

Mathematics Notes

Note 46

July 1977

A Note on Representing a Transient Waveform  
by a Finite Sum of Complex Exponentials

Dr. Tom Cordaro  
Air Force Weapons Laboratory

Abstract

The problem of representing an observed transient waveform by a finite sum of complex exponentials is considered. A least squares technique that involves overfitting the model signal and then estimating the correct order is applied to simulated data as well as data recorded on photographs. In all the examples, the fitted waveform approximates the original waveform well when both functions are plotted on the same graph.

## Acknowledgment

The author would like to thank Mr. John Podlesny of the Air Force Weapons Laboratory for answering questions and making a number of suggestions about this work, and Mr. Harry Season who helped with testing the program and preparing this report.

## CONTENTS

<u>Section</u>		<u>Page</u>
I	INTRODUCTION	5
II	METHODS	7
III	RESULTS	17
IV	CONCLUSIONS	36
	REFERENCES	38

## ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Flowchart for RLS	15
2	Damped Sine Wave for Example 4	20
3	FFT Magnitude of Damped Sine Wave	22
4	Reconstructed Damped Sine Wave Compared with Original	23
5	FFT Magnitude of Reconstructed Damped Sine Wave	24
6	Reconstruction of Filtered Damped Sine Wave	25
7	FFT Magnitude of Filtered Reconstruction for the Damped Sine Wave	26
8	Two Damped Sine Waveforms Used in Example 5	27
9	FFT Magnitude of Two Damped Sine Waveforms	28
10	Reconstruction of Two Damped Sine Waveforms	29
11	FFT Magnitude of Reconstructed Two Damped Sine Waveforms	30
12	Green Box Response	32
13	FFT Magnitude of Green Box Response	33
14	Reconstructed Green Box Response	34
15	FFT Magnitude of Reconstructed Green Box Response	35

SECTION I  
INTRODUCTION

The singularity expansion method (ref. 1) has been used to compute transient electromagnetic responses in terms of a finite number of Laplace transform poles and residues. This has given rise to the question of how these poles and residues can be calculated from experimental data. The purpose of the report is to discuss some of the techniques that have been used with related problems and to describe the results obtained with one particular method using both simulated data and data digitized from experimental transient waveforms. The problem to be considered can be posed as follows.

Let  $y_k$ ,  $k = 1, \dots, N$  be samples  $T$  sec apart of a data waveform. Usually in EMP work a data signal is recorded on a photograph, then digitized resulting in the  $y_k$ . We want to fit a signal of the form

$$x_k = \sum_{i=1}^m A_i \exp [s_i(k-1)T], \quad k = 1, \dots, N \quad (1)$$

to the given data sequence. The sequence  $x_k$  has poles  $s_i$  and residues  $A_i$ . The observations can be written as

$$y_k = x_k + e_k \quad (2)$$

where  $e_k$  represents the error between the  $k^{\text{th}}$  observation and model signals.

The most common approach is to attempt to pick  $m$ ,  $A_i$  and  $s_i$  to minimize the squared error

$$E^2 = \frac{1}{N} \sum_{k=1}^N e_k^2 \quad (3)$$

Thus, given  $y_k$ ,  $k = 1, \dots, N$ , find the values of  $m$ ,  $A_i$  and  $s_i$  to minimize the expression

$$E^2 = \frac{1}{N} \sum_{k=1}^N \left( y_k - \sum_{i=1}^m A_i \exp [s_i T(k-1)] \right)^2 \quad (4)$$

This "pole extraction" problem has appeared in a number of different areas including circuit theory, statistics, and control theory. The following sections review previous work and discuss the results obtained from this investigation.

## SECTION II

### METHODS

#### 1. DIRECT METHOD

A direct approach to the problem is to set the partial derivatives of the squared error function with respect to  $s_j$  and  $A_j$  to zero and solve the resulting equations. That is

$$\frac{\partial E^2}{\partial s_j} = \frac{\partial E^2}{\partial A_j} = 0 \quad j = 1, \dots, m \quad (5)$$

These are necessary conditions for a minimum with a fixed  $m$ . How to minimize with respect to  $m$  is still an open question. The simplest approach is to find the minimum  $E^2$  for  $m = 1, 2, 3, \dots$  and choose the  $m$  that looks best according to some criteria. One method is discussed in section II.3. Equations 5 give

$$\sum_{k=1}^N e_k^{(k-1)} \exp [s_j T(k-1)] = 0 \quad j = 1, \dots, m \quad (6)$$

$$\sum_{k=1}^N e_k \exp [s_j T(k-1)] = 0 \quad j = 1, \dots, m \quad (7)$$

Letting  $Z_j = \exp (s_j T)$ , rearrange equations 6 and 7 to

$$\sum_{k=1}^N (k-1) y_k Z_j^{k-1} = \sum_{k=1}^N \sum_{i=1}^M A_i^{(k-1)} (Z_i Z_j)^{k-1} \quad j = 1, \dots, m \quad (8)$$

$$\sum_{k=1}^N y_k Z_j^{k-1} = \sum_{k=1}^N \sum_{i=1}^M A_i (Z_i Z_j)^{k-1} \quad j = 1, \dots, m \quad (9)$$

