MATHEMATICS NOTES

Note 45

March 1977

User's Manual for SEMPEX: A Computer Code for Extracting Complex Exponentials from a Time Waveform

Darrel L. Lager
H. Gerald Hudson
Andrew J. Poggio

Lawrence Livermore Laboratory
Livermore, California

ABSTRACT

This report is the user's manual for the SEMPEX computer code. Singularity Expansion Method Pole Extraction is a technique in electromagnetics in which the free response of a structure is expressed as a weighted sum of complex exponentials where the damping factors of the exponentials are derived from the poles (i.e., singularities) of the transfer function of the object.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (continued)

TABLES

SECTION I

INTRODUCTION

A recent theoretical advancement in electromagnetics describing the response of a structure in terms of complex exponentials has been called the Singularity Expansion Method (SEM) (refs. 1 and 2). For N uniformly spaced time samples, this can be written as

$$f\,(t_i) = \sum_{j=1}^{M} A_j e^{\alpha_j t_i} \tag{1}$$

where

A = complex amplitude

α = complex natural frequency or damping coefficient

M = number of independent exponentials in the data

t_i = $(i-1)\Delta t$

See reference 3 for a complete discussion relating equation 1 to SEM.

Equation 1 is derived from the Laplace transform of the transfer function of the structure response

$$F(s) = \frac{(s-\gamma_1)(s-\gamma_2)\ \ldots\ (s-\gamma_{M-1})}{(s-\alpha_1)(s-\alpha_2)\ \ldots\ (s-\alpha_M)} \tag{2}$$

where

γ_i = a zero of the transfer function since $s = \gamma_i$ makes $F(s) = 0$

α_i = a pole or singularity of the transfer function since $s = \alpha_i$ makes $F(s) \to \infty$

---

1. Baum, Carl E., "On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems," Air Force Weapons Laboratory, EMP Interaction Note 88 (1971).

2. Tesche, Fredrick M., "On the Singularity Expansion Method as Applied to Electromagnetic Scattering from Thin Wires," EMP Interaction Note 102 (1972).

3. Poggio, A. J., Lager, D. L., and Hudson, H. G., Transient Data Processing Using Complex Exponential Representations, to be published.

By performing a partial fraction expansion we can express equation 2 as

$$F(s) = \sum_{j=1}^{M} \frac{A_j}{s-\alpha_j} \qquad (3)$$

where A_j is defined as the residue for the jth pole α_j. Since the inverse Laplace transform of equation 3 yields equation 1, the complex natural frequencies correspond to poles and the amplitudes correspond to residues; therefore these terms are used interchangeably in the text.

A particularly valuable result of the SEM approach is that the poles are determined by the properties of the structure response. That is, they are independent of the structure's excitation and orientation. Hence, we can characterize a structure by observing its response to a signal only once.

Another advantage of this approach is that it usually takes very few exponentials to sufficiently describe the structure response for late times; thus we can achieve significant data compression. For example, some time responses requiring 512 points for adequate description may be described with as few as 10 poles and 10 residues.

The purpose of SEMPEX (Singularity Expansion Method Pole Extraction Program) is to extract the SEM natural frequencies and complex amplitudes which characterize a given time waveform. The approach used is based on Prony's method (refs. 4 and 5) which he first developed in 1685. The program runs on the CDC 7600. The user must provide a disk file or tape describing the time waveform and a set of control cards describing the time and voltage calibration factors, the number of poles requested, various output control parameters, etc. The output consists of plots showing the original waveform, its amplitude spectrum, the poles extracted, and a reconstruction of the temporal waveform using the extracted poles. Also, tables are printed out which give the values of the poles and residues.

---

4. Householder, A. S., On Prony's Method of Fitting Exponential Decay Curves and Multiple-Hit Survival Curves, Oak Ridge National Laboratory, Oak Ridge, Tenn., Rept. ORNL-455 (1950).

5. Hildebrand, F. B., Introduction to Numerical Analysis (McGraw-Hill Book Co. Inc., New York, 1956).

SECTION II

THEORY

The data processing scheme is based on Prony's method, a technique for solving the nonlinear curve fitting problem

$$f(t_i) = \sum_{j=1}^{M} A_j e^{\alpha_j t_i} \qquad i = 1, 2, \ldots, N \tag{4}$$

for the 2M unknown parameters $\{A_j;\ j=1, 2, \ldots, M\}$ and $\{\alpha_i;\ j-1, 2, \ldots, M\}$. The procedure (ref. 5) for finding the poles begins by defining

$$x_j = e^{\alpha_j \Delta t} \tag{5}$$

so equation 4 can be written in the form

$$\begin{array}{lllllll} A_1 & + & A_2 & + \ldots + & A_M & = f_0 & = f(0) \\ A_1 x_1 & + & A_2 x_2 & + \ldots + & A_M x_M & = f_1 & = f(1\Delta t) \\ \vdots & & \vdots & & \vdots & & \\ A_1 x_1^{N-1} & + & A_2 x_2^{N-1} & + \ldots + & A_M x_M^{N-1} & = f_{N-1} & = f(N-1)\Delta t \end{array} \tag{6}$$

for each of the N equally spaced samples.

Prony (ref. 6) observed that each of the x_i satisfied an *M*th order polynominal of the form

$$x^M - C_1 x^{M-1} - \ldots - C_{M-1} x - C_M = 0 \tag{7}$$

6. Kelly, Louis G., Handbook of Numerical Methods and Applications (Addison-Wesley, Reading, Mass., 1967).

The solution for the $\{C_i;\ i=1, 2, \ldots M\}$ is found by forming M linear equations as follows. Multiply the first row of equation 6 by C_M, the second by C_{M-1}, the third by C_{M-2} ..., the *M*th by C_1, the *(M+1)*th by -1, and add the resulting equations. Manipulating the matrix terms to obtain the form of equation 7 results in

$$0 = C_M f_0 + C_{M-1} f_1 + \ldots C_1 f_{M-1} - f_M \tag{8}$$

which is the first linear equation for the $\{C_j\}$. The second equation is formed by a similar operation on equation 6 beginning with the second row. The third is formed beginning with the third row of equation 6. After M equations for the $\{C_j\}$ have been formed, the resulting system is solved for the $\{C_j\}$. The poles, α_i, are found using equation 5 after finding the roots, x_j, of the polynomial equation 7.

To express the above in matrix notation, we define an (N–M) × M matrix using the notation $f_i = f(i\Delta t)$

$$F = \begin{bmatrix} f_1 & f_2 & \cdots & f_M \\ f_2 & f_3 & \cdots & f_{M+1} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ f_{N-M} & f_{N-M+1} & \cdots & f_{N-1} \end{bmatrix} \tag{9}$$

and an M – N column vector B

$$B = \begin{bmatrix} -f_{M+1} \\ -f_{M+2} \\ \cdot \\ \cdot \\ -f_N \end{bmatrix} \tag{10}$$

The coefficients C_1 through C_M of the polynominal equation 7 are then found by solving with Crout's algorithm (ref. 5) the system

$$FC = B$$

For the situation where the system is over-determined, i.e., $N > 2M$, we need to find the coefficients C which give the least-squares solution to the above system of equations. We wish to find the vector C which minimizes

$$\phi = \sum_{i=1}^{N} r_i^2 \tag{11}$$

where $r = FC - B$, the vector of residuals.

The minimum of ϕ occurs at the point where all the partial derivatives with respect to the coefficient C_i are zero. We want to find the vector C where

$$\nabla\phi = 0 \tag{12}$$

where

$$\nabla = \frac{\partial}{\partial C_1}, \frac{\partial}{\partial C_2} \cdots \frac{\partial}{\partial C_m}$$

This results in the forming of the so-called normal equations for least squares (ref. 7) where we solve for C in the system:

$$\tilde{F}FC = \tilde{F}B \tag{13}$$

where $\tilde{F}$ denotes F transpose.

The situation of an exactly determined system ($M = 2N$) reduces to the square case above. The elements of the vector C are the coefficients of the polynomial equation 7 whose roots x_j are found with the subroutine MULLER (ref. 8).

---

7. Pennington, Ralph H., Introductory Computer Methods and Numerical Analysis (MacMillan, New York, 1965).

8. Lawrence, J. Dennis, Polynomial Root Finder, Lawrence Livermore Laboratory, Rept. C22-001 (1966).

The poles are found by the formula

$$\alpha_j = \frac{1}{\Delta t} \ln x_j \qquad \text{(complex natural logarithm)} \tag{14}$$

After determining the poles $\{\alpha_j\}$, we can find the residues $\{A_j\}$ by defining an $N \times M$ matrix

$$F_1 = \begin{bmatrix} 1 & 1 & . & . & . & 1 \\ x_1 & x_2 & . & . & . & x_M \\ x_1^2 & x_2^2 & . & . & . & x_M^2 \\ x_1^3 & x_2^3 & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_1^{N-1} & . & . & . & . & x_M^{N-1} \end{bmatrix} \tag{15}$$

and a column vector of length N from the original data points

$$Y_1 = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_N \end{bmatrix} \tag{16}$$

and by using subroutine CROUT to solve for A in the system

$$F_1 A = Y_1 \tag{17}$$

For an over-determined system $N > 2M$, we need to again form the normal equations for least squares and solve the system

$$\tilde{F}_1 F_1 A = \tilde{F}_1 Y_1 \tag{18}$$

where $\tilde{F}_1$ denotes F_1 transpose.

SECTION III
USAGE INSTRUCTIONS

The SEMPEX program is used to extract poles and residues from a time waveform f(t) according to the procedure outlined in the block diagram of figure 1. The user supplies appropriate instruction cards and a disk file or tape containing the equally spaced data points of the waveform. The program then plots the filtered amplitude spectrum of f(t), extracts the complex poles and residues and plots them, and compares the filtered f(t) with a reconstruction using all the poles found with a reconstruction using only the poles which meet user-specified criteria. The second reconstruction may be extended beyond the last time value used by Prony's method to give an estimate of the "late time" response of the object. The criteria allow the user to reject poles which have residues of insignificant magnitude or which are nonphysical (i.e., so-called curve fit poles). The pole test criteria are based on either the real or imaginary part of the pole where

$$\begin{aligned} \sigma &= \mathrm{Re}[\alpha] \\ f &= (1/2\pi)\ \mathrm{Im}[\alpha] \end{aligned} \qquad (19)$$

A pole is nonphysical if it lies in the right half-plane due to a positive σ. Positive σ implies a response that increases with time, an impossibility for a passive structure. A pole can also be rejected if σ is a large negative number since this implies an exponential which damps out within the first two or three points in the waveform. Poles which damp out so rapidly are usually not of interest since they result solely from attempts to "fit" the data. A pole can also be rejected as a curve fit pole if its frequency f is higher than the cutoff frequency of the filter.

Two sets of tables are written, one listing all the poles and residues extracted and the other listing those which meet the pole test criteria. There are two tables in each set, one ordered by ascending magnitude of the imaginary part of the pole (i.e., by ascending frequency f defined in equation 19) and the other ordered by descending magnitude of the residue.

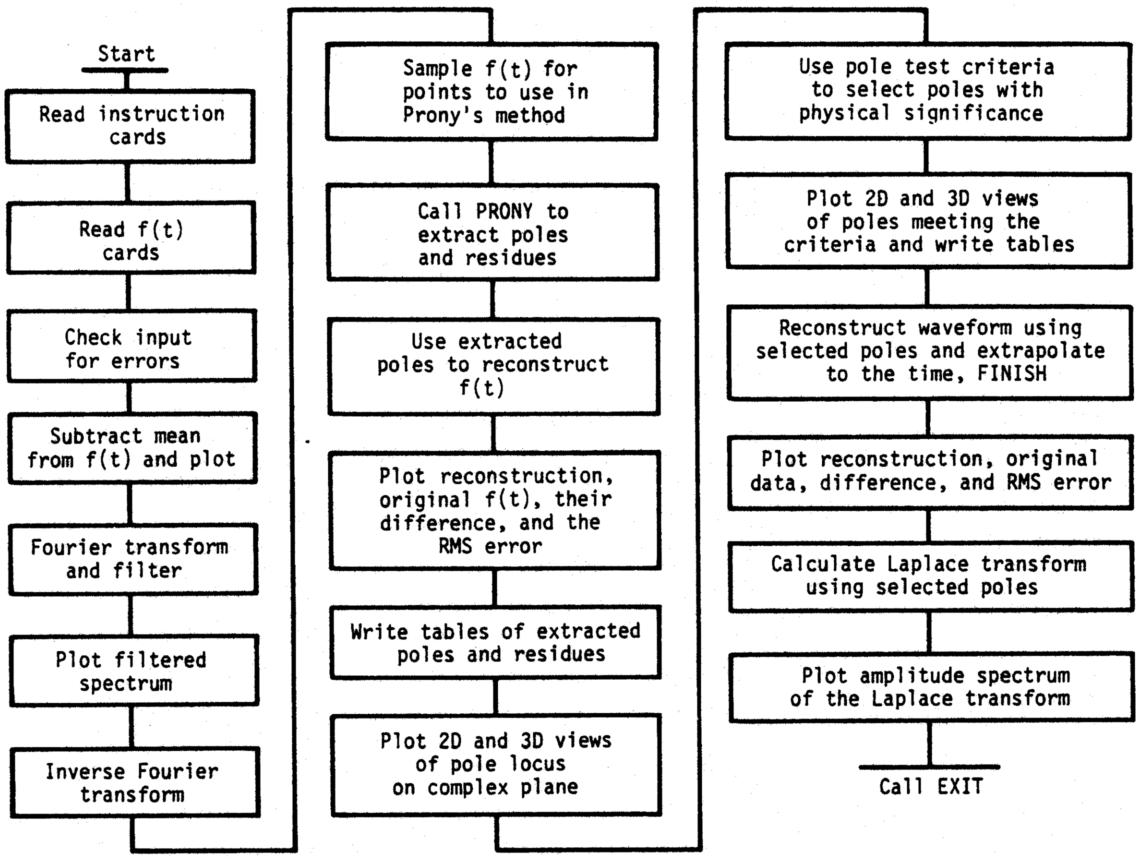

Figure 1. Flow diagram for SEMPEX, a program to extract SEM poles.

The instructions to SEMPEX are specified on cards in the following order:

CARD 1 (8A10) - A line of run description.

CARD 2 (2F15.0) - TIMECAL, VCAL.

TIMECAL = The total time duration of f(t) in seconds.

VCAL = A calibration factor to multiply f(t) for scaling.

CARD 3 (4I5) - NPOLES, NBEGIN, NPTS, NDECI.

NPOLES = The number of poles to be extracted from the data. If the user knows a priori the number of poles in the data, NPOLES is generally given a value of 2 to 4 more than that number. The extra poles allow the program to give a better fit to the data and furthermore will allow the user to verify his knowledge of the number of poles. The extraneous poles will have very small residues.

If the number of poles is unknown, the user typically requests a large number of poles (25 to 50) and waits to see what happens. If the program exits with a singular matrix error (or sometimes an overflow in the CROUT algorithm for solving systems of equations), it usually means far more poles were requested than were required to fit the data, resulting in the matrix F in equation 13 becoming singular (ref. 3). The user must reduce NPOLES and make other runs until the errors cease. Then he can observe the number of poles with significant residues to get more exact estimates of the correct number and make a final run requesting 2 to 4 more poles than the number apparently in the data. For the runs where no errors occur, the user can expect a great deal of movement of the "curve fit" poles from run to run while the large residue poles due to the structure will remain stationary.

Another way of estimating the number of poles in the data is to count the number of peaks in the spectrum of the waveform. Each peak is generally caused by two poles (a conjugate pair). A rule of thumb is to count the peaks with an amplitude greater than a factor of 10^{-3} of the largest peak.

NBEGIN = The number of the first data point to be used in Prony's algorithm as shown in figure 2. NBEGIN should be chosen at a point after the driving function has died to zero (i.e., after the incident pulse has passed the structure) and the free response of the structure has begun. This statement is based on the following reasoning*: Let the response f(t) be the result of the convolution of the structure impulse response g(t) with the excitation h(t)

$$f(t) = g(t)*h(t) = \int_0^t g(t-\tau)\ h(\tau)\ d\tau \tag{20}$$

Let

$$g(t) = \sum_{j=1}^{M} A_j e^{\alpha_j t} \tag{21}$$

and

$h(t)$ = a pulse damping to zero for $t > t_1$

Then

$$f(t) = \sum_{j=1}^{M} e^{\alpha_j t} A_j \int_0^t e^{-\alpha_j \tau}\ h(\tau)\ d\tau \tag{22}$$

or

$$f(t) = \sum_{j=1}^{M} R(t)\ e^{\alpha_j t} \tag{23}$$

where

$$R(t) = A_j \int_0^t e^{-\alpha_j \tau}\ h(\tau)\ d\tau \tag{24}$$

---

*Brittingham, J., Lawrence Livermore Laboratory, personal communication.

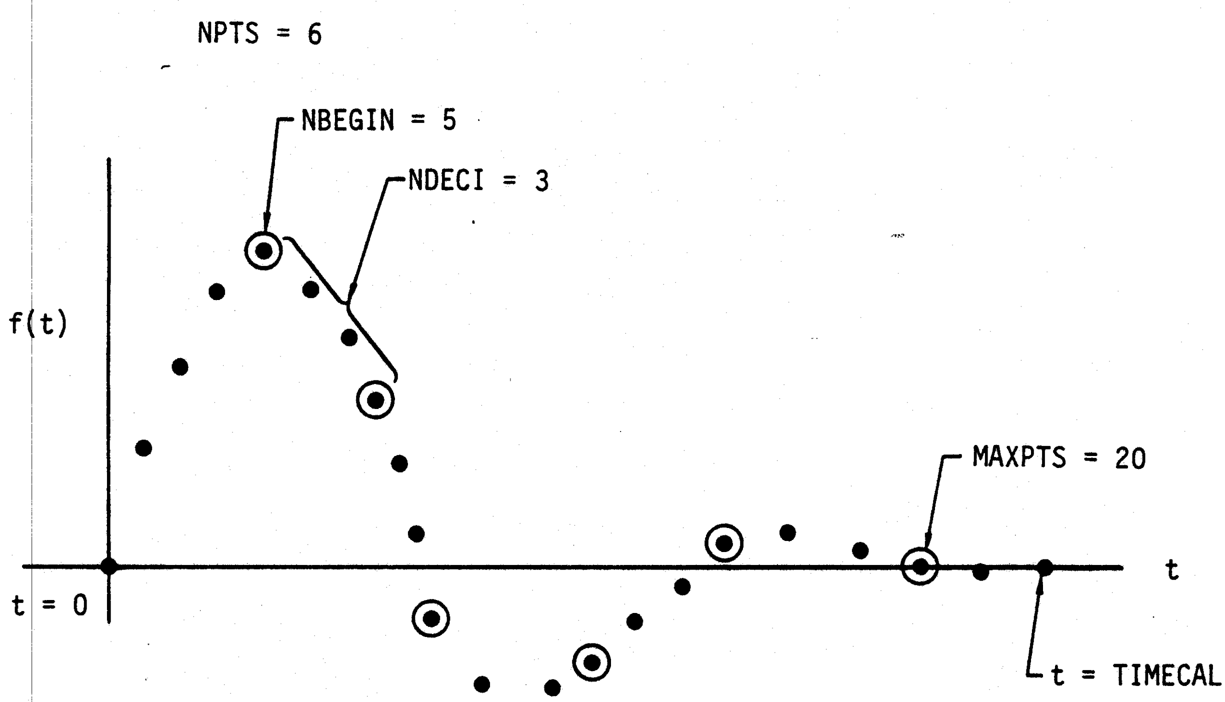

Figure 2. A sample waveform to demonstrate the relation between NBEGIN, NPTS, MAXPTS, and NDECI. The dots indicate the data points of the waveform; the circles indicate the points used in Prony's method to extract poles.

Since the integral in equation 24 is a constant for $t > t_1$, f(t) can be viewed as a sum of exponentials with time-varying residues, R(t), for $t < t_1$ and constant residues for $t > t_1$. Prony's method assumes the residues are time-invariant; therefore, the method cannot be used to extract poles from the waveform while the excitation is present (i.e., for $t < t_1$).

NDECI = Determines the sampling interval for choosing points from the input data to use in Prony's algorithm as shown in figure 2. To avoid the phenomenon of pole foldover (analogous to aliasing in the discrete Fourier transform), NDECI is chosen so the sampling interval between the decimated points satisfies the Nyquist criterion

$$\Delta t_p \leq 1/(2f_{max}) \tag{25}$$

where

Δt_p = spacing between points used by Prony's algorithm,
Δt_p = NDECI * Δt_d

Δt_d = spacing between the data points

f_{max} = highest frequency component in the data

NDECI should be as large as possible to allow the Prony algorithm to work with data over as long a time window as possible.

NPTS = The number of points to be used in Prony's algorithm. NPTS must be greater than or equal to twice the number of poles requested. If a few poles are being requested (say less than 20), the best results are obtained by having NPTS = 2*NPOLES. This gives a so-called square system where the number of unknowns (i.e., residues and poles) to be determined is exactly equal to the number of points. For this case the algorithm gives an exact fit to the data (within the unit roundoff error of the machine, about 10^{-14}).
If a large number of poles is being requested (say greater than 50), the best results are obtained by having NPTS $\geq$ 2*NPOLES. The solution will be a least-squares solution so the fit will no longer be exact; however, the locus of the poles extracted will usually be unaffected by the poorer fit.

NPTS has a further constraint that the index of the last point chosen, MAXPTS, must be less than 512, the number of points in the data record. The formula for determining MAXPTS is

$$\text{MAXPTS} = (\text{NPTS}-1)* \text{NDECI} + \text{NBEGIN} \quad (26)$$

CARD 4 (3F12.0) FMAX, FLOW, FHIGH - Controls truncation filter.

FMAX = A display parameter specifying (in Hz) the maximum frequency of the spectrum of the waveform to be plotted.

FLOW = Specifies (in Hz) the low frequency cutoff of the bandpass filter which processes the waveform before applying Prony's algorithm.

FHIGH = Specifies (in Hz) the high frequency cutoff of the bandpass filter which processes the waveform before applying Prony's algorithm. The filter used is a simple "truncation filter" where the frequency components outside the passband are simply set to zero. Since the transforms from time to frequency and back again are done with a Fast Fourier Transform, the user must choose FLOW and FHIGH judiciously to avoid distortions due to windowing effects. FLOW and FHIGH must be chosen at notches in the spectrum at least two orders of magnitude below the peak.

FHIGH = 0 means to omit the filtering.

CARD 5 (I5) ITEST - Output control card.

ITEST = 0 or a blank card gives an abbreviated output consisting of a plot of the original data, the amplitude spectrum of the filtered data, the resulting time waveform (the inverse FFT of the filtered spectrum), a plot and list of all the poles extracted by Prony's algorithm, a plot of a data reconstruction using all the extracted poles, and a plot of the error between the reconstruction and the filtered time waveform.

ITEST = 1 gives all the above plus plots (and lists) of the poles which meet the pole test criteria (specified on card 6), a plot of another reconstruction (which may be extrapolated beyond the last time value used in Prony's algorithm), a plot of the error between the reconstruction and the filtered time waveform, and the amplitude spectrum derived from the Laplace transform of the poles that meet the test criterion.

The Laplace transform spectrum is determined by

$$F(s) = \sum_{i=1}^{M} \frac{A_i}{s-\alpha_i} \qquad (27)$$

where

s = complex frequency, $\sigma + j2\pi f$

M = number of poles meeting pole test criteria

The spectrum plotted is the magnitude of F(s) evaluated for $s = j2\pi f$ as a function of frequency.

CARD 6 (4F12.0) RES, RHP, PREAL, PIMAG - The pole test criteria.

RES = The residue criterion. It eliminates all poles with residues smaller than RES times the largest residue. For RES = 0, none of the poles are eliminated.

RHP = The right half-plane criterion. Poles which have a real part greater than RHP are eliminated (the real part of desired poles is always negative) as shown in figure 3. RHP is normally 0 or a very small positive number (e.g., 10^{-6}).

PREAL = Eliminates poles which damp out too rapidly. As shown in figure 3, poles with a real part less than PREAL (since they are negative numbers) are eliminated.

PIMAG = Frequency test criterion. As shown in figure 3, poles with an oscillation frequency greater than PIMAG (in Hz) are eliminated. PIMAG is usually set to a number slightly higher than the upper cutoff of the filter, FHIGH.

CARD 7 (E15.5) FINISH - Specifies the extrapolation time.

FINISH = Specifies the time (in seconds) to end the reconstruction using the poles which meet the pole test criteria. The reconstruction begins at the first time used in Prony's algorithm, T(NBEGIN), and ends at FINISH. The spacing between the points in the reconstruction is the same as that used in Prony's algorithm as determined by equation 25. The maximum value for FINISH is set by the dimensions of the array EXTRAP which can contain only 2048 points

$$\text{FINISH} \leq 2048*\Delta t_p + \text{T(NBEGIN)} \qquad (28)$$

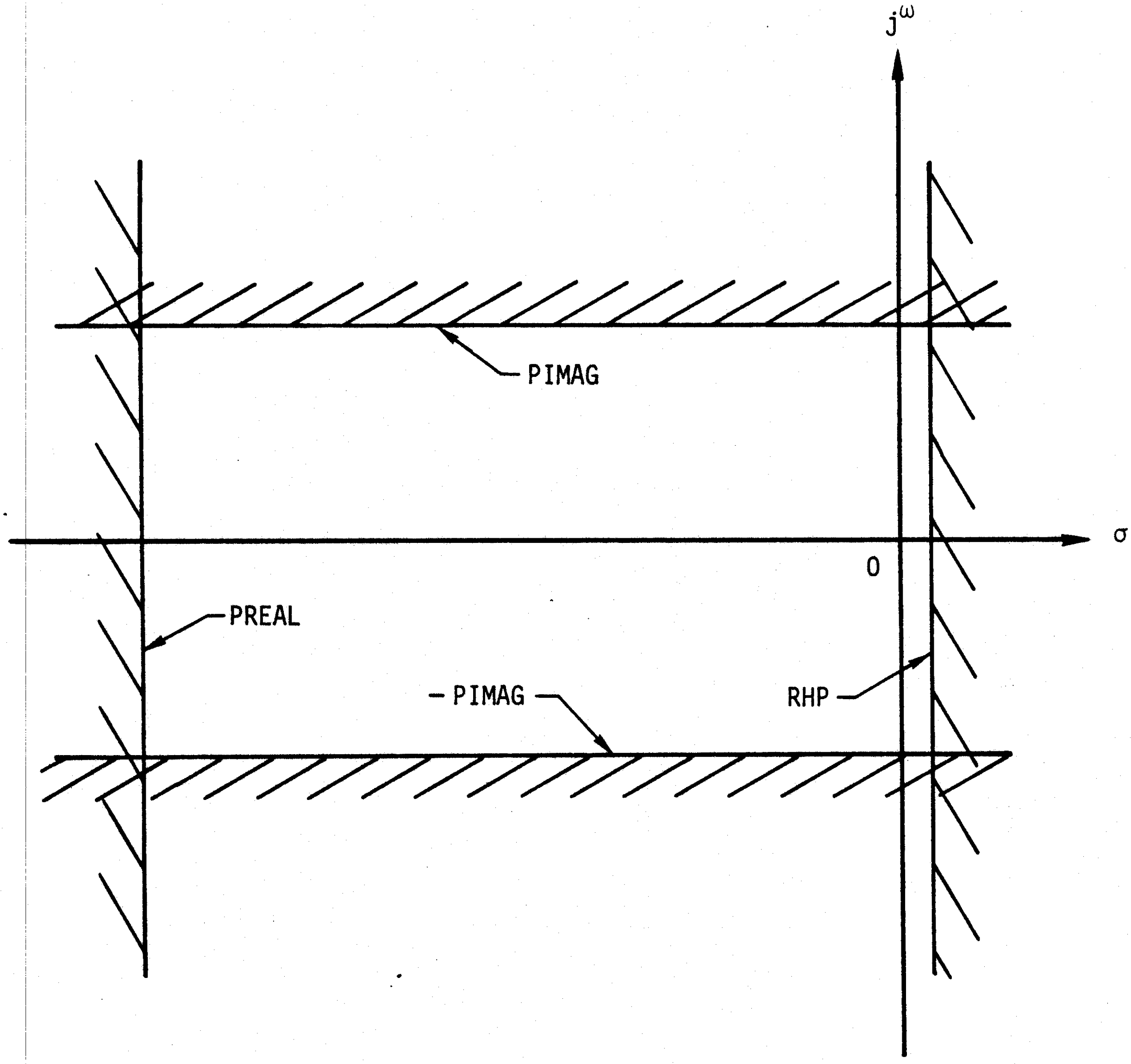

Figure 3. Pole test criteria. Poles lying in any of the crosshatched regions of the complex plane will be eliminated. Variables specifying the region boundaries are shown.

