

UNCLASSIFIED

AD NUMBER

AD845643

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; DEC 1968. Other requests shall be referred to Air Force Technical Applications Center, Washington, DC. This document contains export-controlled technical data.

AUTHORITY

usaf, ltr, 28 feb 1972

THIS PAGE IS UNCLASSIFIED

AD845643

A THEORETICAL DISCUSSION OF OPTIMUM MULTICHANNEL  
FILTER DESIGN

10 December 1968

Prepared For

AIR FORCE TECHNICAL APPLICATIONS CENTER  
Washington, D. C.

By

E. A. Flinn  
TELEDYNE, INC.

Under

Project VELA UNIFORM

JAN 7 1969

Sponsored By

ADVANCED RESEARCH PROJECTS AGENCY  
Nuclear Test Detection Office  
ARPA Order No. 624

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Chief, AFTAC.

# DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST  
QUALITY AVAILABLE.

COPY FURNISHED CONTAINED  
A SIGNIFICANT NUMBER OF  
PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.

A THEORETICAL DISCUSSION OF OPTIMUM MULTICHANNEL  
FILTER DESIGN

SEISMIC DATA LABORATORY REPORT NO. 227

AFTAC Project No.: VELA T/6702  
Project Title: Seismic Data Laboratory  
ARPA Order No.: 624  
ARPA Program Code No.: 8F10  
  
Name of Contractor: TELEDYNE INDUSTRIES, INC.  
  
Contract No.: F 33657-68-C-0945  
Date of Contract: 2 March 1968  
Amount of Contract: \$ 1,251,000  
Contract Expiration Date: 1 March 1969  
Project Manager: Royal A. Hartenberger  
(703) 836-7647

P. O. Box 334, Alexandria, Virginia

AVAILABILITY

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Chief, AFTAC.

*Attn: VSC  
2/10/68 DE 170333*

This research was supported by the Advanced Research Projects Agency, Nuclear Test Detection Office, under Project VELA-UNIFORM and accomplished under technical direction of the Air Force Technical Applications Center under Contract F 33657-68-C-0945.

Neither the Advanced Research Projects Agency nor the Air Force Technical Applications Center will be responsible for information contained herein which may have been supplied by other organizations or contractors, and this document is subject to later revision as may be necessary.

### Abstract

The design equations for both single-channel and multi-channel optimum least-squares ("Wiener") filters are derived and discussed. Specific examples of such filters are presented; for example, inverse filters, signal/noise ratio enhancement filters, prediction filters, and maximum-likelihood filters. The single-channel and multichannel Levinson recursion algorithms for solving the design equations are discussed.

## Table of Contents

	Page No.
ABSTRACT	
INTRODUCTION	
SINGLE-CHANNEL WAVE-SHAPING NORMAL EQUATIONS	2
Minimum error energy	7
The inverse filter	8
The prediction filter	10
General characteristics of the wave-shaping filter	15
The single-channel recursion algorithm	16
Summary of single-channel recursion	22
NORMAL EQUATIONS FOR THE MULTICHANNEL WAVE-SHAPING FILTER	23
RECURSION FOR THE MULTICHANNEL FILTER	27
A time-saver	33
Summary of the multichannel recursion	34
SOME SPECIAL CASES OF MULTICHANNEL FILTERS	35
Multichannel prediction filter	35
Array sum prediction	35
Spatial interpolation	36
Equalization	36
Signal-to-noise ratio enhancement	36
THE OPTIMUM LINEAR FILTER WITH CONSTRAINTS	38
The maximum-likelihood filter	40
REFERENCES	46

## Introduction

Levinson (1947) published an algorithm for recursively extending the length of a digital filter, optimum in the least mean-square-error sense, designed to change a given waveform into another desired waveform. The algorithm was extended to the multichannel input case by Robinson, and to the multi-dimensional case by Wiggins (Simpson et al., 1963).

The general wave-shaping problem includes as special cases the design of inverse filters, prediction, interpolation, and smoothing filters, and the algorithm is required in the design of maximum-likelihood filters of various types (Kelly and Levin, 1964; Simpson et al., 1963). Good discussions of optimum filter design have been presented by Claerbout (1963) and Treitel and Robinson (1966).

There are two advantages to using the recursion algorithm instead of designing the full-length filter directly: first, in the computation of a single-channel filter of length  $L$  points, machine storage requirements are reduced from a multiple of  $L^2$  to a multiple of  $L$  words, and the number of arithmetical operations is reduced from a multiple of  $L^3$  to a multiple of  $L^2$ . For the multichannel filter with  $N$  inputs, the storage reduction is from  $N^2L^2$  to  $N^2L$  and the operational reduction is from  $N^3L^3$  to  $N^2L^2$ . It was not generally practical to design multichannel filters before the extension of the single-channel algorithm.

The second advantage is that the mean-square error, the quantity one is minimizing, can easily be calculated at each step of the recursion. Since the error falls rapidly at first as the filter is extended and then tends to level out, one can stop the process when the error fails to decrease significantly. This saves computation time not only during the filter design but also later when the filter is being applied to data.

Previous statements of the recursion algorithm (Levinson, 1947; Wiggins, 1965; Wiggins and Robinson, 1965; Robinson, 1963; Treitel and Robinson, 1966), although adequate for programming purposes and as mathematical proofs that the recursion gives correct answers, do not seem to me to offer the reader very much insight into why the algorithm works. In addition, some of the published discussions of the multichannel algorithm contain errors which are misleading for the inexperienced reader.

The purpose of this report is to present a simple, easily understood development and discussion of both the single-channel and the multichannel recursion algorithms. Program writeups and listings are appended.

In order for this work to be self-contained, we begin with derivation and discussion of the digital filter design equations the recursion is intended to solve.

#### Single-channel wave-shaping normal equations

Given a digital time series containing  $T+1$  points  $y_0, y_1, \dots, y_T$ , we require a filter of length  $L+1$  points which does the best job in the least mean-square-error sense of converting the input data series  $\vec{y}$  (we are entitled to regard the points  $y_0, y_1, \dots, y_T$  as the elements of a vector  $\vec{y}$ ) into some other desired data series  $\vec{d}$  with known elements  $d_0, d_1, \dots, d_{T+L+1}$ .



we wish to minimize

$$E = |\vec{e}|^2 = |\vec{z} - \vec{d}|^2 = \sum_{t=0}^{T+L+1} \left[ \sum_{j=0}^L y_{t-j} f_j - d_t \right]^2 \quad (4)$$

The minimum value of  $E$  is attained if all the partial derivatives of  $E$  with respect to the  $f_k$ ,  $k = 0, 1, \dots, L$ , are zero:

$$\frac{\partial E}{\partial f_k} = \frac{\partial}{\partial f_k} \sum_{t=0}^{T+L+1} \left[ \sum_{j=0}^L f_j y_{t-j} - d_t \right]^2 = 0$$

or

$$2 \sum_{t=0}^{T+L+1} y_{t-k} \left[ \sum_{j=0}^L f_j y_{t-j} - d_t \right] = 0 \quad k = 0, 1, \dots, L \quad (5)$$

[The quantity in brackets is the error vector, so (5) says that the error vector  $\vec{e}$  is orthogonal or normal to  $\vec{y}$ ; hence the name, normal equations.] We can write this set of equations:

$$\sum_{j=0}^L f_j \left[ \sum_{t=0}^{T+L+1} y_{t-k} y_{t-j} \right] = \sum_{t=0}^{T+L+1} y_{t-k} d_t \quad k=0, 1, \dots, L \quad (6)$$