As can be seen, equations 8 and 9 are a system of coupled nonlinear equations. The continuous time versions of equations 6 through 9 are contained in

reference 2 along with an iterative technique for approximating the solution. Another iterative technique is given in reference 3. One point to be noticed here is that a fairly simple problem statement gives rise to a system of non-linear equations with no known analytical solution.

For what we will do later it is important to notice that equation 9 can be written as

$$\begin{bmatrix} \sum_{k=1}^N y_k z_1^{k-1} \\ \vdots \\ \sum_{k=1}^N y_k z_m^{k-1} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^N (z_1 z_1)^{k-1} & \dots & \sum_{k=1}^N (z_m z_1)^{k-1} \\ \vdots \\ \sum_{k=1}^N (z_1 z_m)^{k-1} & \dots & \sum_{k=1}^N (z_m z_m)^{k-1} \end{bmatrix} \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix} \quad (10)$$

which is just the system of normal equations that occur in least squares curve fitting. Equation 10 shows that if the poles  $s_i$  are known, then the equations for the residues  $A_i$  are linear. So it would appear that the difficult part of the problem is finding the poles.

## 2. NONLINEAR REGRESSION METHODS

In the statistical literature the problem is called a nonlinear regression problem. A good reference is Bard's book (ref. 4). Solutions are usually obtained by iterative techniques. Let  $\underline{W} = (s_1, \dots, s_m, A_1, \dots, A_m)$  = vector of parameters and let  $\underline{W}_i$  be the estimate of  $\underline{W}$  at the  $i^{\text{th}}$  iteration. Also let  $E(i)$  be the resulting error. Then an iterative technique will start from an initial guess  $\underline{W}_0$  and attempt to find a sequence  $\underline{W}_1, \underline{W}_2, \underline{W}_3, \dots$  such that  $E(1) > E(2) > E(3) > \dots$ . The calculations are terminated after  $E(i)$  reaches an acceptable value. The big problem is in finding  $\underline{W}_{i+1}$  given  $\underline{W}_i$ . We can write  $\underline{W}_{i+1} = \underline{W}_i + \underline{V}_i$  where  $\underline{V}_i$  is an increment in the parameter vector. A number of techniques have been proposed for choosing  $\underline{V}_i$ . Gradient methods appear to be best, but there is disagreement on this point (ref. 4, chapter 4). Let  $q_i = \frac{\partial E(i)}{\partial \underline{W}_i}$  = gradient vector evaluated at  $\underline{W}_i$ . Gradient methods choose  $\underline{V}_i = cR_i q_i$  where  $R_i$  is a positive definite matrix and  $c$  a positive scalar. It can be shown that if  $\underline{V}_i$  is chosen



this way, then  $E(i+1) < E(i)$  for some  $c > 0$  (ref. 4). The simplest gradient method is called steepest descent and puts  $R_i = I$ . Newton's method computes the Hessian Matrix,  $H(\underline{w}_i)$ , the matrix of second partial derivatives of  $E(i)$  with respect to the unknown parameters and sets  $R_i = H^{-1}(\underline{w}_i)$  and  $c_i = 1$ . A difficulty is that  $H^{-1}(\underline{w}_i)$  may not be positive definite. There are several modified Newton method schemes designed to overcome this problem. Gradient methods are very general and have been used successfully in pole extraction type problems (refs. 3 and 5). However, they have the disadvantage that the computation is larger than that of the methods described next. (For example, if a 20-pole function is used, then 20 residues are needed so the Hessian matrix is 40 by 40 and this matrix has to be inverted at each step.) Gradient methods need to be initialized with a good initial guess, and they may not converge to a global minimum. On the other hand, gradient methods have an advantage in not requiring equally spaced data.

### 3. DIFFERENCE EQUATION METHODS

The difference equation methods replace the problem of estimating poles with that of estimating the coefficients of a linear difference equation. The result is that a nonlinear estimation problem is replaced with a linear problem, but this presents a difficulty as shown below.