SECTION IV

DATA FORMAT

The data specifying the values of the waveform to be analyzed are specified on cards according to a (1X, F12.7, 12X, F12.7) FORMAT. The program reads 512 points from the cards as shown:

CARD1	f(1)	f(2)
CARD2	f(3)	f(4)
.	.	.
.	.	.
.	.	.
CARD256	f(511)	f(512)

SECTION V

EXAMPLE PROBLEMS

SAMPLE PROBLEM 1

The SEMPEX program was used to extract poles from the time waveform shown in figure 4. The waveform is the computed backscattered field due to a Gaussian pulse incident at an angle 30° from broadside to a 60-m long wire. The calculation was done using the time-domain computer code WT-MBA/LLL1B (ref. 9).

The control cards for SEMPEX were:

```
CARD1 - RUN-205 EXAMPLE PROBLEM
CARD2 - TIMECAL = 1703.7E-9
        VCAL = 1.
CARD3 - NPOLES = 20
        NBEGIN = 75
        NPTS = 40
        NDECI = 7
CARD4 - FMAX = 50.E6
        FLOW = 0
        FHIGH = 18.E6
CARD5 - ITEST = 1
CARD6 - RES = 1.E-4
        RHP = 0
        PREAL = -6.E7
        PIMAG = 17.99E6
CARD7 - FINISH = 4.E-6
```

The spectrum of the waveform in figure 4 is shown in figure 5. The components above FHIGH = 18 MHz were set to zero by the truncation filter. The cutoff of the filter was chosen at a notch in the spectrum to reduce "windowing" effects. The new time waveform obtained after applying the inverse FFT to the filtered spectrum is shown in figure 6.

---

9. Landt, J. A., Miller, E. K., and Van Blaricum, M., WT-MBA/LLL1B: A Computer Program for the Time-Domain Electromagnetic Response of Thin-Wire Structures, Lawrence Livermore Laboratory, Rept. UCRL-51585 (1974).

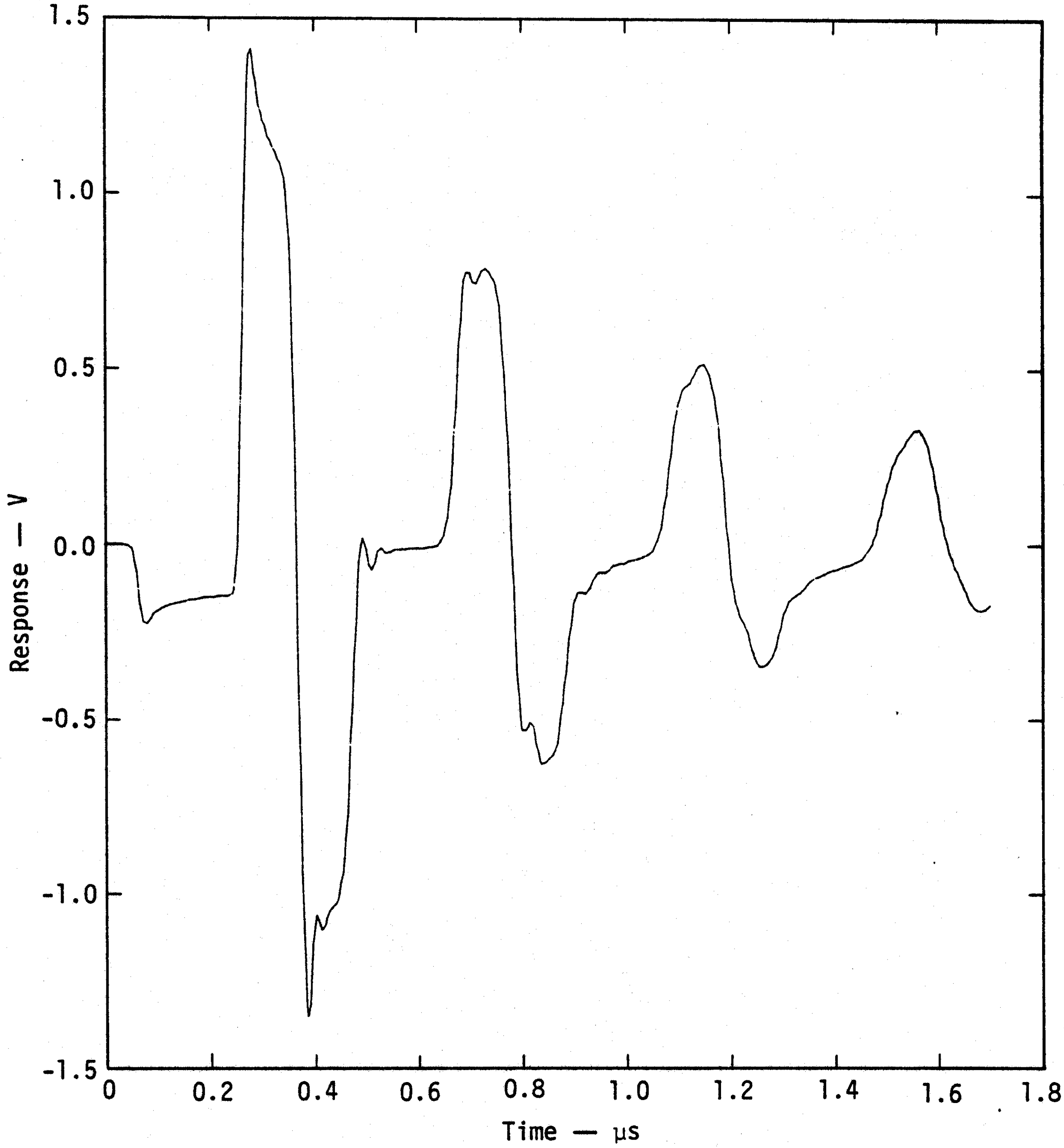

Figure 4. Backscattered field from a 60-m dipole due to Gaussian pulse.

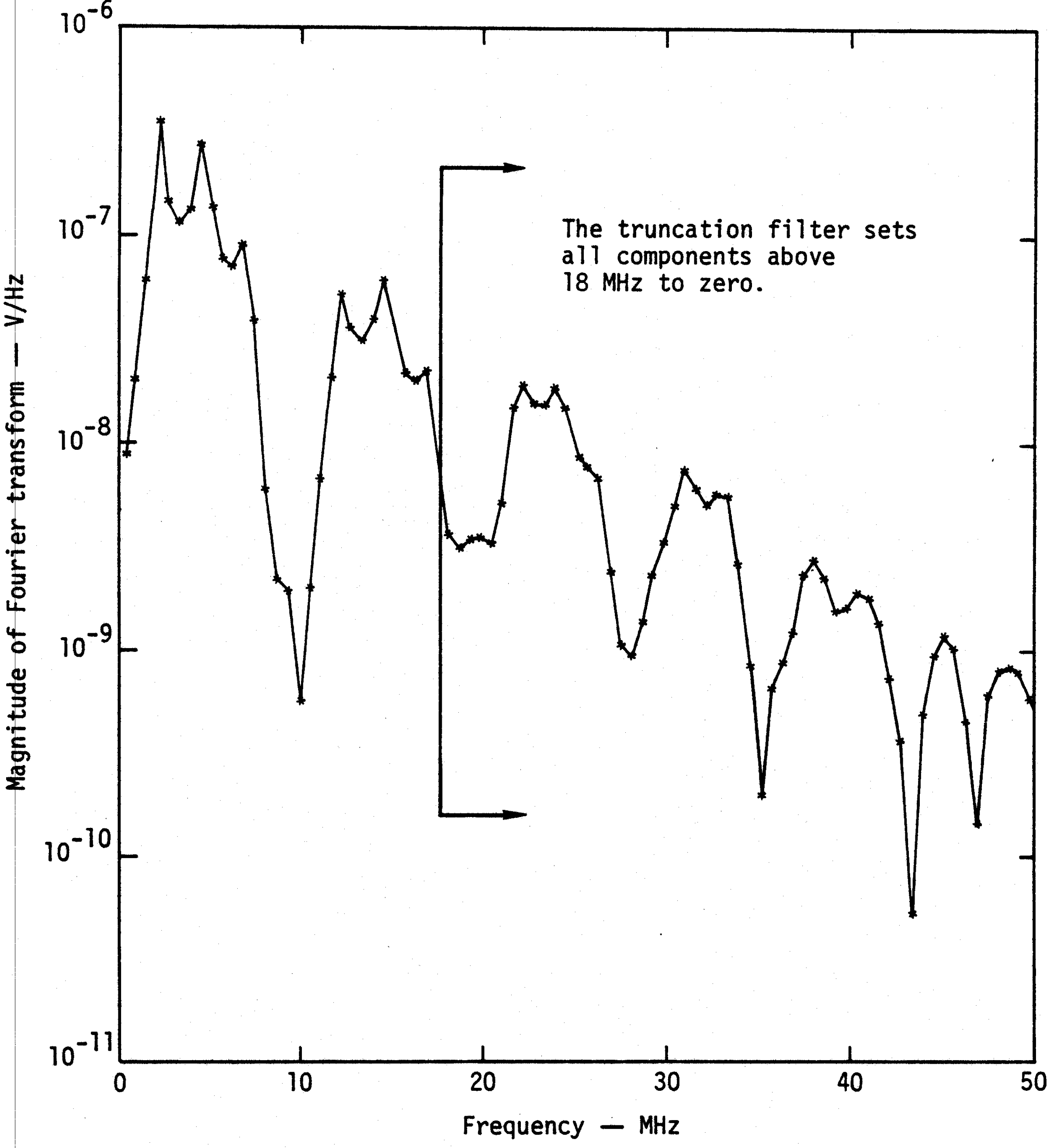

Figure 5. Amplitude spectrum of figure 4 waveform.

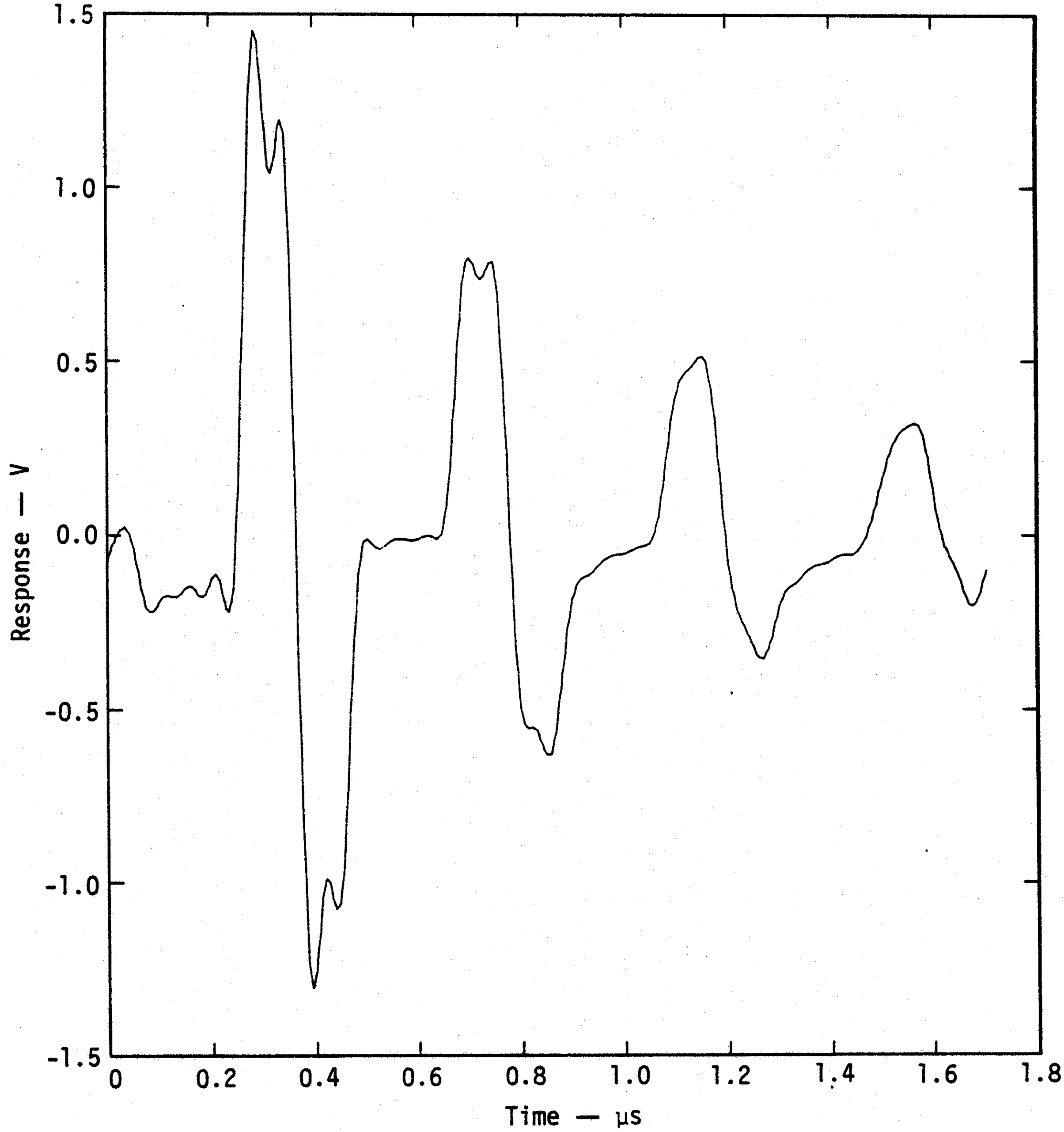

Figure 6. Time waveform after filtering with truncation filter.

The points of the time waveform used in Prony's algorithm are indicated in figure 7 by asterisks. The points chosen are controlled by the parameters on card 3.

All the poles extracted from the waveform are listed in table 1 in the ascending order of frequency (the imaginary part of alpha) and listed in table 2 by descending order of the magnitude of the residues.

The locus of the poles in the complex plane is shown in figure 8 where the location of each of the poles is indicated by an "x." A three-dimensional view of the complex plane is shown in figure 9 where the arrows parallel to the Z axis are proportional to the logarithm of the magnitude of each pole's residue, after normalizing the magnitude of the largest residue to 1. The highest point on the Z-axis is therefore 1. The lowest is 10^{-3} to give a three decade range in magnitudes. For poles with residues less than 10^{-3} of the largest, only an asterisk is plotted on the plane.

Figure 10 is a plot of the locus of the poles that satisfy the pole test criteria specified on card 6. Since the poles occur in conjugate pairs, only the upper left half-plane is shown. Figure 11 is a three-dimensional view of the poles that meet the test criteria. The actual values of the plotted poles and their residues are given in tables 3 and 4.

The poles that passed the test criteria were used to reconstruct the filtered time waveform as shown in figure 12. The data points used in the Prony algorithm are indicated by asterisks. The reconstruction agrees very well with the data points used and extrapolates nicely.

Figure 13 shows the spectrum of the reconstructed waveform derived from the poles that passed the pole test criteria according to equation 27. The spectrum of figure 13 is usually termed the Laplace transform to differentiate it from the Fourier transform used earlier.

A comparison of figures 11 and 13 reveals that a peak in the spectrum occurs across from the poles which have large residues. Furthermore, there are no peaks greater than 18 MHz since the truncation filter has eliminated all the components above that frequency. The Laplace transform does not go to zero above 18 MHz as does the Fourier spectrum because a finite sum of exponentials cannot describe such a discontinuity. The amplitude above 18 MHz

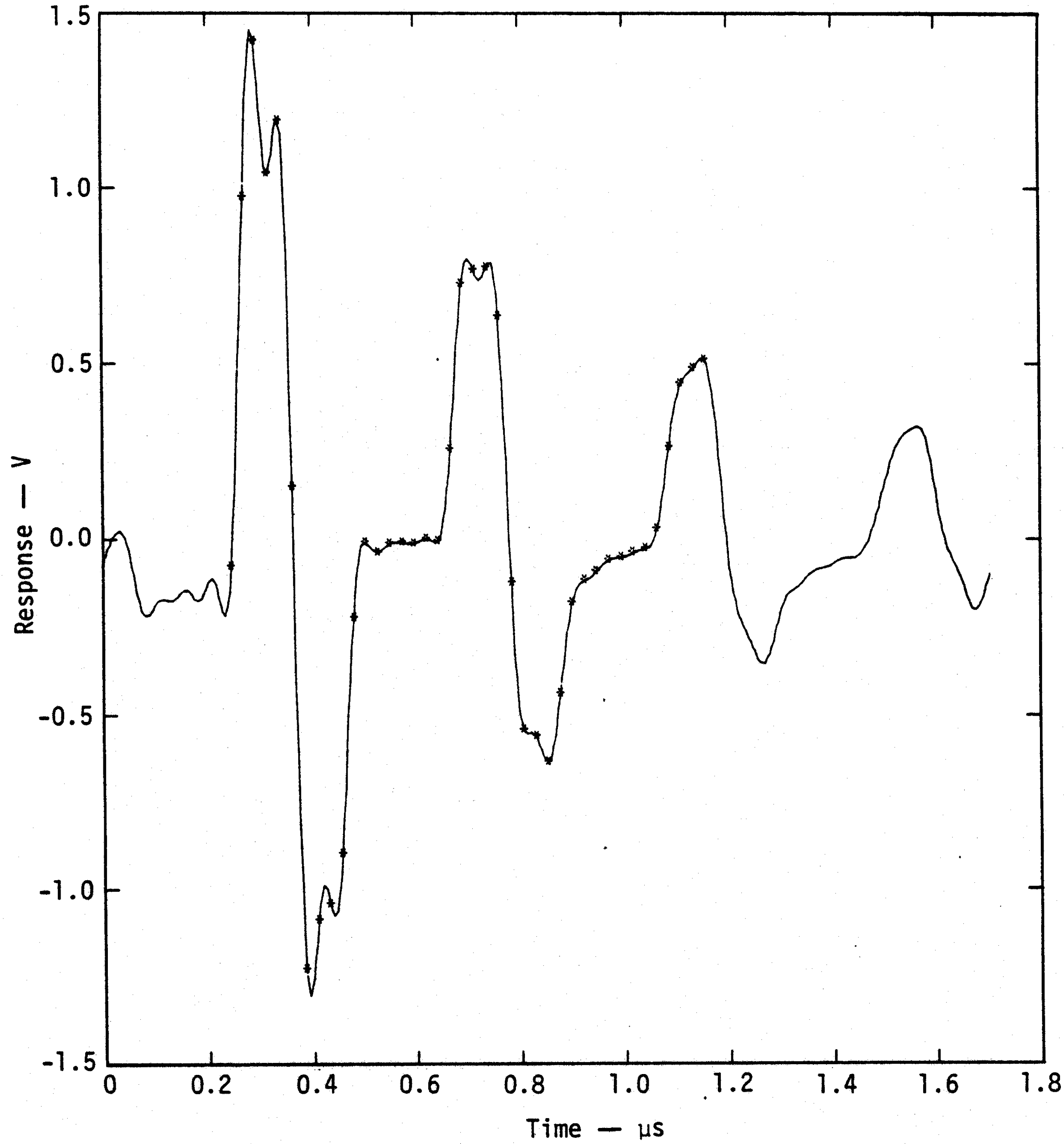

Figure 7. Filtered waveform data points used in Prony's algorithm.

TABLE 1. EXTRACTED POLES -- ASCENDING FREQUENCY ORDER (IMAGINARY PART OF ALPHA).

	A			ALPHA	
	MAG	R	I	R	I
1	2.65871E-03	-2.65871E-03	5.66381E-14	-1.14466E+04	0.
2	4.15409E-01	4.12118E-01	5.21797E-02	-1.06144E+06	-2.34744E+06
3	4.15409E-01	4.12118E-01	-5.21797E-02	-1.06144E+06	2.34744E+06
4	4.24372E-01	-1.69844E-01	3.88902E-01	-1.51629E+06	-4.80982E+06
5	4.24372E-01	-1.69844E-01	-3.88902E-01	-1.51629E+06	4.80982E+06
6	1.31866E-01	-1.03886E-01	8.12177E-02	-1.82476E+06	7.28018E+06
7	1.31866E-01	-1.03886E-01	-8.12177E-02	-1.82476E+06	-7.28018E+06
8	8.01044E-04	7.18980E-04	-3.53184E-04	-2.11901E+06	9.70691E+06
9	8.01044E-04	7.18980E-04	3.53184E-04	-2.11901E+06	-9.70691E+06
10	9.20747E-02	1.37488E-02	-9.10424E-02	-2.24812E+06	1.22084E+07
11	9.20747E-02	1.37488E-02	9.10424E-02	-2.24812E+06	-1.22084E+07
12	1.40992E-01	-7.90689E-02	1.16735E-01	-5.85247E+07	1.41588E+07
13	1.40992E-01	-7.90689E-02	-1.16735E-01	-5.85247E+07	-1.41588E+07
14	1.27327E-01	-1.26602E-01	1.35743E-02	-2.40366E+06	1.46561E+07
15	1.27327E-01	-1.26602E-01	-1.35743E-02	-2.40366E+06	-1.46561E+07
16	4.06613E-02	1.23228E-02	-3.87491E-02	-2.56378E+06	-1.70741E+07
17	4.06613E-02	1.23228E-02	3.87491E-02	-2.56378E+06	1.70741E+07
18	1.29061E-03	1.16046E-03	-5.64810E-04	2.64604E+04	1.79415E+07
19	1.29061E-03	1.16046E-03	5.64810E-04	2.64604E+04	-1.79415E+07
20	1.21449E-04	1.21449E-04	9.24009E-15	3.91563E+05	-2.14240E+07

TABLE 2. EXTRACTED POLES -- DESCENDING RESIDUE MAGNITUDE.

	A			ALPHA	
	MAG	R	I	R	I
1	4.24372E-01	-1.69844E-01	-3.88902E-01	-1.51629E+06	4.80982E+06
2	4.24372E-01	-1.69844E-01	3.88902E-01	-1.51629E+06	-4.80982E+06
3	4.15409E-01	4.12118E-01	5.21797E-02	-1.06144E+06	-2.34744E+06
4	4.15409E-01	4.12118E-01	-5.21797E-02	-1.06144E+06	2.34744E+06
5	1.40992E-01	-7.90689E-02	-1.16735E-01	-5.85247E+07	-1.41588E+07
6	1.40992E-01	-7.90689E-02	1.16735E-01	-5.85247E+07	1.41588E+07
7	1.31866E-01	-1.03886E-01	-8.12177E-02	-1.82476E+06	-7.28018E+06
8	1.31866E-01	-1.03886E-01	8.12177E-02	-1.82476E+06	7.28018E+06
9	1.27327E-01	-1.26602E-01	1.35743E-02	-2.40366E+06	1.46561E+07
10	1.27327E-01	-1.26602E-01	-1.35743E-02	-2.40366E+06	-1.46561E+07
11	9.20747E-02	1.37488E-02	9.10424E-02	-2.24812E+06	-1.22084E+07
12	9.20747E-02	1.37488E-02	-9.10424E-02	-2.24812E+06	1.22084E+07
13	4.06613E-02	1.23228E-02	3.87491E-02	-2.56378E+06	1.70741E+07
14	4.06613E-02	1.23228E-02	-3.87491E-02	-2.56378E+06	-1.70741E+07
15	2.65871E-03	-2.65871E-03	5.66381E-14	-1.14465E+04	0.
16	1.29061E-03	1.16046E-03	5.64810E-04	2.64604E+04	-1.79415E+07
17	1.29061E-03	1.16046E-03	-5.64810E-04	2.64604E+04	1.79415E+07
18	8.01044E-04	7.18980E-04	-3.53184E-04	-2.11901E+06	9.70691E+06
19	8.01044E-04	7.18980E-04	3.53184E-04	-2.11901E+06	-9.70691E+06
20	1.21449E-04	1.21449E-04	9.24009E-15	3.91563E+05	-2.14240E+07

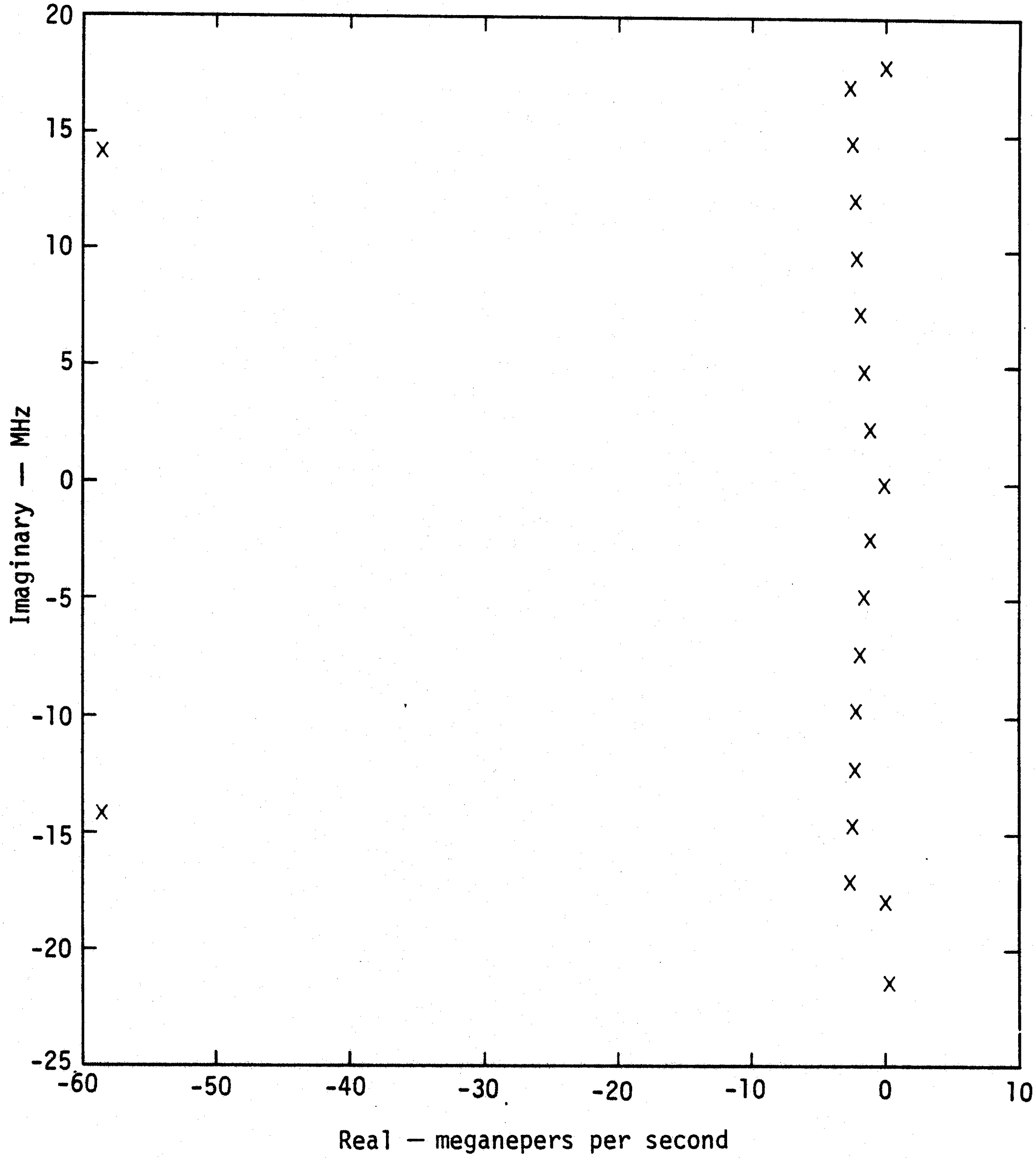

Figure 8. Locus of extracted poles in the complex plane.

