August 1, 1965

HE JOHNS HOPKINS UNIVERSITY
RTMENT OF ELECTRICAL ENGINEERING

REPRESENTATION AND ANALYSIS OF SIGNALS

PART XIX. DIGITAL COMPUTER PROGRAMS FOR SIGNAL ANALYSIS

by

Garrett M. Odell and Robert N. McDonough

The research reported herein was sponsored by the Office of Naval Research under Contract Nonr-4010(13).

Requests for additional copies by Agencies of the Department of Defense, their Contractors, and other Government Agencies should be directed to the :

ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA

Department of Defense Contractors must be established for ASTIA services or have their "need-to-know" certified by the cognizant military agency of their project or contract. All other persons and organizations should apply to the:

U. S. DEPARTMENT OF COMMERCE
OFFICE OF TECHNICAL SERVICES
WASHINGTON 25, D. C.

August 1, 1965

THE JOHNS HOPKINS UNIVERSITY
DEPARTMENT OF ELECTRICAL ENGINEERING

REPRESENTATION AND ANALYSIS OF SIGNALS

PART XIX. DIGITAL COMPUTER PROGRAMS FOR SIGNAL ANALYSIS

by

Garrett M. Odell and Robert N. McDonough

The research reported herein was sponsored by the Office of Naval Research under Contract Nonr-4010(13).

ABSTRACT

This report describes a collection of digital computer subroutines which perform many operations useful in the linear analysis of single epoch signals and linear, stationary, stable systems. Of central importance is the process of expressing a signal as a linear combination of a set of real and/or complex, decaying, exponential functions, which may or may not be orthonormalized.

Most of the routines included have been developed concurrently with previous parts of this report series, and the purposes and methods of the various routines will be understood most readily by those who are at least basically acquainted with the material in those previous reports. The subroutine descriptions are designed to allow a person who is well acquainted with the concepts and calculations involved in signal analysis, but who has only a working knowledge of the FORTRAN computing language, to assemble various subroutines into useful composite programs. To this end, brief numerical examples are included for each subroutine to illustrate precisely how the input arguments define a given problem, and how the output arguments supply its solution.

Most of the subroutines described in this report are coded in the FORTRAN computing language. The others are coded in FAP. In addition to subroutines intended specifically for signal analysis, subroutines which perform common algebraic calculations and convenient input-output operations are included.

The source programs for all subroutines described in this report are available at the originiting activity.

ACKNOWLEDGEMENT

The authors wish to acknowledge Dr. William H. Huggins both as the driving force behind this particular report and as the co-ordinator of the several projects from which have sprung the computer programs described herein.

Special thanks goes out to Dr. Stephen S. Wolff who has given much useful advice.

The pages of this report have been reviewed by many helpful critics. Their suggestions were always welcome and often incorporated into the report. Chief among these are J. R. Sopko and William Wiseman, who, in addition to criticizing, each authored several of the write-ups contained herein.

In every case, the authors of the actual computer programs are acknowledged in the first paragraph of each write-up.

Finally, Miss Mary L. Scheller is responsible for the fine appearance of this report.

TABLE OF CONTENTS

BLANK PAGE

INTRODUCTION

This report describes a collection of digital computer programs, written primarily in the FORTRAN computing language, for performing many of the operations needed in the representation and analysis of signals using generalized exponential components. It provides the detailed descriptions of programs that have been used in the studies presented in the other reports of this series on "Representation and Analysis of Signals." The user of these programs should refer to this series of reports for more complete discussion of the basic concepts and of methods of their application to practical problems (see particularly Part XVII. Reprints on Signal Theory -- A Final Report). We offer below a brief synopsis of some of the operations performed by this set of programs.

Signals of particular interest are those which are of a transient nature, having a more-or-less well-defined beginning and bounded in magnitude by a decaying exponential. Signals of this type are produced when a dissipative dynamical system is impulsively exited. The response of the system is described by its epoch (i.e. the time at which the impulsive exitation occurs) and by the dynamical properties of the system. If the system is linear, the response may be decomposed into components associated with its natural modes. For lumped, stationary systems, these components are generally of exponential form, the damping and oscillatory properties being described by an appropriate set of complex frequencies. A transient signal arising from a single impulsive exitation may therefore be represented most efficiently by: 1) Determining its epoch, 2) Decomposing the signal into the sum of generalized exponential components whose complex frequencies and amplitudes have been chosen to best approximate the given signal.

Complications arise when the signal consists of several wavelets, overlapping in time. This can occur when the system is excited by several impulses in rapid succession, or when the signal is transmitted over several paths having different time delays. The analysis and representation of multiple-epoch signals is evidently more involved than that of single-epoch signals, and in the following report we restrict attention to single-epoch signals, all of whose epochs are assumed to occur at $t = 0$.

We are concerned with expressing the value at any instant t of each member, f, of some collection of signals by the general form

$$f(t) = \sum_{j=1}^{N} a_j \psi_j(t) . \qquad (1)$$

Any set of functions $\{\psi_j\}_1^N$ appropriate for the linear decomposition of each signal of a given collection is said to form a <u>basis</u> for the <u>representation</u> of that set of signals. The ordered set $\{a_j\}_1^N$ of scalar coordinates form the <u>representative</u> of a given signal on the prescribed basis. This set of coordinates will often be arranged as a column of N numbers, written as $\underline{F}\rangle$. The number N is called the dimensionality of the representation. The process of obtaining the representative of a signal on a given basis shall be referred to as <u>projecting</u> the signal on the basis.

Some of the programs discussed in this report deal with <u>exponential</u> bases over the semi-infinite interval; that is, the ordered set of basis components $\{\psi_j\}_1^N$ comprised of functions of the form

$$\psi_j(t) = \begin{cases} \exp(+s_j t), & \text{for } t \geq 0 \\ 0 \quad , & \text{for } t < 0 \end{cases} \qquad (2)$$

If all of the set $\{s_j\}_1^N$ are real negative numbers, the basis is said to be a real exponential basis. If some of the set $\{s_j\}_1^N$ are complex numbers, the real parts of which are negative, the basis is said to be a complex exponential basis.

Another type of basis frequently employed is an orthonormalized exponential basis. The set of functions $\{\psi_j\}_1^N$ forms an orthonormal basis over the time interval (t_1, t_2) if and only if

$$\int_{t_1}^{t_2} \psi_j(t)\psi_k(t)dt = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{if } j \neq k \end{cases} \tag{3}$$

We now define the simple and quadratic Kautz bases.

The simple Kautz basis is comprised of the set of functions $\{\psi_j\}_1^N$ having frequency domain representatives of the form

$$\psi_j(s) = \frac{\sqrt{-2s_j}}{s - s_j} \prod_{k=1}^{j-1} \frac{(s + s_k)}{(s - s_k)} \tag{4}$$

where s_j denotes the jth element of the ordered set $\{s_j\}_1^N$ which defines a real exponential basis; that is, each s_j is a real negative number.

The quadratic Kautz basis is comprised of the set of functions $\{\psi_j\}_1^N$ having frequency domain representatives

$$\psi_{2j-1}(s) = \frac{\sqrt{2p_j}s}{s^2 + p_j s + q_j} \prod_{k=1}^{j-1} \frac{s^2 - p_k s + q_k}{s^2 + p_k s + q_k}$$

$$\psi_{2j}(s) = \frac{\sqrt{2p_j q_j}}{s^2 + p_j s + q_j} \prod_{k=1}^{j-1} \frac{s^2 - p_k s + q_k}{s^2 + p_k s + q_k} \tag{5}$$

In the above, an expression of the form $s^2 + p_j s + q_j$ is called a basis quadratic factor of the quadratic Kautz basis. Each basis quadratic factor may be generated from 2 negative real exponents or from one complex conjugate pair of exponents in the following way

$$s^2 + p_j s + q_j = (s - s_{2j-1})(s - s_{2j}) \qquad (6)$$

where s_j is the jth element of the ordered set $\{s_j\}_1^N$ which defines either a real or a complex exponential basis. Then p_j, q_j is an ordered pair of real positive numbers. The members of the set of ordered pairs $\{p_j, q_j\}_1^{NH}$, where NH is one-half of the dimensionality of the original exponential basis, are called quadratic basis parameters.

It should be emphasized that a quadratic Kautz basis may be generated from either a real exponential basis of even dimensionality, or a complex exponential basis where all complex exponents occur in conjugate pairs; whereas, a simple Kautz basis may be generated only from a real exponential basis. Also note that from an exponential basis of dimension N, the number of basis quadratic factors that can be generated is NH = N/2. These NH basis quadratic factors in turn define a quadratic Kautz basis of dimension N.

The frequency domain representative of a Kautz basis function (eqs. 4, 5, 6) may be interpreted as the system function of a linear stationary operator whose unit-impulse response is the desired time function, vanishing identically prior to $t = 0$. This same operator acting as a filter is described by a weighting pattern which vanishes over the future. To obtain a weighting pattern that matches a signal, one or the other must be reversed in time. In the frequency domain, a reversal of the time scale about the point $t = 0$ corresponds to a change in sign of the

complex variable s. Thus if $f(t) \sim F(s)$, then

$$f(-t) \sim F^*(-s)$$

Time reversal produces a signal that extends backward in time from its epoch at $t = 0$, and vanishes for $t > 0$, or a weighting pattern that extends over the future and vanishes for $t < 0$. Since a physical process cannot depend upon a future event, signal analysis must involve only the past prior to the instant when a measurement is made. That is, to analyze a transient signal which vanishes prior to $t = 0$, we must arrange to reverse the signal in time (or, equivalently, to reverse the weighting pattern of the filter before it is applied to the signal). (See parts I, II, VI, IX, X, and XIII of this report series.)

To denote the two disjointed semi-infinite time intervals prior to and subsequent to the instant $t = 0$, we will use the adjectives "backward" and "forward" respectively. Thus, for example, the signal produced by an impulsive exitation at $t = 0$ of a physical system is a forward signal, whereas the weighting pattern associated with a measurement made at $t = 0$ of the output of a physical system is a backward pattern. In order for a signal to match a pattern, one or the other must be reversed in time.

For present purposes, the time reversal of a signal or a pattern may be denoted by a tilda over the symbol. In order to refer to signals and weighting patterns as abstract entities, we may use Dirac's bra-ket notation as modified by Lai and Ross. Thus $|F\rangle$ denotes a signal and $\langle G|$ denotes the weighting pattern of a measuring process. The result of measuring the signal $|F\rangle$ using the pattern $\langle G|$ is the scalar value $\langle G|F\rangle$. A weighting pattern that matches the signal $|F\rangle$ is denoted by $\langle \tilde{F}|$, and the result of measuring a signal $|F\rangle$ by its matching pattern $\langle \tilde{F}|$ is the signal energy $\|F\|^2 = \langle \tilde{F}|F\rangle$ In terms of the representatives of $|F\rangle$ in the continuous time

and frequency domains, or the discrete sampled data domain, the signal energy is

$$\langle \tilde{F} \mid F \rangle = \int_{-\infty}^{+\infty} f^*(t) f(t) dt \tag{8a}$$

$$= \int_{-j\infty}^{+j\infty} F^*(-s) F(s) \frac{ds}{2\pi j} \tag{8b}$$

$$= \sum_k f_k^* f_k \tag{8c}$$

More generally, the result of measuring a signal $\mid F\rangle$ with a pattern that matches another signal $\mid G\rangle$ is expressed by any of the functionals

$$\langle \tilde{G} \mid F \rangle = \int_{-\infty}^{+\infty} g^*(t) f(t) dt \tag{9a}$$

$$= \int_{-j\infty}^{+j\infty} G^*(-s) f(s) \frac{ds}{2\pi j} \tag{9b}$$

$$= \sum_k g_k^* f_k \tag{9c}$$

Evidently, the result of measuring $\mid G\rangle$ with a pattern that matches $\mid F\rangle$ is the complex conjugate of (9), viz.

$$\langle \tilde{G} \mid F \rangle = \langle \tilde{F} \mid G \rangle^* = \langle F \mid \tilde{G} \rangle = \langle G \mid \tilde{F} \rangle^*$$

which demonstrates the equivalence of the time reversal of either the weighting pattern or the signal. (Note that the order of the symbols in the brackets may be freely interchanged.)

Two signals $|F\rangle$ and $|G\rangle$ (or two patterns $\langle\tilde{F}|$ and $\langle\tilde{G}|$) are <u>orthogonal</u> if and only if

$$\langle\tilde{F}|G\rangle = 0 = \langle\tilde{G}|F\rangle \tag{11}$$

A set of <u>orthonormal</u> components are mutually orthogonal and each of unit energy. Thus $\{\Psi_j\}_1^N$ is a set of orthonormal components if

$$\langle\tilde{\Psi}_j|\Psi_k\rangle = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The least square approximation of a signal by a linear sum of orthonormal components such as

$$|F\rangle \overset{l.s.}{\approx} \sum_k |\Psi_k\rangle\, a_k \tag{13}$$

is easily achieved by measuring $|F\rangle$ with the set of patterns matching the $|\Psi_k\rangle$. Thus, the jth amplitude, a, is found from

$$\langle\tilde{\Psi}_j|F\rangle = \sum_k \langle\tilde{\Psi}_j|\Psi_k\rangle a_k = a_j \tag{14}$$

since the measurement made with the $|\tilde{\Psi}_j\rangle$ pattern is insensitive to all signal components other than $|\Psi_j\rangle$ by assumption of eq. (12). The set of a_j, $j=1, \ldots, n$, may be displayed in a column $\underline{F}\rangle$ as the discrete representative of the signal $|F\rangle$. Geometrically, the a_j are the values of the coordinates of the projection of the abstract signal $|F\rangle$ upon the orthogonal system provided by the set of N orthogonal components $\{\Psi_k\}_1^N$. In this way, a signal may be represented by an ordered set of N numbers.

Similarly, a linear operator or filter, $|H|$, may be represented by an $N \times N$ matrix of numbers, the kth column of which is the representative of the signal formed by $|H|$ operating on the kth basis component $|\psi_k\rangle$. That is, the kth basis component $|\psi_k\rangle$ is transformed by $|H|$ into $|H|\psi_k\rangle$. By eq. (14), the amount of $|\psi_j\rangle$ present in $|H|\psi_k\rangle$ is given by

$$\langle \tilde{\psi}_j \,|H|\, \psi_k \rangle \equiv h_{jk} \tag{15a}$$

These scalars, h_{jk}, for all values of j and k, may be arranged in matrix form to provide a numerical representative of the linear operator $|H|$ on the given basis $\{\psi_j\}_1^N$, viz.

$$\underline{H} = \{h_{jk}\} \tag{15b}$$

This matrix may be used to compute, from the representative $\underline{F}\rangle$, the representative of the signal $|H|F\rangle$ resulting from the operation $|H|$ upon $|F\rangle$. It is simply $\underline{H}\,\underline{F}\rangle$.

In matrix notation, if the discrete representative of a signal $|F\rangle$ is given as a column of N numbers

$$\underline{F}\rangle = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$ after eq. (13), and

the representative of a linear operator $|H|$ is given as the $N \times N$ matrix

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & \cdots & h_{2N} \\ \vdots & & & \\ h_{N1} & \cdots & & h_{NN} \end{bmatrix}$$

then the representative of the signal $|H|F\rangle$ is given by the column of numbers

$$\underline{H}\,\underline{F}\rangle = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} h_{11} \cdots\cdots h_{1N} \\ \vdots \\ \\ h_{N1} \cdots\cdots h_{NN} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

The convention used by FORTRAN of index arrangement in matrix multiplication is followed. That is, the $\{b_i\}_1^N$ are given by

$$b_i = \sum_{j=1}^{N} h_{ij} a_j \,. \tag{16}$$

When the operators are linear and stationary and the basis is composed of generalized exponential components, the representative of two or more operators in cascade can be computed from the matrix product of the representatives of each. That is the representative of $G|H|F\rangle$ is simply $\underline{G}\,\underline{H}\,\underline{F}\rangle$. For bases that are not of exponential type, this computation is only approximate because the basis is then not invariant under linear stationary transformations.

AVAILABILITY OF PROGRAMS

A magnetic tape is available from the Homewood Computing Center containing a tape-loadable IBM 1401 program which will produce a listing of the source programs for all of the routines described in this report and all of the required subprograms (as an option, source cards are punched; and, as another option, any routine may be output individually). Included on this tape are many control programs not described in this report. These control programs read input data, call one or more of the subroutines included in this report, and present the output in appropriate formats. Each prcgram is prefaced by a number of comment cards which define its arguments.

This tape-loadable program is compatible with the IBM 1401 computer with at least an 8k character core, and two magnetic tape drives. Any request for this tape must be accompanied by a blank reel of standard 7 track IBM magnetic tape onto which the master tape may be copied.

Occasionally, in the write-ups which follow, reference is made to the "B.C.C. (Bumblebee Computing Center) library." This is a collection of computer programs, with write-ups, available at The Johns Hopkins Computing Center.

PROGRAM LIST

In the pages that follow, each program of the Signal Analysis Library is separately described. The typical write-up consists of a brief identification of the routine, a discussion of the purpose of the routine and the method employed, a detailed list of the arguments used for input and output, a summary of any restrictions to be imposed on the arguments, any relevant comments, and finally, a numerical example. Very simple, even trivial, examples have been used, for their purpose is to illustrate precisely how the input arguments define the given information of a problem and how the output arguments supply the solution, rather than to demonstrate the applicability of these programs to involved problems.

Two more general examples have been included to demonstrate how various routines may be linked together to solve meaningful problems of signal analysis; see pages 59 and 76.

The programs described in this report are of five kinds: (1) Programs for processing analytically defined signals; (2) Programs for processing operators; (3) Programs for processing empirically defined (sampled-data) signals; (4) Utility programs for performing algebraic calculations; (5) Programs for outputing data. Below is a list of the programs in the Signal Analysis Library sorted into these five groups.

I Programs for processing analytically-defined signals:

SA-1, Projection of signal composed of real exponentials upon a simple Kautz basis.

SA-2, Projection of a pair of quadratic exponential components upon a general quadratic Kautz basis.

SA-3, Projection of a signal composed of general exponential components upon a quadratic Kautz basis.

SA-5, Conversion of a signal representative from a simple to a quadratic Kautz basis, and vice-versa.

SA-18, Inner product of two signals, each specified by rational functions in the frequency domain.

SA-23, Generation of sampled values of an exponential series.

II Programs for processing operators:

SA-6, Conversion of an operator representative from a simple to a quadratic Kautz basis, and vice-versa.

SA-7, Matrix representative on a quadratic Kautz basis of a rational system function specified in the frequency domain.

SA-8, Matrix representative of integration operator on a quadratic Kautz basis.

SA-10, Matrix representative of a special low-pass operator on a quadratic Kautz basis.

III Programs for processing empirically defined (sampled-data) signals

SA-13, Fourier spectrum of a sampled signal.

SA-17, Discrete orthonormal filter.

SA-17.5, Continuous orthonormal filter.

SA-19, Discrete Prony polynomial roots.

SA-19.5, Prony approximation package.

SA-22, Signal projection onto a general exponential basis.

SA-24, Squared error between two sampled signals.

IV Utility programs for performing algebraic calculations:

SA-11, Matrix operations.

SA-12, Eigenvalues and eigenvectors of a real symmetric matrix (Jacobi method).

SA-12.5, Eigenvalues and eigenvectors of a real symmetric matrix (Givens method).

SA-14, Polynomial zeros.

SA-16, Laurent expansion of a rational function.

SA-20, Calculation of exponent from value of exponential function at known time.

SA-20.5, S to XMOD conversion (used in SA-19.5).

SA-25, Real, simultaneous, linear equation solver.

V Programs for outputing data:

SA-17.1, Output package for SA-17.