Prony's method (ref. 6) is a difference equation method. It is based on the fact that the  $x_k$  given by

$$x_k = \sum_{i=1}^m A_i Z_i^{(k-1)} \quad k = 1, \dots, N \quad (11)$$

where

$$Z_i = \exp(s_i T) \quad i = 1, \dots, M \quad (12)$$

satisfy the linear difference equation

$$x_k + a_1 x_{k-1} + \dots + a_m x_{k-m} = 0 \quad k \geq m+1 \quad (13)$$

where the  $a_i$  are related to the  $Z_i$  by

$$z^m + a_1 z^{m-1} + \dots + a_m = (z-Z_1)(z-Z_2) \dots (z-Z_m) \quad (14)$$

So, if the  $a_i$  can be found, then the  $Z_i$  can be found as in equation 14. Then, from equation 12 the poles are given by

$$s_i = \frac{1}{T} \ln Z_i \quad i = 1, \dots, m \quad (15)$$

Once the  $s_i$  are found the problem of finding the  $A_i$  is simply a linear regression problem. So the real effort must be directed towards getting the coefficients  $a_i$ . Reference 6 points out that if  $2m$  values of  $x_k$  are known, then equation 13 can be written as

$$\begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{2m} \end{bmatrix} = \begin{bmatrix} -x_m & -x_{m-1} & \cdots & -x_1 \\ -x_{m+1} & -x_m & \cdots & -x_2 \\ \vdots & \vdots & & \vdots \\ -x_{2m-1} & & \cdots & -x_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \quad (16)$$

so that the  $a_i$  can be found by solving the  $m$  linear equations 16. This procedure is useful where the  $x_k$  are known accurately (ref. 7)--for example, if the  $x_k$  have been calculated on a digital computer. Unfortunately, equation 16 is of little value in experimental work with high noise levels since the  $x_k$  are not known. The data that are available consist of  $x_k$  plus noise. As in equation 2,  $y_k = x_k + e_k$  is the available signal. Substituting this into the difference equation 13 we get

$$y_k + a_1 y_{k-1} + \cdots + a_m y_{k-m} = e_k + a_1 e_{k-1} + \cdots + a_m e_{k-m} \quad (17)$$

so the observations  $y_k$  obey a difference equation. Let  $w_k$  be the right side of equation 17; then

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} \cdots - a_m y_{k-m} + w_k \quad (18)$$

The  $w_k$  are called residuals. Now corresponding to equation 14 we have for the case with observation noise

$$\begin{bmatrix} y_{m+1} \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} -y_m & -y_{m-1} & \cdots & -y_1 \\ \vdots & \vdots & & \vdots \\ -y_{N-1} & -y_{N-2} & \cdots & -y_{N-n} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} w_{m+1} \\ \vdots \\ w_N \end{bmatrix} \quad (19)$$

where in equation 19 we have allowed for  $N \geq 2m$ . Since equation 19 contains the additive noise terms,  $w_k$ , it would seem that we should use  $N \gg 2m$  and find the least squares solution. This is not completely correct. The residuals are correlated as can be seen from equation 17. If we put

$$\underline{Y} = [y_{m+1} \cdots y_{n-1}]^T$$

$$H = \begin{bmatrix} -y_m & \cdots & -y_1 \\ \vdots & & \vdots \\ -y_{N-1} & \cdots & -y_{N-m} \end{bmatrix} \quad (20)$$

$$\underline{a} = [a_1 \cdots a_m]^T$$

$$\underline{W} = [w_{m+1} \cdots w_{N-1}]^T$$

then the minimum variance, unbiased least squares estimate for  $\underline{a}$  is

$$\underline{a} = (H^T R^{-1} H)^{-1} H^T R^{-1} \underline{Y} \quad (21)$$

where  $R$  is the covariance matrix of the residuals. Unfortunately,  $R$  is not known since it depends on  $a_i$ . If one simply uses the least squares solution as though the residuals were uncorrelated--that is, one puts  $R = I =$  identity matrix--then the solution

$$\underline{a} = (H^T H)^{-1} H^T \underline{Y} \quad (22)$$

is biased (ref. 8). This simply means that  $\underline{a}$  will not equal the best  $\underline{a}$  no matter how much data are used. ("Best  $\underline{a}$ " means the  $a_i$  that in turn gives the poles and residues that minimize the squared error  $E^2$ .) To summarize all this, Prony's method has the advantage that it makes the poles easy to find but the disadvantage that when observation noise is present the poles are incorrect.