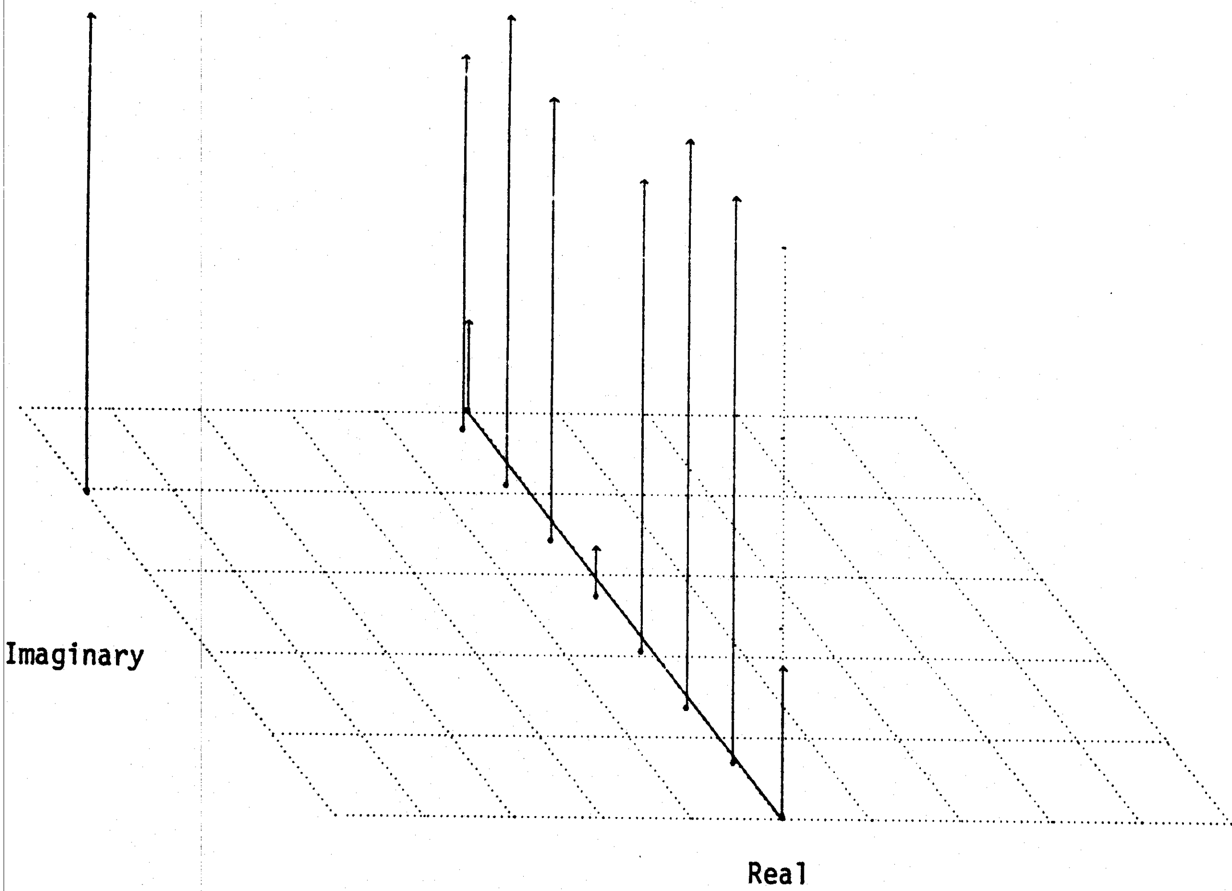

Figure 9. Three-dimensional view of pole locus in the complex plane with residue magnitude represented on Z-axis.

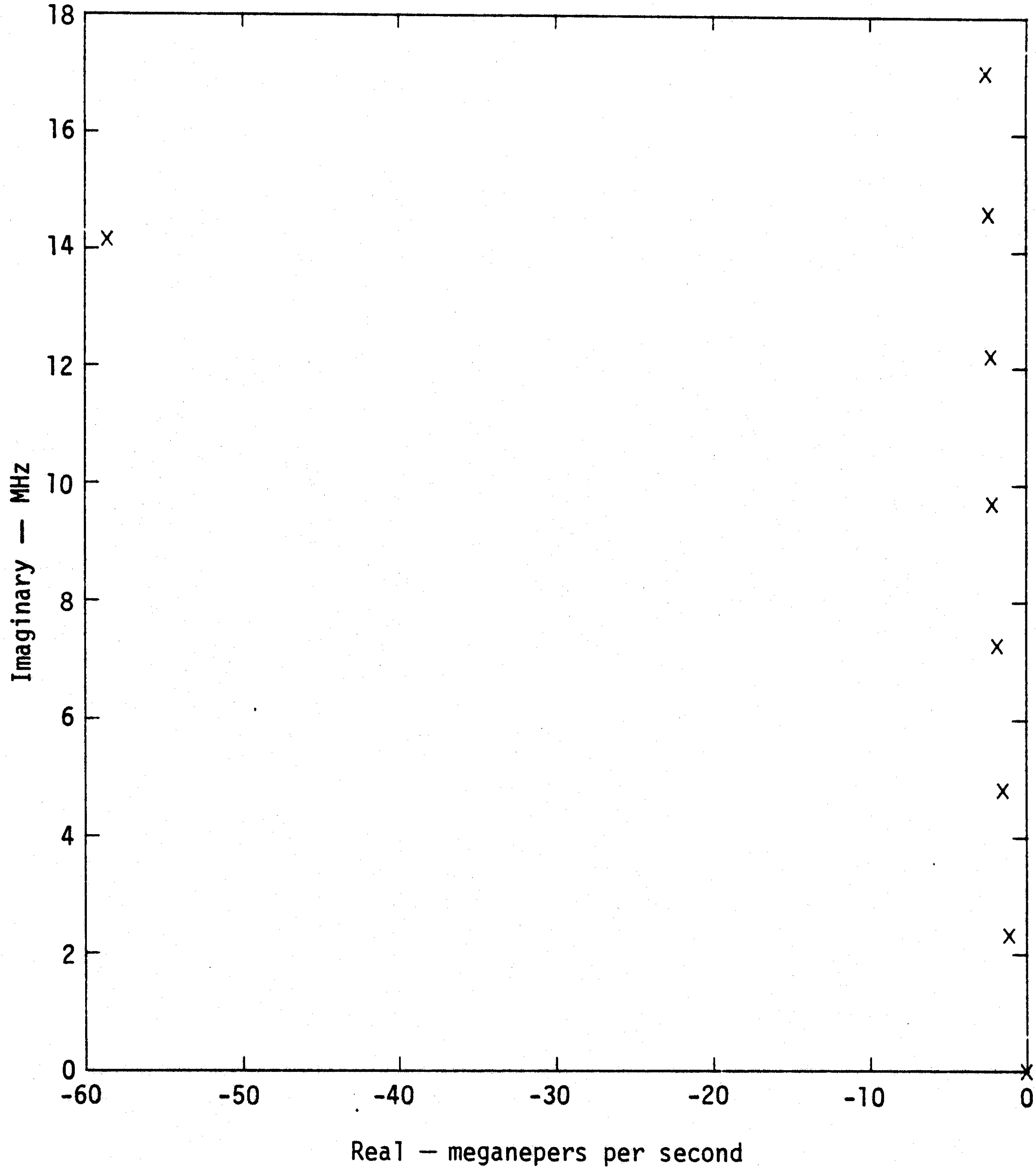

Figure 10. Locus of poles that meet the pole test criteria.

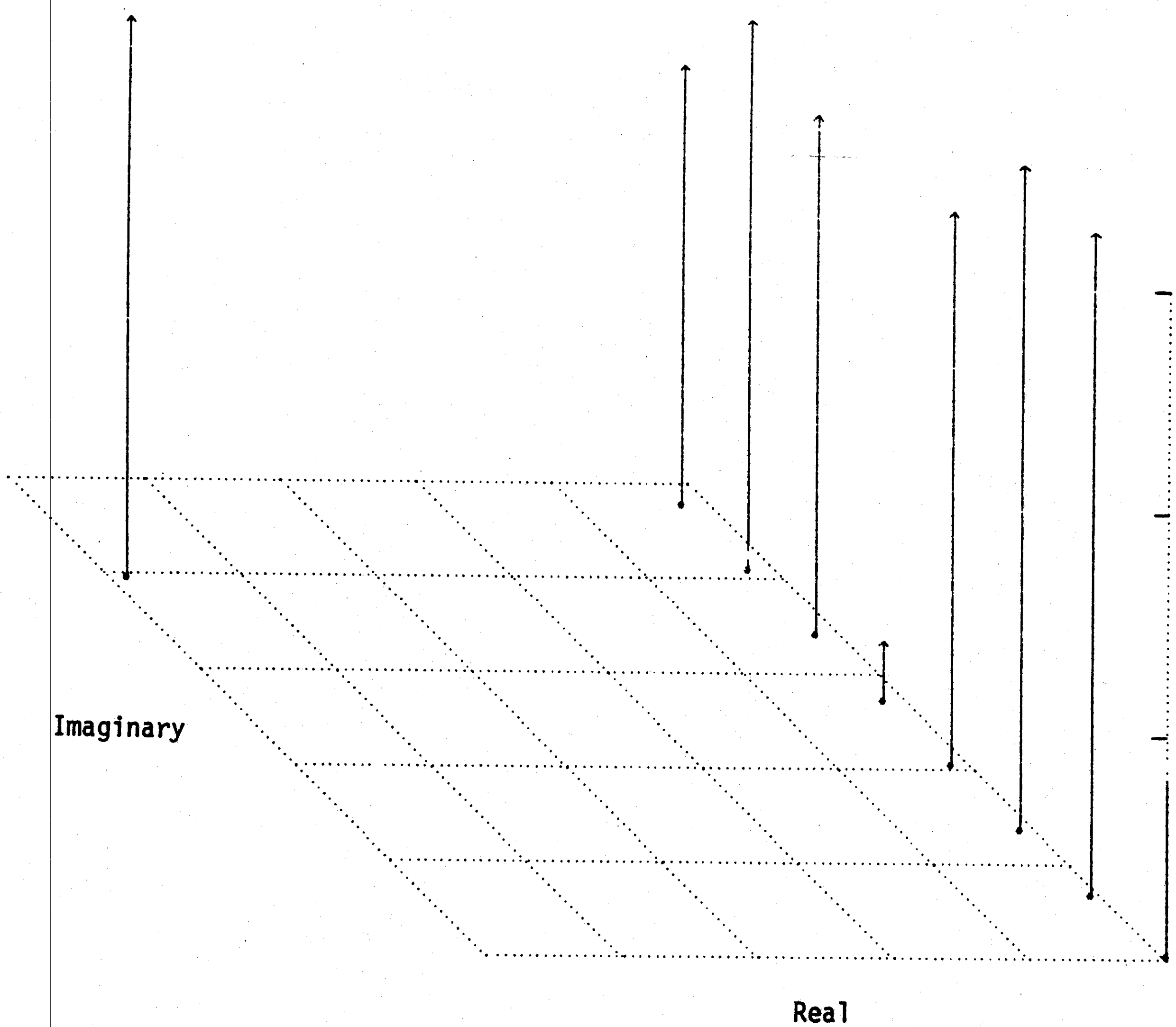

Figure 11. Three-dimensional plot of poles that meet the pole test criteria.

TABLE 3. POLES SATISFYING POLE TEST CRITERIA -- ASCENDING FREQUENCY ORDER.

		A		ALPHA	
	MAG	R	I	R	I
1	2.65871E-03	-2.65871E-03	5.66381E-14	-1.14466E+04	0.
2	4.15409E-01	4.12118E-01	5.21797E-02	-1.06144E+06	-2.34744E+06
3	4.15409E-01	4.12118E-01	-5.21797E-02	-1.06144E+06	2.34744E+06
4	4.24372E-01	-1.69844E-01	3.88902E-01	-1.51629E+06	-4.80982E+06
5	4.24372E-01	-1.69844E-01	-3.88902E-01	-1.51629E+06	4.80982E+06
6	1.31866E-01	-1.03886E-01	8.12177E-02	-1.82476E+06	7.28018E+06
7	1.31866E-01	-1.03886E-01	-8.12177E-02	-1.82476E+06	-7.28018E+06
8	8.01044E-04	7.18980E-04	-3.53184E-04	-2.11901E+06	9.70691E+06
9	8.01044E-04	7.18980E-04	3.53184E-04	-2.11901E+06	-9.70691E+06
10	9.20747E-02	1.37488E-02	-9.10424E-02	-2.24812E+06	1.22084E+07
11	9.20747E-02	1.37488E-02	9.10424E-02	-2.24812E+06	-1.22084E+07
12	1.40992E-01	-7.90689E-02	-1.16735E-01	-5.85247E+07	-1.41588E+07
13	1.40992E-01	-7.90689E-02	1.16735E-01	-5.85247E+07	1.41588E+07
14	1.27327E-01	-1.26602E-01	1.35743E-02	-2.40366E+06	1.46561E+07
15	1.27327E-01	-1.26602E-01	-1.35743E-02	-2.40366E+06	-1.46561E+07
16	4.06613E-02	1.23228E-02	3.87491E-02	-2.56378E+06	1.70741E+07
17	4.06613E-02	1.23228E-02	-3.87491E-02	-2.56378E+06	-1.70741E+07

TABLE 4. POLES SATISFYING POLE TEST CRITERIA -- DESCENDING RESIDUE MAGNITUDE.

		A		ALPHA	
	MAG	R	I	R	I
1	4.24372E-01	-1.69844E-01	-3.88902E-01	-1.51629E+06	4.80982E+06
2	4.24372E-01	-1.69844E-01	3.88902E-01	-1.51629E+06	-4.80982E+06
3	4.15409E-01	4.12118E-01	5.21797E-02	-1.06144E+06	-2.34744E+06
4	4.15409E-01	4.12118E-01	-5.21797E-02	-1.06144E+06	2.34744E+06
5	1.40992E-01	-7.90689E-02	-1.16735E-01	-5.85247E+07	-1.41588E+07
6	1.40992E-01	-7.90689E-02	1.16735E-01	-5.85247E+07	1.41588E+07
7	1.31866E-01	-1.03886E-01	-8.12177E-02	-1.82476E+06	-7.28018E+06
8	1.31866E-01	-1.03886E-01	8.12177E-02	-1.82476E+06	7.28018E+06
9	1.27327E-01	-1.26602E-01	1.35743E-02	-2.40366E+06	1.46561E+07
10	1.27327E-01	-1.26602E-01	-1.35743E-02	-2.40366E+06	-1.46561E+07
11	9.20747E-02	1.37488E-02	9.10424E-02	-2.24812E+06	-1.22084E+07
12	9.20747E-02	1.37488E-02	-9.10424E-02	-2.24812E+06	1.22084E+07
13	4.06613E-02	1.23228E-02	3.87491E-02	-2.56378E+06	1.70741E+07
14	4.06613E-02	1.23228E-02	-3.87491E-02	-2.56378E+06	-1.70741E+07
15	2.65871E-03	-2.65871E-03	5.66381E-14	-1.14465E+04	0.
16	8.01044E-04	7.18980E-04	-3.53184E-04	-2.11901E+06	9.70691E+06
17	8.01044E-04	7.18980E-04	3.53184E-04	-2.11901E+06	-9.70691E+06

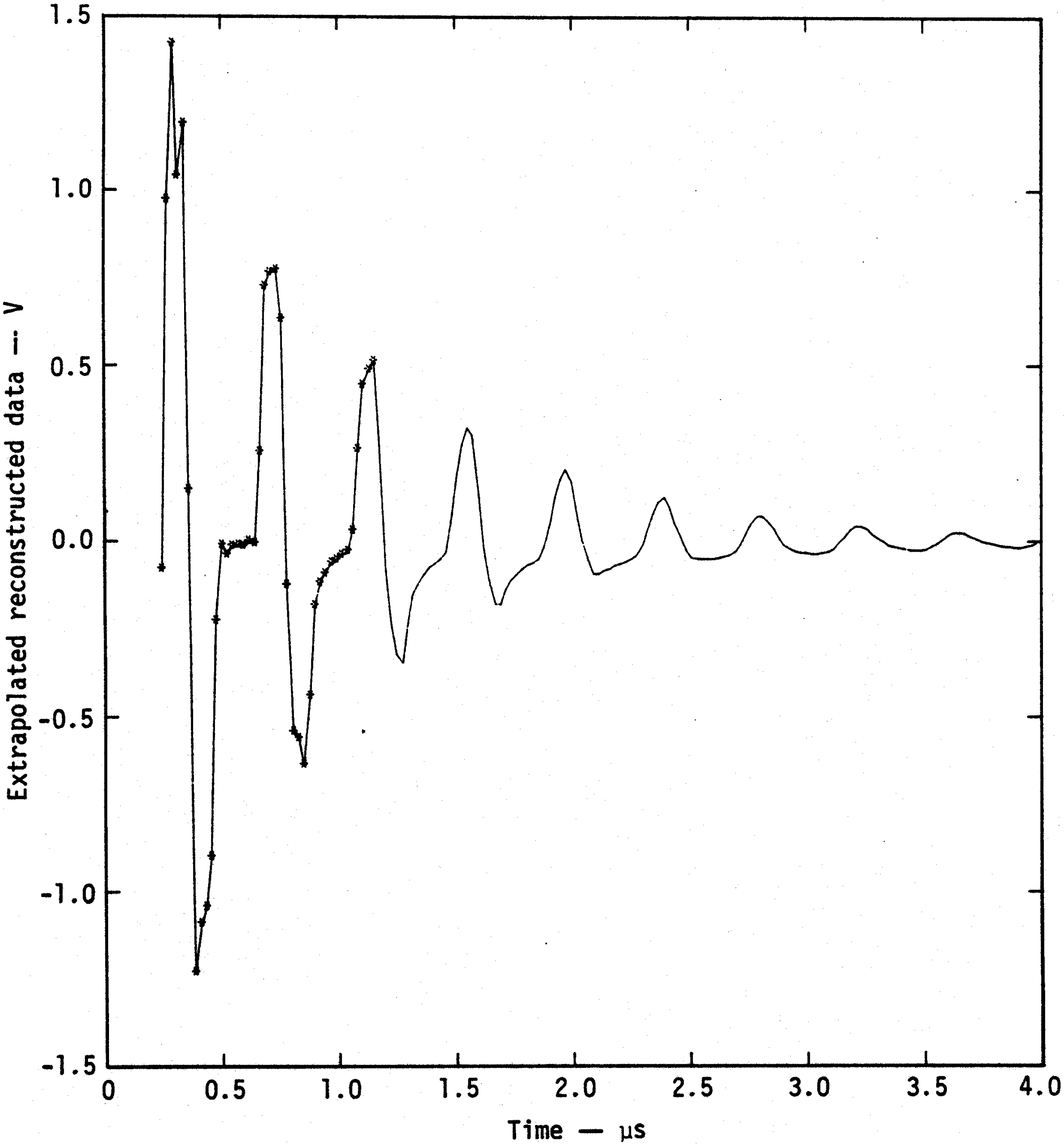

Figure 12. Reconstruction and extrapolation of time waveform using the poles that meet the pole test criteria.

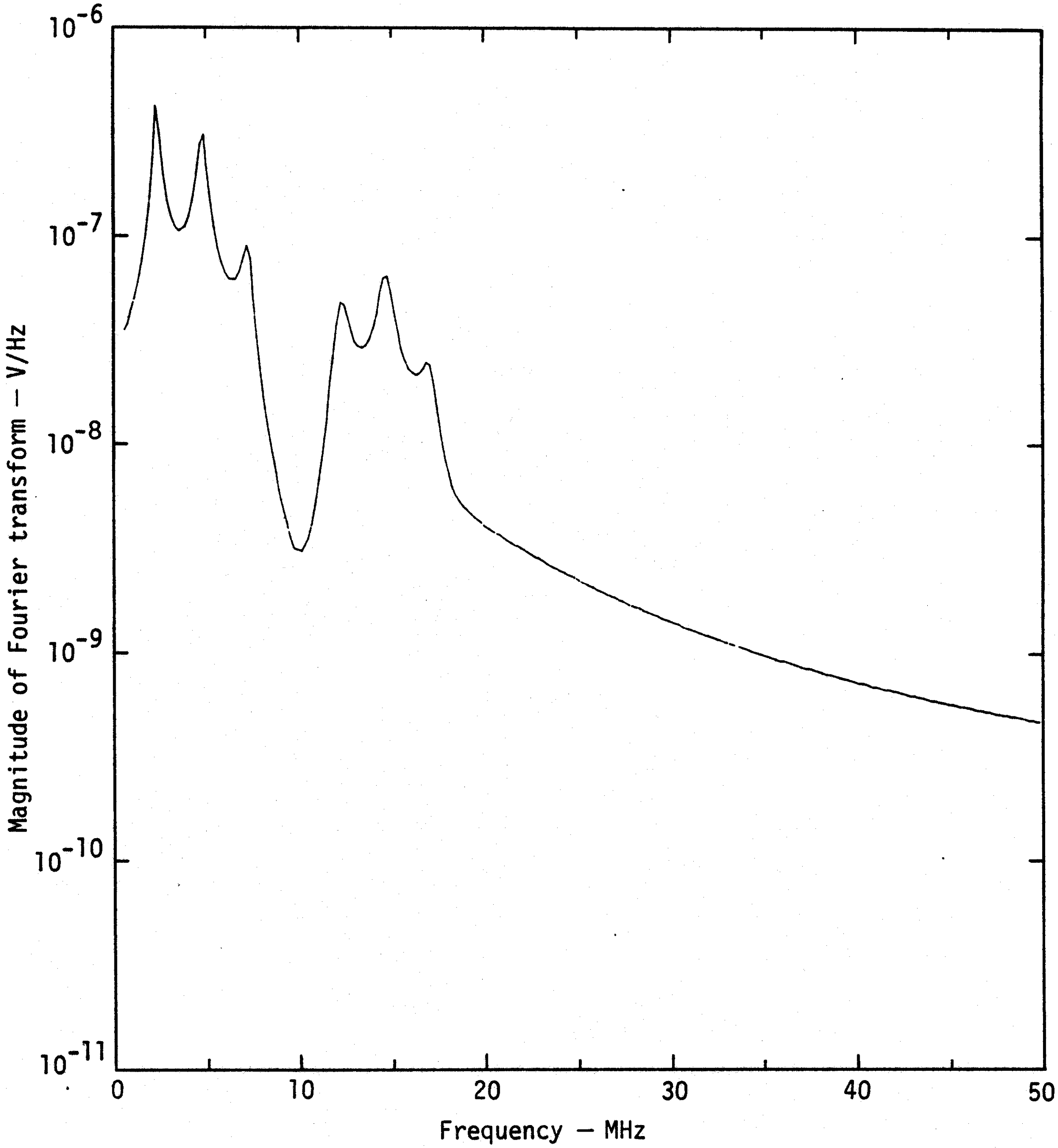

Figure 13. Amplitude spectrum of reconstructed waveform derived from Laplace transform using the poles that meet the pole test criteria.

is sufficiently small, however, to have a negligible effect on the reconstruction. Also, the peaks and valleys of the Laplace transform correspond to those of the Fourier transform of figure 5.

SAMPLE PROBLEM 2

The second sample problem involves a time waveform generated from a known set of poles chosen to have frequencies and residues somewhat representative of an electromagnetic structure. The poles (10 conjugate pairs) are listed in table 5, and the resulting complicated waveform and its spectrum are shown in figures 14 and 15. The data cards for the run were:

CARD1 - RUN 207 TEST PROBLEM WITH KNOWN POLES
CARD2 - TIMECAL = 320.
VCAL = 1.
CARD3 - NPOLES = 25
NBEGIN = 1
NPTS = 50
NDECI = 5
CARD4 - FMAX = .2
FLOW = 0
FHIGH = 0 (do not filter)
CARD5 - ITEST = 1
CARD6 - RES = 0
RHP = 0
PREAL = -.016
PIMAG = .2
CARD7 - FINISH = 1000.

The data points used in Prony's algorithm are indicated by asterisks in figure 16. The extracted poles that met the test criteria are listed in tables 6 and 7. Only two of the 25 requested poles were discarded; both had negligibly small residues, and one was located in the right half-plane. The locus of the poles is shown in figures 17 and 18. The one extraneous pole left after testing has a negligibly small residue The reconstructed and extrapolated time waveform is shown in figure 19, and the Laplace transform of the reconstructed waveform using the extracted poles is shown in figure 20.

TABLE 5. POLE SET USED TO GENERATE TIME WAVEFORM FOR SAMPLE PROBLEM 2.

A (residue)	α (damping coefficient)
0 ± j 1	-0.008 ± j 0.1/2π
0 ± j 0.9	-0.008 ± j 0.2/2π
0 ± j 0.008	-0.008 ± j 0.3/2π
0 ± j 0.7	-0.008 ± j 0.4/2π
0 ± j 0.6	-0.008 ± j 0.5/2π
0 ± j 0.005	-0.008 ± j 0.6/2π
0 ± j 0.4	-0.008 ± j 0.7/2π
0 ± j 0.003	-0.008 ± j 0.8/2π
0 ± j 0.2	-0.008 ± j 0.9/2π
0 ± j 0.1	-0.008 ± j 1.0/2π

TABLE 6. POLES EXTRACTED FROM WAVEFORM OF FIGURE 11 WHICH MEET POLE TEST CRITERIA -- ASCENDING FREQUENCY ORDER.

