EXTENSION OF PENCIL-OF-FUNCTIONS METHOD TO REVERSE-TIME PROCESS---ETC(U)

AUG 80  V K JAIN, T K SARKAR, D D WEINER

UNCLASSIFIED

TR-80-4
EXTENSION OF PENCIL-OF-FUNCTIONS

METHOD TO REVERSE-TIME

PROCESSING WITH FIRST-ORDER DIGITAL FILTERS

BY

Vijay K. Jain
Tapan K. Sarkar
Donald D. Weiner

1. Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620

2. Department of Electrical Engineering
Syracuse University
Syracuse, New York 13210

3. Department of Electrical Engineering
Rochester Institute of Technology
Rochester, New York 14623

DISTRIBUTION STATEMENT A
Approved for public release. Distribution unlimited.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
In this presentation, the data signal is processed in reverse-time by a cascade of first order digital filters to yield a family of information signals. The Gram matrix of these information signals is shown to contain the essential information on the poles of the signal. The entire procedure of the application of pencil-of-function method is thus noniterative. Examples presented demonstrate (i) noiseworthiness in the representation problem when data are corrupted by noise, and (ii) the effectiveness of the method in the approximation problem.
Comparison of the method with the maximum entropy method (or linear predictor) and the Prony method is also included.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>First-Order Filter Based Pencil-of-Functions</td>
<td>3</td>
</tr>
<tr>
<td>II</td>
<td>Method for Modeling Impulse Responses</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>Parallelism and Orthogonality of Information</td>
<td>9</td>
</tr>
<tr>
<td>IV</td>
<td>Program Description</td>
<td>16</td>
</tr>
<tr>
<td>V</td>
<td>Application Examples</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>Appendix A -- Details of Program</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>POF-FILTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Appendix B -- Modeling of a Noisy Test Signal</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Appendix C -- Modeling of a Scatterer Response</td>
<td>88</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Generation of information signals by use of first-order filters</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Magnitude characteristic of ( \mu_1(z) = \frac{1-q}{1-qz^{-1}} )</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Normalized cutoff frequency vs. parameter ( q )</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Magnitude vs Freq. of connection filters</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Idealized connection filters and BP constituents</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Comparison of true and model impulse responses</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Comparison of true and model impulse responses</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Comparison of true and model impulse responses</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>Comparison of true and model impulse responses</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>Comparison of true and model impulse responses</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>Comparison of true and model impulse responses</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Comparison of measured data (of the response of a conducting pipe) and model impulse responses</td>
<td>32</td>
</tr>
<tr>
<td>13</td>
<td>Comparison of the derivative of measured data and the corresponding model impulse response</td>
<td>33</td>
</tr>
<tr>
<td>B1</td>
<td>True impulse response of a third order transfer function</td>
<td>80</td>
</tr>
<tr>
<td>B2</td>
<td>A simulated noisy signal under test</td>
<td>81</td>
</tr>
</tbody>
</table>
EXTENSION OF PENCIL OF FUNCTIONS METHOD TO REVERSE-TIME PROCESSING WITH FIRST-ORDER DIGITAL FILTERS

I. INTRODUCTION

Signal representation and approximation is basic to (a) time-domain extraction of singularities of a scatterer's field pattern. It is also useful in (b) bandwidth compression of signals, and (c) time-domain measurement and testing of networks/channels. This report discusses a unified approach to representing or approximating a given empirical signal by sum of exponentials, i.e., for finding the right hand side of

\[ h_d(t) = \sum_{i=1}^{n} \lambda_i t^i \quad \leftrightarrow \quad H(s) = \sum_{i=1}^{n} \frac{W_i}{s - \lambda_i} \]

\[ = \frac{b_0 s^{n-1} + b_{n-2} s^{n-2} + \ldots + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_0} \]

or, equivalently, the right hand side of the sampled version

\[ h_d(k) = h(k) = \sum_{i=1}^{n} R_i (z_i)^k \quad \leftrightarrow \quad H(z) = \sum_{i=1}^{n} \frac{R_i}{1 - z_i z^{-1}} \]

\[ = \frac{b_0 + b_1 z^{-1} + \ldots + b_{n-1} z^{-n+1}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} \]

The poles \( \lambda_i \) (or \( \gamma_i \) in z-domain) are either real, or they occur in complex conjugate pairs.

In the method described here, the data signal is processed in reverse-time by a cascade of first order digital filters — each \( \mu(z) = 1/(1-qz^{-1}) \), to yield a family of information signals. The Gram matrix \( G \) of these information signals is shown to contain the essential information on the denominator parameters of \( H(z) \). Specifically, it is shown that \( A(z) \) is determined as
\[ A(z) = z^{-n} \left\{ \sum_{i=1}^{n+1} D_i (z-1)^{n+1-i} \right\}/\sqrt{D_1} \]

where \( D_i \) are the diagonal cofactors of the matrix \( G \). The numerator parameters are then determined using a least-squares fit, i.e., \( b = -P^{-1}v \), where \( P \) and \( v \) are defined in the paper.

The entire procedure is thus noniterative and computationally efficient. It is a further generalization of the method developed in [4]. Examples presented demonstrate (i) noiseworthiness in the representation problem when data are corrupted by noise and (ii) the effectiveness of the method in the approximation problem. Comparison of the method with the maximum entropy method (or linear predictor) and the Prony method is also included in the report.

The structure of the report is as follows. The fundamental results relating to the signal representation/approximation problem, through the use of reverse-time processing by first-order digital filters, are obtained in Chapter II. The important question of parallelism and orthogonality of the information signals is explored in Section III. Chapter IV gives the input description and user instructions for the program POF-FILTER which implements the method of Chapter II. Application examples as well as comparison with other methods, are given in Section V. Appendices A gives listing and description of program POF-FILTER and its routines. Appendices B and C contain the computer outputs relating to Example 1 and Example 2, respectively, of Section V.
SECTION II

FIRST-ORDER FILTER BASED PENCIL-OF-FUNCTIONS

METHOD FOR MODELING IMPULSE RESPONSES

We shall be interested in modeling the impulse response \([1]-[3]\)

\[
y(k) = \sum_{i=1}^{n} R_{z}^{i} (z_{2})^{k} \leftrightarrow \frac{B(z)}{A(z)} = \frac{b_{1} z^{-1} + \ldots + b_{n} z^{-n}}{1 + a_{1} z^{-1} + \ldots + a_{n} z^{-n}}
\]  

(1)

from its numerical data. Suppose a suitable \(K\) has been selected such that \(y(k) = 0\) for \(k > K\) (so that use of the upper limit \(K\) instead of \(\infty\) on summations may be permitted). We define the reverse-time first-order filtered signals as

\[
y_{1}(k) = y(k)
\]

\[
y_{2}(k) = q y_{2}(k+1) + y_{1}(k)
\]

\[\vdots\]

\[
y_{N}(k) = q y_{N}(k+1) + y_{n}(k)
\]

where \(N = n + 1\), and \(y_{i}(K+1) = 0\) for \(i = 1, 2, \ldots, N\). Further, \(0 < q < 1\).

This family of signals, which we shall call information signals, exhibits the interesting property stated below.

Lemma 1

\[
y_{i+1}(k) = \sum_{k=1}^{i} \frac{R_{z}^{i}}{(1-q) z_{2}^{i}} (z_{2})^{k}
\]  

(3)

Proof: We prove this by induction. For \(i = 0\) the statement is trivially true since it is identical to (1) for this case. Assuming it to be true for \(i = 1\), let us proceed to prove it is true for \(i\).

From (2)

\[
y_{i+1}(k) = q y_{i+1}(k+1) + y_{i}(k)
\]  

(4)
which is readily shown to be equivalent to

\[ y_{i+1}(k) = \sum_{\nu=k}^{\infty} q^{\nu-k} y_i(\nu) \]

\[ = \sum_{\ell=1}^{n} \frac{R_\ell}{(1-q z_\ell)^{i-1}} \sum_{\nu=k}^{\infty} q^{\nu-k} (z_\ell)^{\nu} \]

(from induction hypothesis)

\[ = \sum_{\ell=1}^{n} \frac{R_\ell}{(1-q z_\ell)^{i-1}} \sum_{\nu=k}^{\infty} q^{\nu-k} (z_\ell)^{\nu-k} \]

The result of equation (3) follows immediately by observing that

\[ \sum_{\nu=k}^{\infty} q^{\nu-k} (z_\ell)^{\nu-k} = \frac{1}{(1-q z_\ell)} \]

This Lemma leads us to the crucial observation stated next.

**Lemma 2**

The set

\[ (q z_m^{-1}) y_2 + y_1, (q z_m^{-1}) y_3 + y_2, \ldots, (q z_m^{-1}) y_n + y_n \]

is linearly dependent for \( m=1,2,\ldots,n \) where \( z_m \) are the poles of the right hand side of (1).

Note: We have used the notation \( y_i \) to denote the sequence \( \{y_i(k)\} \), \( k=0,1,2,\ldots \).

Proof: In view of (3) we find

\[ (q z_m^{-1}) y_{i+1} + y_i = \sum_{\ell=1}^{n} \frac{q(z_m - z_\ell) R_\ell}{(1-q z_\ell)^i} (z_\ell)^k \]

for \( i=1,2,\ldots,n \). Clearly, the sequences \( (q z_m^{-1}) y_{i+1} + y_i \), \( i=1,\ldots,n \) each contain only \( n-1 \) modes \( (z_\ell)^k \), \( \ell \neq m \) hence are linearly dependent.

We can now apply the pencil-of-functions theorem of reference [4] to obtain the central theoretical result of this section.
We will call \( z_k \) (see (1)) the poles of the impulse response, \( R_z \) the corresponding residues, and \( p_\lambda = \{ (z_k)^k \} \) the associated modes. Note that the poles occur in conjugate pairs whenever complex, as do the residues, since \( y \) is real.

Define the \( N \times N \) dimensional Gram matrix (recall, \( N=n+1 \)) [6]

\[
F = \begin{bmatrix}
<y_1,y_1> & \cdots & <y_1,y_N> \\
\vdots & \ddots & \vdots \\
<y_N,y_1> & \cdots & <y_N,y_N>
\end{bmatrix}, \quad <y_i,y_j> = \sum_{k=1}^{K} y_i(k)y_j(k) \quad (7a)
\]

or, equivalently,

\[
F = \sum_{k=1}^{K} f(k)f^T(k) \quad (7b)
\]

where \( f^T(k) = [y_1(k) \ y_2(k) \ \ldots \ y_N(k)] \).

**Theorem 1**

The poles of the impulse response \( y(k) \) must satisfy the equation

\[
\sum_{i=1}^{N} \sqrt[D_i]{(qz-1)^{N-1}} = 0 \quad (8)
\]

where \( D_i \) are the diagonal cofactors of the Gram matrix \( F \) (defined in (7)).

Proof: The theorem follows immediately upon application of the pencil of functions theorem (reference [4]) to the set (5).

Note that the denominator of transform of the impulse response

\[
A(z) = D_1^{-1/2}(qz)^{-n} \sum_{i=1}^{N} \sqrt[D_i]{(qz-1)^{N-1}} \quad (9)
\]

This follows from (8) by dividing through by \( z^n \) and by normalizing the coefficients so that the leading coefficient becomes unity.
Determination of the denominator polynomial $A(z)$ completes the first of the two steps in the decoupled pencil-of-functions method. The next step, finding the numerator coefficients in $B(z)$, or equivalently finding the residues $R_i$, can be accomplished in two alternative ways.

The first method consists in solving for the residues from the equation

$$
\begin{array}{cccc}
\frac{1}{1-qz_1} & \frac{1}{1-qz_2} & \cdots & \frac{1}{1-qz_n} \\
\frac{1}{(1-qz_1)^2} & \frac{1}{(1-qz_2)^2} & \cdots & \frac{1}{(1-qz_n)^2} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{(1-qz_1)^N} & \frac{1}{(1-qz_2)^N} & \cdots & \frac{1}{(1-qz_n)^N}
\end{array}
\begin{array}{l}
R_1 \\
R_2 \\
\vdots \\
R_n
\end{array}
= 
\begin{array}{l}
y_2(0) \\
y_3(0) \\
\vdots \\
y_N(0)
\end{array}
$$

This equation follows from (3) upon setting $k=0$ and letting $i$ range from 1 to $n$. Clearly, the use of this equation requires that the poles $z$ be determined from the denominator $A(z)$ by use of a root-finding routine. This requirement is of no consequence if the final answers are needed in the $s$-domain, for conversion to $s$-domain involves finding the roots of $A(z)$ anyway.

The alternative approach is to find the optimum least-squares numerator coefficients (given the denominator of (9)) through the equation

1 The equation corresponding to $i=0$ is ignored in formulating (10) because of the relatively poor signal/noise statistics of $y_1(0)$ (compared to, say, $y_2(0)$) when the impulse response data contain additive wideband noise. This statement assumes that the bandwidth of $y(k)$ is much smaller than that of the additive noise.
where \( w_i \) denotes the impulse response of \( z^{-i}/A(z) \). Note that \( w_i(k) = w(k-i) \) where \( w(k) \) is the impulse response (i.e., inverse z-transform) of \( 1/A(z) \).

All inner products are summed from \( k=0 \) to \( K \).

**Discussion**

Equations (9) and (11) have been implemented in a computer program "POF-FILTER" written in FORTRAN IV. The program is presented in Section III.

The idea of reverse-time integration was proposed by Carr in [7] and Jain in [8]. Here, we have generalized the concept of reverse-time processing to the case of first-order filter processing. Note that the first order filter \( 1/(1-qz^{-1}) \), used above, encompasses integration; just let \( q=1 \).

It should be borne in mind that the approach developed above is applicable to impulse responses only. With some effort, it may be modified for use to step responses and square-pulse responses. For more general inputs, however, one must use the coupled approach discussed in [9] (which of course involves greatly increased computations, e.g., the Gram matrix involved is of an order twice as high as in the decoupled procedure).

The decoupled approach can be used only if the data is of sufficient length \( K \) such that \( y(k>K)=0 \) for all practical purposes.
The reverse-time processing of \( y(k) \) through the first order filters can be interpreted as forward-time processing of the signal \( h(k) = y(K-k) \) through the same filters.

The use of square-roots of cofactors of the Gram matrix is analogous to the use of square-root factorization [10] of the Gram matrix. Attendant advantages are therefore expected to be realized. A more detailed analysis of this connection will be discussed elsewhere.

The transfer function of the first order filter used in equation (2) is 
\[
\mu(z) = \frac{1}{1-az^{-1}}.
\]
Instead, we could use filters with transfer function 
\[
\mu_1(z) = \frac{1-q}{1-qz^{-1}};
\]
these filters have a d.c. gain equal to unity and the ratio of the output power to the input power is a direct measure of the extent of the rejection of higher frequencies. Equation (9) remains valid even when these unity d.c. gain filters are used.