A number of schemes have been devised to handle the problem of correlated residuals. (See references 8, 9, and 10 for more detailed discussions.) Although the schemes vary a great deal in implementation, they all attempt to whiten the residuals using some iterative technique. To explain how this works,

it will be convenient to change the notation in equation 18. Let

$$A(Z^{-1}) = 1 + a_1 Z^{-1} + \dots + a_m Z^{-m} \quad (23)$$

where  $Z^{-1}$  is the backwards shift operator, that is

$$Z^{-1}y_k = y_{k-1} \quad (24)$$

Using this notation, the difference equation 18 can be written as

$$A(Z^{-1})y_k = w_k \quad (25)$$

Now assume that the correlated residuals  $w_k$  can be represented as

$$w_k = \frac{c(Z^{-1})}{D(Z^{-1})} v_k \quad (26)$$

where  $c(Z^{-1}) = 1 + c_1 Z^{-1} + \dots + c_p Z^{-p}$ ,  $D(Z^{-1}) = 1 + d_1 Z^{-1} + \dots + d_p Z^{-p}$  and where  $v_k$  is a white noise sequence. Then equation 25 becomes

$$A(Z^{-1})y_k = \frac{c(Z^{-1})}{D(Z^{-1})} v_k \quad (27)$$

The least squares estimate for the parameters of this model is the one that

minimizes  $\sum_{k=1}^N v_k^2$ . The problem has evolved to the point that in order to estimate

the  $a_i$ , we also have to estimate the additional parameters  $c_i$  and  $d_i$ . The maximum likelihood method of Astrom and Bohlin (ref. 11) provides a solution to the problem. Unfortunately, the equations are nonlinear in the  $c_i$  and a gradient algorithm is required. However, maximum likelihood appears to be one of the best available techniques. A number of schemes that are simpler to implement come from setting the  $c_i = 0$ . Then equation 27 becomes

$$A(Z^{-1})D(Z^{-1})y_k = v_k \quad (28)$$

Although equation 28 is linear in the parameters, it may be necessary to greatly increase the number of  $d_i$  when the  $c_i$  are set to zero. This is because in

general one expects that a moving average-autoregressive process as in equation 26 will be a more economical representation (that is, have lower order) than an autoregressive process alone. At any rate, estimates for the parameters in equation 28 have been calculated using generalized least squares (refs. 8, 9, and 11) and repeated least squares (refs. 8, 11, and 13). The author attempted to use generalized least squares. The method seemed to work well on simulated data but did not converge when real data were used. The repeated least squares method seems to be moderately successful as discussed below.

Before leaving the general subject of difference equation methods, it should be pointed out that the method will work with arbitrary driving functions. Equation 18 can be generalized to

$$y_k + a_1 y_{k-1} + \dots + a_m y_{k-m} = b_0 u_k + \dots + b_m u_{k-m} + w_k \quad (29)$$

where  $u_k$ ,  $k = 1, 2, \dots, N$  is a known input. Prony's method corresponds to the special case where  $u_k$  is taken as a unit pulse. All the schemes for estimating the coefficients  $a_i$  still work, and there is no need to wait for the driving function to die out as is sometimes done with Prony's method. This would avoid losing data at the beginning of a record. (Unfortunately, the EMP problem is more complicated.)

In this general case, the equation corresponding to equation 19 is

$$\begin{bmatrix} y_{m+1} \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} -y_m & -y_{m+1} & \dots & -y_1 & u_{m+1} & \dots & u_1 \\ \vdots & & & & & & \\ -y_{N-1} & \dots & & & -y_{N-m} & u_M & \dots & u_{N-m} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \\ b_0 \\ \vdots \end{bmatrix} + \begin{bmatrix} w_{m+1} \\ \vdots \\ w_N \end{bmatrix} \quad (30)$$

Here there are  $2m+1$  parameters to be estimated since the  $b_i$  are unknown. However, the residues can be computed directly from the  $b_i$  without requiring another least squares fit to the original data. Expressed in discrete transfer function form, the equations are

$$A(Z^{-1})y_k = B(Z^{-1})u_k + w_k \quad (31)$$

where

$$B(Z^{-1}) = b_0 + b_1 Z^{-1} + \dots + b_m Z^{-m}$$

#### 4. REPEATED LEAST SQUARES (RLS) PROGRAM

The RLS technique (called the covariance method in reference 13) is based on the difference equation 28. In equation 28 the transfer function represents the actual physical system, and  $D(Z^{-1})$  represents the filter required to whiten the noise and remove the bias in the parameter estimates. The scheme is easy to implement. One simply uses least squares to estimate the parameters  $f_i$  where

$$F(Z^{-1}) = 1 + f_1 Z^{-1} + \dots + f_j Z^{-j} = A(Z^{-1})D(Z^{-1}) \quad (32)$$

For a large enough value of  $j$  the residuals  $v_k$  of equation 28 should be white so unbiased estimates can be obtained. Given estimates of the  $f_i$  we still have the problem of finding the poles. Notice these poles are just the zeros of  $Z^m A(Z^{-1})$  (eq. 14), so the zeros of  $Z^j F(Z^{-1})$  consist of the zeros of  $Z^j D(Z^{-1})$  and the zeros of  $Z^j A(Z^{-1})$ . That is, the poles due to the system and to the noise will all be found. Which are which can be decided by making some additional assumptions. Fortunately, when analyzing real data we are usually able to compare the estimated poles with the original data record  $y(t)$  and with the Fourier transform magnitude of  $y_k$  and make a reasonable guess.