		A		ALPHA	
	MAG	R	I	R	I
1	1.10212E-01	-1.10212E-01	1.78192E-14	-2.98098E-10	0.
2	1.00000E+00	-4.72069E-08	-1.00000E+00	-8.00000E-03	1.59155E-02
3	1.00000E+00	-4.72069E-08	1.00000E+00	-8.00000E-03	-1.59155E-02
4	4.77794E-08	3.60855E-08	-3.13162E-08	-8.95737E-03	1.78504E-02
5	4.77793E-08	3.60857E-08	3.13159E-08	-8.95737E-03	-1.78504E-02
6	9.00000E-01	-7.31089E-09	-9.00000E-01	-8.00000E-03	3.18310E-02
7	9.00000E-01	-7.31091E-09	9.00000E-01	-8.00000E-03	-3.18310E-02
8	8.00001E-03	-7.79631E-09	-8.00001E-03	-8.00004E-03	4.77465E-02
9	8.00001E-03	-7.79625E-09	8.00001E-03	-8.00004E-03	-4.77465E-02
10	7.00000E-01	8.20306E-09	7.00000E-01	-8.00000E-03	-6.36620E-02
11	7.00000E-01	8.20275E-09	-7.00000E-01	-8.00000E-03	6.36620E-02
12	6.00000E-01	-1.59336E-09	-6.00000E-01	-8.00000E-03	7.95775E-02
13	6.00000E-01	-1.59342E-09	6.00000E-01	-8.00000E-03	-7.95775E-02
14	5.00000E-03	9.21562E-09	5.00000E-03	-7.99999E-03	-9.54930E-02
15	5.00000E-03	9.21567E-09	-5.00000E-03	-7.99999E-03	9.54930E-02
16	4.00000E-01	-2.29825E-08	-4.00000E-01	-8.00000E-03	1.11408E-01
17	4.00000E-01	-2.29825E-08	4.00000E-01	-8.00000E-03	-1.11408E-01
18	3.00001E-03	-1.82813E-08	3.00001E-03	-8.00003E-03	-1.27324E-01
19	3.00001E-03	-1.82813E-08	-3.00001E-03	-8.00003E-03	1.27324E-01
20	2.00000E-01	-3.14795E-08	2.00000E-01	-8.00000E-03	-1.43239E-01
21	2.00000E-01	-3.14797E-08	-2.00000E-01	-8.00000E-03	1.43239E-01
22	9.99826E-02	-8.35066E-08	9.99826E-02	-7.99985E-03	-1.59155E-01
23	9.99826E-02	-8.35069E-08	-9.99826E-02	-7.99985E-03	1.59155E-01

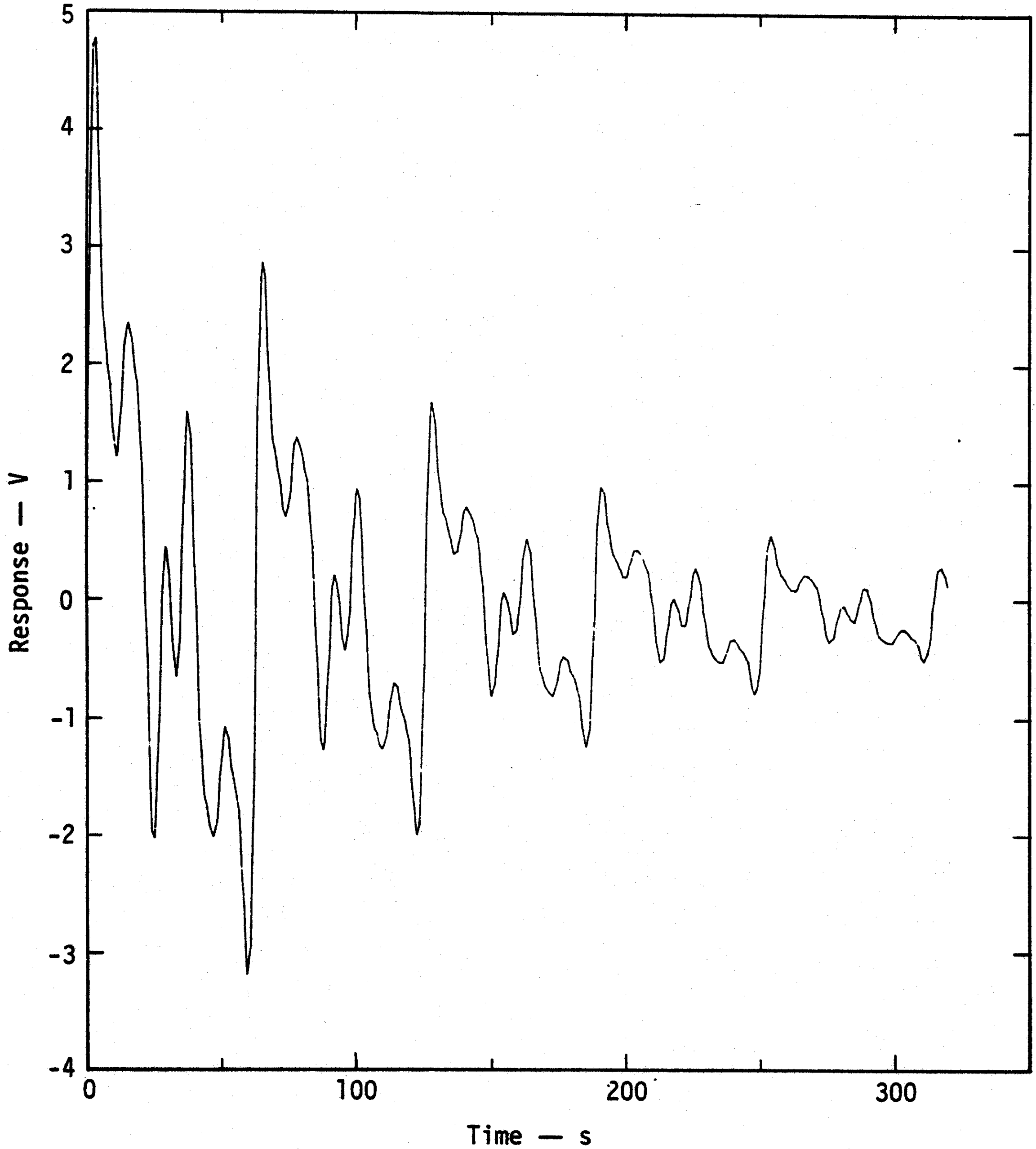

Figure 14. Time waveform for second sample problem generated from a known set of poles.

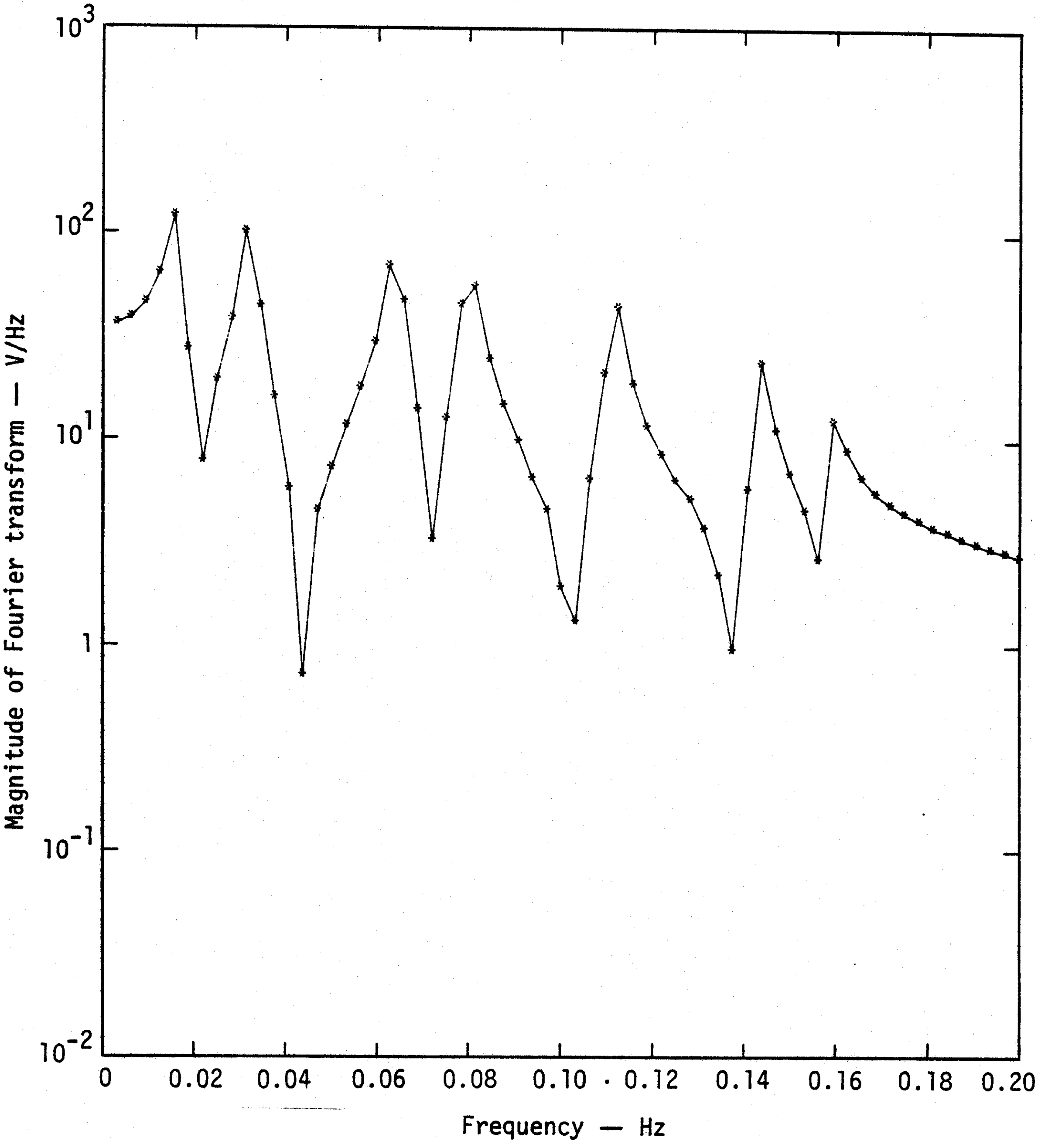

Figure 15. Amplitude spectrum of waveform of figure 14.

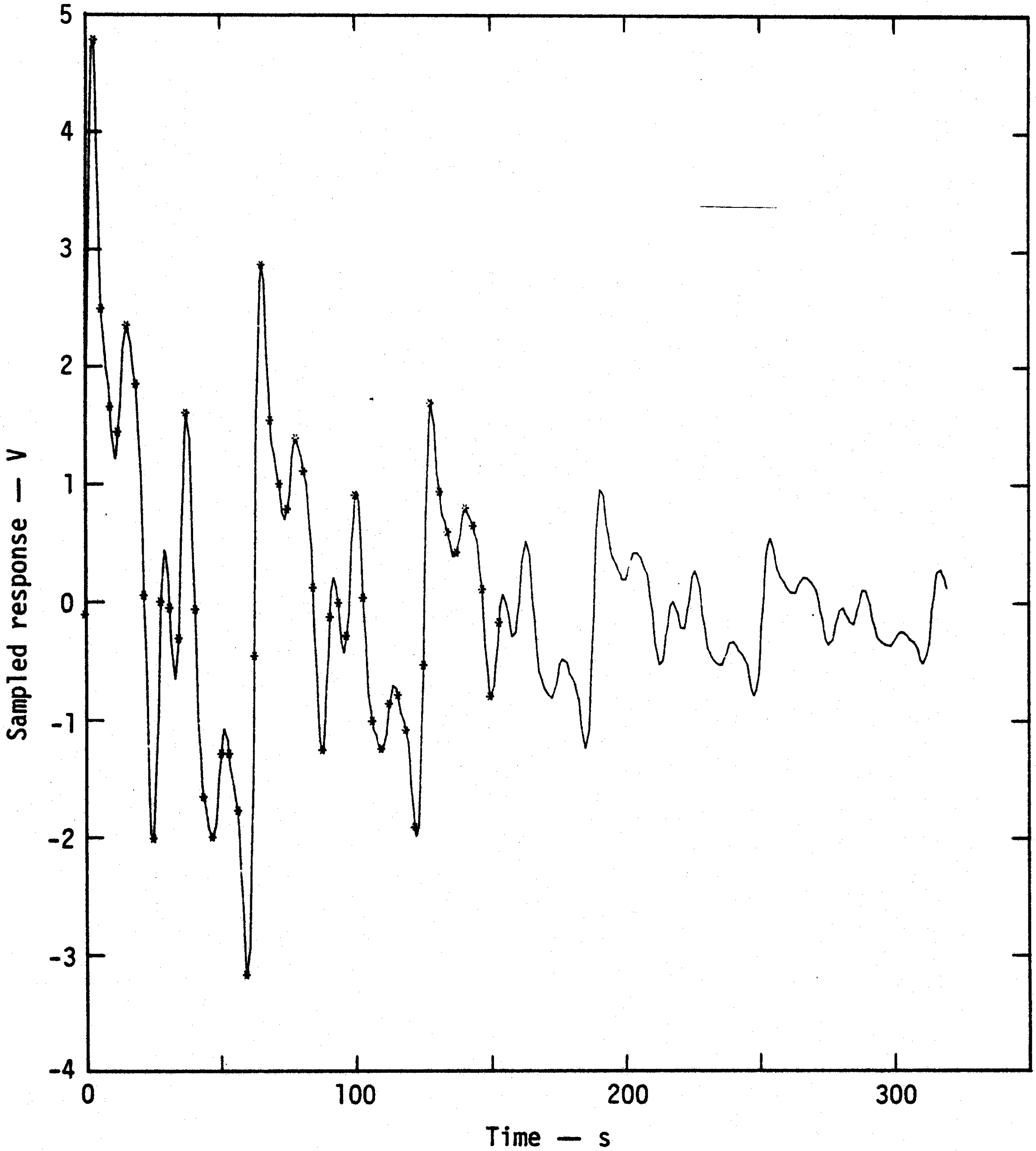

Figure 16. Data points used in Prony's algorithm.

TABLE 7. POLES EXTRACTED FROM WAVEFORM OF FIGURE 11 WHICH MEET POLE TEST CRITERIA -- DESCENDING RESIDUE MAGNITUDE.

		A		ALPHA	
	MAG	R	I	R	I
1	1.00000E+00	-4.72069E-08	1.00000E+00	-8.00000E-03	-1.59155E-02
2	1.00000E+00	-4.72069E-08	-1.00000E+00	-8.00000E-03	1.59155E-02
3	9.00000E-01	-7.31091E-09	9.00000E-01	-8.00000E-03	-3.18310E-02
4	9.00000E-01	-7.31089E-09	-9.00000E-01	-8.00000E-03	3.18310E-02
5	7.00000E-01	8.20275E-09	-7.00000E-01	-8.00000E-03	6.36620E-02
6	7.00000E-01	8.20306E-09	7.00000E-01	-8.00000E-03	-6.36620E-02
7	6.00000E-01	-1.59336E-09	-6.00000E-01	-8.00000E-03	7.95775E-02
8	6.00000E-01	-1.59342E-09	6.00000E-01	-8.00000E-03	-7.95775E-02
9	4.00000E-01	-2.29825E-08	-4.00000E-01	-8.00000E-03	1.11408E-01
10	4.00000E-01	-2.29825E-08	4.00000E-01	-8.00000E-03	-1.11408E-01
11	2.00000E-01	-3.14795E-08	2.00000E-01	-8.00000E-03	-1.43239E-01
12	2.00000E-01	-3.14797E-08	-2.00000E-01	-8.00000E-03	1.43239E-01
13	1.10212E-01	-1.10212E-01	1.78192E-14	-2.98098E-10	0.
14	9.99826E-02	-8.35069E-08	-9.99826E-02	-7.99985E-03	1.59155E-01
15	9.99826E-02	-8.35066E-08	9.99826E-02	-7.99985E-03	-1.59155E-01
16	8.00001E-03	-7.79625E-09	8.00001E-03	-8.00004E-03	-4.77465E-02
17	8.00001E-03	-7.79631E-09	-8.00001E-03	-8.00004E-03	4.77465E-02
18	5.00000E-03	9.21562E-09	5.00000E-03	-7.99999E-03	-9.54930E-02
19	5.00000E-03	9.21567E-09	-5.00000E-03	-7.99999E-03	9.54930E-02
20	3.00001E-03	-1.82813E-08	3.00001E-03	-8.00003E-03	-1.27324E-01
21	3.00001E-03	-1.82813E-08	-3.00001E-03	-8.00003E-03	1.27324E-01
22	4.77794E-08	3.60855E-08	-3.13162E-08	-8.95737E-03	1.78504E-02
23	4.77793E-08	3.60857E-08	3.13159E-08	-8.95737E-03	-1.78504E-02

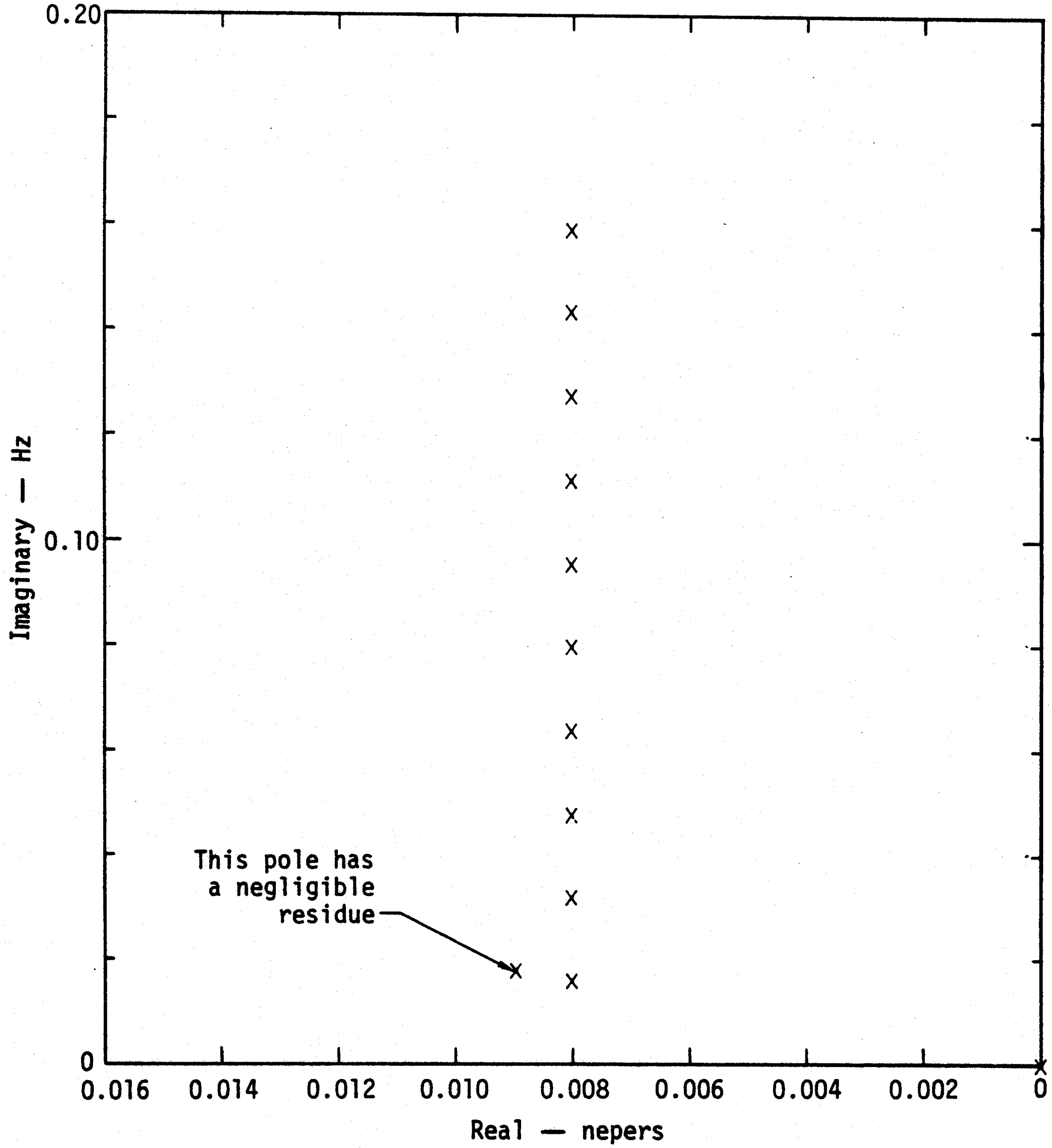

Figure 17. Locus of extracted poles in the complex plane.

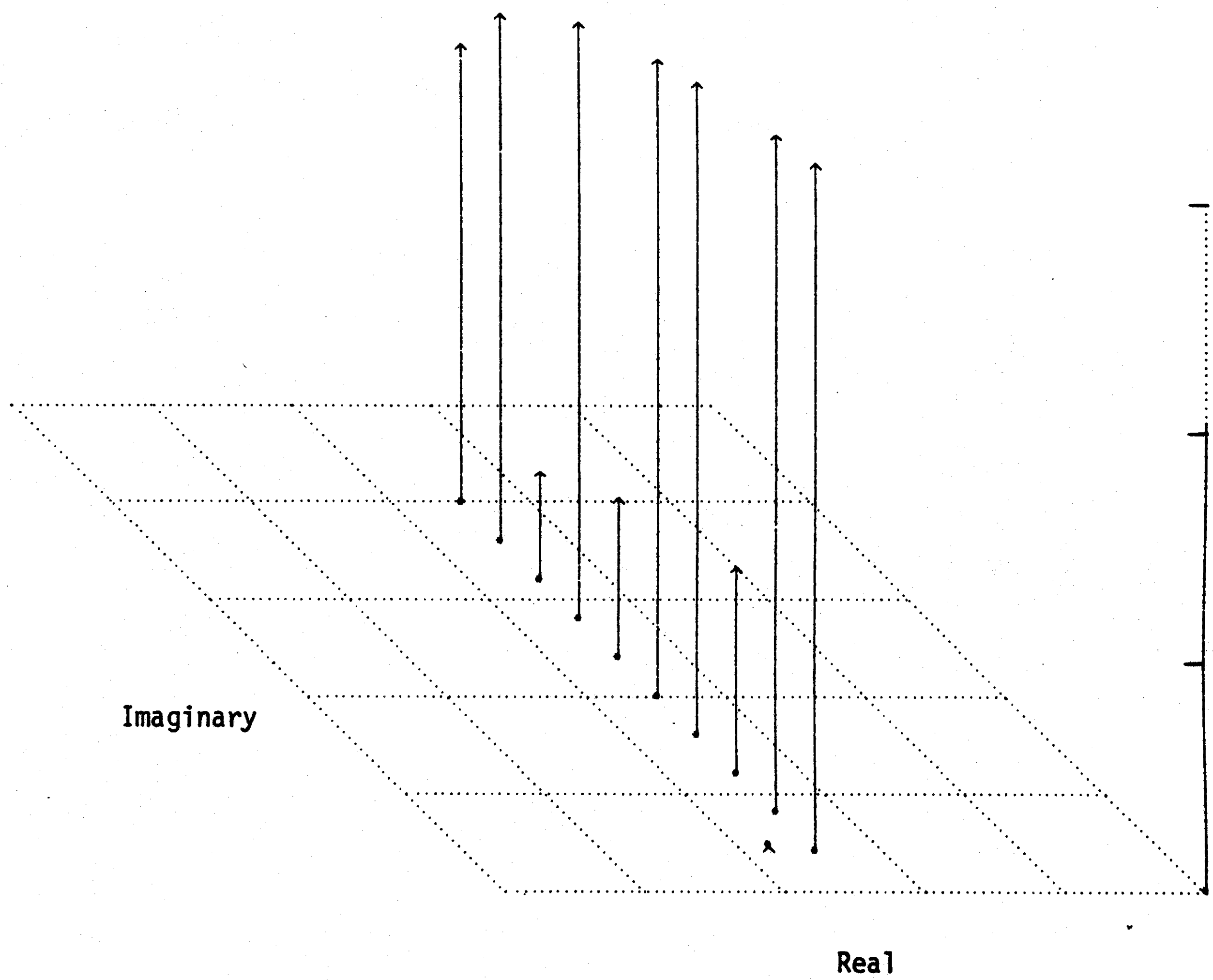

Figure 18. Three-dimensional plot of poles that meet the pole test criteria.

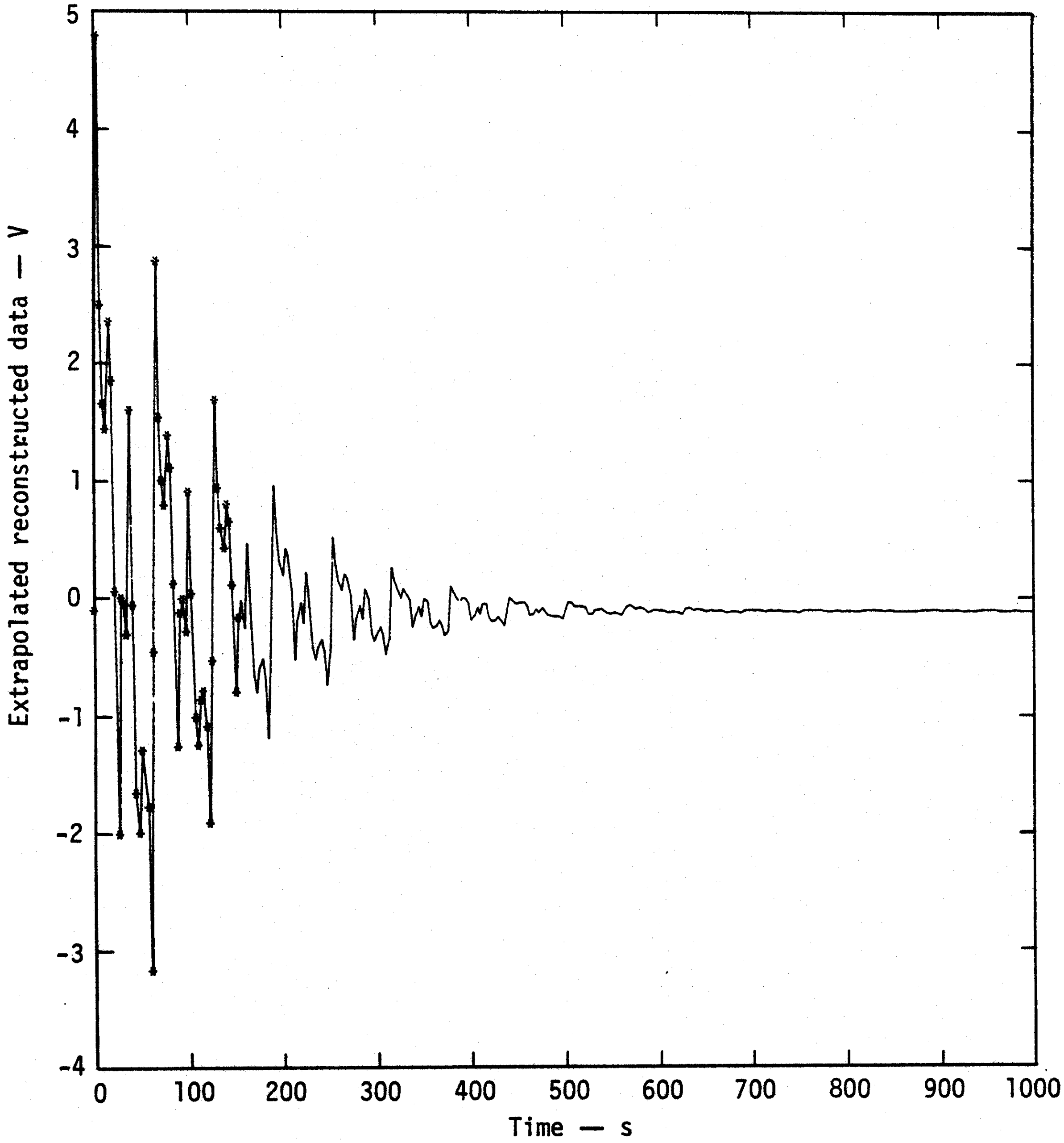

Figure 19. Reconstruction and extrapolation of time waveform using the poles that meet the test criteria.