SA-17.6, Output package for SA-17.5.

SA-26, Real and/or complex number printout.

SA-30, One-line Hollerith output subroutine.

SA-31, Continuous (Calcomp) plotter routine for discrete data points.

SA-31.5, Label routine for SA-31.

SA-32, Data plotter (IBM 1403 printer).

SA-33, printer for 1, 2, or 3 dimensional arrays.

SA-34, Matrix printout subroutine.

Signal Analysis Subroutine SA-1

1. IDENTIFICATION

Projection of a signal composed of real exponentials upon a simple Kautz basis, (FORTRAN) R. N. McDonough

2. PURPOSE

A subroutine to find the representative of a real exponential-type signal:

$$f(t) = \sum_{i=1}^{L} \alpha_i e^{-\eta_i t} \qquad \eta_i > 0 \ , \ t \geq 0$$

on a simple Kautz basis, when given the sets $\{\alpha_i\}_1^L$ $\{\eta_i\}_1^L$, and the set of positive exponents $\{-s_j\}_1^N$ from which the simple Kautz basis is generated.

3. USAGE

A. Calling Sequence:

CALL RXSPRO (L, ALPHA, ETA, N, S, A)

B. Identification of input variables:

L, number of exponentials in the signal,

ALPHA, linear array of the α coefficients of the signal

ETA, array of the exponents (positive) of the signal,

N, dimension of the basis,

S, array of the exponents (positive) of the real exponential basis; i.e. $S(I) = -s_i$

C. Identification of output variables:

$\underline{A}$, an array of the coordinates of the signal on the simple Kautz basis.

D. Restrictions:

$L \leq 20$
$N \leq 20$.

E. Availability:

Source and binary decks for SA-1 are in the SA files.

F. Example:

Suppose we wished to project $e^{-1.5t}$ on the simple Kautz basis corresponding to the following real exponentials:
$e^{-1.0t}$, $e^{-0.5t}$, $e^{-1.5t}$, $e^{-2.0t}$

Then upon entry to RXSPRO, N = 4 = the basis dimension, and the S-array would contain the following values:

S(1) = 1.0 S(2) = 0.5
S(3) = 1.5 S(4) = 2.0

The signal $e^{-1.5t}$ is entered by assigning the following values:

L = 1
ALPHA(1) = 1.0
ETA(1) = 1.5

After exit from RXSPRO the A-array would contain the values:

A(1) = 0.566 A(2) = 0.100
A(3) = 0.056 A(4) = 0.000

Which comprise the representative of $e^{-1.5t}$ on the given simple Kautz basis.

Signal Analysis Subroutine SA-2

1. IDENTIFICATION

Projection of a pair of quadratic exponential components upon a quadratic Kautz basis (FORTRAN) R. N. McDonough

2. PURPOSE

A subroutine to determine the projections on a quadratic Kautz basis of the pair of signals

$$F_1(s) = \frac{\sqrt{2R}\, s}{s^2 + Rs + T} \quad , \quad F_2(s) = \frac{\sqrt{2RT}}{s^2 + Rs + T}$$

for R, T, real, > 0

3. USAGE

A. Calling sequence:

CALL EXPRO (NH, B, R, T, A)

B. Identification of input variables:

NH, one-half of the basis dimension, = the number of basis quadratic factors,

B, linear array of the quadratic basis parameters, ordered p_1, q_1, ..., p_{NH}, q_{NH},

R,T, the signal parameters, as above.

C. Identification of output variables:

A, 2-dimensional array of the signal projections,

A(I,1) contains projections of $F_1(s)$. A(I,2) contains projections of $F_2(s)$.

D. Restrictions:

NH $\leq$ 10, R,T, > 0,

A array must have row dimension 20 in the calling program.

E. Availability:

Source and binary decks are available in the SA files.

F. Comments:

If R or T is less than zero, or if the basis is improper, the program faults and returns N=-1.

G. Example:

Suppose we wish to project the two signals

$$F_1(s) = \frac{\sqrt{2 \cdot 1}\, s}{s^2 + 1s + 2} \quad , \quad F_2(s) = \frac{\sqrt{2 \cdot 2 \cdot 1}}{s^2 + 1s + 2}$$

on a quadratic Kautz basis with basis quadratic factors

$$s^2 + 2.0s + 2.0$$

$$s^2 + 1.0s + 1.0$$

We would enter to EXPRO NH = 2, R = 1.0, T = 2.0,

B(1) = p_1 = 2.0 B(2) = q_1 = 2.0

B(3) = p_2 = 1.0 B(4) = q_2 = 1.0

Upon return from EXPRO the 2 dimensional A array would contain

A(1,1) = 0.9428	A(1,2) = 0.0000
A(2,1) = 0.0000	A(2,2) = 0.9428
A(3,1) = -0.2857	A(3,2) = -0.1347
A(4,1) = 0.0952	A(4,2) = -0.2694

The first column of A is the representative of $F_1(s)$ w.r.t. the chosen basis, and the second column is the representative of $F_2(s)$.

Signal Analysis Subroutine SA-3

1. IDENTIFICATION

Projection of a signal composed of general exponential components upon a quadratic Kautz basis, (FORTRAN)

R. N. McDonough

2. PURPOSE

A subroutine to determine the representative on a given quadratic Kautz basis of a signal of the general form:

$$f(t) = \sum_{i=1}^{2K} a_i e^{-\gamma_i t} + \sum_{i=1}^{L} e^{-\alpha_i t} (b_i \cos\beta_i t + c_i \sin\beta_i t)$$

where $a_i, \gamma_i, \beta_i, b_i, c_i$, are real and $t \geq 0$.

3. USAGE

A. Calling sequence:

CALL EXSPRO(LS,K,EXP,ABC,NH,B,SIG,ICHECK)

B. Identification of input variables:

LS = K + L

K, one-half the number of real signal exponentials

EXP, linear array of the signal exponents, ordered

$\gamma_1, \gamma_2, \ldots, \gamma_{2K}, \alpha_1, \beta_1, \ldots, \alpha_L, \beta_L$,

ABC, linear array of the signal coefficients ordered

$a_1, a_2, \ldots, a_{2K}, b_1, c_1, \ldots, b_L, c_L$,

NH, one-half the basis dimension = the number of quadratic factors of the basis,

B, linear array of the basis parameters ordered

$p_1, q_1, \ldots, p_{NH}, q_{NH}$.

C. Identification of output variables:

SIG, linear array of the signal coordinates on the quadratic Kautz basis,

ICHECK, a fault indicator. ICHECK = 0 on normal exit. If an improper exponent or basis quadratic factor is entered, program faults and ICHECK = -1 is returned.

D. Restrictions:

LS, K, NH $\leq$ 10

$\alpha_1, \gamma_1, > 0$

E. Availability:

Source and binary decks available in the SA files.

F. Comments:

Requires subroutine SA-2, EXPRO.

G. Example:

Suppose we wished to project e^{-t} on a quadratic basis corresponding to the following real basis:

$$e^{-.1t}, e^{-.2t}, e^{-.3t}, e^{-.4t}, e^{-.5t}, e^{-.6t}, e^{-1.t}, e^{-1.5t}, e^{-2.t}, e^{-2.5t}$$

The basis quadratic factors are found to be

$$\underline{s^2 + ps + q}$$

$$(s + .1)(s + .2) = s^2 + 0.3s + 0.02 \quad p_1 = .3,\ q_1 = .02$$

$$(s + .3)(s + .4) = s^2 + 0.7s + 0.12 \quad \text{etc.}$$

$$\text{etc.} \qquad = s^2 + 1.1s + 0.30$$

$$= s^2 + 2.5s + 1.50$$

$$= s^2 + 4.5s + 5.00$$

hence, upon entry to EXSPRO, NH = 5 = one-half the basis dimension; and the B array would contain the following values:

$B(1) = p_1 = .3$ $\quad$ $B(2) = q_1 = .02$

$B(3) = p_2 = .7$ $\quad$ $B(4) = q_2 = .12$

$B(5) = 1.1$ $\quad$ $B(6) = 0.3$

$B(7) = 2.5$ $\quad$ $B(8) = 1.5$

$B(9) = 4.5$ $\quad$ $B(10) = 5.0$

The signal e^{-t} is entered by assigning the following values:

LS = K + L = 1 + 0 = 1

K = 1

EXP(1) = 1.0 $\quad$ EXP(2) = 1.0

ABC(1) = 1.0 $\quad$ ABC(2) = 0.0

After exit from EXSPRO, ICHECK would equal zero, indicating that no fault has occurred and the SIG array would contain the values

SIG(1) = .587
SIG(2) = .083
SIG(3) = .355
SIG(4) = .123
SIG(5) = .078
SIG(6) = .043
SIG(7) = .005
SIG(8) = .006
SIG(9) = .000
SIG(10) = .000

Which is the representative of e^{-t} on the given quadratic Kautz basis.

Signal Analysis Subroutine SA-5

1. IDENTIFICATION

Conversion of signal representative from simple to quadratic Kautz bases, and vice versa. (FORTRAN) R. N. McDonough

2. PURPOSE

A subroutine to replace a signal representative on a simple Kautz basis of even dimension by the representative of the same signal on the equivalent quadratic Kautz basis; and, vice-versa provided the simple basis exists (that is when the basis quadratic factors have only real zeros).

3. USAGE

A. Calling sequence:

CALL BCONS (NH,B,A,ID)

B. Identification of input variables:

NH, one-half the basis dimension, = the number of quadratic factors of the basis.

B, linear array of the basis poles. If the original basis is simple, B contains the positive exponents $-s_1,-s_2,\ldots,-s_{2NH}$; if the original basis is quadratic, B contains the quadratic basis parameters:

$p_1, q_1, \ldots, p_{NH}, q_{NH},$

A, linear array of the signal coordinates. On entry, A contains the coordinates relative to the original basis.

ID, a code. If original basis is simple, enter ID = 1. If original basis is quadratic, enter ID = 2.

C. Identification of output variables:

A, on exit A is a linear array of the computed signal coordinates relative to the new basis.

D. Restrictions:

$NH \leq 10$

E. Availability:

Source and binary decks are available in the SA files.

F. Comments:

If the original basis is quadratic, and one or more of the basis quadratic factors are found to have complex zeros, ID = -1 is returned and A is left unchanged.

G. Example:

Suppose we had a representative of the signal $f(t) = e^{-.00001t}\sin t$, on the simple Kautz basis generated from the exponents $\{s_i\}_1^N$ of the series $e^{-.1t}$, $e^{-.2t}$, $e^{-.3t}$, $e^{-.4t}$, $e^{-.5t}$, $e^{-.6t}$, $e^{-1.0t}$, $e^{-1.5t}$, $e^{-2.0t}$, $e^{-2.5t}$

The representative in this case would be the set:

$$\{A_i\}_1^N = \begin{matrix} 0.4428 \\ 0.5720 \\ 0.4695 \\ 0.0110 \\ -0.6606 \\ -0.9067 \\ 0.0192 \\ 0.9383 \\ -0.6003 \\ -0.0896 \end{matrix}$$

We wish to determine the representative of the signal on the quadratic Kautz basis which is generated from the same exponents as the original simple basis,

We would enter to BCONS NH = N/2 = 5 and ID = 1, indicating the original basis is simple. The B array would contain the (positive) exponents $\{-s_j\}_1^N$, that is,

B(1)	= 0.1	B(6)	= 0.6
B(2)	= 0.2	B(7)	= 1.0
B(3)	= 0.3	B(8)	= 1.5
B(4)	= 0.4	B(9)	= 2.0
B(5)	= 0.5	B(10)	= 2.5

The A array, before entry to BCONS would contain the set $\{A_1\}_1^N$, i.e.

A(1) = .4428
A(2) = .5720
.
.
.
etc.

After exit from BCONS, the A array contains

A(1) = 0.7227
A(2) = 0.0313
A(3) = 0.3157
A(4) = 0.3477
A(5) = -1.1150
A(6) = 0.1235
A(7) = 0.7390
A(8) = -0.5786
A(9) = -0.4672
A(10) = -0.3875

Thus the representative of the signal $f(t) = [e^{-.000001t} \sin t]$, on the quadratic Kautz basis generated from the given exponents, is the set of numbers given above as the

A array. The basis quadratic factors generated from the $\{s_i\}_1^N$ are

$$s^2 + 0.3s + 0.02$$
$$s^2 + 0.7s + 0.12$$
$$s^2 + 1.1s + 0.30$$
$$s^2 + 2.5s + 1.50$$
$$s^2 + 4.5s + 5.00$$

Signal Analysis Subroutine SA-6

1. IDENTIFICATION

Conversion of an operator representative from simple to quadratic Kautz bases and vice-versa, (FORTRAN), R. N. McDonough

2. PURPOSE

A subroutine to replace the matrix representative of an operator on a simple Kautz basis of even dimension by the matrix representative of the same operator on the quadratic Kautz basis generated from the same exponents as the simple basis, and vice-versa, provided the simple basis exists (i.e., when the basis quadratic factors have only real zeros).

3. USAGE

A. Calling sequence:
CALL MATCON(NH,H,B,ID)

B. Identification of input variables:

NH, one-half the basis dimension,

H, the matrix; on entry H contains the matrix representative on the original basis,

B, linear array of the basis poles. If the original basis

is simple, B should contain the exponents $-s_1, \ldots, -s_{2NH}$. If the original basis is quadratic, B should contain the quadratic basis parameters $p_1, q_1, \ldots, p_{NH}, q_{NH}$.

ID, a code; if original basis is simple enter ID = 1; if original basis is quadratic enter ID = 2.

C. Identification of output variables:

H, on exit H contains the matrix representative on the new basis,

ID, if original basis is quadratic and one or more of the basis quadratic factors has complex zeros (i.e. if simple basis does not exist) ID = -1 is returned and H is returned unchanged.

D. Restrictions:

$NH \leq 10$

H must have row dimension 20 in calling program.

E. Example:

Suppose we have the matrix representative of the integration operator on a quadratic Kautz basis, and wish to determine its representative on the corresponding simple basis. In particular, suppose the quadratic basis is characterized by the basis quadratic factors

$$s^2 + 0.3s + 0.02$$

and

$$s^2 + 0.7s + 0.12$$

On the quadratic basis the integration operator would have the following 4x4 matrix representative:

0.000	-7.071	0.000	0.000
7.071	15.000	0.000	0.000
0.000	0.000	0.000	-2.887
0.000	18.710	2.887	5.883

Upon entry to MATCON, we set NH = 2 as one-half of the basis dimension. ID = 2 would be entered to indicate the original basis is quadratic. H would contain the above numbers, i.e. H(1,1) = 0.0, H(1,2) = -7.071, H(2,1) = 7.071, H(2,2) = 15.00, etc. The B array would contain the following values:

$$B(1) = p_1 = 0.3 \quad , \quad B(2) = q_1 = 0.02$$

$$B(3) = p_2 = 0.7 \quad , \quad B(4) = q_2 = 0.12$$

After exit from MATCON, ID would still be = 2, indicating that no fault had occurred, and H would contain the following matrix

10.000	0.000	0.000	0.000
-14.140	5.000	0.000	0.000
11.550	-8.160	3.330	0.000
-10.000	7.071	-5.770	2.500

This output matrix H is the representative of the integration operator on the simple Kautz basis generated from the same exponents as the quadratic basis, that is, in this case, from the exponents of the following set of exponentials:

$$e^{-0.1t}, \quad e^{-0.2t}, \quad e^{-0.3t}, \quad e^{-0.4t}.$$

Signal Analysis Subroutine SA-7

1. IDENTIFICATION

Matrix representative on a quadratic Kautz basis of a rational system function specified in the frequency domain.

2. PURPOSE

A subroutine to determine the matrix representative, H, of a linear, physical, stationary stable operator relative to a quadratic Kautz basis with basis quadratic factors

$s^2 + p_i s + q_i$, i=1,2,...,NH, where the operator has frequency domain representation:

$$H(s) = \frac{HNUM_1 s^M + \ldots + HNUM_M s + HNUM_{M+1}}{\prod_{k=1}^{L}(s^2 + R_k s + T_k)}$$

3. USAGE

A. Calling sequence:

CALL OPROJ(M,HNUM,L,R,T,NH,P,Q,H)

B. Identification of input variables:

M, the degree of the numerator, as above,

HNUM, linear array of the numerator coefficients, as above,

L, as above,

R,T, linear array of R_k, T_k, as above,

NH, one-half of the quadratic basis dimension, as above,

P, linear array of the quadratic basis parameters $p_1,\ldots,p_{NH}$,

Q, linear array of the quadratic basis parameters $q_1,\ldots,q_{NH}$.

C. Identification of output variables:

H, a two dimensional array containing the matrix representative of the operator H(s).

D. Restrictions:

$M \leq 20$; L,NH, ≤ 10; $M \leq 2L$.

The basis may contain only simple left-plane poles (not j-axis poles).

H(s) may contain only simple left-plane and j-axis poles.

H must have row dimension 20 in the calling program.

E. Example:

Suppose we wished to determine the matrix representative of the operator having frequency domain representation $H(s) = \frac{1.0}{s^2 + 1.0}$ on the quadratic Kautz basis with basis quadratic factors $s^2 + 2.0s + 2.0$

$s^2 + 0.05s + 1.0$

We would enter to subroutine OPROJ the following arguments:

M = 0
HNUM(1) = 1.0
L = 1
R(1) = 0.0, T(1) = 1.0
NH = 2
P(1) = 2.0, Q(1) = 2.0
P(2) = 0.05, Q(2) = 1.0

After exit from OPROJ the H array would contain the following values:

-0.20	-0.56	0.00	0.00
0.56	0.60	0.00	0.00
2.53	7.15	0.00	-20.00
-5.06	3.58	20.00	1.00

which is the matrix representative of the given operator on the prescribed basis.

F. Comment:

The 2x2 skew-symetric entries along the diagonal of H are the matrix equivalents of complex numbers. It will be noted that throughout this work complex numbers have been avoided by always using independent real quadratic factors rather than two complex linear factors. These basis elements taken in pairs are entirely equivalent, but by using them we avoid the problems of complex arithmetic.

Signal Analysis Subroutine SA-8

1. IDENTIFICATION

Matrix representative of integration operator on a quadratic Kautz basis, (FORTRAN) R. N. McDonough

2. PURPOSE

A subroutine to determine the matrix representative of the integration operator, D^{-1}, with respect to a quadratic Kautz basis.

3. USAGE

A. Calling sequence:

CALL INTOP(NH,B,H)

B. Identification of input variables:

NH, one-half of the basis dimensionality,

B, linear array of the quadratic basis parameters, ordered $p_1, q_1, \ldots, p_{NH}, q_{NH}$.

C. Identification of output variables:

H, the matrix representative of the integration operator on the given basis.

D. Restrictions:

NH $\leq$ 10

H must have row dimension 20 in the calling program.

E. Example:

Suppose we want to determine the matrix representative of D^{-1} on the quadratic basis specified by these basis quadratic factors

$$s^2 + 0.3s + 0.02$$

$$s^2 + 0.7s + 0.12$$

Upon entry to INTOP, NH = 2, and the B array contains:

$$B(1) = p_1 = 0.30 \ , \ B(2) = q_1 = 0.02$$

$$B(3) = p_2 = 0.70 \ , \ B(4) = q_2 = 0.12$$

After exit from INTOP the two-dimensional H array would contain the following values

0.000	-7.071	0.000	0.000
7.071	15.000	0.000	0.000
0.000	0.000	0.000	-2.887
0.000	18.710	2.887	5.883

This is the matrix representative of the integration operator on the given quadratic Kautz basis.