A flow chart of the program that has been implemented on CDC 7600 and 6600 computers is shown in figure 1. At the present time it uses two AFWL library programs: DECOMP, a program for solving linear equations, and ZRPRC, a program for finding the roots of polynomials. The call to DECOMP can be eliminated with a sequential regression algorithm. This algorithm has been tested and is used in the pole finding part of the program.

The flow chart shows a low-pass filtering operation on  $y_k$  (optional). Usually using the filter allows smaller squared error for a given number of poles, but the filtering operation can cause mistakes. This effect is discussed in the conclusions. The filter used consists of NS sections of 2-pole Butterworth digital filters (ref. 15). The cutoff frequency and NS can be set in the program. The filter is useful if the system poles are known to lie in some low-pass region. The program also has the option of using only poles in the user defined low-pass region when finding the residues. This can be done with or without filtering. The program is initialized by specifying the data file to be read, the number of data points, the upper cutoff frequency, and the number of poles and sampling interval in the data. At present, the number of poles can be between 2 and 80 and can be incremented in a DO loop.

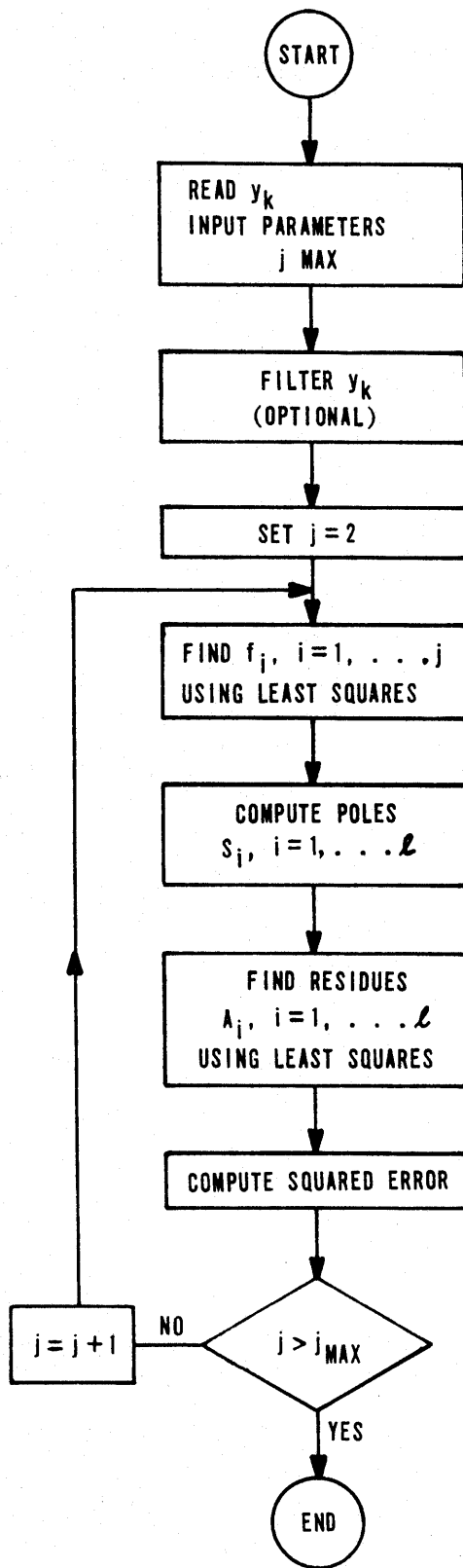


Figure 1. Flowchart for RL3

Output consists of all poles within the specified low-pass region and the corresponding residues plus the computed squared error for each number of poles.



### SECTION III

#### RESULTS

This section contains several examples showing how the RLS program works. Each example was chosen to illustrate some particular point.

EXAMPLE 1. This first example shows what happens when "nice" data are used. A data array was generated on the computer as

$$y(t) = \exp(-1.5t)\sin(20\pi t) + \exp(-0.5t)\cos(10\pi t) \quad (33)$$

for  $t = 0.0, 0.01, 0.02, \dots, 1.99$ . The RLS was run on this data array for two cases.

Case 1. Number of data points =  $2 \times$  (number of poles) = 8. The results are

<u>POLES</u>	<u>RESIDUES</u>
$-1.500001 \pm j 62.83185$	$-0.3955032 \times 10^{-7} \pm j (-0.5000000)$
$-0.5000012 \pm j 31.41593$	$0.5000000 \pm (-0.3950878 \times 10^{-7})$

which agree well enough to justify using the minimum number of data points when the data are known accurately--in this case 14 decimal accuracy.