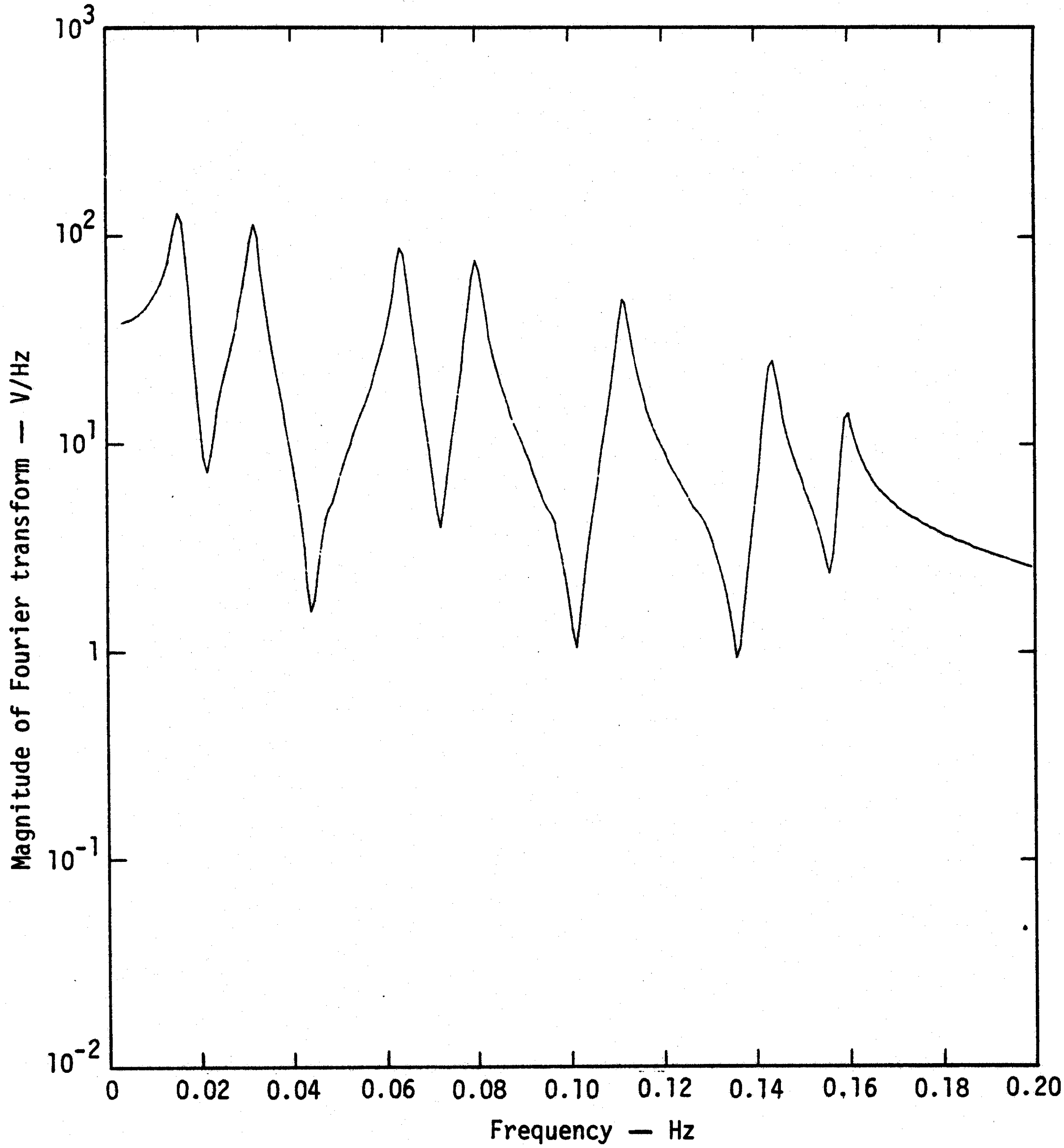

Figure 20. Amplitude spectrum of reconstructed waveform derived from Laplace transform using the poles that meet the pole test criteria.

SAMPLE PROBLEM 3

Figure 21 shows the asymmetric Y-dipole for which the current induced at point A by an incident pulse was obtained by two methods. In the first method, the LLL transient range (ref. 10) was used to obtain the waveform shown in figure 22, which is the voltage V_L across a 50-Ω load due to the current at point A in figure 21. The second method used the time-domain code WT-MBA/LLL1B to calculate the response at the same point for an incident Gaussian-shaped pulse with a vertical E-field. The computed response is shown in figure 23. The control cards for the analysis of the measured data were

```
CARD1 - RUN-225 ANALYSIS OF MEASURED DATA FOR Y-DIPOLE
CARD2 - TIMECAL = 20.E-9
        VCAL = 5.12
CARD3 - NPOLES = 25
        NBEGIN = 75
        NPTS = 50
        NDECI = 8
CARD4 - FMAX = 2.E9
        FLO = 0
        FHIGH = 1.5E9
CARD5 - ITEST = 1
CARD6 - RES = 0
        RHP = 0
        PREAL = -1.5E9
        PIMAG = 1.5E9
CARD7 = FINISH = 50.E-9
```

The control cards for the analysis of the computed data were

```
CARD1 - RUN-226 ANALYSIS OF COMPUTED RESPONSE OF Y-DIPOLE
CARD2 - TIMECAL = 51.1E-9
        VCAL = 1.
```

---

10. Deadrick, F. J., Miller, E. K., and Hudson, H. G., The LLL Transient Electromagnetics Measurement Facility, Lawrence Livermore Laboratory, Rept. UCRL-51933 (1975).

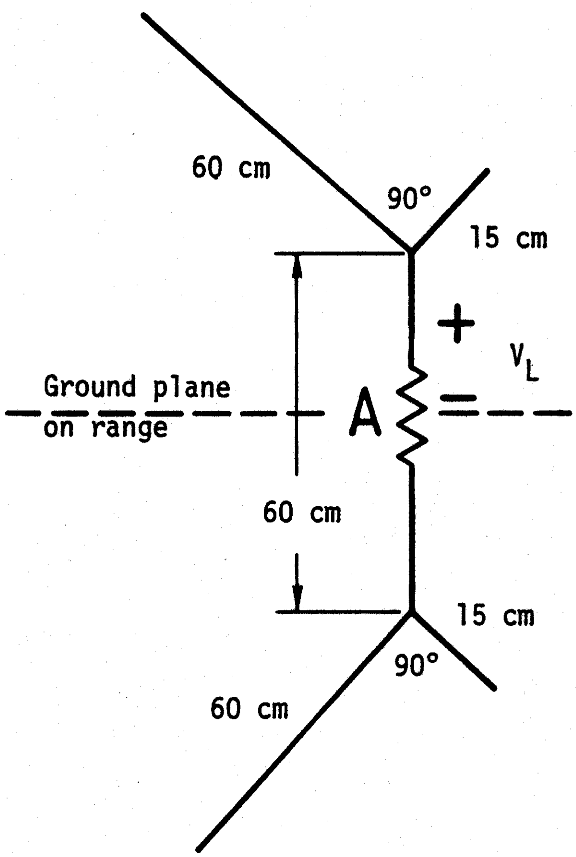

Figure 21. Asymmetric Y-dipole measured on the LLL transient range and computed by the time-domain code. Voltage V_L across a 50-Ω load at point A was determined.

```
CARD3 - NPOLES = 25
        NBEGIN = 53
        NPTS = 50
        NDECI = 3
CARD4 - FMAX = 2.E9
        FLO = 0
        FHIGH = 1.5E9
CARD5 - ITEST = 1
CARD6 - RES = 0
        RHP = 0
        PREAL = -1.5E9
        PIMAG = 1.5E9
CARD7 - FINISH = 50.E-9
```

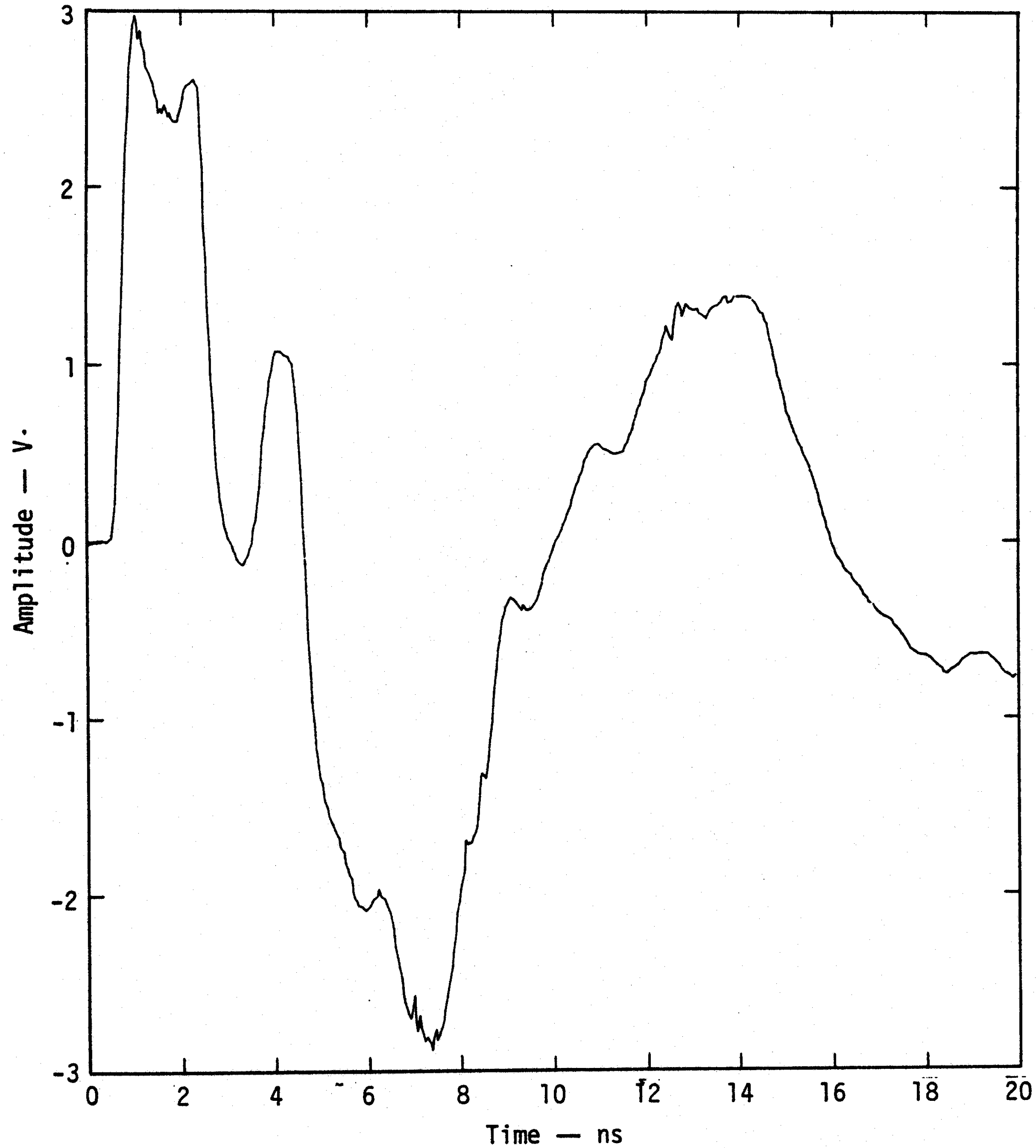

Figure 22. Response of asymmetric Y-dipole measured on LLL transient range.

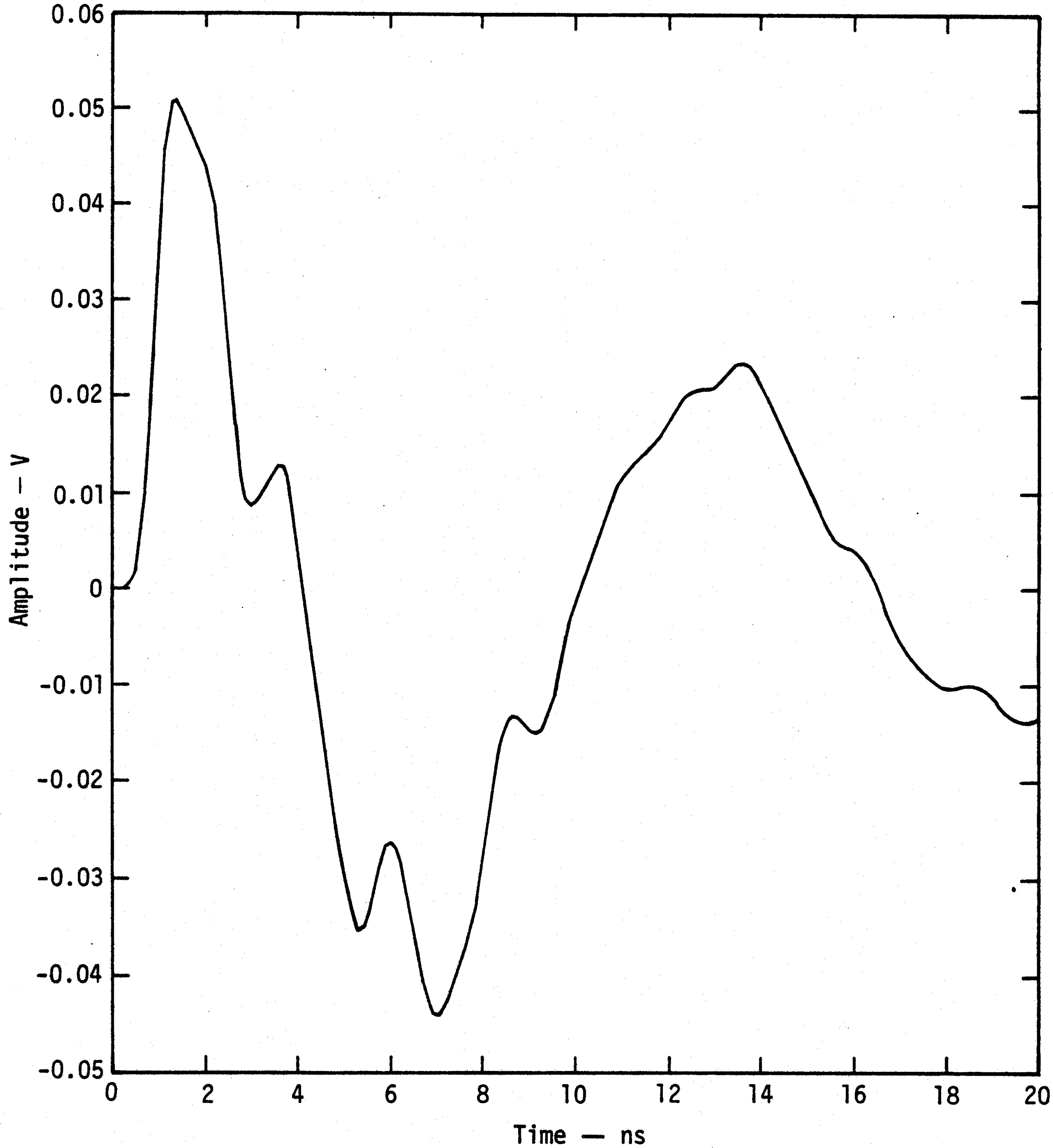

Figure 23. Computed time response of the asymmetric Y-dipole.

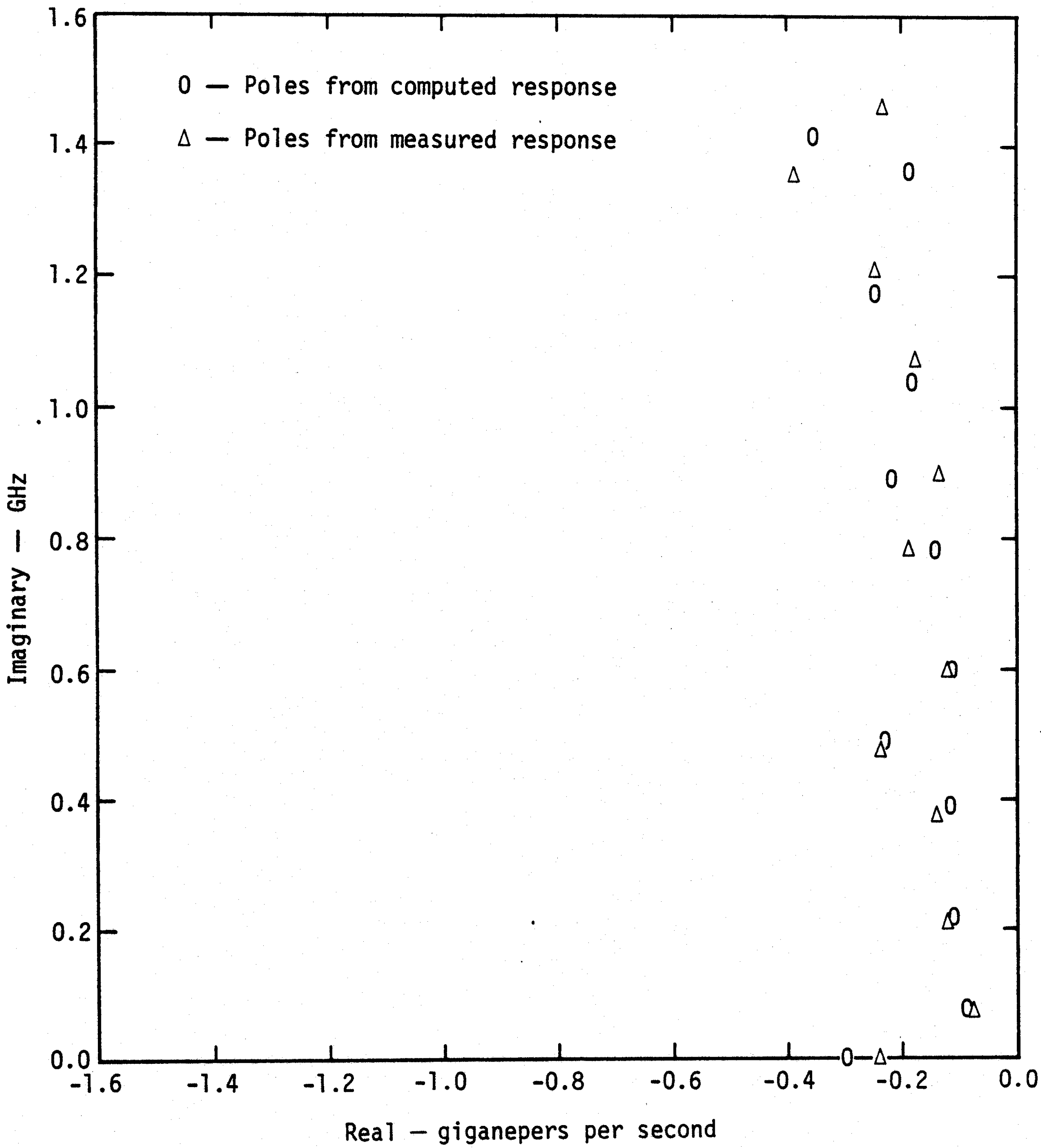

Figure 24. Comparison of poles extracted from measured and computed responses of Y-dipole. Agreement is very good for low-frequency high-residue poles.

The exact values for the poles satisfying the poles test criteria for the measured data are listed in tables 8 and 9, and the values for the poles for the computed data are listed in tables 10 and 11. The locus of the poles in the complex plane is shown in figure 24 where poles from the experimentally measured waveform are indicated by X and the poles from the computed waveform are indicated by 0. For the lower frequency poles, the agreement is quite good. The greater discrepancy at higher frequencies occurs because those poles have lower residues so their positions are perturbed more by noise in the experimental measurements. Figures 25 and 26 show an extrapolation of both the computed and experimental waveforms, respectively.

TABLE 8. POLES EXTRACTED FROM EXPERIMENTALLY MEASURED DATA -- ASCENDING FREQUENCY ORDER.

		A		ALPHA	
	MAG	R	I	R	I
1	3.38067E-01	-3.38067E-01	-1.15961E-11	-4.27076E+08	6.87394E-11
2	1.46535E+00	7.99755E-01	-1.22786E+00	-7.78100E+07	-8.10957E+07
3	1.46535E+00	7.99755E-01	1.22786E+00	-7.78100E+07	8.10957E+07
4	4.47951E-01	-2.96616E-01	3.35677E-01	-1.20830E+08	-2.10150E+08
5	4.47951E-01	-2.96616E-01	-3.35677E-01	-1.20830E+08	2.10150E+08
6	1.43280E-01	-1.16125E-01	-8.39281E-02	-1.51026E+08	3.84108E+08
7	1.43280E-01	-1.16125E-01	8.39281E-02	-1.51026E+08	-3.84108E+08
8	3.58019E-01	-1.47424E-01	-3.26258E-01	-2.23612E+08	-4.76463E+08
9	3.58019E-01	-1.47424E-01	3.26258E-01	-2.23612E+08	4.76463E+08
10	9.28356E-02	4.67759E-03	-9.27177E-02	-1.17042E+08	-6.00356E+08
11	9.28356E-02	4.67759E-03	9.27177E-02	-1.17042E+08	6.00356E+08
12	5.58969E-02	5.29218E-02	-1.79932E-02	-1.72825E+08	-7.84930E+08
13	5.58969E-02	5.29218E-02	1.79932E-02	-1.72825E+08	7.84930E+08
14	5.75979E-02	-4.68041E-02	-3.35694E-02	-1.67747E+08	8.99544E+08
15	5.75979E-02	-4.68041E-02	3.35694E-02	-1.67747E+08	-8.99544E+08
16	1.13725E-01	-1.03814E-01	4.64320E-02	-2.02310E+08	1.07246E+09
17	1.13725E-01	-1.03814E-01	-4.64320E-02	-2.02310E+08	-1.07246E+09
18	5.67586E-02	-3.67239E-02	4.32769E-02	-2.56980E+08	-1.22966E+09
19	5.67586E-02	-3.67239E-02	-4.32769E-02	-2.56980E+08	1.22966E+09
20	2.71043E-01	1.26642E-01	-2.39637E-01	-1.13075E+09	1.33845E+09
21	2.71043E-01	1.26642E-01	2.39637E-01	-1.13075E+09	-1.33845E+09

TABLE 9. POLES EXTRACTED FROM EXPERIMENTALLY MEASURED DATA -- DESCENDING RESIDUE MAGNITUDE.

	A			ALPHA	
	MAG	R	I	R	I
1	1.46535E+00	7.99755E-01	-1.22786E+00	-7.78100E+07	-8.10957E+07
2	1.46535E+00	7.99755E-01	1.22786E+00	-7.78100E+07	8.10957E+07
3	4.47951E-01	-2.96616E-01	-3.35677E-01	-1.20830E+08	2.10150E+08
4	4.47951E-01	-2.96616E-01	3.35677E-01	-1.20830E+08	-2.10150E+08
5	3.58019E-01	-1.47424E-01	3.26258E-01	-2.23612E+08	4.76463E+08
6	3.58019E-01	-1.47424E-01	-3.26258E-01	-2.23612E+08	-4.76463E+08
7	3.38067E-01	-3.38067E-01	-1.15961E-11	-4.27076E+08	6.87394E-11
8	2.71043E-01	1.26642E-01	-2.39637E-01	-1.13075E+09	1.33845E+09
9	2.71043E-01	1.26642E-01	2.39637E-01	-1.13075E+09	-1.33845E+09
10	1.43280E-01	-1.16125E-01	-8.39281E-02	-1.51026E+08	3.84108E+08
11	1.43280E-01	-1.16125E-01	8.39281E-02	-1.51026E+08	-3.84108E+08
12	1.13725E-01	-1.03814E-01	4.64320E-02	-2.02310E+08	1.07246E+09
13	1.13725E-01	-1.03814E-01	-4.64320E-02	-2.02310E+08	-1.07246E+09
14	9.28356E-02	4.67759E-03	-9.27177E-02	-1.17042E+08	-6.00356E+08
15	9.28356E-02	4.67759E-03	9.27177E-02	-1.17042E+08	6.00356E+08
16	5.75979E-02	-4.68041E-02	-3.35694E-02	-1.67747E+08	8.99544E+08
17	5.75979E-02	-4.68041E-02	3.35694E-02	-1.67747E+08	-8.99544E+08
18	5.67586E-02	-3.67239E-02	-4.32769E-02	-2.56980E+08	1.22966E+09
19	5.67586E-02	-3.67239E-02	4.32769E-02	-2.56980E+08	-1.22966E+09
20	5.58969E-02	5.29218E-02	-1.79932E-02	-1.72825E+08	-7.84930E+08
21	5.58969E-02	5.29218E-02	1.79932E-02	-1.72825E+08	7.84930E+08

TABLE 10. POLES EXTRACTED FROM CALCULATED DATA -- ASCENDING FREQUENCY ORDER.