Note that the first-order filtering (in reverse time) is achieved in (2) recursively, without the need to carry out discrete convolution.
SECTION III
PARALLELISM AND ORTHOGONALITY OF INFORMATION SIGNALS

Here we consider the important matter of parallelism and orthogonality of the information signals. In the last section we processed the signal \( h(k) = y(K-k) \) through first order filters \( \mu_1(z) = (1-q)/(1-qz^{-1}) \). This is depicted in Fig. 1. Note that forward time processing of \( h(k) \) is equivalent to reverse-time processing of \( y(k) \). Because of familiarity with forward-time processing we will carry out the discussion below in terms of the signal \( h(k) \). Also note that \( h_i(k) = y_i(K-k) \), \( i = 1,2,\ldots,N \). Finally, we remark that the Gram matrix and the properties of parallelism/orthogonality of the two families of signals \( h_1, h_2, \ldots, h_N \) and \( y_1, y_2, \ldots, y_N \) are identical. Indeed, \( \langle h_i, h_i \rangle = \langle y_i, y_i \rangle \) and \( \langle h_i, h_j \rangle = \langle y_i, y_j \rangle \) for \( i,j = 1,2,\ldots,N \).

\[
\begin{align*}
\text{Fig. 1. Generation of information signals by use of first-order filters.}
\end{align*}
\]

The magnitude (vs. frequency) characteristic of the first order filter \( \mu_1(z) \) is shown in Fig. 2. The cutoff frequency (-3dB point) \( \omega_c \) is related to the parameter \( q \) as [5]

\[
\omega_c = \frac{1}{\Delta} \ln\left(\frac{1}{q}\right)
\]

(12)

where \( \Delta \) is the sampling interval. Defining \( \Omega \) as the normalized frequency (i.e., the ratio \( \omega/\omega_s \) where \( \omega_s \) is the sampling frequency in radian/s) we can express the normalized cutoff frequency \( \Omega_c \) as

\[
\Omega_c = \frac{1}{2\pi} \ln(1/q)
\]

(13)
Fig. 2. Magnitude characteristic of $\mu_1(z) = \frac{1-q}{1-qz-1}$

Fig. 3 Normalized cutoff frequency vs. parameter $q$

The normalized cutoff frequency $\Omega_c$ is plotted against the filter parameter $q$ in Fig. 3.

To analyze the parallelism/orthogonality (PO) properties of the information signals let us define the connection filters

$$M_0(z) = 1$$

$$M_i(z) = (\mu_1(z))^i, \quad i=1,2,\ldots,n$$

(14)
so that we may write

\[ H_{i+1}(z) = M_1(z) H_1(z) \]  

(15)

Note that \( M_1(z) \) is an \( i \)-pole filter with a pole of multiplicity \( i \) at \( z=\alpha \).

The magnitude characteristic of the filters \( M_0, \ldots, M_4 \) are shown in Fig. 4 for \( q=0.8 \).

![Fig. 4. Magnitude vs. Freq. of connection filters](image)

We now present an approximate, and somewhat heuristic, analysis of the PO properties of the family of information signals.

Approximations -

We approximate the magnitude characteristics of the connection filters \( M_1 \) as shown in Fig. 5a.

The phase properties of \( M_1 \) will be ignored (assumed identically zero)

It is then possible to write

\[ M_1(e^{j\omega \Delta}) = L_1(\omega) + \ldots + L_n(\omega) \]  

(16)

where \( L_1(\omega) \) are the bandpass filter characteristics shown in Fig. 5b.

Further, (15) and (16) together yield
Fig. 5. Idealized connection filters and BP constituents.

\[ H_{i+1}(e^{j\omega}) = (L_1(\omega) + \ldots + L_n(\omega)) H_1(e^{j\omega}) \]  
(17a)

\[ = \phi_i(\omega) + \ldots + \phi_n(\omega) \]  
(17b)

where

\[ \phi_i(\omega) = L_i(\omega) H_1(e^{j\omega}) \]  
(18)

Clearly, \( \phi_i(\omega) \), \( i=0,1,\ldots,n \) are orthogonal. Indeed,

\[ \int_{-\omega_s/2}^{\omega_s/2} \phi_i(\omega) \phi_j(\omega) d\omega = \int_{-\omega_s/2}^{\omega_s/2} |H_1(\omega)|^2 L_i(\omega) L_j^*(\omega) d\omega \]

\[ = 0 \text{ for } i \neq j \]  
(19)

because \( L_i(\omega)=0 \) wherever \( L_j(\omega) \neq 0 \) and vice-versa. Let us state this in the time domain as

\[ <\phi_i, \phi_j> = \delta_{ij} \]  
(20)

where \( \delta_{ij} \) is the Kronecker delta and \( \{\phi_i(k)\} = \sum_{j}^{-1} \phi_i(\omega) \).

We can summarize the above discussion as follows.

**Theorem 2**

The information signals are approximately tri-orthogonal.

**Proof:** From (18) and (20) we have
where $\phi_0$, ..., $\phi_n$ are a set of orthogonal signals. The approximation sign arises because of the assumptions made earlier.

Now reversing the time-indices about $k=K$, we have

$$
\begin{bmatrix}
    y_1 \\
    y_2 \\
    y_3 \\
    \vdots \\
    y_N
\end{bmatrix} =
\begin{bmatrix}
    1 & 1 & 1 & \cdots & 1 \\
    0 & 1 & 1 & \cdots & 1 \\
    0 & 0 & 1 & \cdots & 1 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
    \psi_0 \\
    \psi_1 \\
    \psi_2 \\
    \vdots \\
    \psi_n
\end{bmatrix}
$$

(22)

where, by definition $\psi_i(k) = \phi_i(K-k)$. Since

$$
<\phi_i, \phi_j> = <\psi_i, \psi_j>
$$

$$
= \sigma_i \delta_{ij}
$$

(23)

the assertion of the theorem is proved.

Note that the weights $\sigma_i$ can be calculated (in view of (18)) as

$$
\sigma_i = \int_{-\omega_s/2}^{\omega_s/2} |L_i(\omega) H_1(e^{j\omega})|^2 \, d\omega
$$

Else, they may be calculated in the time domain via (20).
Note that (22) may be written more compactly as

\[ \mathbf{\Psi} = \mathbf{U} \mathbf{\Psi} \]  

(25)

where \( \mathbf{\Psi} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^T \), \( \mathbf{\Psi} = [\sigma_0, \sigma_1, \ldots, \sigma_n]^T \) and \( \mathbf{U} \) is the upper triangular matrix of 1's. Secondly, we observe that the Gram matrix of the information signals can be written as

\[ \mathbf{F} = \langle \mathbf{\Psi}, \mathbf{\Psi}^T \rangle \]  

(26)

where the inner-product is taken over each term of the matrix \( \mathbf{y} \mathbf{y}^T \). In terms of the approximate tri-orthogonal characterization in (22), or (25), we have

\[ \mathbf{F} = \mathbf{U} \langle \mathbf{\Psi}, \mathbf{\Psi}^T \rangle \mathbf{U}^T \]

\[ = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^T \]  

(26)

where \( \mathbf{\Sigma} \) is the \( n+1 \) dimensional diagonal matrix \( \text{diag}(\sigma_0, \sigma_1, \ldots, \sigma_n) \).

The lemma below summarizes the PO properties of the information signals.

**Lemma 3**

The correlation coefficient of a pair of information signals \( y_i \) and \( y_j \), \( i > j \) is

\[ \rho_{ij} = \frac{\sigma_{i-1}^2 + \cdots + \sigma_n^2}{\sqrt{\sigma_{i-1}^2 + \cdots + \sigma_n^2} \sqrt{\sigma_{j-1}^2 + \cdots + \sigma_n^2}} \]

(27)

**Proof** - The relation follows readily from the approximate tri-orthogonal characterization of the information signals. Using (22) and (23), we have

\[ \rho_{ij} = \frac{\sigma_{i-1}^2 + \cdots + \sigma_n^2}{\sqrt{\sigma_{i-1}^2 + \cdots + \sigma_n^2} \sqrt{\sigma_{j-1}^2 + \cdots + \sigma_n^2}} \]

which is the same as (27).
Discussion

In any system identification technique it is desirable to have basis functions that differ significantly from each other. Ideally, they should be orthogonal. By varying the parameter $q$ between 0 and 1, it is possible to generate sets of basis functions that vary between a set whose elements are almost identical to approximately a set whose elements are orthogonal.

If $q = 0$ (recall that $q$ is the z-domain pole of the first order filters $u_1(z)$), then the connection filters $N_1$ all have a nearly all-pass characteristic, and as a result $\sigma_i = 0$ for $i \neq n$. Thus $\sigma_n$ dominates all other $\sigma_i$ and we have

$$\rho_{ij} = 1$$

This is undesirable; however, such is indeed the case when unit-delays are employed such as in methods like Prony, Linear Predictive analysis, Maximum-Entropy method, etc. [1]-[2]. The strong correlation between the basis functions leads to numerical difficulties.

On the other extreme, $q=1$ results in pure (digital) integration of $y(k)$, in reverse-time, for generation of the information signals. In this case the analysis of PO properties developed above is not applicable. Therefore, consider $q$ close to one (from the left; e.g., $q = 0.99$. Now the cutoff frequencies $\omega_1, \omega_2, \ldots, \omega_n$ become crowded near the zero frequency. Hence $\sigma_i = 0$ for $i \neq 0$ and $\sigma_0$ dominates all other $\sigma_i$. Therefore, $y_2, \ldots, y_n$ have very small energy (thus very small fraction of signal information), and they are nearly orthogonal to $y_1$.

Intermediate values of $q$ lead to other useful sets of basis functions.

Example

Consider that the signal $y(k)$ has flat low-pass spectrum from $\Omega = 0$ to 0.05 with amplitude level 10. Then $q = 0.8$ and $n = 4$ lead to the values

$$\sigma_0 = 1.45, \sigma_1 = 1.28, \sigma_2 = 0.46, \sigma_3 = 0.31, \sigma_4 = 1.5$$

from which the correlation coefficients can be computed readily. For example $\rho_{23} = 0.893$, and $\rho_{14} = 0.548$. 

15
SECTION IV
PROGRAM DESCRIPTION

The program POF-FILTER is a high accuracy FORTRAN IV program designed to implement the decoupled pencil-of-functions method. Specifically, it models the impulse response of a finite-order linear system by processing the given data through a cascade of first-order linear filters in reverse-time. Some of the features of the program are stated below.

* Decoupled denominator and numerator determination. This permits fairly high order models to be determined, since the order of the Gram matrix is \( n+1 \) where \( n \) is the model order (or \( n+2 \) when data bias is also to be estimated. In contrast, the coupled procedure requires the use of a \( 2n+2 \) order Gram matrix (or \( 2n+3 \) when bias is also to be estimated).

* First-order filter, rather than pure integrators, are used for generating the family of information signals. This results in a nearly tri-orthogonal set of signals which result in a better conditioned Gram matrix than with the use of pure integrators.

* Bias correction option. Data bias can be estimated and thereby more accurate estimates for the model transfer function parameters obtained.

* Noise correction option. A preliminary routine for estimation of noise effect, and correction thereof, has been included. Theoretical work and testing/improvement of this routine remains to be done.

* Direct transmission option. Structural correctness of the model in either the presence or absence of the direct transmission term is preserved by exercising this option.

* Results of modeling are obtained in the z-domain and on an optional
basis in the s-domain also.

* Simulation option. The data modeled may be laboratory test data, or one may generate an impulse response within the program by specifying it either in the form \[ \sum_{k=1}^{n} R_z(z_t)^{k} \] or by the z-domain transfer function \[ H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}}. \]

In the simulation mode, a desired amount of bias and/or additive noise may be incorporated for test purposes.

In the laboratory-data case, a preliminary bias removal and a data scaling procedure (to maximize the effectiveness of the algorithm) has been incorporated.

* The routine which finds the cofactors of the Gram matrix of the information signals has been optimized by incorporating a scaling and corresponding descaling stages.

* For comparison purposes the program provides the option to use two other modeling techniques, the linear predictor and the Prony method. The latter can be the classical Prony, which uses \(2n+2\) data points, or the least-squares Prony. The provision of these methods within this single program is an essential step toward the evaluation of the pencil-of-functions method against other benchmark methods.

The input data cards on the subsequent pages give a description of all input variables, and in so doing provide an understanding of the program use.
INPUT DATA CARDS

CARD # 1
The first card is a title card.

CARD # 1
The first card is a title card. Columns 1 through 70 are available for alphanumeric title.
This title is reproduced in the output

CARD # 2
Another title card (columns 1-70). Not reproduced in output

CARD # 3
Another title card (columns 1-70). Not reproduced in output

CARD # 4
First option card

<table>
<thead>
<tr>
<th>Variable Name (Format)</th>
<th>Description</th>
<th>Columns</th>
<th>Preferred/default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPT (I4)</td>
<td>Number of signal points used in analysis/modeling</td>
<td>1-4</td>
<td>-</td>
</tr>
<tr>
<td>IXX, IYY (2I2)</td>
<td>Blank (unused)</td>
<td>5-8</td>
<td>-</td>
</tr>
<tr>
<td>N (I2)</td>
<td>Model order</td>
<td>9-10</td>
<td>-</td>
</tr>
<tr>
<td>ISIM (I2)</td>
<td>Simulation mode option</td>
<td>11-12</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ISIM = -1 Real (laboratory) data in integer format (10I5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 0 Real (laboratory) data in real format (10F8.6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 1 simulation from digital transfer function</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 2 simulation from sums of exponential*sinusoid form (specified in continuous-time domain)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCOMP (I2)</td>
<td>Number of terms in the sums of exponential*sinusoid form</td>
<td>13-14</td>
<td>-</td>
</tr>
<tr>
<td>IPLT (I2)</td>
<td>Plot option</td>
<td>15-16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPLT = 0 Plot routine not called</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 1 Plot of the original data and model response obtained</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= m same as 1 except that every mth point of the signals are plotted (e.g., with m=2 alternate points are plotted)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNPT (I4)</td>
<td>Number of signal points read or through simulation. NNPT should be greater than or equal to NPT</td>
<td>17-20</td>
<td>NPT</td>
</tr>
</tbody>
</table>
YYY (F10.0)  Blank (unused)  21-30
DT (F10.0)  Sampling interval  31-40
BIAS (F10.0)  Bias to be added to data  41-50

This is for use in the simulation modes (ISIM=1 or 2) when it is desired to study the effect of bias on data.