In matrix form this is

$$Y^T Y \vec{f} = Y^T \vec{d} \quad (7)$$

that is

$$\begin{bmatrix}
 y_0 & y_1 & y_2 & \dots & y_T & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & y_0 & y_1 & \dots & y_T & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & \dots & & \dots & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & & & y_0 & y_1 & y_T \\
 & & & & & & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & & & & & & \begin{matrix} \diagdown \\ 0 \end{matrix}
 \end{bmatrix}
 \begin{bmatrix}
 y_0 & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 y_1 & y_0 & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 y_2 & y_1 & y_0 & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 \vdots & \dots & \dots & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 \dots & \dots & \dots & \dots & y_0 \\
 y_T & \dots & \dots & \dots & y_1 \\
 \begin{matrix} \diagdown \\ 0 \end{matrix} & \dots & \dots & \dots & \vdots \\
 & & y_T & \dots & \vdots \\
 & & & \dots & \vdots \\
 & & & & y_T
 \end{bmatrix}
 \quad (7)$$

$$= \begin{bmatrix}
 y_0 & y_1 & y_2 & \dots & y_T & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & y_0 & y_1 & \dots & y_T & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & \dots & & \dots & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & & & y_0 & y_1 & y_T \\
 & & & & & & \begin{matrix} \diagdown \\ 0 \end{matrix} \\
 & & & & & & & \begin{matrix} \diagdown \\ 0 \end{matrix}
 \end{bmatrix}
 \begin{bmatrix}
 d_0 \\
 d_1 \\
 d_2 \\
 \dots \\
 d_{T+L+1}
 \end{bmatrix}
 \quad (8)$$

We can simplify these equations by introducing the transient autocorrelation function

$$r_s = \sum_{t=0}^{M-s} y_t y_{t+s} \quad (9)$$

where  $M$  is the range of definition of  $y$ . We can define a vector  $\vec{r}$  whose elements are given by (9). With the change of variables  $q = t-k$ , the quantity in brackets on the left side of (6) is

$$\sum_{t=0}^{T+L+1} y_{t-k} y_{t-j} = \sum_{q=0}^{T+L+1-k} y_q y_{q+k-j} = r_{k-j} \quad (10)$$

The bottom limit on the second sum can be taken as zero, since  $y_q = 0$  for  $q < 0$ . Introducing the transient crosscorrelation function

$$\phi_{yd}(s) \equiv g_s \equiv \sum_{t=0}^{M-s} y_t^d y_{t+s} \quad (11)$$

the right side of (6) can be written

$$\sum_{t=0}^{T+L+1} y_{t-k}^d y_t = \sum_{q=0}^{T+L+1-q} y_q^d y_{t+k} = g_k \quad k = 0, 1, \dots, L \quad (12)$$

so that (6) is

$$\sum_{j=0}^L f_j r_{k-j} = g_k \quad k=0, 1, \dots, L \quad (13)$$

It is easy to show that these definitions are consistent with the matrix notation introduced above. We can define

$$R = Y^T Y = \begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_L \\ r_1 & r_0 & r_1 & \dots & r_{L-1} \\ r_2 & r_1 & r_0 & \dots & r_{L-2} \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} \quad (14)$$

so that (8) and (13) can be written

$$R\vec{f} = \vec{g} \quad (15)$$

Minimum error energy:

The minimum value attained by the error energy is obtained by substituting the solution of (15) into (4): if  $\vec{f}$  satisfies  $R\vec{f} = \vec{g}$ , that is,  $Y^T Y \vec{f} = Y^T \vec{d}$ , and if the output is  $\vec{z} = Y\vec{f}$ , then

$$\begin{aligned} E &= \vec{e}^T \vec{e} = (\vec{d} - \vec{z})^T (\vec{d} - \vec{z}) = \vec{d}^T \vec{d} - 2\vec{z}^T \vec{d} + \vec{z}^T \vec{z} \\ &= \vec{d}^T \vec{d} + \vec{z}^T \vec{z} - 2\vec{f}^T Y^T \vec{d} \\ &= \vec{d}^T \vec{d} + \vec{z}^T \vec{z} - 2\vec{f}^T \vec{g} \\ &= \vec{d}^T \vec{d} + \vec{z}^T \vec{z} - 2\vec{f}^T R\vec{f} \\ &= \vec{d}^T \vec{d} + \vec{z}^T \vec{z} - 2(Y\vec{f})^T Y\vec{f} \end{aligned} \quad (16)$$

so

$$E_{\min} = \vec{d}^T \vec{d} - \vec{z}^T \vec{z} \quad (17)$$

so that the energy in the error vector is simply the energy in the desired output minus the energy in the actual output.

The error energy can be calculated without actually constructing the output  $\vec{z}$ : in (16) we can convert  $\vec{z}^T \vec{z}$  into  $\vec{z}^T \vec{d}$  instead of the other way around, and since

$$\vec{z}^T \vec{d} = \vec{f}^T Y^T \vec{d} = \vec{f}^T \vec{g}$$

we have

$$E_{\min} = \vec{d}^T \vec{d} - \vec{f}^T \vec{g} \quad (18)$$

$E$ , being a sum of squares, cannot be negative; nor can it be greater than  $\vec{d}^T \vec{d}$  by (17), so if we divide through (18) by  $\vec{d}^T \vec{d}$  and define

$$P = E/\vec{d}^T \vec{d} = 1 - \frac{\vec{f}^T \vec{g}}{\vec{d}^T \vec{d}} \quad (19)$$

we are assured that

$$0 \leq P \leq 1$$

so that  $P$  can be used as a measure of the performance of a filter  $\vec{f}$  satisfying  $R\vec{f} = \vec{g}$ ; in particular, as the length of  $\vec{f}$  is extended, the performance can be evaluated at each step in the recursion (Treitel and Robinson, 1966).

The inverse filter:

Suppose the desired output  $d_t$  is specified to be unity at time  $t = s$  and zero everywhere else. If it were possible to achieve this desired output perfectly, we would have

$$z_t = \sum_j f_j y_{t-j} = f * y = (0, 0, \dots, 0, 1, 0, \dots, 0) \quad (20)$$

and such a filter is called an inverse filter by virtue of the resemblance between (20) and the definition of an inverse:

$$a \otimes b = I$$

where  $I$  is the identity with respect to the operator  $\otimes$ .

The right side of the normal equations (15) for the inverse filter is just the  $(s+1)$ 'st column of  $Y^T$ . The right side of equation (8) shows that for  $L < s < T+1$ ,  $g$  is an  $(L+1)$ -term segment of the input  $y_t$ , in reverse order:  $(y_s, y_{s-1}, \dots, y_{s-L})$ . For  $0 < s < L$  or  $T+1 < s < T+L+1$ ,  $g$  has fewer than  $L+1$  nonzero terms. We see that for  $s = 0$  and  $s = T+L+1$ ,  $\vec{g}$  has only

one nonzero element. The former case is the zero-delay inverse filter, which satisfies

$$\vec{R}\vec{f} = \begin{bmatrix} y_0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (21)$$

The value of  $y_0$  enters these equations only as a scale factor, which does not affect the shape of the filter or of its output.

The zero-lag inverse filter thus depends not at all on the actual waveform  $\vec{y}$  (except trivially as a scale factor), but only on its autocorrelation  $\vec{r}$ . Since the autocorrelation is symmetric --  $r_k = r_{-k}$  --  $\vec{f}$  does not depend on the phase spectrum of the input data. We might expect that the phase response of the filter  $\vec{f}$  depends only upon some intrinsic property of the autocorrelation, and it does: the phase spectrum of  $\vec{f}$  is such that the total energy in the waveform  $\vec{f}$  is jammed up as much toward the front of the waveform (toward  $f_0$ ) as is consistent with  $\vec{f}$  satisfying (21). This concept of minimum energy delay or minimum phase is the subject of a considerable literature (Robinson, 1954; Robinson and Treitel, 1965; Robinson, 1962).

The performance factor  $P$  of an inverse filter depends drastically on the delay  $s$  at which the desired output is to occur. (Treitel and Robinson, 1966, show examples for which the performance factor is 0.005 for  $s=0$ , but 0.860 for  $s = 15$ ). A recursion exists (Simpson et al., 1963) which shifts the desired output lag  $s$ . This can be used to search relatively inexpensively for the lag at which the performance factor can be maximized for given filter length.

The prediction filter:

Suppose we request a desired output at time  $t$  consisting of the input itself  $p$  time units ahead. This is an extrapolation problem of considerable practical interest, for example, in economics and weather forecasting.