	A			ALPHA	
	MAG	R	I	R	I
1	2.06732E-02	-1.35934E-02	1.55757E-02	-8.14632E+07	7.90177E+07
2	2.06732E-02	-1.35934E-02	-1.55757E-02	-8.14632E+07	-7.90177E+07
3	1.59487E-03	6.82541E-04	1.44144E-03	-1.08156E+08	2.16474E+08
4	1.59487E-03	6.82541E-04	-1.44144E-03	-1.08156E+08	-2.16474E+08
5	2.47167E-03	-9.50684E-04	2.28152E-03	-1.14717E+08	-3.92168E+08
6	2.47167E-03	-9.50684E-04	-2.28152E-03	-1.14717E+08	3.92168E+08
7	2.97438E-03	-2.94500E-03	4.17007E-04	-2.25846E+08	-4.87550E+08
8	2.97438E-03	-2.94500E-03	-4.17007E-04	-2.25846E+08	4.87550E+08
9	2.67607E-04	2.65934E-04	2.98750E-05	-1.13599E+08	-6.01277E+08
10	2.67607E-04	2.65934E-04	-2.98750E-05	-1.13599E+08	6.01277E+08
11	1.10685E-03	-6.03391E-04	9.27919E-04	-1.40301E+08	7.88432E+08
12	1.10685E-03	-6.03391E-04	-9.27919E-04	-1.40301E+08	-7.88432E+08
13	3.02630E-04	1.67907E-04	2.51778E-04	-2.20069E+08	8.96172E+08
14	3.02630E-04	1.67907E-04	-2.51778E-04	-2.20069E+08	-8.96172E+08
15	1.01762E-04	-9.63486E-05	3.27473E-05	-1.67738E+08	-1.04299E+09
16	1.01762E-04	-9.63486E-05	-3.27473E-05	-1.67738E+08	1.04299E+09
17	2.57581E-04	-2.17167E-04	1.38515E-04	-2.43521E+08	1.17758E+09
18	2.57581E-04	-2.17167E-04	-1.38515E-04	-2.43521E+08	-1.17758E+09
19	5.55235E-05	3.86211E-05	-3.98907E-05	-3.30573E+08	1.38052E+09
20	5.55235E-05	3.86211E-05	3.98907E-05	-3.30573E+08	-1.38052E+09

TABLE 11. POLES EXTRACTED FROM CALCULATED DATA -- DESCENDING RESIDUE MAGNITUDE.

		A		ALPHA	
	MAG	R	I	R	I
1	2.06732E-02	-1.35934E-02	-1.55757E-02	-8.14632E+07	-7.90177E+07
2	2.06732E-02	-1.35934E-02	1.55757E-02	-8.14632E+07	7.90177E+07
3	2.97438E-03	-2.94500E-03	4.17007E-04	-2.25846E+08	-4.87550E+08
4	2.97438E-03	-2.94500E-03	-4.17007E-04	-2.25846E+08	4.87550E+08
5	2.47167E-03	-9.50684E-04	-2.28152E-03	-1.14717E+08	3.92168E+08
6	2.47167E-03	-9.50684E-04	2.28152E-03	-1.14717E+08	-3.92168E+08
7	1.59487E-03	6.82541E-04	1.44144E-03	-1.08156E+08	2.16474E+08
8	1.59487E-03	6.82541E-04	-1.44144E-03	-1.08156E+08	-2.16474E+08
9	1.10685E-03	-6.03391E-04	9.27919E-04	-1.40301E+08	7.88432E+08
10	1.10685E-03	-6.03391E-04	-9.27919E-04	-1.40301E+08	-7.88432E+08
11	3.02630E-04	1.67907E-04	2.51778E-04	-2.20069E+08	8.96172E+08
12	3.02630E-04	1.67907E-04	-2.51778E-04	-2.20069E+08	-8.96172E+08
13	2.67607E-04	2.65934E-04	2.98750E-05	-1.13599E+08	-6.01277E+08
14	2.67607E-04	2.65934E-04	-2.98750E-05	-1.13599E+08	6.01277E+08
15	2.57581E-04	-2.17167E-04	1.38515E-04	-2.43521E+08	1.17758E+09
16	2.57581E-04	-2.17167E-04	-1.38515E-04	-2.43521E+08	-1.17758E+09
17	1.01762E-04	-9.63486E-05	3.27473E-05	-1.67738E+08	-1.04299E+09
18	1.01762E-04	-9.63486E-05	-3.27473E-05	-1.67738E+08	1.04299E+09
19	5.55235E-05	3.86211E-05	3.98907E-05	-3.30573E+08	-1.38052E+09
20	5.55235E-05	3.86211E-05	-3.98907E-05	-3.30573E+08	1.38052E+09

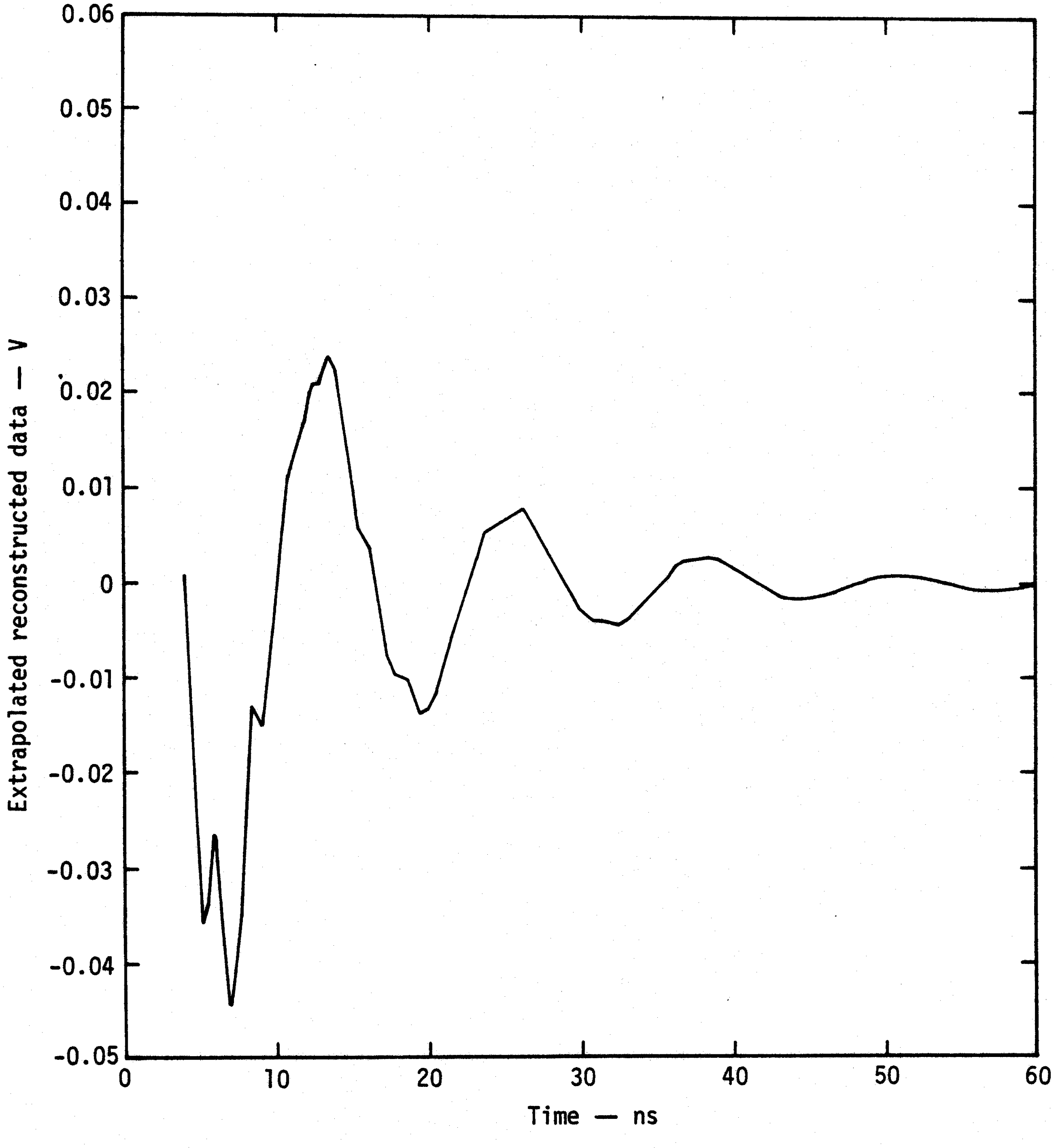

Figure 25. Extrapolation of time waveform using poles extracted from computed Y-dipole response. Compare with figure 23 for $t \geq 0.4 \times 10^{-8}$ s.

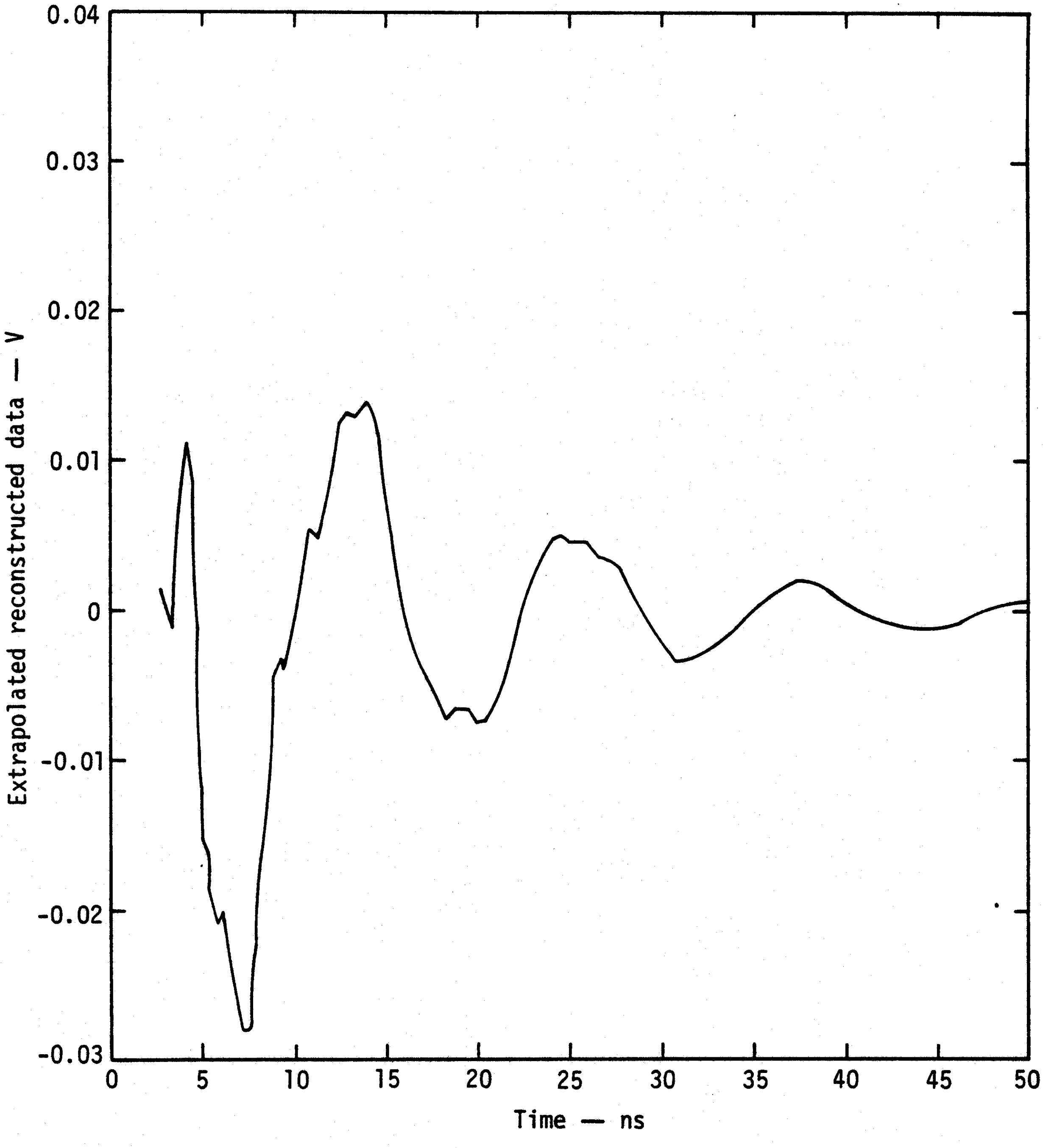

Figure 26. Extrapolation of time waveform using poles extracted from the measured Y-dipole response. Compare with Figure 22 for $t \geq 0.3 \times 10^{-8}$ s.