Signal Analysis Subroutine SA-10

1. IDENTIFICATION

Matrix representative of a special low-pass operator on a quadratic Kautz basis, (FORTRAN) R. N. McDonough

2. PURPOSE

A subroutine to determine the matrix representative, H, of a linear, physical, stationary, stable operator on a quadratic Kautz basis, where the operator has frequency domain representative:

$$H(s) = \frac{s^{2L-1}}{\prod_{i=1}^{L}(s^2 + R_i s + T_i)} \quad , \quad \text{for } R_i, T_i, \text{ real} \geq 0.$$

3. USAGE

A. Calling sequence:

CALL OPRO(L,R,T,NH,P,Q,H)

B. Identification of input variables:

L, as above, the number of operator quadratic factors,

R, linear array $R_1,\ldots,R_L$,

T, linear array $T_1,\ldots,T_L$,

NH, one-half of the quadratic basis dimension,

P, linear array of the quadratic basis parameters, $p_1,\ldots,p_{NH}$,

Q, linear array of the quadratic basis parameters, $q_1,\ldots,q_{NH}$.

C. Identification of output variables:

H, the computed operator matrix representative, H.

D. Restrictions:

$L \leq NH$; $NH \leq 10$,

H must have row dimension 20 in the calling program.

R_1, T_1, must be ≥ 0.

E. Example:

Suppose we wished to determine the matrix representative of an operator, H, having frequency domain representation $H(s) = \frac{s}{s^2 + 1.0}$ on the quadratic Kautz basis having basis quadratic factors

$$s^2 + 2.0s + 2.0$$

$$s^2 + 0.05s + 1.0$$

We would enter to OPRO the following values:

L = 1

R(1) = 0.00, T(1) = 1.0

NH = 2

P(1) = 2.0, Q(1) = 2.0

P(2) = .05, Q(2) = 1.0

After exit from OPRO the H array would contain the following values:

0.40	-0.28	0.00	0.00
0.28	0.80	0.00	0.00
-5.06	3.57	20.00	0.00
-2.53	-7.15	0.00	20.00

This is the matrix representative of the given H(s) on the prescribed quadratic Kautz basis.

Signal Analysis Subroutine SA-11

1. IDENTIFICATION

Matrix operations, (FORTRAN) L. G. Kelly.

2. PURPOSE

A subroutine to perform the following matrix operations:

addition	inversion
subtraction	determinant
transpose	multiplication
equality	scalar multiplication

and to read input matrices, write output matrices, and set a matrix equal to zero.

3. METHOD

The method of determinant and inversion evaluations is that of Crout, which uses the latest Wilkinson prescription for best possible accuracy.

4. USAGE

A. Calling sequence:

CALL S(OP,A,B,C,I,J,K,TEMP)

B. Identification of input variables:

The definitions of the arguments depend on the operation being performed; however, in general the variables have the following meanings:

OP, a code number of the operation to be performed,

A,B,C, the matrices of the equation A = B♦C, where ♦ indicates the operation identified by OP,

I,J,K, parameters specifying the sizes of the matrices,

TEMP, a code, usually used as an error flag.

The following table indicates the definitions of the arguments for each operation:

OPERATION	OP	I,J,K,	RESTRICTIONS
addition A = B+C	1	I= # of rows of A J= # of cols. of A	
subtraction A=B-C	2	I= # of rows of A J= # of cols. of A	
multiplication A = BxC	3	I= # of rows of B J= # of cols. of B K= # of cols. of C	Name of A cannot be name of C, or B.
inversion $A = B^{-1}$	4	I= # of rows of B	A variable TEMP must be specified, as upon return TEMP=0 if B is not singular; Temp=1 if B is singular.
determinant TEMP = det|B|	5	I= # of rows of B	A matrix A must be entered for use as temporary storage. Size of A = size of B
transpose $A = B^{T}$	6	I= # of rows of B J= # of cols. of B	Name of A cannot be name of B
equal A = B	7	I= # of rows of B J= # of cols. of B	
clear A,B,C =0	8	I= # of rows of A,B,C J= # of cols. of A,B,C	
read input matrix A	9	K = the tape unit to be read. The row and col. indices of the first element read will be stored in I and J respectively.	
write output matrix A	10	K= tape unit to be written on I= # of rows of A J= # of cols. of A	

OPERATION	OP	I,J,K,
scalar multiplication A = αB	11	I= # of rows of A J= # of cols. of A TEMP = α, the scalar number
row multiplication $\sum_J A(J)xB(J,K)=C(K)$	12	I = 1 = # of rows of A,C J= # of cols. of A K= # of cols. of B,C

C. Operation notes:

OP=4, inversion; if the matrix is singular, an error message is printed and TEMP = 1 is returned, otherwise TEMP=0 is returned.

OP=5, determinant; for this operation an array A must be entered, and A must be of the same size as B. The A array is used for internal storage of the Crout decomposed matrix of B.

OP=9, read operation; the elements are entered as follows: 4 elements per card, the field for each element is 18 positions long and contains the row, col., and element respectively. The format for each card is 4(2I3,E12.8). The elements may be entered in any order with the exception that the row and col. indices of the first element entered will be stored in I and J respectively. If the matrix is cleared by OP 8 before OP 9 is executed, then zero elements need not be entered. A blank field ends OP 9.

OP=8, clear operation; will clear one to three matrices. In each case there must be a variable entered for A,B, and C. For example:

CALL S(8,A,B,C,IMAX,JMAX,0,0), will clear A,B, and C.
CALL S(8,B,B,B,IMAX,JMAX,0,0), will clear B alone.

OP=12, row multiplication; this operation allows a one dimensional array, A, to be considered as a row matrix, and be multiplied on the right by a two dimensional matrix

B, to obtain the row matrix C. Because A and C are one dimensional arrays, economy of CORE storage can be achieved. A,B, and C must be dimensioned A(N),B(20,N),C(N).

D. Restrictions:

All arrays must have row dimension 20. This row dimension may be altered only by changing the dimension statements in routines S and T, and recompiling.

E. Comments:

This program is a package consisting of two subroutines, S and T.

Subroutine T was developed at LMSD, California.

F. Example:

Suppose we wish to perform the following operations; read in matrices H and G, where H has size NK•NK, and G has size NK•L.

compute $\underline{\omega}^t = \underline{H}^{-1} \bullet \underline{G}$

compute $\underline{\omega} = (\underline{\omega}^t)^t$

The following FORTRAN driver will accomplish the above operations:

```
      CALL S(8,H,G,G,20,20,0,0)     clear H,G
      CALL S(9,H,0,0,NK,NK,5,0)     read H from tape 5
      CALL S(9,G,0,0,NK,L,5,0)      read G from tape 5
      CALL S(4,E,H,0,NK,NK,0,TAG)   invert H
      IF (TAG) 5,4,5
4     CALL S(3,WT,E,G,NK,NK,L,0)    WT = H**-1 • G
      CALL S(6,W,WT,0,NK,L,0,0)     W = WT**t
      CALL S(10,W,0,0,L,NK,6,0)     write W on tape 6
5     CALL EXIT
      END
```

Signal Analysis Subroutine SA-12

1. IDENTIFICATION

Eigenvalues and eigenvectors of a real symetric matrix, (FORTRAN)

R. N. McDonough

2. PURPOSE

A subroutine to determine the eigenvalues and eigenvectors of a real symetric matrix using the method of Jacobi.

3. USAGE

A. Calling sequence:

CALL JACOBI(N,A,T,KTR)

B. Identification of input variables:

N, the order of the matrix A,

A, the given matrix.

C. Identification of output variables:

A, on exit, the diagonal elements of A contain the eigenvalues,

T, a two dimensional array, the columns of which are the eigenvectors, ordered to correspond to the order of the eigenvalues in the diagonal elements of A.

KTR, the number of iterations made by the routine.

D. Restrictions:

$N \leq 20$.

A and T must have row dimension 20 in the calling program.

E. Comments:

If A is the zero matrix, the subroutine returns N = -1, and no results.

If 1000 iterations are made and convergence does not take place, N = 0 is returned and A and T are returned containing the results of the 1000 iterations.

This routine is considerably slower than SA-12.5, subroutine EIGEN, but for a poorly conditioned matrix it is sometimes more accurate.

F. Example:

Suppose the eigenvalues and eigenvectors of the following matrix are desired.

A =

9.0	6.0	-14.0	18.0
6.0	-6.0	8.0	-4.0
-14.0	8.0	8.0	0.0
18.0	-4.0	0.0	0.0

We would enter the above A array and N = 4 to subroutine JACOBI.

After exit from JACOBI the A array would have as its diagonal elements A(1,1) = 26.752, A(2,2) = 0.0696, A(3,3) = -26.659, A(4,4) = -5.162, which are the eigenvalues. The T array, whose columns comprise the eigenvectors, would contain the following values:

T =

0.7871	0.1868	0.5192	-0.2756
0.0022	0.8400	-0.4591	-0.2890
-0.3166	0.5088	0.5864	0.5450
0.5293	0.0230	-0.4194	0.7372

In this case KTR = 16 = the number of iterations required.

Signal Analysis Subroutine SA-12.5

1. IDENTIFICATION

Eigenvalues and eigenvectors of a real symmetric matrix, (FORTRAN) L. G. Kelly

2. PURPOSE

A subroutine to determine the eigenvalues and eigenvectors of a real symmetric matrix using Givens' method.

3. USAGE

A. Calling sequence:

CALL EIGEN (A, VALU, N, M)

B. Identification of input variables:

A, the given matrix

N, the order of the input matrix A,

M, a code; if the eigenvectors are not to be computed enter M = 0; otherwise enter M = any non-zero number

C. Identification of output variables:

VALU, an array containing the eigenvalues,

A, on exit, A contains the eigenvectors stored column-wise.

D. Restrictions:

$2 \leq N \leq 20$.

A must have row dimension 20 in the calling program.

E. Comments:

The routine may occasionally fail to find all the eigenvectors associated with a multiple eigenvalue.

Underflow may occasionally occur from predominantly or near-zero matrices.

F. Example:

If the data presented in the example of the write-up of SA-12 were input to subroutine EIGEN, upon return from EIGEN the A array would contain the same values as the T array of subroutine JACOBI, while the eigenvalues would be stored in the VALU array.

Signal Analysis Subroutine SA-13

1. IDENTIFICATION

Fourier spectrum of a sampled signal,(FORTRAN)

R. N. McDonough

2. PURPOSE

Given 2N+1 values of a signal f(t), sampled at t=0, 1T, 2T, 3T, ..., 2NT, where T is the sampling interval, this subroutine calculates a specified number, K, of the coefficients $\{\alpha_i\}_0^K$, $\{\beta_i\}_1^K$, relevant to the expansion

$$f(nT) = \frac{\alpha_o}{2} + \sum_{i=1}^{N} \alpha_i \cos\left[\frac{2\pi}{2N+1}(i \cdot nT)\right] + \beta_i \sin\left[\frac{2\pi}{2N+1}(i \cdot nT)\right]$$

$$n = 0, 1, 2, \ldots, 2N$$

That is, only the coefficients α_i, i = 0, 1, ...K, and β_i, i = 1, 2, ..., K, are calculated in a given run.

We have here a truncated discrete form of the Fourier representation of a signal f(t) with period τ :

$$f(t) = \frac{\alpha_o}{2} + \sum_{i=1}^{\infty} \left[a_i \cos \frac{2\pi}{\tau} it + b_i \sin \frac{2\pi}{\tau} it \right]$$

2. USAGE

A. Calling sequence:

CALL FORAN (NSP, F, K, A)

B. Identification of input variables:

NSP, the odd number of sample points in the F array; NSP = 2N+1, where N is as above,

F, a linear array of the NSP sampled values of the given signal,

K, a parameter to specify the number of Fourier coefficients to be calculated, that is α_i, i = 0, 1, ...K, and β_i, i = 1, 2, ..., K are calculated.

C. Identification of output variables:

A, linear array of the calculated coefficients, ordered so that

$$A(1) = \alpha_o$$
$$A(2) = \alpha_1, \qquad A(3) = \beta_1,$$
$$\ldots, A(2K) = \alpha_K, \; A(2K+1) = \beta_K$$

D. Restrictions

$K \leq N$

E. Example:

Suppose we wish to calculate the coefficients of the first 3 (sin,cos) pairs of the Fourier expansion of a traingular wave oscillating between +1 and -1 and of period 16, sampled at 17 points, with the time increment T = 1,

Then upon entry to FORAN:

NSP contains the number 17

K contains the number 3

The F-array contains the values:

$$F = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0, -\frac{1}{4}, -\frac{1}{2}, -\frac{3}{4}$$

$$-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, 0$$

Upon exit from FORAN the A-array will contain the values:

A(1) =	0.00	A(5) =	-0.077
A(2) =	0.145	A(6) =	-0.068
A(3) =	0.774	A(7) =	-0.111
A(4) =	0.030		

Then the F-array can be approximated by an F'-array such that:

$$F(I) \approx F'(I) = 0.145 \cos \frac{2\pi(I-1)}{17} + 0.774 \sin \frac{2\pi(I-1)}{17}$$

$$- 0.030 \cos \frac{4\pi(I-1)}{17} - 0.777 \sin \frac{4\pi(I-1)}{17}$$

$$- 0.068 \cos \frac{6\pi(I-1)}{17} - 0.111 \sin \frac{6\pi(I-1)}{17}$$

Where I runs from 1 to 17.

Signal Analysis Subroutine SA-14

1. IDENTIFICATION

Polynomial zeros, (FORTRAN)

R. N. McDonough

2. PURPOSE

A subroutine to determine all zeros of a polynomial of degree N having real coefficients.

3. METHOD

A combination of Newton's and Bairstow's methods is used.

4. USAGE

A. Calling sequence:

CALL ZEROS (N, A, NREAL, ROOTS, ACHECK)

B. Identification of input variables:

N, the degree of the polynomial,

A, linear array of the coefficients of the polynomial ordered as in

$$A_1 X^N + A_2 X^{N-1} + \ldots + A_N X + A_{N+1} .$$

C. Identification of output variables: (When no fault has occurred.)

NREAL, number of real zeros found,

ROOTS, linear array of the zeros found order such that:

$ROOTS_1, \ldots, ROOTS_{NREAL}$ are the real zeros,

$ROOTS_{NREAL+1} \pm j(ROOTS_{NREAL+2})$ is the first conjugate pair of complex zeros.

Etc.

⋮

:

$ROOTS_{N-1} \pm j(ROOTS_N)$ is last conjugate pair of complex zeros,

ACHECK, linear array of coefficients of a polynomial constructed from the zeros found by the routine, ordered same as the A array.

C. Identification of output variables in the case of a fault:

Note: A fault may occur in the subroutine due to failure to converge after a reasonable number of iterations, or because of most unusual symmetry in the locations of the zeros. The presence of a fault is indicated by NREAL returned as negative. In this case output variables are defined as follows:

NREAL, - the number of real zeros found up until the fault occurred +1,

ROOTS, linear array of zeros found up until the fault occurred, ordered in the same way as the ROOTS array for the normal case,

ACHECK, coefficients of a polynomial constructed from the zeros actually found, ordered in the same way as the ACHECK array in the normal case, except that ACHECK (2) is the coefficient of the highest power of X, rather than ACHECK (1),

N, an input variable, is changed upon return to equal the total number of zeros found up until the fault occurred.

D. Restrictions:

$N \leq 50$

ACHECK must have dimenison = N+2 or greater in calling program.

E. Comments:

Accuracy of the zeros is estimated as seven figures, except that a multiple zero will be found as a group of

slightly separated simple zeros and accuracy drops to perhaps four figures.

The A array is destroyed before return.

F. Example:

Suppose we wished to determine the zeros of the following polynomial $x^3 + 3x^2 + x - 5$

On entry to subroutine ZEROS N, the degree of the polynomial, = 3 and the A array of coefficients would have the values A(1) = 1.0, A(2) = 3.0, A(3) = 1.0, A(4) = -5.0

On exit the arguments would contain the following values

NREAL = 1

N = 3

ROOTS(1) = 1.0, ROOTS(2) = -2.0, ROOTS(3) = 1.0

ACHECK(1) = 1.0, ACHECK(2) = 3.0

ACHECK(3) = 1.0, ACHECK(4) = -5.0

This means that N = 3 roots were found, 1 of which was real. The real zero, stored in ROOTS(1) is 1.0, and the conjugate pair of complex zeros is ROOTS(2) ± jROOTS(3), that is -2.0 ± j1.0.

The check polynomial constructed from the computed zeros is $x^3 + 3x^2 + x - 5$ which is identical to the input polynomial.

The A array has been destroyed.

Signal Analysis Subroutine SA-16

1. IDENTIFICATION

Laurent expansion of a rational function, (FORTRAN)

Bell Telephone Laboratories

2. PURPOSE

A subroutine to determine the coefficients $h_k^{(n)}$, for $k < 0$, in the Laurent expansion of a rational function

$$H(s) = C\frac{a_1 s^N + a_2 s^{N-1} + \ldots\ldots + a_{N+1}}{(s-s_1)(s-s_2)\ \ldots\ (s-s_M)} = \sum_{\substack{k=-L \\ L \leq M}}^{\infty} h_k^{(n)} (s-s_n)^k$$

about each of its poles, $s_1, s_2, \ldots, s_n, \ldots, s_M$, given the values of the poles, the numerator coefficients, $a_1, \ldots, a_{N+1}$, and the scale factor, C, thus determining, more particularly the residue of H(s) at each of its poles.

3. USAGE

A. Calling sequence:

CALL RESDU (A,N,C,M,ALPHA,BETA,MULT,K,RR,RI,AM,AN)

B. Identification of input variables:

A, a linear array of the numerator coefficients of H(s), as above,

N, the degree of the numerator, as above

C, the scale factor of H(s), as above,

M, the degree of the denominator of H(s), as above,

ALPHA, a linear array of the real parts of the distinct poles of H(s),

BETA, a linear array of the imaginary parts of the distinct poles of H(s), that is, with reference to the above expression of H(s), s_n = ALPHA(n) ± jBETA(n),

MULT, a linear array of the multiplicities of the distinct poles of H(s), (FORTRAN fixed integers). Note: a conjugate pair of complex poles is to count as one distinct pole. That is, to enter the conjugate pair of poles $\alpha+j\beta$, $\alpha-j\beta$, enter ALPHA(n) = α, BETA(n) = β, MULT(n) = 1,

K, the number of distinct poles of H(s). The note above applies.

C. Identification of output variables:

RR, a two dimensional array of the real parts of the residues, (this array contains the information printed under R-RESIDUE),

RI, a two dimensional array of the imaginary parts of the residues, (this array contains the information printed under J-RESIDUES),

AM, a two dimensional array of the magnitudes of the residues, (this array contains the information printed under AMPLITUDE),

AN, a two dimensional array of the phase angles of the residues, (this array contains the information printed under ANGLE).

D. Restrictions:

For a real pole the multiplicity must be less than or equal to 8.

For a complex conjugate pair of poles the multiplicity must be equal to 1.

$K \leq 64$, i.e., the number of distinct poles cannot exceed 64.

$N < M$, i.e., the degree of the numerator of H(s) must be less than the degree of the denominator.