The desired output is

$$d_t = y_{t+p}$$

Substituting this into (11) we have for the right-hand side vector of the normal equations

$$g_k = \sum_{s=0}^T y_s d_{s+k} \quad k=0, 1, \dots, L$$

$$= \sum_{s=0}^T y_s y_{s+p} = r_p$$

so (13) becomes

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_L \\ r_1 & r_0 & r_1 & \dots & r_{L-1} \\ r_2 & r_1 & r_0 & \dots & r_{L-2} \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_L \end{bmatrix} = \begin{bmatrix} r_p \\ r_{p+1} \\ r_{p+2} \\ \dots \\ r_{p+L} \end{bmatrix} \quad (22)$$

The presence of a segment of the  $\vec{f}$  vector on the right-hand side of (22) is interesting. A trivial prediction filter is that for  $p=0$ ; in this case we obviously have  $\vec{f} = (1, 0, \dots, 0)$ , i.e.,  $\vec{f}$  reproduces the input, as it was

told to do. The next simplest prediction filter attempts to predict only one unit ahead in time:

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_L \\ r_1 & r_0 & r_1 & \dots & r_{L-1} \\ r_2 & r_1 & r_0 & \dots & r_{L-2} \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_L \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \dots \\ r_{L+1} \end{bmatrix} \quad (23)$$

The desired output is  $d_t = y_{t+1}$ . The actual output is

$$z_t = \sum_{k=0}^L f_k y_{t-k} \quad (24)$$

and the error is

$$e_t = d_t - z_t = y_{t+1} - \sum_{k=0}^L f_k y_{t-k} \quad (25)$$

The form of the right-hand side of (25) suggests that we could define a prediction error filter to get  $e_t$  directly, as follows:

$$h_0 = 1; \quad h_j = -f_j, \quad j = 1, 2, \dots, L+1$$

The prediction error filter is thus one unit longer than the prediction filter, and its output is

$$e_t = \sum_{k=0}^{L+1} h_k y_{t-k} \quad (26)$$

where  $t+1$  in (25) has been replaced by  $t$  in (26).

We can inquire what normal equations the prediction error filter  $\vec{h}$  satisfies: we write

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_{L+1} \\ r_1 & r_0 & r_1 & \dots & r_L \\ r_2 & r_1 & r_0 & \dots & r_{L-1} \\ \dots & \dots & \dots & \dots & \dots \\ r_{L+1} & r_L & r_{L-1} & \dots & r_0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \dots \\ h_{L+1} \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \dots \\ c_{L+1} \end{bmatrix} \quad (27)$$

where the  $c_j$ ,  $j=0, L+1$ , are as yet unknown.

Let us use the notation

$$\vec{g}_1 = \begin{bmatrix} r_1 \\ r_1 \\ \dots \\ r_{L+1} \end{bmatrix} \quad \vec{c}_1 = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_{L+1} \end{bmatrix} \quad \vec{h}_1 = \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_{L+1} \end{bmatrix} \quad (28)$$

Then (15) is

$$R\vec{f} = \vec{g}$$

and (27) can be written

$$\left[ \begin{array}{c|c} r_0 & \vec{g}_1^T \\ \hline \vec{g} & R \end{array} \right] \begin{bmatrix} h_0 \\ \vec{h}_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ \vec{c}_1 \end{bmatrix} \quad (29)$$

But  $\vec{h}_1 = -\vec{f}$  by definition. So we have

$$\begin{bmatrix} r_0 & | & \vec{g}_1^T \\ \hline \vec{g}_1 & | & R \end{bmatrix} \begin{bmatrix} 1 \\ \hline -\vec{f} \end{bmatrix} = \begin{bmatrix} c_0 \\ \hline \vec{c}_1 \end{bmatrix} \quad (30)$$

Multiplying this out,

$$\begin{aligned} r_0 - \vec{g}_1^T \vec{f} &= c_0 \\ \vec{g}_1 - R\vec{f} &= \vec{c}_1 = 0 \end{aligned}$$

the last step following from (23). From the previous discussion on error energy, we know that  $\vec{g}_1^T \vec{f}$  is the energy in the predicted output (equation 18).

The zero-lag autocorrelation is the energy in the input, so we have

$$c_0 = r_0 - \vec{z}^T \vec{z} = \vec{y}^T \vec{y} - \vec{z}^T \vec{z} = E$$

where E is the unpredictable energy. Thus the prediction error filter satisfies

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_{L+1} \\ r_1 & r_0 & r_1 & \dots & r_L \\ r_2 & r_1 & r_0 & \dots & r_{L-1} \\ \dots & \dots & \dots & \dots & \dots \\ r_{L+1} & r_L & r_{L-1} & \dots & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ -f_0 \\ -f_1 \\ \dots \\ f_L \end{bmatrix} = \begin{bmatrix} E \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (31)$$

Except for the scale factor E, (31) is the same as the normal equations for the inverse filter (equation 21). An heuristic argument as to why this should be so was given by Claerbout (1963); a rigorous derivation was given by Robinson (1954).

The prediction error filter for spans greater than 1 is not so simple. For example, for  $p = 2$ , we have

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_{L+2} \\ r_1 & r_0 & r_1 & \dots & r_{L+1} \\ r_2 & r_1 & r_0 & \dots & r_L \\ \dots & \dots & \dots & \dots & \dots \\ r_{L+2} & r_{L+1} & r_L & \dots & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -f_0 \\ \dots \\ -f_L \end{bmatrix} = \begin{bmatrix} c \\ c_1 \\ c_2 \\ \dots \\ c_{L+2} \end{bmatrix} \quad (32)$$

Defining  $\vec{c}_2 = (c_2, c_3, \dots, c_{L+2})^T$ ,  $\vec{g}_2 = (r_2, r_3, \dots, r_{L+2})^T$ ,

(32) becomes

$$\begin{bmatrix} \frac{r_0}{\vec{g}_2} & \frac{r_1}{\vec{g}_1} & \frac{\vec{g}_2^T}{R} \\ \frac{r_1}{\vec{g}_2} & \frac{r_0}{\vec{g}_1} & \frac{\vec{g}_1^T}{R} \\ \frac{\vec{g}_2}{R} & \frac{\vec{g}_1}{R} & R \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -\vec{f} \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ \vec{c}_2 \end{bmatrix} \quad (33)$$

Multiplying this out, we have

$$r_0 - \vec{g}_2^T \vec{f} = c_0 \quad (34)$$

$$r_1 - \vec{g}_1^T \vec{f} = c_1 \quad (35)$$

$$\vec{g}_2 - R\vec{f} = \vec{c}_2 = 0 \quad (36)$$

and all this can be calculated from  $\vec{f}$  and  $\vec{r}$ . It is easier to construct the predicted trace and subtract it from the input with the appropriate shift. Alternatively, one can construct an approximation to the output of the prediction error filter for span two by filtering twice with a unit-span prediction error filter.

General characteristics of the wave-shaping filter:

Returning to equations (14) and (15), we notice that the filter does not actually depend on the waveform  $\vec{y}$  but only on its autocorrelation function and crosscorrelation with the desired output. This fact is of central importance in the analysis of stationary stochastic processes where the waveform is a collection of random variables, whose correlation properties nevertheless do not change with time; hence the correlation functions can be measured using one sample of the noise, and the resulting filter will be valid for other samples of the noise.

Notice also that the optimum filters of different length, say  $L$  and  $L+1$ , do not contain the same elements up through  $L$  terms; for example, the one-point filter is

$$f_0 = g_0/r_0$$

and the two-point filter is

$$\begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \frac{1}{r_0^2 - r_1^2} \begin{bmatrix} r_0 g_0 - r_1 g_1 \\ r_0 g_1 - r_1 g_0 \end{bmatrix}$$

The Levinson recursion allows us to construct the filter of length  $L+1$  from the filter of length  $L$ . Since we know the one-point filter  $f_0 = g_0/r_0$ , we can recursively construct a filter of any length. Notice that if  $g_1 = 0$  (zero delay inverse),  $f$  is minimum delay, since  $r_0$  is greater than  $r_1$ .

