

Theorem 3.3-5. A reverse-unitary rational matrix is bounded on the unit circle.

Proof: Suppose $P(z)$ is $m \times n$ and $P(z)\, P_*(z) = 1_n$. Thus $P(e^{i\omega})\, P^*(e^{i\omega}) = 1_n$, and, writing out the diagonal elements in expanded form,

$$\sum_{r=1}^{m} \left|(P)_{rk}(e^{i\omega})\right|^2 = 1 \qquad (k = 1, 2, \dots, n) .$$

$$\therefore \left|(P)_{rk}(e^{i\omega})\right| \leq 1 \qquad (r = 1, 2, \dots, m; \; k = 1, 2, \dots, n),$$

for all ω . Q.E.D.

Theorem 3.3-6. The only regular reverse-unitary matrices $P(z)$ with regular inverses are constant unitary matrices (trivial all-pass systems). If $P(z)$ is real it is real-orthogonal.

Proof: Suppose $P(z)\, P_*(z) = 1_n$, say, where $P(z)$ is a regular $n \times m$ reverse-unitary matrix. The analyticity of its right inverse inside the unit circle implies that of $\overline{P}(1/\bar{z})$ in the same region and therefore that of $\overline{P}(\bar{z})$ outside the unit circle including infinity. Now the poles of $\overline{P}(\bar{z})$ are the complex conjugates of those of $P(z)$. Hence $P(z)$ is analytic in the entire z plane and bounded at infinity. By Liouville's Theorem it must be a constant unitary matrix. If $P(z)$ is real it must be real orthogonal by definition. Q.E.D.

3.36 Delay

The delay of a one-sided matrix-valued wavelet A_i is a measure of how the operator redistributes the energy of an input process. It is defined in terms of the partial energy

$$\mathcal{E}_i = \sum_{j=0}^{i} \operatorname{tr}(A_j A_j') .$$

The following theorem is a discrete analog of a theorem given by <u>Robinson</u> (pp.83-88, 1962). Since his proof is rather long and involved, it will not be repeated here.

<u>Theorem</u> <u>3.3-7</u> (<u>Minimum</u> <u>Delay</u>). The delays of the set of wavelets $A_i(z)$ which have the same invertible Robinson canonical form $A_0(z)$ are greater than or equal to the delay of $A_0(z)$. Equality holds if and only if the all-

pass system $P(z)$ is trivial. That is, the partial energies obey the relation

$$\sum_{j=0}^{k} \operatorname{tr} (A_1)_j (A_1)_j^* \leq \sum_{j=0}^{k} \operatorname{tr} (A_0)_j (A_0)_j^* \qquad \text{for all } k$$

where j is the time index.

3.37 Phase

As in the scalar case, the Fourier transform of the operator $A(z)$ is determined by restricting our attention to $z = e^{-i\omega}$. We may proceed to express each polynomial element of the matrix in terms of an amplitude characteristic and a phase characteristic. The question then arises whether there are any simple measures of this phase matrix, other than the determinant of the polynomial matrix, which would correspond to invertibility. That is, can we formulate a minimum phase theorem for matrix-valued wavelets?

An empirical investigation was made of this question which gave negative answers for all measures tried. These measures included 1) the trace of the phase-lag matrix, 2) the phase-lag of the trace of the polynomial matrix, and 3) the norm of the phase-lag matrix as a function of ω. In every case counter-examples could be found for which the wavelet was invertible but the measure tried did not give a minimum.

Another approach which might prove more successful would be to define a matrix amplitude characteristic $\mathcal{a}(\omega)$ and a matrix phase characteristic $\Phi(\omega)$ such that

$$A(e^{-i\omega}) = \mathcal{a}(\omega)\, e^{-i\Phi(\omega)}$$

This has not been investigated. However, should it prove to have a minimum phase property associated with invertibility, this measure would have limited application because of the difficulty of computation and cognition of such characteristics.

3.4 Multi-Dimensional Wavelets

Our treatment of multi-dimensional wavelets will be brief on two accounts. First, there is no general factorization available for multi-dimensional polynomials; and second, in almost all problems with which we are concerned the multi-dimensional process can be mapped into an equivalent higher ordered one-dimensional matrix-valued process.

The absence of any factorization can be illustrated by attempting to factor the polynomial

$$\begin{aligned} a(x,y) = \; & a_{00} + a_{01}\, x + a_{02}\, x^2 \\ & + a_{10}\, y + a_{11}\, xy + a_{12}\, x^2y \\ & + a_{20}\, y^2 + a_{21}\, xy^2 + a_{22}\, x^2y^2 \quad . \end{aligned}$$

If this were factorable, we should be able to find two polynomials,

$$b(x,y) = b_{00} + b_{01}x + b_{10}y + b_{11}xy \quad \text{and} \quad c(x,y) = c_{00} + c_{01}x + c_{10}y + c_{11}xy$$

such that $b\,c = a$. But a has 9 degrees of freedom and b and c combined have only 8. Thus, unless there are special relationships between the elements a_{ij}, $a(x,y)$ is unfactorable.

3.41 Invertibility

<u>Definition 3.4-1</u>. A scalar multi-dimensional wavelet $a(\underline{z})$ is said to be invertible about its origin $\underline{z} = 0$ if there exists a wavelet $a^{-1}(\underline{z})$ such that $a(\underline{z})\, a^{-1}(\underline{z}) = 1$.

The condition for making an expansion about $\underline{z} = 0$ of $a^{-1}(\underline{z})$ is that $a(\underline{z})$ does not go to zero inside the unit hypercircle

$$|z\, z_1 \ldots z_n| = 1 .$$

3.42 Phase

Perhaps the simplest measure of invertibility involves the phase-lag of the wavelet. The multi-dimensional Fourier transform is found by restricting $\underline{z}$ to the unit hypercircle. Thus

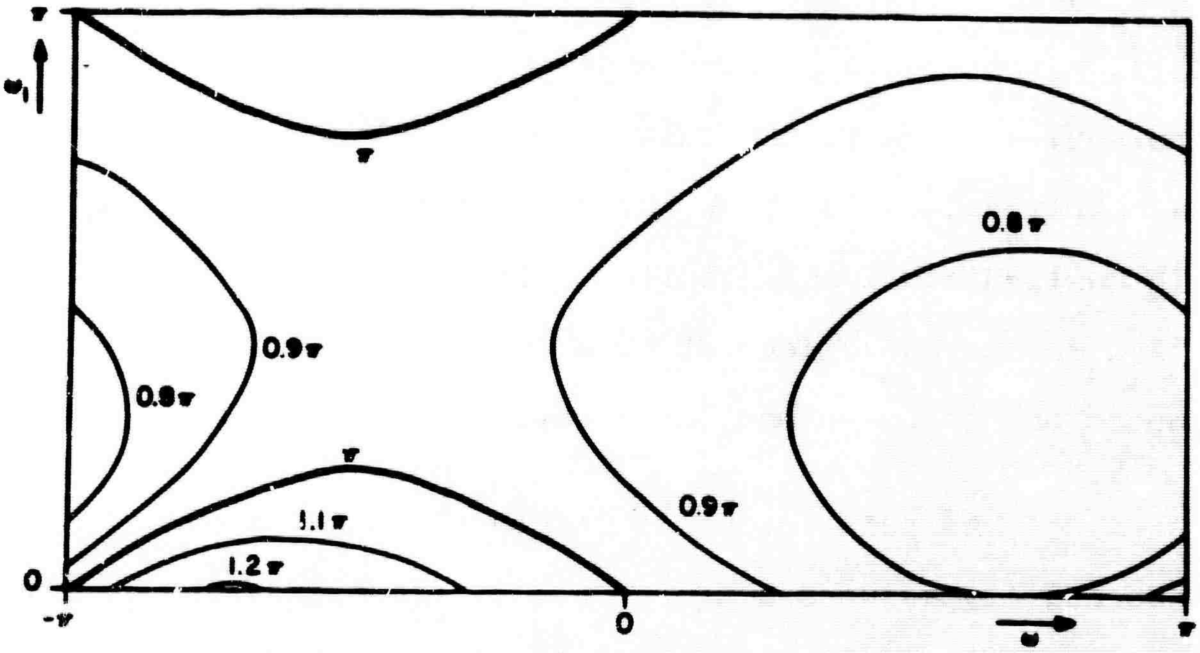
$$a(z, z_1, \ldots, z_n) \rightarrow a(e^{-i\omega}, e^{-i\omega_1}, \ldots, e^{-i\omega_n})$$

and by finding the polar representation of this we can find a multi-dimensional phase-lag characteristic, $-\varpi(\omega, \omega_1, \ldots, \omega_n)$.

<u>Theorem 3.4-1</u> (<u>Minimum Phase</u>). If the phase-lag characteristic $-\varpi(\underline{\omega})$ for the wavelet $a(\underline{z})$ is the same for all $\omega_i = \pi$ or 0 $i = 1, \ldots, n$, then the wavelet $a(\underline{z})$ is invertible.

Figure 3.4-1 illustrates two phase-lag plots for two-dimensional wavelets. The variable z_1 corresponds to the phase-variable ω_1 . Notice that when a hyper-surface $a(\underline{z}) = 0$ cuts across the unit hypercircle, the phase is discontinuous along the intersection. This is analogous to the case of a zero on the unit circle for one-dimensional scalar wavelets.

Invertible wavelet: $1.0 + 0.3\ z + 0.5\ z_1$



Non-invertible wavelet: $1.0 + 0.5\ z + 0.7\ z_1$

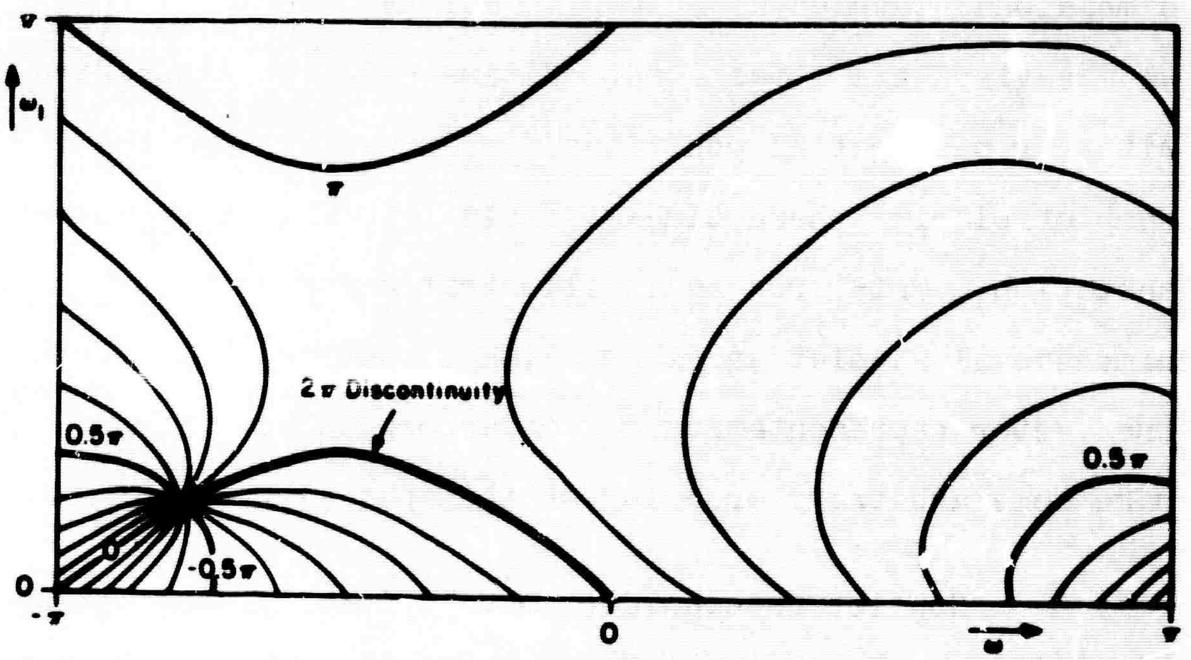


Figure 3.4 - 1: Two dimensional phase-lag characteristics for two wavelets. The phase-lag is discontinuous across the zero hyper-surface in the second case.

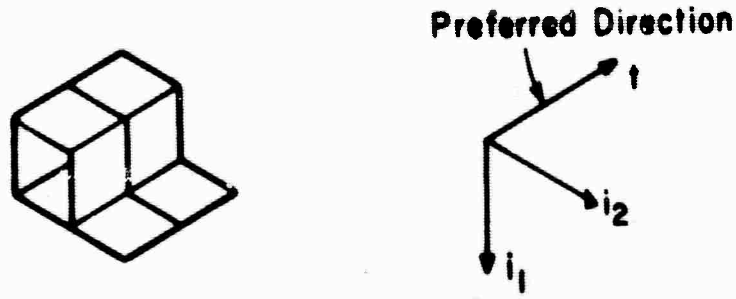
3.43 Mapping into one-dimensional representation

Much of the algebra of multi-dimensional operators and autocorrelations represents a special case of the general matrix-valued, one-dimensional algebra. For this reason, we seek to map multi-dimensional convolution into a matrix-valued notation rather than to develop the algebra in multi-dimensional notation. Thus, this section will give an extensive account of a mapping from multi-dimensional notations to one-dimensional notation.

As pointed out in Chapter 2, this mapping necessarily assumes a preferred, or time-like, direction. It is this dimension that remains undisturbed after the mapping. Thus, rather than thinking of a multi-dimensional wavelet as a lump in multi-dimensional space, we may visualize it as a set of time-wavelets associated with various spatial positions. Then we take the logical step of placing these time-wavelets into a vector representation. This process is illustrated for a three-dimensional wavelet in Figure 3.4-2. Notice that before the vector representation can be accomplished, we must make some arbitrary ordering of the spatial points.

Now let us consider convolution. If the operator wavelet, a, is mapped into a vector of time wavelets, $\underline{a}$, and the output, y, is mapped into a similar vector, $\underline{y}$, then the input, x, must be mapped into a matrix, X .

a_{t,i_1,i_2} :



z-transform in
Preferred - Direction

↓

$a_{i_1,i_2}(z)$: $a_{0,0}(z)$ $a_{1,0}(z)$

$a_{0,0}(z)$ $a_{1,1}(z)$ $a_{1,2}(z)$

↓

Put the indices of $a_{i_1 i_2}(z)$
in some arbitrary monotonic order

↓

$a_j(z)$: $a_1(z)$ $a_3(z)$

$a_2(z)$ $a_4(z)$ $a_5(z)$

↓

Spread these time-wavelets into a
vector representation

↓

$$\underline{a}(z) = \left[a_1(z),\ a_2(z),\ a_3(z),\ a_4(z),\ a_5(z) \right]$$

Figure 3.4 - 2: Mapping of a multi-dimensional wavelet into vector notation.

This process is illustrated in Figure 3.4-3. Each column of X represents the configuration for the dot product for one spatial lag of the convolution. We can think of this mapping for each column as the superposition of the spatially reversed a grid onto the x grid at some lag. The lag for a particular column corresponds to the ordering of the output grid.

Let us now put the methods discussed above on a formal basis. Since there are no well-defined operators for this mapping, we will use short mnemonic words to represent each operator. These operators will be used only to define such mappings as described above.

In nearly all of these discussions the scalar elements may be replaced by matrices, however, to avoid undue confusion we will make the definitions in terms of scalar quantities.

REV - Reversing operator

REV reverses the positive sense of all dimensions of a process:

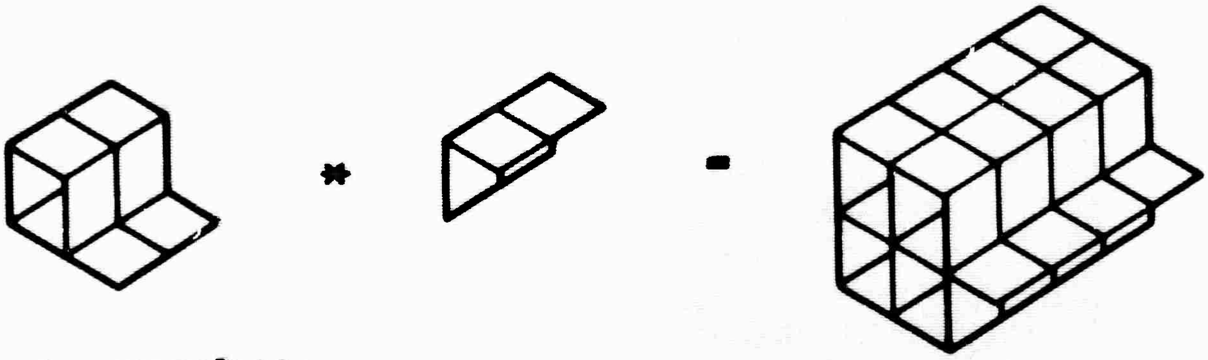
$$\mathrm{REV}(X_{\underline{i}}) = X_{-\underline{i}}$$

$$\mathrm{REV}(X(\underline{z})) = X(1/\underline{z})$$

where we define $-\underline{i} = (-i, -i_1, \ldots, -i_N)$

$$1/\underline{z} = (1/z, 1/z_1, \ldots, 1/z_N) .$$

$$a_{t,i_1,i_2} \quad * \quad x_{t,i_1,i_2} \quad = \quad y_{t,i_1,i_2}$$



* = convolution

z-Transform in Preferred Direction
and order the indices.

$$\begin{matrix} a_1(z) & a_3(z) & \\ a_2(z) & a_4(z) & a_5(z) \end{matrix} \; * \; \begin{matrix} x_1(z) & x_3(z) \\ x_2(z) & \end{matrix} \; = \; \begin{matrix} y_1(z) & y_4(z) & y_7(z) & \\ y_2(z) & y_5(z) & y_8(z) & y_{10}(z) \\ y_3(z) & y_6(z) & y_9(z) & \end{matrix}$$

Map wavelets into matrix notation
(a_1 represents $a_1(z)$, etc.)

$$\begin{bmatrix} a_1, & a_2, & a_3, & a_4, & a_5 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & 0 & x_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_1 & x_2 & 0 & x_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & 0 & x_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & 0 & x_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & x_2 & x_3 \end{bmatrix}$$

$$= \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} \end{bmatrix}$$

Figure 3.4 - 3: Mapping of multi-dimensional convolution into matrix representation.

SHIFT - Origin shifting operator

SHIFT alters the origin of a process by adding a value to each index of the process:

$$\text{SHIFT}\ (\underline{j}, X_{\underline{i}}) = X_{\underline{i} + \underline{j}}$$

$$\text{SHIFT}\ (\underline{j}, X(\underline{z})) = X(\underline{z})\ z^{j}\ z_1^{j_1} \ldots z_N^{j_N}$$

WINDOW - Window operator

WINDOW isolates a portion of a process, Y, by superimposing the grid of a process X onto the grid of Y. The indexing of the new process is that of the window grid X. We assume that Y has zeros wherever X extends beyond the defined limits of Y.

$$\text{WINDOW}\ (X, Y) = Z$$

For example consider the 2-dimensional process

$$X_{\underline{i}} = \begin{matrix} x_{0,-1} & x_{0,0} & x_{0,1} \\ x_{1,-1} & x_{1,0} & x_{1,1} \end{matrix}$$

$$Y_{\underline{i}} = \begin{matrix} y_{-1,0} & y_{-1,1} \\ y_{0,0} & y_{0,1} \\ y_{1,0} & y_{1,1} \end{matrix}$$

Then the WINDOW operator behaves as

$$
\begin{aligned}
Z_{\underline{i}} &= \text{WINDOW}\ (X, Y) \\
&= \begin{matrix} 0 & y_{0,0} & y_{0,1} \\ 0 & y_{1,0} & y_{1,1} \end{matrix} \\
&= \begin{matrix} z_{0,-1} & z_{0,0} & z_{0,1} \\ z_{1,-1} & z_{1,0} & z_{1,1} \end{matrix}
\end{aligned}
$$

ORDER - Ordering operator

ORDER converts a vector of $\underline{i} = (t, i_1, \ldots, i_N)$ into another vector (t,j) such that j takes on a unique value for each of the grid positions $(i_1, \ldots, i_N)$ of a finite process:

$$\text{ORDER}\ (X_{t,i_1,\ldots i_N}) = X_{t,j}$$

The actual process used to select the order of enumeration is entirely arbitrary and need not be specified until a specific application is made.

MAP1 - Mapping operator

MAP1 maps a multi-dimensional process into a vector-valued process. Consider a multi-dimensional process $A_{t,j} = \text{ORDER}\ (A_{t,\underline{i}})$ $1 \leq j \leq N$ then

$$
\begin{aligned}
\text{MAP1}\ (A_{t,j}) &= \underline{A}_t \\
&= \left[A_{t,1}, A_{t,2}, \ldots, A_{t,N} \right].
\end{aligned}
$$

MAP2 - Mapping operator

MAP2 corresponds to the matrix mapping of $X(\underline{z})$

that was made in Figure 3.4-3. It is defined in terms of the operators above.

$$X = \text{MAP2 }(a,x)$$
$$= \text{MAP1}\left(\text{ORDER }(\hat{X}_{t,\underline{k}})\right)$$

where $\hat{X}_{t,\underline{k}} = \text{MAP1}\left\langle\text{ORDER}\left[\text{WINDOW}\left(\text{SHIFT}(\underline{k},\ a),\ \text{REV}(x)\right)\right]\right\rangle.$

That is, $\hat{X}_{t,\underline{k}}$ represents the columns of the matrix X located in an array like that of the convolution of a and x . The individual columns of $\hat{X}_{t,\underline{k}}$ are formed by shifting the grid of a by an amount $\underline{k}$, superimposing it on the spatial reverse of x, and then ordering and mapping this intersection according to the indices of a .

Now, in terms of the z-transforms

$$a(\underline{z})\ x(\underline{z}) = y(\underline{z})$$

corresponds exactly to

$$\text{MAP1 }(a(z))\ \text{ MAP2 }(a,\ x(z)) = \text{MAP1 }(y(z))$$

where a(z) represents the z-transform of a in the preferred direction only.

4. FACTORIZATION OF AUTOCORRELATIONS

The operation of autocorrelation is generally defined as the expected value of the cross-product of a process with itself as a function of time and spatial lags. It has the very useful property of removing all phase information from a stochastic process. If a time-series may be characterized as the convolution of a white light process with a wavelet, then the autocorrelation of the process isolates the amplitude properties of the wavelet. This is because the autocorrelation of white light is zero except for a pulse at lag zero. These properties of stochastic processes have been treated by many authors and from many different viewpoints. Some of the salient works include Riesz (1907 and 1952), Fejer (1916), Kolmogorov (1941a and b), Karhuenen (1947 and 1949), and Szego (1959). Wold (1938) stated the decompositional properties in terms of stochastic time series as follows:

Theorem 4.1-1 (Wold Decomposition). Any stationary process X_t can be uniquely represented as the sum of two mutually uncorrelated process $X_t = U_t + V_t$, where U_t is deterministic, and V_t is the convolution of a one-sided wavelet with a stationary white-light process.

Robinson (1962) and Wiener and Masani (1957) have extended this theorem to specify a particular decomposition in terms of an invertible wavelet.

With this brief discussion of the motivation of autocorrelations (for more detailed discussions see Wiener, 1949; Whittle, 1954; Wiener and Masani, 1957 and 1958; and Anstey, 1964) we will go directly to a discussion of their properties and factorizations. In general, most of the factorizations are made in terms of correlations of finite length; however, some of the cases are easily extendible to infinite lengths.

4.1 One-Dimensional Scalar Autocorrelations

The theory of one-dimensional scalar autocorrelations is well known. Thus we need only state results in this section for the purpose of giving an intuitive introduction to the following sections.

Let $r(z)$ represent a real autocorrelation of length m+ m + 1

$$r(z) = r_{-m} z^{-m} + \ldots + r_{-1} z^{-1} + r_0 + r_1 z + \ldots + r_m z^m$$

then

a) $r(z)$ is reverse-symmetric, that is

$$\begin{aligned} r(z) &= r_*(z) \\ &= r(1/z) \, . \end{aligned}$$

b) $r(e^{-i\omega})$ is non-negative, that is, the cosine transform of the autocorrelation is non-negative.

c) $r_0 \geq r_1$ with equality holding only if the input process is periodic, i.e. deterministic.

d) The real frequency zeros, that is, the zeros on the unit circle, $|z| = 1$, are of even multiplicity.

e) For every zero z_1 of $r(z)$ inside the unit circle, there is a corresponding zero $1/z_1$ outside the unit circle.

It is interesting to note that since the cosine transforms of autocorrelation functions and of minimum delay wavelets are both non-negative (see Section 3.25), the center point and right half of a scalar autocorrelation forms a minimum-delay wavelet.

(Kunetz (1964) has proven that a synthetic seismogram which includes all multiple reflections forms one side of an autocorrelation function. In view of the results obtained above, we can sharpen his result to say that a synthetic seismogram which includes all multiples and the initiating pulse is minimum delay, if, and only if, the initiating pulse is minimum delay (Kunetz took this pulse to be a unit spike, which is certainly minimum delay).)

4.11 Factorization Theorem

An autocorrelation function may always be factored to give a wavelet $a(z)$ such that its autocorrelation, $a(z)\ a_*(z)$, equals the original autocorrelation. In general the factorization is not unique but it may be made unique by requiring that $a(z)$ be a one-sided invertible wavelet, i.e. minimum delay. Then this wavelet is the Robinson canonical minimum-delay form of all other factorizations. These properties are stated more rigorously in the factorization theorem.

Theorem 4.1-1 (Autocorrelation Factorization). Let $r(z)$ be a real scalar autocorrelation of degree $\pm\ m$. Then there exists a real polynomial (wavelet) $a(z)$ of degree m such that

a) $r(z) = a(z)\ a_*(z)$.

b) $a(z)$ and $a^{-1}(z)$ are both analytic inside the unit circle $|z| < 1$, i.e. $a(z)$ is one-sided invertible, minimum delay, or minimum phase.

c) $a(z)$ is unique up to within a trivial all-pass system multiplier, i.e., if $b(z)$ also satisfies a) and b), then $b(z) = p\ a(z)$ where p is a constant such that $p\ \bar{p} = 1$.

d) Any factorization of the form $r(z) = c(z)c_*(z)$ in which $c(z)$ is not invertible is given by

$$c(z) = a(z)\, p(z)$$

$p(z)$ being an arbitrary regular all-pass system.

Since the proofs of parts c) and d) are very similar to that for matrix-valued autocorrelations we will defer the proof of those parts until the next section (also see Robinson, 1963, p. 179). The proofs of parts a) and b) consist of showing that a factorization with the needed properties exist. We will state three factorizations here but will defer again until the next chapter for the discussion of approxiamte factorizations since the scalar methods are just special cases of the matrix-valued techniques.

4.12 Methods of Factorization

4.121 Woldian or spectral analysis

As pointed out at the beginning of section 4.1, every zero, a_i; of the polynomial $r(z)$ is associated with a zero $1/a_i$. Thus if we choose the m zeros a_i $i = 1, \ldots, m$ which fall outside the unit circle to form the polynomial, $a(z)$, then this polynomial will certainly concur with parts a) and b) of the factorization theorem.

4.122 Kolmogorov

If we have the square-gain (that is, the cosine transform), $r(e^{-i\omega})$, of the system, then the wavelet, $a(z)$, is given by

$$a(z) = \sum_{i=0} a_i z^i$$

$$= \exp\left[\frac{1}{4\pi}\int_{-\pi}^{\pi} \frac{e^{i\omega}+z}{e^{i\omega}-z} \log r(e^{i\omega})\, d\omega\right], \ |z| < 1$$

(Robinson, 1963b or Karhunen, 1949).

4.123 Zero-phase

The zero-phase factorization is also based upon the cosine spectrum, however, it does not produce a wavelet that satisfies part b) of Theorem 4.1-1. If we desire the wavelet to be two-sided and symmetrical then we need only take the square root of the spectrum $a(e^{-i\omega}) = \sqrt{r(e^{-i\omega})}$. This wavelet has zero phase.

The spectral and Kolmorgorov factorizations are equivalent (Robinson, 1954). The spectral technique is not a good computational method because of the well known difficulties in finding the zeros of a polynomial. The Kolmorgorov technique becomes approximate in computer applications since we must compute some continuous functions digitally. It has, however, been successfully applied to factorization problems (Galbraith, 1963).

4.2 One-Dimensional Matrix Autocorrelations

The matrix-valued autocorrelation function is very similar to the scalar function.

Let $R(z)$ be an $n \times n$ quasipolynomial autocorrelation matrix of rank r, then

a) $R(z)$ is reverse-symmetric, i.e. $R(z) = R_*(z)$.

b) $R(e^{i\omega})$ is non-negative definite, i.e. $\underline{b}^* R(z) \underline{b} > 0$ for every n vector $\underline{b}$ and every value of z on the unit circle.

c) The determinant of $R(z)$, $d(z) = \det R(z)$ is reverse-symmetrical $d(z) = d_*(z)$.

d) The Smith-McMillan canonical form satisfies

$$D(z) = D_*(z) \ .$$

e) The real frequency zeros, i.e. the zeros on the unit circle, of the diagonal elements of $D(z)$ (and of $d(z)$) are of even multiplicity.

Proof. Statement a) is obvious. If we let $X(z)$ represent an arbitrary finite process, then

$$\begin{aligned} R_*(z) &= (X(z)\, X_*(z))_* \\ &= X_{**}(z)\, X_*(z) \\ &= X(z)\, X_*(z) \\ &= R(z) \end{aligned}$$

Statement b) follows if we note that on the unit circle the determinate of $R(z)$ is

$$\begin{aligned} \left| X(e^{i\omega})\, X_*(e^{i\omega}) \right| &= \left| X(e^{i\omega}) \right| \; \left| X\ (e^{-i\omega}) \right| \\ &= \left| X(e^{i\omega}) \right| \; \left| \overline{X\ (e^{i\omega})} \right| \\ &> 0 \end{aligned}$$

unless $R(z)$ is null.

Statement c) follows directly from a). For statements d) and e) we let $R(z) = C(z)\ D(z)\ F(z)$ be the Smith-McMillan canonical form of $R(z)$. Now, since $R(z) = R_*(z)$, $C(z)\ D(z)\ F(z) = F_*(z)\ D_*(z)\ C_*(z)$. But $D(z)$ and $D_*(z)$ are both canonical to the same matrix $R(z)$ and therefore by the Smith-McMillan Theorem must be the same. Thus every diagonal element of $D(z)$ is reverse-symmetric, and consequently any zero z_ℓ is accompanied by a zero $1/z_\ell$. However, if z_ℓ is a zero of $R(z)$ then it must also have been a zero of $X(z)$. Since $X(z)$ is real, it has a real canonic form $D_1(z)$ and

$$D(z) = D_1(z)\ (D_1)_*\ (z)$$

But since D_1 is real, every root z_ℓ must be accompanied by its complex conjugate $\bar{z}_\ell$. Therefore for every root $|z_\ell| = 1$ on the unit circle we must have four roots z_ℓ, $1/z_\ell$, $\bar{z}_\ell$, and $1/\bar{z}_\ell$. But $z_\ell = 1/\bar{z}_\ell$, $\bar{z}_\ell = 1/z_\ell$ if $|z_\ell| = 1$. Thus all roots on the unit circle must occur in pairs. Q.E.D.

4.21 Factorization Theorem

A matrix-valued autocorrelation may always be factored into the product of a wavelet with its reverse-transpose. This factorization is made unique if we require that the wavelet be one-sided and invertible, i.e. minimum delay. This review is stated more concisely in the Matrix-valued Factorization Theorem.

<u>Theorem</u> <u>4.2-1</u> (<u>Matrix-Valued</u> <u>Autocorrelation</u> <u>Factorization</u>). Let $R(z)$ be a real $n \times n$ quasipolynomial autocorrelation matrix of rank r. Then there exists a real $n \times r$ polynomial matrix $A(z)$ such that

a) $R(z) = A(z)\, A_*(z)$

b) $A(z)$ and $A^{-1}(z)$, its left inverse, are both analytic inside the unit circle. If $R(z)$ is full rank and non-degenerate, $A(z)$ is minimum delay.

c) $A(z)$ is unique up to within a real-orthogonal matrix multiplier on the right (a trivial all-pass system), i.e., if $A_1(z)$ also satisfies a) and b), then $A_1(z) = A(z)\, T$ where T is $r \times r$, constant and unitary, $T\, T^* = 1_r$.

d) Any non-minimum delay factorization of the form $R(z) = C(z)\, C_*(z)$ in which $C(z)$ is $n \times m$, $m \geq r$, and polynomial, is given by the Robinson canonical form

$$C(z) = A(z) \left[1_r \mid 0_{r,m-r} \right] P(z) \quad .$$

$P(z)$ being an arbitrary rational regular $m \times m$ reverse-unitary matrix (that is, $P(z)$ is an $m \times m$ all-pass system).