Case 2. Number of data points = 200. The results are

<u>POLES</u>	<u>RESIDUES</u>
$-1.500000 \pm j 62.83185$	$-0.6065163 \times 10^{-9} \pm j (-0.5000000)$
$-0.5000012 \pm j 31.41593$	$0.5000000 \pm j (-0.5360294 \times 10^{-9})$

which is slightly better than the first case.

EXAMPLE 2. In this example we see what can happen if the data array is not known as accurately as in example 1. An array was generated according to equation 33, multiplied by 250 and truncated. This simulates that kind of data we can expect from a 10-bit ADC since we have roughly three significant figures. This lack of precision has an effect similar to that of observation noise.

Case 1. Number of data points =  $2 \times$  (number of poles) =  $2m$ . Poles with  $m = 4$ :

$$-1.265881 \pm j (53.54892)$$

$$-120.6585 \pm j (106.2812)$$

There is considerable error in the poles. If the number of poles asked for is increased the situation improves. With 12 poles the result is

Poles with  $m = 12$ :

$$3.503043 \pm j (283.7881)$$

$$-4.205054 \pm j (284.3589)$$

$$-9.690239 \pm j (176.6259)$$

$$4.373128 \pm j (119.2774)$$

$$-1.523546 \pm j (62.80804)$$

$$-0.4787273 \pm j (31.40865)$$

The last two pole pairs are the system poles and the first four pairs are the result of truncation noise. Notice that two pairs have positive real parts.

Case 2. Number of data points = 200. With all the available data used and 4 poles requested, RLS computed:

$$-2.143601 \pm j (62.98279)$$

$$-1.397793 \pm j (31.63991)$$

These poles are more accurate than those found in case 1, but show the effect of truncation noise. With 12 poles:

$$-14.56595 \pm j 235.1918$$

$$-12.65917 \pm j 284.3796$$

$$-15.61268 \pm j 180.8489$$

$$- 7.641336 \pm j 124.337$$

$$- 1.501462 \pm j 62.83148$$

$$- 0.5015126 \pm j 31.41628$$

The last two pairs are system poles and the first four pairs are the result of truncation noise. The results for this case are more accurate and there are no poles with positive real parts.

The first two examples show why it is advisable to use all the available data rather than the minimum amount. We will not look at the minimum amount case in the remaining examples. It appears that the real part of poles is harder to find accurately than the imaginary part.

EXAMPLE 3. This example shows what happens when the data have even fewer significant figures than in example 2. This kind of situation can arise when the amplitude of a waveform to be digitized does not span the ADC's input range. The data array was generated by multiplying equation 33 by 100 and then truncating. With four poles asked for, the results are:

$$-5.740473 \pm j 63.58047$$

$$-5.327517 \pm j 32.25558$$

and with 12 poles:

$$-11.19230 \pm j 174.3681$$

$$-10.00480 \pm j 228.2428$$

$$- 9.900576 \pm j 285.1623$$

$$-10.67594 \pm j 121.3942$$

$$- 1.504183 \pm j 62.83713$$

$$- 0.5076418 \pm j 31.41431$$

Again we see that in order to find the system poles accurately, it is necessary to ask for extra poles. This is because of the bias problem explained in section II.3. It is up to the person analyzing the data to decide which poles belong to the system and which do not. At this time the best way to make this decision seems to be to look at the Fourier transform of the original data--a process that is used in the remaining examples.

The first three examples use computer generated data. In order to test RLS further, some tests were run on data digitized from photographs. The photographs in examples 4 and 5 were supplied by Dr. R. L. Hutchins of The BDM Corporation; they were generated by photographing the impulse of real networks. The Fast Fourier Transform (FFT) of the data was computed on an HP9830 computer. All the graphs were done on the 9830. The photographs used in the following examples are pictures of oscilloscope traces. Several types of errors limit data quality. The three best known oscilloscope errors are produced by a nonlinear sweep rate, a variable trace width, and a misalignment of the scope's electron gun and front face which produces a nonlinear distortion called the keystone effect.\* Other errors are introduced when the photographs are digitized.

EXAMPLE 4. The waveform from this example as digitized and plotted on the 9830 is the damped sine wave shown in figure 2. The corresponding FFT magnitude

\*Conversation with Mr. John Podlesny, AFWL.