The single-channel recursion algorithm

The single-channel normal equations (15) were:

$$\begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_L \\ r_1 & r_0 & r_1 & \dots & r_{L-1} \\ r_2 & r_1 & r_0 & \dots & r_{L-2} \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_L \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \dots \\ g_L \end{bmatrix} \quad (15)$$

R is an L+1 by L+1 matrix; to store it requires (L+1)<sup>2</sup> words, and to find  $\vec{f} = R^{-1}\vec{g}$  by inverting R would require (L+1)<sup>3</sup> arithmetical operations. But R is a Toeplitz matrix, i.e., all the elements on a given diagonal are equal, so that there are really only L+1 different elements in R. The Levinson recursion cleverly capitalizes on this fact to calculate  $\vec{f}$  directly from the two vectors (r<sub>0</sub>, r<sub>1</sub>, ..., r<sub>L</sub>) and (g<sub>0</sub>, g<sub>1</sub>, ..., g<sub>L</sub>) without ever storing the whole matrix R. The saving in arithmetical operations comes from the fact that extending the one-point filter to length L+1 requires only a total number of arithmetical operations proportional to (L+1)<sup>2</sup>.

The operation of the algorithm is most easily understood from a concrete example. We will take L = 2, and illustrate the extension from length 3 to length 4. Using primes to denote the unknown elements of the new filter, we seek the solution  $\vec{f}'$  of:

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ f'_3 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad (37)$$

where we already know the solution  $\vec{f}$  of:

$$\begin{bmatrix} r_0 & r_1 & r_2 \\ r_1 & r_0 & r_1 \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} \quad (38)$$

We see the 3-by-3 R matrix of (38) in the upper left corner of the 4-by-4 matrix of (37), and the  $\vec{g}$  of (38) in the upper part of the right-hand side of (37). This suggests trying to express  $\vec{f}'$  in terms of  $\vec{f}$  and some other unknown vector:

$$\begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ f'_3 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ 0 \end{bmatrix} + \begin{bmatrix} b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{bmatrix} \gamma \quad (39)$$

The reason for splitting off the unknown scalar  $\gamma$  and making the indices on  $b'$  go backward will appear later.

We can also split  $\vec{g}$  into a part which depends on  $\gamma$  and a part which does not:

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ v' \end{bmatrix} \gamma \quad (40)$$

Now  $rf$ ,  $v'$ , and  $\gamma$  are unknown. " $rf$ " is a Fortran-like symbol for " $r$  dot  $f$ ", as will be seen later.

From (40), we have

$$v' \gamma + rf = g_3 \quad (41)$$

Substituting (39) and (40) into (37), we have

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \left( \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_0 \end{bmatrix} + \begin{bmatrix} b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{bmatrix} \gamma \right) = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ v' \end{bmatrix} \gamma \quad (42)$$

we can separate (42) into a set of equations which depend on  $\gamma$ :

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ v' \end{bmatrix} \quad (43)$$

and a part which is independent of  $\gamma$ :

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ 0 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} \quad (44)$$

The top three equations of (44) are just (38) again. The bottom equation is

$$rf = \sum_{k=0}^L r_{L-k+1} f_k \quad (45)$$

Using (41), we have

$$\gamma = (g_3 - rf)/v' \quad (46)$$

Now all we need to extend  $\vec{f}$  to  $\vec{f}'$  are  $v'$  and  $\vec{b}'$ . But the remarkable symmetry of  $R$  allows to turn equation (43) upside down and inside out, and identify it with (31), the equation defining an optimum unit-span prediction error filter -- provided we have  $b_0=1$ , which we will show we can always do. Then  $v'$  is just the error energy (the symbol  $v$  having been chosen to connote variance). The fact that (43) is backward with respect to (31) is unimportant in the single-channel situation. For the multichannel filters, we shall see that predicting backward in time is fundamentally different from predicting forward, the reason being that the crosscorrelation functions are not symmetric.

Thus far we have seen that in order to extend the wave-shaping filter  $\vec{f}$ , all we need is a method for extending the backward prediction error filter  $\vec{b}$ . We can anticipate that this will be a simpler task, since (31) is so much simpler than (8).

At each step of the recursion we extend  $\vec{b}$  to  $\vec{b}'$ , calculate  $v'$  from (18), and then extend  $\vec{f}$  to  $\vec{f}'$  using (46), (45), and (39). Thus we turn our attention to the problem of extending  $b$ .

Suppose we know  $\vec{b}$  and  $v$  at this step of the recursion:

$$\begin{bmatrix} r_0 & r_1 & r_2 \\ r_1 & r_0 & r_1 \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v \end{bmatrix} \quad (47)$$

Filling out  $\vec{b}$  with a zero to length 4, substituting it into (43), and using (47), we have:

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} e \\ 0 \\ 0 \\ v \end{bmatrix} \quad (48)$$

where  $e$  is yet unknown. We can rearrange (48) as follows, because of Toeplitz symmetry of  $R$ :

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ 0 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \\ e \end{bmatrix} \quad (49)$$

Now suppose we multiply both sides of (49) by a constant  $\alpha$  (as yet unknown), add the result to (48), and attempt to identify the result with (43):

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \left( \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ 0 \end{bmatrix} \alpha \right) = \begin{bmatrix} e \\ 0 \\ 0 \\ v \end{bmatrix} + \begin{bmatrix} v \\ 0 \\ 0 \\ e \end{bmatrix} \alpha \quad (50)$$

(43) was:

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_1 & r_0 & r_1 & r_2 \\ r_2 & r_1 & r_0 & r_1 \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ v' \end{bmatrix}$$

The bottom equation of the  $\alpha$ -dependent part of (50) determines  $e$ :

$$\sum_{k=0}^L r_{L-k+1} b_k = e \quad (51)$$

Comparison of the top element of the right-hand side of (50) with the top element of the right-hand side of (43) determines  $\alpha$ :

$$e + v\alpha = 0$$

or

$$\alpha = -e/v \quad (52)$$

Similar comparison of the bottom elements determines  $v'$ :

$$v' = v + e\alpha = v - e^2/v \quad (53)$$

and so finally:

$$b'_k = b_k + \alpha b_{L-k} \quad k = 0, 1, \dots, L \quad (54)$$

Equation (54) is rather remarkable. Notice that it implies that  $b_0$  is never changed during the recursion process, so  $\vec{b}$  always remains a unit-span prediction error operator, provided that we start with  $b_0 = 1$ .

We now have everything necessary to extend  $\vec{f}$ .

Summary of single-channel recursion:

Starting values:  $b_0 = 1$ ;  $f_0 = g_0/r_0$ ;  $v = r_0$ . Then for  $L = 0, 1, 2, \dots$ , compute:

$$1. \quad e = \sum_{k=0}^L r_{L-k+1} b_k$$

$$2. \quad a = -e/v$$

$$3. \quad b'_k = b_k + ab_{L-k} \quad k = 0, L+1$$

$$4. \quad v' = v + ae$$

$$5. \quad rf = \sum_{k=0}^L r_{L-k+1} f_k$$

$$6. \quad \gamma = (g_{L+1} - rf)/v'$$

$$7. \quad f'_k = f_k + \gamma b_{L-k+1} \quad k=0, L+1$$

Notice that in order to construct a zero-delay inverse filter recursively, we need carry out only the first four steps.

Normal equations for the multichannel wave-shaping filter

Suppose we have  $N$  input channels  $y_i(t)$ ,  $i = 1, N$ , and  $M$  output channels  $z_j(t)$ ,  $j = 1, M$ , related to the inputs by:

$$z_j(t) = \sum_{i=1}^N \sum_{s=0}^L f_{ij}(s) y_i(t-s) \quad j=1, M \quad (55)$$

so that each input contributes to each output via the  $N \cdot M$  filters  $f_{ij}(s)$ .

Suppose also that we have a given set of desired output functions  $d_j(t)$ ,  $j=1, M$ . Then we seek the set of filter coefficients which minimizes the set of  $M$  error functions

$$E_j = \sum_{t=0}^{T+L+1} [z_j(t) - d_j(t)]^2 \quad j = 1, M \quad (56)$$

[We see from (56) that since the  $M$  outputs are decoupled as far as the design criterion is concerned, we could just as well work with the single output channel case, and superimpose the results.]