The arrays RR, RI, AM, AN, must all have row dimension 64, and column dimension at least as large as the highest

A(1) = 1.000 A(4) = 5.333

A(2) = 5.583 A(5) =-3.500

A(3) = 9.917 A(6) =-3.333

N = 5

C = 12.00

M = 6

ALPHA(1) = 0.0 BETA(1) = 0.0 MULT(1) = 1

ALPHA(2) = 1.0 BETA(2) = 0.0 MULT(2) = 1

ALPHA(3) =-2.0 BETA(3) = 0.0 MULT(3) = 2

ALPHA(4) = 1.0 BETA(4) = 1.0 MULT(4) = 1

K = 4

The following printout would be returned:

ROOT	ORDER	R-RESIDUE	J-RESIDUE	AMPLITUDE	ANGLE
1	1	5.0	0.0	5.0	0.0
2	1	4.0	0.0	4.0	0.0
3	1	3.0	0.0	3.0	0.0
3	2	1.0	0.0	1.0	0.0
4	1	1.0	2.0	2.236	63.43

indicating that

$$H(s) = \frac{5.0}{s} + \frac{4.0}{s-1} + \frac{3.0}{(s+2)^2} + \frac{1.0}{s+2} + \frac{1+2j}{s+j+1} + \frac{1-2j}{s+j-1}$$

The RR array, for example, would be returned as follows:

RR =

5.0	0.0
4.0	0.0
3.0	1.0
1.0	0.0

order pole of H(s).

E. Output:

This subroutine causes the arrays RR, RI, AM, AN to be printed out in the following way. The poles are assigned serial numbers and the pole with which each coefficient is associated is indicated by printing the serial number of that pole in the column headed ROOT. The term in the series about a particular pole with which each coefficient is associated is indicated by a number in the column headed ORDER, the lowest order term being the multiplier of the algebraically least power of $(s-s_n)$ in the series about that pole, (thus the highest "order" term is the residue of that pole). For each coefficient, the real part, the imaginary part, the magnitude, and the phase angle in degrees are printed in columns headed R-RESIDUE, J-RESIDUE, AMPLITUDE, and DEGREES, respectively.

The numbers printed by this subroutine under ROOT and ORDER are the row and column indices, respectively, of the RR, RI, AM, AN arrays in which the information on that line of printing will be found.

F. Comments:

SA-16 is a package of eight interlocking subroutines named: RESDU, POLYEV, FACTEV, POLY, NFAC, DERIV, COLL, CMPLX.

G. Example:

Suppose we wished to process

$$H(s) = 12.0\frac{s^5 + 5.583s^4 + 9.917s^3 + 5.333s^2 - 3.50s - 3.33}{s(s-1)(s+2)^2(s+1+j)(s+1-j)}$$

We would enter the following arguments to subroutine RESDU:

Signal Analysis Subroutine SA-17

1. IDENTIFICATION

Discrete orthonormal filter routine, (FORTRAN),
W. H. Huggins, T. Y. Young.

2. PURPOSE

The purpose of this subroutine is, given the p and q parameters defining a quadratic Kautz basis, and given sampled values of a signal, to calculate the coordinates on the given quadratic basis of that signal, either at some prescribed epoch, or at the calculated epoch when the projection of the input signal is a maximum. Provision is made to perform a linear transformation on these coordinates.

3. METHOD

The orthonormal filters are formed according to the prescription of Kautz as modified by Young to be orthonormal in the sampled data representation. See introduction for definition of the quadratic Kautz basis.

4. USAGE

A. Calling sequence:

CALL YOUNG(NSP,TINC,F,NH,B,FB,KEPOCH,EF,EFBC,M,G,FBG,FBC)

B. Identification of input variables:

NSP, the number of sample points of the input signal,

TINC, the time increment between successive sample points of the input signal,

F, a linear array of NSP sampled values of the input signal,

NH, one-half of the quadratic basis dimensionality,

B, a linear array of the quadratic basis parameters,

in order $p_1, q_1, \ldots, p_{NH}, q_{NH}$,

KEPOCH, index of the input signal at the epoch instant; to have the subroutine search out the instant at which the energy of the projection of the input signal on the quadratic basis is a maximum, enter KEPOCH=0,

M, a control parameter;
M = 0, for no output from the linear transformation,
M =-1, for no output from G and for no complementary signal,
Otherwise M equals the number of outputs of the linear operator G desired, that is, M = the column dimension of the G array,

G, a (2NH)x(M) matrix defining the auxiliary linear operator G.

C. Identification of output variables:

FB, a linear array of the coordinates of the input signal on the quadratic Kautz basis at the epoch instant,

KEPOCH, if KEPOCH = 0 is entered, then after exit from YOUNG, KEPOCH contains the calculated epoch index,
EF, the maximum energy of the projection of the input signal on the prescribed basis,

EFBC, the energy of the complementary signal at the epoch instant,

FBG, an (NSP)x(M) array containing the time sampled outputs of the auxiliary linear operator G,

FBC, linear array of sampled values of the complementary signal.

D. Options available:

To prevent initialization of the filters to zero state, enter NSP = -NSP.

To process the signal in reverse order, (that is to pro-

cess the array F(NSP),F(NSP-1),...,F(1)), enter
TINC = -TINC.
To prevent recomputation of the filter parameters, enter
NH = -NH.

E. Restrictions:

NH $\leq$ 10
NSP $\leq$ 500
G, if used, must have row dimension 20,
FBG, if used, must have row dimension 500.

F. Comments:

If no transformation operation is to be performed on the time sampled filter outputs, enter G as the unit matrix.

Note that a signal of any number of sample points may be processed by entering 500 points at a time, and on each call after the first, entering NSP = -NSP to prevent initialization of the filters.

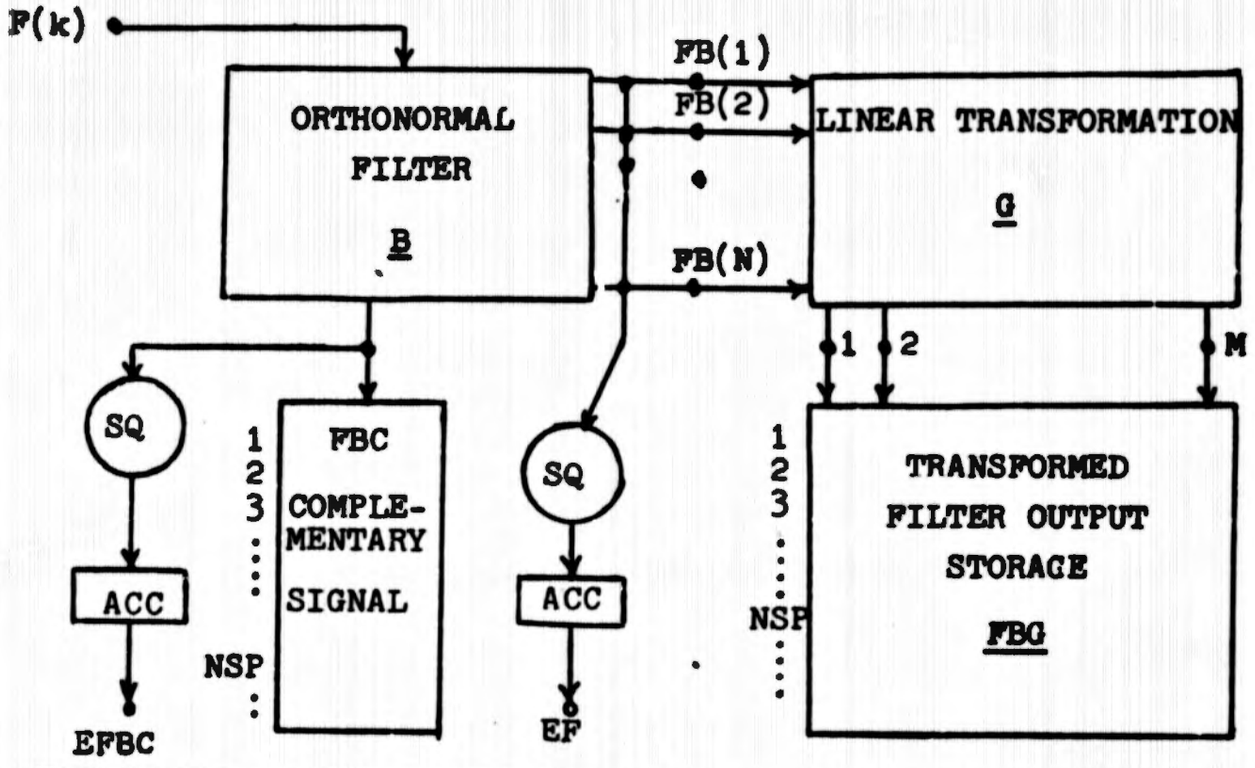
An output package, YGOUT, is available under the listing SA-17.1. This subroutine has the same argument list as YOUNG and causes the printout of all relevant information if called immediately after a call to YOUNG.

G. Example:

See the example following the write-up of SA-17.6.

H. Schematic:

The sketch on the following page is intended to clarify the inter-relationships of the program variables.



Signal Analysis Subroutine SA-17.1

1. IDENTIFICATION

Output package for SA-17, (FORTRAN), J. R. Sopko.

2. PURPOSE

A subroutine to cause the printout of the values of all input and output arguments of subroutine YOUNG, SA-17. Plots of the input signal and the complimentary signals are also made.

3. USAGE

A. Calling sequence:

CALL YGOUT(NSP,TINC,F,NH,B,FB,KEPOCH,EF,EFBC,M,G,FBG,FBC)

B. Operation:

Call YGOUT immediately after a call to YOUNG, and with the same argument list.

Signal Analysis Subroutine SA-17.5

1. IDENTIFICATION

Continuous orthonormal filter routine, (FORTRAN),
W. H. Huggins, J. R. Sopko.

2. PURPOSE

The purpose of this subroutine is, given the p and q parameters defining a set of orthonormal quadratic Kautz filters, to:

a. generate sampled values of the continuous unit impulse responses of the Kautz filters,

b. establish the coordinates on the quadratic Kautz basis of a continuous input signal, f(t), at some prescribed epoch, given sampled values of that signal,

c. establish the epoch at which the projection on the Kautz basis of a given input signal, f(t), is a maximum, and to measure the coordinates at this instant.

3. METHOD

This subroutine establishes all necessary linkages for the formulation, solution, and storage of the differential equations relevant to filtering with Kautz filters. The actual solution of the differential equations is accomplished using BCC Library routine 5.02.05, NOL1, which employs the Adams Moulton prediction-correction method. The user is referred to the BCC Library write-up of routine 5.02.05 for details concerning this technique.

4. USAGE

A. Calling sequence:

```
CALL KAUTZ(M,NSP,F,TINC,DELT,N,P,Q,YD,KEPOCH,FK,FC,EF,EFC)
```

B. Identification of input variables:

M, a control parameter;
M = 1, for purpose a.,
M = 0, for purpose b.,
M =-1, for purpose c.,

NSP, the number of sampled values in the input signal array, F, for the M = 0,-1 operations; for the M = +1 operation, NSP = the number of unit impulse response values to be generated for each filter,

F, a linear array of the sampled values of the signal to be filtered, (for the M = 1 operation enter a dummy argument),

TINC, a constant time interval between successive sample points of the F array; for the M = 1 operation, TINC = the time increment between successive computed values of the unit impulse responses,

DELT, a computational time increment used in the solution of the filtering equations, (see restrictions),

N, the dimensionality of the quadratic Kautz basis, that is, the number of filters to be used,

P, a linear array of the p quadratic basis parameters,

Q, a linear array of the q quadratic basis parameters,

KEPOCH, the index of the F array at the epoch instant; must be specified for the M = 0 operation, is calculated in the M = -1 operation, (for the M = 1 operation, enter a dummy variable).

C. Identification of output variables:

YD, a linear storage array of filter output, input arguments, etc., see comments for contents,

KEPOCH, the calculated index of the F array at the epoch instant of the input signal F, for the M = -1 operation,

FK, a linear array of coordinates of the input signal on

the quadratic Kautz basis at KEPOCH,

FC, a linear array of NSP sampled values of the complementary signal at intervals of TINC,

EF, the computed energy of the input signal at KEPOCH,

EFC, the energy of the complementary signal at KEPOCH.

D. Restrictions:

NSP $\leq$ 500, N $\leq$ 20, DELT $\leq$ TINC, DELT $\geq$ 0.010,

TINC/DELT = an integer,

For stability reasons, DELT should meet the following restriction with respect to the set of exponents s_i, from which the p and q parameters are generated:

DELT $\leq \frac{\pi}{10.0 \cdot \beta_{max}}$, where β_i is the imaginary part of

$s_i = \alpha_i + j\beta_i$.

For all calls to KAUTZ an array of dimension 11100 must be entered for YD.

E. Auxiliary subroutines used:

KAUTZ employs the following auailiary subroutines:
NOL1, BCC Library routine 5.02.05,
DERIV, required by NOL1,
TERM, required by NOL1,
OUT, required by NOL1.

A source deck for routines KAUTZ, DERIV, TERM, and OUT combined into a single package is available on the Signal Analysis Source Library tape. NOL1 is available under a separate listing on this same tape.

F. Comments:

KAUTZ assumes initial time is zero and runs until time TMAX = (NSP-1)*TINC in all modes of operation.

KAUTZ carries out a linear interpolation on the input signal when DELT < TINC.

An output package, KZOUT, is available for printing out all relevant variables of KAUTZ in an appropriate format. See SA-17.5.

The input array, F, is processed by KAUTZ in the order F(1),F(2),...,F(NSP). Thus any reversals of the input signal must be accomplished before calling KAUTZ.

See the introduction for definitions of the quadratic Kautz basis, p and q basis parameters, etc.

The YD array is alloted as follows:

YD(1) thru YD(20) - working space for NOL1
YD(21) thru YD(40) - P(1),Q(1),...,P(NH),Q(NH)
YD(41) = 0.0
YD(42) = DELT
YD(43) = TINC
YD(44) = TMAX
YD(45) = N
YD(46) = 1.0
YD(47) = NSP
YD(48) = M
YD(49) = a storage control parameter for NOL1, LI
YD(50) = the current value of the complementary signal
YD(51) thru YD(90) - normalizing constants $\sqrt{2p_1}$, $\sqrt{2p_1q_1}$,... ...,$\sqrt{2p_{NH}}$, $\sqrt{2p_{NH}q_{NH}}$.
YD(91) = the current value of EF
YD(92) = the current value of EFC
YD(93) = the current maximum value of EF-EFC
YD(94) = KEPOCH
YD(95) = EF at KEPOCH
YD(96) = EFC at KEPOCH
YD(97) thru YD(100) - nothing
YD(101) thru YD(600) - the input signal array, F

YD(601) thru YD(1100) - output of first Kautz filter
YD(1101) thru YD(1600) - output of the second Kautz filter
.
.
.
YD(10101) thru YD(10600) - output of twentieth Kautz filter
YD(10601) thru YD(11100) - the complementary error signal.

G. Example:

See example beginning on following page.

Signal Analysis Subroutine SA-17.6

1. IDENTIFICATION

Output package for SA-17.5, (FORTRAN), J. R. Sopko.

2. PURPOSE

A subroutine to cause the printout of the values of all input and output arguments of subroutine KAUTZ, SA-17.5. Where relevant, plots of the input signal and the complementary signals are made.

3. USAGE

A. Calling sequence:

CALL KZOUT(M,NSP,F,TINC,DELT,N,P,Q,YD,KEPOCH,FK,FC,EF,EFC)

B. Operation:

Call KZOUT immediately after a call to KAUTZ, and with the same argument list.

An example to illustrate a possible use of subroutines SA-17, and SA-17.6.

In Part I of this report series ("The Use of Orthgonalized Exponentials," by W. H. Huggins, page 53 et. seq.) the representation of a triangular pulse, implemented with analog equipment, was described. An orthonormalized exponential basis (the "simple Kautz basis" of the present report) was constructed from the exponentials $e^{-1.6t}$, $e^{-1.2t}$, $e^{-0.8t}$, $e^{-0.4t}$, and the representative of a triangular pulse on this basis was measured at various prescribed epochs. As an example of a possible use of subroutines KAUTZ and YOUNG we will carry out an equivalent analysis of a triangular pulse.

As in the example referenced, we will use an orthonormalized exponential basis constructed from the exponents $e^{-1.6t}$, $e^{-1.2t}$, $e^{-0.8t}$, $e^{-0.4t}$, the quadratic Kautz basis (see introduction for definitions of quadratic and simple Kautz bases). The components of this 4 dimensional quadratic Kautz basis are expressed in the frequency domain as

$$|\phi_1\rangle \sim \phi_1(s) = \frac{\sqrt{5.6}s}{s^2 + 2.8s + 1.92}$$

$$|\phi_2\rangle \sim \phi_2(s) = \frac{\sqrt{10.7}}{s^2 + 2.8s + 1.92}$$

$$|\phi_3\rangle \sim \phi_3(s) = \frac{\sqrt{2.40}s}{s^2 + 1.20s + 0.32}\left[\frac{s^2 - 2.8s + 1.92}{s^2 + 2.8s + 1.92}\right]$$

$$|\phi_4\rangle \sim \phi_4(s) = \frac{\sqrt{0.769}}{s^2 + 1.20s + 0.32}\left[\frac{s^2 - 2.8s + 1.92}{s^2 + 2.8s + 1.92}\right]$$

Figure I contains plots of these four quadratic Kautz basis components as functions of time. Each plot, generated by subroutine CPLOT, SA-31, is a continuous curve through 130 sampled values of the unit-impulse response of the corresponding

Kautz filter. In each plot the signal is normalized to its extreme values. These sets of sampled valued were generated by subroutine KAUTZ (SA-17.5), responding to the following call.

CALL KAUTZ(1, 130, DUM, T, DELT, 4, P, Q, YD, DUM, FK, FC, DUM, DUM)

where: T = DELT = 0.10 seconds

P(1) = $-s_1 - s_2$ = 2.80

Q(1) = $s_1 \cdot s_2$ = 1.92

P(2) = $-s_3 - s_4$ = 1.20

Q(2) = $s_3 \cdot s_4$ = 0.32

DUM = a dummy argument.

The sampled values of the filter unit-impulse responses are returned, stored in the YD array.

For comparison, Figure II contains similar plots of the four corresponding simple Kautz basis components. These components comprise the basis collection $F\{-1.6, 0.4, 4\}$ described and used in Part I of this report series. They are:

$$|F_1\rangle \sim \begin{cases} f_1(s) = \dfrac{\sqrt{2.8}}{s + 1.6} \\ f_1(t) = \dfrac{4.}{\sqrt{5.}}\left[e^{-1.6t}\right] \end{cases}$$

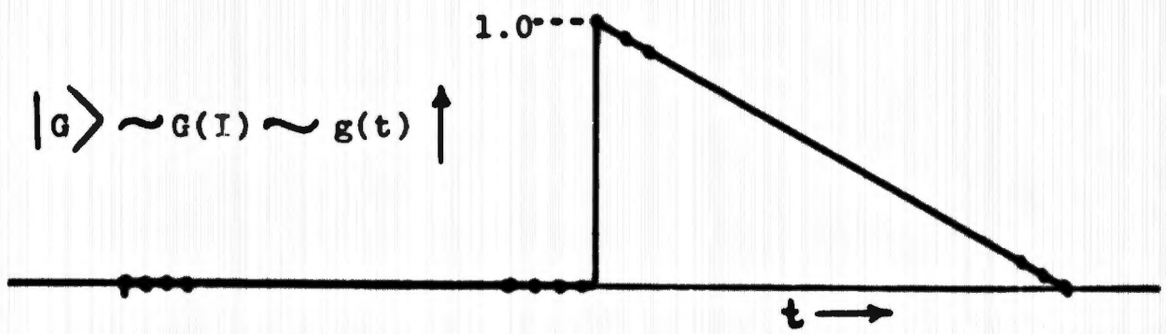
$$|F_2\rangle \sim \begin{cases} f_2(s) = \dfrac{\sqrt{2.4}}{s+1.2}\left[\dfrac{s - 1.6}{s + 1.6}\right] \\ f_2(t) = 2.\sqrt{\dfrac{3.}{5.}}\left[-8.0e^{-1.6t} + 7.0e^{-1.2t}\right] \end{cases}$$

$$|F_3\rangle \sim \begin{cases} F_3(s) = \dfrac{\sqrt{1.6}}{s + 0.8}\left[\dfrac{s - 1.6}{s + 1.6}\right]\left[\dfrac{s - 1.2}{s + 1.2}\right] \\ F_3(t) = 2.\sqrt{\dfrac{2.}{5.}}\left[28.0e^{-1.6t} - 42.0e^{-1.2t} + 15.0e^{-0.8t}\right] \end{cases}$$

$$|F_4\rangle \begin{cases} F_4(s) = \dfrac{\sqrt{0.8}}{s + 0.4}\left[\dfrac{s - 1.6}{s + 1.6}\right]\left[\dfrac{s - 1.2}{s + 1.2}\right]\left[\dfrac{s - 0.8}{s + 0.8}\right] \\ F_4(t) = \sqrt{\dfrac{4.}{5.}}\left[-56.0e^{-1.6t} + 105.0e^{-1.2t} - 60.0e^{-0.8t} + 10.0e^{-0.4t}\right] \end{cases}$$

We will analyze a pulse, g(t), which rises abruptly at its epoch, t_e, to a positive value, (1.0), then decays linearly, assuming the value 0.0 after 10. seconds. The pulse will be represented by an array of 200 sample points, stored in the array G, separated by equal time increments, T = 0.10 seconds. The actual epoch index of the signal is 100, corresponding to the epoch time t_e = 10.0 seconds.