ANBIAS (F10.0)  Number of points used for a preliminary estimate of bias. That is, NBIAS= Integer(ANBIAS) number of points from the right are used to find a crude estimate of bias; this crude bias is subtracted from data.

This is useful only when real data is analyzed (ISIM=-1 or 0), and ignored when simulation data is generated.

VAR (F10.0)  Variance of noise to data  61-70

Must be used only in the simulation mode (ISIM=1 or 2) and should be left blank when real data is analyzed.

CARD # 5.1-5.n  If ISIM=-1 or 0 these cards contain real data.  1-50 if ISIM=-1
1-80 if ISIM=0

If ISIM=1 these cards contain (in (5F10.0) format) the coefficients of the z-domain transfer function; first denominator coefficients, and on succeeding card(s) the numerator coefficients.

If ISIM=2 these cards contain (in (5F10.0) format) the coefficients of the exponential*sinusoid terms; one such term on each card. Each card contains a) the weighting coefficient, b) the exponent (real), c) the radian frequency of the sinusoid, and d) the phase of the sinusoid. If the phase is zero, the sinusoid is a sine wave with zero value at k=0.

Example: If \( y(t) = e^{-t} + 7e^{-3t} \cos(2t) \) is to be simulated, then ISIM=2, NCOMP=2 and these cards are as follows:

\[
\begin{align*}
+1.00000 & \quad -1.00000 \\
+7.00000 & \quad -3.00000 \\
+2.00000 & \quad +1.57079
\end{align*}
\]
CARD # 6
Subtitle card. This alphanumeric subtitle is reproduced in the output

CARD # 7
Second option card. This numeric information is also reproduced on the output (for user convenience)

<table>
<thead>
<tr>
<th>Variable Name (Format)</th>
<th>Description</th>
<th>Preferred/Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPR (I2)</td>
<td>Print option. Increasing value results in more printing. Use 0, 1 or 2 for normal use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Values 3, 4 and 5 are useful for diagnostic purposes, or when results of intermediate computations are needed (for example, the first-order filtered signals, i.e., the information signals are printed only if IPR.GE.4)</td>
<td></td>
</tr>
<tr>
<td>IZTS (I2)</td>
<td>z-domain to s-domain conversion option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IZTS = -1 conversion to s-domain is not performed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 0 conversion performed; only the poles in s-domain printed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 1 conversion performed; in addition to poles in s-domain the s-domain denominator and numerator also printed.</td>
<td></td>
</tr>
<tr>
<td>IREM (I2)</td>
<td>Number of coefficients in the numerator (of the z-domain model), counted from the right, and set to zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-6 0</td>
<td></td>
</tr>
<tr>
<td>ISPN</td>
<td>Method option</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ISPN = -1 pencil-of-functions method employed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 1 pencil-of-functions method employed; noise added to simulated data. Must be used only when ISIM=1 or 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 0 Analysis of noise only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= -2 Covariance equations used (for LPC or Prony)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= -3 Autocorrelation equations used (for LPC or Prony)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Warning: Do not use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For LPC set IREM=19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For LPC set IREM=19</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>IFIX (12)</td>
<td><strong>Noise correction option</strong>&lt;br&gt;(under development - ignore)</td>
<td>9-10</td>
</tr>
<tr>
<td>NFIX (12)</td>
<td><strong>Auxiliary parameter for use with IFIX</strong> - ignore</td>
<td>11-12</td>
</tr>
<tr>
<td>IBIAS (12)</td>
<td><strong>Bias extraction option</strong>&lt;br&gt;IBIAS = 0 Bias extraction not exercised&lt;br&gt;IBIAS = 1 bias extracted&lt;br&gt;Warning: Do not use bias extracted values other than 0 or 1</td>
<td>13-14</td>
</tr>
<tr>
<td>IBΦ (12)</td>
<td><strong>Direct transmission term option</strong>&lt;br&gt;IBΦ = 0 Constrains b₀=0 in numerator determination&lt;br&gt;IBΦ = 1 Model assumes direct transmission is present, and b₀ is determined together with other coefficients</td>
<td>15-16</td>
</tr>
<tr>
<td>MNPT (I4)</td>
<td><strong>Controls the number of points used in the computation of error, and for printing (when IPR._GE.2) and plotting</strong></td>
<td>17-20</td>
</tr>
<tr>
<td>QI (F10.0)</td>
<td><strong>z-domain pole of the first order filter</strong>&lt;br&gt;(i.e., the parameter q of Section I)</td>
<td>21-30</td>
</tr>
<tr>
<td>DFAC (F10.0)</td>
<td><strong>Auxiliary variable for use with IFIX</strong>&lt;br&gt;(under development)</td>
<td>31-40</td>
</tr>
</tbody>
</table>
SECTION V
APPLICATION EXAMPLES

Two examples will be presented in this section. The first deals with a simulated noisy signal, \( x(k) = y(k) + w(k) \) where \( y(k) \) is the response of a third order transfer function and \( w(k) \) is a zero mean noise component. The performance of the pencil-of-functions method, with and without Gram matrix enhancement, is compared with that of other methods [1]-[3]. The second example pertains to the transient response of a conducting pipe tested at the ATHAMAS-I EMP simulator. These examples demonstrate the effectiveness of the pencil-of-functions method as a practical modeling technique.

Example 1

Let

\[
y(k) \leftrightarrow \frac{1 - 1.92z^{-1} + z^{-2}}{1 - 2.68z^{-1} + 2.476z^{-2} - 0.782z^{-3}}
\]

where the sequence \( y \) is truncated at \( K=99 \). The signal under test is obtained as

\[ x(k) = y(k) + w(k) \]

where \( w(k) \) is a zero mean, uncorrelated sequence with standard deviation equal to 0.0316. The energies of the signal and noise components are 1.82 and 0.1, respectively, hence the sequence under test has a signal-to-noise ratio of 12.6 dB. The numerical data and a plot of the signal under test are given in Appendix B.

The signal under test was analyzed by the following methods [2].

1. Pencil-of-functions method
   a) without enhancement of the Gram matrix
   b) with enhancement of the Gram matrix

2. Linear predictive coder (LPC)
   a) using autocorrelation equations
   b) using covariance equations

22
   a) using autocorrelation equations
   b) using covariance equations [3]

The computer output for these six runs is given in Appendix B. Here we summarize the fractional energy error, defined as

\[ \nu = \frac{\sum_{k=0}^{K} (\widehat{y}(k) - y(k))^2}{\sum_{k=0}^{K} y^2(k)} \]  

(28)

where \( \widehat{y}(k) \) is the impulse response of the model. Note that in simulation mode the model response can be, and is, compared with the true signal. When real data is tested the true signal is not available, hence \( x(k) \) must be used in this equation instead of \( y(k) \).

For the present example the true signal is available, hence the fractional energy error is computed by exact application of (28). The values for the various methods are tabulated in Table 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fractional energy error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencil-of-functions</td>
<td>0.0167</td>
</tr>
<tr>
<td>Pencil-of-functions with enhancement</td>
<td>0.0016</td>
</tr>
<tr>
<td>Linear Predictor (autocorrelation eqn)</td>
<td>0.1444</td>
</tr>
<tr>
<td>Linear Predictor (covariance eqn)</td>
<td>0.1567</td>
</tr>
<tr>
<td>Prony (autocorrelation eqn)</td>
<td>0.1431</td>
</tr>
<tr>
<td>Prony (covariance eqn)</td>
<td>0.1397</td>
</tr>
</tbody>
</table>

The model responses are compared with the true signal \( y(k) \) in Figures 6 through 11. The reader is cautioned that the solid line in each of these figures represents the true signal, and not the noisy signal under test. The latter is shown in Fig. B2 in Appendix B. The dotted line in each of these figures represents the model response.
METHOD: Pencil-of-functions
(reverse-time with first-order filters; q=0.8)

Fig. 6. Comparison of true and model impulse responses
METHOD: Pencil-of-functions (reverse-time with first-order filters; \(q=0.8\))

Enhancement of Gram matrix applied; IFIX=1

Fig. 7. Comparison of true and model impulse responses
Fig. 8. Comparison of true and model impulse responses

METHOD: Linear predictor (LPC) Autocorrelation equations used

--- true impulse response
----- model impulse response
True impulse response
---
model impulse response

METHOD: Linear predictor (LPC)
Covariance equations used

Fig. 9. Comparison of true and model impulse responses
Fig. 10. Comparison of true and model impulse responses
METHOD: Prony
Covariance equations used (see Markel and Gray [3])

Fig. 11. Comparison of true and model impulse responses
This example demonstrates the superiority of the pencil-of-functions method over two other widely used methods (for modeling impulse responses), namely the all pole linear predictor and the prony method.

Example 2

As a real world application we consider the use of pencil-of-functions method to the transient response of a conducting pipe tested at the ATHAMAS-I EMP simulator. The conducting pipe is 10 m long and 1 m in diameter. Hence, the true resonance of the pipe is expected to be in the neighborhood of 14 MHz. Also, the pipe has been excited in such a way that it is reasonable to expect only odd harmonics at the scattered fields. The data measured are the integral of the E-field; i.e., the measured variable is a voltage. The transient response used for analysis is shown in Fig. 12 by the solid line. The results of analysis by the pencil-of-functions method are given in Appendix C for the case of an 8th order model; the model response, with an error of 0.0125, is shown in Fig. 12 by the dotted line. Model poles are:

- fundamental: \(-4.280 + j 67.686 \text{ Mrad/s} \approx (10.794 \text{ MHz})\)
- 3rd harmonic: \(-22.470 + j218.200 \approx (34.911 \text{ MHz})\)
- curve-fit pair: \(-2.543 + j 12.890 \approx (2.051 \text{ MHz})\)
- curve-fit pair: \(-16.547 + j 88.981 \approx (14.404 \text{ MHz})\)

Note that a pole at the origin (due to the integrator) has not been obtained because we have used the bias extraction option. On the other hand, the curve-fit pole pair arises because the data do not truly pertain to the impulse response of a finite order (lumped) linear system.

Next the data were differentiated (actually differenced), and analyzed by the pencil-of-functions method. The results are given in Appendix C for 8th order analysis; the model response, with a fractional
energy error of 0.0369 is shown in Fig 13 by the dotted line (the solid line depicts the differentiated data). The model poles are:

- **fundamental**: $-9.453 + j71.596 \text{ Mrad/s} = 11.494 \text{ MHz}$
- **3rd harmonic**: $-25.996 + j222.340 \text{ Mrad/s} = 35.628 \text{ MHz}$
- **5th harmonic**: $-113.303 + j617.095 \text{ Mrad/s} = 99.855 \text{ MHz}$
- **curve-fit pair**: $-22.268 + j59.213 \text{ Mrad/s} = 10.068 \text{ MHz}$

Here again a curve-fit pole pair arises because the data do not truly pertain to the impulse response of a finite order (lumped) linear system.

We also give below the poles arising from a 6th order model (the computer output is not given, nor the graphical display of the model response):

- **fundamental**: $-9.575 + j75.106 \text{ Mrad/s} = 12.050 \text{ MHz}$
- **3rd harmonic**: $-14.866 + j221.693 \text{ Mrad/s} = 35.363 \text{ MHz}$
- **curve-fit pair**: $-14.123 + j34.045 \text{ Mrad/s} = 5.866 \text{ MHz}$

The fractional energy error in modeling is 0.0566.

Here again a curve-fit pole pair arises because the data do not truly pertain to the impulse response of a finite order (lumped) linear system.
CONDUCTING PIPE TEST

- Measured data
- Model impulse response

Method: Pencil-of-functions
Enhancement used
Bias extracted

Fig. 12. Comparison of measured data (of the response of a conducting pipe) and model impulse response
CONDUCTING PIPE TEST

- Derivative of Measured data
- Model impulse response

Method: Pencil-of-functions Enhancement used Bias extracted

Fig. 13. Comparison of the derivative of measured data and the corresponding model impulse response
REFERENCES


APPENDIX A

DETAILS OF PROGRAM POF-FILTER

LISTINGS

PURPOSE, EQUATIONS

FLOWCHART

VARIABLES

FOR THE MAIN AND SUBROUTINES
MAIN PROGRAM

FLOWCHART

Initialization
Read option cards
Read or Generate signal y

Generate information signals (CALL FILTER)

Compute Gram matrix

Perform corrections (CALL BUILDZ) (CALL FIX)

Find denominator (CALL GKRDCT)

Calculate numerator

Find model response and error. Print
Find s-domain parameters

Plot
LISTING OF POF-FILTER

PROGRAM "POF-FILTER"
IMPULSE-RESPONSE MODELING
BY PENCIL-OF-FUNCTIONS METHOD
APRIL 19;0
* DECOUPLED DENOM. AND NLM. DETERMINATION*
* FIRST-ORDER FILTERS USED*
* NOISE CORRECTION OPTION*
* BIAS CORRECTION OPTION*
* DIRECT TRANSMISSION OPTION*
* RESULTS IN BOTH Z- AND S- DOMAINS*

"POF-FILTER" MODELS IMPULS RESPONSE OF A SCATTERER/CHANNEL/NETWORK
IT CAN BE USED IN SIMULATION MODE
OR ON EXPERIMENTALLY RECORDED RESPONSES.
FOR COMPARISON PURPOSES IT ALSO PROVIDES
ON AN OPTIONAL BASIS THE FOLLOWING METHODS
# LINEAR PREDICTOR (COV OR ALTOCORR)
# PRONY
# LEAST-SQUARES PRONY

NOTE: 1350 LINES OF CODE. THIS CAN BE REDUCED SUBSTANTIALLY
FOR PARTICULAR APPLICATIONS. IN PARTICULAR, ROUTINES "PLOP"(47),
"ZTOS"(157), AND "FCLRT"(261) MAY BE ELIMINATED
IF ONLY Z-DOMAIN RESULTS NEEDED