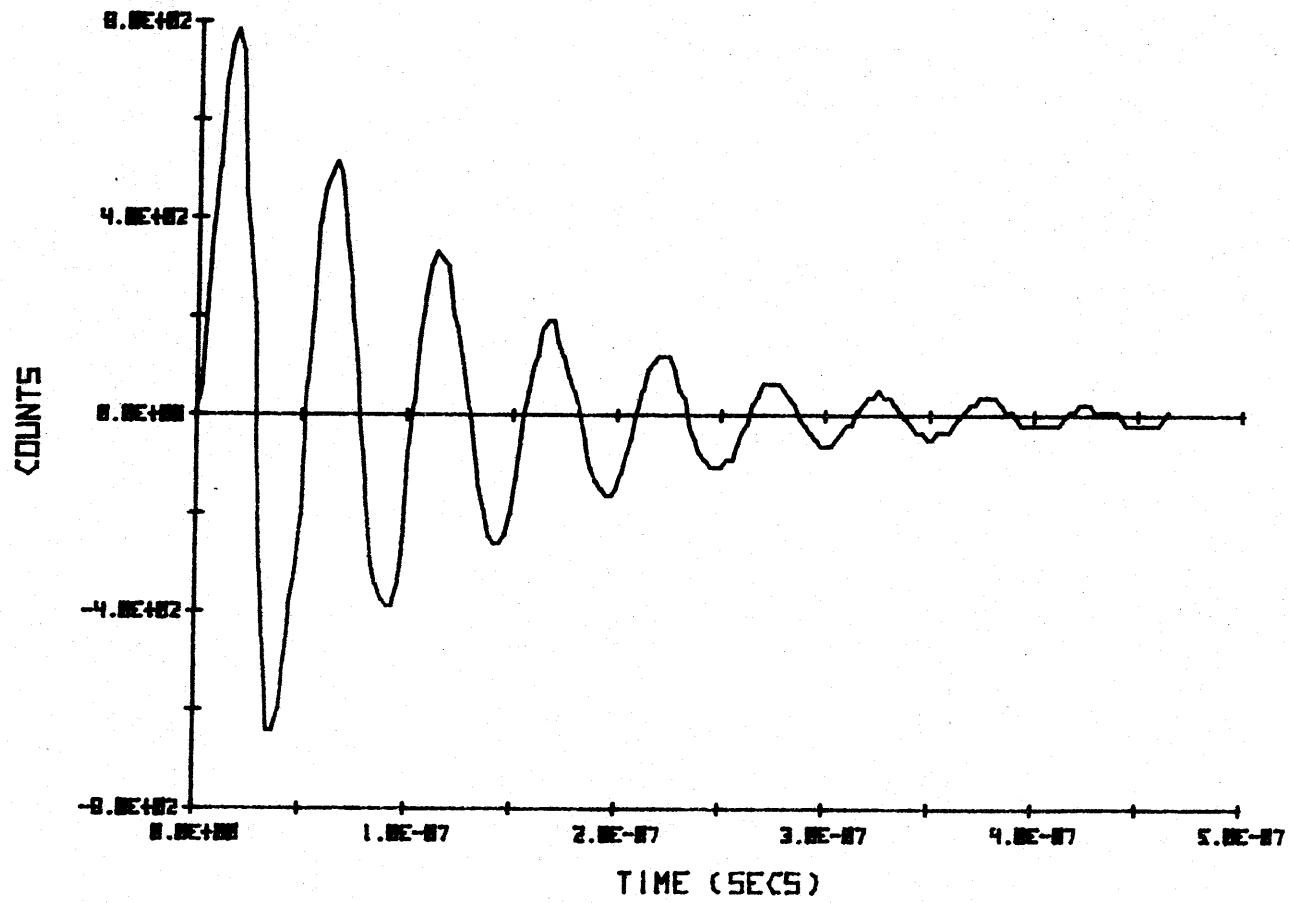


Figure 2. Damped Sine Wave for Example 4

appears in figure 3. The fundamental frequency is approximately 19 MHz. Smaller peaks at about 38 or 57 MHz, etc., suggest some kind of harmonic distortion. The remaining peaks at higher frequencies are beyond the 30 to 40 dB resolution of the recording system and can be taken as the result of noise.

Figure 4 shows a reconstruction of the waveform using poles and residues from the time waveform calculated by RLS. The corresponding FFT magnitude is in figure 5. The number of data points used in the calculation was 185. The poles were found by asking for 60 poles. Only poles with frequencies below 60 MHz (in this case a total of 20 poles) were used in the residue calculation and reconstruction. The squared error was 162. This graph is fairly smooth above 60 MHz because only poles with lower frequencies were used in the reconstruction. When all 60 poles are used in the reconstruction, the squared error is 42.6. The error is less since the extra poles fit the noise. This brings up an important point. Increasing the number of poles will usually decrease the squared error. We can decide which poles correspond to the system by looking at the original waveform in figure 2 and at its FFT magnitude in figure 3 and inferring that the only "real" pole-pair is the one with frequency around 19 MHz. An interpretation like this is always required to separate system and noise or distortion poles.

Next, the original waveform was analyzed by filtering it with  $NS = 2$  sections and a cutoff frequency  $FC = 60$  MHz. Figures 6 and 7 show the results. With 60 poles the squared error was only 17.6. Twenty-four poles were used in the reconstruction. Here the error is that between the filtered waveform and the reconstruction. The error is much less than that in the reconstruction shown in figure 4 since the high frequency noise was removed.