Diagram:



t=0.1, 0.2,... ...,t=10.0=t_e,... ...,t=20.0

I=1, 2, ...,I=100,.... ...,I=200

Where t is time in seconds; I is the index of the G array.

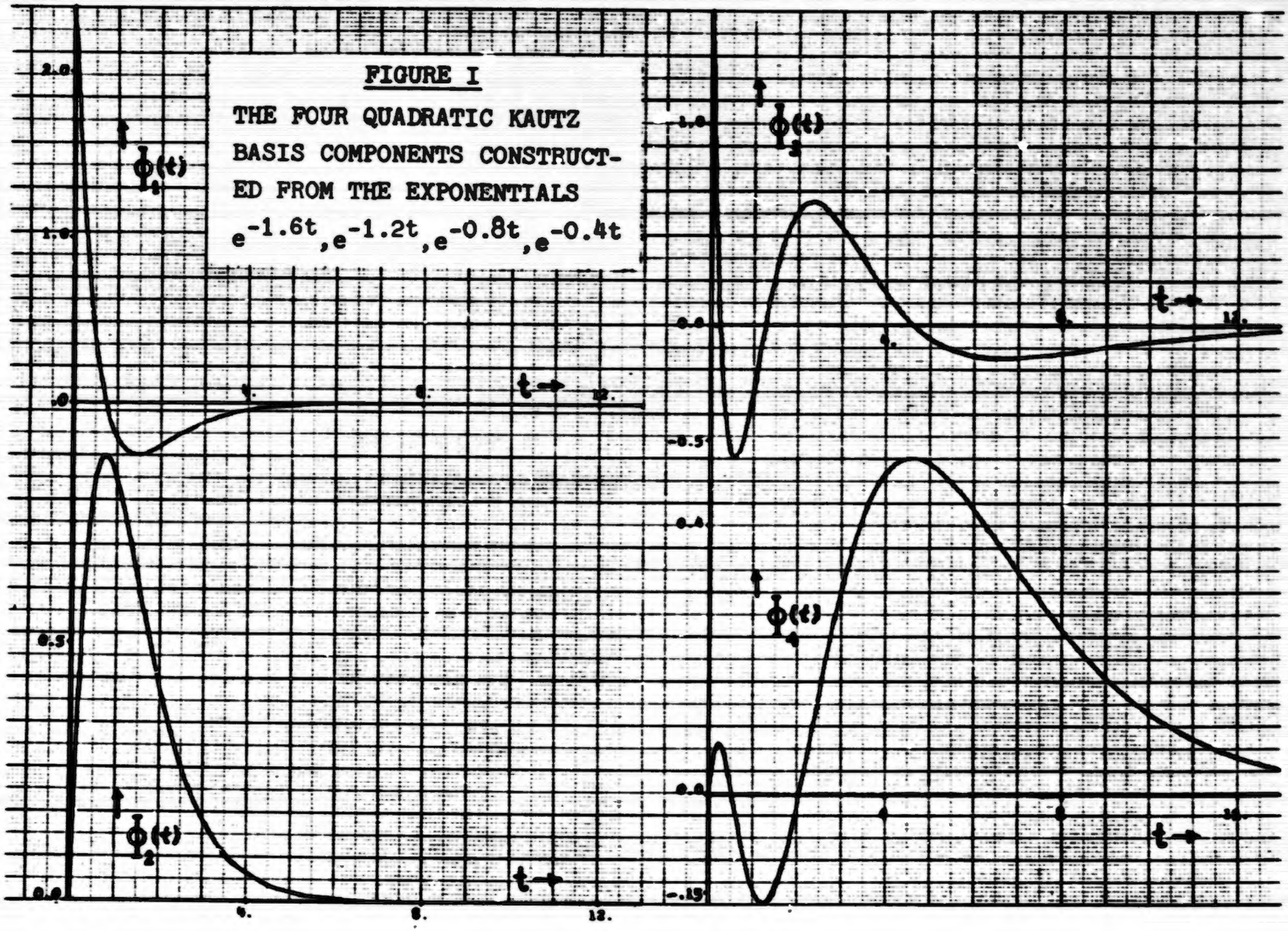


FIGURE I
THE FOUR QUADRATIC KAUTZ BASIS COMPONENTS CONSTRUCTED FROM THE EXPONENTIALS $e^{-1.6t}, e^{-1.2t}, e^{-0.8t}, e^{-0.4t}$
$\phi_1(t)$
$\phi_2(t)$
$\phi_3(t)$
$\phi_4(t)$
t →
2.0
1.0
.5
0.0
-.5
-.15
4.
8.
12.
16.

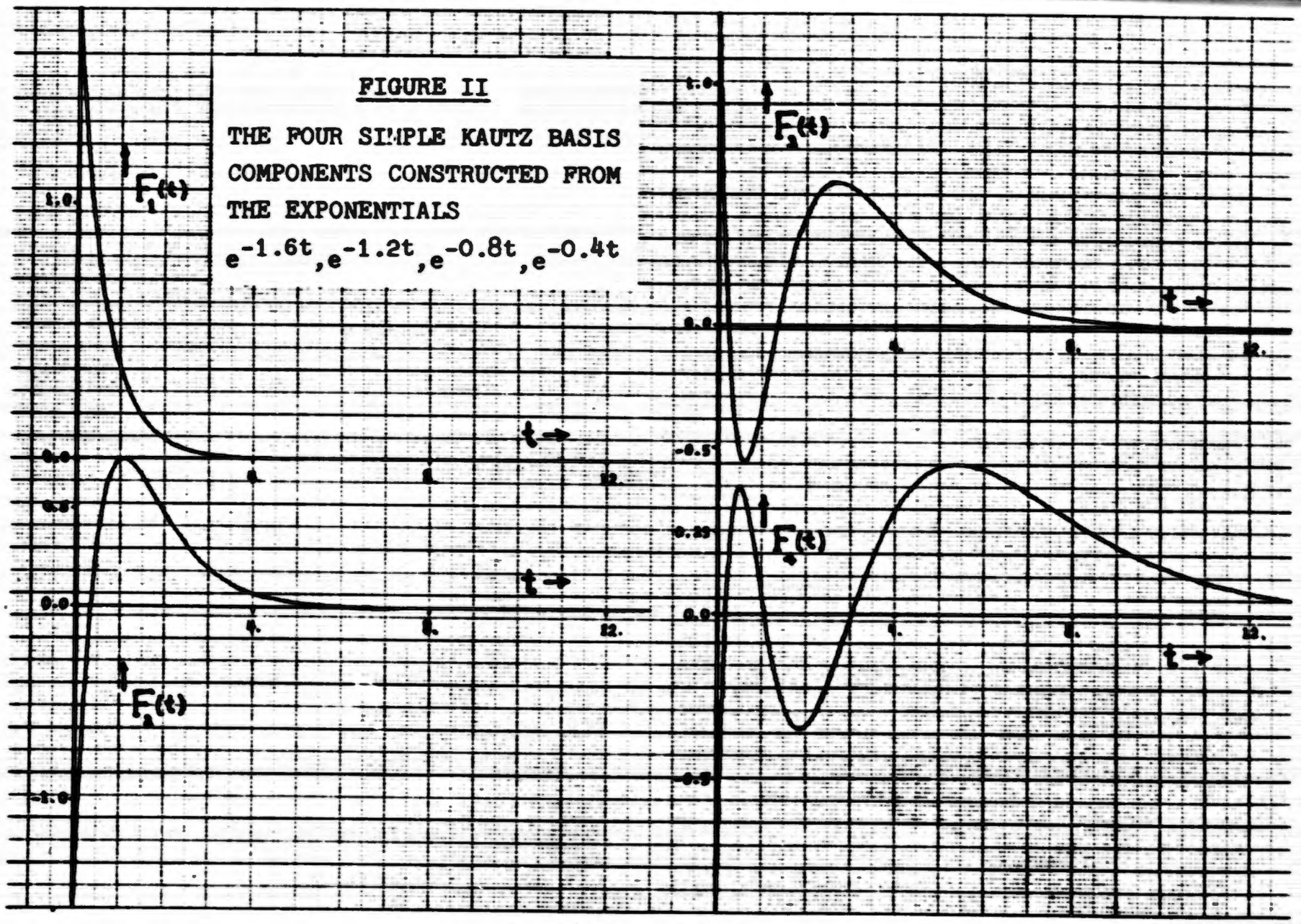


FIGURE II
THE FOUR SIMPLE KAUTZ BASIS COMPONENTS CONSTRUCTED FROM THE EXPONENTIALS
$e^{-1.6t}, e^{-1.2t}, e^{-0.8t}, e^{-0.4t}$
$F_1(t)$
$F_2(t)$
$F_3(t)$
$F_4(t)$
t →
1.0
0.0
-1.0
-0.5
4.
8.
12.

We wish to determine the optimum representative (under the criterion of least mean squared error) of this signal $|G\rangle$ on the <u>prescribed</u> four dimensional quadratic Kautz basis. We use subroutine YOUNG, setting KEPOCH = 0, initially, to indicate that the subroutine is to set KEPOCH equal to the index of the G array which corresponds to that instant when the energy of the projection of $|G\rangle$ on the prescribed basis is a maximum. To carry out the filtering process correctly we must reverse $|G\rangle$ about t = 0 (see introduction). That is, the sampled values representing $|G\rangle$ must be input to the YOUNG filter network in reverse order, viz: G(200), G(199), ..., G(1). Setting TINC = -T = -0.10 in the calling sequence of YOUNG accomplishes this. The following call to subroutine YOUNG is made:

```
TINC = -0.10
NH = 2
M = -1
KEPOCH = 0
B(1) = P(1)
B(2) = Q(1)
B(3) = P(2)
B(4) = Q(2)
CALL YOUNG(200, TINC, G, NH, B, FB, KEPOCH, EF, EFBC, M, DUM,
                                                  DUM, FBC).
```

Upon return KEPOCH = 100; thus, the subroutine has correctly determined the natural signal epoch. The representative of $|G\rangle$ at this epoch, stored in the FB array, is

$$\underline{G}\rangle = \begin{bmatrix} 0.12355 \\ 1.46808 \\ 0.42857 \\ 1.00562 \end{bmatrix}$$

The energy of this projection is EF = 3.383500

Figure III is a plot of the actual traingular pulse, $|G\rangle$, together with the approximation determined by subroutine YOUNG. This approximation is

$$g(t) \approx g'(t) = 0.12355\,\phi_1(t) + 1.46805\,\phi_2(t) + 0.42857\,\phi_3(t) + 1.00562\,\phi_4(t)$$

To demonstrate the effect of choosing an epoch for the representation of $|G\rangle$ which differs from its natural epoch, KEPOCH was specified as 62, and the above call to YOUNG was repeated. The representative returned was

$$\underline{G}\rangle = \begin{bmatrix} -0.04446 \\ 0.05500 \\ -0.40401 \\ 1.64704 \end{bmatrix}$$

The energy of this projection is 2.73133. In Figure IV, the approximation generated from this representative is plotted, superimposed upon the original pulse $|G\rangle$. As expected this inaccurate choice of signal epoch (KEPOCH = 62 $\sim$ t = 6.2 seconds = t_e - 3.8 seconds) leads to a much poorer representation of the triangular pulse.

Figure III, The signal $|G\rangle$ together with the approximation when KEPOCH = 100 $\sim t_e$.

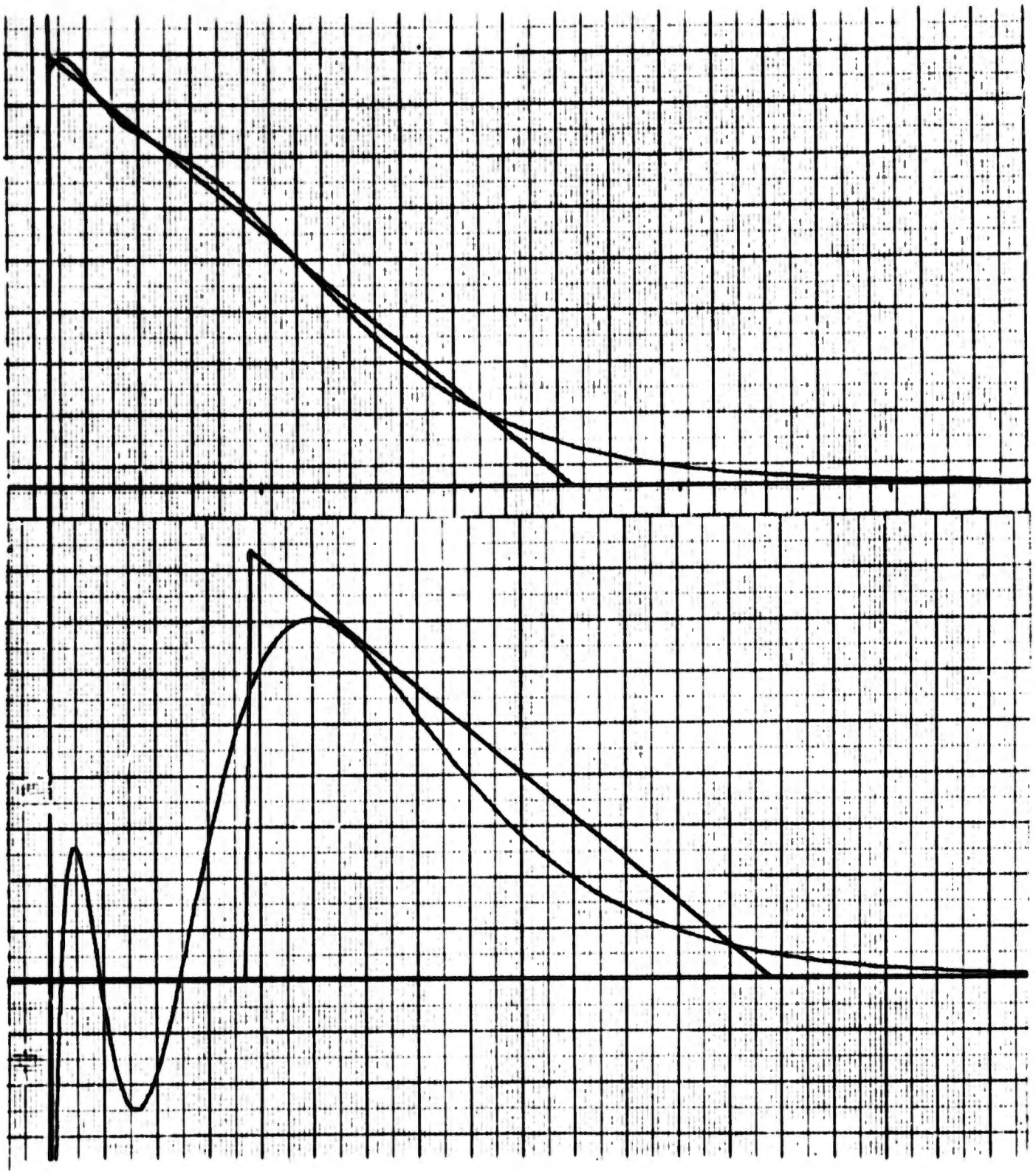


Figure IV, The signal $|G\rangle$ together with the approximation when KEPOCH is set = 62 $\sim t_e$ - 3.8 seconds.

Signal Analysis Subroutine SA-18

1. IDENTIFICATION

Inner product of two signals, each specified by rational functions in the frequency domain, (FORTRAN)

R. N. McDonough, G. M. Odell

2. PURPOSE

A subroutine to determine the inner product of two forward signals $|F\rangle$ and $|G\rangle$ having frequency domain representation:

$$F(s) = \frac{FN_1 s^K + FN_2 s^{K-1} + \ldots + FN_{K+1}}{FD_1 s^L + FD_2 s^{L-1} + \ldots + FD_{L+1}}$$

$$G(s) = \frac{GN_1 s^M + GN_2 s^{M-1} + \ldots + GN_{M+1}}{GD_1 s^N + GD_2 s^{N-1} + \ldots + GD_{N+1}}$$

3. METHOD

The inner product is calculated as the sum of the residues at the left-half-plane poles of either $F(-s)G(s)$ or $F(s)G(-s)$; $F(-s)G(s)$ if $F(s)$ has the greatest number of left-half-plane poles, $F(s)G(-s)$ otherwise.

4. USAGE

A. Calling sequence:

CALL INPRO (FN, K, FD, L, GN, M, GD, N, PRO, IFAULT)

B. Identification of input variables:

FN, FD, GN, GD, arrays of polynomial coefficients as above.
K, L, M, N, polynomial degrees, as defined under "Purpose."

C. Identification of output variables:

PRO, the inner product of $|F\rangle$ and $|G\rangle$, i.e. either $\langle F | G\rangle$

or $\langle \tilde{G} \mid F \rangle$ depending on which of F(s) and G(s) has the most left-half-plane poles.

IFAULT, a fault indicator; if routine fails to operate IFAULT = -1 and no results are returned, otherwise IFAULT = 0.

D. Restrictions:

K, L, M, N $\leq$ 20,

K+M < L+N,

No multiple poles in F(s) or G(s),

No poles common to both F(s) and G(-s),

No j-axis poles in F(s) or G(s).

E. Comments:

The following subroutines are required: SA-14, ZEROS - SA-16, RESDU - BCC 4.01.03, POLMUL.

The residues are determined by subroutine RESDU, which causes the printout of a table of information concerning the residues. In general this printout may be ignored.

F. Example:

Suppose we wished to know the inner product of the following two signals expressed in the frequency domain:

$$F(s) = \frac{s-1}{s^2+3s+2} = \frac{s-1}{(s+1)(s+2)}$$

$$G(s) = \frac{s^2-3s+2}{s^3+6s^2+11s+6} = \frac{(s-1)\ (s-2)}{(s+1)\ (s+2)\ (s+3)}$$

Upon entry to INPRO the FN, FD, GN, and GD arrays would contain the following values:

FN(1) = 1.0, FN(2) = -1.0

FD(1) = 1.0, FD(2) = 3.0, FD(3) = 2.0

GN(1) = 1.0, GN(2) = -3.0, GN(3) = 2.0,

GD(1) = 1.0, GD(2) = 6.0, GD(3) = 11.0, GD(4) = 6.0

And the polynomial degrees would be entered as follows:

K=1, L=2, M=2, N=3

After exit from INPRO, IFAULT = 0 would indicate that no fault had occurred, and PRO would equal 0.0, which is the inner product of the above two signals.