The proof to this important theorem is divided into two parts. First we prove parts c) and d). Then parts a) and b) are proven in the next section by demonstrating factorization algorithms which produce wavelets having the given properties. Four such algorithms are known. Two produce $A(z)$ by analytical manipulations and two give $A^{-1}(z)$ by approximate techniques.

<u>Proof</u>. Consider statement d) first. Let $C(z)=A(z)Q(z)$ where $A(z)$ satisfies a) and b). Then

$$C(z)\ C_*(z) = A(z)\ Q(z)\ Q_*(z)\ A_*(z)$$

$$= A(z)\ A_*(z)$$

$$Q(z)\ Q_*(z) = 1_r$$

where $Q(z) = A^{-1}(z)\ C(z)$ is obviously analytic inside the unit circle, i.e. $P(z)$ is an arbitrary $m \times m$ reverse-unitary matrix that incorporates $Q(z)$ in its first r rows; i.e.,

$$Q(z) = \left[1_r \mid 0_{r,m-r} \right] P(z) \quad .$$

Now let us consider statement c). Let $A(z)$ and $A_1(z)$ be two matrices satisfying a) and b), and let $A(z) = A_1(z)\,Q(z)$. Then

$$A(z)\,A_*(z) = A_1(z)\,(A_1)_*\,(z)$$

$$= A_1(z)\,Q(z)\,Q_*(z)\,(A_1)_*\,(z)$$

$$Q(z)\,Q_*(z) = 1_r$$

where $Q(z) = A_1^{-1}(z)\,A(z)$ is analytic inside the unit circle. But we also have $Q(z) = (A_1)_*\,(z)\,A_*^{-1}(z)$ and it is therefore analytic outside the unit circle. By Theorem 3.3-6, $Q(z)$ is a constant real orthogonal matrix.

4.22 Analytic Factorization Methods

Both of the analytic factorizations depend upon the factorization of an elementary autocorrelation matrix. We will discuss this technique first. The algorithm was first presented by Oona and Yasuura (1954, pp. 125-177) and later expanded upon by Youla (1961, pp. 176-178) for paraconjugate-hermitian matrices. The following statement has been altered to account for the properties of reverse-symmetric matrices.

4.221 Elementary autocorrelation matrix

Consider an $r \times r$ positive elementary quasi-polynomial reverse-symmetri matrix, i.e., an elementary autocorrelation matrix, $R(z)$. Because of t... positive nature of $R(e^{i\omega})$, all its diagonal elements are reverse-symmetric and positive on the unit circle. Let $q_1 \leq q_2 \leq, \ldots, \leq q_r$ be the maximum degrees of the diagonal entries arranged in non-decreasing order. Since $R(z)$ is reverse-symmetric, the q's are non-negative integers. Again invoking the positive character of $R(e^{i\omega})$, it follows that no element in $R(z)$ has degree exceeding q_r. Thus $q_r = 0$ if and only if $R(z)$ is a constant symmetric positive-definite $r \times r$ matrix, in which case it can be written as AA^* by a number of standard techniques. Excluding this relatively trivial situation, we will assume $q_r > 0$.

We begin by interchanging the rows and columns of $R(z)$ so as to make its diagonal elements $(R)_{11}, (R)_{22}, \ldots, (R)_{rr}$ possess the degrees $q_1, q_2, \ldots, q_r$, respectively. Call the rearranged matrix $R_1(z)$. Then there exists a permutation matrix K such that

$$R_1(z) = K\,R(z)\,K' \tag{4.2-1}$$

R_1 is also elementary, reverse-symmetric and positive.

Next we force each diagonal term to have degree q_r. Let us begin by defining a non-increasing sequence of non-negative integers $\sigma_1, \sigma_2, \ldots, \sigma_r$ by

$$\sigma_i = q_r - q_i \qquad i = 1, 2, \ldots, r \tag{4.2-2}$$

and the $r \times r$ diagonal matrix $H(z)$ by

$$H(z) = \operatorname{diag}\left[(1 - z^{k_1})^{\sigma_1}, (1 - z^{k_2})^{\sigma_2}, \ldots, (1 - z^{k_r})^{\sigma_r}\right]. \tag{4.2-3}$$

where $k_i = \pm 1$ chosen so that the degree of the non-diagonal terms do not exceed q_r. Note that $\sigma_r = 0$. The $r \times r$ matrix

$$R_2(z) = H(z)\, R_1(z)\, H_*(z) \tag{4.2-4}$$

is quasipolynomial, reverse-symmetric and positive. Moreover all of its diagonal elements have the same degree q_r. Since R_1 is elementary, it is clear that

$$\operatorname{Det} R_2(z) = 0(z^{\sigma}) \tag{4.2-5}$$

where

$$\sigma = \sigma_1 + \sigma_2 + \ldots + \sigma_r \tag{4.2-6}$$

But from equation 4.2-2

$$\sigma \leq (r - 1)\, q_r . \tag{4.2-7}$$

$R_2(z)$ may be written in expanded form as

$$R_2(z) = T'_{q_r} z^{-q_r} + \ldots + T'_1 z^{-1} + T_0 + T_1 z + \ldots + T_{q_r} z^{q_r} \tag{4.2-8}$$

where the T's are constant $r \times r$ matrices. The important observation is that T_{q_r} is singular, i.e. $\operatorname{Det} T_{q_r} = 0$ for otherwise equation 4.2-8 would yield

$$\text{Det } R_2(z) = O(z^{rq_r}) \tag{4.2-9}$$

which contradicts equations 4.2-5 and 4.2-7. This deduction implies that T_{q_r} contains a principal minor G of order $s \times s$ which is non-singular and such that the minor $\tilde{G}$ created by adding the $(s+1)$th row and column to G is singular. Thus we may add a linear combination of the first s rows of T_{q_r} to the $(s+1)$th row and the <u>same</u> linear combination of the first s columns to the $(s+1)$th column such that $(T_{q_r})_{s+1,\, s+1}$ is reduced to zero and no other diagonal term is affected. Hence for the correct choice of a constant $r \times r$ non-singular matrix K_1,

$$\tilde{T}_{q_r} = K_1 \, T_{q_r} \, K_1' \tag{4.2-10}$$

has a zero element in the $(s+1, s+1)$ place. From 4.2-8

$$R_3(z) = K_1 \, R_2(z) \, K_1' = \sum_{i=-q_r}^{q_r} (K_1 \, T_i \, K_1') z^i \tag{4.2-11}$$

has a diagonal element in the $(s+1, s+1)$ position of degree $< q_r$.

The matrix

$$R_4(z) = H^{-1}(z) \, R_3(z) \, H_*^{-1}(z) \tag{4.2-12}$$

is reverse-symmetric, positive and elementary. According to the definition of R_2 (see equation 4.2-4) $(R_2)_{\ell,m}$ is divisible by $(1 - z^{k_\ell})^{\sigma_\ell}$ $(1 - z^{-k_m})^{\sigma_m}$, and according to the definition of $R_3(z)$ (see equation 4.2-11) and the definition of K_1, $R_3(z)$ differs from $R_2(z)$ only in its $(s + 1)$th row and column. More specifically,

$$(R_3)_{\ell,s+1} = (R_2)_{\ell,s+1} + \sum_{i=1}^{s} c_i \, (R_2)_{\ell,i} \quad (\ell=1,\ 2,\ \ldots,\ r), \tag{4.2-13}$$

the c's being scalars. By construction $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$, thus every term on the right-hand side of equation 4.2-13 is divisable by $(1 - z^{k_{s+1}})^{\sigma_{s+1}} (1 - z^{-k_m})^{\sigma_m}$, $(k = 1,\ 2,\ \ldots,\ r)$. The same considerations apply to the $(s + 1)$th row, whence, for all ℓ and m, $(R_3)_{\ell,m}$ is divisible by $(1 - z^{k_\ell})^{\sigma_\ell} (1 - z^{-k_m})^{\sigma_m}$, and $R_4(z)$ is a quasipolynomial matrix. Since

$$\text{Det } R_4(z) = \text{Det}(K_1^{\,2} K^2) \text{ Det}(R(z)) = \text{constant},$$

$R_4(z)$ is elementary.

But $R_4(z)$ is simpler than $R_1(z)$ because the degree of its $(s+1,\ s+1)$ entry is at least 1 less than the same entry in the latter matrix, while all other corresponding diagonal elements have the same degree as before. Consequently, after one cycle of the algorithm,

$$R(z) = G_1(z) \, R_4(z) \, (G_1)_*(z) \tag{4.2-14}$$

where

$$G_1(z) = K^{-1} H^{-1}(z) \, K_1^{-1} H(z)$$

is an elementary polynomial matrix and $R_4(z)$ is at least 1 deg. less than $R(z)$.

We now replace $R(z)$ by $R_4(z)$ and repeat the algorithm. After a maximum of rq_r cycles $R(z)$ is reduced to a constant symmetric positive-definite matrix $R_4 = CC'$, so that finally

$$R(z) = A(z)\, A_*(z)$$

where

$$A(z) = G_1(z)\, G_2(z) \cdots G_{rq_r}(z)\, C .$$

This factorization does not guarantee that $A(z)$ is one-sided. This is because of the ambiguity in the definition of $H(z)$. To the author's knowledge no one-sided factorizations exist for cases in which this algorithm does not give a right-sided factorization. For example, the elementary autocorrelation

$$R(z) = \begin{bmatrix} -z^{-1} + 3 - z & -z^{-1} + 1 \\ 1 - z & 1 \end{bmatrix}$$

may be factored either as

$$R(z) = \begin{bmatrix} -z^{-1} + 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 - z & 1 \\ 1 & 0 \end{bmatrix}$$

or as

$$R(z) = \begin{bmatrix} 1 & -z^{-1} + 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 - z & 1 \end{bmatrix}$$

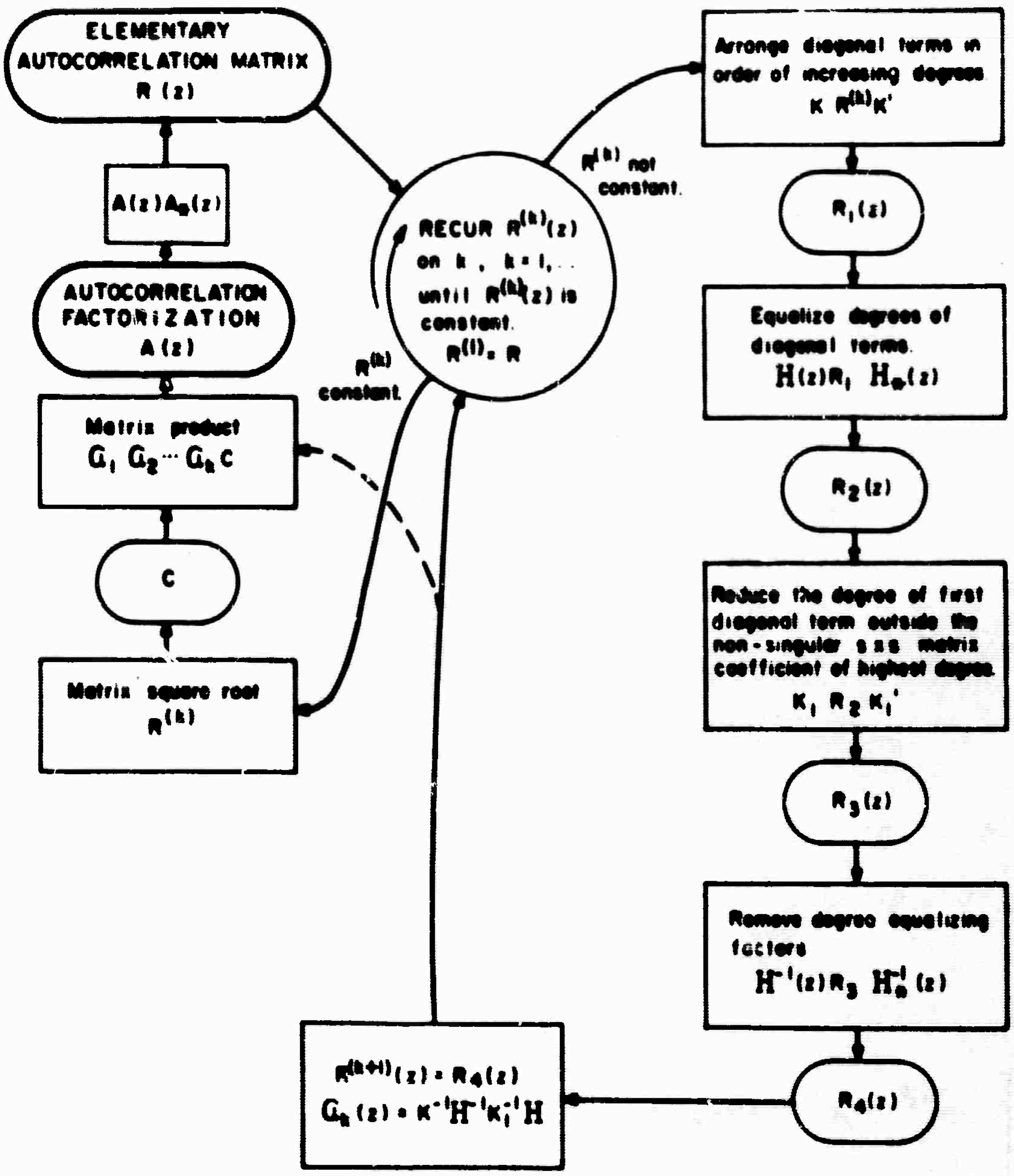


Figure 4.2 - 1: Elementary autocorrelation matrix factorization.

but no right-sided **form has** been found.

The important steps in this reduction are shown in Figure 4.2-1.

<u>Example</u> <u>4.2-1</u>. Let us consider the elementary autocorrelation matrix

$$R(z) = \begin{bmatrix} -2z^{-1}+6-2z & -4z^{-2}+14z^{-1}-14+4z \\ 4z^{-1}-14+14z-4z^{2} & 8z^{-2}-32z^{-1}+50-32z+8z^{2} \end{bmatrix}$$

We will follow the steps of this factorization in detail.

Recursion 1.

Since the degrees of the diagonal terms of $R(z)$ are already in ascending order we may skip the first step. Thus

$$R_1^{(1)}(z) = R(z)$$

Next we make all of the terms have the same degree by forming the product

$$R_2^{(1)}(z) = H^{(1)} R_1^{(1)} H_*^{(1)}, \quad H^{(1)} = \mathrm{diag}\left[1-z,\ 1\right]$$

$$= \begin{bmatrix} 2z^{-2}-10z^{-1}+16-10z \ +2z^{2} & -4z^{-2}+18z^{-1}-28+18z-4z^{2} \\ -4z^{2}\ +18z^{-1}-28+18z^{-1}-4z^{2} & 8z^{-2}-32z^{-1}+50-32z+8z^{2} \end{bmatrix}$$

The matrix for the z^2 terms is

$$T_2^{(1)} = \begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix}$$

Thus the diagonal term $(T_2)_{2,2}$ can be reduced to zero by the product

$$R_3^{(1)}(z) = K_1^{(1)}\, R_2^{(1)}\, (K_1^{(1)})' \quad \text{where} \quad K_1 = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2z^{-1} - 10z^{-1} + 16 - 10z + 2z^2 & -2z^{-1} + 4 - 2z \\ -\ 2z^{-1} + 4 - 2z & 2 \end{bmatrix}$$

Finally we remove the H multipliers to obtain:

$$R_4^{(1)}(z) = H^{(1)^{-1}}\, R_3^{(1)}\, H_*^{(1)^{-1}}$$

$$= \begin{bmatrix} -2z^{-1} + 6 - 2z & -2z^{-1} + 2 \\ 2 - 2z & 2 \end{bmatrix}$$

These steps can now be combined so that

$$R(z) = G_1\, R_4^{(1)}\, (G_1)_*$$

where

$$G_1 = H^{(1)^{-1}}\, K_1^{(1)^{-1}}\, H^{(1)}$$

$$= \begin{bmatrix} 1 & 0 \\ -2+2z & 1 \end{bmatrix}$$

Recursion 2.

Our beginning point is the matrix $R_4^{(1)}$ from the last recursion. This time we must exchange the positions of the diagonal terms so that they will have ascending degrees:

$$R_1^{(2)} = K^{(1)} R_4^{(1)} K^{(1)'} \quad \text{where} \quad K = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 2 - 2z \\ -2z^{-1} + 2 & -2z^{-1} + 6 - 2z \end{bmatrix}$$

Here we must now multiply by $H^{(2)} = \text{diag}\left[(1-z^{-1}),1\right]$ in order for the off diagonal terms to be the same degree as the diagonal terms.

$$R_2^{(2)}(z) = H^{(2)} R_1^{(2)} H_*^{(2)}$$

$$= \begin{bmatrix} -2z^{-1} + 4 - 2z & -2z^{-1} + 4 - 2z \\ -2z^{-1} + 4 - 2z & -2z^{-1} + 6 - 2z \end{bmatrix}$$

From the coefficients of z we may select $K_1^{(2)}$ such that the degree of $(R_2^{(2)})_{2,2}$ is reduced,

$$R_3^{(2)} = K_1^{(2)} R_2^{(2)} K_1^{(2)1} \quad \text{where} \quad K_1^{(2)} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2z^{-1} + 4 - 2z^{-1} & 0 \\ 0 & 2 \end{bmatrix}$$

and, proceeding as before, we have

$$R_4^{(2)} = (H^{(2)})^{-1} R_3^{(2)} (H_*^{(2)})^{-1}$$
$$= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

This set of operations may also be grouped

$$G_2 = K^{(2)^{-1}} H^{(2)^{-1}} K_1^{(2)^{-1}} H^{(2)} .$$

Now we see that the factorization of $R(z)$ is given by

$$R(z) = A(z)\, A_*(z)$$

where $A(z) = G_1 G_2 \sqrt{R_4^{(2)}}$

$$= \sqrt{2} \begin{bmatrix} -z^{-1} + 1 & 1 \\ 2z^{-1} - 3 + 2z & -2 + 2z \end{bmatrix} .$$

4.222 Spectral analysis

We will begin by illustrating the decomposition for an $n \times n$ full rank $(r = n)$, non-degenerate ($p=2mn$ zeros in the determinant, where m is the greatest number of zeros in any of the quasipolynomial elements) autocorrelation.

Let us assume that statement a) is true. We begin then by examining the latent zeros and vectors of $R(z)$ in terms of those of $A(z)$. The latent roots as specified by the determinat

$$\left|R(z)\right| = \left|A(z)\, A_*(z)\right|$$

$$= \left|A(z)\right| \left|A(z^{-1})\right| . \qquad (4.2\text{-}15)$$

are $z_i^{\pm 1}$ $i = 1, 2, \ldots, p$ where the z_i are roots of

$A(z)$. The latent vectors are either

$$\begin{aligned} \operatorname{Adj} R(z_1) &= \operatorname{Adj} (A(z_1)\, A_*(z_1)) \\ &= \operatorname{Adj} A_*(z_1)\, \operatorname{Adj} A(z_1) \\ &= \operatorname{Adj} A_*(z_1)\, \underline{u}_1\, \underline{v}_1' \\ &= \underline{w}_1\, \underline{v}_1' \end{aligned} \tag{4.2-16}$$

or

$$\begin{aligned} \operatorname{Adj} R(z_1^{-1}) &= \underline{v}_1\, \underline{u}_1' \;\operatorname{Adj} A(z_1^{-1}) \\ &= \underline{v}_1\, \underline{w}_1' \;. \end{aligned}$$

If we choose the p zeros outside the unit circle we will satisfy condition b). These zeros and their associated vectors $\underline{v}_1$ may be used to construct $\mathfrak{A}(z)$ according to either of the two methods illustrated in the proofs to Spectral Theorem or the Spectral Corollary (Section 3.32). We must now determine the constant multiplier A_0 from the autocorrelation

$$\begin{aligned} R(z) &= \mathfrak{A}(z)\, A_0 (A_0)_*\, \mathfrak{A}_*(z) \\ A_0 A_{0_*} &= \mathfrak{A}^{-1}(z)\, R(z)\, \mathfrak{A}_*^{-1}(z) \;. \end{aligned} \tag{4.2-17}$$

Thus we can only determine A_0 to within a real orthogonal multiplier.

The factorization used above is similar in intent to the Woldian factorization for scalar autocorrelations. The sequence of operations is illustrated in Figures 4.2-2 or 4.2-3.

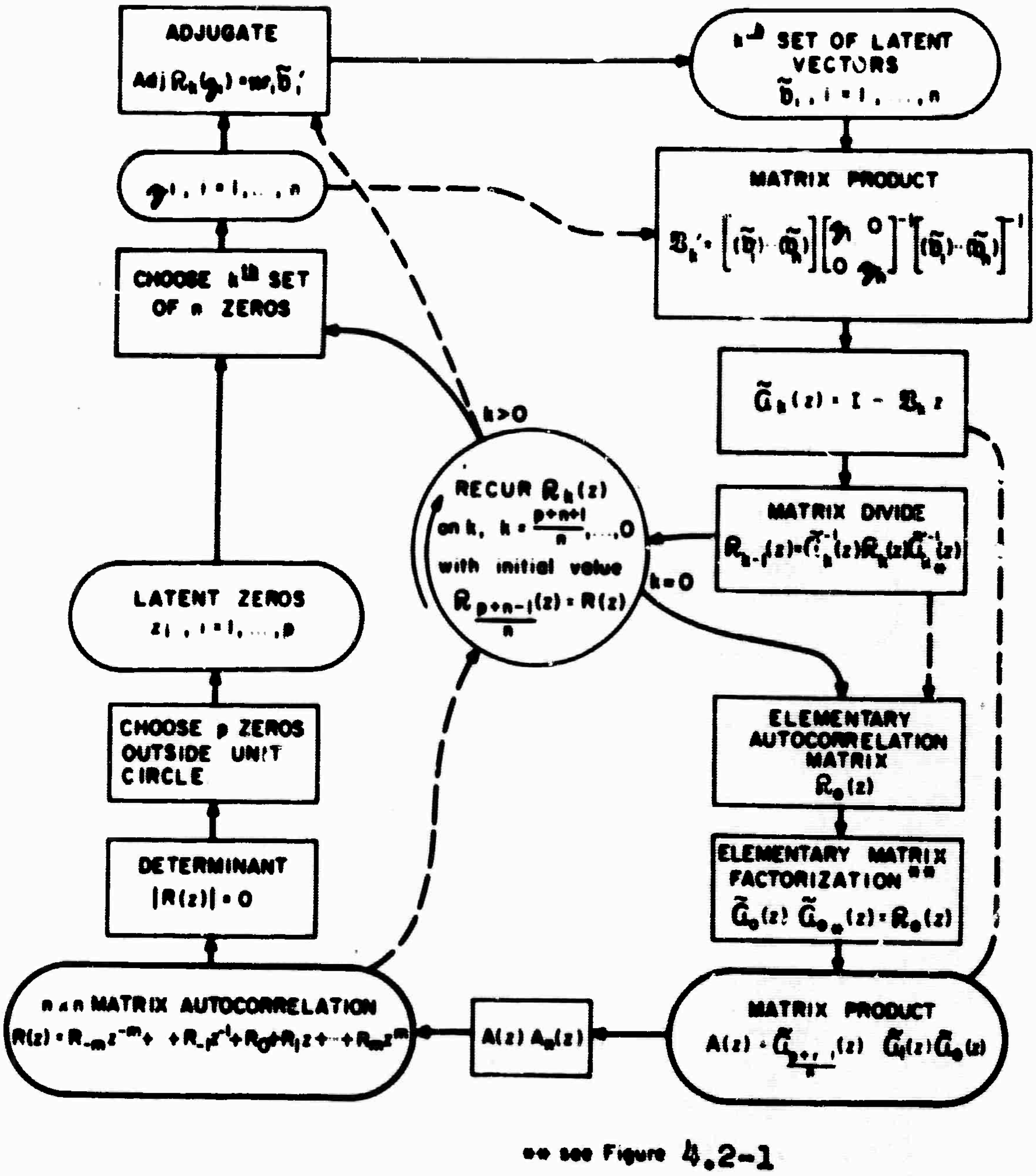


Figure 4.2 - 2: Spectral factorization of a matrix autocorrelation according to Theorem 3.3-1.

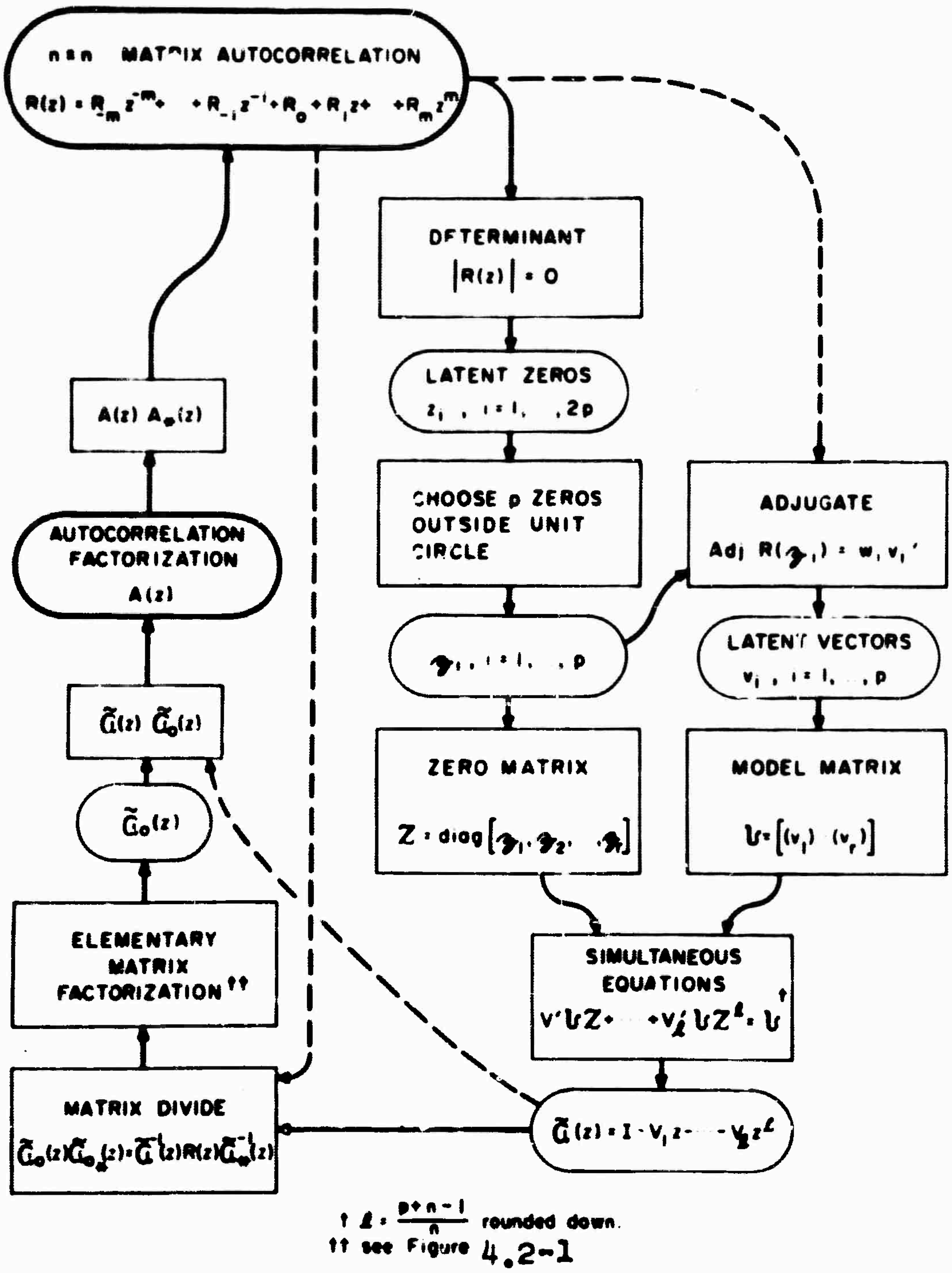


Figure 4.2 - 3: Spectral factorization of a matrix autocorrelation according to Corallary 3.3-1.

If $R(z)$ is full-rank but degenerate, $p < nm$, then the factorization is not complete when we reach equation 4.2-17. In this case we will have

$$\tilde{\mathcal{R}}(z) = \tilde{\mathcal{A}}^{-1}(z)\, R(z)\, \tilde{\mathcal{A}}_*^{-1}(z)$$

where $\tilde{\mathcal{R}}(z)$ is an elementary autocorrelation matrix which must be factored according to the method of Section 4.221 to give

$$\tilde{\mathcal{R}}(z) = \tilde{\mathcal{A}}_0(z)(\tilde{\mathcal{A}}_0)_*(z) \; .$$

The complete factorization is

$$A(z) = \tilde{\mathcal{A}}(z)\, \tilde{\mathcal{A}}_0(z) \; .$$

If $R(z)$ is not full rank, then the factorization must be done in terms of full rank submatrices of $R(z)$. Thus, we partition $R(z)$ symmetrically about the main diagonal such that each $r_1 \times r_1$ submatrix $\mathcal{R}_{11}(z)$ is full rank. For example

$$R(z) = \begin{array}{cc} \begin{array}{ccc} \longleftarrow r \longrightarrow & \leftarrow r_1 \rightarrow & \leftarrow r_2 \rightarrow \end{array} & \\ \left[\begin{array}{c|c|c} \mathcal{R}_{00} & \mathcal{R}_{01} & \mathcal{R}_{02} \\ \hline \mathcal{R}_{10} & \mathcal{R}_{11} & \mathcal{R}_{12} \\ \hline \mathcal{R}_{20} & \mathcal{R}_{21} & \mathcal{R}_{22} \end{array}\right] & \begin{array}{c} r \\ r_1 \\ r_2 \end{array} \end{array} \tag{4.2-18}$$

Each of the diagonal submatrice $(R)_{ii}$, is then factored according to the spectral theorem technique (given earlier in this section) to obtain the $r_i \times r_i$ matrices $\hat{G}_{ii}(z)$.
Now form the matrices

$$\hat{G}(z) = \left[\begin{array}{c|c} \multicolumn{2}{c}{\hat{G}_{00}} \\ \hline \hat{G}_{11} & 0_{r_1,\, r-r_1} \\ \hline \hat{G}_{22} & 0_{r_2,\, r-r_2} \end{array}\right] \tag{4.2-19}$$

and the left inverse

$$\hat{G}^{-1}(z) = C \left[\begin{array}{c|c|c} \hat{G}_{00}^{-1} & \begin{array}{c} \hat{G}_{11}^{-1} \\ \hline 0_{r-r_1,\, r_1} \end{array} & \begin{array}{c} \hat{G}_{22}^{-1} \\ \hline 0_{r-r_2,\, r_2} \end{array} \end{array}\right] \tag{4.2-20}$$

where C is a constant diagonal $r \times r$ matrix.
Now, the matrix

$$\tilde{R}(z) = \hat{G}^{-1}(z)\, R(z)\, \hat{G}_*^{-1}(z)$$

is an $r \times r$ elementary quasipolynomial matrix (this will not be proven here) which may be factored according to Section 4.221.
Thus

$$\tilde{R}(z) = \hat{G}_0(z)\, (\hat{G}_0)_*(z)$$

and $$A(z) = \hat{G}(z)\, \hat{G}_0(z)$$ Q.E.D.

Example 4.2-2. (Full rank, non-degenerate case)

Consider the autocorrelation matrix

$$R(z) = \begin{bmatrix} -2z^{-1} + 6 - 2z & -z^{-1} + 1 \\ 1 - z & -z^{-1} + 2 - z \end{bmatrix}.$$

We begin by finding the latent roots z_1 and latent vectors $\underline{v}_1$ and $\underline{w}_1$ by the technique outlined in the section 3.32.

$$\begin{aligned} R(z) &= 2z^{-2} - 9z^{-1} + 14 - 9z + 2z^2 \\ &= (1 - 2z)(1 - z)(1 - z^{-1})(1 - 2z^{-1}) \end{aligned}$$

Substituting these roots into Adj $R(z)$ we find

zero	vector $\underline{w}$	vector $\underline{v}$
1/2	(1, 1)	(-1, 2)
1	(0, 1)	(0, 1)
1	(0, 1)	(0, 1)
2	(-1, 2)	(1, 1)

We choose one of the roots on the unit circle and the root outside the unit circle to find $A_{\cdot}$. Using the notation of section 3.32, we have

$$\begin{aligned} V' &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1/2 & 0 \\ -1/2 & 1 \end{bmatrix} \end{aligned}$$

Thus $\mathcal{T}(z) = I - V z$

$$= \begin{bmatrix} 1 - 1/2\, z & 1/2\, z \\ 0 & 1 - z \end{bmatrix}$$

and

$$A_0 A_0' = \mathcal{T}^{-1}(z)\, R(z)\, \mathcal{T}_*^{-1}(z)$$

$$= \frac{1}{(1-3/2z+1/2z^2)^2} \begin{bmatrix} 1-z & 1/2z \\ 0 & 1-1/2z \end{bmatrix} \begin{bmatrix} -2z^{-1}+6-2z & -z^{-1}+1 \\ 1-z & -z^{-1}+2-z \end{bmatrix} \cdot \begin{bmatrix} -z^{-1}+1 & 0 \\ 1/2z^{-1} & -1/2z^{-1}+1 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

Consequently we find the desired minimum delay wavelet

$$A(z) = \mathcal{T}(z)\, A_0$$

$$= \begin{bmatrix} 2-z & 1 \\ 0 & 1-z \end{bmatrix} .$$

4.223 Smith-McMillan

This factorization is based upon the Smith-McMillan canonical form. The algorithm is very similar to a factorization technique for paraconjugate-hermitian matrices given by Youla (1961). The system is quite elegant in its conception since the algorithm is independent of the rank and degeneracy of the autocorrelation; however, a flaw remai so that for some cases a one-sided factorization cannot be guaranteed even for the full-rank non-degenerate matrix autocorrelation.