Substituting (55) into (56), we have

$$E_j = \sum_{t=0}^{T+L+1} \left[ \sum_{i=1}^N \sum_{s=0}^L f_{ij}(s) y_i(t-s) - d_j(t) \right]^2 \quad j=1, M \quad (57)$$

The  $E_j$  are all minimized when the partial derivatives with respect to the filter coefficients all vanish:  $\partial E_m / \partial f_{km}(u) = 0$  for  $m = 1, M$ ;  $k = 1, N$ ;  $u = 0, L$ . Writing this out, we have:

$$\frac{\partial F_m}{\partial f_{km}(u)} = 2 \sum_{t=0}^{T+L+1} \left[ \sum_{i=1}^N \sum_{s=0}^L f_{im}(s) y_i(t-s) - d_m(t) \right] y_k(t-u) = 0$$

$$k = 1, N; m = 1, M; u = 0, L \quad (58)$$

which we can write:

$$\sum_{i=1}^N \sum_{s=0}^L f_{im}(s) \sum_{t=0}^{T+L+1} y_i(t-s) y_k(t-u) = \sum_{t=0}^{T+L+1} y_k(t-u) d_m(t)$$

$$k = 1, N; m = 1, M; u = 0, L \quad (59)$$

We can define correlation functions as in the single-channel case:

$$r_{hn}(q) = \sum_{t=0}^{T+L+1} y_h(t) y_n(t+q) \quad (60)$$

$$g_{hn}(q) = \sum_{t=0}^{T+L+1} y_h(t) d_n(t+q) \quad (61)$$

From the cross-symmetry of the crosscorrelation function, we have

$$r_{hn}(q) = r_{nh}(-q) \quad (62)$$

Thus we can write (59):

$$\sum_{i=1}^N \sum_{s=0}^L f_{im}(s) r_{ik}(s-u) = g_{km}(u) \quad \begin{matrix} k=1, N \\ m=1, M \\ u=0, L \end{matrix} \quad (63)$$

We can use (62) to make this set of equations resemble a matrix equation:

$$\sum_{i=1}^N \sum_{s=0}^L r_{ki}(u-s) f_{im}(s) = g_{km}(u) \quad \begin{matrix} k=1,N \\ m=1,M \\ u=0,L \end{matrix} \quad (64)$$

As an illustration, the set of equations for N=2, L=2, M=2, is:

$$\left[ \begin{array}{cc|cc|cc} r_{11}(0) & r_{12}(0) & r_{11}(-1) & r_{12}(-1) & r_{11}(-2) & r_{12}(-2) \\ r_{21}(0) & r_{22}(0) & r_{21}(-1) & r_{22}(-1) & r_{21}(-2) & r_{22}(-2) \\ \hline r_{11}(1) & r_{12}(1) & r_{11}(0) & r_{12}(0) & r_{11}(-1) & r_{12}(-1) \\ r_{21}(1) & r_{22}(1) & r_{21}(0) & r_{22}(0) & r_{21}(-1) & r_{22}(-1) \\ \hline r_{11}(2) & r_{12}(2) & r_{11}(1) & r_{12}(1) & r_{11}(0) & r_{12}(0) \\ r_{21}(2) & r_{22}(2) & r_{21}(1) & r_{22}(1) & r_{21}(0) & r_{22}(0) \end{array} \right] \left[ \begin{array}{c} f_{11}(0) \ f_{12}(0) \\ f_{21}(0) \ f_{22}(0) \\ \hline f_{11}(1) \ f_{12}(1) \\ f_{21}(1) \ f_{22}(1) \\ \hline f_{11}(2) \ f_{12}(2) \\ f_{21}(2) \ f_{22}(2) \end{array} \right] = \left[ \begin{array}{c} g_{11}(0) \ g_{12}(0) \\ g_{21}(0) \ g_{22}(0) \\ \hline g_{11}(1) \ g_{12}(1) \\ g_{21}(1) \ g_{22}(1) \\ \hline g_{11}(2) \ g_{12}(2) \\ g_{21}(2) \ g_{22}(2) \end{array} \right] \quad (65)$$

The dashed partition lines we have drawn in (65) suggest a matrix formulation:

define

$$r_s = \begin{bmatrix} r_{11}(s) & r_{12}(s) & \dots & r_{1N}(s) \\ r_{21}(s) & r_{22}(s) & \dots & r_{2N}(s) \\ \dots & \dots & \dots & \dots \\ r_{N1}(s) & r_{N2}(s) & \dots & r_{NN}(s) \end{bmatrix} \quad (66)$$

$$f_s = \begin{bmatrix} f_{11}(s) & f_{12}(s) & \dots & f_{1M}(s) \\ f_{21}(s) & f_{22}(s) & \dots & f_{2M}(s) \\ \dots & \dots & \dots & \dots \\ f_{N1}(s) & f_{N2}(s) & \dots & f_{NM}(s) \end{bmatrix} \quad (67)$$

$$g_s = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \dots & g_{1M}(s) \\ g_{21}(s) & g_{22}(s) & \dots & g_{2M}(s) \\ \dots & \dots & \dots & \dots \\ g_{N1}(s) & g_{N2}(s) & \dots & g_{NM}(s) \end{bmatrix} \quad (68)$$

We can now define supermatrices (matrices whose elements are themselves matrices):

$$R = \begin{bmatrix} r_0 & r_{-1} & r_{-2} & \dots & r_{-L} \\ r_1 & r_0 & r_{-1} & \dots & r_{-L+1} \\ r_2 & r_1 & r_0 & \dots & r_{-L+2} \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} = \begin{bmatrix} r_0 & r_1^T & r_2^T & \dots & r_L^T \\ r_1 & r_0 & r_1^T & \dots & r_{L-1}^T \\ r_2 & r_1 & r_0 & \dots & r_{L-2}^T \\ \dots & \dots & \dots & \dots & \dots \\ r_L & r_{L-1} & r_{L-2} & \dots & r_0 \end{bmatrix} \quad (69)$$

$$F = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_L \end{bmatrix} \quad G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \dots \\ g_L \end{bmatrix} \quad (70)$$

From (62) we see that R is symmetric and block-Toeplitz. The normal equations (64) become simply:

$$RF = G \quad (71)$$

where  $R$  is an  $L$ -by- $L$  matrix whose elements are each  $N$ -by- $N$  matrices;  
 $F$  and  $G$  are both column vectors each of whose elements is an  $N$ -by- $M$  matrix.

### Recursion for the Multichannel filter

The Levinson-Robinson recursion for the multichannel waveshaping filter goes as above, with the following exceptions:

1. Elements of the normal equations are matrices, not scalars.
2. Since crosscorrelations are not symmetric, superdiagonal elements of  $R$  must be written  $r_k^T$  or  $r_{-k}$ , rather than  $r_k$ .
3. Since matrices do not necessarily commute, we must be careful about the order of elements in a product.
4. For reciprocals we need to use inverses.
5. Most important, the backward multichannel prediction error operator is not simply the time reverse of the forward prediction error operator. We find that instead of one auxiliary sequence  $\vec{b}$ , we will need two,  $\vec{a}$  and  $\vec{b}$ , which correspond to the forward and backward prediction error operators. Instead of the single variance and error terms  $v$  and  $e$ , we will need a variance and an error term for both the forward and the backward prediction error operators.

As before, we illustrate the recursion for  $L = 2$ . We seek the solution  $\vec{f}'$  of:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ f'_3 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad (72)$$

where we know the solution  $\vec{f}$  of:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T \\ r_1 & r_0 & r_1^T \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} \quad (73)$$

and where now the r's, f's, and g's are given by (69) and (70)

As before, we express  $\vec{f}'$  in terms of  $\vec{f}$  and an unknown correction:

$$\begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ f'_3 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ 0 \end{bmatrix} + \begin{bmatrix} b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{bmatrix} \gamma \quad (74)$$

where now  $\gamma$  and the elements of  $\vec{f}$  and  $\vec{f}'$  are N-by-M matrices, and the elements of  $\vec{b}'$  are N-by-N matrices.

We split  $\vec{g}$  into a part which depends on  $\gamma$  and a part which does not:

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ vb' \end{bmatrix} \gamma \quad (75)$$

where  $rf$  is an N-by-M matrix and  $vb'$  is an N-by-N matrix. "vb" connotes "variance on the bottom"; we will later need a variance on the top, error on the top, and error on the bottom.