Note that in this case, the inner product would be evaluated as the sum of the residues at the left-half-plane poles of F(s) G(-s) since G(s) has the most left-half-plane poles.

Signal Analysis Subroutine SA-19

1. IDENTIFICATION

Discrete Prony polynomial roots, (FORTRAN),

R. N. McDonough

2. PURPOSE

A subroutine to determine a set of N exponentials for the representation of a sampled signal using the overdetermined discrete form of Prony's method.

3. METHOD

Given the NSP numbers, F_1, ..., F_{NSP}, this routine determines the N solutions, X_1, X_2, ..., X_N of

$$\sum_{i=1}^{N+1} \alpha_i X^{i-1} = 0, \quad \alpha_{N+1} = 1, \quad \text{where } \alpha_1, \ldots, \alpha_N \text{ are}$$

such that the equations

$$\sum_{i=1}^{N+1} F(K+i-1)\,\alpha_i = 0, \quad \alpha_{N+1} = 1, \quad K=1, \ldots \text{NSP-N}$$

are satisfied under the criterion of least squared error, that is, such that

$$\sum_{K=1}^{NSP-N} \left[\sum_{i=1}^{N+1} F(k+i-1)\alpha_i \right]^2 \quad \text{is a minumum.}$$

4. USAGE

A. Calling sequence:

CALL DISPRO(NSP,F,X,N,NR)

B. Identification of input variables:

NSP, the number of sampled signal values,

F, linear array of the sampled signal values,

N, the number of exponentials to be found.

C. Identification of output variables:

X, linear array of the exponentials, ordered in the same way as array RØØTS of subroutine SA-14, ZERØS,

NR, the number of real exponentials found.

D. Restrictions:

$N \leq 20$

$NSP \geq 2N$

E. Comments:

Note that the X_i are exponentials, not exponents; however, the X array is not the same as the XMØD array of subroutines SA-22 and SA-23. Subroutine SA-20, EXCØN will convert the X array of exponentials to the S array of exponents. Subroutine SA-20.5, CØNVRT will convert that S array to the XMØD array.

For a discussion of Prony's method, the user is referred to Representation and Analysis of Signals, Part XV, by R. N. McDonough.

Error returns: if the normal equations for the determination of the α_i are singular, NR = N + 1 and no results are returned. If the polynomial zeros routine fails to converge, returns NR negative and no results.

The following subroutines are required: SA-11, S; SA-14, ZERØS.

F. Example:

Suppose we were interested in representing the signal $f(t) = 2.0e^{-t} - 1.0e^{-2.0t} + 0.25e^{-0.1t}$ using a 2 dimensional exponential basis; that is, using a basis composed of

$$\{\Psi_i\}_1^2 = \{e^{+s_1 t}, e^{+s_2 t}\}$$

We wish to determine the two exponents s_1, s_2, matched to the signal f(t), in the Prony sense.

To accomplish this we might generate 10 sampled values of the signal f(t) separated by the time interval T = 1.0. This array of values would be entered as the F array; that is, we enter to DISPRØ

F(1)	= 0.8266	F(6)	= 0.1421
F(2)	= 0.4570	F(7)	= 0.1260
F(3)	= 0.2823	F(8)	= 0.1130
F(4)	= 0.2039	F(9)	= 0.1019
F(5)	= 0.1651	F(10)	= 0.0921

We would enter NSP = 10, and N = 2.

Upon return from LISPRØ NR = 2 indicates that no fault has occurred and that the exponents s_1, s_2, are real. The X array would contain

$$X(1) = 0.93009$$
$$X(2) = 0.44923$$

The above X(1) and X(2) are exponentials, that is $X(1) = e^{s_1 T}$, $X(2) = e^{s_2 T}$, where T is the sampling interval = 1.0. It is easily found, perhaps using SA-20, EXCØN, that

$$s_1 = -0.07246$$
$$s_2 = -0.80020$$

Thus the two dimensional exponential basis matched to the given signal f(t) in the Prony sense consists of the functions, $\psi_1 = e^{-0.07246t}$, $\psi_2 = e^{-0.80020t}$

Signal Analysis Subroutine SA-19.5

1. IDENTIFICATION

Prony approximation package (FORTRAN).

J. R. Sopko

2. PURPOSE

This subroutine is essentially a coordinator of several other subroutines described in this library. The purpose of this routine is, given sampled values of a signal, f(t), to:

Calculate the Prony exponents $\{s_i\}_1^N$ and least square coefficients $\{a_i\}_1^N$ relevant to the approximation

$$f(t) \approx \sum_{i=1}^{N} a_i e^{+s_i t}$$

Perform an error analysis with respect to this approximation, print out all relevant data in an appropriate format, and provide plots of the input, approximation, and error signals (if so desired).

3. METHOD

To accomplish the above purpose this subroutine makes use of the following auxiliary subroutines, also described in this library:

SA-19 DISPRO

calculates the Prony exponentials using the over-determined discrete form of Prony's method,

SA-20 EXCON, and SA-20.5 CONVRT

calculates the Prony exponents from the Prony exponentials; CONVRT calculates the XMOD array for SA-22, 23,

SA-22 COMPO

calculates the least square coefficients relevant to the Prony exponents,

SA-23 EXSGEN

generates the approximation,

SA-24 ERROR

performs the error analysis,

SA-26 APRINT (modified)

prints out arrays of complex numbers in appropriate format,

SA-32 PPLOT

prints out plots of input, approximation, and error signals,

BCC routines 1.04.06 and 1.04.061 calcomp plotter routines for the same purpose as PPLOT.

For specific details concerning the method of each auxiliary subroutine, the user is referred to the individual write-ups included in this library, and the BCC files.

4. USAGE

A. Calling sequence:

CALL PRONY (NSP, F, TINC, N, KPLOT, S, A, PCERR)

B. Identification of input variables:

NSP, number of samples in the input data array F(I),

F(I), sampled values of input signal to be approximated,

TINC, time increment between successive samples in F(I),

N, number of exponents to be used in the approximation

KPLOT, plotting control parameter

KPLOT = 0 for no plots
KPLOT = 1 for PPLOT plot only
KPLOT = 2 for PPLOT and CALCOMP plots
KPLOT = 3 for CALCOMP plot only.

C. Identification of output variables:

S(I), array of the (negative) Prony exponents; i.e. $S(I) = +s_1$. (Note: this S array is the negative of the S array of Programs SA-1, 5, 6.)

A(I), array of least square coefficients,

PCERR, indicator of goodness of approximation (refer SA-24)

D. Restrictions:

$NSP \leq 500$, $N \leq 20$, $NSP \geq 2N$

If CALCOMP PLOTS are desired the F(I) array should be normalized to its maximum value and a card * CALCOMP PLOT LABEL IS UPRONY must be included after the JOB card.

E. Output:

This subroutine, as part of the regular output, provides printout of the following items:

All input data,

Prony exponents and least square coefficients,

sampled values of the approximation and error signals,

results of the error analysis,

plots of input, approximation, and error signals (if desired).

F. Comments:

This subroutine has a built-in feature to discard extraneous roots introduced by SA-19. This is accomplished by discarding any negative real exponentials output by SA-19 before proceeding in the program.

In addition to the subprograms listed under 3, METHOD, the user must supply routines SA-14, ZEROS (used by DISPRO) and SA-11, S (used by COMPO and DISPRO). Subroutines PRONY and APRINT (modified) have been combined into a

single package. BCC routines 1.04.06 and 1.04.061 are not written up in this report; however 1.04.06 is included on the source program tape.

G. Example:

In the example following routines SA-17 and SA-17.5, the representative of a triangular pulse signal $|G\rangle$ on a prescribed orthonormal exponential basis was measured using the YOUNG digital filter. The basis used was the four dimensional quadratic Kautz basis constructed from the four real exponents: $s_1 = -1.6$, $s_2 = -1.2$, $s_3 = -0.8$, $s_4 = -0.4$. In that example, these four exponents were given information, along with the sample points defining the signal $|G\rangle$. Often, for a well-behaved, exponential-like signal, subroutine PRONY (by using SA-19, DISPRO) can discover a given number of exponentials suitable to serve as basis components for the representation of the signal. Thus, to use subroutine PRONY to analyze a signal it is necessary only to define the signal by a number of equally sapced sampled values, and to select the dimensionality of the exponential basis onto which the signal is to be projected.

We will use a triangular pulse signal, $|G\rangle$, of the same form as the signal used in the example following the write-up of SA-17.5. This forward signal vanishes identically prior to $t = 0$, has the value 1.0 at $t = 0$, and decays linearly with increasing t, assuming the value 0.0 for all $t \geq 10.0$ seconds. We may represent this signal with say NSP = 29 sampled values, stored in an array G, made at $t = 0, 0.5, 1.0, 1.5, \ldots, 10.0, 10.5, \ldots, 14.0$. The sampling interval is TINC = 0.5. If we want to project $|G\rangle$ on a three-dimensional exponential basis, we set N = 3 and make the following call to subroutine PRONY:

```
CALL PRONY (NSP, G, TINC, N, KPLOT, S, A, PCERR)
```

where KPLOT = 1, indicates that only PPLOTS of the relevant signals are desired.

Subroutine PRONY provides a detailed printout of all relevant data, selected parts of which have been reproduced as Figure V. As can be seen from the figure, the three exponents, determined by SA-19, consist of one real exponent and one complex conjugate pair. They are

$$s_1 = -19.5506$$
$$s_{2,3} = -0.2164 \pm j\ 0.1542$$

These numbers are returned in the S array. The representative on the exponential basis characterized by the above exponents, labeled "PROJECTION COEFFICIENTS" in Figure V, is

$$\underline{G}\rangle = \begin{bmatrix} -0.0334 \\ 1.0334 \\ 0.8693 \end{bmatrix}$$

These numbers would be returned in the A array.

Figure VI is a reproduction of the plot generated by subroutine PPLOT (SA-32) of the input signal $|G\rangle$ together with the approximation constructed from the above representative. This approximation, written in the continuous time domain, is:

$$G = -0.0334\, e^{-19.5506t}$$
$$+ e^{-0.2164t}\left[1.033 \cos(0.1542t) + 0.8693 \sin(0.1542)\right]$$

In Figure VI, lines have been drawn, connecting the asterisks (*) representing sampled values of the approximation, and connecting the dots (·) representing sampled values of

FIGURE V - Prony approximation to triangular pulse using three exponents.

NUMBER OF EXPONENTIALS TO REPRESENT SIGNAL = 3

DATA CONTAINS 29 SAMPLE POINTS AT INTERVALS OF 0.5000 TIME UNITS

INPUT SIGNAL

0.10000E 01 0.95000E 00 0.90000E 00 0.85000E 00 0.80000E 00 0.75000E 00 0.70000E 00 0.65000E 00 0.60000E 00 0.55000E 00
0.50000E 00 0.45000E-00 0.40000E-00 0.35000E-00 0.30000E-00 0.25000E-00 0.20000E-00 0.15000E-00 0.10000E-00 0.50000E-01
0. 0. 0. 0. 0. 0. 0. 0. 0.

EXPONENTIALS (OUTPUT OF SA-19)

(0.56838275E-04)
(0.89476223E 00) (0.69136156E-01)

NUMBER OF REAL EXPONENTS = 1

EXPONENTIALS IN CORRECT FORMAT FOR SA-22 AND SA-23

(0.56638274E-04)
(0.89742924E 00) (0.77114408E-01)

EXPONENTS (OUTPUT OF SA-20)

(-0.19550601E 02)
(-0.21644201E-00) (0.15422381E-00)

PROJECTION COEFFICIENTS
(OUTPUT OF SA-22)

(-0.33436671E-01)
(0.10334347E 01) (0.85932409E 00)

APPROXIMATION AT INTERVALS OF 1

1.00000E 00 0.9d478E 00 0.92998E 00 0.87110E 00 0.809d7E 00 0.74772E 00 0.68560E 00 0.62507E 00 0.56624E 00 0.50976E 00
0.45642E-00 0.40611E-00 0.35916E-00 0.31566E-00 0.27561E-00 0.23879E-00 0.20571E-00 0.17564E-00 0.14884E-00 0.12454E-00
0.10315E-00 0.84295E-01 0.67770E-01 0.53387E-01 0.40956E-01 0.30245E-01 0.21229E-01 0.13591E-01 0.72234E-02

ENF = 2.050E 01 ERR = 1.509E-01 PCERR = 0.7359

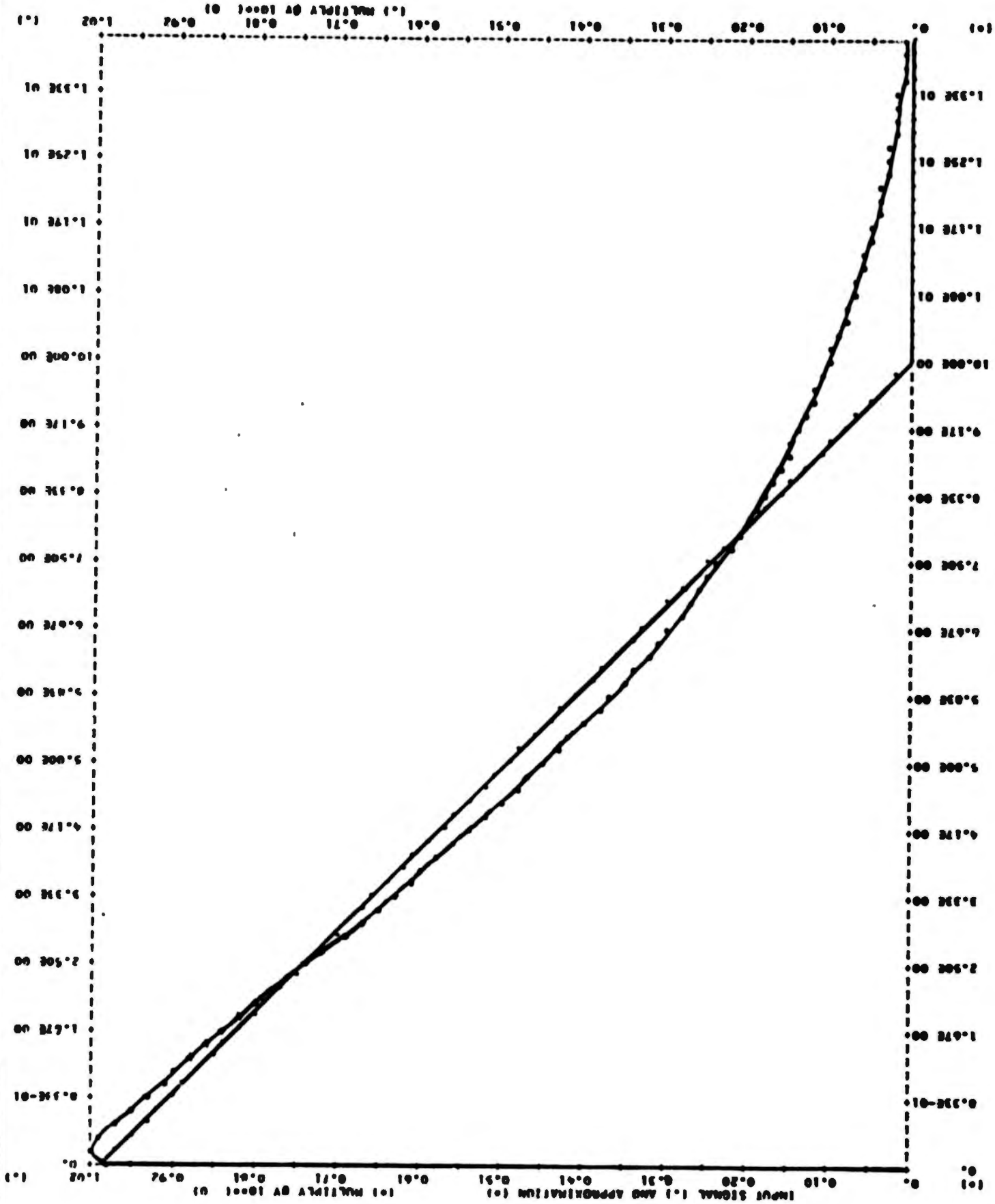


FIGURE VI - Triangular pulse together with 3 exponent approximation.

the triangular pulse $|G\rangle$. Note that the points of the plot are separated by time increments of TINC/3. for greater clarity of the plot. The approximation is PCERR = 0.7359 percent in error. (See write-up of SA-24 for definition of PCERR.)

For comparison purposes we next try to represent the signal $|G\rangle$ using a 2 dimensional exponential basis. We simply set N = 2, and repeat the previous call to subroutine PRONY. Figures VII and VIII contain the output for N = 2, corresponding to figures V and VI where N = 3. It will be noted that the complex pair of exponents chosen when N=2 closely resembles the pair chosen when N=3. While the error associated with the two-dimensional representative of the triangular pulse is somewhat larger than that associated with the three-dimensional representative, (0.8656 as compared to 0.7359) the plots of the approximations seem very nearly identical.

FIGURE VII - Prony approximation to triangular pulse using 2 exponents.

NUMBER OF EXPONENTIALS TO REPRESENT SIGNAL = 2

DATA CONTAINS 29 SAMPLE POINTS AT INTERVALS OF 0.5000 TIME UNITS

INPUT SIGNAL

0.10000E 01	0.95000E 00	0.90000E 00	0.85000E 00	0.80000E 00	0.75000E 00	0.70000E 00	0.65000E 00	0.60000E 00	0.55000E 00
0.50000E 00	0.45000E-00	0.40000E-00	0.35000E-00	0.30000E-00	0.25000E-00	0.20000E-00	0.15000E-00	0.10000E-00	0.50000E-01
0.	0.	0.	0.	0.	0.	0.	0.	0.	