APPENDIX: LISTING OF SEMPEX

```
OBOX,012.
OPRINT,LIST.
XLATE,I=DATA,O=SIGNAL,AS.
RUN,SC.
RFL,125000.
MAKE,BINARY.
LINK,F,B=BINARY,CS=ZERO,LO.
BINARY.
]ER
      PROGRAM SEMPEX(SIGNAL,TAPE1=SIGNAL,INPUT,TAPE2=INPUT,
     .OUTPUT,TAPE3=OUTPUT,TAPE99)
      DIMENSION XTIME(512),RESP(256),RESPIN(512),RCNSTR(512),TIME(2048)
      DIMENSION RBLDR(2048),XTIME1(512),ERROR(512),ERRORS(512)
      COMPLEX A,ALPHA,RESPR,EXTRAP,FFT,CRESP
      DIMENSION  A(35), ALPHA(35), RESPR(512),EXTRAP(2048),CRESP(256)
      DIMENSION FREQ(256),FFTMAG(256),  FFT(256)
      DIMENSION FFTRSP(1024),SINES(128)
C     DIMENSION XTIT1(4),YTIT1(4),YTIT2(4),TIT1(8),TIT2(8)
      DIMENSION POLESR(35),POLESI(35),RUNTIT(8)
C    . YTIT3(4),TIT3(8),XTIT2(4),YTIT4(4),TIT4(8),TIT5(8)
C    ..YTIT5(4),TIT6(8),YTIT6(4),TIT7(8),YTIT7(4),TIT8(8)
C    .XTIT3(4),YTIT8(4),TIT9(8),XTIT4(4),TIT10(8),TIT11(8)
      ECUIVALENCE(ERRORS,ERROR)
C     DATA (XTIT1=40H      TIME IN SECONDS                     ),
C    .(YTIT1=40H        RESPONSE                           ),
C    .(YTIT2=40H       SAMPLED RESPONSE                    ),
C    .(YTIT3=40H      RECONST DATA                         ),
C    .(TIT1=40H       ORIGINAL DATA                        ),
C    .(TIT2=40H     SAMPLED DATA                           ),
C    .(TIT3=40H   RECONSTRUCTED TIME RESPONSE              ),
C    .(TIT4=40H        UN-NORMALIZED POLES                 ),
C    .(TIT5=40H     UN-NORMALIZED POLES UPPER L.H.P.       ),
C    .(YTIT4=40H           IMAGINARY (HZ)                  ),
C    .(XTIT2=40H            REAL (NEPERS)                  )


C     DATA(TIT6=40H ERR. BET. RECONSTRUCTED DATA AND INPUT ),
C    .(YTIT5=40H           ERROR(RECONST-INPUT)            ),
C    .(YTIT6=40H           EXTRAP RECONST DATA             ),
C    .(TIT7=40H ---EXTRAPOLATED DATA ***SAMPLED DATA       ),
C    .(YTIT7=40H         ERROR (EXTRAP.-INPUT)             ),
C    .(TIT8=40HERROR BET. EXTRAP. DATA & SAMPLED DATA      ),
C    .(XTIT3=40H        FREQUENCY IN HZ                    ),
C    .(YTIT8=40H  MAGNITUDE OF FOURIER TRANSFORM (V/HZ)    ),
C    .(TIT9=40HSPECTRUM DERIVED FROM POLES & RESIDUALS     ),
C    .(TIT10=40HFFT OF THE INPUT DATA                      ),
C    .(TIT11=40H        TIME RESPONSE AFTER FILTERING      )


C------ FIRST CARD
C---------- THIS CARD IS FOR A RUN DESCRIPTION.

      READ(2,400) RUNTIT
  400 FORMAT(1X,8A10)

C------SECOND CARD
C---------- READ IN THE TOTAL TIME DURATION OF THE DATA IN SECONDS
C           AND A VOLTAGE CALIBRATION FACTOR.

      READ(2,401) TMAX,VCAL
  401 FORMAT(1X,2F15.0)

C------THIRD CARD
C---------- READ IN NPOLES,NBEGIN,NPTS,NDECI
```

```
C      -NPOLES-  IS THE INTEGER NUMBER OF POLES DESIRED TO FIT THE DATA.
C       NPOLES CANNOT BE GREATER THAN 35.

C       -NBEGIN-  AN INTEGER NUMBER DEFINING THE FIRST DATA POINT IN THE
C       INPUT DATA ARRAY.(NBEGIN NEVER LESS THAN OR EQUAL TO ZERO)

C       -NPTS-  THE NUMBER OF DATA POINTS TO BE FIT.  FOR AN EXACT FIT
C              NPTS=2 X NPOLES.

C      -NDECI-  AN INTEGER NUMBER WHICH DETERMINES HOW MUCH THE
C      INPUT DATA IS SAMPLED.  FOR EXAMPLE: IF NBEGIN=1
C      AND NDECI=4 THEN EVERY FOURTH DATA POINT IN THE INPUT DATA ARRAY
C      WILL BE SAMPLED.

      READ(2,402) NPOLES,NBEGIN,NPTS,NDECI
 402  FORMAT(1X,4I5)

C------ FOURTH CARD
C---------- THIS CARD CONTROLS THE TRUNCATION FILTER.
C---------- FMAX,FLOW,FHIGH

C      -FMAX- MAXIMUM FREQUENCY PLOTTED FOR FREQ. DOMAIN PLOTS.

C      -FLOW-  THIS SPECIFIES THE LOW FREQUENCY CUTOFF POINT.

C      -FHIGH- THIS SPECIFIES THE HIGH FREQUECY CUTOFF POINT.
C              IF FHIGH IS 0. THEN THE INPUT DATA WILL NOT BE
C              FILTERED.

C     NOTE-----  THIS FILTER WORKS BEST IF THE CUTOFF POINTS CHOSEN ARE
C                PICKED TO COINCIDE WITH A NATURAL NULL IN THE
C                FREQUENCY SPECTRUM.  IF NO OBVIOUS NULL EXISTS THEN
C                THE CUTOFF POINTS MUST BE APPROXIMATELY 2 ORDERS OF
C                MAGNITUDE DOWN FROM THE PEAK VALUES IN THE SPECTRUM.


      READ(2,403) FMAX,FLOW,FHIGH
 403  FORMAT(1X,3F12.0)

C------FIFTH CARD
C             THIS CARD CONTROLS THE AMOUNT OF OUTPUT GENERATED
C             BY THE SEMPEX CODE.
C             TO OBTAIN A COMPLETE OUTPUT TYPE A  1 IN COLUMN 2.
C             TO OBTAIN A QUICK LOOK AT THE POLES WITH NO SUBSEQUENT
C             POLE TESTING LEAVE THIS CARD BLANK.
C             IF THE FIFTH CARD IS BLANK THEN THE SIXTH AND SEVENTH
C             CARDS ARE NOT REQUIRED.

      READ(2,432) ITEST
 432  FORMAT(1X,I5)

      IF(ITEST .EQ. 0) GO TO 222
C------  SIXTH CARD
C            THIS CARD CONTROLS THE POLE TESTING PARAMETERS
C      -RES-  ELIMINATES POLES WITH LOW RESIDUES AS COMPARED TO THE LARGEST
C      RESIDUE.  IE. RES=.001 MEANS THAT ALL POLES WHOSE RESIDUES
C      ARE LESS THAT .001 OF THE ABSOLUTE VALUE OF THE LARGEST RESIDUE WILL
C      BE ELIMINATED.  IF RES=0. NO POLES ARE ELIMINATED DUE TO WEAK
C      RESIDUES.

C      -RHP-   DEFINES WHICH REGION OF THE RIGHT HALF PLANE POLES ARF
C      ALLOWED.  THIS PARAMETER IS NORMALLY 0.
```

```
C        -PREAL-    DEFINES THE POINT ON THE -SIGMA AXIS POLES WILL BE
C        ELIMINATED. IE. IF PREAL= -1. AND A POLE HAS A DAMPING COEFFICIENT
C        OF -1.01 THE POLE WILL BE ELIMINATED.
C        THIS PARAMETER ALSO DEFINES THE -SIGMA AXIS FOR PLOTTING.

C        -PIMAG-  DEFINES WHICH POINT ON THE IMAGINARY AXIS ABOVE WHICH
C        POLES WILL BE ELIMINATED.  DEFINES THE IMAGINARY AXIS FOR PLOTTING.

       READ(2,433) RES,RHP,PREAL,PIMAG
  433  FORMAT(1X,4F12.0)

C------  CARD SEVEN
C            THIS CARD CONTROLS THE EXTRAPOLATION OF THE
C            RECONSTRUCTED DATA.
C        -FINISH-  ONCE THE POLES HAVE BEEN TESTED
C        THE REMAINING POLES ARE USED TO RECONSTRUCT/EXTRAPOLATE A TIME WAVEFORM
C        -FINISH-  DEFINES THE TIME IN SECONDS WHERE THIS RECONSTRUCTION
C        ENDS.

       READ(2,437) FINISH
  437  FORMAT(1X,E15.5)
  222  CONTINUE

C------ READ IN THE INPUT DATA

       READ(1,404) (RESPIN(I),RESPIN(I+1),I=1,512,2)
  404  FORMAT(1X,F12.7,12X,F12.7)
       TP = 8.*ATAN(1.)
       MAXPTS=(NPTS-1)*NDECI+NBEGIN
       DT=TMAX/511.
       DDT=NDECI*DT

C        CALCULATE THE HIGHEST FREQUENCY BASED ON NDECI AND
C        THE NYQUIST CRITERIA.

       FNYQ=1./(2.*NDECI*DT)

C------ RUN TESTS ON THE INPUT PARAMETERS

       IF(FHIGH .EQ. 0.) GO TO 227
       IF(FNYQ .GE. FHIGH) GO TO 227
       WRITE(3,438)
  438  FORMAT(*-ERROR- FHIGH IS GREATER THAN NYQUIST RATE*)
       CALL EXIT
  227  CONTINUE

       IF(NPOLES .LE. 35) GO TO 200
       WRITE(3,405)
  405  FORMAT(1X,*-ERROR-  NPOLES GREATER THAN 35.*)
       CALL EXIT
  200  CONTINUE

       IF(TMAX .GT. 0.) GO TO 201
       WRITE(3,406)
  406  FORMAT(1X,*-ERROR-  TMAX LESS THAN ZERO.*)
       CALL EXIT
  201  CONTINUE

       IF(2*NPOLES .LE. NPTS) GO TO 202
       WRITE(3,407)
  407  FORMAT(1X,*-ERROR-  2 X NPOLES GREATER THAN NPTS*)
       CALL EXIT
  202  CONTINUE
```

```
      IF(NBEGIN .LT. MAXPTS) GO TO 203
      WRITE(3,408)
408   FORMAT(1X,*-ERROR-  NBEGIN IS GREATER THAN MAXPTS.*)
      CALL EXIT
203   CONTINUE

      IF(NBEGIN .NE. 0) GO TO 204
      WRITE(3,410)
410   FORMAT(1X,*-ERROR-  NBEGIN EQUALS 0.*)
      CALL EXIT
204   CONTINUE

      IF(RHP .GE.0.)  GO TO 205
      WRITE(3,411)
411   FORMAT(1X,*-ERROR-  RHP IS NEGATIVE*)
      CALL EXIT
205   CONTINUE

      IF(RES .GE. 0.) GO TO 206
      WRITE(3,425)
425   FORMAT(1X,*-ERROR-  RES IS NEGATIVE*)
      CALL EXIT
206   CONTINUE

      IF(ITEST .EQ. 0) GO TO 207
      IF(PREAL .LT. 0.) GO TO 207
      WRITE(3,412)
412   FORMAT(1X,*-ERROR-  PREAL IS POSITIVE*)
      CALL EXIT
207   CONTINUE

      IF(NPOLES .GT. 0) GO TO 208
      WRITE(3,413)
413   FORMAT(1X,*-ERROR-  NPOLES IS ZERO OR NEGATIVE*)
      CALL EXIT
208   CONTINUE

      IF(ITEST .EQ. 0) GO TO 209
      IF(PIMAG .GT. 0.) GO TO 209
      WRITE(3,414)
414   FORMAT(1X,*-ERROR-  PIMAG IS 0. OR NEGATIVE*)
      CALL EXIT
209   CONTINUE

      IF(MAXPTS .LE. 512) GO TO 215
      WRITE(3,428)
428   FORMAT(1X,*-ERROR-  NPTS,NBEGIN,OR NDECI TOO LARGE*)
      CALL EXIT
215   CONTINUE

      IF(FINISH .GE. 0.) GO TO 214
      WRITE(3,415)
415   FORMAT(1X,*-ERROR-  FINISH IS NEGATIVE*)
      CALL EXIT
214   CONTINUE

      IF(FHIGH .EQ. 0.) GO TO 224
      IF(FHIGH .GT. 0.) GO TO 226
      WRITE(3,434)
434   FORMAT(1X,*-ERROR-  FHIGH IS NEGATIVE.*)
      CALL EXIT
226   CONTINUE

      IF(FLOW .GE. 0.) GO TO 225
```

```
      WRITE(3,435)
435   FORMAT(1X,*-ERROR-  FLOW IS NEGATIVE.*)
      CALL EXIT
225   CONTINUE

      IF (FHIGH .GT. FLOW) GO TO 224
      WRITE(3,436)
436   FORMAT(1X,*-ERROR-  FHIGH IS .LT. FLOW.*)
      CALL EXIT
224   CONTINUE

      ISTART=NBEGIN/NDECI
      IFINI=FINISH/DDT
      ITOPTS = IFINI - ISTART + 1
      IF(ITOPTS.LE.2048) GO TO 213
      WRITE(3,423)
423   FORMAT(1X,*--ERROR-- FINISH TOO LARGE*)
      CALL EXIT
213   CONTINUE

C------CALCULATE THE MEAN OF THE INPUT DATA.

      SUM=0.
      DO 800 I=1,512
      SUM=SUM+RESPIN(I)
800   CONTINUE
      AMEAN=SUM/512.

C------SET UP A TIME ARRAY AND SUBTRACT OUT THE MEAN.

      DO 801 I=1,512
      XTIME1(I)=(I-1)*DT
801   RESPIN(I)=(RESPIN(I)-AMEAN)*VCAL

C------PLOT THE INPUT DATA.

      CALL AMINMX(RESPIN,1,512,1,VMIN,VMAX)
      CALL PEEK(1,1,9,0,1H*,XTIME1,RESPIN,512,0.,TMAX,VMIN,VMAX,XTIT1,
     .YTIT1,TIT1,RUNTIT,IERR)

C-------COMPUTE THE FFT USING THE FORT ROUTINE

C-------FIRST SET UP A FREQUENCY ARRAY

      FMIN = 1./TMAX
      DO 802 I = 1,256
802   FREQ(I) = I*FMIN

C-------SET UP THE INPUT DATA FOR FORT

      DO 803 I=1,1024,2
803    FFTRSP(I)=RESPIN(I/2+1)
      CALL FORT(FFTRSP,9,SINES,-1,IERR)


C      TRUNCATION FILTER

      IF(FHIGH .EQ. 0.) GO TO 221
C-----NOW ZERO OUT THE FREQUENCY COMPONENTS ABOVE FHIGH.

      DO 833 IY=1,256
      IF(FREQ(IY) .GT. FHIGH) GO TO 218
833   CONTINUE
218   CONTINUE
```

```
      NC=IY-1
      IF(NC .EQ.256) GO TO 219
      DO 834 I=NC,254
      FFTRSP(2*I+3)=1.E-14
      FFTRSP(2*I+4)=1.E-14
      FFTRSP(1024-2*I-1)=1.E-14
      FFTRSF(1024-2*I)=1.E-14
      FFTRSP(513)=1.E-14
      FFTRSP(514)=1.E-14
834   CONTINUE
219   CONTINUE


C---- ZERO OUT FREQUENCY COMPONENTS BELOW FLOW.

      DO 835 IY=1,NC
      IF(FREQ(IY) .GT. FLOW) GO TO 220
835   CONTINUE
220   CONTINUE
      IC=IY-1

      IF(IC .LT. 1) GO TO 221
      DO 836 I=1,IC
      FFTRSP(2*I+1)=1.E-14
      FFTRSP(2*I+2)=1.E-14
      FFTRSP(1024-2*I+2)=1.E-14
      FFTRSP(1024-2*I+1)=1.E-14
836   CONTINUE

221   CONTINUE

C-------PLOT THE MAGNITUDE OF THE FFT

      DO 804 IS=1,256
      IF(FREQ(IS) .GT. FMAX) GO TO 210
804   CONTINUE
210   NF=IS-1
      DO 805 I = 1,NF
805    FFTMAG(I) = SQRT(FFTRSP(2*I+1)**2 + FFTRSP(2*I+2)**2)*(TMAX)

      CALL AMINMX(FFTMAG,1,NF,1,FFTMIN,FFTMAX)
      FFTMIN=FFTMAX*1.E-4
      CALL PEEK(1,1,11,2,1H*,FREQ,FFTMAG,NF,FMIN,FREQ(NF),FFTMIN,
     .FFTMAX,XTIT4,YTIT8,TIT10,RUNTIT,IERR)

C------  TRANSFORM THE TRUNCATED FREQUENCY DOMAIN DATA BACK INTO
C         THE TIME DOMAIN.

      IF(FHIGH .EQ. 0.) GO TO 223
      CALL FORT(FFTRSP,9,SINES,1,IERR)

C------COPY THE NEW FILTERED TIME HISTORY BACK INTO THE INPUT
C       DATA ARRAY.

      DO 837 I=1,1024,2
      J=J+1
      RESPIN(J)=FFTRSP(I)
837   CONTINUE
      CALL AMINMX(RESPIN,1,512,1,VMIN,VMAX)
      CALL PEEK(1,1,9,0,1H*,XTIME1,RESPIN,512,0.,TMAX,VMIN,VMAX,XTIT1,
     .YTIT11,TIT11,RUNTIT,IERR)
223   CONTINUE
C------SAMPLE THE INPUT DATA FOR PRONY ROUTINE.
```

```
      J=0
      DO 806 I=NBEGIN,MAXPTS,NDECI
      J=J+1
      RESP(J)=RESPIN(I)
      XTIME(J)=XTIME1(I)
806   CONTINUE
      CALL AMINMX(RESPIN,1,512,1,VMIN,VMAX)

C------PLOT THE SAMPLED DATA.

      CALL PEEK(1,1,9,0,1H*,XTIME1,RESPIN,512,0.,TMAX,VMIN,VMAX,XTIT1,
     .YTIT2,TIT2,RUNTIT,IERR)
      CALL PEEK(1,0,9,3,1H*,XTIME,RESP,NPTS,0.,TMAX,VMIN,VMAX,XTIT1,
     .YTIT1,TIT1,RUNTIT,IERR)

C------NOW CALL PRONY TO DO AN EXPONENTIAL CURVE
C------FIT TO THE TIME DATA.

      DO 807 I=1,NPTS
807   CRESP(I)=RESP(I)

      CALL PRONY(DDT,CRESP,A,ALPHA,NPOLES,NPTS)



C---NOW TRY TO RECONSTRUCT THE ORIG TIME FUNCTION FROM THE EXP COEFFS

      DO 808 I=1,NPTS
      DO 809 IT=1,NPOLES
809   RESPR(I)=RESPR(I)+A(IT)*CEXP((I-1)*ALPHA(IT)*DDT)
808   CONTINUE

C---PLOT THE REAL PART OF THE RECONSTRUCTED ARRAY

      DO 810 IR=1,NPTS
      RCNSTR(IR)=REAL(RESPR(IR))
810   CONTINUE
      CALL AMINMX(RCNSTR,1,NPTS,1,VMIN,VMAX)
      CALL PEEK(1,1,9,0,1H*,XTIME,RCNSTR,NPTS,0.,
     .TMAX,VMIN,VMAX,XTIT1,YTIT3,TIT3,RUNTIT,IERR)
C------- COMPUTE THE ERROR BETWEEN THE INPUT
C------- AND RECONSTRUCTED RESPONSE


      WRITE(3,416)
416   FORMAT(//,1X,*RECONSTRUCTED DATA    ORIGINAL DATA    DIFFERENCE*)
      DO 811 I=1,NPTS
      ERRORS(I)=RCNSTR(I)-RESP(I)
      WRITE(3,417) RCNSTR(I),RESP(I),ERRORS(I)
417   FORMAT(1X,E15.5,3X,E15.5,2X,E15.5)
811   CONTINUE

C------- PLOT THE ERRORS

      CALL AMINMX(ERRORS,1,NPTS,1,ERMIN,ERMAX)
      CALL PEEK(1,1,9,0,1H*,XTIME,ERRORS,NPTS,0.,
     .TMAX,ERMIN,ERMAX,XTIT1,YTIT5,TIT6,RUNTIT,IERR)

C-------COMPUTE THE RMS VOLTAGE OF THE ORIGINAL DATA

      RMSRSP = 0.
      DO 812 I=1,NPTS
812   RMSRSP = RMSRSP + RESP(I)**2
      RMSRSP = SQRT(RMSRSP/NPTS)
```

```
C------- COMPUTE THE PERCENT RMS ERROR AND WRITE ON THE PLOT

      RMSERR=0
      DO 813 I=1,NPTS
813   RMSERR=RMSERR+ERRORS(I)**2
      RMSERR=SQRT(RMSERR/NPTS)
      PCNTER = RMSERR*100./RMSRSP

      CALL SETCH(50.,40.,1,0,1)
C     WRITE(99,418) PCNTER
418   FORMAT(1X,*PERCENT RMS ERR =*,E12.4)



      DO 814 I=1,NPOLES
      POLESR(I)=REAL(ALPHA(I))
      POLESI(I)= AIMAG(ALPHA(I))/TP
814   CONTINUE

C------REORDER THE POLES FROM LOWEST TO HIGHEST FREQUENCY.

      CALL ORDHZ(POLESR,POLESI,A,NPOLES)

      CALL SETCH(0.,100.,1,0,1)
C     WRITE(99,419)
419   FORMAT(*1  POLE LISTING    -ORDERED BY FREQUENCY-*)
C     WRITE(99,420)
      WRITE(3,419)
      WRITE(3,420)

      DO 829 I=1,NPOLES
      X=CABS(A(I))
      Y=REAL(A(I))
      Z=AIMAG(A(I))
      RR=POLESR(I)
      RI=POLESI(I)
C     WRITE(99,421) I,X,Y,Z,RR,RI
      WRITE(3,421) I,X,Y,Z,RR,RI
829   CONTINUE
C------NOW REORDER THE POLE LIST BY MAGNITUDES OF THE RESIDUES.

      CALL ORDMAG(POLESR,POLESI,A,NPOLES)

C------WRITE OUT THE RESIDUE ORDERED POLE LIST.

      CALL SETCH (0.,100.,1,0,1)
C     WRITE(99,427)
427   FORMAT(*1  POLE LIST   -ORDERED BY RESIDUES-*)
C     WRITE(99,420)
      WRITE(3,427)
      WRITE(3,420)
      DO 830 I=1,NPOLES
      X=CABS(A(I))
      Y=REAL(A(I))
      Z=AIMAG(A(I))
      RR=POLESR(I)
      RI=POLESI(I)
C     WRITE(99,421) I,X,Y,Z,RR,RI
      WRITE(3,421) I,X,Y,Z,RR,RI
830   CONTINUE

      DO 815 I=1,NPOLES
```

```
 815  ALPHA(I)=CMPLX(POLESR(I),POLESI(I))


 424  FORMAT(*1--POLE DATA*,//,1X,8A10,//,1X,*TMAX =*,E12.4,/,
     .1X,*VCAL =*,E12.4,/,1X,*NPOLES =*,I5,/,1X,*NBEGIN =*,I5,/,1X,
     .*NPTS =*,I5,/,1X,*NCECI =*,I5,/,1X,*MAXPTS =*,I5,/,1X,*FMAX =*,
     .E12.4,/,1X,*FLOW =*,E12.4,/,1X,*FHIGH =*,E12.4,/,1X,*ITEST =*,I5,
     ./,1X,*RES =*,E12.4,/,1X,*RHP =*,E12.4,/,1X,*PREAL =*,E12.4,/,1X,
     .*PIMAG =*,E12.4,/,1X,*FINISH =*,E12.4,/,1X,*DT =*,E12.4,/,
     .1X,*DDT =*,E12.4,/,1X,*FNYQUIST =*,E12.4)
 439  FORMAT(1X,*ERROR WITH ALL POLES =*,E12.4,/,
     .1X,*POLES REMAINING AFTER TESTS =*,I5,/,
     .1X,*ERROR WITH TESTED POLES =*,E12.4)
 420  FORMAT(///,29X,*A*,30X,*ALPHA*,//,13X,*MAG*,13X,*R*,13X,*I*,
     .17X,*R*,13X,*I*)
 421  FORMAT(1X,I5,3E15.5,3X,2E15.5)


C------ FIND THE RANGE OF THE POLES

      CALL AMINMX(POLESR,1,NPOLES,1,PMINR,PMAXR)
      CALL AMINMX(POLESI,1,NPOLES,1,PMINI,PMAXI)
      CALL PEEK(1,1,9,3,1HX,POLESR,POLESI,NPOLES,PMINR,PMAXR,PMINI,
     .PMAXI,TIT2,YTIT4,TIT4,RUNTIT,IERR)

      CALL FRAME
      CALL PLPLTI(A,POLESR,POLESI,NPOLES,PMINR,PMAXI)

      IF(ITEST .EQ. 0) GO TO 217
C------- PLOT THE POLES AGAIN WITH FIXED AXIS.

      CALL PEEK(1,1,9,3,1HX,POLESR,POLESI,NPOLES,PREAL,-PREAL,-PIMAG,
     .PIMAG,XTIT2,YTIT4,TIT4,RUNTIT,IERR)

      CALL FRAME
      CALL PLPLTI(A,POLESR,POLESI,NPOLES,PREAL,PIMAG)
C---NOW PLOT ONLY THE UPPER QUADRANT OF THE LHP


C---FURTHER TRY TO CLASSIFY THE POLES BY THROWING OUT POLES IN THE RHP
C---AND THROWING OUT POLES WHOSE RESIDUALS ARE BELOW SOME CUTOFF VALUE

      RESMAX=0.
      DO 817 I=1,NPOLES
      IF(REAL(ALPHA(I)).GT. RHP) GO TO 211
      RESID=CABS(A(I))
      IF(RESID .GT. RESMAX) RESMAX=RESID
 211  CONTINUE
 817  CONTINUE

C   NOW, WHAT IS THE MAX ALLOWABLE RESIDUAL?

      LPOLES=1
      RESTST=RESMAX*RES
      DO 818 I=1,NPOLES

C------ELIMINATE ANY POLE IN THE RHP.

      IF(REAL(ALPHA(I)).GT. RHP) GO TO 212

C------ELIMINATE ANY POLE WITH RESIDUES < RES*MAXRESIDUAL.
      IF(CABS(A(I)) .LT. RESTST) GO TO 212

C------ELIMINATE ANY POLE WITH A FREQUENCY HIGHER THAN PIMAG
```

```
      IF(ABS(AIMAG(ALPHA(I))) .GE. PIMAG) GO TO 212
C------ELIMINATE ANY POLE WITH A SIGMA LESS THAN PREAL.
      IF(REAL(ALPHA(I)) .LE. PREAL) GO TO 212
      A(LPOLES)=A(I)
      ALPHA(LPOLES)=ALPHA(I)
      POLESR(LPOLES) = POLESR(I)
      POLESI(LPOLES) = POLESI(I)
      LPOLES=LPOLES+1
212   CONTINUE
818   CONTINUE
      LPOLES=LPOLES-1
C------THE POLES IN THE FOLLOWING PLOT ARE THE ONLY ONES USED
C------IN THE RECONSTRUCTION OF THE TIME WAVEFORM LATER ON.
      CALL PEEK(1,1,9,3,1HX,POLESR,POLESI,LPOLES,PREAL,RHP,0.,PIMAG,
     . XTIT2,YTIT4,TIT5,RUNTIT,IERR)
C------PLOT THE UPPER QUADRANT OF THE LHP IN THREE DIMENSIONS
C------SHOWING THE MAGNITUDES OF THE RESIDUES FOR EACH POLE
      CALL FRAME
      CALL PLPLOT(A,POLESR,POLESI,LPOLES,PREAL,PIMAG)
C------ REORDER THE TESTED POLES BY FREQUENCY.
      CALL ORDHZ(POLESR,POLESI,A,LPOLES)
C   NOW WRITE OUT THE POLES AGAIN
      CALL SETCH(0.,100.,1,0,1)
C     WRITE(99,422)
      WRITE(3,422)
      WRITE(3,420)
422   FORMAT('1   POLES LEFT AFTER POLE TESTS  -ORDERED BY FREQUENCY-')
C     WRITE(99,420)
      DO 831 I=1,LPOLES
      X=CABS(A(I))
      Y=REAL(A(I))
      Z=AIMAG(A(I))
      RR=POLESR(I)
      RI=POLESI(I)
C     WRITE(99,421) I,X,Y,Z,RR,RI
      WRITE(3,421) I,X,Y,Z,RR,RI
831   CONTINUE
C------ NOW REORDER THE TESTED POLES BY THE MAGNITUDE OF THE RESIDUES.
      CALL ORDMAG(POLESR,POLESI,A,LPOLES)
C------WRITE OUT THE TESTED POLES BY DESREASING MAGNITUDE OF RESIDUES.
      CALL SETCH(0.,100.,1,0,1)
C     WRITE(99,426)
426   FORMAT('1  POLES LEFT AFTER TESTS  -ORDERED BY RESIDUES-')
C     WRITE(99,420)
      WRITE(3,426)
      WRITE(3,420)
      DO 832 I=1,LPOLES
      X=CABS(A(I))
```

```
      Y=REAL(A(I))
      Z=AIMAG(A(I))
      RR=POLESR(I)
      RI=POLESI(I)
C     WRITE(99,421) I,X,Y,Z,RR,RI
      WRITE(3,421) I,X,Y,Z,RR,RI
832   CONTINUE

C------- NOW EXTRAPOLATE THE RECONSTRUCTED TIME RESPONSE

C------- SET UP NEW TIME ARRAY FOR THE EXTRAPOLATED TIME RESPONSE

      DO 819 I=ISTART,IFINI

      II=II+1
819   TIME(II)=XTIME(I)+(I-ISTART)*DDT
C-------  NOW EXTRAPOLATE THE RECONSTRUCTED TIME RESPONSE

      DO 820 I=ISTART,IFINI

      IK=IK+1
      DO 821 IT=1,LPOLES
821   EXTRAP(IK)=EXTRAP(IK)+A(IT)*CEXP((I-ISTART)
     .*CMPLX(REAL(ALPHA(IT)),AIMAG(ALPHA(IT))*TP)*DDT)
820   CONTINUE

C------- NOW PLOT THE REAL PART OF THE EXTRAPOLATED RESPONSE

      II=0
      DO 822 IR=ISTART,IFINI
      II=II+1
822   RBLDR(II)=REAL(EXTRAP(II))
      CALL AMINMX(RESPIN,1,512,1,VMIN,VMAX)
      CALL PEEK(1,1,9,0,1H*,TIME,RBLDR,II,0.,FINISH,
     .VMIN,VMAX,XTIT1,YTIT6,TIT7,RUNTIT,IERR)
      CALL PEEK(1,0,9,3,1H*,XTIME,RESP,NPTS,START,FINISH,VMIN,
     .VMAX,XTIT1,YTIT6,TIT7,RUNTIT,IERR)


C-------COMPUTE THE ERROR BETWEEN THE EXTRAPOLATED
C-------RESPONSE AND THE ORIGINAL DATA.

      DO 823 I = 1,NPTS
823   ERROR(I) = RBLDR(I) - RESP(I)

C-------NOW PLOT THE ERROR

      CALL AMINMX(ERROR,1,NPTS,1,ERMIN,ERMAX)
      CALL PEEK(1,1,9,2,1H*,XTIME,ERROR,NPTS,0.,TMAX,
     .ERMIN,ERMAX,XTIT1,YTIT7,TIT8,RUNTIT,IERR)

C-------COMPUTE THE RMS ERROR AND WRITE IT ON THE PLOT.

      RMSERR = 0.
      DO 824 I = 1,NPTS
824   RMSERR = RMSERR + ERROR(I)**2
      RMSERR = SQRT(RMSERR/NPTS)

      EXTPER = RMSERR*100./RMSRSP

      CALL SETCH(50.,10.,1,0,1)
C     WRITE(99,418) EXTPER

      WRITE(3,431)
```

```
431   FORMAT('1            TIME RECONSTRUCTION AFTER POLE TESTING')
      WRITE(3,416)
      WRITE(3,417) (RBLDR(I),RESP(I),ERROR(I),I=1,NPTS)


      IF (ITEST .EQ. 0) GO TO 216

C-------COMPUTE THE FOURIER TRANSFORM FROM THE POLES THAT
C-------ARE LEFT AFTER POLE TESTING.


C-------FIRST SET THE FREQUENCCY ARRAY.

      FREQ0 = 1./TMAX
      DFREQ = (FMAX - FREQ0)/255
      DO 825 I = 1,256
825   FREQ(I) = (I-1)*DFREQ + FREQ0

C-------NOW COMPUTE THE FOURIER TRANSFORM.

      DO 826 I = 1,256
      OMEGA = TP*FREQ(I)
      DO 827 J = 1,LPOLES
827   FFT(I) = FFT(I) + A(J)/(CMPLX(0.,OMEGA) +
     .CMPLX(REAL(ALPHA(J)),TP*AIMAG(ALPHA(J))))
826   CONTINUE


C-------NOW PLOT THE MAGNITUDE OF THE FOURIER TRANSFORM.


      DO 828 I = 1,256
828   FFTMAG(I) = CABS(FFT(I))

      CALL PEEK(1,1,11,0,1H*,FREQ,FFTMAG,256,FREQ(1),FREQ(256),
     .FFTMIN,FFTMAX,XTIT4,YTIT8,TIT9,RUNTIT,IERR)



217   CONTINUE
      CALL SETCH(0.,100.,1,0,1)
C     WRITE(99,424) RUNTIT,TMAX,VCAL,NPOLES,NBEGIN,NPTS,NDECI,MAXPTS,
C    .FMAX,FLOW,FHIGH,ITEST,RES,RHP,PREAL,PIMAG,FINISH,DT,DDT,FNYQ
C     WRITE(99,439) PCNTER,LPOLES,EXTPER
      WRITE(3,424) (RUNTIT(I),I=1,8),
     .TMAX,VCAL,NPOLES,NBEGIN,NPTS,NDECI,MAXPTS,
     .FMAX,FLOW,FHIGH,ITEST,RES,RHP,PREAL,PIMAG,FINISH,DT,DDT,FNYQ
      WRITE(3,439) PCNTER,LPOLES,EXTPER

216   CALL EXIT(1)
      END
```

```
      SUBROUTINE PRONY(DELX,YVALS,A,ALPHA,NTERMS,NPOINTS)
      DIMENSION RINR(36), RINI(36), ROOTR(35), ROOTI(35)
      COMPLEX YVALS, F,  FBARF, B, FBARB, FI, FIBARY,EIG,
     .    SOLN, A, ALPHA,FIBARFI
      COMMON /BLOCK1/ F(128,35),FBARF(35,35),B(128),FBARB(35)
      EQUIVALENCE (F,FI),(FBARF,FIBARFI),(FBARB,FIBARY)
      DIMENSION FI(128,35),  FIBARFI(35,35), FIBARY(35)
      DIMENSION SOLN(35), A(35), ALPHA(35)
      DIMENSION YVALS(256)
      IDIFF = NPOINTS - NTERMS

      IF(IDIFF .LE. 128) GO TO 200
      WRITE(3,430)
 430  FORMAT(1X,*--ERROR--   NPTS-NPOLES .GT. 128*)
 200  CONTINUE

COMMENT----NOW TO GENERATE F MATRIX

      DO  800  IA=1, NTERMS
      DO  800  IB=1, IDIFF
      IPASS=NTERMS
  800 F(IB,IA) = YVALS(IA+IB-1)

COMMENT----NOW GENERATE B VECTOR

      DO  801  IE=1, IDIFF
  801 B(IE) = - YVALS(IE+IPASS)


COMMENT----COMPUTE FBARF AND FBARB
      DO 802 I=1,NTERMS
      DO 802 J=1,NTERMS
      FBARF(I,J)=0.
      DO 802 K=1,IDIFF
802    FBARF(I,J)=FBARF(I,J)+F(K,I)*F(K,J)

      DO 803 I=1,NTERMS
      FBARB(I)=0.
      DO 803 K=1,IDIFF
803    FBARB(I)=FBARB(I)+F(K,I)*B(K)

COMMENT----USE CROUT TO SOLVE EQN FBARF*S=FBARB FOR S

      CALL CROUT(FBARF,SOLN, FBARB, NTERMS, 1)

COMMENT----CIN NOW CONTAINS SOLUTIONS FOR S
COMMENT----NOW TO FIND ROOTS OF POLYNOMIAL WITH S VECTOR AS COEFFS

      RINR(1)=1.
      RINI(1)=0.
      DO  804  IM=1,NTERMS
      IZ=NTERMS+1-IM
      IY=IM+1
      RINR(IY) = REAL(SOLN(IZ))
 804  RINI(IY) = AIMAG(SOLN(IZ))
      CALL MULLER (RINR, RINI, NTERMS, ROOTR, ROOTI)


COMMENT----NOW USE LOG (COMPLEX) TO FIND T1 AND T2, ETC.

      DO  805  IQ=1,NTERMS
 805  ALPHA(IQ) = CLOG(CMPLX(ROOTR(IQ),ROOTI(IQ)))/DELX
```

```
COMMENT----NOW TO DO A LEAST SQUARES FIT FOR THE A COEFFS IN A*EXP(ALPHA

COMMENT----FIRST GENERATE THE FI MATRIX

      DO  806  JA=1,NTERMS
      FI(1,JA)=1.
      FI(2,JA) = CMPLX(ROOTR(JA), ROOTI(JA))
      DO  806  JB=3,NPOINTS
  806 FI(JB,JA) = FI(JB-1,JA) * FI(2,JA)

COMMENT----NOW FIND FIBAR


COMMENT----COMPUTE FIBARFI AND FIBARY


      DO 807 I=1,NTERMS
      DO 807 J=1,NTERMS
      FIBARFI(I,J)=0.
      DO 807 K=1,NPOINTS
807    FIBARFI(I,J)=FIBARFI(I,J)+FI(K,I)*FI(K,J)

      DO 808 I=1,NTERMS
      FIBARY(I)=0.
      DO 808 K=1,NPOINTS
808    FIBARY(I)=FIBARY(I)+FI(K,I)*YVALS(K)


COMMENT----USE CROUT TO SOLVE EQN FIBARFI*A=FIBARY  FOR A


      CALL CROUT (FIBARFI, A, FIBARY, NTERMS, I)


      RETURN
      END
```

```
      SUBROUTINE ORDHZ(PR,PI,ACOEF,NPOLES)
      COMPLEX ACOEF,TEMPC
      DIMENSION PR(1),PI(1),ACOEF(1)
      NPM1=NPOLES-1
      DO 800 J=1,NPM1
      AMIN=ABS(PI(J))
      INDX=J
      K=J+1
      DO 801 I=K,NPOLES
      IF(ABS(PI(I)).GE.AMIN) GO TO 200
      AMIN=ABS(PI(I))
      INDX=I
200   CONTINUE
801    CONTINUE
      TEMP=PI(J)
      PI(J)=PI(INDX)
      PI(INDX)=TEMP
C------- NOW REORDER THE PR AND A ARRAYS IN THE SAME
C------- MANNER AS THE PI ARRAY

      TEMP=PR(J)
      PR(J)=PR(INDX)
      PR(INDX)=TEMP
      TEMPC=ACOEF(J)
```

```
      ACOEF(J)=ACOEF(INDX)
      ACOEF(INDX)=TEMPC
  800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ORDMAG(PR,PI,ACOEF,NPOLES)
      COMPLEX ACOEF,TEMPC
      DIMENSION PR(1),PI(1),ACOEF(1)
      NPM1=NPOLES-1
      DO 800 J=1,NPM1
      AMAX=CABS(ACOEF(J))
      INDX=J
      K=J+1
      DO 801 I=K,NPOLES
      IF(CABS(ACOEF(I)).LE.AMAX) GO TO 200
      AMAX=CABS(ACOEF(I))
      INDX=I
  200 CONTINUE
  801  CONTINUE
      TEMPC=ACOEF(J)
      ACOEF(J)=ACOEF(INDX)
      ACOEF(INDX)=TEMPC
C------- NOW REORDER THE PR AND PI ARRAYS IN THE SAME
C------- MANNER AS THE ACOEF ARRAY

      TEMP=PR(J)
      PR(J)=PR(INDX)
      PR(INDX)=TEMP
      TEMP=PI(J)
      PI(J)=PI(INDX)
      PI(INDX)=TEMP
  800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULLER(COE,COEI,N1,ROOTR,ROOTI)
*     FORTRAN              MULLER    C2.2-001B
C MULLER
      DIMENSION COE(21),ROOTR(20),ROOTI(20),COEI(21)
      N2=N1+1
      N4=0
      I=N1+1
   19 IF(COE(I))9,7,9
    7 IF(COEI(I)) 9,1,9
    1 N4=N4+1
      ROOTR(N4)=0.
      ROOTI(N4)=0.
      I=I-1
      IF(N4-N1)19,37,19
    9 CONTINUE
   10 AXR=0.97
      AXI=0.
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GO TO 99
```

```
   11 BET1R=TEMR
      BET1I=TEMI
      AXR=0.98
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GO TO 99
   12 BET2R=TEMR
      BET2I=TEMI
      AXR=0.99
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GO TO 99
   13 BET3R=TEMR
      BET3I=TEMI
   14 TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.
      TE9=TE3*TE3-TE4*TE4
      TE10=2.*TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
      TEM=SQRT(TE1*TE1+TE2*TE2)
      IF(TE1)113,113,112
  113 TE4=SQRT(.5*(TEM-TE1))
      TE3=.5*TE2/TE4
      GO TO 111
  112 TE3=SQRT(.5*(TEM+TE1))
      IF(TE2)110,200,200
  110 TE3=-TE3
  200 TE4=.5*TE2/TE3
  111 TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.*TE15
      TE2=2.*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
  204 TE7=TE9
      TE8=TE10
  205 TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
```

```
      ALP4I=AXI
      M=4
      GO TO 99
   15 N6=1
   38 IF(ABS(HELL)+ABS(BELL)-1.E-20)18,18,16
   16 TE7=ABS(ALP3R-AXR)+ABS(ALP3I-AXI)
      IF(TE7/(ABS(AXR)+ABS(AXI))-1.E-7)18,18,17
   17 N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3 .LT. 100) GO TO 14
      IFLAG=1
      WRITE(3,400)
 400  FORMAT(1X,*--ERROR-- BAD ROOTS IN SUBROUTINE MULLER*)
   18 N4=N4+1
      ROOTR(N4)=ALP4R
      ROOTI(N4)=ALP4I
      IF(IFLAG .EQ. 1) WRITE(3,401) N4,ROOTR(N4),ROOTI(N4)
 401  FORMAT(1X,I5,2E15.5)
      IFLAG=0
      N3=0
   41 IF(N4-N1)10,37,37
   37 RETURN
   99 TEMR=COE(1)
      TEMI=0.0
      DO 100  I=1,N1
      TE1=TEMR*AXR-TEMI*AXI
      TEMI=TEMI*AXR+TEMR*AXI +COEI(I+1)
  100 TEMR=      TE1+COE(I+1)
      HELL=TEMR
      BELL=TEMI
   42 IF(N4)102,103,102
  102 DO 101  I=1,N4
      TEM1=AXR-ROOTR(I)
      TEM2=AXI-ROOTI(I)
      TE1=TEM1*TEM1+TEM2*TEM2
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
  101 TEMR=TE2
  103 GO TO(11,12,13,15),M
      END
```

```
      SUBROUTINE CROUT (C, X, Y, NCOLSC, NCOLSY)
*     FORTRAN            CROUT
       COMPLEX   SUM,HIGH,SUM1,DETERM
      COMPLEX A, C, X, Y
      DIMENSION A(35,36), C(35,35), X(35,1), Y(35,1), INDEX(35)

C
C     CROUT REDUCTION



C       THIS IS A CROUT REDUCTION PROGRAM WHICH CAN EITHER SOLVE FOR THE
C      SOLUTION OF A MATRIX EQUATION, OR AS A SPECIAL CASE, IT CAN COMPU
C      INVERSE OF A SPECIFIED N BY N MATRIX
C
C     THE TECHNIQUE IS CLEARLY EXPLAINED ON PAGES 429 TO 435 OF THE BOOK
C      F.B. HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS, PUBLISHED BY
C      MCGRAW HILL IN 1956

C        THIS PROGRAM SOLVES THE MATRIX EQUATION HX = B. THE H AND THE B
C       MATRICES ARE SPECIFIED PARAMETERS, AND THE PROGRAM CALCULATES TH
C      MATRIX.

C        H IS AN N BY N MATRIX OF NUMBERS, B IS AN N BY M MATRIX OF NUMB
C       AND X IS AN N BY M MATRIX OF NUMBERS TO BE DETERMINED. IF THE IN
C        OF THE H MATRIX IS DESIRED, ONE SETS THE B MATRIX EQUAL TO AN N
C       IDENTITY MATRIX, THUS X BECOMES THE INVERSE MATRIX.

C      THE A MATRIX SPECIFIED IN THE ARGUMENT OF THE SUBROUTINE IS AN
C      AUGMENTED MATRIX CONTAINING THE N BY N MATRIX H IN THE FIRST N BY
C     LOCATIONS, AND IS AUGEMENTED IN THE N + 1  TO N + M COLUMNS BY THE
C        MATRIX B.
C
C       NN IS BY DEFINITION EQUAL TO N + M.
C
C     THE SOLUTION MATRIX, X, IS STORED IN THE FIRST N BY M (ROWS BY COL

C      ELEMENTS OF THE AUGMENTED MATRIX A, THE ORIGINAL H AND B MATRICES
C     DESTROYED, AND SHOULD BE SAVED ELSEWHERE IF THEY ARE TO BE USED AF
C          USING THIS ROUTINE.

COMMENT----SOLVES CX=Y FOR X    C=COEFF MATRIX
COMMENT----COPIES C AND Y INTO A MATRIX
      N=NCOLSC
      M=NCOLSY
      NN=N+M
      DO 200 IQ1=1,N
      DO 200 IQ2=1,N
  200 A(IQ1,IQ2) = C(IQ1,IQ2)
      DO 210 IQ3=1,N
      DO 210 IQ4=1,M
  210 A(IQ3,IQ4+N) = Y(IQ3,IQ4)
       ABJC = 1.E-200
       JZ=N-1
       JA=N+1

        DO 30 I = 1, N
 30     INDEX(I)=I



      DO 700 J=1,NN