Before doing the factorization we must investigate several more properties of quasipolynomial matrices.

Definition 4.2-1. Let $G(z)$ be an $n \times m$ rational matrix of normal rank r. A decomposition of the form

$$G(z) = P(z)\,\Delta(z)\,Q(z)$$

is said to be an inner-standard factorization if

a) $\Delta(z)$ is $r \times r$, canonic and analytic together with its inverse in the entire z plane with the possible exception of a finite number of points on the unit circle.

b) $P(z)$ is $n \times r$ and analytic together with its left inverse inside and on the unit circle.

c) $Q(z)$ is $r \times m$ and analytic together with its right inverse outside and on the unit circle.

Interchanging the roles of P and Q gives rise to an <u>outer-standard factorization</u>. Obviously any inner-standard factorization of $G(z)$ generates an outer-standard factorization of $G'(z)$, $G^{-1}(z)$ and $G(1/z)$. For example $G'(z) = Q'(z)\,\Delta(z)\,P'(z)$, etc.

It follows from the Smith-McMillan Theorem that any rational matrix $G(z)$ possesses an inner- and outer-standard factorization. For, let $G(z) = C(z)\,D(z)\,F(z)$ where C and F are elementary and D is canonic. By factoring the φ's and ψ's (see Smith-McMillan Theorem, section 3.34) appearing in the diagonal elements of $D(z)$ into the product of three quasipolynomials, the first without zeros $|z| \leq 1$, the second without zeros $|z| \neq 1$, and the third without zeros in $|z| \geq 1$, it is possible then to write $D(z) = D^{+}(z)\,\Delta(z)\,D^{-}(z)$: $D^{+}(z)$ and its inverse are analytic in $|z| \leq 1$, $\Delta(z)$ and $\Delta^{-1}(z)$ in $|z| \neq 1$, and $D^{-}(z)$ and its inverse in $|z| \geq 1$. Now, choosing $P(z) = C(z)\,D^{+}(z)$ and $Q(z) = D^{-}(z)\,F(z)$ we have the desired breakdown.

<u>Lemma 4.2-1</u>. Let $G(z)$ possess two right-standard factorizations $G = P_1\,\Delta_1\,Q_1$. Then,

a) $\Delta(z) = \Delta_1(z)$

b) $P_1(z) = P(z)\,M^{-1}(z)$ and $Q_1(z) = N(z)Q(z)$, where $M(z)$ and $N^{-1}(z)$ are any two $r \times r$ elementary quasipolynomial matrices which transform $\Delta(z)$ into itself, vis,

$$M(z)\,\Delta(z)\,N^{-1}(z) = \Delta(z) .$$

Proof. We have

$$G = P\,\Delta\,Q = P_1\,\Delta_1\,Q_1 \tag{4.2-21}$$

Then

$$\Delta_1^{-1}\,P_1^{-1}\,P\Delta = Q_1\,Q^{-1} \tag{4.2-22}$$

By definition the right hand side of equation 4.2-22 is analytic in $|z| \geq 1$. Thus Q_1Q^{-1} is analytic in the entire z plane. According to equation 4.2-21 the inverse of Q_1Q^{-1} is $\Delta^{-1}\,P^{-1}\,P_1\,\Delta_1 = QQ_1^{-1}$ and is therefore also analytic in the entire z plane. By Theorem 3.3-4 Q_1Q^{-1} is therefore an elementary $r \times r$ quasipolynomial matrix $N(z)$. Similarly $P_1^{-1}P$ is an $r \times r$ elementary quasipolynomial matrix $M(z)$. From equation 4.2-21

$$M(z)\,\Delta(z)\,N^{-1}(z) = \Delta_1(z) .$$

Since $\Delta(z)$ and $\Delta_1(z)$ are both canonic, $\Delta(z) = \Delta_1(z)$ by the Smith-McMillan Theorem. Thus

$$M(z) = \Delta_1(z)\,N(z)\,\Delta^{-1}(z)$$

$$Q_1(z) = N(z)\,Q(z)$$

$$P_1(z) = P(z)\,\Delta(z)\,N^{-1}(z)\,\Delta^{-1}(p)$$

$$= P(z)\,M^{-1}(z)$$

Q.E.D.

<u>Corollary 4.2-1</u>. The canonic matrix $\Delta(z)$ appearing in either an inner-standard or outer-standard factorization of an $n \times m$ matrix $G(z)$ of rank $r(G)$ is equal to the $r \times r$ identity matrix 1_r if and only if $G(z)$ is analytic and $r(G)$ is constant on the unit circle. In this case, if $P\ Q$ and $P_1\ Q_1$ are any two standard factorizations of G, $P_1(a) = P(z)\ N^{-1}(z)$ and $Q_1(z) = N(z)\ Q(z)$, $N(z)$ being an arbitrary $r \times r$ elementary quasipolynomial matrix.

<u>Proof</u>. The if part is immediate. The analyticity of $G(z)$ on the unit circle implies that all of the denominator terms of $\Delta(z)$ are unity. This in turn leads to the conclusion that $r(G)$ is constant on the unit circle only if the numerator quasipolynomials in $\Delta(z)$ are unity. Thus $\Delta(z) = 1_r$. The remaining statements are consequences of Lemma 4.2-1. Q.E.D.

<u>Corollary 4.2-1a</u>. If $G(z)$ is reverse-symmetric then

$$N(z) = M_*(z)$$

where $M(z)$ is any $r \times r$ elementary quasipolynomial matrix satisfying $\Delta(z)\ M_*(z) = M(z)\ \Delta_*(z)$.

<u>Proof</u>. Since $G(z) = G_*(z)$, $Q_*(z)\Delta(z)\ P_*(z)$ is also a right standard factorization of $G(z)$ by arguments similar to those used for theorem 3.3-4. Thus, according to Lemma 4.2-1

$$P_*(z) = N(z)\, Q(z)$$

$$Q_*(z) = P(z)\, M^{-1}(z)$$

$$\therefore \quad P_*(z) = N(z)\, M_*^{-1}(z)\, P_*(z)$$

Since $P_*(z)$ has a right inverse,

$$N(z) = M_*(z)$$

and according to Lemma 4.2-1

$$\Delta(z)\, M_*(z) = M(z)\, \Delta_*(z) \; . \qquad \text{Q.E.D.}$$

The factorization algorithm discussed here is based upon the Smith-McMillan canonical form for the autocorrelation matrix. Unfortunately, because of the arbitrariness of the sequence of steps involved in finding a particular realization of the Smith-McMillan canonical form, the solution is not unique. The solution matrix $\bar{\Lambda}(z)$ is not one-sided (and therefore not analytic inside the unit circle). This matrix $\bar{\Lambda}(z)$ will differ from the proper answer by a unitary matrix.

Step 1. Reduce the matrix $R(z)$ to its Smith-McMillan canonic form. Since $R(z)$ is a quasipolynomial matrix, this procedure is a standard but arbitrary one as illustrated in section 3.34. Thus we will have

$$R(z) = C(z)\, D(z)\, F(z) \; .$$

Step 2. According to Theorem 4.2-1, $D(z)$ is of the form that it may be factored as

$$D(z) = D^+(z)\,\Delta(z)\,D^-(z)$$

$$= D^+(z)\,\Delta(z)\,D^+_*(z)$$

where

(1) $D^+(z)$ is $r \times r$, diagonal and analytic, together with its inverse $(D^+)^{-1}(z)$ for $|z| \leq 1$.

(2) $\Delta_*(z) = \Delta(z) = \theta(z)\,\theta_*(z)$ in which all diagonal elements of $\theta(z)$ are reverse-symmetric. Furthermore, $\Delta(z)$ is canonic and non-zero for $|z| \neq 1$.

Let

$$P(z) = C(z)\,D^+(z)$$

$$Q(z) = D^+_*(z)\,F(z)$$

Then we have an inner-standard factorization

$$R(z) = P(z)\,\Delta(z)\,Q(z)$$

Step 3. Now we wish to factor $\Delta(z)$. Since $R(z)$ is reverse-symmetric, a second left standard factorization is

$$R(z) = Q_*(z)\,\Delta_*(z)\,P_*(z)$$

and according to Lemma 4.2-1 and its Corollaries

$$Q_*(z) = P(z)\,M^{-1}(z) \tag{4.2-23}$$

where $M^{-1}(z)$ is an $r \times r$ elementary quasipolynomial matrix such that

$$\Delta(z)^{-1}\, M(z)\, \Delta(z) \;=\; N(z) \tag{4.2-24}$$

is also quasipolynomial.

Thus we may write

$$\begin{aligned} R(z) &= P(z)\, M^{-1}(z)\, \Delta(z)\, P_*(z) \\ &= P\,\theta\,\theta^{-1}\, M^{-1}\,\theta\,\theta_*\, P_* \end{aligned} \tag{4.2-26}$$

or

$$\theta^{-1}\, P^{-1}\, R\, P_*^{-1}\, \theta_*^{-1} \;=\; \theta^{-1}\, M^{-1}\, \theta \tag{4.2-26}$$

Hence

$$\tilde{M}(z) \;\equiv\; \theta^{-1}(z)\, M^{-1}(z)\, \theta_*(z)$$

is $r \times r$, reverse-symmetric and non-negative on the unit circle (by the properties of equation 4.2-26). Actually we can say a good deal more. Let us write equation 4.2-24 in terms of its elements:

$$(M)_{rk}(z)\,\frac{(\Delta)_{kk}(z)}{(\Delta)_{rr}(z)} \;=\; (M)_{rk}(z)\,\frac{(\theta)^2_{kk}(z)}{(\theta)^2_{rr}(z)} \;.$$

Since each element must be quasipolynomial

$$(M)_{rk}(z)\,\frac{(\theta)_{kk}(z)}{(\theta)_{rr}(z)}$$

must also be quasipolynomial. Thus $\tilde{M}(z)$ is a quasipolynomial matrix. But $\tilde{M}(z) = M^{-1}(z) =$ constant, i.e.,

$\tilde{M}(z)$ is a positive reverse-symmetric $r \times r$ elementary quasipolynomial matrix. In Section 4.221 we demonstrated that such a matrix is factorable as

$$\tilde{M}(z) = S(z)\ S_*(z)\ ,$$

$S(z)$ being an $r \times r$ elementary quasipolynomial matrix. After this is achieved, a factorization for $R(z)$ is obtained as $R(z) = \tilde{A}(z)\ \tilde{A}_*(z)$ with

$$\begin{aligned}\tilde{A}(z) &= P(z)\ \theta(z)\ S(z) \\ &= C(z)\ D^+(z)\ \theta(z)\ S(z)\ .\end{aligned}$$

where $\tilde{A}(z)$ differs from the desired factorization $A(z)$ by a unitary matrix. By straight forward algebra

$$\begin{aligned}\tilde{A}(z)\ \tilde{A}_*(z) &= C\ D^+\ \theta\ S\ S_*\ \theta_*\ D_*^+\ C_* \\ &= C\ D^+\ M^{-1}\ \theta_*^2\ D_*^+\ C_* \\ &= P\ M^{-1}\ \Delta\ P_* \\ &= Q_*\ \Delta\ P_* \\ &= R\end{aligned}$$

The pertinent computational steps involved in this algorithm are illustrated in Figure 4.2-4.

The advantage of this factorization is that the degenerate and singular autocorrelation matrices need not be treated as special cases (as contrasted to the spectral

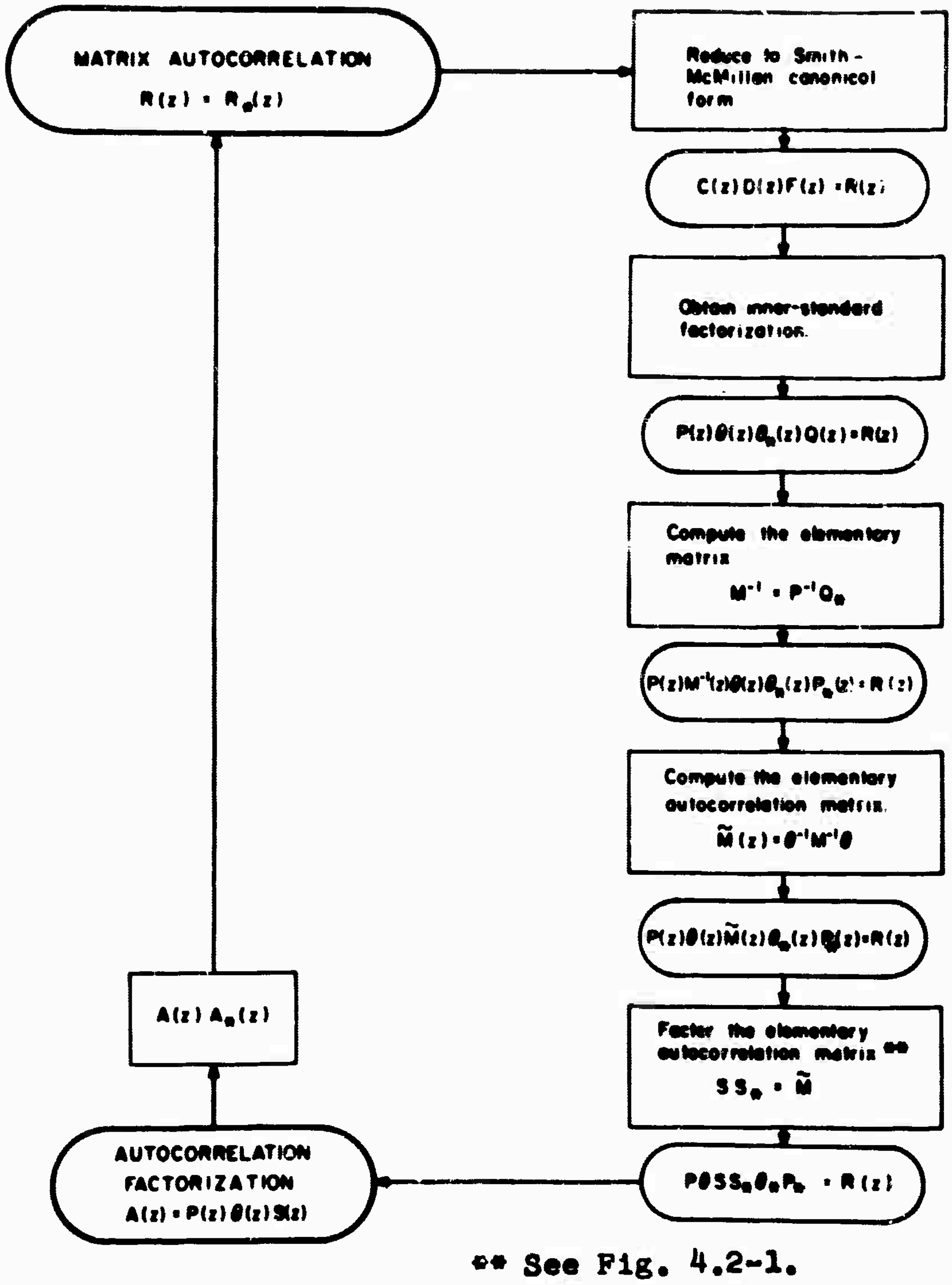


Figure 4.2 - 4: Smith-McMillan factorization of a matrix autocorrelation.

approach). If an algorithm can be found for forming the Smith-McMillan canonical form so that the factorizations must be one-sided then this formulation would become more important than the spectral approach.

Example 4.2-3. As in Example 4.2-2 we will consider the autocorrelation matrix

$$R(z) = \begin{bmatrix} -2z^{-1} + 6 - 2z & -z^{-1} + 1 \\ 1 - z & -z^{-1} + 2 - z \end{bmatrix} .$$

Step 1 of the factorization consists of reducing $R(z)$ to the Smith-McMillan canonical form. Since this process was illustrated in Example 3.3-3 we will merely give a particular result:

$$R(z) = C(z)\ D(z)\ F(z)$$

where $$C(z) = \begin{bmatrix} 1 & 0 \\ 2 - 9/2z + 7/2z^2 - z^3 & 1 \end{bmatrix}$$

$$D(z) = \text{diag}\left[1,\ z^{-2} - 9/2z^{-1} + 7 - 9/2z + z^2\right]$$

and $$F(z) = \begin{bmatrix} -2z^{-1} + 6 - 2z & -z^{-1} + 1 \\ 4z - 2z^2 & z \end{bmatrix} .$$

Notice that the factorization has been made in such a way that $C(z)$ is one-sided.

Step 2 consists of forming the left-standard factorization from the Smith-McMillan canonical form. We write

$$D(z) = \text{diag}\left[1,\ (1 - 1/2z)(1 - z)\ 2\ (1 - z^{-1})(1 - 1/2z^{-1})\right]$$

$$= D^{+}\ \theta\ \theta_{*}\ D^{+}_{*}$$

where $D^{+} = \text{diag}\ \left[1,\ (1 - 1/2z)\right]$

$\theta = \sqrt{2}\ \text{diag}\ \left[1,\ (1 - z)\right]$.

Now, the left standard factorization $R = P\ \Delta\ Q$ is given by setting

$$P = C\ D^{+}$$

$$= \begin{bmatrix} 1 & 0 \\ 2 - 9/2z + 7/2z^2 - z^3 & 1 - 1/2z \end{bmatrix},$$

$$Q = D^{+}_{*}\ F$$

$$= \begin{bmatrix} -2z^{-1} + 6 - 2z & -z^{-1} + 1 \\ -2 + 5z - 2z^2 & -1/2 + z \end{bmatrix}$$

and $\Delta = \theta\ \theta_{*}$

Step 3 involves extracting the elementary reverse-symmetric polynomial $M(z)$ from the left-standard factorization. We have

$$R = P\ \Delta\ Q$$

$$= P\ M^{-1}\ \Delta\ P_{*}$$

where $M^{-1} = P^{-1}\ Q_{*}$ (see equations 4.2-23 and 4.2-25). Thus we compute M^{-1}

$$M^{-1} = \begin{bmatrix} -2z^{-1} + 6 - 2z & -2z^{-2} + 5z^{-1} - 2 \\ 2z^{-1} - 9 + 14z - 9z^2 + 2z^3 & 4z^{-2} - 16z^{-1} + 25 - 16z + 4z^2 \end{bmatrix}$$

Now, we also had

$$R = P\,\theta\,\theta^{-1}\,M^{-1}\,\theta\,\theta_*\,P_*$$

$$= P\,\theta\,\tilde{M}\,\theta_*\,P_*$$

where $$\tilde{M} = \theta^{-1}\,M^{-1}\,\theta$$

$$= \begin{bmatrix} -2z^{-1}+6-2z & -2z^{-2}+7z^{-1}-7+2z \\ 2z^{-1}-7+7z-2z^2 & 4z^{-2}-16z^{-1}+25-16z+4z^2 \end{bmatrix}$$

This is indeed an elementary reverse-symmetric polynomial matrix. The next step is the factorization of this matrix into the form $\tilde{M} = S\,S_*$. A very similar elementary auto-correlation matrix was factored in the Example 4.2-1. The result is

$$S(z) = \begin{bmatrix} -z^{-1}+1 & 1 \\ 2z^{-1}-3+2z & -2+2z \end{bmatrix}.$$

The factorization of $R(z)$ is given by

$$R(z) = A(z)\;A_*(z)$$

where $$A(z) = P(z)\;\theta(z)\;S(z)$$

$$= \sqrt{2}\begin{bmatrix} -z^{-1}+1 & 1 \\ 1/2 - 1/2z & 1/2z - 1/2z^2 \end{bmatrix}.$$

This is not a one-sided factorization as we had obtained in the spectral decomposition example; however, this solution can be forced to be one-sided by post-multiplying it be the proper unitary matrix.

4.23 Approximate Factorization Methods

The factorization methods outlined in the last two sections are exact, but are difficult computationally. The spectral approach suffers from the well-known difficulties of determining the zeros of polynomials. The Smith-McMillan canonical form approach is complicated and has not yet been refined to give one-sided factorizations.

In this section we will discuss two approximate schemes for determining the Taylor expansion of the inverse operator $\mathfrak{T}^{-1}(z)$ from the autocorrelation $R(z)$. Both of these techniques depend upon the fact that

$$\mathfrak{T}^{-1}(z)\, R(z) \;=\; A_0\, \hat{A}_{0*}\, \mathfrak{T}_*(z)$$

is one-sided (specifically, right-sided). Thus if we have an approximation

$$A_1(z) \;\approx\; \mathfrak{T}^{-1}$$

then we can improve the approximation be examining the non-zero right side of

$$A_1(z)\, R(z) \;=\; \mathcal{E}(z)\ .$$

The first technique is a recursive method that may be associated with least-squares. It was advanced independently by Robinson (1963a) and J. P. Burg (personal communication) based on the work of Levinson (1947).

The second technique is an iterative method based on the vector projections of linear algebra. It was developed by Wiener and Masani (1957 and 19) and by Masani (1960). Wunsch (1965) has also published a heuristic interpretation of the projection technique. Both computational schemes have been programmed and tested for computational efficiency. For all cases tested, the projection technique was an order of magnitude slower than the least-squares recursive method.

4.231 Least-squares

The approximate least-squares wavelet, A_M, of degree M

$$A_M = A_{0,M} + A_{1,M}\, z + \ldots + A_{M,M}\, z^M$$

has the properties that

a) $A_{0,M} = 1_n$, and

b) $\mathcal{E}_{i,M} = 0 \quad i = 1, \ldots, M$ where

$$\mathcal{E}_M(z) = A_M(z)\, R(z) \,. \tag{4.2-27}$$

If we write out the equations for $\mathcal{E}_{i,M}$ $i = 1, \ldots, M$

$$\begin{aligned} A_{M,M} R_{-M+1} + \ldots + A_{1,M} R_0 &= -A_{0,M} R_1 \\ A_{M,M} R_{-M+2} + \ldots + A_{1,M} R_1 &= -A_{0,M} R_2 \\ \vdots \qquad\qquad \vdots \quad &\ \ \vdots \qquad \vdots \\ A_{M,M} R_0 \quad + \ldots + A_{1,M} R_{M-1} &= -A_{0,M} R_M \end{aligned} \tag{4.2-27a}$$

we see that this defines a set of nM simultaneous equations which can always be solved for $A_M(z)$. We will provide a recursive technique for extending the length (degree) of $A_M(z)$ without directly resolving the set of simultaneous equations given above.

For the recursion we sill need a second wavelet

$$B_M(z) = B_{0,M} + B_{1,M} z + \dots + B_{M,M} z^M$$

that has the properties

a) $B_{0,M} = 1_n$, and

b) $\mathcal{E}_{i,M} = 0 \quad i = -1, \dots, -M$ where

$$B_M(1/z)\, R(z) = \mathcal{E}_M(z) \qquad (4.2\text{-}28)$$

and where $\mathcal{E}_{i,M}$ $i = -1, \dots, -M$ is given by

$$\begin{aligned}
B_{1,M} R_{-M+1} + \dots + B_{M,M} R_0 &= -B_{0,M} R_{-M} \\
B_{1,M} R_{-M+2} + \dots + B_{M,M} R_1 &= -B_{0,M} R_{-M+1} \\
\vdots \qquad\qquad\quad \vdots \quad & \;\;\vdots \qquad \vdots \\
B_{1,M} R_0 \quad + \dots + B_{M,M} R_{M-1} &= -B_{0,M} R_{-1} \quad .
\end{aligned} \qquad (4.2\text{-}28a)$$

In the spectral factorization of the autocorrelation (see section 4.222) we obtained the minimum delay wavelet $\tilde{\mathfrak{a}}(z)$ by choosing all of the roots outside the unit circle. We could also have formed a maximum delay wavelet $\tilde{B}(z)$ by choosing all of the roots inside the unit circle such that $R(z) = \tilde{B}(z)\, B_0\, B_0'\, \tilde{B}_*(z)$. Thus $\tilde{B}_*(z)$ is minimum delay. The wavelets $A_M(z)$ and $B_M(z)$ are the least-squares approximations to the wavelets $\tilde{\mathfrak{a}}^{-1}(z)$ and $\tilde{B}_*^{-1}(z)$.

It can be shown (Robinson, 1963b) that $\mathcal{A}_M(z)$ and $\mathcal{B}_M(z)$ are also minimum delay.

Notice that if weight and add $\mathcal{A}_M(z)$ and $\mathcal{B}_M(z)$, we find

$$K\,\mathcal{A}_M(z) + z^{M+1}\,\tilde{K}\,\mathcal{B}_M(1/z)\,R(z) = K\,\mathcal{E}_M(z) + z^{M+1}\,\tilde{K}\,\tilde{\mathcal{E}}_M(z) \quad .$$

Thus, if we choose K and $\tilde{K}$ so that

$$\text{a)}\quad K = 1_n\ ,\quad \mathcal{E}_{M+1,M} + \tilde{K}\,\tilde{\mathcal{E}}_{0,M} = 0$$

$$\therefore\ \text{define}\quad K_{a,M} = -\mathcal{E}_{M+1,M}\,\tilde{\mathcal{E}}_{0,M}^{-1}\ ,$$

or

$$\text{b)}\quad \tilde{K} = 1_n\ ,\quad K\,\mathcal{E}_{0,M} + \tilde{\mathcal{E}}_{-M-1,M} = 0$$

$$\therefore\ \text{define}\quad K_{b,M} = -\tilde{\mathcal{E}}_{-M-1,M}\,\mathcal{E}_{0,M}^{-1}\ ,$$

we find a recurrence relationship

$$\mathcal{A}_{M+1}(z) = \mathcal{A}_M(z) + z^{M+1}\,K_{a,M}\,\mathcal{B}_M(1/z)$$

$$\mathcal{B}_{M+1}(z) = \mathcal{B}_M(z) + z^{M+1}\,K_{b,M}\,\mathcal{A}_M(1/z)\ .$$

These polynomials are multivalued counterparts of the polynomials orthogonal of the unit circle treated by Geronimus (1960) and Szego (1959).

Likewise note that

$$\mathcal{E}_{M+1}(z) = \mathcal{E}_M(z) + z^{M+1}\,K_{a,M}\,\tilde{\mathcal{E}}_M(z)$$

$$\tilde{\mathcal{E}}_{M+1}(z) = \tilde{\mathcal{E}}_M(z) + z^{-M-1}\,K_{b,M}\,\mathcal{E}_M(z)\ .$$

There are two other relationships that are important computationally. First, if $R(z)$ is symmetric (see Section 4.31) then $A_M(z) = B_M(z)$ for obvious reasons. Second, for all cases

$$\mathcal{E}_{M+1,M} = \mathcal{T}'_{-M-1,M} .$$

Proof. (According to J. P. Burg, personal communication.) We first map equations 4.2-27a and 4.2-28a into matrix notation:

$$[A_{0,M}, \ldots, A_{M,M}, 0] \, \mathcal{R}_{M+1} = [\mathcal{E}_{0,M}, 0, \ldots, 0, \mathcal{E}_{M+1,M}]$$

$$[0, B_{M,M}, \ldots, B_{0,M}] \, \mathcal{R}_{M+1} = [\mathcal{T}_{-M-1,M}, 0, \ldots, 0, \mathcal{T}_{0,M}] \tag{4.2-29}$$

where

$$\mathcal{R}_{M+1} = \begin{bmatrix} R_0 & R_1 & \cdots & R_{M+1} \\ R_{-1} & R_0 & \cdots & R_M \\ \vdots & \vdots & \ddots & \vdots \\ R_{-M-1} & R_{-M} & \cdots & R_0 \end{bmatrix} .$$

The solution to the next recursion will then give

$$[A_{0,M+1}, \ldots, A_{M,M+1}, A_{M+1,M+1}] \, \mathcal{R}_{M+1} = [\mathcal{E}_{0,M+1}, 0, \ldots, 0]$$

$$[B_{M+1,M+1}, B_{M,M+1}, \ldots, B_{0,M+1}] \, \mathcal{R}_{M+1} = [0, \ldots, 0, \mathcal{T}_{0,M+1}] . \tag{4.2-30}$$

Equations 4.2-30 show that the first and last rows of $\mathcal{R}^{-1}_{M+1}$ are

$$\mathcal{E}^{-1}_{0,M+1} \, [A_{0,M+1}, \ldots, A_{M+1,M+1}]$$

and $$\mathcal{T}^{-1}_{0,M+1} \, [B_{M+1,M+1}, \ldots, B_{0,M+1}] .$$

Since $\mathcal{R}$ is symmetrical, $\mathcal{R}^{-1}$ is also symmetrical, and therefore

$$\left(\mathcal{E}_{0,M+1}^{-1} \ \mathcal{A}_{M+1,M+1}\right)' = \mathcal{T}_{0,M+1}^{-1} \ \mathcal{B}_{M+1,M+1} \qquad (4.2\text{-}31)$$

(note that $\mathcal{E}_{0,M}$ and $\mathcal{T}_{0,M}$ must be non-singular for all M since $\mathcal{R}$ is non-singular.)

There exists an $n \times n$ matrix Q such that if $R_{M+1} = Q$, then $\mathcal{E}_{M+1,M} = 0$. If $R_{M+1} = Q$, then $\mathcal{A}_{M+1} = \mathcal{A}_M$. However, since

$$\mathcal{B}_{M+1,M+1} = \mathcal{T}_{0,M+1} \ \mathcal{A}'_{M+1,M+1} \ \mathcal{E}_{0,M+1}^{-1}$$

(from equation 4.2-31) if $\mathcal{A}_{M+1,M+1} = 0$, then $\mathcal{B}_{M+1,M+1} = 0$ and $\mathcal{B}_{M+1}(z) = \mathcal{B}_M(z)$. For an arbitrary R_{M+1} we can write

$$R_{M+1} = (R_{M+1} - Q) + Q \quad .$$

If we substitute this into equations 4.2-29 we find

$$\mathcal{E}_{M+1,M} = R_{M+1} - Q$$

$$\mathcal{T}_{-M-1,M} = (R_{M+1} - Q)' \qquad \text{since} \quad R_{-M} = R'_M$$

$$= \mathcal{E}'_{M+1,M} \quad . \qquad \text{Q.E.D.}$$

The left part of the flow diagram in Figure 5.1-6 shows the steps involved in the recursive computations.

4.232 Wiener-Masani Projections

The projection technique for factoring a matrix valued autocorrelation involves the theory of linear algebra. It will be convenient for our development to consider vectors of matrices rather than polynomials with matrix coefficients, i.e., we will work in the time domain rather than in the z-transform domain.

Let us begin by defining the elements $\underline{A}_t$ of a complete subset M of the linear space of vectors of matrices S

$$\underline{A}_t = A_t, A_{t+1}, \ldots, A_{t+n}, \ldots$$

That is, each element $\underline{A}_t$ of S is a time shifted reproduction of the minimum delay operator $\underline{A}_0$ (Robinson, p. 75, 1962).

We shall also define an inner product

$$(\underline{A}_i, \underline{A}_j) = \sum_{s=1}^{\infty} \underline{A}_{s+j-i} \underline{A}_s' \qquad (4.2\text{-}32)$$

$$= R_{i-j}$$

This definition conforms to all the requirements for a linear product:

a) $(\underline{A}_i, \underline{A}_j)' = (\underline{A}_j, \underline{A}_i)$,

b) $(\underline{A}_i, \underline{A}_i) > 0$ if $\underline{A}_i \neq 0$,

c) $(a\,\underline{A}_i, \underline{A}_j) = a(\underline{A}_i, \underline{A}_j)$ where a is a scalar,

d) $(\underline{A}_i + \underline{A}_j, \underline{A}_k) = (\underline{A}_i, \underline{A}_k) + (\underline{A}_j + \underline{A}_k)$.