The bottom equation in (75) gives

$$g_3 = rf + vb'\gamma$$

so we have

$$\gamma = (vb')^{-1}(g_3 - rf) \tag{76}$$

We now substitute (74) and (75) into (72):

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \left\{ \begin{bmatrix} f_0 \\ f_1 \\ f_1 \\ 0 \end{bmatrix} + \begin{bmatrix} b_3' \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} \gamma \right\} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ vb' \end{bmatrix} \gamma \tag{77}$$

The  $\gamma$ -dependent part of (77) is

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b_3' \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ vb' \end{bmatrix} \tag{78}$$

and the part which is independent of  $\gamma$  is:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ 0 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ rf \end{bmatrix} \tag{79}$$

The top three equations of (79) are just (73) again. The bottom equation of (79) determines  $r_f$ :

$$r_f = \sum_{k=0}^L r_{L-k+1} f_k \quad (80)$$

Now all we need to calculate  $\vec{f}'$  from  $\vec{f}$  are  $vb'$  and  $\vec{b}'$ .

Since the elements of  $R$  now are matrices containing crosscorrelations, we cannot rearrange (75) in the same way we got (49) from (48). It turns out that to pull the same trick as was used in forming (50) and comparing it to (43), we must define a forward prediction error operator  $\vec{a}'$ , which satisfies:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} a_0' \\ a_1' \\ a_2' \\ a_3' \end{bmatrix} = \begin{bmatrix} vt' \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (81)$$

where  $vt' \neq vb'$ . (81) will play the same role equation (49) did in the single-channel case. Thus two auxiliary operators are required for the recursion procedure, whereas the single-channel case required only one.

Suppose, then, that at this stage  $\vec{a}'$ ,  $\vec{b}'$ ,  $vt'$ , and  $vb'$  are unknown, but that we know  $vt$  and  $vb$ , and  $\vec{a}$  and  $\vec{b}$  which satisfy:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T \\ r_1 & r_0 & r_1^T \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} vt \\ 0 \\ 0 \end{bmatrix} \quad (82)$$

$$\begin{bmatrix} r_0 & r_1^T & r_2^T \\ r_1 & r_0 & r_1^T \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ vb \end{bmatrix} \quad (83)$$

We now fill out  $\vec{a}$  with a zero matrix and substitute into (81); using (82)

we have:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} = \begin{bmatrix} vt \\ 0 \\ 0 \\ eb \end{bmatrix} \quad (84)$$

where  $eb$  is an unknown N-by-N matrix. Similarly extending  $\vec{b}$ , substituting into (78), and using (83), we have:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} et \\ 0 \\ 0 \\ vb \end{bmatrix} \quad (85)$$

where  $et$  is an unknown N-by-N matrix.

Now we postmultiply (84) by an unknown N-by-N matrix  $\alpha$ , postmultiply (85) by an unknown N-by-N matrix  $\beta$ , and add:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \left( \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} \beta \right) = \begin{bmatrix} vt \\ 0 \\ 0 \\ eb \end{bmatrix} \alpha + \begin{bmatrix} et \\ 0 \\ 0 \\ vb \end{bmatrix} \beta \quad (86)$$

The top equation of the  $\beta$ -dependent part of (86) determines  $et$ :

$$et = \sum_{k=0}^L r_{L-k+1}^T b_k \quad (87)$$

and the bottom equation of the  $\alpha$ -dependent part of (86) determines  $eb$ :

$$eb = \sum_{k=0}^L r_{L-k+1} a_k \quad (88)$$

Now let us postmultiply (78) by  $\beta$ , postmultiply (81) by  $\alpha$ , and add:

$$\begin{bmatrix} r_0 & r_1^T & r_2^T & r_3^T \\ r_1 & r_0 & r_1^T & r_2^T \\ r_2 & r_1 & r_0 & r_1^T \\ r_3 & r_2 & r_1 & r_0 \end{bmatrix} \left( \begin{bmatrix} a_0' \\ a_1' \\ a_2' \\ a_3' \end{bmatrix} \alpha + \begin{bmatrix} b_3' \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} \beta \right) = \begin{bmatrix} vt' \\ 0 \\ 0 \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} 0 \\ 0 \\ 0 \\ vb' \end{bmatrix} \beta \quad (89)$$

We now identify (89) with (86). To do this, it turns out to be necessary to select

$$(vt)\alpha + et = 0$$

or

$$\alpha = -(vt)^{-1} et \quad (90)$$

and

$$(vb)\beta + eb = 0$$

or

$$\beta = -(vb)^{-1} eb \quad (91)$$

Substituting (90) and (91) into (86), we have

$$R \left( \begin{bmatrix} a_0' \\ a_1' \\ a_2' \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} 0 \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} \beta \right) = \begin{bmatrix} -et \\ 0 \\ 0 \\ (eb)\alpha \end{bmatrix} + \begin{bmatrix} (et)\beta \\ 0 \\ 0 \\ -eb \end{bmatrix} \quad (92)$$

We can get rid of the  $-et$  and the  $-eb$  simply by adding (84) and (85) to (92):

$$R \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} \beta + \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} \alpha = \begin{bmatrix} vt + (et)\beta \\ 0 \\ 0 \\ vb + (eb)\alpha \end{bmatrix} \quad (93)$$

Now our identification is complete if we put:

$$vt' = vt + (et)\beta \quad (94)$$

$$vb' = vb + (eb)\alpha \quad (95)$$

$$a'_k = a_k + b_{L-k+1} \beta \quad k = 0, L \quad (96)$$

$$b'_k = b_k + a_{L-k+1} \alpha \quad k = 0, L \quad (97)$$

A time-saver:

The fact that the matrix  $et$  is always the transpose of the matrix  $eb$  was proved by J. P. Burg in 1962 (personal communication) and much more simply by D. W. McCowan (personal communication) in 1966 as follows:

Premultiplying (84) by  $(0, b_2^T, b_1^T, b_0^T)$ , we have

$$(0, b_2^T, b_1^T, b_0^T) R \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} = (0, b_2^T, b_1^T, b_0^T) \begin{bmatrix} vt \\ 0 \\ 0 \\ eb \end{bmatrix} = eb \quad (98)$$

the last equality following from the fact that we have  $b_0 = I$  throughout the recursion process. Similarly premultiplying (85) by  $(a_0^T, a_1^T, a_2^T, 0)$  we have

$$(a_0^T, a_1^T, a_2^T, 0) R \begin{bmatrix} 0 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = (a_0^T, a_1^T, a_2^T, 0) \begin{bmatrix} et \\ 0 \\ 0 \\ vb \end{bmatrix} = et \quad (99)$$

the last equality following from the fact that we have  $a_0 = I$  throughout the recursion process. Since  $R = R^T$ , it follows that the left side of (98) is the transpose of the left side of (99), and hence that

$$et = eb^T \quad (100)$$

Summary of the multichannel recursion:

Starting values:  $a_0 = b_0 = I$ ;  $vt = vb = r_0$ ,  $f_0 = r_0^{-1} g_0$ . Then for  $L = 0, 1, \dots$ , compute:

1.  $et = \sum_{k=0}^L r_{L-k+1} b_k$
2.  $eb = et^T$
3.  $\alpha = -(vt)^{-1} et$
4.  $\beta = -(vb)^{-1} eb$
5.  $a'_k = a_k + b_{L-k+1} \beta \quad k=0, L$
6.  $b'_k = b_k + a_{L-k+1} \alpha \quad k=0, L$

$$7. vt' = vt + (et)\beta$$

$$8. vb' = vb + (eb)\alpha$$

$$9. rf = \sum_{k=0}^L r_{L-k+1} f_k$$

$$10. \gamma = (vb')^{-1}(g_{L+1} - rf)$$

$$11. f'_k = f_k + b_{L-k+1}\gamma \quad k = 0, L$$

Some special cases of multichannel filters

Multichannel prediction filter: This predicts the input channels, making use of the crosscorrelation between channels. For examples, see Claerbout (1964).

We take  $M = N$ , and

$$d_j(t) = y_j(t + p)$$

where  $p$  is the prediction span. The normal equations are:

$$R \begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_L \end{bmatrix} = \begin{bmatrix} r_p \\ r_{p+1} \\ \dots \\ r_{p+L} \end{bmatrix}$$

Array sum prediction:

This predicts the array sum. For a single output,

$$d(t) = \frac{1}{N} \sum_{i=1}^N y_i(t + p)$$

where  $p$  is the prediction span. We can predict the array sum at several different spans simultaneously:

$$d_j(t) = \frac{1}{N} \sum_{i=1}^N y_i(t + p_j) \quad j = 1, M$$

where  $p_j, j=1, M$ , are a set of  $M$  different prediction spans.