******************************************************************************
DIMENSION F(2500),U(800),LU(800),X(800,11),G(11,11),AM(11,11)
DIMENSION GM(11,11),GST(11,11),GPQH(11,11),EN(11,11)
DIMENSION V(22),VV(22),AP(11),SR(11),SI(11),SPH(11)
DIMENSION TITLE(70),IBUF(512),IDUM(10)
DOUBLE PRECISION DT,AC,BD,ERROR
COMMON /DA0/ISP,DELTA,SIG2,DT,DE,IAS,IBIAS,DFAC
COMMON /DA1/FBAR,EBAR,FESLM,EESLM
COMMON /IO/I,J,LT,IFR,ITFR,IZFR,IZROUND,IPLOT
REIIND5
MAXPL=800
MAX=11
MAX2=2*MAX
IR=5
ILT=6
ISKIP=0
CALL VE3(ULF,0,10)
CALL VE3(LUF,0,0,11)
CALL VE3(LUF,0,0,0)
WRITE(ILT,2)
READ(IR,6) TITLE(I),I=1,70
WRITE(ILT,6) TITLE(I),I=1,70
READ(IR,6) TITLE(I),I=1,70
READ(IR,6) TITLE(I),I=1,70
READ(IR,6) TITLE(I),I=1,70
READ(IR,4) NPT,IXX,IIY,N,IAS,NCCMP,IPLOT,NNPT,
+YYYY,JT,BIAS,ANVIAS,VAR
NPI=NPI+1
NP2=NP1+1
NP3=NP1+3
NP=NP2+N+2
NP=NP1+N+1
IF (\NPT,EQ,0) \NPT=NPT
IF (DT,EQ,0) DT=1.0
IFUN=2
IF (IPLT.LT.0) IFUN=1
IF (IPLT.LT.0) IPLT=-IPLT
FMAX=1.0
IF (ISIM.EQ.3) GO TO 63
IF (ISIM.NE.1) GO TO 59
IMAX=0
DO 98 I=1,NNPT,10
READ(IR,161)(IDUM(J),J=1,10)
DO 93 J=1,10
IF (IABS(IDUM(J)).GT.IMAX) IMAX=IABS(IDUM(J))
   K=I-1+J
3 F(K)=IDUM(J)
8 CONTINUE
FMAX=IMAX
GO TO 45
9 CONTINUE
IF (ISIM.EQ.1) GO TO 49
READ(IR,153)(F(K),K=1,NNPT)
FMAX=0.0
DO 41 K=1,NNPT
   IF (ABS(F(K)).GT.FMAX) FMAX=ABS(F(K))
41 CONTINUE
NBIAS=AN3IAS
IF (NBIA-EQ.0) NBIA=0.2*NNPT
NDIE=NNPT+1-NBIAS
F9=0.0
DO 57 K=NDIE,NNPT
   F3=F3+F(K)
   F9=F9/NBIAS
57 CONTINUE
DO 56 K=1,NNPT
   F(K)=(F(K)-F3)/FMAX
56 CONTINUE
IF (ISIM.EQ.1) READ(IR,160)(V(I),I=1,NP1)
IF (ISIM.EQ.1) READ(IR,160)(V(I),I=NP2,NP2)
CALL VEOLAT(NP1,V(NP2),-1.0,0.3)
IF (ISIM.EQ.1) CALL RESPON(F,L,N,V,VV,NNPT)
IF (ISIM.EQ.1) GO TO 61
DO 60 I=1,NCOMP
   READ(IR,5)(AMP(I),SR(I),SI(I),SPH(I))
   WRITE(ILT,11)(AMP(I),SR(I),SI(I),SPH(I))
   CALL SIGNAL(F,NNPT,AMP,SR,SI,SPH,DT,NCOMP)
   CONTINUE
IF (IPLT.GE.1) CALL PLOTS(IBUF,512,9)
11 READ(IR,8)(TITLE(I),I=1,1,70)
IF (EOF(IR).NE.0) GO TO 998
WRITE(ILT,3)
WRITE(ILT,18)(TITLE(I),I=1,70)
READ(IR,9)(TITLE(I),I=1,70)
WRITE(ILT,18)(TITLE(I),I=1,70)
BACKSPACE 5
READ(IR,6)(IFR,IZTS,IREM,ISP,IFIX,NFIX,NP1,IESAS,IES0,NNPT,QI,DFAC
NSTAT=2
IF (ISPNEQ-2) NSTAT=NP1
IF (ISPNEQ-2) NSTAT=1
IZPR=0
IF (IP.GE.2J AND IPR.LE.39) IZPR=(IP-10)/10
IPR=0
IF (IP.GE.30) IPR=1
IPR=IPR-10*(IPR/10)
IF(ISPN LE -2 AND IREM GT N) IREM=N
IF(OFAC EQ 0) OFAC=0.1
IBS=1
IF(IBIAS EQ 0) IBS=0
IF(IBIAS EQ 0) IBIAS=0
IROUNO=0

CORRPT SIGNAL IF DESIRED.
PROCESS WITH FIRST ORDER FILTERS

10 CONTINUE
IF(VAR GE 1.0E-6) CALL CORRPT(F,X,VAR,NPT,MAXPL)
3 CONTINUE
IF(VAR GE 1.0E-6) GO TO 99
DO 30 K=1,NPT
30 X(K,1)=F(K)+BIAS
9 CONTINUE
INT=1
IF(ISPN LE -2) INT=2
IF(NP1 GT 1) CALL FILTER(X,NPT,NP1,MAXPL,INT)

COMPUTE GRAM MATRIX

NPP=NP1
IF(IBIAS NE 0) NPP=NP2
DO 44 I=1,NPP
DO 44 J=1,NPP
AO=0.0
IF(ISPN EQ 0.0 .AND. IROUND EQ 0) GO TO 43
DO 42 K=NSTRT,NPT
AO=AO+X(K,I)*X(K,J)
42 GN(I,J)=AO*DT
41 IJ=IABS(I-J)+1
IF(ISPN EQ -3 .AND. I GE 2) GN(I,J)=GN(1,IJ1)
4 GN(I,J)=GN(I,J)
CONTINUE
IF(IROUND EQ 0) G(I,J)=GN(I,J)
CONTINUE
IF(ISPN NE 0 .OR. IROUND NE 0)
1CALL SKROCT(GH,E,DET,V,NPP,NPP,MAX,1)
IF(IROUND EQ 0) WRITE(ILT,171) DET
IF(IROUND EQ 0) WRITE(ILT,172) DET
IF(IPR GE 1) CALL PRTMAT(GH,NPP,NPP,MAX,-1)
WRITE(ILT,1)
IRO=IRUND
IF(IROUND EQ 0) IROUND=IROND+1
IF(IRO EQ 0 .AND. ISPN GT -1) GO TO 410
IF(IFIX EQ -1) GO TO 203

ESTIMATE OF ** G

6 CALL BUIL0Z(AM,V,NP1,NPT,MAX,NFIX)
- - - - - N P1 REPLACED BY N Pp NEXT 3 CARDS - - - - -
CALL FIX(GDML,AM,GEST,E,V,NPP,NPP,SG2,MAX,IFIX)
IF(IFIX EQ 1) WRITE(ILT,482) SIG2
CALL SKROCT(GEST,E,DET,V,NPP,NPP,MAX,1)
WRITE(ILT,162) DET
IF(IPR GE 1) CALL PRTMAT(GEST,NP1,NP1,MAX,0)
DO 154 I=1,NP1
154 J=1,NP1
DO 155 J=1,NP1
155 GDML(I,J)=GEST(I,J)
NFIX=NFIX+1
39 (MAIN-4)
IF (NFIX GE 1) GO TO 156
ISKIP = 1
IROUND = 3

DETERMINE NUMERATOR

CONTINUE
IF (IBIAS EQ 1) IBIAS = 1
IF (ISPNEQ 0) GO TO 996
CALL VEQLAT(NP1, V(NP2), VV, 0, 0)
V(NP2) = -1.0
CALL RESPON(X(1,1), U, N, V, VV, NPT)
IF (IPRGE 4) WRITE(ILT, 174)
IF (IPRGE 4) WRITE(ILT, 210) (X(K,1), K =1, NPT)
CALL FILTER(K+NPT, NP1-IREM, MAXPL, 2)
L = N-IREM
IF (IBIASNE 0) L = L + 1
LP1 = L + 1
LP2 = L + 2
IF (IBIASNE 0) CALL VEQUAT(NPT, X(1-P1), L, 0, 11)
CALL VEQLAT(NPT, X(1-LP2), F, 0, 1)
CALL VEQLAT(NPT, X(1-LP2), SIAS, 0, 4)
IF (IPRGE 5) WRITE(ILT, 110) (X(K,I), K =1, NPT), I =1, LP2
L = L + 16
LP1 = L + 1
DO 216 I =1, L
DO 216 J =1, LP1
AD = 0.0
DO 215 K =1, NPT
5 AD = AD + X(K, I+1-I80) * X(K, J+1-I80)
6 G(I, J) = AD / DT
5 IF (IPRGE 5) CALL PRMTAT(G, L, LP1, MAX, 205)
CALL GROOT(G, DET, VV, L, MAX, 0)
7 IF (IPRGE 5) CALL PRMTAT(E, L, MAX, 207)
CALL VEQLAT(NP1, VV, AMP, 0, 0)
DO 220 I =1, L
AD = 0.0
DO 219 J =1, L
9 AD = AD + E(I, J) * G(J, LP1)
0 VV(I) = AD
FMEAN = 0.0
IF (IBIASNE 0) FMEAN = VV(1)
CALL VEQUAT(IREM, VV(N+I30-IREM+1), AMP, 0, 0)
V(NP2) = 0.0
CALL VEQLAT(N+I30, V(NP3-I80), VV, 0, 1)
IF (IPRGE 2) WRITE(ILT, 12) FMAX
2 WRITE(ILT, 303) FMAX
WRITE(ILT, 210) (V(I), I =1, NP1)
WRITE(ILT, 210) (V(I), I =NP2+NPNP2)
IF (IBIASNE 0) WRITE(ILT, 305) FMEAN

MODEL RESPONSE AND ERROR

IF (MNPTGE 0) MNPT = NNPT
CALL VEQLAT(NP1, V(NP2), -1.0, 0, 3)
CALL RESPON(X(1,2), U, N, V, VV, MNPT)
CALL VEQLAT(MNPT, X(1,2), FMEAN, 0, 4)
ERROR = 0.0
FFSUM = 0.0
DO 213 K =1, M, NP
X(K,1) = F(K)*FMAX*BIAS
40 (MAIN-6)
FFSuM=FFSuM+X(K,1)*X(K,1)
X(K,3)=X(K,1)-X(K,2)
ERROR=ERROR+X(K,3)*X(K,3)
FFSuM=FFSuM+DT
ERROR=ERROR*DT
RATIO=ERROR/FFSuM
WRITE(ILT,304)ERROR,FFSuM,RATIO
IF (IPR.GE.2) WRITE (ILT,112)
IF (IPR.GE.2) WRITE (ILT,110) (X(K,1),K=MNPT)
IF (IPR.GE.2) WRITE (ILT,113)
IF (IPR.GE.2) WRITE (ILT,110) (X(K,2),K=MNPT)
IF (IPR.GE.2) WRITE (ILT,114)
IF (IPR.GE.2) WRITE (ILT,110) (X(K,3),K=MNPT)
DELT=DT
IF (IZTS.GE.0) CALL ZTOS(V(1),V(NPZ),N,DELT,IZTS)

IF (IPLT.EQ.0) GO TO 239
IF (IPLT.EQ.1) GO TO 238
DO 230 I=1,2
K=0
DO 239 K=1,MNPT,IPLT
K=K+1
10 X(KK,I)=X(K,I)
MNPT=MNPT/IPLT
16 T0=0.0
DELT=DT*IPLT
CALL PLOP(MNPT,IPFUNXMAXFLTO,OELTlhYIHTIBUF)
9 CONTINUE

FORMAT STATEMENTS

FORMAT (I4,6I2,I4,F10.0)
FORM (5F10.3)
FORMAT (8I2,I4,F10.0)
FORMAT (7A1)
FORMAT (2X,7A1)
3 FORMAT (10(X,F5.0))
4 FORMAT (10(5X,I5))
FORMAT (2X,I2,*AMP=*,F8.2,* S=*,F10.4,* J=*,F10.4,
1* PHASE=*,F10.4)
FORMAT (2X,*WAVEFORMS AND NUMER. SCALLED BY XMAX=*,F12.5)
0 FORMAT (10X,6H MATRIX)
0 FORMAT (10X,6H MATRIX)
2 FORMAT (10X,11!GEƒT MATRIX,* (DET=*,G13.6,*))
0 FORMAT (2X,5(2X,G13.6))
10 FORMAT (2D10(F5.2))
0 FORMAT (2X,10(F7.4))
2 FORMAT (2X,*ORIGINAL SIGNAL(INCLUDES BIAS, IF ANY)*,/
3 FORMAT (2X,*IMPL. RESP OF MODEL (INC.3-MAT, IF IBIAS.NE.0)*,/
4 FORMAT (2X,*ERROR=F(K)-FREC(K)*,/
8 FORMAT (10X,14!NOISY X MATRIX)
9 FORMAT (10X,6H X MATRIX)
1 FORMAT (10X,16!TRUE GRAM MATRIX,* (DET=*,G13.6,*))
2 FORMAT (10X,17!NOISY GRAM MATRIX,* (DET=*,G13.6,*))
4 FORMAT (2X,*IMPLE SEED RESPONSE OF ±/A(ZI)*)
3 FORMAT (5F10.0)
1 FORMAT (10I5)
3 FORMAT (10F8.6)
3 FORMAT (2X,*ST TF 9(Z)/A(Z)*, (FMAX=*,E12.5,*))
5 FORMAT (/2X,*SS ERROR=*,G13.6,*SS SIGNAL=*,G13.6,
9* RATIO=*,G13.6,/)
FORMAT(1H1)
FORMAT(/////)

GO TO 1111
98 CONTINUE
CALL PLOT(0., 0., 999)
STOP
END
SUBROUTINE: BUILDR

PURPOSE: To generate a conversion matrix to go from $\sqrt{D_1}$ to parameters $a_i$ of $A(z)$. See equation (9) of Section I

EQUATIONS: Conversion matrix (shown below for n=3)

$$
\begin{bmatrix}
1 \\
-3 & 1 \\
3 & -2 & 1 \\
-1 & 1 & -1 & 1
\end{bmatrix}
= \text{Diag}(q^3, q^2, q, 1)
$$

ROUTINE VARIABLES

A Conversion matrix (not to be confused with the denominator polynomial)

X Vector which brings in $[\sqrt{D_1}, \sqrt{D_2}, \ldots, \sqrt{D_N}]$ to the routine and takes back the denominator parameters $[a_0, a_1, \ldots, a_n]$

N System order plus one

MAX Maximum dimensionality of matrix A

FURTHER DESCRIPTION:
SUBROUTINE BUILOR (A, X, N, MAX)

--------------------------------

CONVERSION MATRIX: REVERSE FOE PROCESSING -- I.R. MODELING

DIMENSION A(MAX,1), X(1), Y(20)

DOUBLE PRECISION DT, Y

COMMON /D4O/ISPN, DELTA, SIG2, DT, Q, IAS, ISIAS, OFAC

COMMON /I3/IR, ILT, IFR, ITPR, IZFR, IRJUND, IPLT

NM1=N-1

DO 11 I=1,N

Y(I)=0.0

DO 11 J=1,N

A(I,J)=0.0

A(N+N)=1.0

DO 20 JJ=1,NM1

J=N-JJ

DO 15 KK=1,2

K=KK-1

DO 15 I=J,NM1

A(I+K, J) = A(I+K, J) + A(I+1, J+1) * (1.0-K-K)