Spaces which are linear, complete, and contain an inner product are called closed linear manifolds. Neumann (p. 51, 1950) states the following definition and theorem concerning projections:

Definition 4.2-2. If M is a closed linear manifold in S, if $\underline{B} \in S$, and if $\underline{B} = \underline{B}_1 + \underline{B}_2$, where $\underline{B}_1 \in M$ and $\underline{B}_2 \in -M$. Then $\underline{B}_1$ is called the projection of $\underline{B}$ on M and the operation of projecting $\underline{B}$ on M is denoted by $P_M \underline{B} = \underline{B}_1$.

Theorem 4.2-2. A necessary and sufficient condition that an operator E be a projection P is that

a) E is single valued, linear with domain in S,

b) $(E\,\underline{A}_i, \underline{A}_j) = (\underline{A}_i, E\,\underline{A}_j)$ for every $\underline{A}_i$ and $\underline{A}_j$ in S,

c) $E\,E = E$.

M is uniquely defined by E.

Finally Neumann (1950) states the crucial projection theorem.

Theorem 4.2-3. (Projection). If $E_1 = P_{M_1}$ and $E_2 = P_{M_2}$, then the sequence of operators $E_1, E_2E_1, E_1E_2E_1, E_2E_1E_2E_1$ has a limit E; the sequence $E_2, E_1E_2, E_2E_1E_2, E_1E_2E_1E_2$ has the same limmit E; and $E = P_{M_1M_2}$.

Wiener and Masani (p. 106, 1958) state the following corollary.

Corollary 4.2-3. If F is the projection on $M \perp M_1 \cup M_2$, then

$$F = I - E_1 - E_2 + E_1 E_2 + E_2 E_1 - \ldots$$

The convergence being in the strong sense.

Wiener and Masani (1958) then give a lengthy development to generalize this equation to include an infinite number of projections. They find

$$F = I - \sum_{i=1}^{\infty} E_i + \sum_{i,j=1}^{\infty} E_i E_j - \sum_{i,j,k=1}^{\infty} E_i E_j E_k + \ldots \qquad (4.2\text{-}33)$$

where the projection operator is defined by the inner product

$$E_k = (\quad , \underline{A}_k) \underline{A}_k \ .$$

The normalization that they use to insure convergence is the requirement that $(R_k)_{i,i} = \delta_k$. To make this normalization, we factor each of the diagonal terms such that

$$(R(z))_{i,i} = a_i(z)\, r_i\, a_{i*}(z) \qquad i = 1, \ldots, n$$

where $a_i(z)$ is normalized so that its constant term is equal to one. Now let

$$a(z) = \operatorname{diag}\left[a_1(z), a_2(z), \ldots, a_n(z)\right]$$
$$\sqrt{r} = \operatorname{diag}\left[\sqrt{r_1}, \sqrt{r_2}, \ldots, \sqrt{r_n}\right],$$

then the normalized autocorrelation is

$$\hat{R}(z) = \sqrt{r}^{\,-1} a^{-1}(z)\; R(z)\; a_*^{-1}(z) \sqrt{r}^{\,-1}.$$

Our problem then, is to find the vector, A , that is orthogonal to each of the $\underline{A}_t$ $t = 1, \ldots, \infty$. Thus we substitute $\underline{A}$ into the projection sequence 4.2-33:

$$F\,\underline{A} = I\,\underline{A} - \sum_{i=1}^{\infty} E_i\underline{A} + \sum_{i,j=1}^{\infty} E_iE_j\underline{A} - \sum_{i,j,k=1}^{\infty} E_iE_jE_k\underline{A} + \ldots$$

$$= I\,\underline{A} - \Sigma\,(\underline{A},\underline{A}_i)\underline{A}_i + \Sigma\,(\underline{A},\underline{A}_j)(\underline{A}_j,\underline{A}_i)\underline{A}_i$$

$$- \Sigma\,(\underline{A},\underline{A}_k)(\underline{A}_k,\underline{A}_j)(\underline{A}_j,\underline{A}_i)\underline{A}_i + \ldots$$

$$= I\,\underline{A} + \sum_{i=1}^{\infty}\left[-R_{-i} + \sum_{j=1}^{\infty}\hat{R}_{-j}\hat{R}_{j-i} - \sum_{j,k=1}^{\infty}\hat{R}_{-k}\hat{R}_{j-k}\hat{R}_{i-j} + \ldots\right]\underline{A}_i$$

Therefore the orthogonal operator

$$\hat{A}_\infty = A_{0\infty},\; A_{1\infty},\; \ldots,\; A_{n\infty},\; \ldots$$

is given in terms of the autocorrelation only:

$$A_{0\infty} = 1_n$$

$$A_{i\infty} = -\hat{R}_{-i} + \sum_{j=1}^{\infty}\hat{R}_{-j}\hat{R}_{j-i} - \sum_{k,j=1}^{\infty}\hat{R}_k\hat{R}_{j-k}\hat{R}_{i-j} + \ldots$$

For computational purposes we define a vector $e_{i,j}$ which represents the $j^{\underline{th}}$ projection of the $i^{\underline{th}}$ term of $\hat{A}_i$. Clearly then, the first projection is

$$\hat{A}_{i,0} = \delta_i$$

$$e_{i,1} = -\hat{R}_{-i} \qquad i = 1, 2, \ldots$$

BLANK PAGE

and the $(j + 1)^{\underline{th}}$ projection is

$$\epsilon_{i,j+1} = -\epsilon_{1,j} \hat{R}_i - \epsilon_{2,j} \hat{R}_{i-1} - \ldots - \epsilon_{i+1} \hat{R}_0 - \ldots$$

$$\lambda_{i,j+1} = \lambda_{i,j} + \epsilon_{i,j+1} \qquad i = 1, 2, \ldots$$

The iteration is continued until $\epsilon_{i,j}$ becomes smaller than some given value.

The only problem that remains is that of scaling $\hat{\lambda}_{i\infty}$ so that is represents the inverse $\mathfrak{a}^{-1}$ (see equation 4.2-31). We have

$$\hat{R}(z) = \lambda_{\infty}^{-1}(z) \hat{A}_0 \hat{A}_0' \lambda_{\infty *}^{-1}(z)$$

and $$R(z) = \mathfrak{a}(z) A_0 A_0' \mathfrak{a}_*(z)$$

from the spectral factorization of a non-degenerate autocorrelation. But we had

$$\hat{R}(z) = \sqrt{r}^{-1} a^{-1}(z) R(z) a_*^{-1}(z) \sqrt{r}^{-1}$$

$$\therefore \mathfrak{a}(z) A_0 A_0' \mathfrak{a}_*(z) = a(z)\sqrt{r} \; \lambda_{\infty}^{-1}(z)\hat{A}_0\hat{A}_0'\lambda_{\infty}^{-1}(z)\sqrt{r} \; a_*(z)$$

and $$\mathfrak{a}^{-1}(z) = \sqrt{r} \; \hat{\lambda}_{\infty}(z) \sqrt{r}^{-1} a^{-1}(z) \quad .$$

This development is intended to be a quick summary of the projection technique. It is by no means rigorous. The step from the Neumann theorem to the actual projection definitions that converge is certainly not immediate. One must either follow the path that Wiener and Masani (1958) did, or generalize Neumann's scalar theorems to matrix space.

4.3 Multi-Dimensional Autocorrelation

Since the properties of matrix-valued multi-dimensional autocorrelations follow directly from those of scalar autocorrelations we will limit the discussion here to scalar values.

The properties of multi-dimensional auto-correlations have the same features that we observed for one-dimensional correlations.

Let $R(\underline{z}) = R(z, z_1, \ldots, z_k)$ be a $(k+1)$-dimensional scalar autocorrelation function, then

a) $R(\underline{z})$ is centro-symmetric; that is,
$R(z, z_1, \ldots, z_k) = R(1/z, 1/z_1, \ldots, 1/z_k)$.

b) $R(\underline{z})$ is non-negative definite on the unit hyper-circle $|z\ z_1 \ldots z_k| = 1$.

4.31 Mapping into One-Dimensional Representation

Perhaps the most important thing that we will establish here is the mapping of multi-dimensional autocorrelations into a matrix representation. We will begin by making the transformation in terms of the mapping operators defined in Section 3.43 and then proceed to direct transformation from multi-dimensional to matrix valued autocorrelations.

The mapping is illustrated in Figure 4.3-1. The configuration of the autocorrelation matrix is the same that we would obtain if we had treated the process, $x(\underline{z})$, as three separate wavelets, $x_1(z)$, $x_2(z)$, and $x_3(z)$ and defined the autocorrelations as

$$\begin{bmatrix} x_1(z) \\ x_2(z) \\ x_3(z) \end{bmatrix} \begin{bmatrix} x_{1*}(z), & x_{2*}(z), & x_{3*}(z) \end{bmatrix} = \begin{bmatrix} r_0(z) & r_1(z) & r_2(z) \\ r_{-1}(z) & r_0(z) & r_3(z) \\ r_{-2}(z) & r_{-1}(z) & r_0(z) \end{bmatrix} .$$

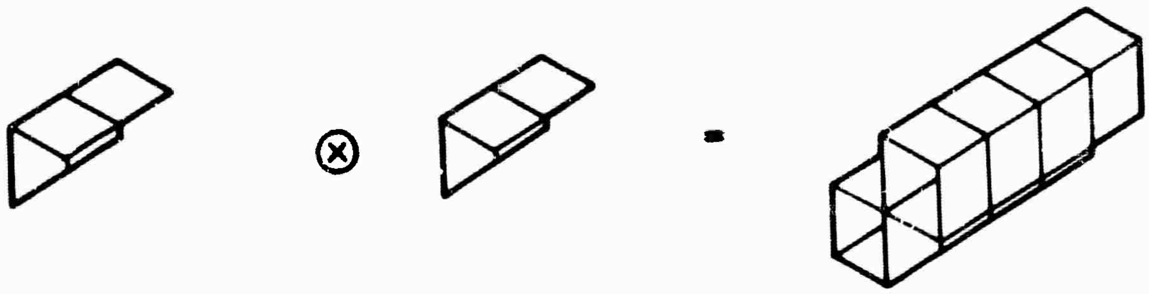
However, for each spatial lag we must take the sum of the correlations of all the wavelets that overlap at that lag. The matrices shown in Figure 4.3-1 fulfill this requirement. They are defined formally by

$$\mathrm{MAP2}(x,\ x(z)) \cdot \mathrm{MAP2}(x,\ x(z))' = R(z) .$$

Frequently, in practical applications we are presented with the multi-dimensional autocorrelation $r(\underline{z})$ and we wish to map it directly into the matrix representation $R(z)$. The procedure here is very similar to that taken above. We map the spatial positions of the process into a vector and form the symbolic product

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\ \vdots & & \ddots & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} \end{bmatrix}$$

$$x_{t,i_1,i_2} \quad \otimes \quad x_{t,i_1,i_2} \quad = \quad r_{t,i_1,i_2}$$



$\otimes$ = autocorrelation

z-Transform in preferred direction and order the indices.

$$\begin{array}{cc} x_1(z) & x_3(z) \\ x_2(z) & \end{array} \qquad \begin{array}{cc} x_{1*}(z) & x_{3*}(z) \\ x_{2*}(z) & \end{array} \quad = \quad \begin{array}{ccc} & r_1(z) & r_3(z) \\ r_{-2}(z) & r_0(z) & r_2(z) \\ r_{-3}(z) & r_{-1}(z) & \end{array}$$

Map wavelets into matrix notation.

$$\begin{bmatrix} x_1 & x_2 & 0 & x_3 & 0 & 0 \\ 0 & x_1 & x_2 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} x_{1*} & 0 & 0 \\ x_{2*} & x_{1*} & 0 \\ 0 & x_{2*} & 0 \\ x_{3*} & 0 & x_{1*} \\ 0 & x_{3*} & x_{2*} \\ 0 & 0 & x_{3*} \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & r_2 \\ r_{-1} & r_0 & r_3 \\ r_{-2} & r_{-3} & r_0 \end{bmatrix}$$

Figure 4.3 - 1: Mapping of a multi-dimensional autocorrelation into a matrix representation.

Then the subscripts of each element $r_{i,j}$ define the spatial separation for that term. If the spatial process is stationary, that is, if $r_{i,j}$ depends only upon the spatial separation between position i and position j, then $R(z)$ is symmetric. This leads to a number of simplifications.

4.32 Methods of Factorization

Except for one method of factorization, all of the techniques that are used are made in terms of the matrix mapping of the autocorrelation. The fact that the autocorrelation matrix is symmetrical may lead to some important simplifications in some cases. For example, in the elementary autocorrelation matrix factorization, symmetricallity forces the algorithm to give a one-sided wavelet. Also, in the least-squares approximate technique the operator $\mathcal{A}_M(z) = \mathcal{B}_M(z)$.

It is instructive to consider the meaning of the minimum delay wavelets that one obtains from the matrix factorization. Each row of the matrix $A(z)$ will be a vector representation of a spatial minimum phase wavelet. This vector representation is the same as that used for mapping the original process. The origin of the wavelet is located at the spatial position corresponding to the diagonal term in the matrix. Thus the autocorrelation of a spatial process having n lattice points will produce n minimum phase wavelets; and each wavelet well have its origin at a different lattice point.

Occasionally in physical problems we know that the factorization should have zero-phase, i.e., should be symmetrical in all directions. For this case we may proceed as in the one-dimensional scalar case. Thus we need only evaluate the expression

$$a(e^{-i\underline{\omega}}) = \sqrt{r(e^{-i\underline{\omega}})} \quad .$$

That is, we find the cosine-transform of $r_{\underline{i}}$, take its square root, and retransform back to space-time.

5. LEAST-SQUARE FILTERING IN THE PRESENCE OF NOISE

In Chapter 4. a number of techniques were discussed for finding a minimum delay wavelet from a given autocorrelation. Of the techniques discussed, the least-squares approximation was found to be the best method in the sense of computational efficiency. In this chapter, the least-squares decompositional method will be extended to include signal shaping (in addition to straight prediction) in the presence of random noise with a given coherency. This approach will give an optimum linear operator for a given length and output lag.

The normal equations for the one-dimensional matrix-valued process only will be developed here. As was illustrated in the last two chapters, all other dimensionalities are but a special case for this representation.

5.1 Derivation and Recursive Solution of the Normal Equations

The solution of the problem of determining the optimum least-squares linear operator is based upon the following assumptions:

a) The known $n \times m$ matrix-valued signal S_t is the additive combination of K uncorrelated stationary random processes $\mathfrak{S}_{i,t}$.

b) The $n \times m$ matrix-valued noise N_t is a random process with zero mean, $E \left\langle N_t \right\rangle = 0_{n,m}$, and known covariance, $E \left\langle N_i \, N_j' \right\rangle$.

c) The observed random process, X_t , is the additive combination of the signal and the noise

$$X_t = S_t + N_t \; .$$

d) The observed random process is convolved with an undetermined $\ell \times n$ matrix-valued wavelet F_i $i = 1, \ldots, M$ to obtain the $\ell \times m$ matrix-valued actual output Y_t .

e) The $\ell \times m$ matrix-valued desired output, D_t , is the additive combination of K independent desired outputs $\tilde{D}_{i,t}$ where $\tilde{D}_{i,t}$ is uncorrelated with $\tilde{X}_{j,t}$, $i \neq j$, i.e.

$$E \left\langle \tilde{D}_{i,t+\tau} \, \tilde{X}_{j,0} \right\rangle = 0 \quad i \neq j \; .$$

5.12 Normal Equations

The linear least-squares operator wavelet is determined by requiring that the norm of the difference between the actual output and the desired output is minimum for all time. That is, we require that ε_0 be minimized, where

$$\begin{aligned} \varepsilon_0 &= E \, ((\, e_t \,)) \\ &= E \left\langle \operatorname{tr} e_t \, e_t' \right\rangle , \end{aligned}$$

$$e_t = D_t - Y_t \quad ,$$

and E stands for expected value.

To find the minimum, we take the derivative of $\mathcal{E}_0$ with respect to the coefficients of the wavelet F_j $j = 1, \ldots, M$ and set it equal to zero. Thus

$$\frac{\partial \mathcal{E}_0}{\partial F_j} = 0 \qquad j = 1, \ldots, M$$

implies that the error e_i is normal to the input X_{i-j+1}:

$$E \left\langle e_i X'_{i-j+1} \right\rangle = 0 \qquad j = 1, \ldots, M \, .$$

This orthogonality was the basis for the development of the Wiener-Masani projections (see Section 4.232). This is also the origin of the name "normal equations."

Now, let us expand the normal equations:

$$\begin{aligned} E \left\langle e_i X'_{i-j+1} \right\rangle &= 0 \qquad\qquad j = 1, \ldots, M \\ &= E \left\langle \left(D_i - \sum_{k=1}^{M} F_k X_{i-k+1}\right) X'_{i-j+1} \right\rangle \end{aligned}$$

Also, we have $X_i = S_i + N_i$ and $E \left\langle N_i \right\rangle = 0$. Thus the normal equations have the form

$$\begin{aligned} E \Big\langle D_i & \left(S_{i-j+1} + N_{i-j+1}\right) \\ & - \sum_{k=1}^{M} F_k \left(S_{i-k+1} + N_{i-k+1}\right)\left(S'_{i-j+1} + N'_{i-j+1}\right) \Big\rangle = 0 \end{aligned}$$

$$E \left\langle \sum_{k=1}^{M} F_k \left(S_{i-k+1} S'_{i-j+1} + N_{i-k+1} N'_{i-j+1}\right) \right\rangle = E \left\langle D_i S'_{i-j+1} \right\rangle$$

$$\sum_{k=1}^{M} F_k \; E\left\langle S_{j-k} \, S_0' \right\rangle + E\left\langle N_{j-k} \, N_0' \right\rangle = E\left\langle D_{j-1} \, S_0' \right\rangle$$

for $j = 1, \ldots, M$. From the assuptions we see that

$$E\left\langle S_{j-k} \, S_0' \right\rangle = \sum_{i=1}^{K} E\left\langle \mathfrak{S}_{i,j-k} \, \mathfrak{S}_{i,0}' \right\rangle$$

$$E\left\langle D_{j-k} \, S_0' \right\rangle = \sum_{i=1}^{K} E\left\langle \mathfrak{D}_{i,j-k} \, \mathfrak{S}_{i,0}' \right\rangle .$$

Therefore the autocorrelation of S_t is the sum of the autocorrelations of $\mathfrak{S}_{i,j}$. If we define an autocorrelation

$$R_i = E\left\langle S_i \, S_0' \right\rangle + E\left\langle N_i \, N_0' \right\rangle$$

and a cross-correlation

$$G_i = E\left\langle D_i \, S_0' \right\rangle ,$$

then the normal equations may be written in the simple form

$$\sum_{k=1}^{M} F_i \, R_{j-k} = G_{j-1} , \quad j = 1, \ldots, M . \qquad (5.1\text{-}1)$$

We may also obtain a simple form for the expected error $\mathcal{E}_0$:

$$\mathcal{E}_0 = \operatorname{tr} \, E\left\langle e_i e_i' \right\rangle$$

$$= \operatorname{tr} \, E\left\langle e_i \, (D_i - \sum_{k=1}^{M} F_k \, X_{i-k+1})' \right\rangle$$

$$= \operatorname{tr} \, E\left\langle e_i \, D_i' \right\rangle$$ since e_i is normal to X_{i-k+1} for $k = 1, \ldots, M$

$$\rho_0 = \operatorname{tr} E\left\langle D_i D_i' - \sum_{k=1}^{M} F_k X_{i-k+1} D_i' \right\rangle$$

$$\rho_0 = \operatorname{tr} E\left\langle D_i D_i' - \sum_{k=1}^{M} F_k G_{k-1}' \right\rangle . \qquad (5.1\text{-}2)$$

The normal equations that were obtained above are very closely related to the simultaneous equations defined in Section 4.231 for $A_{i,M}$

$$\sum_{k=0}^{M} A_{k,M} R_{j-k} = \rho_{j,M} \qquad j = 0, \ldots, M$$

where $A_{0,M} = I$, and $\rho_{j,M} = 0$, $j = 1, \ldots, M$. For if there is no noise, $N_i = 0$ for all i , and if we make the desired output equal to the input one lag ahead in time (that is, we ask the filter to predict the next value of the process) then the $n \times n$ filter wavelet F_i $i = 1, \ldots, M$ is identical to the wavelet $-A_{i,M}$ $i = 1, \ldots, M$ and the equations above for $j = 1, \ldots, M$ are the same as equation 5.1-1. Also, the equation for the expected error $\rho_{0,M}$ is the same as equation 5.1-2.

The filter $-A_{i,M}$ $i = 1, \ldots, M$ is called the least-squares approximate prediction filter with unit prediction distance. The difference between the actual value and the predicted value is the output of $A_{i,M}$ $i = 0, \ldots, M$. Thus it is called the prediction error filter or the foresight error filter.

On the other hand, had we first reversed the direction of time and then solved for the prediction error filter as above, we would have obtained the filter $B_{i,M}$ $i = 0, \ldots, M$:

$$\sum_{k=0}^{M} B_{k,M} R_{k-j} = \mathcal{E}_{j,M} \qquad j = 0, \ldots, M$$

where $B_{0,M} = I$ and $\mathcal{E}_{j,M} = 0$ $j = 1, \ldots, M$. Since this filter actually predicts the past from future values of the process, it is called the hindsight error filter.

Example 5.1-1. Let us consider a symbolic two-dimensional scalar problem of signal shaping in the presence of noise. The problem is specified by the arrays illustrated in Figure 5.1-1. That is, we wish to design a filter which produces the desired output array when convolved with the signal array and which produces a zero output array when convolved with the noise array.

The actual design of the filter is based on the two-dimensional autocorrelations of the signal and of the noise, and on the two-dimensional crosscorrelation of the desired output with the signal. These arrays are shown in Figures 5.1-2 and -3.

These arrays are then mapped into matrix notation (see Chapter 4.31) and substituted into the normal equations. The recursive procedure described in the next section was then followed to compute the 20 x 20

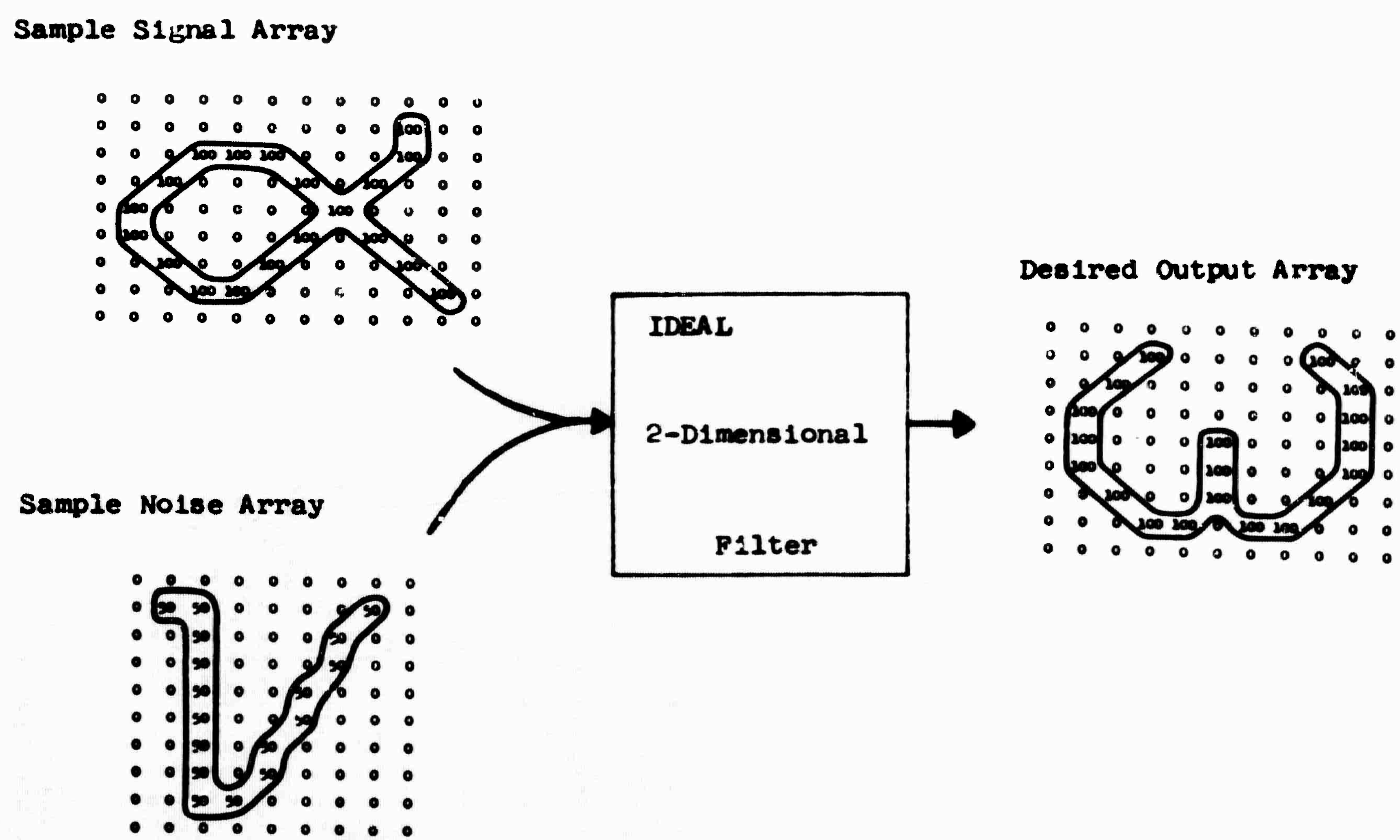
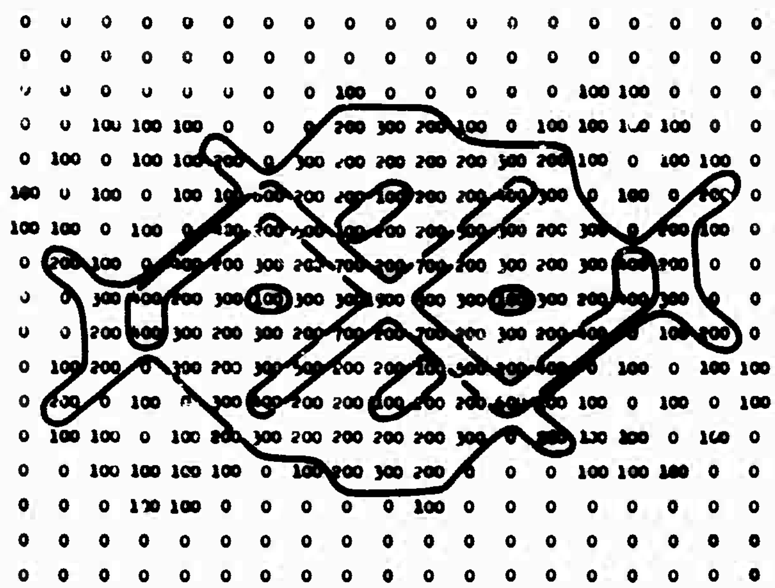


Figure 5.1 - 1: Sample arrays for designing least-squares filters.

Autocorrelation of the Signal Array



Autocorrelation of the Noise Array

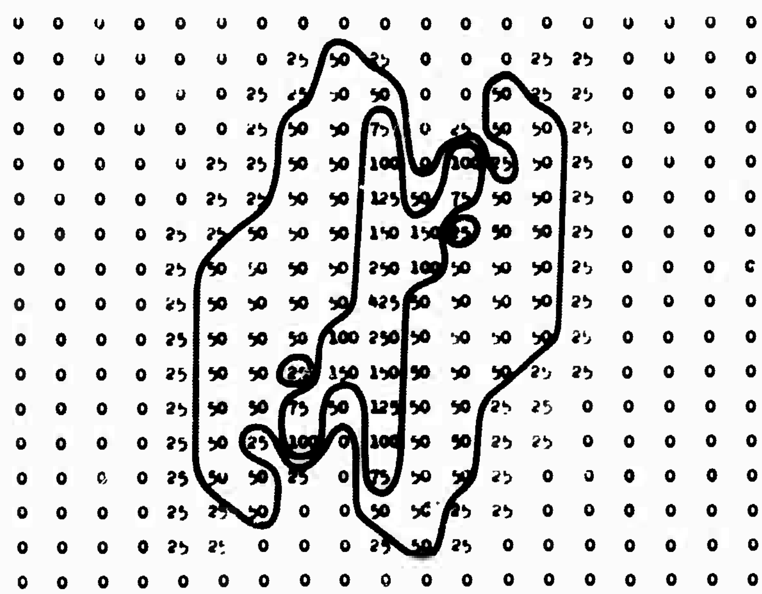


Figure 5.1 - 2: Complete 2-dimensional autocorrelations of the signal array and of the noise array.

term filter shown in Figure 5.1-4. This recursion was carried out by first extending the length of the filter in a positive direction and then extending it in a negative direction alternately until the final length was reached. After each extension, the expected error was computed. A plot of these values is also shown in Figure 5.1-4.

Finally, to gain a visual idea of the quality of the filter, it was convolved (in two dimensions) with the signal array and with the noise array to produce the two output arrays shown in Figure 5.1-5. Only the center portions of the convolutions are shown.

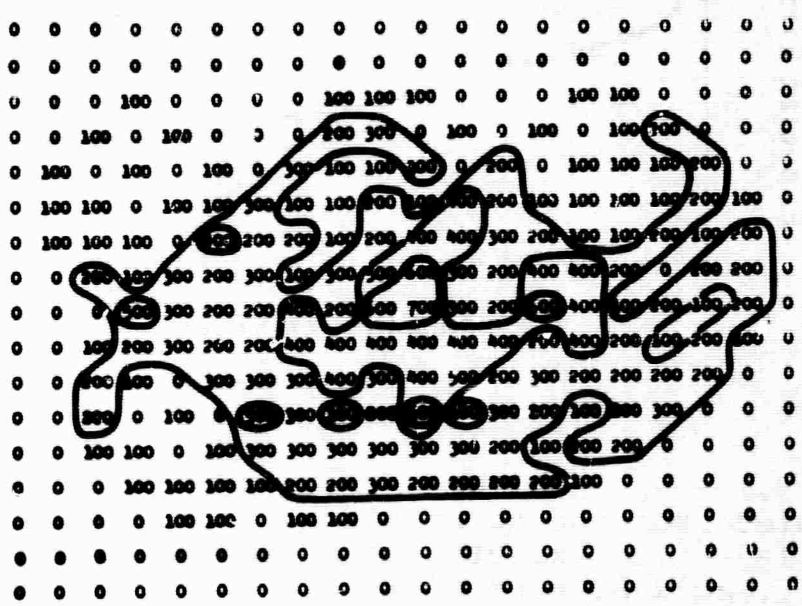


Figure 5.1 - 3: Complete crosscorrelation of the signal array with the desired output array.