EXPONENTIALS (OUTPUT OF SA-19)

(0.90000534E 00) (0.65505197E-01)

NUMBER OF REAL EXPONENTS = 0

EXPONENTIALS IN CORRECT FORMAT FOR SA-22 AND SA-23

(0.90238602E 00) (0.72655006E-01)

EXPONENTS (OUTPUT OF SA-20)

(-0.20542578E-00) (0.14531001E-00)

PROJECTION COEFFICIENTS (OUTPUT OF SA-22)

(0.10386112E 01) (0.73574129E 00)

APPROXIMATION AT INTERVALS OF T

0.10386E 01	0.98295E 00	0.92358E 00	0.86204E 00	0.79960E 00	0.73734E 00	0.67610E 00	0.61657E 00	0.55928E 00	0.50465E 00
0.45294E-00	0.40437E-00	0.35903E-00	0.31699E-00	0.27822E-00	0.24268E-00	0.21027E-00	0.18087E-00	0.15435E-00	0.13054E-00
0.10924E-00	0.90430E-01	0.73777E-01	0.59161E-01	0.46415E-01	0.35372E-01	0.25875E-01	0.17771E-01	0.10918E-01	

ENF = 2.0508E 01 ERR = 1.775E-01 PCERR = 0.8656

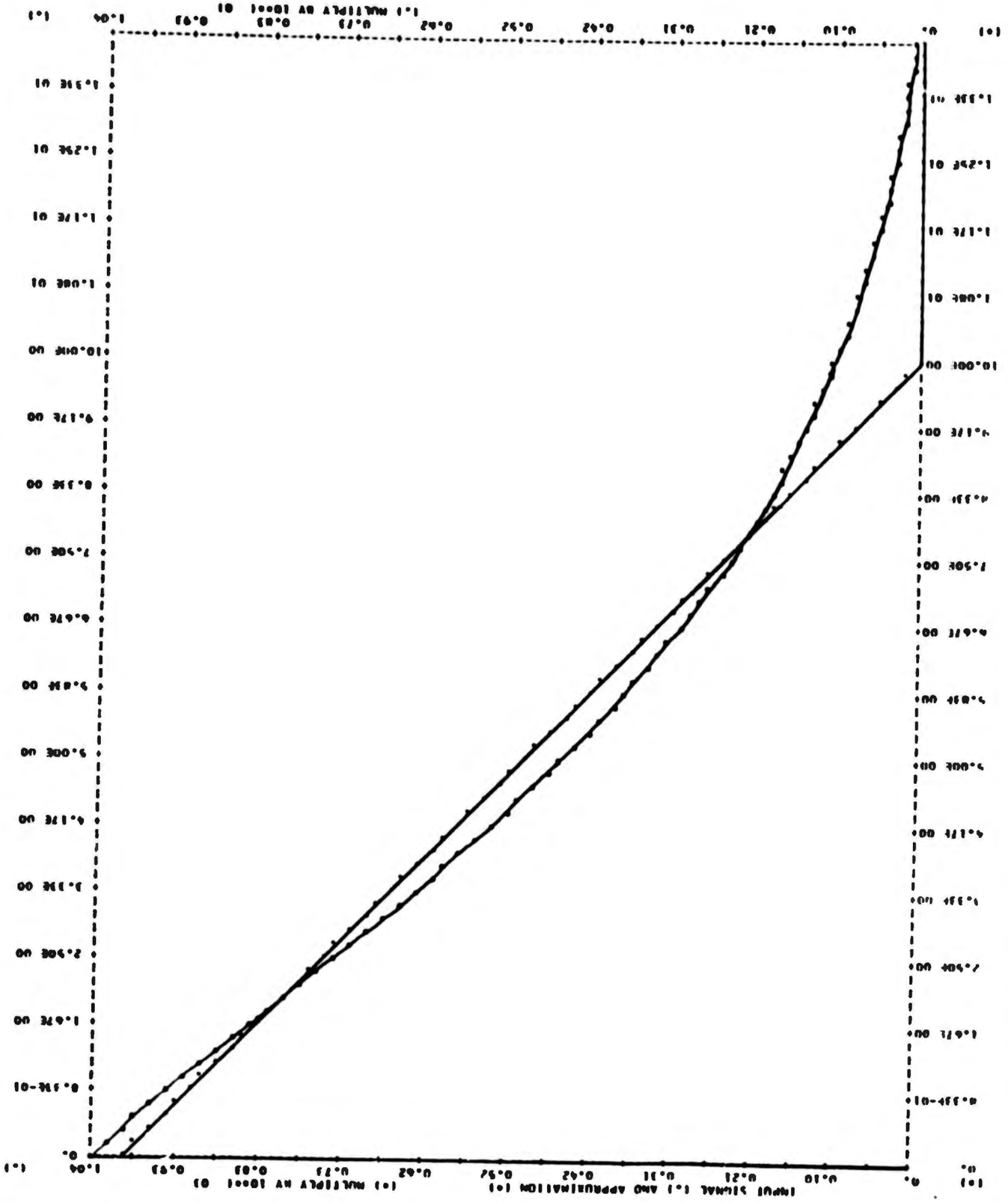


FIGURE VIII - triangular pulse together with 2 exponent approximation

Signal Analysis Subroutine SA-20

1. IDENTIFICATION

Calculation of exponent from value of exponential function at known time (FORTRAN), R. N. McDonough

2. PURPOSE

A subroutine to determine the real or complex exponents s_j corresponding to the given real or complex exponentials $x_j = e^{+s_j T}$, where T is a constant.

3. USAGE

A. Calling sequence:

CALL EXCON (X, N, NR, T, S, ITAG)

B. Identification of input variables:

X, linear array of the exponentials ordered such that

$X_1, \ldots, X_{NR}$ contain the real exponentials,

X_{NR+1} contains the real part of the first complex exponential,

X_{NR+2} contains the imaginary part of the first complex exponential,

⋮

X_{N-1} contains the real part of the last complex exponential,

X_N contains the imaginary part of the last complex exponential,

N, the number of values in the X array

NR, the number of real exponentials,

T, a constant interval of time, as defined above.

C. Identification of output variables:

S, linear array of computed negative exponents, ordered the same as the X array; i.e. $S(I) = +s_1$; (Note this S array is the negative of the S array of programs SA-1, SA-5, SA-6.)

ITAG, linear array of fault indicators; if no fault occurs ITAG(J) = 0, J = 1, ..., NR is returned. However, if some X_j is real and $X_j \leq 0$, there is no corresponding real exponent s_j and $s_j = 0$, $ITAG_j = -1$ is returned in this case.

D. Restrictions:

Dimension of ITAG must be at least NR in calling program.

E. Comments:

The X array above is not the same as the XMOD array of programs SA-22 and SA-23. Subroutine SA-20.5 will convert the S array into the XMOD array.

F. Example:

Suppose we had the following X array of exponentials, perhaps the output of SA-19, DISPRO:

$$.63210 = e^{s_1 T}$$

$$.66620 + j.711406 = e^{s_2 T}$$

$$.22557 + j.88256 = e^{s_3 T}$$

where $T = 1.0$

and we wished to determine s_1, s_2, and s_3.

Upon entry to subroutine EXCON the X array would contain

X(1) = .63210
X(2) = .66620, X(3) = .711406
X(4) = .22557, X(5) = .88256,

N would = 5, NR = 1, T = 1.0.

After exit from subroutine EXCON, ITAG(I) = 0, I = 1, 5, indicating that no improper exponentials were entered, and the S array would contain:

S(1) = -.45871
S(2) = -.25685
S(3) = .818199
S(4) = -.084552
S(5) = 1.28895

Thus the exponents have been determined to be:

s_1 = -.45871
s_2 = -.25685 + j0.818199
s_3 = -.084552+ j1.28895

Signal Analysis Subroutine SA-20.5

1. IDENTIFICATION

S to XMOD conversion, (FORTRAN), Brown, Sopko

2. PURPOSE

A subroutine to convert the S array of exponents of SA-20 to the XMOD array of subroutines SA-22 (COMPO) and SA-23 (EXSGEN).

3. USAGE

A. Calling sequence:

CALL CONVRT (S, XMOD, N, NR, T)

B. Identification of input variables:

S, array of (negative) exponents, ordered the same as array ROOTS of program SA-14, i.e.

S_1, ..., S_{NR} are the real exponents,

S_{NR+1} is real part of first complex conjugate pair of exponents,

S_{NR+2} is imaginary part of first complex conjugate pair of exponents, etc. ... ,

S_{N-1} is real part of last complex exponent,

S_N is imaginary part of last complex exponent.

This S array is the same as the S array of SA-20; but is the negative of the S arrays of programs SA-1, SA-5, SA-6.

N, number of exponents in S array, also number of exponentials in the XMOD array,

NR, number of real exponents in S array, also number of real exponentials in XMOD array,

T, arbitrary time increment as defined below.

C. Identification of output variables:

XMOD, array of exponentials ordered as in SA-22 and SA-23, i.e.

$$XMOD_1 = e^{S_1 T}, \ldots, XMOD_{NR} = e^{S_{NR} T},$$

$$XMOD_{NR+1} = e^{\alpha_1 T}, \; XMOD_{NR+2} = \beta_1 T, \ldots$$

$$XMOD_{N-1} = e^{\alpha_{Nc/2} T}, \; XMOD_N = \beta_{Nc/2} T.$$

where NC = N - NR; and α_1 = S(NR+1), B_1 = S(NR+2), etc.

D. Comments

Complex exponents must occur in conjugate pairs.

The X array of SA-19 may be transformed into the XMOD array of SA-22 and SA-23 by letting T = sampling interval of equally spaced input data in EXCON, SA-20, which changes the X array to the S array. Then SA-20.5 will change the S array to XMOD.

For usage with SA-23 to change the sampling interval of the generated signal from T to T', the XMOD array must be modified by putting T = T' in CONVRT, SA-20.5.

E. Example:

Suppose we wished to determine the XMOD array corresponding to the time increment T = 0.25 and the S array

$$s_1 = -1.0$$

$$s_{2,3} = -2.0 \pm j3.0$$

We would enter to CONVRT T = 0.25, N = 3, NR = 1, and

S(1) = -1.0

S(2) = -2.0 S(3) = 3.0

Upon return from CONVRT, the XMOD array would contain

XMOD(1) = $e^{(-1.0)(0.25)}$ = 0.77880

XMOD(2) = $e^{(-2.0)(.25)}$ = 0.60653

XMOD(3) = (3.0)(.25) = 0.7500

Signal Analysis Subroutine SA-22

1. IDENTIFICATION

Signal projection onto a general exponential basis (FORTRAN)

R. N. McDonough

2. PURPOSE

This subroutine determines the coordinates of a time sampled signal on a given exponential basis. The squared error is minimized over the finite point set.

3. METHOD

Given NSP equally spaced sampled values of a signal, $F_1, \ldots, F_{NSP}$, and N numbers, $XMOD_1, \ldots, XMOD_N$, this subroutine determines N numbers, $A_1, \ldots, A_N$, such that

$\sum_{i=1}^{NSP} (F_i - G_i)^2$ is minimum over the set $\{A_j\}_1^N$, where

$$G_{i+1} = \sum_{j=1}^{NR} A_j (XMOD_j)^i + \sum_{\substack{j=NR+1 \\ (by\ 2)}}^{N-1} (XMOD_j)^i \left[A_j \cos (i) XMOD_{j+1} + A_{j+1} \sin(i) XMOD_{j+1} \right]$$

where NR = number of real exponents characterizing the given basis. Here superscripts denote exponentiation.

We have here the finite discrete analog of the following approximation of a signal f(t):

$$f(t) \simeq \sum_{j=1}^{NR} a_j e^{+s_j t} + \sum_{j=1}^{\frac{NC}{2}} e^{\alpha_j t}\left[b_j \cos \beta_j t + c_j \sin \beta_j t \right]$$

where NC = Number of complex exponents in the given basis, and $s_i = +\alpha_i + j\beta_i$ for $i = NR + 1, \ldots, NC$

4. USAGE

A. Calling sequence:

CALL COMPO (NSP, F, XMOD, N, NR, A, IFAULT)

B. Identification of input variables:

NSP, number of time-samples in the F array,

F, array of signal time-samples where $F_i = f\left[(i - 1)T\right]$, $i = 1, \ldots, NSP$,

XMOD, array of N numbers which define the exponential basis as follows

$$XMOD_1 = e^{S_1 T}, \quad XMOD_{NR} = e^{S_{NR} T},$$

$$XMOD_{NR+1} = e^{\alpha_1 T}, \quad XMOD_{NR+2} = \beta_1 T, \ \ldots$$

$$\ldots, \quad XMOD_{N-1} = e^{\alpha_{NC/2} T}, \quad XMOD_N = \beta_{NC/2} T$$

where T = time increment between sample points,

N, dimensionality of the given exponential basis,

NR, number of real exponents in the given basis.

C. Identification of output variables:

A, array of N coefficients as defined under 3. ,

IFAULT, a fault indicator, where IFAULT = -1 is returned if the normal equations of the process are singular; IFAULT = 0 otherwise.

D. Restrictions:

$N \leq 20$, $NR \leq N$, $NSP \geq N$.

E. Comments:

Subroutine SA-11 is required.

No orthogonalization process is used. If $NSP \gg N$ the normal equations of the process may become unstable.

The XMOD array of SA-22 is not the same as the X array in SA-19 and SA-20. This XMOD array may be obtained from the S array of SA-20 through use of subroutine CONVRT (SA-20.5).

F. Example

Suppose we wished to project the signal $f(t) = -e^{-t}$ on the exponential basis characterized by the exponents:

$$S(1) = -0.2$$
$$S(2) = -1.0$$
$$S(3) = -2.0$$

And suppose we had available the following ten time samples of f(t):

F(1)	= -1.000	F(6)	= -0.286
F(2)	= -0.779	F(7)	= -0.263
F(3)	= -0.607	F(8)	= -0.174
F(4)	= -0.472	F(9)	= -0.135
F(5)	= -0.368	F(10)	= -0.105

The sampling interval here is T = 0.25.

First we use subroutine CONVRT (SA-20.5) to determine the XMOD array of exponentials corresponding to the above exponents. Here, also, the time increment is T = 0.25. That is, we use SA-20.5 to calculate

$$XMOD(1) = e^{(-0.2)(0.25)} = 0.95123$$
$$XMOD(2) = e^{(-1.0)(0.25)} = 0.77880$$
$$XMOD(3) = e^{(-2.0)(0.25)} = 0.60653$$

We then enter the above F and XMOD arrays to COMPO, along with NSP = 10, N = 3, and NR = 3.

Upon return from COMPO, IFAULT = 0, indicating that no fault has occurred, and the A array contains:

$$A(1) = 0.0$$
$$A(2) = -1.0$$
$$A(3) = 0.0$$

We have the obvious result that the signal represented by the given array of sample points may be represented on the given exponential basis as

$$f(t) = 0.0e^{-0.2t} - 1.0e^{-1.0t} + 0.0e^{-2.0t}$$

Signal Analysis Subroutine SA-23

1. IDENTIFICATION

Generation of time-samples of an exponential series, (FORTRAN)

R. N. McDonough

2. PURPOSE

A subroutine to generate equally spaced time-samples of the function

$$g(t) = \sum_{i=1}^{N} a_i e^{s_i t} .$$

3. METHOD

Given N numbers $XMOD_1$, ..., $XMOD_N$, and N numbers A_1, ..., A_N, this subroutine generates NSP numbers G_1, ..., G_{NSP} which are defined as follows:

$$G_{i+1} = \sum_{j=1}^{NR} A_j XMOD_j{}^{i}$$

$$+ \sum_{\substack{j=NR+1 \\ (\text{by } 2)}}^{N-1} XMOD_j{}^{i} \left[A_j \cos(i) XMOD_{j+1} + A_{j+1} \sin(i) XMOD_{j+1} \right]$$

where NR = the number of real exponents in the series. Here the superscripts denote exponentiation.

4. USAGE

A. Calling sequence:

CALL EXSGEN (XMOD, N, NR, A, G, NSP)

B. Identification of input variables:

XMOD, an array of N numbers which are defined as follows:

$$XMOD_1 = e^{S_1 T}, \ldots, XMOD_{NR} = e^{S_{NR} T},$$

$$XMOD_{NR+1} = e^{\alpha_1 T}, XMOD_{NR+2} = \beta_1 T \ldots$$

$$\ldots, XMODN_{-1} = e^{\alpha_{NC/2} T}, XMOD_N = \beta_{NC/2} T,$$

where: $s_i = \alpha_i + j\beta_i$; for $NR < i \leq NC$.

T = time increment between desired time samples,

NC = number of complex conjugate pairs of exponentials,

N, number of exponential components in the series, = NR + NC,

NR, number of real exponential components in the series,

A, array of N coefficients as defined in SA-22, COMPO,

NSP, number of sample points desired.

C. Identification of output variables:

G, array of NSP time-samples of the function g(t) as defined under 3. above.

D. Restrictions:

$N \leq 20$, $NR \leq N$

E. Comments:

All arguments are as defined in COMPO, SA-22.

The XMOD array is not the same as the X array of SA-19 and SA-20. This XMOD array may be obtained from the S array of SA-20 by means of subroutine CONVRT, SA-20.5.

C. Identification of output variables:

ENF, the approximation of the energy in signal F

$$= \sum_{i=1}^{NSP} F_i{}^2 T,$$

ERR, the approximation of the squared error

$$= \sum_{i=1}^{NSP} (F_i - G_i)^2 T,$$

PCERR, the percent error, = 100x(ERR/ENF).

D. Comments:

If ENF = 0, PCERR = -1 is returned.

E. Example

Suppose we have two signals F and G, each given as an array of 5 sampled values, and wish to know the squared error between them. For example, let the signals be given by the following arrays, where the time increment between sample points is T = .50.

F(1)	= 0.00	G(1)	= 0.00
F(2)	= 0.25	G(2)	= 0.30
F(3)	= 0.50	G(3)	= 0.60
F(4)	= 0.75	G(4)	= 0.90
F(5)	= 1.00	G(5)	= 1.20

To determine the squared error between F and G we would enter to ERROR the above F and G arrays, and NSP = 5, T = 0.50.

EXSGEN generates sampled values only at equidistant time intervals T, as specified in SA-20.5. To obtain sampled values at equidistant intervals T', the XMOD array should be obtained from SA-20.5 by setting T = T'.

G. Example:

Suppose we wished to generate 10 equally spaced sampled values, where the sampling interval is T = 0.25 units, of the signal

$$g(t) = 0.1e^{-0.2t} - 1.0e^{-1.0t} + 1.0e^{-2.0t}$$

We first use subroutine CONVRT (SA-20.5) to generate the XMOD array from the given exponents. SA-20.5 would provide, given T = 0.25, the numbers:

$$XMOD(1) = e^{+s_1 T} = e^{(-0.2)(0.25)} = 0.95123$$

$$XMOD(2) = e^{+s_2 T} = e^{(-1.0)(0.25)} = 0.77880$$

$$XMOD(3) = e^{+s_3 T} = e^{(-2.0)(0.25)} = 0.60653$$

Upon entry to subroutine EXSGEN, the XMOD array would be as above and we would also enter N = 3, NR = 3, NSP = 10, and A(1) = 0.1

A(2)= -1.0

A(3)= +1.0

After exit from EXSGEN, the G array would contain:

(Continued on the next page)

G(1)	= .100	G(6)	= -.128
G(2)	= -.167	G(7)	= -.099
G(3)	= -.149	G(8)	= -.074
G(4)	= -.163	G(9)	= -.050
G(5)	= -.151	G(10)	= -.030

The above numbers are the time-samples of g(t) as initially given, evaluated at equal time intervals of 0.25 units.

Signal Analysis Subroutine SA-24

1. IDENTIFICATION

Squared error between sampled signals,(FORTRAN), R. N. McDonough

2. PURPOSE

A subroutine to determine the squared error between two sampled signals, F(t) and G(t).

3. USAGE

A. Calling sequence:

CALL ERROR (F, G, NSP, T, ENF, ERR, PCERR)

B. Identification of input variables:

F, linear array of NSP sampled values of the first signal,

G, linear array of NSP sampled values of the second signal,

NSP, the number of sampled values in the F array, which must equal the number of sampled values in the G array,

T, the time increment between sampled values.

After exit from ERROR we would have:

ENF = 0.88750

ERR = 0.03750

PCERR = 4.2253

Signal Analysis Subroutine SA-25

1. IDENTIFICATION

Real simultaneous linear equation solver, (FORTRAN),

L. G. Kelly

2. PURPOSE

A subroutine to determine the solution, $\underline{x}$, of the set of real, linear, simultaneous equations: $\underline{B}\,\underline{x} = \underline{v}$.

3. USAGE

A. Calling sequence:

CALL CROSIM(A, N)

B. Identification of input variables:

A, an augmented matrix of which the first N columns contain the matrix $\underline{B}$ and the $N+1^{st}$ column contains $\underline{v}$,

N, the number of simultaneous equations to be solved.

C. Identification of output variables:

After exit from the routine the $N + 1^{st}$ column of the A array contains the solution $\underline{x}$.

D. Restrictions:

$N \leq 20$

The A array must have row dimension 20 in the calling program.

E. Comments:

The operation time for CROSIM varies approximately as N^3 so that if $N = 20$ the routine should take about 8 seconds.