CONTINUE

QQ=1.0

CHANGED THRU 22 3-17-80

DO 22 II=1,NM1

I=II

QQ=QQ*CI

DO 22 JJ=1,N

A(I,J) = QQ*A(I,J)

DO 25 I=1,N

IF(IPR.GE.3) WRITE (ILT, 5) (A(I,J), JJ=1,N)

DO 25 J=1,N

Y(I) = Y(I) + A(I,J) * X(J)

DO 28 JJ=1,N

X(I) = Y(I)/Y(1)

IF(IPR.GE.3) WRITE (6,7) (X(I), I=1,N)

FORMAT (2X,10G12.5)

FORMAT (*: ESTIMATED PARAMETER VECTOR*, /*, 1G13.6)

RETURN

END

44
SUBROUTINE: BUILDZ

PURPOSE: To calculate unit noise covariance matrix for reverse-time first-order filtering case (under further development)

EQUATIONS:

\[
Z = \sum_{k=1}^{K} (K-k+1) m(k) m^T(k)
\]

where \( m(k) \) is the vector of unit-pulse responses of the connection filters

ROUTINE VARIABLES

- \( Z \) Covariance matrix for unit noise
- \( R \) Work vector
- \( NPI \) System order plus one
- \( NDIM \) Maximum dimension of the matrix \( Z \)
- \( NPT \) Number of points in signal
- \( NFIX \) Not used (blank)

FURTHER DESCRIPTION:

45
SL3ROUTINE BUILDZ(Z,R,NP1,NPT,NUIM,NFIX)

ALTERNATIVE NOISE CCV PGM FOR *GNN*

DIMENSION Z(NOIM,1),R(1)
DOUBLE PRECISION DT
COMMON /DA0/ISPA,DELT4,SI2,DT,CI,3IAS,3IAS,DFAC
COMMON /IO/I1,ILT,IPR,ITPR,IZPR,IROUND,IPLT
Q=Q l
TIME=DT*NPT
IOPT=NFIX+1
GO TO(201,1G19101,201),ICFT
1
SC=DT
N=NP1-1
R(1)=1.0
DO 1 I=1,NP1
IF(I.GE.2) R(I)=R(I-1)
DO 1 J=1,NP1
Z(I,J)=0.0
DO 2 K=1,NPT
NPTK=NPT+1-K
DO 3 J=1,NP1
DO 3 I=J,NP1
Z(I,J)=Z(I,J)+R(I)*R(J)*NPTK
R(I)=0.0
DO 4 I=1,N
R(I+1)=R(I)+R(I)
CONTINUE
DO 7 J=1,NP1
DO 7 I=J,NP1
Z(I,J)=Z(I,J)*DT**(I+J-1)
GO TO 301
1 CONTINUE
DO 210 J=1,NP1
DO 210 I=J,NP1
IF(I.EQ.1) Z(I,J)=TIME
Z(I,J)=(TIME**(I+J-1))/(I+J-1)
WRITE(ILT,161)
1 CONTINUE
DO 174 I=1,NP1
DO 166 J=1,NP1
3 Z(I,J)=Z(J,I)
IF(IPR.GE.2) WRITE(ILT,220) (Z(I,J),J=1,NP1)
4 CONTINUE
1 FORMAT(10X,* QUANT. NOISE *)
1 FCRMAT(10X,*3IAS EFFECT*)
3 FORMAT(2X,5(2X,E13.6))
RETURN
END
SUBROUTINE: CORUPT

PURPOSE: This routine, useful in simulation mode, can be called when it is desired to add random noise of given variance to the signal

EQUATIONS:

ROUTINE VARIABLES

F Input signal
X First column of X would contain the corrupted signal; the second column temporarily contains the noise added to signal
VAR Variance of noise added to signal
NPT Number of signal points
NDIM Maximum column dimensionality of X

FURTHER DESCRIPTION:

This routine needs a library routine to produce random numbers (gaussian, zero mean and uncorrelated)
SU3ROUTINE CORRUPT(F,X,VAR,NPT,NOIM)

-----------------------------
ADD'S NOISE
DIMENSION F(1),X(NOIM,1)
DOUBLE PRECISION DT,A0,30
COMMON /DA0/ISP,DELTA,SIG2,DT,QI,3IAS,IBIAS,DFAC
COMMON /DA1/FEAR,EBAR,FESUM,EESUM
COMMON /IO/IR,ILT,IPR,ITFR,IZPR,IROUND,ILT
F3BAR=0.
EBAR=0.
FESUM=0.
EESUM=0.
WRITE(ILT,469)VAR
IS=2458169
IS2=397665
SIGMA=SQRT(VAR)
CALL NRML(NPT,1,1,0.,SIGMA,IS,IS2,X(1,2),0)
DO 26 K=1,NPT
26 X(K,1)=F(K)+3IAS+X(K,2)
DO 211 K=1,NPT
211 F3=F(K)+3IAS
F3BAR=F3BAR+F3
EBAR=EBAR+X(K,2)
EESUM=EESUM+X(K,2)*X(K,2)
FESUM=FESUM+F3*X(K,2)
IF(ISPN.EQ.0)X(K,1)=X(K,2)
CONTINUE
EESUM=EESUM+DT
FESUM=2.0*FESUM+DT
F3BAR=F3BAR/NPT
EBAR=EBAR/NPT
WRITE(ILT,462)F3BAR,EBAR,FESUM,EESUM
IF(IPR.LE.2) GO TO 411
WRITE(ILT,8)
WRITE(ILT,110)(X(K),K=1,NPT)
IF(ISPN.EQ.0)GO TO 411
WRITE(ILT,18)
WRITE(ILT,115)(X(K),K=1,NPT)
WRITE(ILT,1)
CONTINUE
CONTINUE

FCRMAT STATEMENTS

FORMAT(10X,16=,ROUNDED F SIGNAL)
FORMAT(10X,16=,ROUNDED ERROR E)
FORMAT(2X,5(2X,611,4))
FORMAT(20(1X,F5.2))
FORMAT(1X,20(1X,F5.3))
FORMAT(2X,9=,VAR hANCE OF NOISE=,E11,4)
FORMAT(5)
RETURN
END
SUBROUTINE: FILTER

PURPOSE: To produce the information signals. Specifically, this routine performs reverse-time first order filtering upon the given signal (stored in the first column of the matrix X)

EQUATIONS:

\[ X(k,i+1) = q X(k+1,i+1) + X(k,i) \]

ROUTINE VARIABLES

- X Matrix of information signals. First column brings in the signal to be processed
- NPT Number of signal points
- NPI Number of information signals (model order plus one)
- NDIM Maximum column dimensionality of X
- INT Option parameter. -1 for reverse-time filtering, +2 for unit-shifts (or delays)

FURTHER DESCRIPTION:
SL340. TIM. FILTER(X, NPT, NP1, NOIM, INT)

DIMENSION X(NOIM, 1)
DOUBLE PRECISION DT, SC, 30
COMMON /DO4/ ISPA, DELTA, SIG2, DT, Q1, IBIAS, IBIAS, DFAC
COMMON /I3/ IR, I2L, IPR, ITR, IERP, IROUND, IPLT

GENERATE FIRST-ORDER FILTER PROCESSED SIGNALS FROM DATA IN X(K, 1)
INT=1 COMPARE FOR FORWARD, -1 FOR REVERSE TIME
FIRST-ORDER FILTERING

INT=2 FOR UNIT DELAYS (X(K, I+1) = X(K-1, I))
N=NP1=1
NP2=NP1+1
IOPT=INT+2
GO TO(51, 11, 11, 91, 60, IOPT)

FORWARD FIRST-ORDER FILTERING
CONTINUE
DO 40 J=1, N
JJ=J+1
X(1, JJ)=X(1, 1)
DO 40 K=2, NPT
K1=K-1
X(K, JJ)=QI*X(K1, JJ)+X(K, J)
CONTINUE
GO TO 70

REVERSE-TIME FIRST-ORDER FILTERING
CONTINUE
DO 60 J=1, N
JJ=J+1
X(NPT, JJ)=X(NPT, 1)
X(NPT, JJ)=0.0
BD=X(NPT, JJ)
DO 60 KK=2, NPT
K=NPT+1-KK
K1=K+1
CHANGED NEXT CARD 3/17/80
BD=BD+QI*X(K, J)
BD=QI*BD*X(K, J)
X(K, JJ)=BD
CONTINUE
IF(QI*IAS.EQ.0.0)GO TO 62
IPWR=QI*IAS-1
DO 61 KK=2, NPT
TIME=DT*KK
K=NPT+1-KK
X(K, NPT)=TIME**IPWR
CONTINUE
GO TO 70

GENERATE UNIT DELAYS
CONTINUE
DO 93 I=2, NP1
I1=I-1
X(I1, I)=0.0
DO 93 K=2, NPT
K1=K-1
X(K, I)=X(K1, I1)
GO TO 81
CONTINUE
SC#1, J
DO 90 I=2, NP2
SC=SC*CT
90:0<=1, NPT
X(K, I)=SC*X(K, I)
90 (FILTER 1)
CONTINUE
IF (IFR = -T.4) GO TO 99
IF (IROUND .EQ. 1) WRITE (ILT, 178) INT
IF (IROUND .EQ. 0) WRITE (ILT, 179) INT
DO 10 I = 1, NPT
10 WRITE (ILT, 110) (X(K, I), K = 1, NPT)
CONTINUE
110 FORMAT (4(1X, F12, 6))
10 FORMAT (10(1X, F7.3))
79 FORMAT (10X, "FILTER, INT=", I2, " FOF PROCESSED X")
78 FORMAT (10X, "FILTER, INT=", I2, " NOISY FOF PROCESSED X")
FORMAT (/)
RETURN
END

51 (FILTER-2)
SUBROUTINE: FIX

PURPOSE: To enhance the Gram matrix of the information signals and to effect noise corrections
(Under further development)

EQUATIONS:

\[ F = G - \sigma_{est} P \]

where \( P \) is the unit noise covariance matrix

ROUTINE VARIABLES

- \( G \): Noisy Gram matrix of reverse-time first-order filtered signals
- \( P \): Covariance matrix of unit noise (also reverse-time first-order filtered)
- \( C \): Corrected matrix
- \( D \): Work matrix
- \( N \): Dimensionality of the matrices
- \( NC \): Not used
- \( SIG \): Estimated noise variance
- \( NDIM \): Maximum dimensionality of the matrices
- \( IFIX \): Option parameter (Use IFIX=1)

FURTHER DESCRIPTION:

Under further development
SUBROUTINE FIX(G,P,C,Q,X,N,NC,SIG,NOIM,IFIX)

ESTIMATE NOISE INTENSITY SIG (ASSUME WHITE NOISE)
CORRECT NOISY MATRIX = C
[I] DENOTES NOISE MATRIX FOR UNIT NOISE
NC IS THE NONZERO SUBMATRIX OF P = CCOV OF NOISE

DIMENSION G(NOIM,1),P(NOIM,1),C(NOIM,1),O(NOIM,1),X(1)
COMMON /DQ/ISP,DELTA,SIG2,DT,QT,BIAS,IEIAS,DFAC
COMMON /ID/IR,ILT,IPR,ITFR,IZPR,IROUND,ILPT
DOUBLE PRECISION SUMDET,DT
SI=SIG
IF(IFIX.EQ.0)GO TO 51
GDP=G(1,1)/P(1,1)
JCT=0
SIG=0.0
JCT=JCT+1
SUMDET=0.0
CALL GKROCT(G,0,GOET,X,0,N,NOIM,0)
IF(JCT.EQ.1)DETG=GOET
DO 7 I=1,N
DO 7 J=1,N

SUMDET=SUMDET+G(I,J)*P(I,J)
IF(SUMDET.LT.0.0.AND.IFIX.NE.2)GO TO 11
SI=1.0/SUMDET
WRITE(ILT,32)JCT,ICT,GOET,SUMDET,SI
IF(SIG.DT.G0.EQ.1)WRITE(ILT,31)
ICT=0
CONTINUE
DO 9 I=1,N
DO 9 J=1,N
C(I,J)=G(I,J)-SI*P(I,J)
IF(IFIX.EQ.0)GO TO 11
CALL GKROCT(C,0,CDEXT,X,0,N,NOIM,0)
IF(CDEXT.LT.0.0.OR.CDEXT.GT.GOET)ICT=ICT+1
IF(ICT.GT.5)GO TO 10
IF(ICT.GT.0)SI=SIG/ICT/2.0
IF(ICT.GT.0)GO TO 51
IF(ICT.GE.5)GO TO 11
THR=DETG/DFAC
IF(ICT.EQ.1)WRITE(ILT,33)DETG,DFAC,THR
SIG=SIG+SI
IF(CDEXT.GT.0.0)CALL MEQUAT(N,N,G,C,NOIM,1)
WRITE(ILT,34)JCT,ICT,GOET,SI,CDEXT
IF(CDEXT.GT.0.0)GO TO 3
FORMAT(2X,'NOISE VAR EXCESSIVE, SIG/G0.EQ.1*)
FORMAT(1X,*J,1 GOET,SUMDET,SI*,2I2,4E11.2)
FORMAT(1X,*GOET,DFAC,THR*,2I2,4E11.2)
FORMAT(1X,*J,1 SI,CDEXT*,2I2,4E11.2)
RETURN
END

53
SUBROUTINE: GKRDC1

PURPOSE: Basically, this routine finds the cofactors and/or the inverse of a square matrix. It also calculates the denominator parameters through pencil-of-functions method (if ISP1.GE.-1) or the LPC/Prony methods (ISP1.LE.-2).

EQUATIONS:
\[ A(z) = D_i^{-1/2} (y^z)^{-n} \sum_{i=1}^{N} \sqrt{D_i} (y^{z-1})^{N-i} \]

ROUTINE VARIABLES

X Gram matrix of information signals
Y The adjoint or the inverse matrix of X is returned in Y
DET The determinant of X is returned in this variable
XLAMDA Vector of computed denominator parameters
NN Not used (blank)
N Dimensionality of X and Y
MAX Maximum dimensionality of X and Y
IOPT Option parameter; 0 for finding the inverse and determinant of the matrix X, 1 to find the denominator parameters

FURTHER DESCRIPTION:

Scaling has been introduced in this routine to enable accurate computations even for high order modeling (say 6 to 10).