-2	-4	[illegible]	5	[illegible]	-19	2	13	-4	-17	7	-0	-3	[illegible]	-5	-2	-19	15	-8	1
-1	3	2	-13	[illegible]	2[illegible]	6	-20	-13	11	-9	-30	[illegible]	38	2	3	-8	2	15	-18
13	4	7	17	2[illegible]	-12	-10	29	-10	-5	11	22	-27	15	11	6	6	-16	7	-20
-6	[illegible]	-4	3	-34	[illegible]	-2	-31	30	-23	-3	2	-12	-17	0	8	-14	-11	4	5
14	-17	[illegible]	34	[illegible]	-3	23	37	-5	-2	34	46	-19	-11	43	-3	27	-26	16	4
-7	2[illegible]	-14	-4[illegible]	15	-1[illegible]	-19	-71	[illegible]7	[illegible]	-65	32	23	-61	-59	11	68	-13	-23	13
9	-3	-1	1	-14	-9	-42	103	-20	-21	95	-73	33	-90	-12	29	1	63	-83	46
-47	[illegible]	[illegible]	-3	11	24	20	-69	-27	43	-100	26	32	-45	-1	-23	-7	53	-7	-2
-1	4	3[illegible]	47	-40	52	13	1	-70	-36	64	43	-15	34	-112	-27	0	0	33	-71
[illegible]	-3	1	14	1[illegible]	9	-3[illegible]	-25	-40	126	102	9	-39	105	62	16	-90	6	92	-82
-15	3	-19	113	2[illegible]	32	14	60	-90	60	183	6	-84	161	45	-70	41	-4	57	-42
-17	1[illegible]	-20	[illegible]	34	13	-60	52	-33	6	4	29	12	-74	60	-24	10	-4	10	2
12	-34	40	-14	-40	12	48	-11	111	-14	122	83	-50	5	60	3	38	29	-4	35
-14	-34	60	-[illegible]1	36	-104	114	-21	-26	-90	115	-145	15	-82	-44	52	50	-15	-23	2
-[illegible]	-9	70	-32	-49	-4[illegible]	61	-25	60	6	22	23	22	-45	-12	5	-2	-35	-62	-2
24	4	-47	-22	11	-57	-12	-21	42	133	31	55	108	82	-21	11	-43	2	-90	-23
2	29	3	15	14	52	-33	13	-13	-54	-98	-74	-81	-46	-30	2	20	14	1	-16
13	34	-80	16	-19	-3	-45	13	-59	9	-9	-8	-13	37	6	0	10	-12	13	1
-1[illegible]	12	2	-5	-[illegible]	26	2	24	21	25	19	4	-19	19	13	-17	-6	7	80	23
3	-6	-6	17	3	-4	5	-13	-16	-22	-15	-17	16	1	22	13	13	-9	-4	-4

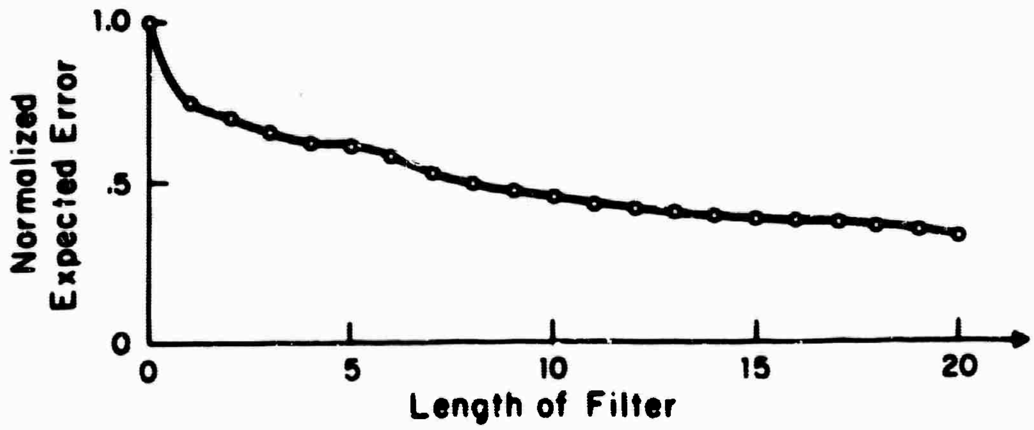


Figure 5.1 - 4: Optimum 20-row x 20-column filter computed from the correlations in Figures 5.1-2 and 5.1-3. The expected error is for 20 x 1, 20 x 2, ..., 20 x 20 term filters.

Convolution of Filter with Signal Array

-2	4	4	0	5	5	-2	-1	10	0	-1	4	4	0	4	4	-7	-1	-1
-2	-4	0	1	-5	-3	6	5	-2	0	4	0	-4	3	0	9	-6	-11	7
1	-1	-5	-6	5	3	0	-1	3	-2	-6	0	-5	-4	0	2	-7	4	5
-4	-1	-3	[illegible]	1	-9	2	-6	-6	3	5	-4	3	-12	6	-4	3	2	-4
1	2	6	-4	5	-3	-6	-16	10	-8	1	4	-11	2	1	-15	10	-2	2
6	2	2	-6	-2	-1	-3	[illegible]	3	-6	2	-5	62	3	-7	-10	6	9	-3
10	12	1	0	5	-5	[illegible]	1	14	2	6	13	13	[illegible]	-1	-4	2	6	-13
6	6	-2	-6	4	[illegible]	[illegible]	12	7	2	9	9	14	[illegible]	9	6	-3	-9	7
-3	4	-2	-6	17	[illegible]	13	2	19	61	6	16	3	[illegible]	14	6	1	8	1
5	-6	6	2	5	[illegible]	2	7	6	62	20	2	1	66	2	10	5	1	1
7	-12	16	5	-10	2	[illegible]	6	1	[illegible]	7	1	72	-1	12	1	14	6	4
-2	10	2	6	-6	5	7	71	70	[illegible]	75	72	[illegible]	1	12	-3	1	14	-3
-19	20	7	-4	12	4	4	-4	-9	1	4	-10	11	-4	4	3	-1	2	1
-5	0	1	5	2	3	8	-9	5	2	-5	0	6	0	3	3	-7	-5	6
-6	-2	-4	1	7	-1	-4	0	2	-6	2	5	3	4	9	-5	-6	-6	-5
10	-15	-4	4	2	3	2	5	9	3	1	1	7	-4	-5	-3	2	-6	1
2	1	-2	-2	3	-5	2	2	0	-4	2	-5	-6	0	-1	2	6	2	0

Convolution of Filter with Noise Array

0	-2	-2	2	3	-2	0	2	[illegible]	-3	5	2	4	-3	0	-3	1	3	-2
0	-3	3	0	2	3	-2	-2	2	-5	-2	11	2	-3	-5	2	5	-8	2
2	0	-5	-6	2	-2	-5	6	-4	-7	2	[illegible]	[illegible]	-1	-7	3	2	1	0
-1	3	-1	-5	-5	3	4	5	3	-16	-6	1	6	-4	-3	-3	-4	5	-2
-2	3	-7	-4	3	14	10	-1	3	-1	-1	7	-4	2	1	-3	3	-5	[illegible]
0	2	3	-5	2	24	17	5	8	9	5	10	-9	[illegible]	24	-7	-2	1	[illegible]
3	0	-2	7	-5	20	13	3	-7	6	20	-8	-7	16	6	-3	-4	5	4
-4	11	-7	15	-5	22	20	5	-5	23	15	-7	12	9	11	-5	1	10	-3
-2	4	6	9	-9	7	11	6	-1	16	0	4	1	5	9	-11	9	6	-7
-1	-1	17	8	-9	-6	35	1	9	15	13	-2	0	0	-2	-4	10	-4	-5
-5	-2	16	-2	4	2	35	25	20	12	19	-2	5	2	1	-3	6	-10	-2
-1	5	8	-2	-11	12	25	5	4	22	6	6	7	-9	3	2	-2	-4	-2
5	2	2	0	-3	9	21	7	3	11	2	-1	4	-1	3	5	-6	-1	-2
4	-4	3	-7	12	4	10	0	19	-6	-4	4	2	1	3	1	-2	-3	0
5	-12	6	-7	15	1	0	2	13	-11	2	2	0	5	2	-3	-1	-1	-1
0	-8	0	-1	6	-6	7	2	-4	-2	-5	-2	2	1	-5	-3	-4	-1	0
-3	-6	-1	3	1	6	1	2	1	2	-3	-1	1	0	-5	-3	-1	-1	1

Figure 5.1 - 5: Actual outputs of filter when it is convolved with the signal array and with the noise array.

5.12 Recursive Computation Algorithm

The recursive scheme that was presented in section 4.231 can be expanded to apply to the extension of the length of the wavelet F_i . In addition, Simpson (Simpson, et al., 1963) has proposed a similar recursive scheme to shift the lag between the desired output D_i and the input X_i . Thus this recursion allows and efficient search for the optimum lag.

5.121 Extension of filter length

The recursive algorithm will be stated in terms of the z-transform. The normal equations for $F_M(z)$ may be written as

$$F_M(z) \quad R(z) \quad = \quad \hat{e}_M(z)$$

where $\hat{e}_{j,M} = G_{j-1} \quad j = 1, \ldots, M$.

If we weight $B_M(1/z)$ and add it to $F_M(z)$ we find

$$\left[F_M(z) + z^{M+1} K_{F,M} B_M(1/z)\right] R(z) = \hat{e}_M(z) + z^{M+1} K_{F,M} \varepsilon_M(z)$$

Since $\varepsilon_{i,M} = 0 \quad i = -1, \ldots, -M$, if we choose $K_{F,M}$ such that

$$\hat{e}_{M+1,M} + K_{F,M} \varepsilon_{0,M} = G_M ,$$

then we obtain the recursive relationship

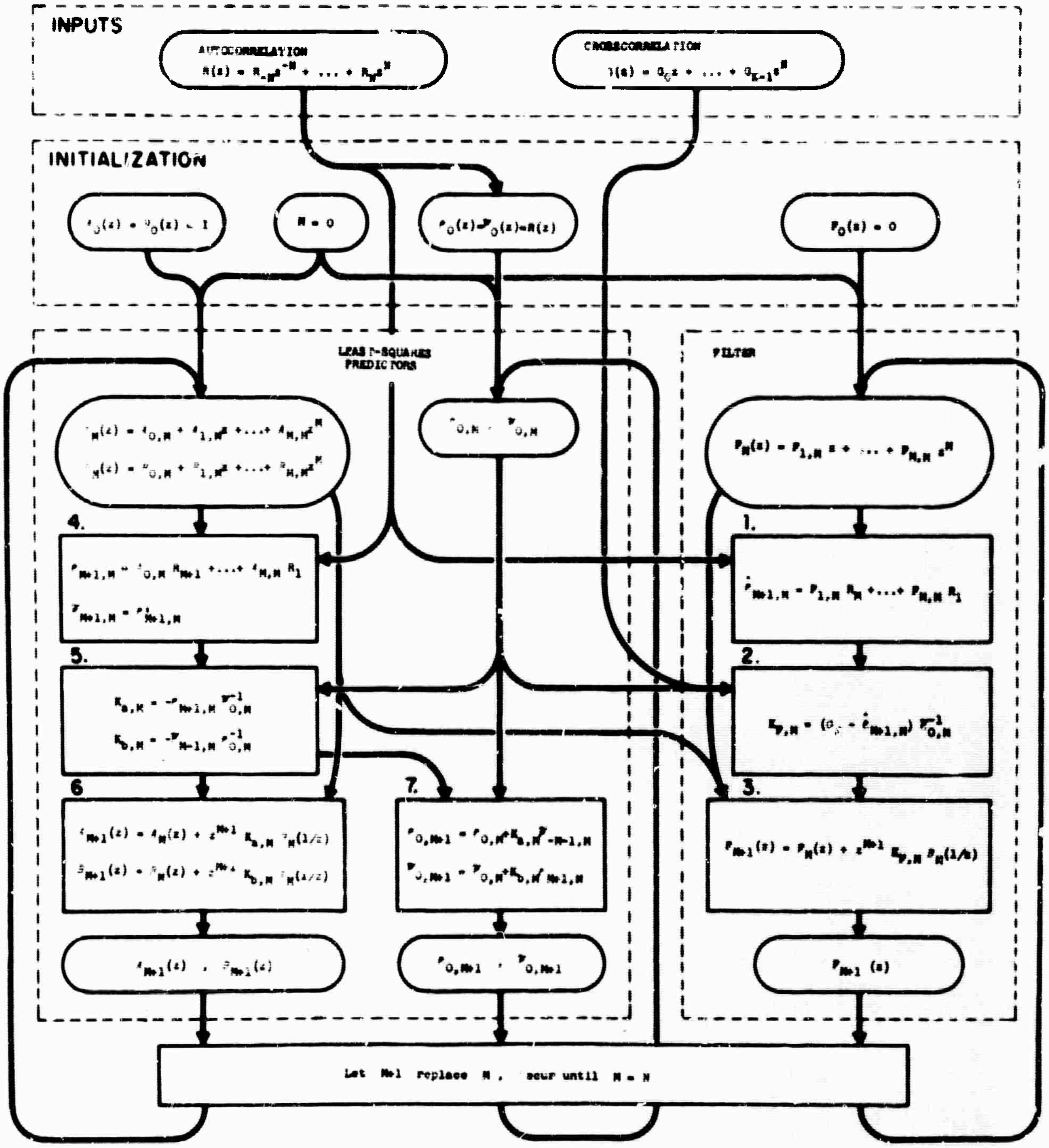


Figure 5.1 - 6: Recursion to extend the length of an optimum least-squares filter. The numbers on the boxes illustrate a possible computational sequence.

$$F_{M+1}(z) = F_M(z) + z^{M+1} K_{F,M} \mathcal{B}_M(1/z) .$$

This recursion is illustrated in detail in Figure 5.1-6.

Similarly, if we had wished to extend $F_M(z)$ in the other direction, we would weight $\mathcal{A}_M(z)$ and add it to $F_M(z)$

$$F_M(z) + \hat{K}_{F,M} \mathcal{A}_M(z) = \hat{\mathcal{E}}_M(z) + \hat{K}_{F,M} \mathcal{E}_M(z) .$$

where we choose $\hat{K}_{F,M}$ such that

$$\hat{\mathcal{E}}_{0,M} + \hat{K}_{F,M} \mathcal{E}_{0,M} = G_{-1}$$

and we have extended the length of $F_M(z)$ in the negative direction according to

$$F_{M+1}(z) = z \left[F_M(z) + \hat{K}_{F,M} \mathcal{A}_M(z)\right] .$$

5.122 Shift of output origin

This recursion is slightly more complicated than the simple length extension. Our objective is to find a filter $F_M^{(1)}(z)$ such that

$$F_M^{(1)}(z) \; R(z) = \hat{\mathcal{E}}_M^{(1)}(z)$$

where $\hat{\mathcal{E}}_{j,M}^{(1)} = G_{j-2} \quad j = 1, \ldots, M$ given $F_M(z)$ as described in the last section. The first step is to shift $F_M(z)$ and $\mathcal{B}_{M-1}(1/z)$ right one unit and subtract

$$z \left[F_M(z) - z^M F_{M,M} \mathcal{B}_{M-1}(1/z)\right] R(z) = \Upsilon_M(z)$$

where $\gamma_{i,M} = G_{i-2}$ $i = 2, \ldots, M$. Thus the error that is introduced by shifting $F_M(z)$ is compensated for, $i = 2, \ldots, M$, by subtracting $B_{M-1}(z)$ multiplied on the left by $F_{M,M}$. Now we add a weighted version of $A_{M-1}(z)$ to adjust the value of $\gamma_{1,M}$ to be G_{-1} :

$$z \left[F_M(z) - z^M F_{M,M} B_{M-1}(1/z) + K_{F;M} A_{M-1}(z) \right] R(z) =$$

$$\gamma_M(z) + K_{F;M} e_{M-1}(z)$$

Thus if we set

$$\gamma_{1,M} + K_{F;M} e_{0,M-1} = G_{-1}$$

we find that

$$F_M^{(1)}(z) = z \left[F_M(z) - z^M F_{M,M} B_{M-1}(1/z) + K_{F;M} A_{M-1}(z) \right] .$$

Similarly, we could have chosen to left shift the cross-correlation to solve for $F_M^{(-1)}(z)$

$$F_M^{(-1)}(z) \ R(z) = \hat{e}_M^{(-1)}(z)$$

where $\hat{e}_{i,M}^{(-1)} = G_j$ $j = 1, \ldots, M$. This filter is given by

$$F_M^{(-1)}(z) = 1/z \left[F_M(z) - F_{1,M} \, z \, A_{M-1}(z) + \right.$$

$$\left. K_{F",M} \, z^{M+1} B_{M-1}(1/z) \right]$$

where $\gamma_M" - K_{F",M} e_{M-1,M-1} = G_M$, and

$$\gamma_M" = (F_{2,M} - F_{1,M} A_{1,M-1}) R_{-1} + \cdots$$

$$+ (F_{M,M} - F_{1,M} A_{M-1,M-1}) R_{-M} .$$

5.2 Computational Properties

The normal equations derived in the first part of this section will certainly provide the optimum linear least-squares filter for a given specification of a problem. The success of the application of such filters depends to a large extent upon the design of the specifications so that a small expected error is obtained. The rest of this chapter will be devoted to examining some qualitative design criteria for scalar least-squares problems. Most of the conclusions may be easily extended to multi-variable problems.

In most least-squares filter problems there are at least three parameters that are left to the discretion of the designer:

1) The inclusion of an arbitrary amount of white noise to represent uncertainty in the design criteria.

2) The specification of the shape of the desired output process.

3) The specification of the lag between the desired output and the input process.

A set of computational experiments were performed to test the effects of the variation of these and other parameters. In each experiment, after an autocorrelation series and a crosscorrelation series were specified, the expected errors for all filters with lengths less than some maximum (50 points in these examples), and with all relevent lags, were computed. The array of expected errors were normalized between 1. and 0. and were contoured in terms of decibels ($10 \log_{10} \varepsilon^2$ where ε^2 is the expected error). Thus each experiment provides a complete test of the effect of the output lag.

Nearly all of the experiments were performed using the wavelets shown in Figure 5.2-1. The mixed delay and minimum delay wavelets shown have the same amplitude spectrum. The mixed delay wavelet was chosen with a large dynamic range so that the expected error plots would have definite character.

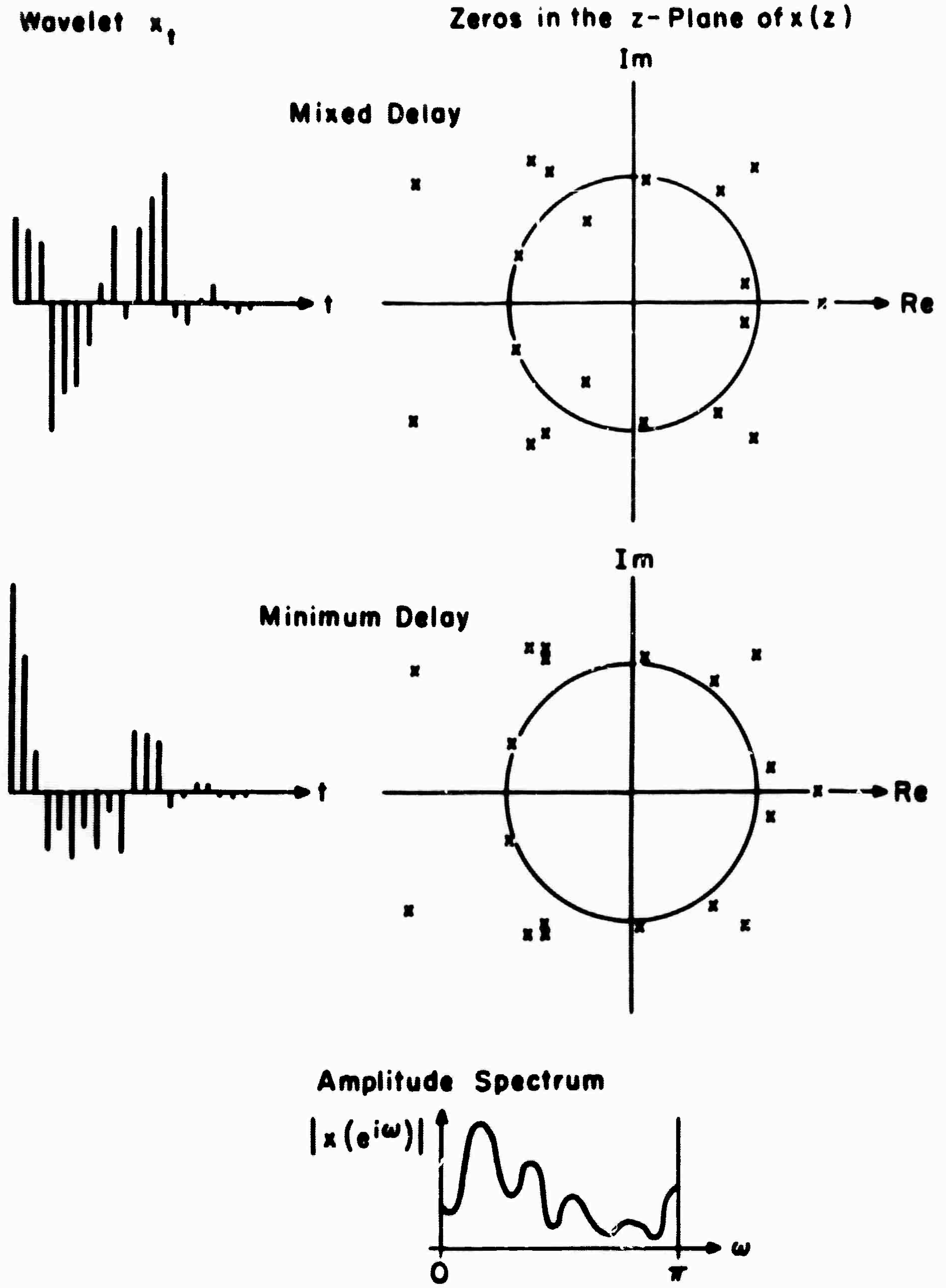


Figure 5.2 - 1: Time domain, z-transform, and amplitude spectrum of a mixed delay wavelet and its equivalent minimum delay form.

5.21 White Noise

The first experiment involved signal shaping in the presence of white noise. Since the autocorrelation of white noise is a spike at zero lag and zeros elsewhere, white noise is included by adding a constant to the center term of the autocorrelation.

The filters for which the expected errors were computed were asked to compress the mixed delay wavelet shown in Figure 5.2-1 into a spike while rejecting varying amounts of white noise. The resulting expected error arrays are shown in Figure 5.2-2. The primary point to notice here is that as the relative power of the white noise is increased, the filter shape becomes stable for shorter lengths but the expected error approaches a constant greater than zero. This constant is related to the relative powers of the white noise and the wavelet. The position of the minimum relative to the output lag is also somewhat dependent upon the amount of noise added.

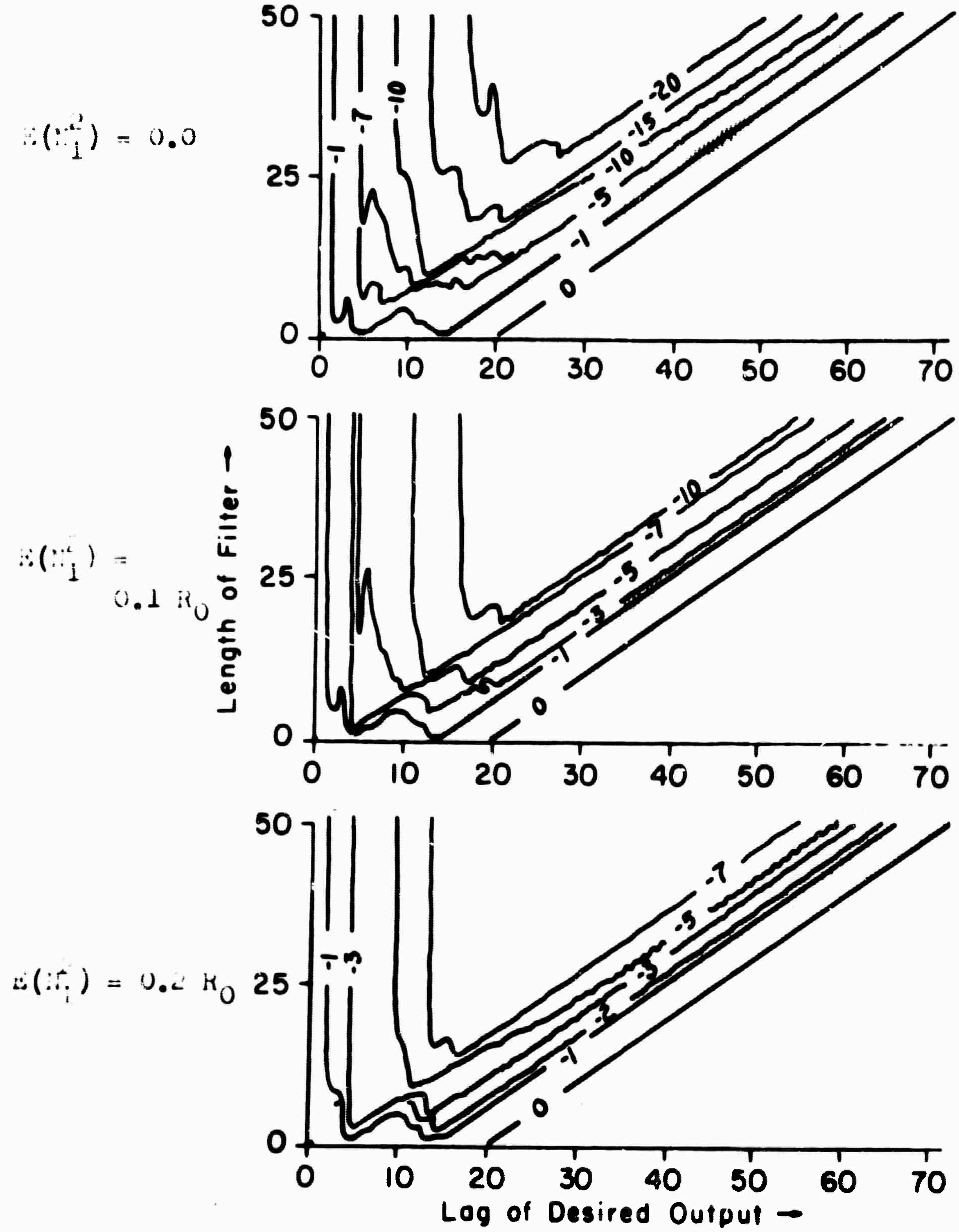


Figure 5.2 - 2: Contours (in decibels) of the error arrays for optimum shaping filters in the presence of white noise. R_0 is the zero lag of the wavelet autocorrelation.

5.22 Delay Properties of the Input Wavelet

Although it is usually not readily alterable, the delay of the input wavelet will certainly affect the design of the filter problem. This experiment illustrates the effect. For each case the least-squares filter was asked to compress the energy of the input wavelet into a spike. The contours of the expected error arrays are shown in Figure 5.2-3. The first plot is for the minimum delay wavelet shown in Figure 5.2-1. The second plot is for the mixed delay wavelet, and the third plot is for a maximum delay wavelet obtained by taking the time reverse of the minimum delay wavelet.

Clearly the location of the minimum is strongly affected by the delay of the input wavelet. The relationship between the position of the expected error minimum and the delay is not simple. A longer set of experiments showed that this position of the minimum is most highly dependent upon the zeros that are far from the unit circle and least dependent upon zeros close to the unit circle. That is, the change in position of the minimum when a distant root is reflected about the unit circle is much greater than when a root with nearly unit magnitude is reflected. It is possible to say, though, that for all the possible minimum positions for wavelets with a given amplitude spectrum, the minimum delay wavelet will have the position closest to the left side of the plot, and

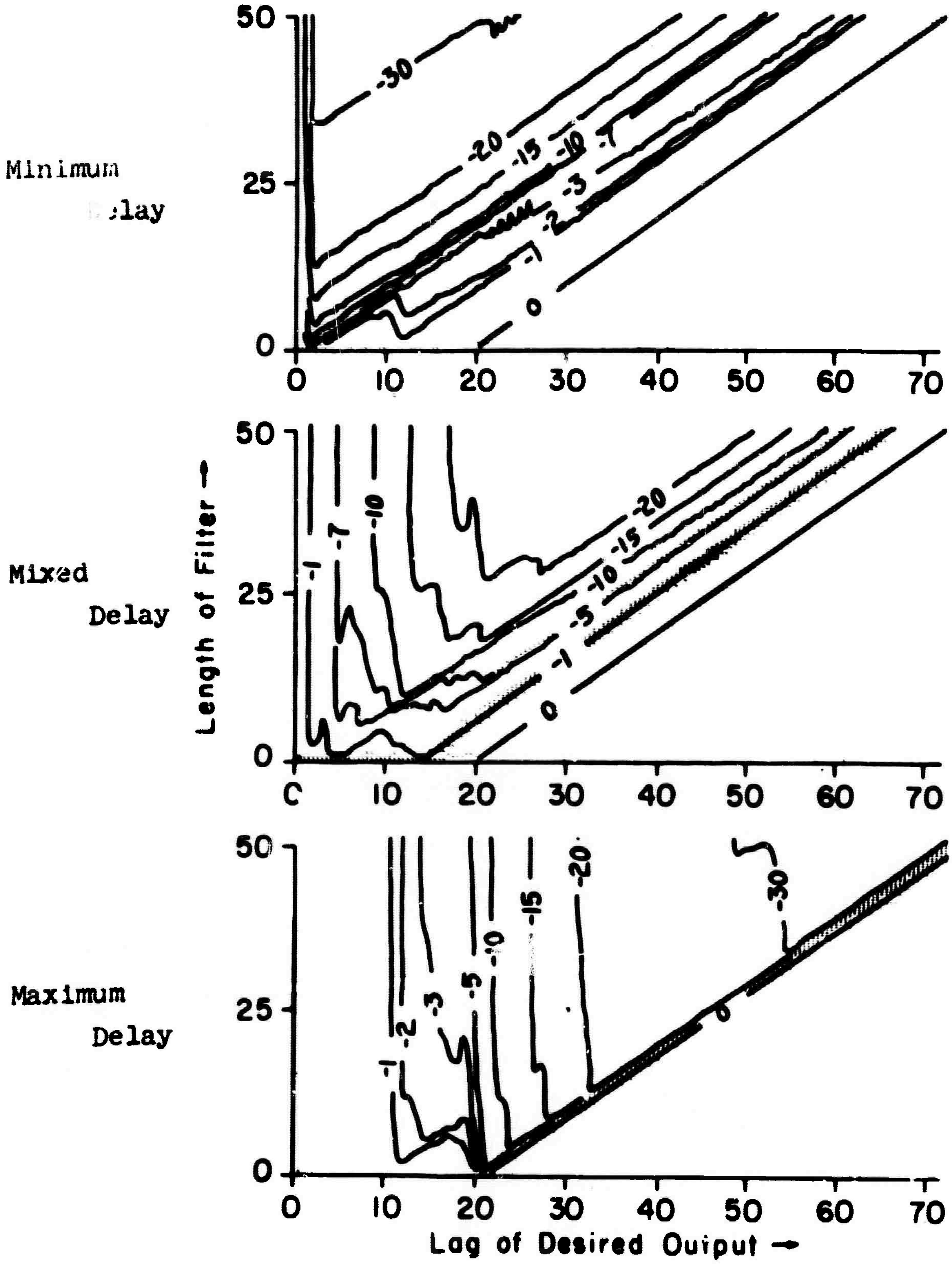


Figure 5.2 - 3: Contours (in decibels) of the expected error arrays of spiking filters for 3 wavelets that have the same amplitude spectrum.

the maximum delay wavelet will have the position closest to the right side. As the expected error at these optimum lags approach zero the optimum position will approach lag one for a minimum delay wavelet.

5.23 Desired Output Spectrum

The last set of experiments tests the effect on the expected error of the shape of the desired output wavelet relative to that of the input wavelet. The definition of such an experiment is necessarily very vague since it is difficult to define a measure of relative shape. For the examples shown in Figure 5.2-4 the amplitude spectrum of the desired output was varied relative to that of the input wavelet.

The input wavelet for all of the cases was the mixed delay wavelet shown in Figure 5.2-1. The desired output for the first case had the same amplitude spectrum as the input wavelet but was maximum delay. The desired output for the second case was a spike, that is, it had a unit amplitude spectrum. For the last case the desired output was a 20-term minimum delay wavelet that had an amplitude spectrum that was approximately reciprocal to the amplitude spectrum of the input case.

This set of examples verifies the conclusion that one would make intuitively, i.e. that the closer

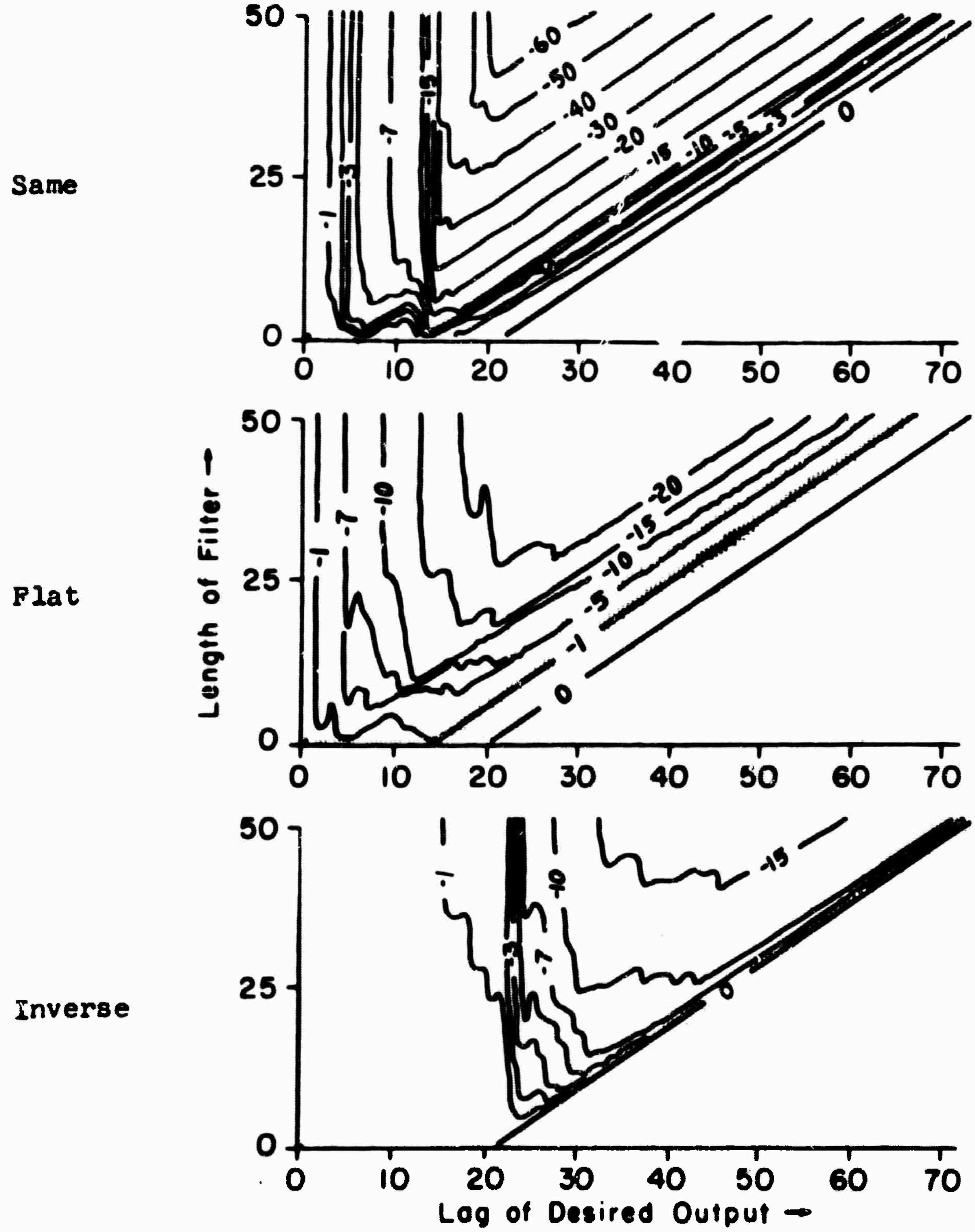


Figure 5.2 - 4: Contours (in decibels) of the expected error arrays of optimum shaping filters for which the desired output has the same, a flat, and an inverse amplitude spectrum relative to the signal wavelet.

the spectrum of the desired output wavelet is to that of the input wavelet, the better the filter is able to perform.