The right-hand side contains matrices  $g_s, s = 0, L$ :

$$(g_s)_{ij} = \frac{1}{N} \sum_{k=1}^N r_{ik}(s + p_j)$$

Spatial interpolation: We take as input  $N-1$  of the  $N$  input channels, and as desired output the remaining input channel. The normal equation matrix is  $L \cdot (N-1)$ -by- $L \cdot (N-1)$ , and the right-hand side contains matrices  $g_s, s=0, L$ , where

$$(g_s)_i = r_{iN}(s)$$

The frequency spectrum of the interpolation error output, formed by subtracting the  $N$ 'th channel from the interpolation filter output, can be used to determine the spatial coherence of the data (Flinn and McCowan, 1967).

Equalization: The "equalization" filter simply attempts to convert each input trace to the array sum, and hence is identical to the  $N$ -output array sum predictor at zero prediction span.

The equalization error output is a measure of the variability of the waveform across the array (for an example, see Backus, 1966).

Signal-to-noise ratio enhancement: Suppose that the input channels consist of signal mixed with noise, and we wish the output channels to contain only

the signal. This situation can be handled within the framework we have outlined above.

The inputs are:

$$y_i(t) = n_i(t) + s_i(t) \quad i = 1, N \quad (101)$$

where  $n_i(t)$  represents the noise and  $s_i(t)$  the signal; we assume that the two have different correlation properties across the array. The desired outputs are:

$$d_j(t) = s_j(t) \quad j = 1, N \quad (102)$$

We may ask for fewer output channels in order to get a more effective estimate of the signal. For example, if the signal is a wave travelling across the array, we might ask only for the time-shifted array sum:

$$d(t) = \frac{1}{N} \sum_{i=1}^N s_i(t - c_i) \quad (103)$$

where  $c_i$  is the time delay appropriate to the  $i$ 'th channel.

The normal equations are set up and solved as before. Frequently we measure the noise correlation functions and use a theoretical model for the noise. In this case we have an adjustable parameter in (101), which becomes:

$$y_i(t) = n_i(t) + \lambda s_i(t) \quad i = 1, N \quad (101a)$$

where  $\lambda$  determines the signal-to-noise ratio. If the actual inputs have a different  $\lambda$  than that for which the filters were designed, the performance is degraded.

The optimum linear filter with constraints

The previous filters were designed using no criterion other than the efficiency of converting one waveform into another. In some cases, however, it is desirable to impose constraints on the filter.

As a simple example of the single-channel filter with a constraint, we consider the maximum output energy filter (Claerbout, 1963). The design criterion is simply that when the input is a certain waveform  $x(t)$ , the energy in the output should be as large as possible. Obviously some sort of constraint is needed to keep the filter coefficients finite. We can choose to require that the energy in the filter be unity:

$$\sum_{t=0}^L f^2(t) = 1 \quad (104)$$

The output energy is:

$$E = \sum_{t=0}^{T+L+1} z^2(t) = \sum_{t=0}^{T+L+1} \left[ \sum_{\tau=0}^L f(t-\tau) x(t) \right]^2 \quad (105)$$

Using a Lagrange multiplier, we seek to maximize

$$Q = E + \lambda \left[ 1 - \sum_{t=0}^L f^2(t) \right] \quad (106)$$

Using the vector and matrix notation set up earlier (equations 2 and 2a), we have:

$$\begin{aligned} Q &= \vec{z}^T \vec{z} + \lambda (1 - \vec{f}^T \vec{f}) \\ &= (\mathbf{Y}\vec{f})^T \mathbf{Y}\vec{f} + \lambda (1 - \vec{f}^T \vec{f}) \\ &= \vec{f}^T \mathbf{Y}^T \mathbf{Y} \vec{f} + \lambda (1 - \vec{f}^T \vec{f}) \end{aligned}$$

Using (14),

$$Q = \vec{f}^T R \vec{f} + \lambda [1 - \vec{f}^T \vec{f}] \quad (107)$$

Now  $\partial Q / \partial \lambda = 0$  gives us the constraint, and  $\partial Q / \partial \vec{f}_j = 0$  gives:

$$\frac{\partial Q}{\partial \vec{f}_j} = R \vec{f} - \lambda \vec{f} = 0$$

$$\text{or } R \vec{f} = \lambda \vec{f} \quad (108)$$

which is a simple eigenvalue equation. We see that  $\lambda$  is the largest eigenvalue of the  $R$  matrix, and  $\vec{f}$  is the corresponding eigenvector. If we had chosen to minimize the output energy when the input is  $x(t)$ , we would find the smallest eigenvalue of  $R$  and the corresponding eigenvector.

The actual output energy for this filter is:

$$E = \vec{z}^T \vec{z} = \vec{f}^T R \vec{f} = \lambda \vec{f}^T \vec{f} = \lambda$$

so that the eigenvalue is the output energy.

We could have arrived at equation (108) by arguing that maximizing  $\vec{z}^T \vec{z}$  subject to the constraint  $\vec{f}^T \vec{f} = 1$  is equivalent to maximizing the ratio

$$\lambda = \frac{\vec{z}^T \vec{z}}{\vec{f}^T \vec{f}} = \frac{\vec{f}^T R \vec{f}}{\vec{f}^T \vec{f}} \quad (109)$$

In a slightly more complicated version of this filter, we not only ask that the output be as large as possible (in the energy sense) when  $x(t)$  is the input, but also that the output be as small as possible when some other waveform  $w(t)$  is input. This is equivalent to maximizing the ratio

$$\lambda = \frac{\vec{z}^T \vec{z}}{\vec{f}^T W^T W \vec{f}} \quad (110)$$

where  $W$  is a matrix formed from  $w(t)$  in the same way that  $Y$  was formed from  $y(t)$ --see equation (2a).

Writing the autocorrelation matrix of  $W$  as  $P = W^T W$ , (110) becomes

$$\vec{f}^T (R\vec{f} + \lambda P\vec{f}) = 0$$

$$\text{or} \quad R\vec{f} + \lambda P\vec{f} = 0 \quad (111)$$

which is a generalized eigenvalue problem. (111) can always be solved, since correlation matrices are non-negative definite. If  $L$  is not too large, we can convert (111) into (108) by premultiplying by  $P^{-1}$ :

$$(P^{-1}R)\vec{f} + \lambda\vec{f} = 0 \quad (112)$$

The maximum-likelihood filter:

The maximum-likelihood or minimum-variance filter (the two terms are synonymous if the input is a Gaussian random process) was designed by Levin (Kelly and Levin, 1965). It is an example of a multichannel filter with a constraint.

We assume a single output channel, and ask that the energy in it be as small as possible, i.e., we wish to minimize

$$E = \vec{z}^T \vec{z} = \sum_{t=0}^{T+L+1} z^2(t) \quad (113)$$

We obviously need a constraint to keep the filter coefficients from turning out to be zero. Levin suggested a linear constraint based on prior knowledge of the nature of the input. In Levin's case the input channels consisted of signal and noise, but the signal was known to occur simultaneously on all the

channels with the same waveform and amplitude:

$$y_i(t) = n_i(t) + s(t) \quad i = 1, N \quad (114)$$

Levin's constraint was that the sum (across channels) of the filter coefficients should be a single spike at  $t = 0$ :

$$\sum_{i=1}^N f_i(t) \cong h(t) = (1, 0, 0, \dots, 0) \quad (115)$$

Using this constraint, the output is:

$$\begin{aligned} z(t) &= \sum_{i=1}^N \sum_{\tau=0}^L u_i(t-\tau) f_i(\tau) \\ &= \sum_{i=1}^N \sum_{\tau=0}^L n_i(t-\tau) f_i(\tau) + \sum_{i=1}^N \sum_{\tau=0}^L s(t-\tau) f_i(\tau) \end{aligned}$$

or

$$z(t) = \sum_{i=1}^N \sum_{\tau=0}^L n_i(t-\tau) f_i(\tau) + N s(t) \quad (116)$$

Thus the effect of minimizing the power in the output by varying the filter coefficients, under the constraint (115), is to reduce the first term on the right side of (116) relative to the second term (since the second term does not depend on the filter coefficients at all)--i.e., the effect is to increase the signal-to-noise ratio. Kelly and Levin showed that the output signal is unbiased, i.e., that the signal comes through undistorted. The filter can, of course, be designed so that the output signal arrives with any desired delay, i.e., the right side of (115) might be

$$h(t) = 0, 0, \dots, 0, 1, 0, \dots$$

Actually, any waveform  $h(t)$  could be used as a constraint, e.g., the impulse response of some desirable bandpass prefilter could be used as  $h(t)$ , and this would save doing the prefiltering. It will be apparent later that there are compensating advantages in using an  $h(t)$  which has a single nonzero element.