F. Example:

Suppose we wished to solve the following set of equations

$$\begin{aligned} 1.0x_1 + 1.0x_2 + 1.0x_3 + 1.0x_4 &= 4.0 \\ 1.0x_1 + 3.0x_2 - 2.0x_3 + 1.0x_4 &= 3.0 \\ 1.0x_1 + 1.0x_2 - 1.0x_3 - 1.0x_4 &= 0.0 \\ 1.0x_1 - 1.0x_2 + 2.0x_3 - 1.0x_4 &= 1.0 \end{aligned}$$

We would enter to CROSIM $N = 4$, and the following A array

A =

1.0	1.0	1.0	1.0	4.0
1.0	3.0	-2.0	1.0	3.0
1.0	1.0	-1.0	-1.0	0.0
1.0	-1.0	2.0	-1.0	1.0

Upon exit from CROSIM the $N + 1^{st}$ column of the A array would be:

$$\begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

Therefore the solution desired is:

$x_1 = 1.0, \quad x_2 = 1.0, \quad x_3 = 1.0, \quad x_4 = 1.0$

Signal Analysis Subroutine SA-26

1. IDENTIFICATION

Real and/or complex number print-out (FORTRAN), R. N. McDonough

2. PURPOSE

A subroutine to print an array of real and/or complex numbers in an appropriate format.

3. USAGE

A. Calling sequence:

CALL ARRINT(F, N, NR)

B. Identification of input variables:

F, array of N numbers to be printed, with NR real numbers first then complex pairs,

N, number of numbers to be printed,

NR, number of real numbers in F array.

C. Output:

The numbers will be printed in E-format in the following arrangement:

$$F(1)$$
$$\vdots$$
$$F(NR)$$
$$F(NR+1) \pm j\, F(NR+2)$$
$$\vdots$$
$$F(N-1) \pm j\, F(N)$$

D. Restrictions

$NR \leq N$

E. Example:

Suppose we wished to print the following set of real and complex numbers

$$1.0 \pm j.5$$

$$1.0$$

$$2.0$$

$$-3.0 \pm j.01$$

To use subroutine APRINT we must arrange the array so that the real numbers precede the complex pairs, that is we would enter the numbers in the F array as follows:

F(1) = 1.0

F(2) = 2.0

F(3) = 1.0 F(4) = .50

F(5) = -3.0 F(6) = .01

In addition we would enter to APRINT, N = 6, NR = 2.

The above entries would cause the following printout:

(.10000000E+01)

(.20000000E+01)

(.10000000E+01) ± J(.50000000E+00)

(-.30000000E+01) ± J(.10000000E-01)

Signal Analysis Subroutine SA-30

1. IDENTIFICATION

One line Hollerith output subroutine, (FAP), J. Reeder.

2. PURPOSE

A subroutine to center and print one line of from 1 to 120 Hollerith characters.

3. USAGE

A. Calling sequence

CALL TITLE (1H-, 19HTITLE TO BE PRINTED)

B. Identification of input variables:

1H-, the first argument is 1H*, where * is any legal carriage control character. This argument determines how many lines are to be skipped before printing the line. For example,

*	# of lines skipped
0	skip 2 lines
-	skip 3 lines
+	stay on same line
blank	skip 1 line
1	new page

19HTITLE TO BE PRINTED, the second argument is the line to be printed. The Hollerith count (in the example 19H) must be correct for the given line, and may have any value from 1H to 120H.

C. Comments:

A title is automatically centered on the output sheet.

Signal Analysis Subroutine SA-31

1. IDENTIFICATION

Continuous plotter routine for discrete data points (FORTRAN-FAP), G. M. Odell.

2. PURPOSE

A subroutine to generate appropriately scaled continuous plots of one, two, or three signals, when given arrays of discrete sample points taken at equal time intervals.

3. METHOD

The routine generates data for BCC subroutine PL570 which writes a tape which controls the Calcomp plotting device. In order to avoid the use of internal storage arrays for independent variables, routine CPLOT generates and plots one pair of coordinates at a time.

4. USAGE

A. Calling sequence:

CALL CPLOT(6HULABEL, SIZE, NPLOTS, LETERS, ISCALE, Y1, NSP1, REF1, Y2, NSP2, REF2, Y3, NSP3, REF3)

B. Identification of input variables:

6HULABEL, a six-character Hollerith lable which associates the plot made with a given job. This argument is ignored on all but the first call to CPLOT, and on subsequent calls a dummy may be entered,

SIZE, a parameter specifying the total length in inches of the plot, (floating point),

NPLOTS, a parameter to control tne number of plots in the present picture, that is:

NPLOTS=1, for one signal to be plotted, and normalized to fill the entire plot,

NPLOTS=2, for two signals to be plotted, the first, Y1, on the top half of the paper; the second, Y2, on the bottom half,

NPLOTS=3, for three signals to be plotted, Y1 in the top third, Y2 in the middle third, Y3 in the bottom third,

NPLOTS=4, for two signals to be plotted with the entire page available for each plot,

LETERS, the number of letters in the longest title to be written on the present picture. If no title is to be written on the present picture, enter LETERS $\leq$ 0, and the picture will be terminated by CPLOT. If LETERS $>$ 0, subroutine WRITE must be used.

ISCALE, a parameter to control the scaling of the signals, that is,

ISCALE=1, for separate X and Y scales for all plots in the picture,

ISCALE=2, for common X and separate Y scales for all plots in the picture,

ISCALE=3, for common X and common Y scales for all plots in the picture,

ISCALE=4, for separate X and common Y scales for all plots in the picture,
(Note: 'separate' scales means each signal is normalized to fill the entire region of the paper allotted to it, and 'common' scales means the first signal is normalized and all other signals are plotted on this normalized scale. Obviously then, if two or three plots are to have a common X or Y scale, the signal entered as Y1 must have the maximum X or Y value of the common scale.)

Y1, linear array of equally spaced sample points for the first signal,

NSP1, the number of sample points in the Y1 array,

REF1, the value of Y1 at which a horizontal reference line is to be drawn for the Y1 signal,

Y2,NSP2,REF2,Y3,NSP3,REF3, these variables refer to the second and third signals respectively, each variable having the same definition as the corresponding variable for signal 1.

C. Option:

Normally some number included in the range of Y values is entered for REF, however, if no horizontal axis is desired for a particular signal, enter REF as less than the minimum value of that Y array.

To cause the horizontal axis to be centered between the maximum and minimum values of the Y array, enter REF as greater than the maximum value of that Y array.

D. Restrictions:

There is no limit either to the length of a plot or to the number of sample points to be plotted.

E. Availability:

Both CPLOT and the required PL570 are available on the Signal Analysis library source tape.

F. Comments:

Plotted signals may be individually labeled by using subroutine WRITE immediately after calling CPLOT, if the number of letters in the longest title to be written has been entered as argument LETERS. In any case Subroutine WRITE, SA 31.5 is required by this routine.

Each picture must be ended before another can begin. The two ways of doing this are: i) to enter LETERS ≤ 0 in which case CPLOT ends the picture, and no labels may be written on that picture, or ii) to make the proper call to subroutine WRITE after the last title has been written.

After the final call to either CPLOT or WRITE in a given job the user must make the following call to PL570:

```
CALL PL570(2,0,0,0,0,0,0,0,0,0,0,0)
```

This call is necessary to separate the plots made by different jobs.

Note that each Hollerith character takes .15 inches, hence the length of the plot and labels combined will be SIZE + (.15)*(LETERS).

An asterisk card containing the following information must be included after the XEQ card:

```
*       CALCOMP PLOT LABEL IS ULABEL
```

where ULABEL is the first argument of the calling sequence.

This program is a package of six subroutines: CPLOT, CCPLOT, YPLOT, REFPLT, PL570, and WRITE.

The subroutine calling sequence may have a variable length argument list. That is, if only one signal is to be plotted the arguments Y2,NSP2,REF2,Y3,NSP3,REF3 may be altogether omitted and the calling sequence will have only eight arguments. If only two signals are to be plotted Y3,NSP3, and REF3 may be omitted.

G. Space required:

CPLOT - 82
CCPLOT - 521
YPLOT - 82
REFPLT - 90
WRITE - 167
PL570 - 2145
3050 (decimal)

H. Example:

See example included in the write-up of SA-31.5, WRITE.

Signal Analysis Subroutine SA-31.5

1. IDENTIFICATION

Label routine for continuous plotter subroutine CPLOT, (FORTRAN), G. M. Odell.

2. PURPOSE

A subroutine to label plots generated by routine CPLOT, SA-31. This subroutine may be used only in conjunction with CPLOT, and only after a call to CPLOT has been made.

3. USAGE

A. Calling sequence:

```
CALL WRITE(NFUNCT,NCHAR,19HTITLE TO BE WRITTEN)
```

B. Identification of input variables:

NFUNCT, the number of the signal with which the label is to be associated, =1,2, or 3 (for example, NFUNCT = 2 will cause the lable to be written in front of the plot of the Y2 array of the last call to subroutine CPLOT),

NCHAR, the number of Hollerith characters in the label,

19HTITLE TO BE WRITTEN, the label to be written; the Hollerith count must be correct.

C. Output:

The labels will be written as a single string of letters immediately preceeding the signal number NFUNCT. The space per letter is 0.15 inches, hence the label is NCHAR*0.15 inches long.

D. Restrictions:

A label may have any number of letters as long as the number of letters in the longest title of a given picture has been specified as argument LETERS of subroutine CPLOT.

Since a maximum of 3 signals can be plotted by one call to CPLOT, a maximum of 3 labels can be written by subroutine WRITE on any particular picture, each being written by a different call.

Each picture must be ended before another can begin, as explained in the CPLOT write-up. If WRITE is used the following call must be made after the last label of a picture is written:

```
CALL WRITE(DUM,DUM,6HENDPIC)
```

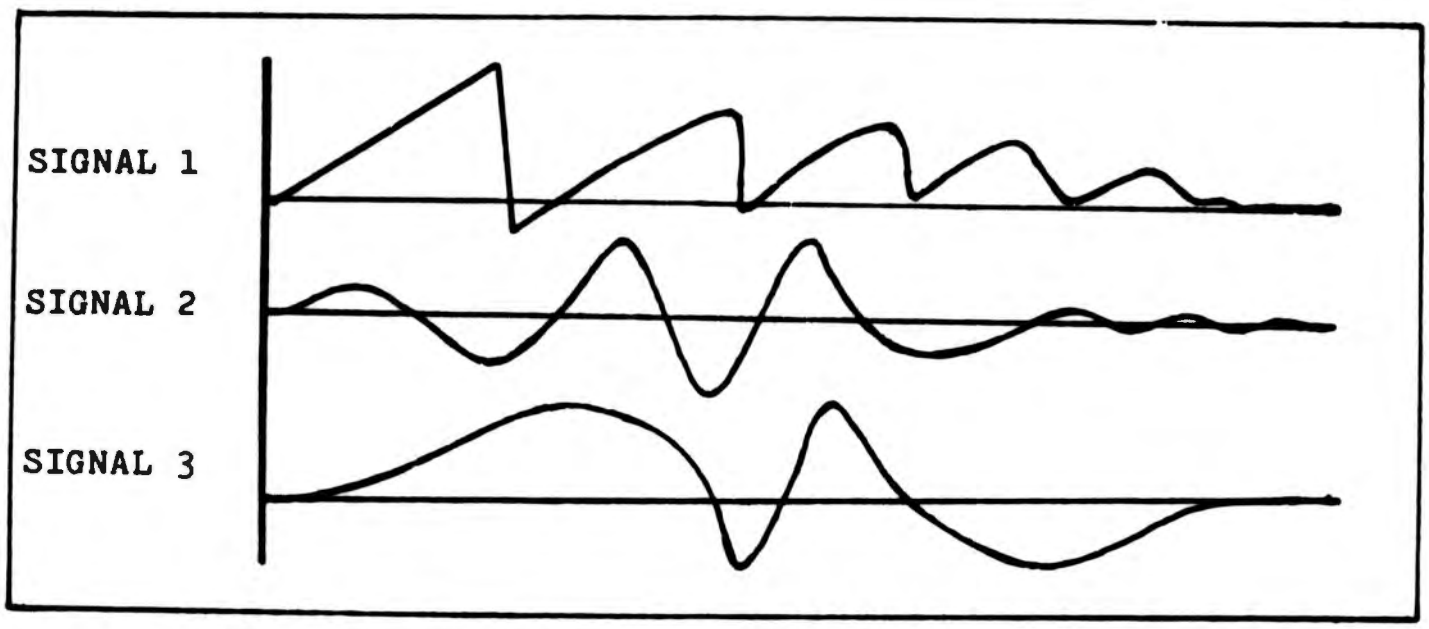
E. Example:

Suppose we wished to plot 3 signals on one page. They are to have common X-axis scales, but separate vertical (Y-axis) scales. They are all to have horizontal axes drawn at Y = 0.0, and they are to be labeled "SIGNAL 1," "SIGNAL 2," and "SIGNAL 3," respectively. The length of the plot is to be 18 inches.

We would make the following calls:

```
CALL CPLOT(6HUSMITH,18.0,3,8,2,Y1,NSP1,0.0,Y2,NSP2,0.0,Y3,
NSP3,0.0)
CALL WRITE(1,8,8HSIGNAL 1 )
CALL WRITE(2,8,8HSIGNAL 2 )
CALL WRITE(3,8,8HSIGNAL 3 )
CALL WRITE(3,8,6HENDPIC )
```

The output might look something like:



Signal Analysis Subroutine SA-32

1. IDENTIFICATION

   Data Plotter,(FAP), Rathbun.

2. PURPOSE

   A subroutine to plot sampled values of two signals as part of the regular 1401 printer output.

3. USAGE

   A. Calling sequence:

CALL PPLOT (NSP, X1, Y1, X2, Y2, K, Y1MAX, Y1MIN, Y2MAX, Y2MIN)

B. Identification of input variable:

NSP, number of sample points to be plotted,

X1, array of values of independent variable for first signal,

Y1, array of corresponding values of dependent vraiable for first signal,

X2, as above (second signal),

Y2, as above (second signal),

K, Y-scale control parameter. To provide a common scale for both Y1 and Y2, set K=1. To provide independent scales for Y1 and Y2, set K=0.

Y1MAX, maximum value of Y1,

Y1MIN, minimum value of Y1,

Y2MAX, as above, (Y2),

Y2MIN, as above, (Y2).

C. Comments:

The YMAX and YMIN values need not be known a priori, for, by setting YMAX = YMIN, (i.e. both equal to the same dummy variable) the routine will search out the maximum and minimum values of the data to be plotted and automatically adjust its scales to provide the most expanded plots possible in 100 positions.

When Y falls outside of the specified range for (YMAX, YMIN), the data are converted modulo this range.

The Y1 axis may be labeled in columns 10-70 by writing on output tape 6 just prior to calling PPLOT. (The paper is not advanced on printing the first line by PPLOT.) Similarly, the Y2 axis may be labeled in columns 10-70 immediately after calling PPLOT by writing on output

tape 6, with a "+" in column 1 to prevent paper advance.

When the plot of only one signal is desired, treat the signal as if it were two identical signals.

There are no restrictions on NSP.

One coordinate pair is plotted on each line of the plot, regardless of the numerical value of the increment between successive elements of the X arrays.

Signal Analysis Subroutine SA-33

1. IDENTIFICATION

Printer for 1, 2, or 3 dimensional arrays (FORTRAN)

J. R. Reeder, Garry Odell

2. PURPOSE

A subroutine to print a part or the whole of a given one, two or three dimensional array in an appropriate E format.

3. USAGE

A. Calling sequence:

CALL PRT (NROWS, NCOLS, NLAYER, U, NRODIM, NCOLDM, MODULO)

B. Identification of input variables:

NROWS, the number of rows to be printed

NCOLS, the number of columns to be printed,

NLAYER, the number of layers to be printed,

U, the name of the array to be printed.

NRODIM, the row dimension of U in the calling program,

NCOLDM, the column dimension of U in the calling program,

MODULO, the number of columns to be printed per page.

C. Output:

The NCOLS columns are printed MODULO columns to a page. If NCOLS > MODULO, the remaining columns are printed immediately thereafter in the same format. The rows and columns are individually labeled. Each layer is printed entirely before the next, and the layers are labeled.

The format used is determined according to the value of modulo entered, such that for $4 \leq$ MODULO ≤ 10 the entire width of the output page is filled, e.g. for MODULO = 10, a row is printed in (10 E 12.4) format. For $1 \leq$ MODULO ≤ 4 the NCOLS columns are printed MODULO columns to a page in (E 30.9) format.

D. Restrictions:

NROWS, NCOLS, NLAYER, < 1000

$1 \leq$ MODULO ≤ 10

E. Comments:

Selected portions of an array may be printed by judicious entry into the subroutine, i.e. U(1, 1, 4), U(300), etc., are all legitimate input arguments and will cause that particular location to be the first number printed

F. Example:

Suppose we have a three dimensional array, dimensioned (2, 2, 2), which contains the following values:

A(1,1,1) = 1.	A(1,2 1) = 2.
A(2,1,1) = 3.	A(2,2,1) = 4.
A(1,1,2) = 10.	A(1,2,2) = 20.
A(2,1,2) = 30.	A(2,2,2) = 40.

and we wished to print out some part, or all of it.

If we wished to print the entire array we could make the following entry to PRT:

```
CALL PRT(2,2,2,A,2,2,10)
```

The above call would cause the following print-out:

```
                    DEPTH INDEX = 1
                       COL 1              COL 2
      ROW 1          .1000E+01          .2000E+01
      ROW 2          .3000E+01          .4000E+01

                    DEPTH INDEX = 2
                       COL 1              COL 2
      ROW 1          .1000E+02          .2000E+02
      ROW 2          .3000E+02          .4000E+02
```

Note: in this example MODULO = 10 was entered, which causes the values to be printed in E12.4 format; however, any other value from 1 to 9 could have been entered which would cause various degrees of format expansion up to E30.9 for MODULO = 1.

If we wished to print only the elements A(1,1,1) and A(1,1,2) we might make the following entry to PRT:

```
CALL PRT(1,1,2,A,2,2,6)
```

The above call would result in the following print out:

```
                    DEPTH INDEX  = 1
                                  COL 1
      ROW 1                  .100000000E+01

                    DEPTH INDEX = 2


      ROW 1                  .100000000E+02
```

Note that MODULO = 6 has caused a format of E20.9 to be used.

Signal Analysis Subroutine SA-34

1. IDENTIFICATION

Matrix printout subroutine, (FORTRAN), J. R. Reeder

2. PURPOSE

A subroutine to printout a part or the whole of a given array in an appropriate format; that is, given a P x Q array this routine prints an M x N part of it, where $M \leq P$, $N \leq Q$.

3. METHOD

The subroutine treats the input matrix as a one-dimensional array of numbers and causes M x N numbers of this array to be printed in 10 E 12.4 format.

4. USAGE

A. Calling sequence:

CALL PRTOUT (NCOL, NROW, U, NRODIM)

B. Identification of variables in calling sequence:

NCOL, the number of columns to be printed,

NROW, the number of rows to be printed,

U, the name of the matrix to be printed,

NRODIM, the row dimension of the matrix U in the calling program.

C. Output:

The NCOL columns are printed 10 at a time in 10 E 12.4 format, and the rows and columns are individually labeled.

D. Restrictions

$NCOL \leq 99$

This subroutine cannot be used to print three dimensional arrays.

E. Example:

Suppose we had a two dimensional array, H, of row dimension 3, and wished to print out the first 2 rows and 5 columns.

H =

1.0	2.0	3.0	4.0	5.0
6.0	7.0	8.0	9.0	10.0
11.0	12.0	13.0	14.0	15.0

We would make the following call:

```
CALL PRTOUT (5, 2, H, 3)
```

The above would cause the following output:

```
        1            2            3            4            5
1)  .1000E+01    .2000E+01    .3000E+01    .4000+01     .5000E+01
2)  .6000E+01    .7000E+01    .8000E+01    .9000E+01    .1000E+02
```

PAGE INDEX

PAGE INDEX (Continued)