5.24 Output Lag and Filter Length

All of the examples illustrated in Figures 5.2-2, -3, and -4 illustrate properties of expected error verses output lag and filter length. We can draw the following conclusion: The expected error is a non-increasing function of filter length for any particular lag of the desired output relative to the input. The value of the expected error levels out to some value which depends on the output lag. Indeed the expected error is strongly dependent upon the output lag and may vary quite rapidly for small changes in the output lag. Generally, as the length of the filter becomes long with respect to the lengths of the input and desired output wavelets, the expected error curve, plotted with respect to the output lag, has one relative minimum.

The discussion above may be sharpened somewhat by examining the results of Claerbout and Robinson (1964). They showed that the sum of the expected errors for all possible lags of the desired output relative to the input is independent of the length of the filter if the problem involves no noise suppression. Thus as the filter becomes longer the total expected error will be spread over a greater region. Consequently, there must be some lag for which the expected error approaches zero at least as $1/M$ where M is the length of the filter.

The examples illustrated in Figures 5.2-2, -3, and -4 show that the performance of an optimum filter is strongly dependent upon the design of the input parameters. Thus the successful application of least-squares techniques should include all possible means of optimizing the design of the problem before finding the optimum filter for a particular case.

6. GEOPHYSICAL APPLICATIONS

The mathematics developed in the preceeding three chapters is general in its applicability to geophysical, as well as non-geophysical, problems. Since the range of possible applications is great, this chapter will be restricted to two examples for which the setup of the problems do not require long derivations.

The first example is an illustration of the usefulness of the zero-phase multi-dimensional factorization. The second example is of a two-dimensional least-squares problem that is derived with a slightly different initial criteria than that used in Chapter 5. The end result, however, is a set of normal equations for which the recursive procedures given in Chapters 4 and 5 are applicable.

6.1 World-Wide Average Gravity Anomalies

The Army Map Service (1959) has reported on a statistical analysis of available world-wide gravity data. Part of the results reported were numerical estimates of the average gravity covariance for continental regions and for oceanic regions. These estimates are shown in Figure 6.1-1. The curves are an eye-ball smoothing of the data.

If we follow the assuption of the Army Map Service that the shape of gravity anomalies are stationary with respect to azimuth, then the curves illustrated would

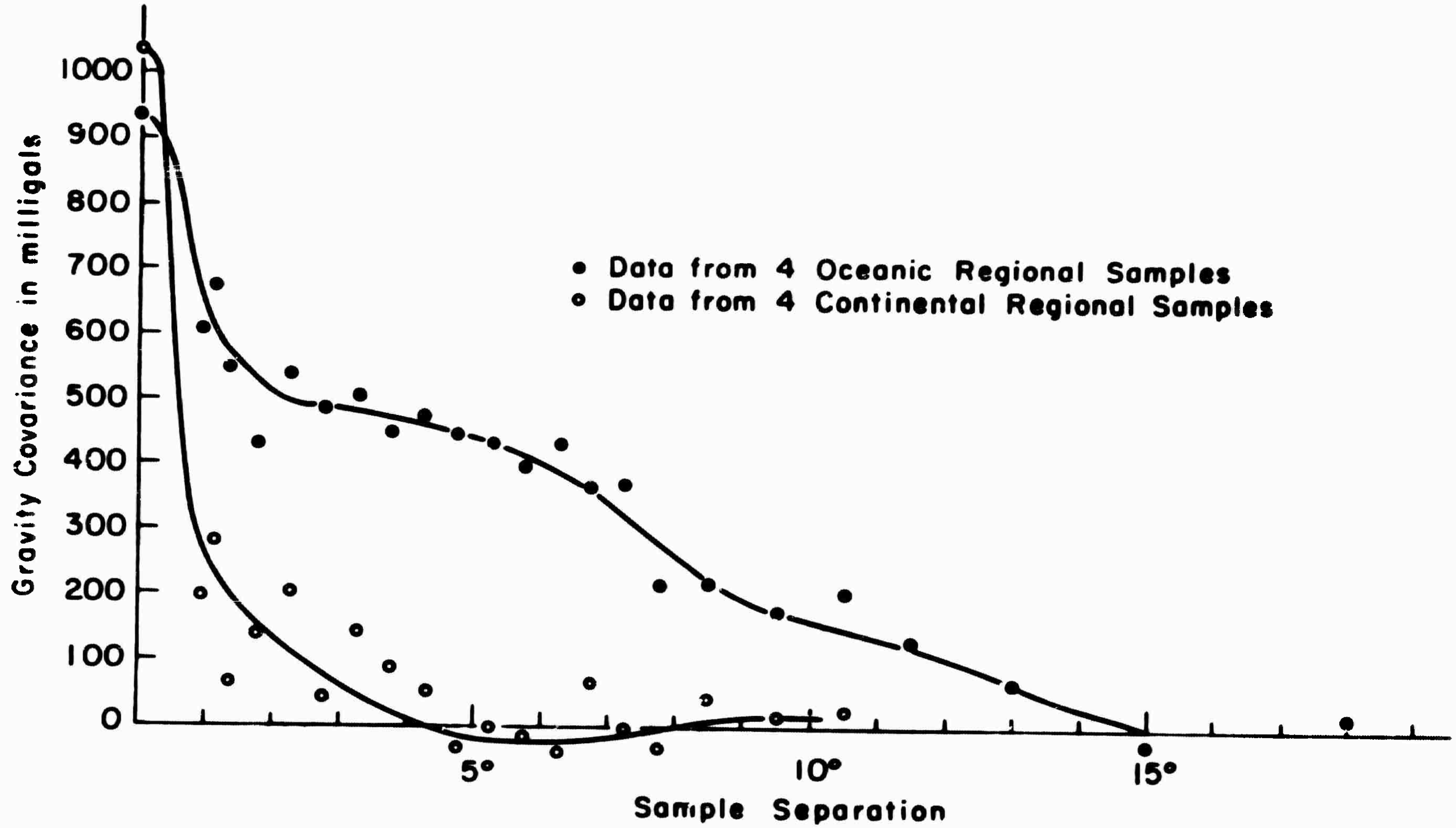


Figure 6.1 - 1: Numerical estimates of the gravity anomaly covariance for oceanic and for continental regions.

represent one-half of the cross-section of the two-dimensional autocorrelation of an average gravity anomaly. Clearly we should be able to apply the zero-phase (see Section 4.32) factorization technique to determine the average symmetrical gravity anomaly. This was carried out as follows:

1. A two-dimensional autocorrelation array was made by sampling the curves in Figure 6.1-1 at 1° intervals.

2. The two-dimensional cosine transform was computed.

3. Since the sample arrays are covarience estimates, they may not represent possible autocorrelation functions, that is, some values of the cosine transform may be negative. In this example, a few small values for the oceanic samples were negative. These values were arbitrarily set positive. (As a check, these values were also set to zero. The resultant solution was modified only slightly.)

4. The square root of each value in the transform was taken.

5. These values were than inverse transformed to obtain an average gravity anomaly.

Cross-sections of the resultant average gravity arrays are plotted in Figure 6.1-2. A preliminary interpretation was made by determining the depth of a point

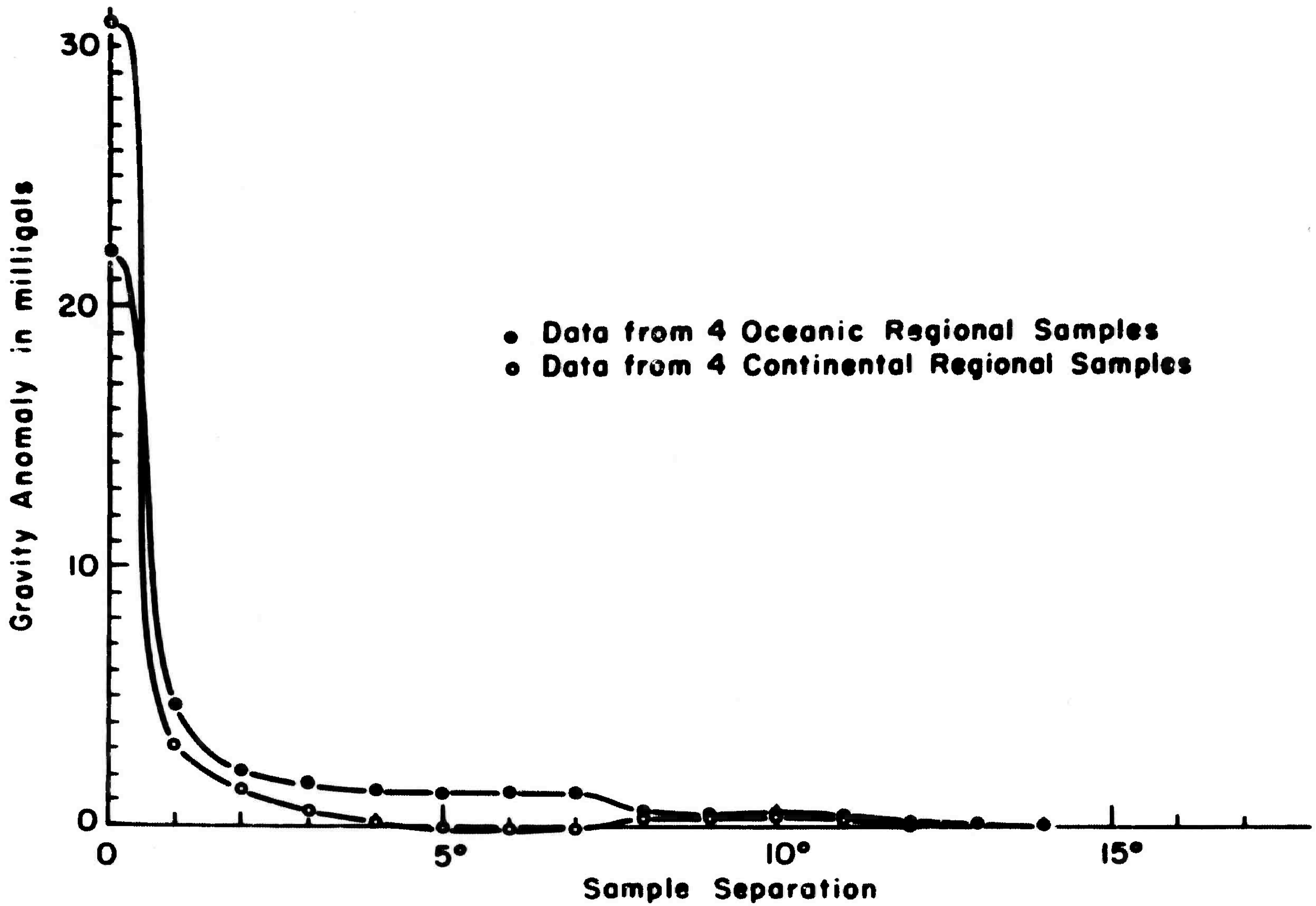


Figure 6.1 - 2: Numerical estimates of the average gravity anomaly for oceanic and for continental regions.

source that would give the same gravity values at 0° and at 1°. Such sources would have a depth of ~60 km for the continental regions and ~80 km for the oceanic regions. The anomalous masses would be ~10% larger in the oceanic regions than in the continental regions.

The point source interpretation does not account for the large values of the oceanic anomaly out to 7°. The total mass of an anomalous body is proportional to the volumn under the two-dimensional gravity anomaly. Using this property as a criterion for comparing masses, we see that the oceanic anomalous masses are approximately 3.5 times as large as the continental masses.

6.2 ω-k Filtering

The idea of band-pass or band-reject filtering is perhaps one of the oldest concepts in one-dimensional, scalar filter design. This concept can be readily extended to two-dimensional filtering by defining band-pass or band-reject areas in the two-dimensional Fourier-transform domain (sometimes called the ω-k plane). One very simple form of such filters has wide application in exploration seismology for descrimination between plane waves on the basis of the direction of arrival of the wave (see Fail and Grau, 1963; or Embree, et al., 1963).

In this section we will consider the general problem of optimally designing a discrete two-dimensional filter from band-pass and band-reject area specification in the ω-k plane.

Let us first consider the two-dimensional band-pass filter. The criterion for designing this filter is that we wish to minimize the square of the difference between the desired band-pass configuration and the actual Fourier-transform of the filter. Thus, if we let f_j $j = 1, \ldots, n$ represent a discrete filter with lattice points at (x_j, t_j) $j = 1, \ldots, n$, and $d(\omega,t)$ represent the desired band-pass configuration, then the error to be minimized is

$$\mathcal{E} = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left[\sum_j f_j \quad e^{i(\omega t_j - kx_j)} - d(\omega,k) \right]^2 d\omega \, dk \qquad (6.2\text{-}1)$$

where $a^2 = a\bar{a}$. Taking the derivative of $\mathcal{E}$ with respect to f_j, we find a minimum given by

$$\frac{\partial \mathcal{E}}{\partial f_j} = 0 \qquad j = 1, \ldots, n$$

or

$$0 = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{i(\omega t_\ell - kx_\ell)} \cdot$$

$$\left[\sum_j f_j \, \bar{e}^{i(\omega t_j - kx_j)} - d(\omega,k) \right] d\omega \, dk$$

$$j = 1, \ldots, n, \quad \ell = 1, \ldots, n \qquad (6.2\text{-}2)$$

which leads to the particularly simple result:

$$f_j = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{i(\omega t_j - kx_j)} \, d(\omega,k) \, d\omega \, dk$$

Thus the filter coefficients are determined by the Fourier-transform of the desired band-pass configuration. The expected error $\mathcal{E}$, is given by (beginning from equation 6.2-1)

$$\mathcal{E} = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} d(\omega,k) \left[\sum_j f_j \, \bar{e}^{i(\omega t_j - kx_j)} - \overline{d(\omega,k)} \right] d\omega \, dk$$

which gives (according to equation 6.2-2)

$$= \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} d^2(\omega,k) \, d\omega \, dk - \sum_j f_j^2 \quad .$$

This is reminiscent of the expression for the expected error that was found in chapter 5.11.

The simplicity of this method of filter computation makes it feasible to examine the properties of such filters in several ways. Figures 6.2-1 and -2 illustrate several of the possible computations that are useful in studying the properties for any given ω-k configuration. Each of the figures contains 4 contour plots that represent

1. The coefficients for a square 13 x 13 term filter.

2. The actual Fourier transform of the 13 x 13 term filter contoured in decibels. This plot is superimposed upon a diagram of the desired pass region.

3. The array defined by

$$e_j = f_j^2 \Big/ \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} d^2(\omega,k) \quad d\omega \; dk$$

contoured at values of e_j = 0.00001, 0.0001, 0.001, 0.01, and 0.1 . The value at any point in this array is just the value by which the normalized expected error is decreased by the addition of that point to the filter. Thus these contours represent the optimum filter shapes for minimizing both the expected error and the number of filter coefficients. It is interesting to notice that the maxima in the e_j array are parallel to the boundaries of the pass region in the ω-k definition of the filter.

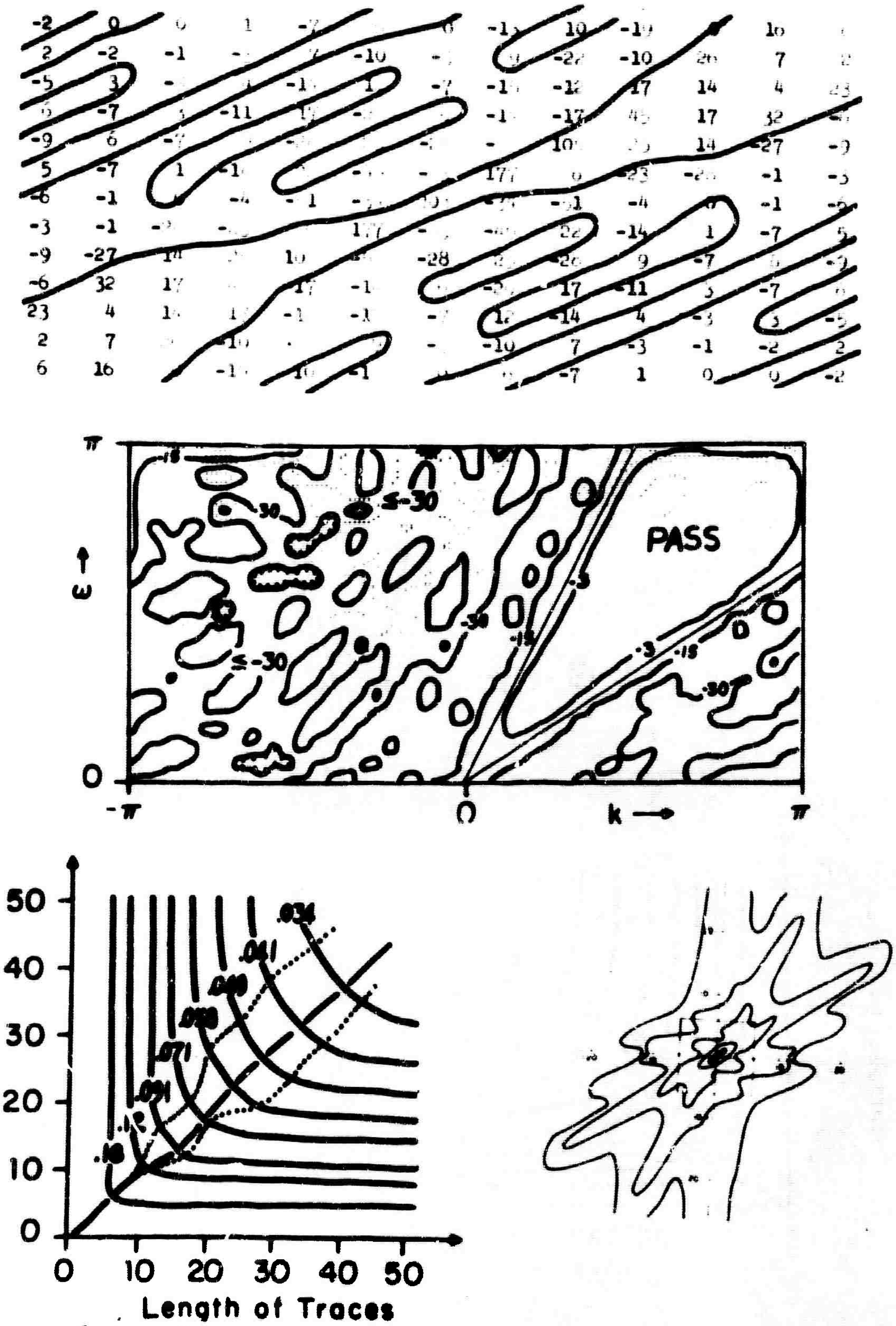


Figure 6.2 - 1: The filter coefficients for an optimum least-squares band-pass filter; the contours (in decibels) of the actual ω-k plane reponse of the filter; the array of expected errors for all rectangular filters smaller than 51 x 51 terms; and the optimum filter shapes for this pass-band configuration.

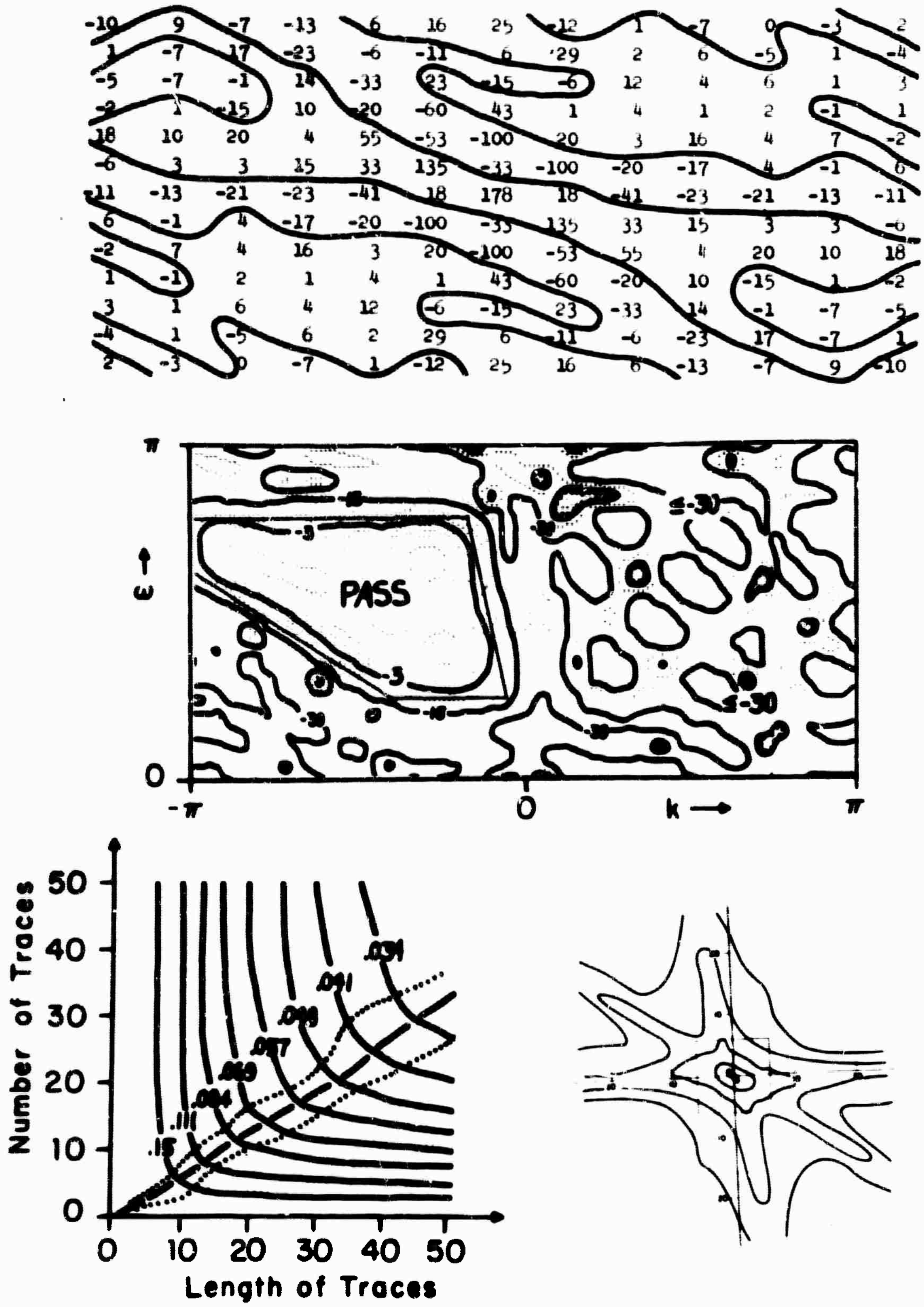


Figure 6.2 - 2: The filter coefficients for an optimum least-squares band-pass filter; the contours (in decibels) of the actual ω-k plane response of the filter; the array of expected errors for all rectangular filters smaller than 51 x 51 terms; and the optimum filter shapes for this pass-band configuration.

4. The array o normalized expected errors for all possible rectangular dimensions of the filters smaller than 51 x 51 terms. The center dotted line represents the optimum length for any given number of traces while the area between the outer dotted lines include all dimensions which are quite close to the optimum rectangular shape.

In nearly all physical problems the contributions to the error in some parts of the ω-k plane are more important than the contribution in other parts of the plane. Let us define a weighting function $W(\omega,k)$ that expresses this importance. The larger $W(\omega,k)$ is in any given area, the more important the error is in that area. Let us introduce $W(\omega,k)$ into the least-squares band-pass problem:

$$\mathcal{E} = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} W(\omega,k)\left[\sum_j f_j\, e^{i(\omega t_j - kx_j)} - d(\omega,k)\right]^2 d\omega\, dk$$

Setting the derivatives of $\mathcal{E}$ to zero as before, we find

$$\frac{\partial \mathcal{E}}{\partial f_j} = 0$$

$$= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} W(\omega,k)\, e^{i(\omega t_\ell - kx_\ell)} \cdot$$

$$\left[\sum_j f_j\, e^{-i(\omega t_j - kx_j)} - d(\omega,k)\right] d\omega\, dk$$

we find

$$\sum_j f_j \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} W(\omega,k) \exp i\left[\omega(t_\ell - t_j) - k(x_\ell - x_j)\right] d\omega dk =$$

$$\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} W(\omega,k)\, d(\omega,k) \exp i\left[\omega t_\ell - k x_\ell\right] d\omega dk$$

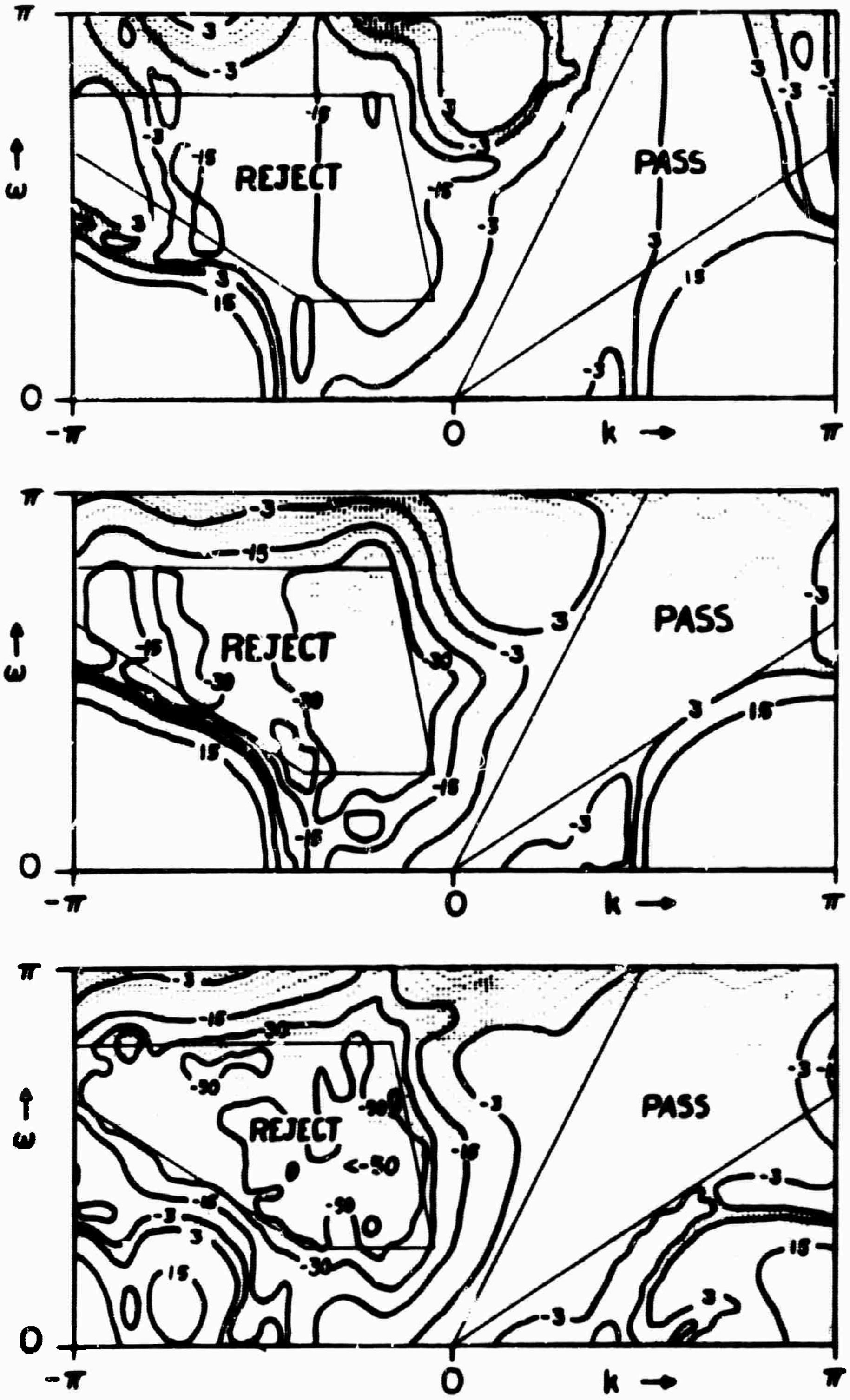
$$\ell = 1, \ldots, n .$$

Furthermore, the expected error becomes

$$\mathcal{E} = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} W(\omega,k)\, d(\omega,k) \left[\sum_j f_j\, e^{-i(\omega t_j - k x_j)} - \overline{d(\omega,k)}\right] d\omega dk .$$

Thus we find a set of simultaneous equations which must be solved for the f_j. Clearly, since the coefficients multiplying f_j depend upon $(t_\ell - t_j)$ and $(x_\ell - x_j)$ they have the same symmetry properties that were stated for multi-dimensional autocorrelations. If follows then that once the integrals have been evaluated, the recursive computation procedure given in Chapter 5.121 can be used to solve the simultaneous equations.

Figure 6.2-3 shows the ω-k transform of three filters computed by the least-squares method. In each case the weighting function, $W(\omega,k)$, was zero everywhere except in the PASS and REJECT regions. It was held constant over each of these regions. In the three examples shown, the ratio of the weighting function in the REJECT region to that in the PASS region (which is labelled N/S in the figure) was set at 1., 10., and 100., respectively. The remarkable improvement in the filter performance as the



N/S = 1.

N/S = 10.

N/S = 100.

Figure 6.2 - 3: Contours (in decibels) of the ω-k plane transforms of 3 least-squares optimum band-pass and band-reject filters calculated with various values for the noise-to-signal weighting parameter.