```

```
      DO 800 II=1,N
      SUM=0.0
      I=INDEX(II)
      IF(II-J)33,34,34

C       THE SECTION AFTER STATEMENT 33 IS EQUIVALENT TO FORMULA 10.4.5 AN
C                10.4.6 OF HILDEBRAND

  33   IF (II - 1) 9000,9200,9000
 9000  LLLL = II - 1

       DO 9100 K = 1,LLLL
       IPPP = INDEX(K)
 9100   SUM = SUM + A(I,K)*A(IPPP,J)

 9200   A(I,J) = (A(I,J)-SUM)/A(I,II)

        GO TO 800

C       THE SECTION AFTER STATEMENT 34 IS EQUIVALENT TO FORMULA 10.4.4
C             OF HILDEBRAND

  34   IF (J - 1) 8000,8200,8000
 8000   LLLL = J - 1

      DO 8100 K=1,LLLL
      IPPP=INDEX(K)
 8100 SUM=SUM+A(I,K)*A(IPPP,J)

 8200 A(I,J)=A(I,J)-SUM

  800   CONTINUE


      IF(J-N)41,700,700
   41 L=INDEX(J)

C           THIS SECTION SHIFTS AND REORDERS THE COLUMNS AND ROWS ( THEY
C        RESHUFFLED AT THE END OF THE PROCESS TO BE PUT IN THE ORIGINAL O

      KA=L
      HIGH=A(L,J)
      KZ=0

      DO 35 I=J,JZ
      JC=I+1
      L=INDEX(JC)
       IF (CABS(HIGH) - CABS(A(L,J))) 36,35,35
   36 HIGH=A(L,J)
      KA=L
      KZ=1
   35 CONTINUE

       IF (CABS(HIGH) - ABJC) 31,31,3200
31    WRITE(3,32) ABJC
  32   FORMAT (2X,24HPIVOT ELEMENT LESS THAN ,E20.8)

 3200 DO 37 K=1,N
      KK=K
       IF(INDEX(K)-KA)37,38,37
   37 CONTINUE

   38 ITEMP=INDEX(J)
      INDEX(J)=INDEX(KK)
```

```
      INDEX(KK)=ITEMP
  700 CONTINUE
      IF(M)2000,1000,2000
 2000 L=N-1
       DO 39 J = JA,NN
       LL = 1
       DO 42 K = 1,N
C      THIS SECTION IS USED TO SEE IF ONLY A SINGULAR TYPE SOLUTION IS P
       IF (CABS(A(K,J)) - 0.0) 43,42,43
   42 CONTINUE
      IZ=INDEX(N)
       IF (CABS(A(IZ,N)) - 1.E-2) 46,46,44
   44  CONTINUE
      WRITE(3,45)
45    FORMAT(1X,* ONLY SOLUTION IS ZERO VECTOR*)
      GO TO 10
   46  CONTINUE
      WRITE(3,45)
       GO TO 10
C        THIS LOOP IS EQUIVALENT TO 10.4.7 OF HILDEBRAND
   43 DO 40 IJ=LL,L
      SUM1=0.0
      II=N-IJ
      I=INDEX(II)
      LLL=II+1
      DO 9300 K=LLL,N
      IP=INDEX(K)
 9300 SUM1=SUM1+A(I,K)*A(IP,J)
      A(I,J)=A(I,J)-SUM1
   40 CONTINUE
   39 CONTINUE
 1000   CONTINUE
C       THIS SECTION SHIFTS THE SOLUTION MATRIX INTO THE FIRST N BY M
C           LOCATIONS (ROWS BY COLUMNS)
      DO 400 I=1,N
       DO 400 J = JA,NN
      K=INDEX(I)
      L=J-N
  400 A(I,L)=A(K,J)
COMMENT----WRITE ANSWER INTO X MATRIX
```

```
      DO 250 IQ5 = 1,N
      DO 250 IQ6=1,M
  250 X(IQ5,IQ6) = A(IQ5,IQ6)

   10  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PEEK (NCRT, NF, LG, KP, AP, X, Y, NXY, XMIN, XMAX,
     1                YMIN, YMAX, ALABX, ALABY, ALABTT,PTITLE,IND)
      RETURN
      END
```

```
      SUBROUTINE FORT(A,M,S,IFS,IFERR)
C
C     FOURIER TRANSFORM SUBROUTINE, PROGRAMMED IN SYSTEM/360,
C     BASIC PROGRAMMING SUPPORT, FORTRAN IV, FORM C28-6504
C     THIS DECK SET UP FOR IBSYS ON IBM 7094.
C
C     DOES EITHER FOURIER SYNTHESIS,I.E.,COMPUTES COMPLEX FOURIER SERIES
C     GIVEN A VECTOR OF N COMPLEX FOURIER AMPLITUDES,OR, GIVEN A VECTOR
C     OF COMPLEX DATA X DOES FOURIER ANALYSIS, COMPUTING AMPLITUDES.
C     A IS A COMPLEX VECTOR OF LENGTH N=2**M COMPLEX NOS. OR 2*N REAL
C     NUMBERS. A IS TO BE SET BY USER.
C     M  IS AN INTEGER 0.LT.M.LE.13, SET BY USER.
C     S IS A VECTOR S(J)= SIN(2*PI*J/NP ), J=1,2,....,NP/4-1,
C     COMPUTED BY PROGRAM.
C     IFS IS A PARAMETER TO BE SET BY USER AS FOLLOWS-
C     IFS=0 TO SET NP=2**M AND SET UP SINE TABLE.
C     IFS=1 TO SET N=NP=2**M, SET UP SIN TABLE, AND DO FOURIER
C     SYNTHESIS, REPLACING THE VECTOR A BY
C
C     X(J)= SUM OVER K=0,N-1 OF A(K)*EXP(2*PI*I/N)**(J*K),
C     J=0,N-1, WHERE I=SQRT(-1)
C
C     THE  X'S  ARE STORED WITH   RE X(J) IN CELL  2*J+1
C     AND  IM X(J)  IN CELL  2*J+2  FOR  J=0,1,2,...,N-1.
C     THE  A'S  ARE STORED IN THE SAME MANNER.
C
C     IFS=-1   TO SET  N=NP=2**M,SET UP SIN TABLE, AND DO FOURIER
C     ANALYSIS, TAKING THE INPUT VECTOR A AS X AND
C     REPLACING IT BY THE A  SATISFYING THE ABOVE FOURIER SERIES.
C     IFS=+2 TO DO FOURIER SYNTHESIS ONLY, WITH A PRE-COMPUTED S.
C     IFS=-2 TO DO FOURIER ANALYSIS ONLY, WITH A PRE-COMPUTED S.
C     IFERR   IS SET BY PROGRAM TO-
C     =0 IF NO ERROR DETECTED.
C     =1 IF M IS OUT OF RANGE., OR, WHEN IFS=+2,-2, THE
C     PRE-COMPUTED S TABLE IS NOT LARGE ENOUGH.
C     =-1 WHEN IFS =+1,-1, MEANS ONE IS RECOMPUTING S TABLE
C     UNNECESSARILY.
C
C     NOTE- AS STATED ABOVE, THE MAXIMUM VALUE OF M FOR THIS PROGRAM
C     ON THE IBM 7094 IS 13. FOR 360 MACHINES HAVING GREATER STORAGE
C     CAPACITY, ONE MAY INCREASE THIS LIMIT BY REPLACING 13 IN
C     STATEMENT 3 BELOW BY LOG2 N, WHERE N IS THE MAX. NO. OF
C     COMPLEX NUMBERS ONE CAN STORE IN HIGH-SPEED CORE.  ONE MUST
C     ALSO ADD MORE  DO  STATEMENTS TO THE BINARY SORT ROUTINE
C     FOLLOWING STATEMENT  24  AND CHANGE THE EQUIVALENCE STATEMENTS
```

```
C     FOR THE  K'S.
C
      DIMENSION A(1), S(1), K(14)
      EQUIVALENCE (K(13),K1),(K(12),K2),(K(11),K3),(K(10),K4)
      EQUIVALENCE (K( 9),K5),(K( 8),K6),(K(7),K7),(K( 6),K8)
      EQUIVALENCE (K( 5),K9 ),(K( 4),K10),(K( 3),K11),(K( 2),K12)
      EQUIVALENCE (K( 1),K13),( K(1),N2)
      IF(M)2,2,3
    3 IF(M-13) 5,5,2
    2 IFERR=1
    1 RETURN
    5 IFERR=0
      N=2**M
      IF( IABS(IFS) - 1 ) 200,200,10
C     WE ARE DOING TRANSFORM ONLY. SEE IF PRE-COMPUTED
C     S TABLE IS SUFFICIENTLY LARGE
   10 IF( N-NP )20,20,12
   12 IFERR=1
      GO TO 200
C     SCRAMBLE A, BY SANDE'S METHOD
   20 K(1)=2*N
      DO 22 L=2,M
      K(L)=K(L-1)/2
   22 CONTINUE
      DO 24 L=M,12
      K(L+1)=2
   24  CONTINUE
C     NOTE EQUIVALENCE OF KL AND K(14-L)
C     BINAR\ SORT-
      IJ=2
      DO 30 J1=2,K1,2
      DO 30 J2=J1,K2,K1
      DO 30 J3=J2,K3,K2
      DO 30 J4=J3,K4,K3
      DO 30 J5=J4,K5,K4
      DO 30 J6=J5,K6,K5
      DO 30 J7=J6,K7,K6
      DO 30 J8=J7,K8,K7
      DO 30 J9=J8,K9,K8
      DO 30 J10=J9,K10,K9
      DO 30 J11=J10,K11,K10
      DO 30 J12=J11,K12,K11
      DO 30 JI=J12,K13,K12
      IF(IJ-JI)28,30,30
   28 T=A(IJ-1 )
      A(IJ-1)=A(JI-1)
      A(JI-1)=T
      T=A(IJ)
      A(IJ)=A(JI)
      A(JI)=T
   30 IJ=IJ+2
      IF(IFS)32,2,36
C     DOING FOURIER ANALYSIS,SO DIV. BY N AND CONJUGATE.
   32 FN = N
      DO 34 I=1,N
      A(2*I-1) = A(2*I-1)/FN
      A(2*I)=-A(2*I)/FN
   34  CONTINUE
C     SPECIAL CASE- L=1
   36 DO 40 I=1,N,2
      T = A(2*I-1)
      A(2*I-1) =T + A(2*I+1)
      A(2*I+1)=T-A(2*I+1)
```

```
      T=A(2*I)
      A(2*I) = T + A(2*I+2)
      A(2*I+2)= T - A(2*I+2)
 40    CONTINUE
      IF(M-1) 2,1   ,50
C     SET FOR L=2
   50 LEXP1=2
C     LEXP1=2**(L-1)
      LEXP=8
C     LEXP=2**(L+1)
      NPL= 2**MT
C     NPL = NP* 2**-L
   60 DO 130 L=2,M
C     SPECIAL CASE- J=0
      DO 80 I=2,N2,LEXP
      I1=I + LEXP1
      I2=I1+ LEXP1
      I3 =I2+LEXP1
      T=A(I-1)
      A(I-1) = T +A(I2-1)
      A(I2-1) = T-A(I2-1)
      T =A(I)
      A(I) = T+A(I2)
      A(I2) = T-A(I2)
      T= -A(I3)
      TI = A(I3-1)
      A(I3-1) = A(I1-1) - T
      A(I3   ) = A(I1 )    - TI
      A(I1-1) = A(I1-1) +T
      A(I1)    = A(I1    )   +TI
 80    CONTINUE
      IF(L-2) 120,120,90
   90 KLAST=N2-LEXP
      JJ=NPL
      DO 110 J=4,LEXP1,2
      NPJJ=NT-JJ
      UR=S(NPJJ)
      UI=S(JJ)
      ILAST=J+KLAST
      DO 100 I= J,ILAST,LEXP
      I1=I+LEXP1
      I2=I1+LEXP1
      I3=I2+LEXP1
      T=A(I2-1)*UR-A(I2)*UI
      TI=A(I2-1)*UI+A(I2)*UR
      A(I2-1)=A(I-1)-T
      A(I2  )=A(I    ) - TI
      A(I-1) =A(I-1)+T
      A(I)    =A(I)+TI
      T=-A(I3-1)*UI-A(I3)*UR
      TI=A(I3-1)*UR-A(I3)*UI
      A(I3-1)=A(I1-1)-T
      A(I3)    =A(I1   )-TI
      A(I1-1)=A(I1-1)+T
      A(I1)    =A(I1)    +TI
 100   CONTINUE
C     END OF I LOOP
      JJ=JJ+NPL
 110   CONTINUE
C     END OF J LOOP
  120 LEXP1=2*LEXP1
      LEXP = 2*LEXP
      NPL=NPL/2
```

```
 130     CONTINUE
C      END OF L LOOP
   140 IF(IFS)145,2,1
C      DOING FOURIER ANALYSIS. REPLACE A BY CONJUGATE.
   145 DO 150 I=1,N
       A(2*I) =-A(2*I)
 150     CONTINUE
   160 GO TO 1
C      RETURN
C      MAKE TABLE OF S(J)=SIN(2*PI*J/NP),J=1,2,....NT-1,NT=NP/4
   200 NP=N
       MP=M
       NT=N/4
       MT=M-2
       IF(MT) 260,260,205
   205 THETA=.7853981634
C      THETA=PI/2**(L+1)     FOR L=1
   210 JSTEP = NT
C      JSTEP = 2**( MT-L+1 ) FOR L=1
       JDIF = NT/2
C      JDIF = 2**(MT-L)  FOR L=1
       S(JDIF) = SIN(THETA)
       IF (MT-2)260,220,220
   220 DO 250 L=2,MT
       THETA = THETA/2.
       JSTEP2 = JSTEP
       JSTEP = JDIF
       JDIF = JDIF/2
       S(JDIF)=SIN(THETA)
       JC1=NT-JDIF
       S(JC1)=COS(THETA)
       JLAST=NT-JSTEP2
       IF(JLAST-JSTEP)250,230,230
   230 DO 240 J=JSTEP,JLAST,JSTEP
       JC=NT-J
       JD=J+JDIF
   240 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)
   250 CONTINUE
   260 IF(IFS)20,1,20
       END
```

```
      SUBROUTINE PLPLOT(A,ALPR,ALPI,NP,PLTR,PLTI)
      COMPLEX A
      DIMENSION A(1),ALPR(1),ALPI(1),XG(6),YG(6)

C---DEFINE SOME PROGRAM CONSTANTS

      THETA=0.785
      ST=SIN(THETA)
      CT=COS(THETA)

C---ESTABLISH MAPPING AND DRAW A GRID ON THE SIGMA-OMEGA PLANE

      CALL MAP(-1.707,0.,0.,1.707,.1,.9,.1,.9)

      DATA (XG=0.,-.2,-.4,-.6,-.8,-1.),
     . (YG=0.,.1414,.2828,.4243,.5657,.7071)

      DO 800 I=1,6
      X1=-YG(I)
      X2=-1.+X1
      Y1=Y2=YG(I)
      CALL LINEP(X1,Y1,X2,Y2,3)
      Y1=0.
      Y2=YG(6)
      X1=XG(I)
      X2=X1-YG(6)
800    CALL LINEP(X1,Y1,X2,Y2,3)
      CALL LINEP(0.,0.,0.,1.,3)
      CALL LINE(0.,.333,-.01,.333)
      CALL LINE(0.,.666,-.01,.666)
      CALL LINE(0.,1.,-.01,1.)

C---FIND THE MAGNITUDE OF THE LARGEST RESIDUAL

      AMAX=CABS(A(1))
      DO 802 I=2,NP
802    IF(CABS(A(I)).GT.AMAX)AMAX=CABS(A(I))

C---NOW PLOT THE POLES IN THE UPPER LEFT HALF PLANE

      DO 803 I=1,NP
      IF(ALPI(I).LT.0)GO TO 803
      IF(ALPR(I).GT.0)GO TO 803
      ALPSR=ALPR(I)/ABS(PLTR)
      ALPSI=ALPI(I)/ABS(PLTI)
      AMAG=CABS(A(I))/AMAX
      AMAG=ALOG10(1000.*AMAG)/3.
      IF(AMAG.LT.0)AMAG=.001
      X1=X2=(ALPSR-ALPSI*CT)
      Y1=ALPSI*ST
      Y2=ALPSI*ST+AMAG
      CALL PLOTV(X1,Y1,X2,Y2,.005)
      CALL SETLCH(X1-.002,Y1+.002,1,0,0)
      CALL CRTBCD(1H*)
803    CONTINUE
      CALL SETCH(57.,3.,1,0,1)
      CALL CRTBCD(4HREAL)
      CALL SETCH(16.,10.,1,0,1)
      CALL CRTBCD(4HIMAG)
      RETURN
      END
```

```
      SUBROUTINE PLPLTI(A,ALPR,ALPI,NP,PLTR,PLTI)
      COMPLEX A
      DIMENSION A(1),ALPR(1),ALPI(1),XG(6),YG(6)

C---DEFINE SOME PROGRAM CONSTANTS

      THETA=0.785
      ST=SIN(THETA)
      CT=COS(THETA)

C---ESTABLISH MAPPING AND DRAW A GRID ON THE SIGMA-OMEGA PLANE

      CALL MAP(-1.707,1.,0.,1.707,.01,.99,.1,.9)

      DATA (XG=0.,-.2,-.4,-.6,-.8,-1.),
     . (YG=0.,.1414,.2828,.4243,.5657,.7071)

      DO 800 I=1,6
      X1=-YG(I)+1.
      X2=-1.-YG(I)
      Y1=Y2=YG(I)
      CALL LINEP(X1,Y1,X2,Y2,3)
      Y1=0.
      Y2=YG(6)
      X1=XG(I)
      X2=X1-YG(6)
      CALL LINEP(X1,Y1,X2,Y2,3)
      X1=-X1
      X2=X1-YG(6)
800   CALL LINEP(X1,Y1,X2,Y2,3)
      CALL LINEP(0.,0.,0.,1.,3)
      CALL LINE(0.,.333,-.01,.333)
      CALL LINE(0.,.666,-.01,.666)
      CALL LINE(0.,1.,-.01,1.)
      CALL LINE(0.,0.,-YG(6),YG(6))

C---FIND THE MAGNITUDE OF THE LARGEST RESIDUAL

      AMAX=0.
      DO 802 I=1,NP
      IF(ALPR(I).LT.PLTR) GO TO 802
      IF(ALPI(I).GT.PLTI) GO TO 802
      IF(CABS(A(I)).GT.AMAX)AMAX=CABS(A(I))
802   CONTINUE

C---NOW PLOT THE POLES IN THE UPPER LEFT HALF PLANE

      DO 803 I=1,NP
      IF(ALPI(I).LT.-1.E-4)GO TO 803
      IF(ALPR(I).LT.PLTR) GO TO 803
      IF(ALPI(I).GT.PLTI) GO TO 803
      ALPSR=ALPR(I)/ABS(PLTR)
      ALPSI=ALPI(I)/ABS(PLTI)
      AMAG=CABS(A(I))/AMAX
      AMAG=ALOG10(1000.*AMAG)/3.
      IF(AMAG.LT.0)AMAG=.001
      X1=X2=(ALPSR-ALPSI*CT)
      Y1=ALPSI*ST
      Y2=ALPSI*ST+AMAG
      CALL PLOTV(X1,Y1,X2,Y2,.005)
      CALL SETLCH(X1-.002,Y1+.002,1,0,0)
      CALL CRTBCD(1H*)
803   CONTINUE
      CALL SETCH(52.,3.,1,0,1)
```

```
      CALL CRTBCD(4HREAL)
      CALL SETCH(4.,10.,1,0,1)
      CALL CRTBCD(4HIMAG)
      RETUPN
      END
```

```
      SUBROUTINE FRAME
      RETURN
      END
```

```
      SUBROUTINE SETCH(A,B,C,D,E,F)
      RETURN
      END
```

```
      SUBROUTINE CRTID(A,B)
      RETURN
      END
```

```
      SUBROUTINE MAPX(A,B,C,D,E,F,G,H,I)
      RETURN
      END
```

```
      SUBROUTINE SETPCH(A,B,C,D,E)
      RETURN
      END
```

```
      SUBROUTINE POINTC(A,B,C,D)
      RETURN
      END
```

```
      SUBROUTINE TRACE(A,B,C)
      RETURN
      END
```

```
      SUBROUTINE DUMP
      RETURN
      END
```

```
      SUBROUTINE MAP(A,B,C,D,E,F,G,H)
      RETURN
      END


      SUBROUTINE LINEP(A,B,C,D,E)
      RETURN
      END


      SUBROUTINE LINE(A,B,C,D)
      RETURN
      END


      SUBROUTINE PLOTV(A,B,C,D,E)
      RETURN
      END


      SUBROUTINE SETLCH(A,B,C,D,E)
      RETURN
      END


      SUBROUTINE CRTBCD(A)
      RETURN
      END


      SUBROUTINE AMINMX(A,B,C,D,E,F)
      RETURN
      END


)ER
 LATEST VERSION 35 POLE CAPABILITY  SCATDIP30 INPUT.
        1703.7E-9              1.
    18   75   36    8
          5.E7            0.        1.8E7
     1
            0.            0.       -1.5E7        5.0E7
          20.E-7
)EF
```

References

1. Baum, Carl E., "On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems," Air Force Weapons Laboratory, EMP Interaction Note 88 (1971).

2. Tesche, Fredrick M., "On the Singularity Expansion Method as Applied to Electromagnetic Scattering from Thin Wires," EMP Interaction Note 102 (1972).

3. Poggio, A. J., Lager, D. L., and Hudson, H. G., Transient Data Processing Using Complex Exponential Representations, to be published.

4. Householder, A. S., On Prony's Method of Fitting Exponential Decay Curves and Multiple-Hit Survival Curves, Oak Ridge National Laboratory, Oak Ridge, Tenn., Rept. ORNL-455 (1950).

5. Hildebrand, F. B., Introduction to Numerical Analysis (McGraw-Hill Book Co. Inc., New York, 1956).

6. Kelly, Louis G., Handbook of Numerical Methods and Applications (Addison-Wesley, Reading, Mass., 1967).

7. Pennington, Ralph H., Introductory Computer Methods and Numerical Analysis (MacMillan, New York, 1965).

8. Lawrence, J. Dennis, Polynomial Root Finder, Lawrence Livermore Laboratory, Rept. C22-001 (1966).

9. Landt, J. A., Miller, E. K., and Van Blaricum, M., WT-MBA/LLL1B: A Computer Program for the Time Domain Electromagnetic Response of Thin-Wire Structures," EMP Interaction Note 210, 6 May 1974.

10. Deadrick, F. J., Miller, E. K., and Hudson, H. G., The LLL Transient Electromagnetics Measurement Facility, Lawrence Livermore Laboratory, Rept. UCRL-51933 (1975).