This routine calls BUILDR to go from \( D_i \) to the parameters \( a_i \) of \( A(z) \).
SUBROUTINE GKRDCI (X,Y,DET,XLAMDA,NN,N,MAX,IOPT)
-----------------------------------------------
DIMENSION XLAMDA(1)
DIMENSION X(MAX,1),Y(MAX,1)
DOUBLE PRECISION A,B,C,D,E
DIMENSION N (10),SCAL (10),RSC(10),Z(10,10)
DOUBLE PRECISION DT,SC,AD,90
COMMON /DAO/ISPN,DELTA,SIG2,DT,QI,3I,AS,IBIAS,IFAC
COMMON /IO/IR,IT,IPR,ITPR,IZPR,IROUND,IPLT
IGK=1
IF(N.NE.1)GO TO 3
Y(1,1)=1.0/X(1,1)
DET=X(1,1)
GO TO 61
CONTINUE
NMAT=NN
LPC=0
IF(ISPN.LE.-2.AND.IOPT.EQ.1)LPC=1
IF(LPC.EQ.1)NMAT=N-1
-----SCALE-----
DO 11 I=1,NMAT
SCAL(I)=1.0
11 IF(X(I+LPC,1+LPC).GT.0.1E-20)SCAL(I)=SQRT(X(I+LPC,1+LPC))
RSC(I)=1.0/SCAL(I)
CONTINUE
DO 16 I=1,NMAT
DO 16 J=1,NMAT
Y(J,I)=X(J+LPC,1+LPC)*RSC(I)*RSC(J)
Z(J,I)=Y(J,I)
16 IF(ITPR.GE.1)CALL PRTMAT(Z,NMAT,NMAT,10,0)
---BEGIN GK REDUCTION---
A=1.0
DO 43 I=1,NMAT
B=0.0
L=I
M=I

FIND LARGEST ENTRY A(L,M) IN THE LOWER DIAGONAL SUBMATRIX

DO 18 J=I,NMAT
DO 18 K=I,NMAT
IF(ABS(Y(K,J)).LE.DABS(3))GO TO 18
B=ABS(Y(K,J))
L=K
M=J
CONTINUE

INTERCHANGE ROWS

IF(L.EQ.1)GO TO 24
DO 23 J=I,NMAT
C=Y(L,J)
Y(L,J)=Y(I,J)
Y(I,J)=C
23 CONTINUE

INTERCHANGE COLUMNS

IF(M.EQ.1)GO TO 29

55 (GKRDCI-1)
DO 20 J=1,NMAT
C = Y(J, M)
Y(J, M) = Y(J, I)
Y(J, I) = C
NUM(1, I) = L
NUM(2, I) = M
S = Y(I, I)
Y(I, I) = A
DO 42 J=1,NMAT
IF(J.EQ.1) GO TO 42
C = Y(I, J)
Y(I, J) = 0.0
DO 41 K = 1, NMAT
D = C * Y(K, I)
E = S + Y(K, J) + D
IFABS(E) LT .1E+00 10ABS(0) E = 0.0
Y(K, J) = E / A
CONTINUE
A = 3

RESTORE COLMNAS
DO 56 I = 2, NMAT
J = NMAT + I - 1
K = NUM(2, J)
IF(K.EQ.0) GO TO 52
DO 51 L = 1, NMAT
C = Y(K, L)
Y(K, L) = Y(J, L)
Y(J, L) = C
K = NUM(1, J)
CONTINUE
DO 56 I = 1, NMAT
DSCAL = 1.0
DO 59 I = 1, NMAT
DSCAL = SCAL(I) * DSCAL
DET = DET * SCAL(I) * SCAL(I)
IF(IPSNS.EQ.1) WRITE (13, 337) DET, A*(SCAL(I), I = 1, NMAT)
FORMAT(1X, 2ENTGA, RSC(I)) + 7E-11/4
IF(IPSNS.EQ.1) AND (IPSNS.EQ.1) GO TO 61
IF(IPSNS.EQ.1) CALL PRRTMAT(Y, NMAT, 10, MAX, 0)
IF(IPSNS.EQ.1) CALL PRRTMAT(Z, Y, NMAT, 10, MAX, 0)
DO 60 I = 1, NMAT
DO 60 J = 1, NMAT
Y(I, J) = Y(I, J) * RSC(I) * RSC(J) / A
CONTINUE
IF(IPSNS.LT.0) GO TO 1000
IF(IPSNS.GT.-2) GO TO 440
IF(Y(L, I).LE.0.0) GO TO 1000
****
SC = 1.0
DO 200 I = 2, NMAT
SC = SC * CT
SC = SC / RSC(I)
SC = SC / RSC(I)
IF(IGCM.EQ.1) Y(LMCI) = RSC(I) * Y(I, 1) / Y(I, 1)
IF(IGCM.EQ.1) GO TO 199
A=Y(I,I)
IF(Y(I,I).LT.C.0)A=0.0
IF(IQR.EQ.2)A=ABS(Y(I,I))
XLAMDA(I)=RSC(I)*SQRT(A/Y(I,I))
IF(Y(I,I).LT.0.0)XLAMDA(I)=-XLAMDA(I)
CONTINUE
XLAMDA(I)=RSC*XLAMDA(I)
AT=RSC(I)*OSCAL*SQR(Y(I,I))
IF(IPR.GE.1)WRITE(6,106)(XLAMDA(I),I=1,NMAT),AT
NPP=N
IF(IBIAS.NE.0)NPP=N-1
CALL BUILDR(Y,XLAMDA,NPP,NAX)
FORMAT(5X,*SYNTHETIC VECTOR, AND SQRT(Y11)*,/,10G12.5)
GO TO 1000
CONTINUE
IF(ITPR.LE.0)GO TO 449
WRITE(ILT,446)
CALL PRDMAT(Y,N,NAX,0)
DO 442 I=1,NMAT
DO 442 J=1,NMAT
Z(I,J)=X(I+1,J+1)
CALL PRDMAT(Z,Y,NMAT,10,NAX,0)
FORM(2X,* INVERSE AND PRODUCT MATRICES*)
CONTINUE
XLAMDA(I)=1.0
DO 450 I=2,N
XLAMDA(I)=0.0
DO 450 J=1,NMAT
XLAMDA(I)=XLAMDA(I)-Y(I-1,J)*X(J+1,1)
CONTINUE
RETURN
END
SUBROUTINE MEQUAT

PURPOSE: To set an m x n dimensional matrix B equal to a matrix A of the same dimensionality

Equation: \( B = A \)

ROUTINE VARIABLES: 
- \( M \) Row dimensionality of B and A
- \( N \) Column dimensionality of B and A
- \( B \) Matrix to be set
- \( A \) Matrix to which the matrix B is set
- \( NDIM \) Maximum number of rows permissible
- \( IOPT \) Print option; 0 for no printing, 2 or greater for printing

```fortran
SUBROUTINE MEQUAT(M,N,B,A,NDIM,IOPT)

IOPT=0 SET B TO ZERO
1 5 EQUAL TO A
10 3 TO IDENTITY
DIMENSION A(NDIM,1),B(NDIM,1)
DO 33 I=1,M
DO 33 J=1,N
IF(IOPT.NE.1) B(I,J)=0.0

IF(IOPT.EQ.10.AND.I.EQ.J) B(I,J)=1.0
IF(IOPT.EQ.1) B(I,J)=A(I,J)
CONTINUE
RETURN
END
```
SUBROUTINE:       PLOP

PURPOSE:           To plot a pair of columns of the array X
                   (This routine may be substituted by user's own routine)

SUBROUTINE PLOP(NPT,NF,Y,NDIM,TO,DT,LABEL,INDEP,IBUF)
                   -------------------------------
NPT=NUMBER OF TIME PTS  (WARNING: NDIM SHOULD BE GE. NPT+2)
NF=NUMBER OF FUNS
Y(K,T) DATA ARRAY OF DIMENSION NDIM,NF
T0=INITIAL TIME, DT=TIME INCREMENT
LABEL, INDEP = TITLES FOR Y AND X AXES
DIMENSION Y(NDIM,NF),YY(2),LABEL(1),INDEP(1)
DIMENSION X(512),IBUF(512)
COMMON /IO/IR,ILT,IPR,ITPR,IZPR,IROUND, IPLT
* =NF*NDIM
  M1=M+1
  M2=M+2
NPT1=NPT+1
NPT2=NPT+2
X(1)=T0
DO 2 K=NPT1,NPT
X(K)=X(K-1)+DT
DO 2 I=1,NF
DO 2 K=NPT1,NDIM
Y(K,I)=Y(NPT,I)

INITIALIZE(IDO, INK, 12IN, PAPER)  MAX.LENGTH=60IN
CALL PLOTMX(60.0)
SET ORIGIN
CALL PLOT(0.,0.-5,3)
CALL FACTOR(5.0/6.5)
BEGIN PLOTTING
CALL SCALE(X,6.5,NPT,1)
CALL SCALE(Y(1,1),10.0, M,1)
CALL AXIS(0.,0.,11TIME (SEC.))
* =16.6.5.0.*X(NPT1),X(NPT2))
CALL AXIS(0.,0.,16RESPONSES Y ,Y,
* =16.10.,30.,Y(M1),Y(M2))
WRITE(6,6)X(NPT1),X(NPT2)
WRITE(6,7)Y(M1),Y(M2)
FORMAT(1X,*TO,0,DIV (6.5 DIV)*.4(1X,F7.3))
FORMAT(1X,*YO,0,DIV (10 DIV)*.4(1X,F7.3))
DO 10 I=1,NF
Y(NPT1,I)=Y(M1)
Y(NPT2,I)=Y(M2)
IF (I.lt.6.1) CALL LINE(X,Y(1,I),NPT,1, I-1,I)
IF (I.ge.2) CALL DASHLN(X,Y(1,2),NPT,1)
CONTINUE
CALL PLOT(10.,0.,-3)
RETURN
END
SUBROUTINE: POLCON

PURPOSE: To combine the factors of a polynomial in order to produce the coefficients.

SUBROUTINE POLCON(C,R2,K,N)

DIMENSION C(1),R2(1)
COMPLEX C,R2,COMP
DIMENSION DC(2)
EQUIVALENCE (COMP,DC)
NP1=N+1
DO10I=2,NP1
R2(I)=0.0000
R2(1)=1.0000
DO4I=1,N

COMP=C(I)
IF(I.EQ.K.OR.(DC(1).EQ.0.000 .AND. DC(2).EQ.0.000))GO TO 4
DO2JJ=1,I
J=I-JJ+1
R2(J+1)=R2(J+1)*C(I)+R2(J)
R2(1)=R2(1)*C(I)
CONTINUE
RETURN
END

EQUATION: \[(x - c_1)(x - c_2) \ldots (x - c_n)\]

\[= r_1 + r_2x + \ldots + r_nx^{n-1} + r_{n+1}x^n\]

VARIABLES: C Vector containing the roots of the factors
R2 Vector returning the coefficients of the polynomial
K Exclude Kth factor if K#0
N Number of roots contained in C
SUBROUTINE: POLRT

PURPOSE: To find the roots of a polynomial

EQUATIONS:
\[ a_1 + a_2 x + \ldots + a_{n+1} x^n \]
\[ (x - p_1 - jq_1)(x - p_2 - jq_2) \ldots (x - p_n - jq_n) \]

ROUTINE VARIABLES
- **XCOF**: Coefficients of the polynomial (XCOF(1)=a_1)
- **COF**: Work vector
- **M**: Order of polynomial
- **ROOTR**: Real parts of the roots are returned in this vector
- **ROOTI**: Imaginary parts of the roots are returned in this vector
- **IER**: Type of error, if any, returned in this integer variable

FURTHER DESCRIPTION:
SUBROUTINE POLRT(XCOF, COF, M, ROCR, RROI, IER)

COMPUTES THE REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL

DESCRIPTION OF PARAMETERS
XCOF = VECTOR OF M+1 COEFFICIENTS OF THE POLYNOMIAL
ORDERED FROM SMALLEST TO LARGEST POWER
COF = WORKING VECTOR OF LENGTH M+1
M = ORDER OF POLYNOMIAL
ROCR = RESULTANT VECTOR OF LENGTH M CONTAINING REAL ROOTS
OF THE POLYNOMIAL
RROI = RESULTANT VECTOR OF LENGTH M CONTAINING THE
CORRESPONDING IMAGINARY ROOTS OF THE POLYNOMIAL
IER = ERROR CODE WHERE
IER=0 NO ERROR
IER=1 M LESS THAN ONE
IER=2 M GREATER THAN 36
IER=3 UNABLE TO DETERMINE ROOT WITH 500 INTERATIONS
ON 5 STARTING VALUES
IER=4 HIGH ORDER COEFFICIENT IS ZERO

DIMENSION XCOF(1), COF(1), ROCR(1), RROI(1)

DOUBLE PRECISION X0, Y0, X, Y, XPR, YPR, UX, UY, V, XT, U, XT2, YT2, SUMSQ, 1 DX, DY, TEMP, ALPHA, XCOF, COF, ROCR, RROI, ER1, ER2, XR, XS, YS, YS, TCL COMMON /lO/ IR, ILT, IPRT, IPR, IZPR, IROUND, IFLT

LIMITED TO 36TH ORDER POLYNOMIAL OR LESS.
FLOATING POINT OVERFLOW MAY OCCUR FOR HIGH ORDER
POLYNOMIALS BUT WILL NOT AFFECT THE ACCURACY OF THE RESULTS.

METHOD
NEWTON-RAPHSON ITERATIVE TECHNIQUE. THE FINAL ITERATIONS
ON EACH ROOT ARE PERFORMED USING THE ORIGINAL POLYNOMIAL
RATHER THAN THE REDUCED POLYNOMIAL TO AVOID ACCUMULATED
ERRORS IN THE REDUCED POLYNOMIAL.