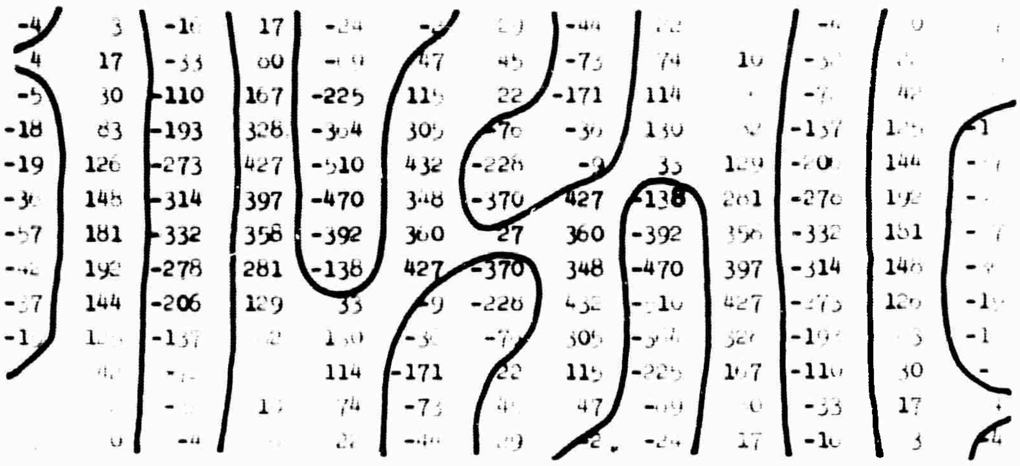
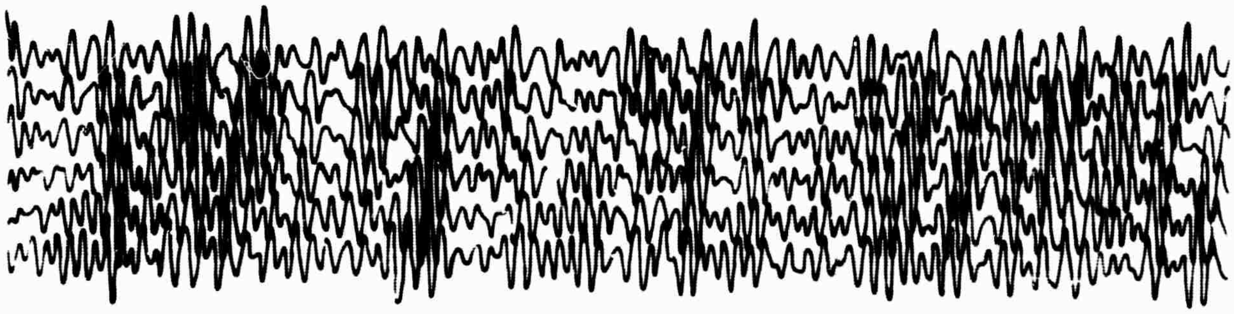


Figure 6.2 - 4: Coefficients for third filter (N/S = 100.) illustrated in Fig. 6.2-3.

ratio increases reflects the fact that we are much more interested in the reject region having very nearly zero amplitude at all points while we are less concerned with small fluctuations in the pass-region. The filtercoefficients for the last filter are given in Figure 6.2-4.

As a final illustration of the effectiveness of these filters when applied to data, a set of traces was constructed which contained noise whose spectrum is primarily in the REJECT region and signals that are in the PASS region. These traces were convolved with the transform filter illustrated in Figure 6.2-1 and with the least-squares filter illustrated at the bottom of Figure 6.2-3. The results are shown in Figure 6.2-5. The signals had stepouts of 1.5, 1.0, and 0.5 digitization units per trace. Thus the signals that were on the edges of the pass band of the transform filters were attenuated by ~2. On the other hand, because of the difficulty of constructing

Sample of noise + signal traces.



Noise + signal traces convolved with band-pass filter (see Fig. 6.2-2).

Noise + signal traces convolved with band-pass, band-reject filter (see Fig. 6.2-4).

Figure 6.2 - 5: Example of application of band-pass and band-pass, band-reject filters to simulated noise and signal traces.

noise that is completely restricted to the REJECT region, some very low frequency waves were actually amplified by the least squares filter.

APPENDIX

PROGRAM LISTINGS

Nearly all of the subroutines used for the computations made for this thesis were performed by programs from the set documented by Simpson (1965). A complete description of the writing format, the abbreviations, and other programs referred to will be found in the work cited above.

All of the subroutines listed here were written for the purpose of investigating polynomial matrices and their spectral decomposition. Several of the programs (BRAINY, MATML3, and SIMEQC) are, however, much more general in their application.

```
*     BRAINY (SUBROUTINE)
*     LABEL
CBRAINY
      SUBROUTINE BRAINY (NRA,NCA,LPA,A,NCB,LPB,B,C)
C
C                     ----ABSTRACT----
C
C TITLE - BRAINY
C         COMPLETE TRANSIENT CONVOLUTION OF TWO MATRIX VALUED SERIES
C
C               SUBROUTINE BRAINY COMPUTES THE COMPLETE TRANSIENT
C               CONVOLUTION
C
C
C                     C(I) = SUM OVER ALL VALUES ( A(J-I+1)*B(I) )
C
C               WHERE A, B, AND C ARE MATRIX-VALUED VECTORS.
C
C LANGUAGE   - FORTRAN II SUBROUTINE
C EQUIPMENT  - 709 OR 7090 (MAIN FRAME ONLY)
C STORAGE    - 265 REGISTERS
C SPEED      -
C AUTHOR     - R.A. WIGGINS AND J.F. CLAERBOUT    1/64
C
C                     ----USAGE----
C
C TRANSFER VECTOR CONTAINS ROUTINES - SETKS,STZ
C        AND FORTRAN SYSTEM ROUTINES - NONE
C
C FORTRAN USAGE
C     CALL BRAINY(NRA,NCA,LPA,A,NCB,LPB,B,C)
C
C INPUTS
C
C    NRA        NUMBER ROWS IN A MATRICES.
C
C    NCA        NUMBER COLUMNS IN A MATRICES, NUMBER ROWS IN B MATRICES.
C
C    LPA        NUMBER OF MATRICES IN A.
C
C    A(I)       I=1...NRA,1...NCA,1...LPA IS A CLOSELY SPACED VECTOR OF
C                 MATRICES.
C
C    NCB        NUMBER COLUMNS IN B MATRICES.
C
C    LPB        NUMBER OF MATRICES IN B.
C
C    B(I)       I=1...NCA,1...NCB,1...LPB IS A CLOSELY SPACED VECTOR OF
C                 MATRICES.
C
C -NOTE- IF NRA, NCA, NCB, LPA, OR LPB ARE LSTHN 1, BRAINY RETURNS WITH
C          NO COMPUTATIONS.
C
C OUTPUTS
C
C    C(I)       I=1...NRA,1...NCB,1...LPA+LPB-1  IS THE CLOSELY SPACED
C                 VECTOR OF MATRICES OF THE CONVOLUTION OF A WITH B.
C
C EXAMPLES
C
C 1. INPUTS  - NRA = 2  NCA = 1  LPA = 3  NCB = 2  LPB = 2
C              A(1...2,1...1,1...3) = 1.,2.,2.,3.,3.,4.
C              B(1...1,1...2,1...2) = -1.,-3.,-2.,-4.
C    OUTPUTS - C(1...2,1...2,1...4) = -1.,-2.,-3.,-6.,-4.,-7.,-10.,-17.,
C                                     -7.,-10.,-17.,-24.,-6.,-8.,-12.,-16.
C
C PROGRAM FOLLOWS BELOW
C
      DIMENSION A(2),B(2),C(2)
      DIMENSION SPACE(20)
      CALL SETKS (NRA,M,NCA,L,NCB,N,LPA,LA,LPB,LB)
      IF (XMINOF(N,M,L,LA,LB)) 100,100,5
5     CONTINUE
      CALL SETKS (M*N,MN,M*L,ML,L*N,LN,1,M1,1,K3)
      CALL STZ (MN*(LA+LB-1),C)
```

```
************************
*     BRAINY           *
************************
(PAGE  2)

          DO 50 I1=1,LB
          L1=M1
          L2=1
          DO 40 I2=1,LA
          K1=L1
          K2=L2
          DO 30 I3=1,M
          J1=K1
          J3=K3
          DO 20 I4=1,N
          J2=K2
          CK=0.
          DO 10 I5=1,L
          CK=CK+A(J2)*B(J3)
          J2=J2+M
    10    J3=J3+1
          C(J1)=CK+C(J1)
    20    J1=J1+M
          K1=K1+1
    30    K2=K2+1
          L1=L1+MN
    40    L2=L2+ML
          M1=M1+MN
    50    K3=K3+LN
   100    RETURN
          END
```


```
************************
*     BRAINY           *
************************
                (PAGE  2)
```

```
*     DIXMAC (SUBROUTINE)
*     LABEL
CDIXMAC
      SUBROUTINE DIXMAC (ZR,ZI,NZRA,NCA,AR,AI,BR,BI)
C
C
C                        ----ABSTRACT----
C
C TITLE - DIXMAC
C        COMPLEX DIAGONAL MATRIX TIMES COMPLEX SQUARE MATRIX.
C
C
C             DIXMAC  COMPUTES THE MATRIX PRODUCT
C
C
C                        DIAG ( Z ) * A  =  B
C
C
C             WHERE  Z  IS A COMPLEX VECTOR OF LENGTH  NZRA,
C                    A  IS A COMPLEX  NZRA BY NCA  MATRIX, AND
C                    B  IS A COMPLEX  NZRA BY NCA  MATRIX.
C
C
C LANGUAGE   - FORTRAN II SUBROUTINE
C EQUIPMENT  - 709, 7090, 7094 (MAIN FRAME ONLY)
C STORAGE    - 93 REGISTERS
C SPEED      -
C AUTHOR     - R.A. WIGGINS   7/64
C
C
C                        ----USAGE----
C
C
C TRANSFER VECTOR CONTAINS ROUTINES - NOT ANY
C       AND FORTRAN SYSTEM ROUTINES - NOT ANY
C
C
C FORTRAN USAGE
C      CALL DIXMAC (ZR,ZI,NZRA,NCA,AR,AI,BR,BI)
C
C
C INPUTS
C
C    ZR(I)      I=1,....,NZRA  CONTAINS THE REAL PART OF THE VECTOR  Z.
C
C    ZI(I)      I=1,....,NZRA  CONTAINS THE IMAGINARY PART OF THE VECTOR
C                 Z.
C
C    NZRA       NUMBER OF ELEMENTS IN  Z  AND NUMBER OF ROWS IN  A AND B.
C               MUST BE GRTHN= 1
C
C    NCA        NUMBER OF COLUMNS IN  A  AND  B.
C               MUST BE GRTHN= 1
C
C    AR(I)      I=1,....,NZRA*NCA  CONTAINS THE REAL PART OF THE MATRIX  A
C                 STORED CLOSELY SPACED BY COLUMNS.
C
C    AI(I)      I=1,....,NZRA*NCA  CONTAINS THE IMAGINARY PART OF THE
C                 MATRIX  A  STORED CLOSELY SPACED BY COLUMNS.
C
C
C OUTPUTS
C
C    BR(I)      I=1,....,NZRA*NCA  CONTAINS THE REAL PART OF THE MATRIX  B
C                 STORED CLOSELY SPACED BY COLUMNS.
C
C    BI(I)      I=1,....,NZRA*NCA  CONTAINS THE IMAGINARY PART OF THE
C                 MATRIX  B  STORED CLOSELY SPACED BY COLUMNS.
C
C
C EXAMPLES
C
C 1. INPUTS  - ZR(1)=2.  ZI(1)=1.  NZRA=1  NCA=2
C              AR(1...2)=1.,3.  AI(1...2)=2.,0.
C    OUTPUTS - BR(1...2)=0.,6.  BI(1...2)=5.,3.
```

```
      C
      C 2. INPUTS  - ZR(1...2)=2.,3.  ZI(1...2)=0.,0.  NZRA=2  NCA=1
      C              AR(1...2)=1.,4.  AI(1...2)=2.,3.
      C    OUTPUTS - BR(1...2)=2.,12. BI(1...2)=4.,9.
      C
      C
      C PROGRAM FOLLOWS BELOW
      C
      C
      C DUMMY DIMENSION
      C
            DIMENSION ZR(2),ZI(2),AR(2),AI(2),BR(2),BI(2)
      C
      C BRING IN SCALAR VARIABLES.
      C
            N=NZRA
            M=NCA
      C
      C FORM USEFUL COMBINATIONS
      C
            NM=N*M
            IB=1
      C
      C DO MULTIPLICATION
      C
            DO 20 IZ=1,N
            DO 10 IAB=IZ,NM,N
            BR(IAB)=ZR(IZ)*AR(IAB)-ZI(IZ)*AI(IAB)
            BI(IAB)=ZR(IZ)*AI(IAB)+ZI(IZ)*AR(IAB)
       10   CONTINUE
       20   CONTINUE
      C
      C THAT'S ALL
      C
            RETURN
            END
```

```
*     LAVEC (SUBROUTINE)
*     LABEL
CLAVEC
      SUBROUTINE LAVEC (LZ,ZR,ZI,NRCA,LPA,ADJA,ZIFU,UR,UI)
C
C
C                    ----ABSTRACT----
C
C TITLE - LAVEC
C         LATENT VECTORS FOR A POLYNOMIAL MATRIX
C
C
C               LAVEC  FINDS THE  LZ  LATENT VECTORS  U(I)  OR  V(I)  OF
C               A POLYNOMIAL MATRIX  A(Z)
C
C                                                      T
C                  ADJUGATE ( A ( Z(I) )  )  =  U(I) * V(I) .
C
C
C               GIVEN THE COMPLEX LATENT ZEROS  Z(I), I=1,...,LZ  AND THE
C               MATRIX COEFFICIENTS OF THE ADJUGATE OF  A(Z).  THE
C               VECTORS ARE SCALED SO THAT THE FIRST NON-ZERO ELEMENT IS
C               EQUAL TO (1.,0.).
C
C
C LANGUAGE    - FORTRAN II SUBROUTINE
C EQUIPMENT   - 709, 709C, 7094 (MAIN FRAME ONLY)
C STORAGE     - 328 REGISTERS
C SPEED       -
C AUTHOR      - R.A. WIGGINS   10/64
C
C
C                    ----USAGE----
C
C
C TRANSFER VECTOR CONTAINS ROUTINES - CHOOSE,IPLYEV,MATRA,MAXAB
C        AND FORTRAN SYSTEM ROUTINES - NOT ANY
C
C
C FORTRAN USAGE
C      CALL LAVEC (LZ,ZR,ZI,NRCA,LPA,ADJA,ZIFU,UR,UI)
C
C
C INPUTS
C
C    LZ         LENGTH OF VECTOR OF LATENT ZEROS
C
C    ZR(I)      I=1,...,LZ  IS REAL PART OF THE VECTOR OF LATENT ZEROS,
C                 Z,  OF  A(Z).
C
C    ZI(I)      I=1,...,LZ  IS IMAGINARY PART OF THE VECTOR OF LATENT
C                 ZEROS,  Z,  OF  A(Z).
C
C    NRCA       NUMBER OF ROWS OR COLUMNS IN THE MATRICES OF  ADJA.
C               MUST BE GRTHN= 1
C
C    LPA        LENGTH OF POLYNOMIALS IN ADJA,  I.E. THE NUMBER OF
C                 COEFFICIENT MATRICES IN ADJA.
C               MUST BE GRTHN= 1
C
C    ADJA(I)    I=1,...,NRCA*NRCA*LPA  CONTAINS THE MATRIX COEFFICIENTS
C                 OF THE ADJUGATE OF  A(Z).
C
C    ZIFU       =0.     IF THE  U(I)  VECTORS ARE DESIRED.
C               NOT=0.  IF THE  V(I)  VECTORS ARE DESIRED.
C
C OUTPUTS
C
C    UR(I)      I=1,...,NRCA*LZ  CONTAINS THE REAL PARTS OF THE LATENT
C                 VECTORS  U(J), J=1,...,LZ,  IF  ZIFU=0.  OR  V(J)  IF
C                 ZIFU  NOT= 0.
C    UI(I)      I=1,...,NRCA*LZ  CONTAINS THE IMAGINARY PARTS OF THE
C                 LATENT VECTORS  U(J), J=1,...,LZ  IF  ZIFU=0.  OR  V(J)
C                 IF  ZIFU  NOT= 0.
C
```

```
C
C EXAMPLES
C
C 1. INPUTS  - LZ=2  ZR(1...2)=1.,2.  ZI(1...2)=0.,0.  ZIFU=0.
C              NRCA=1  LPA=2  ADJA(1...2)=1.,3.
C    OUTPUTS - UR(1...2)=1.,1.
C              UI(1...2)=0.,0.
C
C 2. INPUTS  - LZ=2  ZR(1...2)=2.,2.  ZI(1...2)=1.,-1.  ZIFU=0.
C              NRCA=2  LPA=2  ADJA(1...8)=(1., 0.)  (-.46667, -.11111)
C                                         (0., 1.), ( .40000  -.33333)
C              NOTE THAT  ADJA  IS STORED AS
C              ADJA(1...8)=1.,0.,0.,1.,-.466667,.4,-.11111,-.33333
C    OUTPUTS - UR(1...4)=1.,-0.6,1.,-0.6
C              UI,(1...4)=0., 1.8,0.,-1.8
C
C 3. INPUTS  - LZ=2  ZR(1...2)=-1.,2.  ZI(1...2)=0.,0.  ZIFU=0.
C              NRCA=2  LPA=2  ADJA(1...8)=(1.,0.)  (-.5,0.)
C                                         (0.,1.), ( .5,1.)
C    OUTPUTS - UR(1...4)=1.,-.3333,0.,1.
C              UI(1...4)=0.,0.,0.,0.
C
C
C 4. INPUTS  - SAME AS EXAMPLE 3. EXCEPT  ZIFU=1.
C    OUTPUTS - UR(1...4)= 1.,0.,1.,3.
C              UI(1...4)= 0.,0.,0.,0.
C
C
C PROGRAM FOLLOWS BELOW
C
      DIMENSION ZR(2),ZI(2),ADJA(2),UR(2),UI(2)
      N=NRCA
      L=LPA
      NN=N*N
      LN=L*N
      CALL CHOOSE (ZIFU, ILXLN,L,LN, ILNXL,LN,L)
      CALL MATRA (ADJA,NN,L ,ADJA)
      K1=1
      DO 70 I1=1,LZ
      ZRE=ZR(I1)
      ZIM=ZI(I1)
      J2=1
      DO 40  I2=1,N
      J3=J2
      K3=K1
      DO 20  I3=1,N
      CALL IPLYEV (L ,ADJA(J3),ZRE,ZIM,UR(K3),UI(K3))
      K3=K3+1
 20   J3=J3+ILXLN
      CALL MAXAB  (N,UR(K1),UMX,IUMX)
      IF (ABSF(UMX)-1.E-06) 30,30,45
 30   CALL MAXAB  (N,UI(K1),UMX,IUMX)
      IF (ABSF(UMX)-1.E-06) 40,40,45
 40   J2=J2+ILNXL
 45   CONTINUE
      K4=K1
      DO 50  I4=1,N
      URL=UR(K4)
      UIL=UI(K4)
      ABSU=URL*URL+UIL*UIL
      IF (ABSU-1.E-06) 50,50,55
 50   K4=K4+1
 55   CONTINUE
      K5=K1+N-1
      DO 60  I5=K4,K5
      URT=UR(I5)
      UIT=UI(I5)
      UR(I5)=(URT*URL+UIT*UIL)/ABSU
 60   UI(I5)=(UIT*URL-URT*UIL)/ABSU
 70   K1=K1+N
      CALL MATRA (ADJA,L,NN,ADJA)
      RETURN
      END
```

```
*       MATML3 (SUBROUTINE)
*       FAP
*MATML3
        COUNT   200
        LBL     MATML3
        ENTRY   MATML3 (N,M,L,AA,BB,ZFNBTR,CC,GZFADD)
*
*
*                       ----ABSTRACT----
*
* TITLE - MATML3
*          GENERAL MATRIX MULTIPLICATION
*
*
*                 MATML3  MULTIPLIES AN  N BY M  MATRIX,  A,  BY AN  M BY L
*                 MATRIX,  B,  TO OBTAIN AN  N BY L  PRODUCT MATRIX,  C.
*
*                             M        L            L
*                         (       )   ( )          ( )
*                       N (   A   ) * ( )     =  N (C)
*                         (       )   (B) M        ( )
*                                     ( )
*                                     ( )
*
*                 A  IS ASSUMED TO BE STORED BY COLUMNS.  B  MAY BE STORED
*                 EITHER BY COLUMNS OR BY ROWS.  C  WILL BE STORED BY
*                 COLUMNS.
*
* LANGUAGE    - FAP SUBROUTINE  (FORTRAN II COMPATIBLE)
* EQUIPMENT   - 709, 7090, 7094 (MAIN FRAME ONLY)
* STORAGE     - 79 REGISTERS
* SPEED       - ABOUT  ((18*M+10)*N+10)*L + 120 MACHINE CYCLES ON THE
*                 7090.
* AUTHOR      - R.A. WIGGINS          9/27/64
*
*
*                       ----USAGE----
*
* TRANSFER VECTOR CONTAINS ROUTINES - (NOT ANY)
*        AND FORTRAN SYSTEM ROUTINES - (NOT ANY)
*
*
* FORTRAN USAGE
*
*     CALL MATML3 (N,M,L,AA,BB,ZFNBTR,CC,GZFADD)
*
*
* INPUTS
*
*    N          NUMBER OF ROWS IN  A  AND  C.
*               MUST BE GRTHN= 1
*
*    M          NUMBER OF COLUMNS IN  A,  ROWS IN  B.
*               MUST BE GRTHN= 1
*
*    L          NUMBER OF COLUMNS IN  B  AND  C.
*               MUST BE GRTHN= 1
*
*    AA(I)      I=1,....,N*M  CONTAINS THE MATRIX  A(I,J) I=1...N,
*                 J=1...M  STORED CLOSELY SPACED BY COLUMNS.
*
*    BB(I)      I=1,....,M*L  CONTAINS THE MATRIX  B(I,J) I=1...M,
*                 J=1...L  STORED EITHER BY ROWS OR COLUMNS.
*
*    ZFNBTR     =0. IF  BB  IS STORED BY COLUMNS.
*               NOT=0. IF  BB  IS STORED BY ROWS.
*
*    GZFADD     GRTHN 0. IF THE PRODUCT IS TO BE ADDED INTO THE OUTPUT
*                 AREA.
*               LSTHN=0. IF THE PRODUCT REPLACES THE OUTPUT AREA.
*
*
* OUTPUTS
*
*    CC(I)      I=1,....,N*L  CONTAINS THE PRODUCT MATRIX  C(I,J).
```

```
*                   I=1...N, J=1...L   STORED CLOSELY SPACED BY
*                   COLUMNS.
*
*
* EXAMPLES
*
* 1. INPUTS  - N=1 M=1 L=1  AA(1)=2.  MM(1)=3.  ZENBTR=0.  GZFADD=0.
*    OUTPUTS - CC(1)=6.
*
* 2. INPUTS  - N=3 M=2 L=2  AA(1...6) = 1.,1.,3.,2.,7.,1.
*              ZENBTR=0.    MM(1...4) = 1.,5.,3.,7.
*              GZFADD=1.    CC(1...6) = 1.,0.,0.,0.,0.,0.
*    OUTPUTS - CC(1...6) = 17.,36.,8.,17.,52.,16.
*
* 3. INPUTS  - SAME AS EXAMPLE 2. EXCEPT  ZENBTR 1.  GZFADD=0.
*    OUTPUTS - CC(1...6) = 7.,22.,6.,19.,54.,22.
*
*
* PROGRAM FOLLOWS BELOW
*
XR1     HTR     0
XR2     HTR     0
XR4     HTR     0
        BCI     1,MATML3
MATML3  SXD     XR4,4
        SXD     XR1,1
        SXD     XR2,2
        CLA     4,4
        ADD     K1
        STA     AA
        CLA     5,4
        ADD     K1
        STA     BB
        CLA     7,4
        ADD     K1
        STA     LPB
        STA     CC
        STA     CC1
        LDQ     NOP
        CLA*    8,4
        CAS     ZERO
        TRA     *+3
NOP     NOP
        LDQ     STZ
        SLQ     LPA
        CLA*    1,4
        STD     N
        STD     N1
        XCA
        MPY*    2,4
        ARS     1
        PAX     ,1
        SXA     BOT,1
        CLA*    2,4
        LDQ     K10
        ZET*    6,4
        XCA
        STD     MURD
        XCA
        STD     DORM
        LDQ*    2,4
        MPY*    3,4
        ALS     17
        STD     BOT
        LDQ*    1,4
        MPY*    3,4
        ARS     1
        PAX     ,4
N1      TIX     *+2,1,**           *0*N
        AXT     0,1
        SXD     NM1,1
        LXA     BOT,1
        LXD     BOT,2
        TRA     LPB
LOOP    TIX     NM1,1,1
```

```
         LXA     BOT,1
         LXD     BOT,2
MORO     TIX     *+1,2,**        ***M OR I
         SXD     BOT,2
         TRA     L R
NMI      TXI     *+1,1,**        ***N*(M-1)
         LXD     BOT,2
LPB      STZ     **,4            ***ADR(CC), NOP IF GZFADD LSTHN=0.
AA       LDQ     **,1            ***ADR(AA)
BB       FMP     **,2            ***ADR(BB)
CC       FAD     **,4            ***ADR(CC)
CCI      STO     **,4            ***ADR(CC)
OORM     TIX     *+1,2,**        ***I OR M
N        TIX     AA,1,**         ***N
         TIX     LOOP,4,1
         LXD     XR1,1
         LXD     XR2,2
         LXD     XR4,4
         TRA     9,4
BOT      PZE     **,,**          ***N*M,,***M*L AT BEGINNING.
ZERO     PZE
STZ      STZ     ,,
KID      PZE     ,,1
KI       PZE     1,,
         END
```

```
*     POMAIN (SUBROUTINE)
*     LABEL
CPOMAIN
      SUBROUTINE POMAIN (NRCA,LPA,A,LADJ,ADJUG,DETPOL,SPACE)
C
C                    ----ABSTRACT----
C
C TITLE - POMAIN
C        POLYNOMIAL MATRIX ADJUGATE AND DETERMINANT
C
C              SUBROUTINE POMAIN FINDS THE ADJUGATE AND DETERMINANT OF
C              A LAMBDA-MATRIX (A LAMBDA-MATRIX IS MATRIX IN WHICH EACH
C              TERM IS A POLYNOMIAL, OR EQUIVALENTLY, IT IS A POLYNOMIAL
C              HAVING MATRIX COEFFICIENTS). THE METHOD USED IS AN
C              EXTENSION OF THE INVERSION TECHNIQUE DESCRIBED BY
C              FADDEJEW AND SOMINSKI (PROBLEM-BOOK IN HIGHER ALGEBRA,
C              SECOND EDITION, GOSTECHISDAT 1949). THIS EXTENSION WAS
C              MADE BY J. CLAERBOUT, STATISTICS INSTITUTE, UNIVERSITY
C              OF UPPSALA, SWEDEN, AND B. JANSSON, RESEARCH INSTITUTE
C              OF NATIONAL DEFENSE, STOCKHOLM, SWEDEN.
C
C LANGUAGE    - FORTRAN II SUBROUTINE
C EQUIPMENT   - 709 OR 7090 (MAIN FRAME ONLYI
C STORAGE     - 272 REGISTERS
C SPEED       -
C AUTHOR      - R.A. WIGGINS AND  BO JANSSON  1/64
C
C                    ----USAGE----
C
C TRANSFER VECTOR CONTAINS ROUTINES - BRAINV,MOVE,SETKS
C        AND FORTRAN SYSTEM ROUTINES - NONE
C
C FORTRAN USAGE
C     CALL POMAIN(NRCA,LPA,A,LADJ,ADJUG,DETPOL,SPACE)
C
C INPUTS
C
C    NRCA       NUMBER OF ROWS OR COLUMNS IN THE MATRICES OF A.
C               MUST BE GRTHN= 1
C
C    LPA        LENGTH OF THE POLYNOMIALS IN A (I.E. THE NUMBER OF MATRIX
C                 POLYNOMIAL COEFFICIENTS IN A).
C               MUST BE GRTHN= 1
C
C    A(I)       I=1...NRCA,1...NRCA,1...LPA IS THE VECTOR OF MATRIX
C                 COEFFICIENTS FOR THE POLYNOMIAL.
C               MUST BE STORED CLOSELY PACKED.
C
C    SPACE(I)   I=1...NRCA*NRCA*((NRCA)*(LPA-1)+1)  IS COMPUTATION SPACE
C                 NEEDED BY POMAIN.
C
C OUTPUTS
C
C    LADJ       LENGTH OF POLYNOMIALS IN ADJUGATE OF A.
C               = (NRCA)*(LPA-1)+1
C
C    ADJUG(I)   I=1...NRCA,1...NRCA,1...LADJ IS THE VECTOR OF MATRIX
C                 COEFFICIENTS FOR THE ADJUGATE OF A.
C
C    DETPOL(I)  I=1...LADJ+LPA-1 IS THE VECTOR OF COEFFICIENTS OF THE
C                 POLYNOMIAL DETERMINANT OF A.
C
C EXAMPLES
C
C 1. INPUTS   - NRCA = 1  LPA = 2  A(1...1,1...1,1...2) = 1.,1.
C    OUTPUTS  - LADJ = 1  ADJUG(1...1,1...1,1...1) = 1.
C               DETPOL(1...2) = 1.,1.
C
C 2. INPUTS   - NRCA = 2  LPA = 2
C               A(1...2,1...2,1...2) = 1.,1.,0.,1.,4.,0.,6.,3.
C    OUTPUTS  - LADJ = 2  DETPOL (1...3) = 1.,1.,12.
C               ADJUG(1...2,1...2,1...2)  = 1.,-1.,0.,1.,3.,0.,-6.,4.
C
C 3. INPUTS   - NRCA = 2  LPA = 1
C               A(1...2,1...2,1...1) = 1.,2.,3.,3.
```

****
*   PDMAIN   *
****
(PAGE 2)


```
      C     OUTPUTS - LADJ = 1    DETPOL(1) = -3.
      C                ADJUG(1...2,1...2,1...1) = 3.,-2.,-3.,1.
      C
      C PROGRAM FOLLOWS BELOW
      C
            DIMENSION A(8),ADJUG(8),      DETPOL(2),SPACE(8)
            CALL SETKS (NRCA,N,LPA,LA)
            CALL SETKS (N*N,NN,N+1,N1,LA,LAD)
            CALL SETKS (NN*LA,NNLA,NNLA,NNLA1)
            IF (N-1) 5,5,6
       5    CONTINUE
            LADJ=1
            ADJUG(1)=1.
            CALL MOVE (LA,A,DETPOL)
            GO TO 80
       6    CALL MOVE  (NNLA,A,SPACE)
       2000 Z=1.
            DO 40 L=1,N
      C CALCULATE COEFFICIENTS P(K) OF CHARACTERISTIC POLYNOMIAL
            K1=1
            DO 20 K=1,NNLA1,NN
            PLK=0.
            NN1=K+NN-1
            DO 10 I=K,NN1,N1
       10   PLK=PLK+SPACE(I)
            DETPOL(K1)=PLK/FLOATF(L)
       20   K1=K1+1
            IF (L-N) 25,50,50
       25   CONTINUE
      C SUBTRACT P(K)*IDENTITY MATRIX
            CALL MOVE (NNLA1,SPACE,ADJUG)
            DO 30 I=1,NN,N1
            K1=1
            DO 30 K=1,NNLA1,NN
            ADJUG(K)=ADJUG(K)-DETPOL(K1)
       30   K1=K1+1
      C MULTIPLY BY INPUT MATRIX
            CALL BRAINY (N,N,LA,A,N,LAD,ADJUG,SPACE)
       2010 Z=1.
            LAO=LAD+LA-1
       40   NNLA1=NNLA1+NNLA-NN
      C  CHANGE SIGN OF DETERMINANT AND ADJUGATE IF N EVEN
       50   CONTINUE
            IF(XMODF(N,2)) 60,60,70
       60   CONTINUE
            DO 61 I=1,LAD
       61   DETPOL(I)=-DETPOL(I)
            DO 62 I=1,NNLA1
       62   ADJUG(I)=-ADJUG(I)
       70   CONTINUE
            LADJ=LAO-LA+1
       80   RETURN
            END
```

```
*     SIMEQC (SUBROUTINE)
*     LABEL
CSIMEQC
      SUBROUTINE SIMEQC (NRADIM,NRAB,NCB,AR,AI,BR,BI,DET,IS,ERR)
C
C
C TITLE - SIMEQC
C         SOLUTION OF COMPLEX SIMULTANEOUS EQUATIONS
C
C
C               SIMEQC  SOLVES THE COMPLEX SIMULTANEOUS EQUATIONS
C
C                         A X  =  B
C
C               FOR  X, WHERE  A  HAS  NRAB  ROWS AND  NRAB  COLUMNS,
C                              B  HAS  NRAB  ROWS AND  NCB   COLUMNS,
C                              X  HAS  NRAB  ROWS AND  NCB   COLUMNS.
C
C               THE SOLUTION MATRIX,  X,  IS STORED IN  A.
C
C               THE SOLUTION OF THE MATRIX EQUATION IS ACCOMPLISHED BY
C               UPPER TRIANGULARIZATION OF THE  A  MATRIX USING A
C               MODIFIED MAXIMUM PIVOT (ONLY THE INDIVIDUAL COLUMNS ARE
C               SEARCHED FOR A MAXIMUM)  FOR EACH REDUCTION.  THE
C               DETERMINANT IS COMPUTED AT THE SAME TIME.  IF THE MATRIX
C               A  IS SINGULAR A ZERO VALUE OF THE DETERMINANT IS
C               RETURNED.
C
C               BOTH THE  A  AND  B  MATRICES ARE DESTROYED BY SIMEQC.
C
C
C LANGUAGE    - FORTRAN II SUBROUTINE
C EQUIPMENT   - 709, 7090, 7094 (MAIN FRAME ONLY)
C STORAGE     - 675 REGISTERS
C SPEED       - RELATIVELY SLOW
C AUTHOR      - R.A. WIGGINS    7/64
C
C
C                    ----USAGE----
C
C
C TRANSFER VECTOR CONTAINS ROUTINES - NOT ANY
C        AND FORTRAN SYSTEM ROUTINES - NOT ANY
C
C
C FORTRAN USAGE
C     CALL SIMEQC (NRADIM,NRAB,NCB,AR,AI,BR,BI,DET,IS,ERR)
C
C
C INPUTS
C
C    NRADIM     THE DIMENSIONED SIZE OF THE COLUMNS OF THE  A  AND  B
C                 MATRICES.  THAT IS, THE CALLING PROGRAM WOULD CONTAIN
C                 STATEMENTS LIKE
C                      DIMENSION AR(NRADIM,IGNORD), AI(NRADIM,IGNORD)
C                      DIMENSION BR(NRADIM,IGNORD), BI(NRADIM,IGNORD)
C                 WHERE  IGNORD  MAY BE ANY CONSISTENT VALUE.  IF THE
C                 MATRICES ARE STORED CLOSELY SPACED THEN  NRADIM = NRAB
C                 AND  AR, AI, BR,  AND  BI  MAY BE DIMENSIONED AS
C                 VECTORS.
C               MUST BE GRTHN=  NRAB.
C
C    NRAB       NUMBER OF ROWS IN  A  AND  B,  NUMBER OF COLUMNS IN  A.
C               MUST BE GRTHN= 1 .
C
C    NCB        NUMBER OF COLUMNS IN  B.
C               MUST BE GRTHN= 1 .
C
C    AR(I,J)    I=1,...,NRAB,  J=1,...,NRAB  CONTAINS THE REAL PART OF  A
C                 STORED BY COLUMNS.
C               NOTE-  AR  IS ALSO AN OUTPUT.
C
C    AI(I,J)    I=1,...,NRAB,  J=1,...,NRAB  CONTAINS THE IMAGINARY PART
C                 OF  A  STORED BY COLUMNS
C               NOTE-  AI  IS ALSO AN OUTPUT.
```