The normal equations for the maximum-likelihood filter are considerably more complicated than in the case of the wave-shaping filter. Using Lagrange multipliers, the quantity we minimize is:

$$Q = \sum_{t=0}^{T+L+1} \left[ \sum_{i=1}^N \sum_{s=0}^L y_1(t-s) f_1(s) \right]^2 - 2 \sum_{v=0}^L \lambda_v \left[ \sum_{i=1}^N f_1(v) - h(v) \right] \quad (117)$$

where we have split off a factor of (-2) from the Lagrange multipliers for later convenience.

Writing out  $\partial Q / \partial f_k(u) = 0$ ,  $k=1, N$ ;  $u=0, L$ , gives:

$$\sum_{t=0}^{T+L+1} \sum_{i=1}^N \sum_{s=0}^L y_1(t-s) f_1(s) y_k(t-u) = \lambda_u \quad (118)$$

$\partial Q / \partial \lambda_u = 0$ ,  $u=0, L$ , gives:

$$\sum_{i=1}^N f_1(u) = h(u) \quad (119)$$

Using (60) and (62), we can write (118) as:

$$\sum_{i=1}^N \sum_{s=0}^L r_{ki}(s-u) f_i(s) = \lambda_u \quad u = 0, L \quad (120)$$

We define R and F as in (69) and (70), as well as:

$$\Lambda = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \dots \\ \lambda_L \end{bmatrix} \quad H = \begin{bmatrix} h(0) \\ h(1) \\ \dots \\ h(L) \end{bmatrix} \quad (121)$$

and a summing matrix

$$S^T = \begin{matrix} \leftarrow N \rightarrow & \leftarrow N \rightarrow & \dots & \leftarrow N \rightarrow \\ \left[ \begin{array}{cccccccccccccccc} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & 1 & \dots & 0 \end{array} \right] \begin{matrix} \uparrow \\ L \\ \downarrow \end{matrix} \end{matrix} \quad (122)$$

an L-by-N·L matrix, each row of which contains N adjacent ones distributed as shown. S can be thought of as a stretched unit matrix. Using (122), we can write (119) as:

$$S^T F = H \quad (123)$$

and we can write (120) as:

$$RF = SA \quad (124)$$

We need to solve (123) and (124) for F. Observe first that

$$S^T S = N I \quad (125)$$

where I is the identity matrix. Starting with (123), we have:

$$S^T F = H$$

$$S^T R^{-1} R F = H$$

$$S^T R^{-1} S S^T R F = N H$$

$$S^T R F = N (S^T R^{-1} S)^{-1} H$$

$$S^T R F = S^T S (S^T R^{-1} S)^{-1} H$$

which is satisfied if

$$F = R^{-1} S (S^T R^{-1} S)^{-1} H \quad (126)$$

This can be abbreviated

$$F = P (S^T P)^{-1} H \quad (127)$$

where  $P = R^{-1} S \quad (128)$

Notice that P is a sort of optimum multichannel filter in its own right, since it is a solution of the normal equations  $R P = S$ . Because of this we can compute P using the multichannel recursion, and form F by selecting the j'th column of  $P (S^T P)^{-1}$ , where the single nonzero element of H is the j'th element.

Using (126), the output energy is:

$$E = \mathbf{z}^T \mathbf{z} = (\mathbf{YF})^T \mathbf{YF} = \mathbf{NH}^T (\mathbf{S}^T \mathbf{R}^{-1} \mathbf{S})^{-1} \mathbf{H} \quad (129)$$

When  $\mathbf{H}$  contains only one nonzero element (say the  $j$ 'th), the output error is thus  $N$  times the  $j$ 'th diagonal element of  $(\mathbf{S}^T \mathbf{R}^{-1} \mathbf{S})^{-1}$ , so we can compute the filter performance without actually calculating the output.

REFERENCES

- Backus, M.M., 1966, Teleseismic signal extraction: Proc. Roy. Soc., Vol. 29, p. 343-367.
- Claerbout, J.F., 1963, Digital filters and applications to seismic detection and discrimination: M.Sc. Thesis, Massachusetts Institute of Technology, Cambridge, Mass.
- Claerbout, J.F., 1964, Detection of P waves from weak sources at great distances: Geophysics, Vol. 29, p. 197-210.
- Flinn, E.A., and D.W. McCowan, 1967, Optimum multichannel filter design: Theory, practice, and examples: Seismic Data Laboratory Report (number not yet assigned), Seismic Data Laboratory, Alexandria, Virginia.
- Kelly, E.J., and M.J. Levin, 1964, (for specific reference, see references cited by Green, Capon, etc., in the December, 1965, Proc. IEEE).
- Levinson, N., 1947, An heuristic exposition of Wiener's mathematical theory of prediction and filtering: J. Math. Phys., Vol. 26, p. 110-119.
- McCowan, D.W., 1968, Design and evaluation of certain multichannel filters: Seismic Data Laboratory Report 209, Seismic Data Laboratory, Alexandria, Virginia.
- Robinson, E.A., 1954, Predictive decomposition of stationary time series, with application to seismic exploration: Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Mass. (reprinted in Geophysics, Vol. 32, No. 3, pp. 418-484, 1967)
- Robinson, E.A., and S. Treitel, 1965, Dispersive digital filters: Reviews of Geophysics, Vol. 29, p. 395-404.

- Simpson, S.M., E.A. Robinson, R.A. Wiggins, and C.L. Wunsch, 1963, Studies in optimal filtering of single and multiple stochastic processes: Scientific Rept. No. 9, Contract AF 19(604)7378. Defense Documentation Center accession number AD 428-509.
- Treitel, S., and E.A. Robinson, 1966, The design of high-resolution digital filters: *IEEE Trans. Geoscience Electronics*, Vol. GE-4, p. 25-38.
- Wiggins, R.A., 1965, On factoring the correlations of discrete multivariate stochastic processes: Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Mass.
- Wiggins, R.A., and E.A. Robinson, 1965, Recursive solution to the multi-channel filtering problem: *J. Geophys. Res.*, Vol. 70, p. 1885-1891.

**DOCUMENT CONTROL DATA - R&D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

1. ORIGINATING ACTIVITY (Corporate author) TELEDYNE INDUSTRIES, INC. ALEXANDRIA, VIRGINIA		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE A THEORETICAL DISCUSSION OF OPTIMUM MULTICHANNEL FILTER DESIGN			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Scientific			
5. AUTHOR(S) (Last name, first name, initial) Flinn, E. A.			
6. REPORT DATE 10 December 1968	7a. TOTAL NO. OF PAGES 48	7b. NO. OF REFS 13	
8a. CONTRACT OR GRANT NO. F 33657-68-C-0945	8b. ORIGINATOR'S REPORT NUMBER(S) 227		
8c. PROJECT NO. VELA T/6702			
8d. ARPA Order No. 624	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned to the report)		
8e. ARPA Program Code No. 8F10			
10. AVAILABILITY/LIMITATION NOTICES This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Chief, AFTAC.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY ADVANCED RESEARCH PROJECTS AGENCY NUCLEAR TEST DETECTION OFFICE WASHINGTON, D. C.	
13. ABSTRACT The design equations for both single-channel and multi-channel optimum least-squares ("Wiener") filters are derived and discussed. Specific examples of such filters are presented; for example, inverse filters, signal/noise ratio enhancement filters, prediction filters, and maximum-likelihood filters. The single-channel and multichannel Levinson recursion algorithms for solving the design equations are discussed.			
14. KEY WORDS Multichannel filters			