ER2=1.0D+50
TCL=1.0D-8
IFIT=0
N=M
IER=0
IF (XCOF(N+1)) 10, 25, 10
10 IF (N) 15, 15, 32

SET ERROR CODE TO 1
15 IER=1
25 IF (IER) 200, 201, 200
   WRITE (6, 203) IER
   FORMAT (1X, *ERROR CALLED FROM FOLRT, IER = *, I3)
   RETURN

SET ERROR CODE TO 4
25 IER=4
   GO TO 23

SET ERROR CODE TO 2
23 (FOLRT-1)
30 IER=2
   GO TO 20
32 IF (N=36) 35,35,30
35 NX=N
   NXX=N+1
   N2=1
   KJ1 = N+1
   DO 40 L=1,KJ1
      MT=KJ1-L+1
      COF(MT)=XCOF(L)
40
SET INITIAL VALUES

45 XO= .01500101
   YO= .01000101

ZER0C INITIAL VALUE COUNTER

--------BEGIN ITERATION---------------------
X AND Y ARE THE REAL AND IMAG PARTS OF ROOT
IN=0
50 X=XO

INCREMENT INITIAL VALUES AND COUNTER

XO=-10.0*YO
   YO=-10.0*X

SET X AND Y TO CURRENT VALUE

X=XO
   Y=UC
   IN=IN+1
   GO TO 59
55 IF IT=1
   XPR=X
   YPR=Y

EVALUATE POLYNOMIAL AND DERIVATIVES

59 ICT=0
60 UX=0.0
   UY=0.0
   V =0.0
   YT=0.0
   XT=1.0
   U=COF(N+1)
IF(L) 65,130,65
65 DO 70 I=1,N
   L =N-I+1
   TEMP=COF(L)
   YT2=X*X+Y*YT
   YT2=X*YT+Y*XT
   U=U+TEMP*XT
   V=V+TEMP*YT
   FIT=I
   UX=UX+FIT*XT
   UY=UY+FIT*YT
   YT=YT2
70 3 (ITRT-2)
70 YT=YT2
    SUMSC=LX*UX+LY*UY
    IF (SLMSQ) 75, 110, 75
75  OX=(V*LY-U*UX)/SUMSQ
    X=X+OX
    OY=-(U*JY+V*UX)/SUMSQ
    ILQC=75
    IF (ITPR.GE.1) WRITE (ILT, 442) ILQC
    Y=Y+OY
    XSS=X
    YSS=Y
    IF (YSS.EQ.0.0D0) XSS=1.0D0
    IF (XSS.EQ.0.0D0) XSS=1.0D0
    RMAG=SCRT(XSS*YSS)
    MODIF=APR.80 *(SS+0.0001*RMAG IN NEXT CARD
    ERI=ABS(DX/(XSS+0.0001*RMAG)) +ABS(DY/(YSS+0.0001*RMAG))
    IF (ITPR.GE.1) WRITE (ILT, 444) CX, XSS, OY, YSS, ERI, ER2
    ILQC=77
    IF (ITPR.GE.1) WRITE (ILT, 442) ILQC
    IF (ER1.GT.ER2) GO TO 78
    ER2=ER1
    XSS=XXS
    YSS=YSS
    IF (ER1-TCL) 100, 86, 86
    STEP ITERATION COUNTER
80  ICT=ICT+1
    IF (ICT.EQ.50) 60, 85, 85
65  IF (IFIT) 130, 96, 100
90  IF (IN=5) 50, 95, 95
     ---------------------EXIT FROM ITERATIONS------------------

    SET ERROR CODE TO 3
95  IER=3
    X=XS
    Y=YS
    ER1=ER2
100 DO 105 L=1,4XX
    MT=KJ=1.+1.
    TEMP=XCOF(MT)
    XCOF(MT)=COF(L)
105 COF(L)=TEMP
    ITEP=N
    N=NX
    NX=ITEP
    IF (IFIT) 120, 55, 120
110 IF (IFIT) 115, 50, 115
115 X=XPR
    Y=YPR
20 IF IT=0
    IF (ABS(Y)-1.0D-8*ABS(X)) 135, 125, 125
25 ALP=I=XX
    SLMSQ=Z*X+Y*Y
    N=N-2
    GO TO 140
30 X=0
    N=N-1
    UX=X+1
64 (PCLRT-3)
135    Y=0.0
136    SUMSQ=0.0
137    ALPHA=X
138    N=N-1
139    COF(2)=COF(2)+ALPHA*COF(1)
140    DO 150 L=2,N
141    COF(L+1)=COF(L+1)+ALPHA*CCF(L)*SUMSQ*COF(L-1)
142    ROOTT(N2)=Y
143    ROOTR(N2)=X
144    IF (ER1.GT.TCL) WRITE(6,554)N2,ER1
145    FORMAT(1X,*ERROR ON *,I3,* TH ROOT IS *,10.3)
146    ER2=1.00+50
147    N2=N2+1
148    IF (SUMSQ) 160,165,160
149    Y=-Y
150    SUMSQ=0.0
151    GO TO 155
152    IF (N) 20,20,45
153    FORMAT(2X,*TEST IN POLRT*,I5)
154    FORMAT(1X,*DX,XSS,OY,YSS,ER1=2*,2E8.1,2X,2E8.1,2X,2E9.2)
155    RETURN
156    END
SUBROUTINE: PRDMAT

PURPOSE: This subroutine computes the product of two square matrices.

EQUATIONS: $A \times A \times B$

ROUTINE VARIABLES: A, B N x N matrices
NDIM1 Maximum row dimension of A
NDIM2 Maximum row dimension of B
LOC An integer which is printed
SUBROUTINE PRMAT(A,B,N,NCIM1,NDIM2,LOC)

DIMENSION A(NDIM1,1),B(NDIM2,1),C(10,10)

IF (LOC.GE.1) WRITE (6,5) LOC
DO 31 I=1,N
DO 31 J=1,N
C(I,J)=0.0
DO 21 K=1,N
C(I,J)=C(I,J)+A(I,K)*B(K,J)
CONTINUE
DO 41 I=1,N
DO 41 J=1,N
A(I,J)=C(I,J)
DO 45 I=1,N
WRITE (6,19) (A(I,J),J=1,N)
FORMAT (*) LOCATION/INTEGER=*,I5)
FORMAT (2X,10G13.6)
RETURN
END

FUNCTION COMB(N,M)
CALCULATES COMBINATION M OUT OF N
IF (N.LE.0) GO TO 99
L=1
LD=1
IF (M.LE.0) GO TO 8
MN1=A-M+1
DO 5 I=MN1,N
L=L*I
DO 7 I=1,N
LD=LD*I
COMB=L/ld
RETURN
END
SUBROUTINE: PRTMAT, PRTVEC, PRCEVC, PRVEC

PURPOSE: These four subroutines perform printing of arrays, PRTMAT of matrices and the other three of vectors. See subroutines for comments.

EQUATIONS:

ROUTINE VARIABLES

Subroutine PRTMAT

A Matrix being printed
M Its row dimensionality
N Its column dimensionality
NDIM Maximum number of rows permitted
LOC Use LOC=0. If nonzero, this same number is printed.

FURTHER DESCRIPTION:
SUBROUTINE PRTMAT(A,M,N,ADIM,LOC)

PRINTS A MATRIX, AND AN INTEGER (PERHAPS A LOCATION) IF LOC .GE. 1

DIMENSION A(M,ADIM,1)
IF(LOC .GE. 1) WRITE (6,5) LOC
DO 31 I=1,M
  WRITE (6,10) (A(I,J),J=1,N)
  FORMAT(* LOCATION/INTEGER=*,I5)
31 CONTINUE

FORMAT(2X,10G13.6)
RETURN
END

SUBROUTINE PRTCVEC(A,N)

PRINTS A COMPLEX VECTOR, MAX N=5
WHEN VARIABLE IS SINGLE PRECISION

COMPLEX A(N)
WRITE (6,1) (A(I),I=1,N)
FORMAT(1X,2E12.5,4(3X,2E12.5))
RETURN
END

SUBROUTINE PRTCVEC(A,N)

PRINTS A COMPLEX VECTOR, MAX N=5
WHEN VARIABLE IS DOUBLE PRECISION

COMPLEX A
DIMENSION A(1)
WRITE (6,1) (A(I),I=1,N)
FORMAT(1X,20D16.6,4(3X,20D16.6))
RETURN
END

SUBROUTINE PRTCVD(A,N)

THIS SUBROUTINE OUTPUTS DOUBLE PRECISION SINGLE DIMENSIONED ARRAY

DIMENSION A(1)
WRITE (6,1) (A(I),I=1,N)
FORMAT(1X,6D16.6)
RETURN
END
SUBROUTINE: RESPON

PURPOSE: To determine the response of the digital transfer function \( H(z) \) to an input sequence \( v(k) \). The coefficients of \( H(z) \) are given to the routine in the \( \text{NPNP2-N+N+2} \) vector GAMMA.

EQUATIONS:
\[
H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}
\]
\[
x_k = -a_1 x_{k-1} - \cdots - a_N x_{k-N} + b_0 v_k + \cdots + b_N v_{k-N}
\]

ROUTINE VARIABLES

- \( X \) The vector which returns the response of \( H(z) \)
- \( V \) Vector containing the input sequence \( v(k) \)
- \( N \) Order of transfer function \( H \)
- \( \text{GAMMA} \) Vector of coefficients of \( H \)
  \( \text{GAMMA} = (1, a_1, \ldots, a_N, -b_0, -b_1, \ldots, -b_N) \)
- \( XLAMDA \) Work vector
- \( \text{MP1} \) Number of response points generated

FURTHER DESCRIPTION:

The routine assumes zero initial conditions.
SUBROUTINE RESPRN(X,V,N,GAMMA,XLAMDA,MP1)

DIMENSION X(1),V(1),GAMMA(1),XLAMDA(1)
DOUBLE PRECISION XSAV,AC,20
NM1=N-1
NP1=N+1
NPNP1=N+N+1
NPNP2=N+N+2
DO 19 I=1,NPNP1
XLAMDA(I)=0.0
XSAV=0.0
DO 20 K=1,MP1
IF(N.EQ.1)GO TO 25
DO 21 I=1,NM1
J=NPI-I
XLAMDA(J)=XLAMDA(J-1)
CONTINUE
DO 22 I=1,N
J=NPNP2-I
XLAMDA(J)=XLAMDA(J-1)
XLAMDA(1)=XSAV
XLAMDA(NP1)=V(K)
XSAV=0.0
DO 23 I=1,NPNP1
XSAV=XSAV-GAMMA(I+1)*XLAMDA(I)
IF(ABS(XSAV).GE.1.0E10)XSAV=0.0
X(K)=XSAV
RETURN
END
SUBROUTINE: SIGNAL

PURPOSE: This routine generates a signal which is a weighted sum of exponential*sinusoid terms

EQUATIONS:

\[ f(t) = \sum_{i=1}^{m} w_i e^{-\alpha_i t} \sin(\beta_i t + \phi_i) \]

\[ F(k) = f(k\Delta) \]

ROUTINE VARIABLES

- F Vector returning the generated signal
- NPT Number of signal points generated
- AMP Vector of weights associated with each exp*sinusoid term
- SR Vector of exponents
- SI Vector of radian frequencies
- SPH Vector of phases
- DT Sampling interval
- NCOMP Number of terms

FURTHER DESCRIPTION:

This routine is useful only in the simulation mode and is called when ISIM=2
SUBROUTINE SIGNAL(F,NPT,AMP,SR,SI,SPH,DT,NCOMP)

DIMENSION F(1),AMP(1),SR(1),SI(1),SPH(1)
COMMON /10/IR,ILT,IPR,I1P,IZPR,IROUND,IPLT
DOUBLE PRECISION A,B,C,X
DO 12 K=1,NPT
   F(K)=0.0
DO 20 I=1,NCOMP
   A=SR(I)*DT
   B=SI(I)*DT
   C=SPH(I)
   DO 15 KK=1,NPT
      K=KK-1
      X=AMP(I)
      IF (A.NE.0.0)X=X*DEXP(A*K)
      IF (B.NE.0.0)X=X*DSIN(B*K+C)
      F(KK)=X+F(KK)
      CONTINUE
   IF (IPR.LT.2)GO TO 30
   WRITE(ILT,9)
   WRITE(ILT,6)(F(K),K=1,NPT)
   WRITE(ILT,1)
   CONTINUE
   FORMAT(/)
   FORMAT(20(1X,F5.2))
   FORMAT(10X,* F SIGNAL*)
RETURN
END
SUBROUTINE: \texttt{VEQUAT}

\textbf{PURPOSE:} To equate a vector \textbf{Y} to another suitable vector \textbf{EQUATIONS:} 

\begin{align*}
Y(k) &= 0 \text{ vector if } IOPT=0 \\
&= X(k) \quad \text{if } IOPT=1 \text{ or } 2 \text{ (print also)} \\
&= Y(k) \times X(1) \quad \text{if } IOPT=3 \\
&= Y(k) + X(1) \quad \text{if } IOPT=4 \\
&= \delta_k \quad (1,0,0,0..) \quad \text{if } IOPT=10 \\
&= 1 \quad (1,1,1,..) \quad \text{if } IOPT=11
\end{align*}

\textbf{VARIABLES:}

\begin{itemize}
  \item \texttt{NPT} Dimensionality of \textbf{X} 
  \item \texttt{Y} The vector to be set 
  \item \texttt{X} Auxiliary vector 
  \item \texttt{NPUL} Not used 
  \item \texttt{IOPT} Option parameter (see above)
\end{itemize}

\begin{verbatim}
SUBROUTINE VEQUAT(NPT,Y,X,NPUL,IOPT)

IOPT=0 SET Y TO ZERO
  1 SET Y=X  (PRINT IF 2)
  3 SET Y= Y*CONST X(1)
  4 SET Y= Y+CONST X(1)
  9 SET Y TO ZERO
 10 SET Y= IMPULSE
 11 SET Y=STEP

DIMENSION X(1),Y(1)
IF(IOPT.EQ.0) IOPT=9
00 33 K=1,NPT
IF(IOPT.EQ.1 OR IOPT.EQ.2) Y(K)=X(K)
IF(IOPT.EQ.3) Y(K)=Y(K)*X(1)
IF(IOPT.EQ.4) Y(K)=Y(K)+X(1)
IF(IOPT.GE.9) Y(K)=0.0
IF(IOPT.EQ.11) Y(K)=1.0
CONTINUE
IF(IOPT.EQ.2) WRITE(6,6) (Y(K),K=1,NPT)
FORMAT(2X,10G12.5)
IF(IOPT.EQ.10) Y(1)=1.0
RETURN
END
\end{verbatim}

74
SUBROUTINE: ZTOS

PURPOSE: Given the z-domain transfer function $H(z)$ this routine computes the equivalent s-domain transfer function $H(s)$

EQUATIONS: $H(z)$ $H(s)$
- Impulse invariant if $IZTS=0$
- Pulse invariant " =1
- Trapezoid invariant " =2

ROUTINE VARIABLES

B Vector of denominator parameters
A Vector of numerator parameters

Comment - unfortunately, these names are opposite to the convention used in the rest of the report. However, the reader need not be concerned about this unless he wishes to study this routine; in the latter case, he should bear this in mind.