****
*   SIMEQC   *
****
(PAGE 2)

```
C
C     BR(I,J)    I=1,...,NRAB,  J=1,...,NCB  CONTAINS THE REAL PART OF  B
C                  STORED BY COLUMNS.
C                IS DESTROYED BY  SIMEQC.
C
C     BI(I,J)    I=1,...,NRAB,  J=1,...,NCB  CONTAINS THE IMAGINARY PART
C                  OF  B  STORED BY COLUMNS
C                IS DESTROYED BY  SIMEQC.
C
C     SPACE(I)   I=1,...,NRAB  IS TEMPORARY COMPUTATION SPACE.
C
C
C OUTPUTS
C
C     AR(I,J)    I=1,...,NRAB,  J=1,...,NCB  CONTAINS THE REAL PART OF  X
C                  STORED BY COLUMNS.
C
C     AI(I,J)    I=1,...,NRAB,  J=1,...,NCB  CONTAINS THE IMAGINARY PART
C                  OF  X  STORED BY COLUMNS.
C
C     DET(I)     I=1,...,2  CONTAINS THE REAL AND IMAGINARY PARTS, RESPEC-
C                  TIVELY, OF THE DETERMINANT OF  A.
C
C     ERR        =0. IF ALL OK.
C                =2. IF  A  IS SINGULAR (DET(1...2) = 0.).
C
C
C EXAMPLES
C
C 1. INPUTS   - NRADIM=2  NRAB=1  NCB=2
C                AR(1)=2.  AI(1)=1.  BR(1...2)=2.,3.  BI(1...2)=  ,-2.
C     OUTPUTS - AR(1...2)=1.,.8  AI(1...2)=0.,-1.4  DET(1...2)=2.,  .
C                ERR=0.
C
C 2. INPUTS   - NRADIM=2  NRAB=2  NCB=2
C                AR(1...4)=1.,-1.,-1.,-3.  BR(1...4)=1.,0.,0.,1.
C                AI(1...4)=1., 3., 1.,-2.  BI(1...4)=0.,0.,0.,0.
C                   THAT IS,  A = ( ( 1., 1.)  (-1., 1.) )
C                                 ( (-1., 3.)  (-3., 2.) )
C     OUTPUTS - AR(1...4)=-0.5, 2., 1., 0.     DET(1...2) = (-1.,1.)
C                AI(1...4)=-2.5,-1., 0., 1.     ERR=0.
C                   THAT IS,  X = ( (-.5,-2.5)  (1.,0.) )
C                                 ( ( 2.,-1.0)  (0.,1.) )
C
C 3. SAME NUMBERS AS EXAMPLE 2. BUT WITH A DIFFERENT DIMENSION.
C     INPUTS  - NRADIM=3  NRAB=2  NCB=2
C                AR(1...6)=1.,-1.,0.,-1.,-3.,0.
C                AI(1...6)=1., 3.,0., 1.,-2.,0.
C                BR(1...6)=1., 0.,0., 0., 1.,0.
C                BI(1...6)=0., 0.,0., 0., 0.,0.
C     OUTPUTS    AR(1...6)=-0.5, 2.,0.,1.,0.,0.   DET(1...2) = (-1.,1.)
C                AI(1...6)=-2.5,-1.,0.,0.,1.,0.   ERR=0.
C
C 4. SECOND ROW IS COMPLEX CONJUGATE OF FIRST ROW.
C     INPUTS  - NRADIM=2  NRAB=2  NCB=1
C                AR(1...4)= 1., 1.,-1.,-1.  BR(1...2)= 1., 2.
C                AI(1...4)= 1.,-1., 1.,-1.  BI(1...2)= 1.,-2.
C                   THAT IS, A = ( (1., 1.) (-1., 1.) )  B = ( (1., 2.) )
C                                ( (1.,-1.) (-1.,-1.) )      ( (1.,-2.) )
C     OUTPUTS - AR(1...2)=1.5,0.   DET = (0.,-4.)  ERR=0.
C                AI(1...2)= 0.,.5
C
C
C 5. SINGULAR CASE.
C     INPUTS  - NRADIM=2  NRAB=2  NCB=1
C                AR(1...4)= 1.,1.,-1.,-1.  BR(1...2)= 1.,0.
C                AI(1...4)= 1.,1.,-1.,-1.  BI(1...2)= 0.,0.
C     OUTPUTS - DET(1...2)=0.,0.  ERR=2.  AR,AI CONTAIN MEANINGLESS
C                NUMBERS.
C
C
C PROGRAM FOLLOWS BELOW
C
C
C DUMMY DIMENSIONS
```

```
C
      DIMENSION AR(2),AI(2),BR(2),BI(2),DET(2),IS(2)
C
C BRING IN SCALAR VARIABLES
C
      ND=NRADIM
      N=NRAB
      K=NCB
      ERR=0.
C
C IF   NRAB = 1  COMPUTE DIRECTLY AND LEAVE
C
      IF (N-1) 2,2,4
 2    CONTINUE
      DET(1)=AR
      DET(2)=AI
      DETM=(DET(1)*DET(1)+DET(2)*DET(2))
      J=1
      DO 3 I=1,K
      AR(J)=(BR(J)*DET(1)+BI(J)*DET(2))/DETM
      AI(J)=(BI(J)*DET(1)-BR(J)*DET(2))/DETM
 3    J=J+ND
      GO TO 500
 4    CONTINUE
C
C COMPUTE THE NEEDED COMBINATIONS AND SET UP THE INITIAL VALUES.
C
      NND=N*ND
      KND=K*ND
      DET(1)=1.
      DET(2)=0.
      ITR=1
      NDITR1=0
      DO 10 I=1,N
 10   IS(I)=I
C
C  ITR    =INDEX OF TRACE TERM
C  NDITR1=ND*(ITR-1)
C
C
C FIND LARGEST VALUE IN THIS COLUMN WITH INDEX GRTHN= ITR
C
 100  CONTINUE
      IR=ITR
      AMX=0.
 110  CONTINUE
      ISIR=IS(IR)
      ISIRA=ISIR+NDITR1
      AMXT=AR(ISIRA)*AR(ISIRA)+AI(ISIRA)*AI(ISIRA)
      IF (AMX-AMXT)  120,130,130
 120  CONTINUE
      AMX=AMXT
      ICC=ISIR
      IS(IR)=IS(ITR)
      IS(ITR)=ICC
 130  CONTINUE
      IR=IR+1
      IF (IR-N)  110,110,140
 140  CONTINUE
      ICC=IS(ITR)
 2000 Q=1.
C
C DIVIDE ALL TERMS TO THE RIGHT OF ICC BY DIAGONAL TERM
C
      ICCA=ICC+NDITR1
      ARITR=AR(ICCA)
      AIITR=AI(ICCA)
 200  CONTINUE
      ICCA=ICCA+ND
      IF (ICCA-NND)  210,210,220
 210  CONTINUE
      ARICCA=AR(ICCA)
      AIICCA=AI(ICCA)
      AR(ICCA)=(ARICCA*ARITR+AIICCA*AIITR)/AMX
      AI(ICCA)=(AIICCA*ARITR-ARICCA*AIITR)/AMX
```

```
       GO TO 200
  220  CONTINUE
       ICCB=ICC
  230  CONTINUE
       BRICCB=BR(ICCB)
       BIICCB=BI(ICCB)
       BR(ICCB)=(BRICCB*ARITR+BIICCB*AIITR)/AMX
       BI(ICCB)=(BIICCB*ARITR-BRICCB*AIITR)/AMX
       ICCB=ICCB+ND
       IF (ICCB-KND)  230,230,240
  240  CONTINUE
 2010  Q=1.
C
C INCREMENT THE DETERMINANT
C
       DET1=DET(1)
       DET2=DET(2)
       DET(1)=DET1*ARITR-DET2*AIITR
       DET(2)=DET1*AIITR+DET2*ARITR
C
C SUBTRACT THIS ROW FROM ALL SUCCEEDING ROWS
C
       IR=ITR
  250  CONTINUE
 2020  Q=1.
       IR=IR+1
       IF (IR-N)  260,260,310
  260  CONTINUE
       ICCA=IS(IR)+NDITRI
       ICCS=ICC+NDITRI
       ARICC=AR(ICCA)
       AIICC=AI(ICCA)
  270  CONTINUE
       ICCA=ICCA+ND
       IF (ICCA-NND)  280,280,290
  280  CONTINUE
       ICCS=ICCS+ND
       AR(ICCA)=AR(ICCA)-(AR(ICCS)*ARICC-AI(ICCS)*AIICC)
       AI(ICCA)=AI(ICCA)-(AR(ICCS)*AIICC+AI(ICCS)*ARICC)
       GO TO 270
  290  CONTINUE
       ICCB=IS(IR)
       ICCS=ICC
  300  CONTINUE
       BR(ICCB)=BR(ICCB)-(BR(ICCS)*ARICC-BI(ICCS)*AIICC)
       BI(ICCB)=BI(ICCB)-(BR(ICCS)*AIICC+BI(ICCS)*ARICC)
       ICCB=ICCB+ND
       ICCS=ICCS+ND
       IF (ICCB-KND)  300,300,250
  310  CONTINUE
C
C LOOP TO NEXT ROW
C
       ITR=ITR+1
       NDITRI=NDITRI+ND
       IF (ITR-N)  100,100,320
  320  CONTINUE
C
C IF DETERMINANT IS ZERO, LEAVE
C
       IF (ABSF(DET(1))+ABSF(DET(2))-1.E-6)  999,999,330
  330  CONTINUE
C
C NOW CLEAR THE UPPER TRIANGLE
C
       ITR=N
       NDITRI=NND-ND
  340  CONTINUE
       IR=ITR
  345  CONTINUE
 2030  Q=1.
       IR=IR-1
       IF (IR) 370,370,350
  350  CONTINUE
       ICCB=IS(IR)
```

****
*   SIMEQC   *
****
(PAGE   5)


```
      ICCS=IS(ITR)
      ICCA=ICCB+ND(ITR)
      ARICC=AR(ICCA)
      AIICC=AI(ICCA)
  360 CONTINUE
      BR(ICCB)=BR(ICCB)-(BR(ICCS)*ARICC-BI(ICCS)*AIICC)
      BI(ICCB)=BI(ICCB)-(BI(ICCS)*ARICC+BR(ICCS)*AIICC)
      ICCB=ICCB+ND
      ICCS=ICCS+ND
      IF (ICCS-KND)  360,360,345
  370 CONTINUE
      ITR=ITR-1
      ND(ITR)=ND(ITR)-ND
      IF (ITR-1)  380,380,340
  380 CONTINUE
C
C UNSCRAMBLE B INTO A
C
      IR=1
      NDIC=0
  390 CONTINUE
      IA=NDIC+1
  400 CONTINUE
      ISIRA=IS(IR)+NDIC
      AR(IA)=BR(ISIRA)
      AI(IA)=BI(ISIRA)
      IA=IA+1
      IR=IR+1
      IF (IR-N)  400,400,410
  410 CONTINUE
      IR=1
      NDIC=NDIC+ND
      IF (NDIC-KND)  390,420,420
  420 CONTINUE
C
C THAT'S ALL
C
  500 CONTINUE
      RETURN
  999 CONTINUE
      ERR=2.
      GO TO 500
      END
```

```
*     SNLMAC (SUBROUTINE)
*     LABEL
CSNLMAC
      SUBROUTINE SNLMAC (NZ,ZR,ZI,NRU,UR,UI,LA,AA,SPACE,ERR)
C
C
C                    ----ABSTRACT----
C
C TITLE - SNLMAC
C         SYNTHESIZE LAMBDA MATRIX FROM COMPLEX VECTORS AND ZEROS.
C
C
C                SNLMAC  CONSTRUCTS A LAMBDA (POLYNOMIAL) MATRIX FROM
C                THE COMPLEX LATENT VECTORS  U(I)  AND LATENT ZEROS  Z(I).
C                SINCE THE POLYNOMIAL MATRIX WILL BE REAL, THE NON-REAL
C                LATENT VECTORS AND ZEROS MUST APPEAR IN CONJUGATE PAIRS.
C
C                IF WE FORM THE MODAL MATRIX
C
C                                     M
C                             ( (U(1)) )
C                    T        ( (U(2)) )
C                   U    =    (    .   ) M
C                             (    .   )
C                             ( (U(M)) )
C
C                AND THE SPECTRAL MATRIX
C
C                   Z   =  DIAG ( Z(1),Z(2),...,Z(M) )
C
C                THEN THE  N BY N  MATRIX COEFFICIENTS, A(I), I=1,...,L,
C                ARE GIVEN BY THE SIMULTANEOUS EQUATIONS
C
C                                          L
C                   A(1)*U*Z +...+ A(L)*U*Z    =  U.
C
C                WHERE  L*N = M .
C                SNLMAC  SETS UP AND SOLVES THESE EQUATIONS.
C
C
C  LANGUAGE  - FORTRAN II SUBROUTINE
C EQUIPMENT  - 709, 7090, 7094 (MAIN FRAME ONLY)
C STORAGE    - 348 REGISTERS
C SPEED      -
C AUTHOR     - R.A. WIGGINS    8/64
C
C
C                    ----USAGE----
C
C TRANSFER VECTOR CONTAINS ROUTINES - DIXMAC, MATRA, MOVREV, SIMEQC
C       AND FORTRAN SYSTEM ROUTINES - NOT ANY
C
C
C FORTRAN USAGE
C     CALL SNLMAC (NZ,ZR,ZI,NRU,UR,UI,LA,AA,SPACE,ERR)
C
C
C INPUTS
C
C    NZ          NUMBER OF LATENT ZEROS AND VECTORS.
C                = M IN THE ABSTRACT.
C                MUST BE GRTHN= 1
C          *
C    ZR(I)       I=1,...,NZ  CONTAINS THE REAL PARTS OF THE LATENT ZEROS
C                  Z(I).
C          *
C    ZI(I)       I=1,...,NZ  CONTAINS THE IMAGINARY PARTS OF THE LATENT
C                  ZEROS  Z(I).
C
C    NRU         NUMBER OF ELEMENTS IN EACH VECTOR  U(I).
C                MUST BE GRTHN= 1, MUST BE A MULTIPLE OF  NZ.
C          *
C    UR(I)       I=1,...,NRU*NZ  CONTAINS THE REAL PARTS OF THE LATENT
C                  VECTORS  U(I)  STORED CLOSELY SPACED.
C          *
```

```
C     UI(I)       I=1,....,NRU*NZ  CONTAINS THE IMAGINARY PARTS OF THE
C                   LATENT VECTORS  U(I)  STORED CLOSELY SPACED
C
C    *THE LATENT VECTOR  U(I)  CORRESPONDS TO THE LATENT ROOT  Z(I).
C     BOTH VECTORS MUST CONTAIN THE COMPLEX CONJUGATES EXPLICITLY.
C
C     SPACE(I)  I=1,....,2*NZ*2*(NZ*NRU)*NRU  IS TEMPORARY COMPUTATION
C                   SPACE.
C
C
C OUTPUTS
C
C     LA          IS THE NUMBER OF COEFFICIENT MATRICES FOUND.
C                 =(NRU/NZ)+1
C                 = L+1  IN THE ABSTRACT.
C
C     AA(I)       I=1,....,NRU*NRU*LA  CONTAINS THE COEFFICIENT MATRICES
C                   A(I), I=0...L.  STORED CLOSELY SPACED BY COLUMNS WHERE
C                   A(0) = IDENTITY MATRIX.
C
C     ERR         =0.  IF ALL OK.
C                 =2.  IF THE SIMULTANEOUS EQUATIONS ARE SINGULAR - THIS
C                        HAPPENS IF  U(I)=U(J)  WHEN  Z(I)=Z(J).
C
C
C EXAMPLES
C
C 1. INPUTS   - NZ=2  ZR(1...2)=2.,3.  ZI(1...2)=0.,0.
C               NRU=1  UR(1...2)=1.,1.  UI(1...2)=0.,0.
C    OUTPUTS - LA=3  AA(1...3)=1.,-5.,6.  ERR=0.
C
C 2. INPUTS   - NZ=2  ZR(1...2)=2.,2.  ZI(1...2)=1.,-1.  NRU=2
C               UR(1...4)=5.,-3.,5.,-3.  UI(1...4)=0.,9.,0.,-9.
C    OUTPUTS - LA=2  AA(1...8)=1.,0.,0.,1.,-.33333,-.4,.11111,-.46667
C               ERR=0.
C
C 3. INPUTS   - NZ=4  ZR(1...4)=2.,2.,5.,5.  ZI(1...4)=1.,-1.,-4.,4.
C               NRU=2  UR(1...8)=5.,-3.,5.,-3.,40.,39.,40.,39.
C                      UI(1...8)=0., 9.,0.,-9., 0.,-1., 0., 1.
C    OUTPUTS - LA=3  ERR=0.
C               AA(1...12)=1.,0.,0.,1.,-.31364, .51018, .07061,-.73026,
C                                         .00644,-.19796, .01735, .22414
C
C
C PROGRAM FOLLOWS BELOW
C
C
C DUMMY DIMENSIONS
C
      DIMENSION ZR(2),ZI(2),UR(2),UI(2),AA(2),SPACE(2)
C
C BRING IN THE SCALAR PARAMETERS
C
      L=NZ
      N=NRU
C
C SET UP THE COMBINATIONS NEEDED
C
      M=L/N
      LA=M+1
      LN=L*N
      NN=N*N
      NN1=NN+1
      LL=L*L
      IAI=3
      IAR=IAI+LL
      IBI=IAR+LL
      IBR=IBI+LN
      ISP=IBR+LN
      IURI=IBR
      IUII=IBI
      IARI=IAR
      IAII=IAI
C
C SET UP RIGHT SIDE OF SIMULTANEOUS EQUATIONS
```

BLANK PAGE

```
C
      CALL MATRA (UR,N,L,SPACE(IBR))
      CALL MATRA (UI,N,L,SPACE(IBI))
C
C SET UP LEFT SIDE OF SIMULTANEOUS EQUATIONS
C
      DO 10  I=1,M
      CALL DIXMAC (ZR,ZI,L,N,SPACE(IURI),SPACE(IUII),SPACE(IARI),
     1 SPACE(IAII))
      IURI=IARI
      IUII=IAII
      IARI=IARI+LN
 10   IAII=IAII+LN
 2000 G=1.
C
C SOLVE THE EQUATIONS
C
      CALL SIMEQC (L,L,N,SPACE(IAR),SPACE(IAI),SPACE(IBR),SPACE(IBI),
     1 SPACE,SPACE(ISP),ERR)
C
C TRANSPOSE THE OUTPUT AND CHANGE THE SIGN OF OUTPUT
C
      CALL MATRA  (SPACE(IAR),L,N,AA(NN1))
      CALL MOVREV (LN,1,AA(NN1),1,AA(NN1),-1)
C
C PUT AN IDENTITY MATRIX AT THE BEGINNING
C
      CALL MOVREV (NN,0,0,1,AA,1)
      CALL MOVREV (N,0,1.,N+1,AA,1)
C
C THAT'S ALL
C
      RETURN
      END
```

REFERENCES

Anstey, N.A., 1964, Correlation techniques - a review: Geophys. Prosp. v. 12, no. 4, p. 355-382.

Army Map Service, 1959, Statistical and harmonic analysis of gravity: Tech. Rept. 24, Project No. MO-011, p. 58.

Bellman, R., 1960, Introduction to matrix analysis: McGraw-Hill Book Co., Inc.

Backus, M., Burg, J., Baldwin, D., Bryan, E., 1964, Wide band extraction of mantle P waves from ambient noise: Geophysics, v. 29, no. 5, p. 672-692.

Bodewig, E., 1956, Matrix calculus: North-Holland Pub. Co., Amsterdam.

Burg, J. P., 1964, Three dimensional filtering with an array of seismometers: Geophysics, v. 29, no. 5, p. 693-713.

Claerbout, J. F., 1963, Digital filters and applications to seismic detection and discrimination: Sci. Rep. 5 of Contract AF19(604)7378 ARPA Project VELA UNIFORM, M. I. T.

__________________ 1964, Detection of p-waves from weak sources at great distances: Geophysics, v. 29, no. 2, p. 197-211.

Claerbout, J. F., and Robinson, E. A., 1964, The error in least-squares inverse filtering: Geophysics, v. 29, no. 1, p. 118-120.

Embree, P., Burg, J. P., and Backus, M. M., 1963, Wide band velocity filtering: Geophysics, v. 28, p. 948-974.

Fail, M. D., and Grau, G., 1963, Les filtres en eventail: Geophys. Prosp., v. 11, p. 131-163.

Fejer, L., 1916, Uber trigonometrische Polynome: Z. reine u. angew. Mat., v. 146, p. 53-82.

Frazer, R. A., Duncan, W. J., and Collar, A. R., 1947, Elementary matrices: The MacMillon Co., New York.

Galbraith, J. N., Jr., 1963, Computer studies of microseism statistics with applications to prediction and detection: Sci. Rept. 6 of Contract AF19(604)7378, ARPA Project VELA UNIFORM, M. I. T.

Gantmacher, F. R., 1959, The theory of matrices: Chelsea Publishing Co. New York, v. I.

Geronimus, Ya. L., 1960, Polynomials orthogonal on a circle and interval: Pergamon Press, London.

Helson, H., and Lowdenslager, D., 1958, Prediction theory and Fourier series in several variables: Acta Math., 99, p. 165-202.

Hoffman, K., and Kunze, R., 1961, Linear algebra: Prentice-Hall, Englewood Cliffs, N. J., 323 p.

Jury, E. I., 1964, Theory and application of the z-transform method: John Wiley and Sons, Inc., New York.

Karhunen, K., 1947, Uber lineare Methoden in derWahrscheinlichkeitsrechnung: Ann. Acad. Sci. Fennicae Ser. A, I. Math. Phys. 37, p. 1-79.

____________ 1949, Uber die Strukture stationarer zufalliger Funktionen: Arkiv for Matematik, 1, p. 141-160.

Kolmogorov, A., 1941a, Stationary sequences in Hilbert space. (Russion): Bull. Math. Univ., Moscow, v. 2, no. 6, 40 p. (English translation by Natasha Artin.)

____________ 1941b, Interpolation und Extrapolation von staionaren zufalligen Folgen: Bull. Acad. Sci. (Nauk) U.S.S.R. Ser. Math., v. 5, p. 3-14.

Kunetz, G., 1964, Generalisation des operateurs d'anti-resonance a un nombre quelconque de reflecteurs: Geophys. Prosp., v. 12, no. 3. p. 283-289.

Lee, Y. W., 1960, Statistical theory of communications: John Wiley & Sons, Inc., New York, 501 p.

Levinson, N., 1947, The Wiener RMS (root mean square) error criterion in filter design and prediction: Appendix B of Wiener (1949), p. 129-148.

Masani, P., 1960, The prediction theory of multivariate stochastic process III: Acta Math., v. 30, no. 104.

McMillan, B., 1952, Introduction to formal realizability theory - II: Bell System Tech. Jour. 31, p. 541-600.

Neumann, J. von, 1950, Functional operators, vol II: the geometry of orthogonal spaces: Princeton University Press, Princeton, N. J.

Oona, Y., and Yasuura, K., 1954, Synthesis of finite passive 2n-terminal networks with prescribed scattering matrices: Mem. Kyushu Univ. (Engineering), Japan, v. 14, no. 2, p. 125-177.

Quenouille, M. H., 1957, The analysis of multiple time-series: Griffins Statistical Monographs & Courses, Hafner Publishing Co., New York.

Riesz, F., 1907, Sur les systemes orthogonaux de fonctions: C. R. Acad. Sci. Paris, 144, p. 615-734.

Riesz, F., and Sz.-Nagy, B., 1952, Lecons d'analyse fonctionelle: English trans., F. Ungar, New York.

Robinson, E. A., 1954, Predictive decomposition of time series with application to seismic exploration: M.I.T. G.A.G. Rept. 7.

_______________ 1962, Random wavelets and cybernetic systems: Griffins Statistical Monographs & Courses, Hafner Publishing Co., New York.

_______________ 1963a, Mathematical development of discrete filters for detection of nuclear explosions: Jour. Geophys. Res., v. 68, p. 5559-5567.

_______________ 1963b, Structural properties of stationary stochastic processes with applications: in Time Series Analysis edited by M. Rosenblatt, John Wiley & Sons, Inc., p. 170-196.

Robinson, E. A., and Treitel, S., 1964, Principles of digital filtering: Geophysics, v. 29, no. 3, p. 395-404.

Schneider, W. A., Larner, K. L., Burg, J. P., and Backus, M. M., 1964, A new data processing technique for the elimination of ghost arrivals on reflection seismograms: Geophysics, v. 29, no. 5, p. 783-805.

Simpson, S. M., Jr., 1965, Time series computations in Fortran and Fap - Vol. I - A program library: to be published by Addison-Wesley, Reading, Mass.

Simpson, S. M., Jr., Director, 1961a, Initial studies on underground nuclear detection with seismic data prepared by a novel digitization system: Sci. Rept. 1 of Contract AF 19(604)7378 ARPA Project VELA UNIFORM.

__________ 1961b, Time series techniques applied to underground nuclear detection and further digitized seismic data: Sci. Rept. 2 of Contract AF 19(604)7378 ARPA Project VELA UNIFORM.

__________ 1962a, Continued numerical studies on underground nuclear detection and further digitized seismic data: Sci. Rept. 3 of Contract AF 19(604)7378 ARPA Project VELA UNIFORM.

__________ 1962b, Magnetic tape copies of M.I.T. geophysics program set I (Time series programs for the IBM 709, 7090): Sci. Rept. 4 of Contract AF 19(604)7378 ARPA Project VELA UNIFORM.

Simpson, S. M., Jr., Robinson, E. A., Wiggins, R. A., and Wusch C. I., 1963, Studies in optimum filtering of single and multiple stochastic processes: Sci. Rept. 7 of Contract AF 19(604)7378 ARPA Project VELA UNIFORM.

Szego, G., 1939, Orthogonal polynomials: American Mathematical Society, New York.

Wadsworth, G. P., Robinson, E. A., Bryan, J. G., and Hurley, P. M., 1953, Detection of reflections on seismic records by linear operators: Geophysics, v. 18, p. 539-586.

Whittle, P., 1954, On stationary processes in the plane: Biometrika no. 41, pp 434-449.

__________ 1956, On the variation of yield variance with plot size: Biometrika no. 43, p. 337-343.

__________ 1963, Prediction and regulation by linear least-squares methods: The English Universities Press Ltd., London.

Wiener, N., 1949, Extrapolation, interpolation, and smoothing of stationary time series: The Technology Press of M.I.T.,Cambridge, Mass.

Wiener, N., and Masani, P., 1957, The prediction theory of multivariate stochastic processes: Acta Math., v. 98, p. 111-180.

__________ 1958, The prediction theory of multivariate stochastic processes II: Acta Math., v. 99, pp 93-137.

Wiggins, R. A., and Robinson, E. A., 1965, Recursive solutions to the multichannel filtering problem: To be published in Jour. of Geophys. Res.

Wold, H. O. A., 1938, Stationary time series: Almqvist and Wiksell.

Wolf, E., (Editor), 1964, Progress in optices, Vol. III: John Wiley & Sons, New York.

Wunsch, C. I., 1965, Multiple spectral factorization method of Wiener and Masani: To be published in IEEE, PTGIT.

Yaglom, A. M., 1960, Effective solutions of linear approximation problems for multivariate stationary processes with rational spectrum: Theory of Probability and Its Applications, v. 5, no. 3.

Youla, D. C., 1961, On the factorization of rational matrices: IRE Trans. IT-7, p. 172-189.

Zadeh, L. A., and Desoer, C. A., 1963, Linear system theory: the state space approach: McGraw-Hill Book Co., Inc., New York.

BIOGRAPHICAL NOTE

PII Redacted

The author [redacted] He completed high school at the Broadwater Public Schools in 1957. From 1957 until 1961 he was enrolled at the Colorado School of Mines. He received the William B. Waltman award and the Society of Exploration Geophysicist's "Outstanding Senior Geophysicist" award during his senior year. In June, 1961, he graduated with honors with a Geophysical Engineering degree. He entered the Graduate School at the Massachusetts Institute of Technology in the Department of Geology and Geophysics in September, 1961, and has since been a research assistant under Dr. S. M. Simpson, Jr. He is the co-author, with Enders A. Robinson, of the paper "Recursive Solution to the Multichannel Filtering Problem" to be published in the Journal of Geophysical Research.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstra[ct an]d indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Dept. of Geology and Geophysics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

ON FACTORING THE CORRELATIONS OF DISCRETE MULTIVARIABLE STOCHASTIC PROCESSES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Scientific Report

5. AUTHOR(S) *(Last name, first name, initial)*

Wiggins, Ralph A.

6. REPORT DATE

February 28, 1965

7a. TOTAL NO. OF PAGES

196

7b. NO. OF REFS

60

8a. CONTRACT OR GRANT NO.

AF 19(604)7378

b. PROJECT NO.

8652

c. TASK

865203

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Number 9

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

AFCRL - 65-207

10. AVAILABILITY/LIMITATION NOTICES

Agencies of the Dept. of Defense, their contractors, and Government agencies may obtain copies of this report from DDC. All others U.S. DEPT. OF COMMERCE, OFFICE OF TECH. SERVICES

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Air Force Camb. Res. Labs., Office of Aerospace Research, U.S. Air Force, Bedford, Massachusetts

13. ABSTRACT

This thesis is an extension of the theory of discrete scalar time series analysis to multivariable processes. This extension is facilitated by expanding the algebra of polynomial matrices (matrices with polynomial elements).

Multivariable processes may have a multiplicity of either the independent or the dependent variable. Such processes are called multi-dimensional or multi-channel, respectively. All multi-dimensional processes may be formally mapped into matrix notation. Once this mapping is made the properties of all multivariable linear operators and autocorrelations can be studied in terms of the polynomial matrices that represent their z-transforms.

Polynomial matrices can be decomposed into three related forms: the spectral factorization, the Smith-McMillan canonical form, or the Robinson canonical form. Each of these representations leads to the concept of an invertible or minimum delay wavelet.

The algorithms for finding the spectral factorization and for finding the Smith-McMillan canonical form can be extended to

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Multivariable correlation factorization						
Polynomial matrix algebra						
Multivariable least-squares filters						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

13. Abstract (Continued)

provide an analytic factorization of a multi-channel autocorrelation in terms of invertible wavelets. In addition the autocorrelation may be approximately factored by a recursive least-squares algorithm, or by a projection technique.

Of the factorization methods available, the recursive algorithm is the most efficient and is therefore extended to include the more general problem of signal shaping in the presence of noise.

Finally, as an illustration, the problem of designing a finite optimum two-dimensional band-pass, band-reject filter is solved and the characteristics of a few particular realizations of such filters are presented. (U)