N Order of the transfer function
DELTA Sampling interval
IZTS Option for type of conversion (see above)

FURTHER DESCRIPTION:

See Gold and Rader, or Oppenheim and Schafer, or Stanley for the theory of z-domain to s-domain conversion.
SUBROUTINE ITGS (B, A, N, DELTA, IZTS)

GIVEN THE DISCRETE DESCRIPTION THIS SUBROUTINE COMPUTES THE
EQUIVALENT CONTINUOUS DOMAIN DESCRIPTION OF A LINEAR DYNAMIC
SYSTEM.
THE INPUT ARRAYS A AND B ARE FILLED ACCORDING TO THE DIFFERENCE
EQUATION
\[ E(1) \cdot Y(K) + B(2) \cdot Y(K-1) + \ldots + B(N+1) \cdot Y(K-N) \\
- A(1) \cdot U(K) - A(2) \cdot U(K-1) - \ldots - A(N+1) \cdot U(K-N) = 0 \]

IS(1) MUST E QUAL 1
POLES OF THE CONTINUOUS DOMAIN MUST BE DISTINCT AND NON-ZERO
FOR THE TRANSFORMATION TO BE VALID

UPON RETURNING ARRAYS A AND B CONTAIN THE EQUIVALENT CONTINUOUS
DESCRIPTION STORED ACCORDING TO THE DIFFERENTIAL EQUATION
\[ B(1) \cdot \dot{Y}(T) + B(2) \cdot \dot{Y}(T) + \ldots + B(N+1) \cdot \dot{Y}(T) \\
A(1) \cdot U(T) + A(2) \cdot U(T) + \ldots + A(N+1) \cdot U(T) = 0 \]
WHERE D(M, F(T)) = THE MTH TIME DERIVATIVE OF FUNCTION, F

N + 1 ALWAYS IS 1

N = ORDER OF SYSTEM
N (MAXIMUM) = ONE LESS THAN THE DIMENSION SUBSCRIPT

IZTS = 0 ---> IMPULSE
IZTS = 1 ---> PULSE
IZTS = 2 ---> TRAPAZOIDAL
DELTA = SAMPLING INTERVAL = 1/(SAMPLING FREQUENCY)

COMPLEX CR, CA, CF, CG, CF1, CON1, CON2, COFT
COMPLEX POLE(20), ZRO(20)
DIMENSION 3(1), A(1), TEMP(20), RR(20), RI(20), CR(20), CA(20), CF(20),
CG(20), CF1(20), ZR(20), ZI(20)
COMMON /IO/ IR, ILT, IPR, IFPR, IZPR, IROUND, IPLT
CONT=0.0000
NP1=N+1
IF(A(NP1)) 410, 411, 410
1 ICHCK=2
GO TO 401
1 ICHCK=1
GO TO 400
1 CONT=(NP1)/3(NP1)
DO=+02I=1,N
A(I)=A(I)-CONT+B(I)
IF(IFPR.GE.1) WRITE (ILT, 41)
CALL PCLRT (B, TEMP, NR, RR, RI, IER)
IF(IFPR.GE.1) WRITE (ILT, 42) IER
FORMAT (2X, *GOING INTO PCLRT*)
FORMAT (2X, *RETURNED FROM PCLRT*, IER=* I2)
DO6I=1,N
CR(I)=CMPLX(RR(I), RI(I))
CF(I)=1.0000/CR(I)
IF(IIPR.GE.1) WRITE (6, 1002)
2 FORMAT (* THE POLES OF THE Z-omain*)
IF(IIPR.GE.1) CALL PROCVEC (CF, *)
IF(IIPR.LE.1) GO TO 1101
DO3I=1,N
CONT=+0010
78 (2238-1)
CON2=1.0000
DO J=1,N
CON2=CON2*CR(I) + A(N-J+1)  
IF(I-J) <= 5
CON1=CON1 + 1.0000 - CR(I) + CF(J)
CONTINUE
CA(I) = COM2/CON1  
IF(IZTS = 1) 225, 224, 230
25 DO 222 I=1,N
CA(I) = CA(I)/DELT A
GO TO 226
22 CR(I) = CGS(CR(I))/DELT A
GO TO 226
?4 DOZI = 1,N
CON1 = CLG(CR(I))/DELT A
CA(I) = CA(I) + CR(I) * CON1/(CR(I) - 1.0000)
CR(I) = CON1
GO TO 226
10 ICH = CK = 2
DO 231 I=1,N
CON1 = CLG(CR(I))/DELT A
CON2 = CA(I) * CR(I)/((1.0000 - CR(I))*(1.0000 - CR(I)))
CONT = CONT - CON2*(1.0000 + CON1*DELT A - CR(I))
CA(I) = CON2*CON1*CCN1*DELT A
* CR(I) = CON1
WRITE (6,2131)
31 FORMAT (2X, * TIME SCALED FOR SCATTERER*)
6 WRITE (6, 1004) IZTS
DT = 0.00193325
RDT = DELTA/DT
RDT = 1.0
DO 228 I=1,N
SSR = REAL(CR(I))*RDT
SSI = IMAG(CR(I))*RDT
SSM = CABS(CR(I))*RDT
SSFR = (SSM/6.2631853)
6 WRITE (6, 1010) I, SSR, SSI, SSM, SSFR
IF(IPR = LE.0) GO TO 1101
10 FORMAT (3X, I2, 5F12.4)*
IF(IPR = GE.1) WRITE (6, 1003)
13 FORMAT (* NUMERATOR CONSTANTS OF FACTORIZED H(S)*)
IF(IPR = GE.1) CALL PRCVEC(CA, N)
DO 240 I=1, N
1 POLE(I) = CR(I)
CALL POLCON(CR, CG, 0, N)
107 I = 1, N
CF(I) = 0.0000
DO 109 K = 1, N
CALL POLCON(CR, CF1, K, N)
109 J = 1, N
CF(J) = CF(J) + CF1(J) + CA(K)
CF(NP1) = 0.0000
GO TO (431, 464), ICH
IF(CF(NP1) = CONT + 2)
CG05I = 1, N
CF(I) = CF(I) + CONT*CG(I)
CONT = NE
IF(IPR = LE.1) GC TO 520
FINITE ZEROs

DO 507 I=1,NP1
  A(I)=CF(I)
  NC=3
  DO 510 I=1,NP1
    IF(ABS(A(I)) GT.1.D-6)NC=I-1
  10 CONTINUE
  IF(NC.EQ.0)GO TO 520

  N1=N0+1
  AK=A(N1)
  DO 515 I=1,N1
    A(I)=A(I)/AK
    CALL PCLR(A,TEMP,NC,RR,RI,IER)
    DO 517 I=1,N0
      ZR(I)=RR(I)
      ZI(I)=RI(I)
      ZRO(I)=CMPLX(ZR(I),ZI(I))
  15 CONTINUE
    IF(IPR.GE.2)WRITE(6,1007)AK

  07 FORMAT(* ZEROS OF H(S), NUMERATOR CONSTANT =*,E13.6)
  IF(IPR.GE.2)CALL PRVEC(ZRO,N0)

  CONTINUE
  DO20 I=1,NP1
  B(I)=CG(I)
  A(I)=CF(I)
  IF(IPR.GE.1)WRITE(6,1005)
  05 FORMAT(* S-DO MAIN DENOMINATOR*)
  IF(IPR.GE.1)CALL PRVEC(B,NP1)
  IF(IPR.GE.1)WRITE(6,1006)
  06 FORMAT(* S-DO MAIN NUMERATOR*)
  IF(IPR.GE.1)CALL PRVEC(A,NP1)
  RETURN
END
APPENDIX B

Modeling of a Noisy Test Signal

The signal considered in example 1 of Section IV is

\[ x(k) = y(k) + w(k) \]

where \( y(k) \) is the impulse response of \( (1 - 1.92z^{-1} + z^{-2})/(1 - 2.68z^{-1} + 2.476z^{-2} - 0.782z^{-3}) \) and \( w(k) \) is additive white noise. The true signal of interest, \( y(k) \), is shown in Fig. B1 and the signal under test, \( x(k) \), is shown in Fig. B2.

Given below are the (card deck) input to program POF-FILTER and, succeeding it, the printer output from the program.

**INPUT CARDS**
True impulse response

\[ y(k) \leftrightarrow \frac{1 - 1.92z^{-1} + z^{-2}}{1 - 2.68z^{-1} + 2.476z^{-2} + 0.782z^{-3}} \]

Fig. B1 True impulse response of a third order transfer function
SIGNAL UNDER TEST
- true signal + noise
- \( y(k) + w(k) \)

(see Fig. B1 for \( y(k) \))

Fig. B2. A simulated noisy signal under test
| WAVE FORM | AC | JF | F2 | T | V | S | C* | X | 60 | 72 | 74 | 76 | 78 | 80 | 82 |
|-----------|----|----|----|---|---|---|----|---|----|----|----|----|----|----|----|----|
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |
| 1.033 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 | 555 |

1.033: 555

**SIGMA** = 555

**AC SIGNAL** = 555

**TF** = 555

**IF** = 555
APPENDIX C

MODELING OF A SCATTERER RESPONSE

The signal of interest is the recorded response of a conducting pipe, considered in Example 2 of Section IV. On site digital sampling of the response was carried out as follows:

- For $k = 0$ to $k = 300$, the sampling interval is $0.390625$ ns.
- For $k = 301$ to $302$, the sampling interval is $0.703125$ ns.
- For $k = 303$ to the end, the sampling interval is $1.953125$ ns.

For analysis purposes, we resampled the first 300 data points by picking up every 5th point; the remaining data were used as such to gather a total of 245 data points. Note that we have ignored the intermediate sampling rate of the original data points 301-302.

The reconstituted signal was shown in Fig. 12 by the solid line.

Two runs will be presented below. First pertaining to the signal obtained above (from original data). The second pertains to differentiated (actually, differenced) signal produced from the resampled response (Fig. 13).

Given below are the (card deck) input to program POF-FILTER and, succeeding it, the printer output from the program; first for the signal itself and next for the differentiated signal.

**INPUT CARDS (RESAMPLED SIGNAL)**
PRINTER OUTPUT (CONDUCTING PIPE RESPONSE ANALYSIS)

RESPONSE GF = SCATTER G.

PR IMP SPF, IFIX MAP, ISS, IEP. MAPT - PENCIL OF FUNCTIONS, ENHANCED

C J-1 = 1/50 .C 0.125C .C 0.501

TRUE .25 AM MATIX (OLT) 1 2356205-31

J1 0.005 0.001 0.005 0.025 0.05 0.075 0.24

J2 0.01 0.02 0.03 0.04 0.05 0.06 0.07

J3 0.02 0.03 0.04 0.05 0.06 0.07 0.08

J4 0.03 0.04 0.05 0.06 0.07 0.08 0.09

J5 0.04 0.05 0.06 0.07 0.08 0.09 0.10

J6 0.05 0.06 0.07 0.08 0.09 0.10 0.11

J7 0.06 0.07 0.08 0.09 0.10 0.11 0.12

J8 0.07 0.08 0.09 0.10 0.11 0.12 0.13

J9 0.08 0.09 0.10 0.11 0.12 0.13 0.14

J10 0.09 0.10 0.11 0.12 0.13 0.14 0.15

J11 0.10 0.11 0.12 0.13 0.14 0.15 0.16

STIMULUS V.5= 1.55X3.62-50

ST MATIX (OLT) 1 2 3 4 5 6 7

Fz T5 I(Z) / Hz (z) = 1.7858 .30

= 7.549 23.268 44.222 63.251 82.321 101.434

= 9.24 57.19 267.63 823.21


S-FOLDS SF SI CM-S FG IZTS 0

1 -1237 2420 5677 1 4 2 5

2 -1472 4069 90572 2 4 2 5

3 -245 102 43672 12 4 20 6 2 8 6 9

4 -245 2 4 12 4 20 6 2 8 6 9

5 -245 102 43672 12 4 20 6 2 8 6 9

6 -245 2 4 12 4 20 6 2 8 6 9

7 -245 2 4 12 4 20 6 2 8 6 9

8 -245 2 4 12 4 20 6 2 8 6 9

9 -245 2 4 12 4 20 6 2 8 6 9


2 4 2 5 2 8 6 9

1 2 4 2 0 6 2 8 6 9

2 4 2 5 2 8 6 9

3 1 2 4 2 0 6 2 8 6 9

4 1 2 4 2 0 6 2 8 6 9

5 1 2 4 2 0 6 2 8 6 9

6 1 2 4 2 0 6 2 8 6 9

7 1 2 4 2 0 6 2 8 6 9

8 1 2 4 2 0 6 2 8 6 9

9 1 2 4 2 0 6 2 8 6 9


2 4 2 5 2 8 6 9

1 2 4 2 0 6 2 8 6 9

2 4 2 5 2 8 6 9

3 1 2 4 2 0 6 2 8 6 9

4 1 2 4 2 0 6 2 8 6 9

5 1 2 4 2 0 6 2 8 6 9

6 1 2 4 2 0 6 2 8 6 9

7 1 2 4 2 0 6 2 8 6 9

8 1 2 4 2 0 6 2 8 6 9

9 1 2 4 2 0 6 2 8 6 9

EXTENSION OF PENCIL OF FUNCTION METHOD TO REVERSE-TIME PROCESS--ETC(IJ)

AUG 80

V K JAIN, T K SARKAR, D D WEINER

UNCLASSIFIED

TR-80-ML-560
DIFFERENTIATED SIGNAL

\[
\frac{T_2 - T_1}{S} + \frac{T_2 - T_1}{L} + \frac{T_2 - T_1}{M} + \frac{T_2 - T_1}{N} + \frac{T_2 - T_1}{O} + \frac{T_2 - T_1}{P} + \frac{T_2 - T_1}{Q} + \frac{T_2 - T_1}{R} + \frac{T_2 - T_1}{S} + \frac{T_2 - T_1}{T} + \frac{T_2 - T_1}{U} + \frac{T_2 - T_1}{V} + \frac{T_2 - T_1}{W} + \frac{T_2 - T_1}{X} + \frac{T_2 - T_1}{Y} + \frac{T_2 - T_1}{Z}
\]

\[
\begin{align*}
T_1 & = 0.5 \times (T_2 - T_1) \\
L & = 0.5 \times (T_2 - T_1) \\
M & = 0.5 \times (T_2 - T_1) \\
N & = 0.5 \times (T_2 - T_1) \\
O & = 0.5 \times (T_2 - T_1) \\
P & = 0.5 \times (T_2 - T_1) \\
Q & = 0.5 \times (T_2 - T_1) \\
R & = 0.5 \times (T_2 - T_1) \\
S & = 0.5 \times (T_2 - T_1) \\
T & = 0.5 \times (T_2 - T_1) \\
U & = 0.5 \times (T_2 - T_1) \\
V & = 0.5 \times (T_2 - T_1) \\
W & = 0.5 \times (T_2 - T_1) \\
X & = 0.5 \times (T_2 - T_1) \\
Y & = 0.5 \times (T_2 - T_1) \\
Z & = 0.5 \times (T_2 - T_1)
\end{align*}
\]

\[
\begin{align*}
\text{SS DIFFERENTIATED SIGNAL} & = 0.52408 \\
\text{RATIO} & = 0.365624 = 0.61
\end{align*}
\]