EXAMPLE 5. In this example, a photograph of a signal containing two damped sine waves was used. The photograph was digitized to 180 points. A graph of the digitized waveform is shown in figure 8. Its FFT magnitude in figure 9 shows two large peaks and a number of smaller peaks of much lower amplitude. Experience has shown that a waveform with an FFT like this is rather easy to analyze with RLS. When 56 poles were asked for and only those with frequencies less than 35 MHz (a total of 14 poles) used in the reconstruction, the squared error was 118. The reconstructed function and its FFT magnitude are shown in figures 10 and 11. As can be seen, the reconstruction agrees very well with the original waveform. When all 56 poles were used in the reconstruction, the

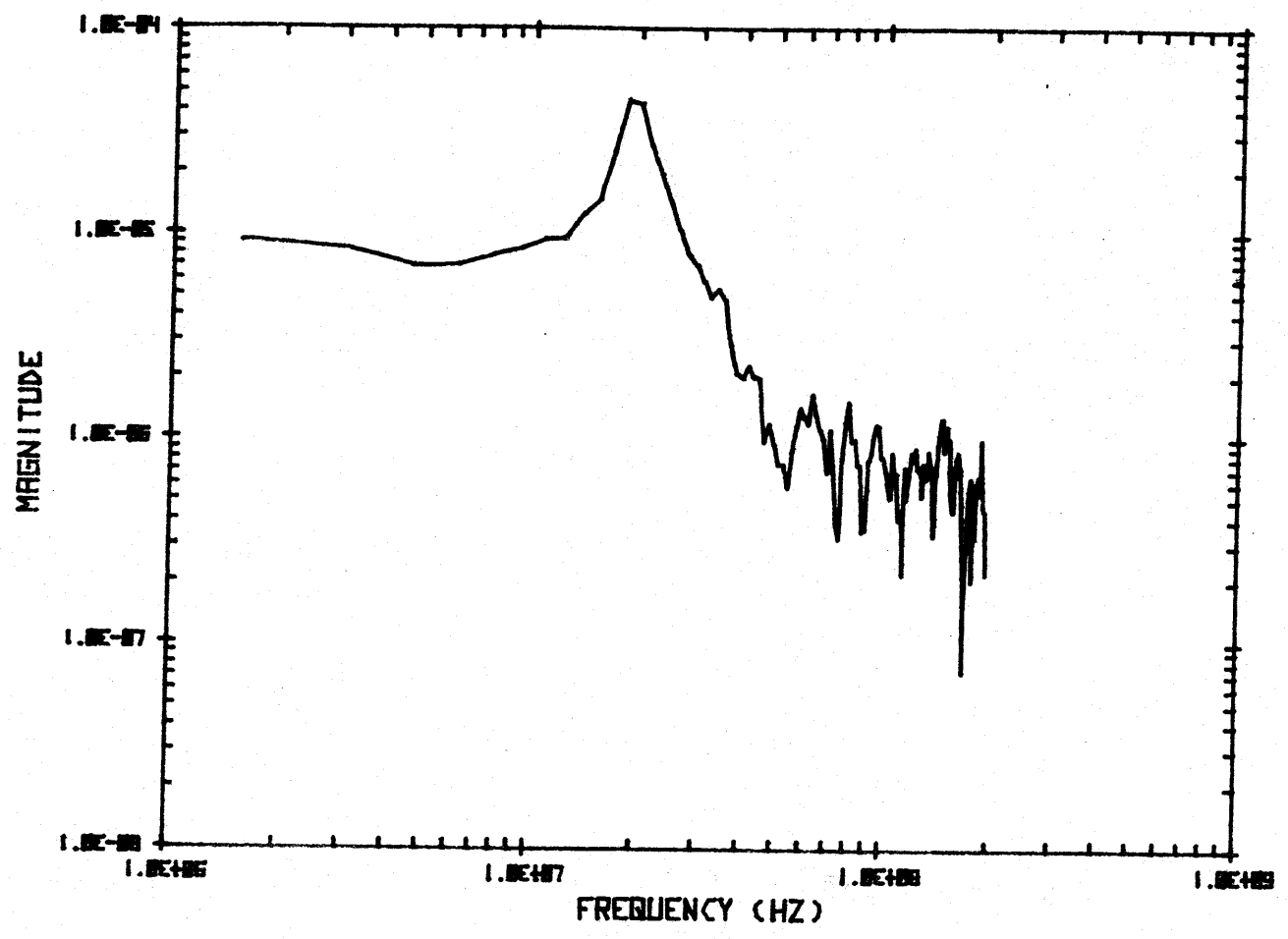


Figure 3. FFT Magnitude of Damped Sine Wave

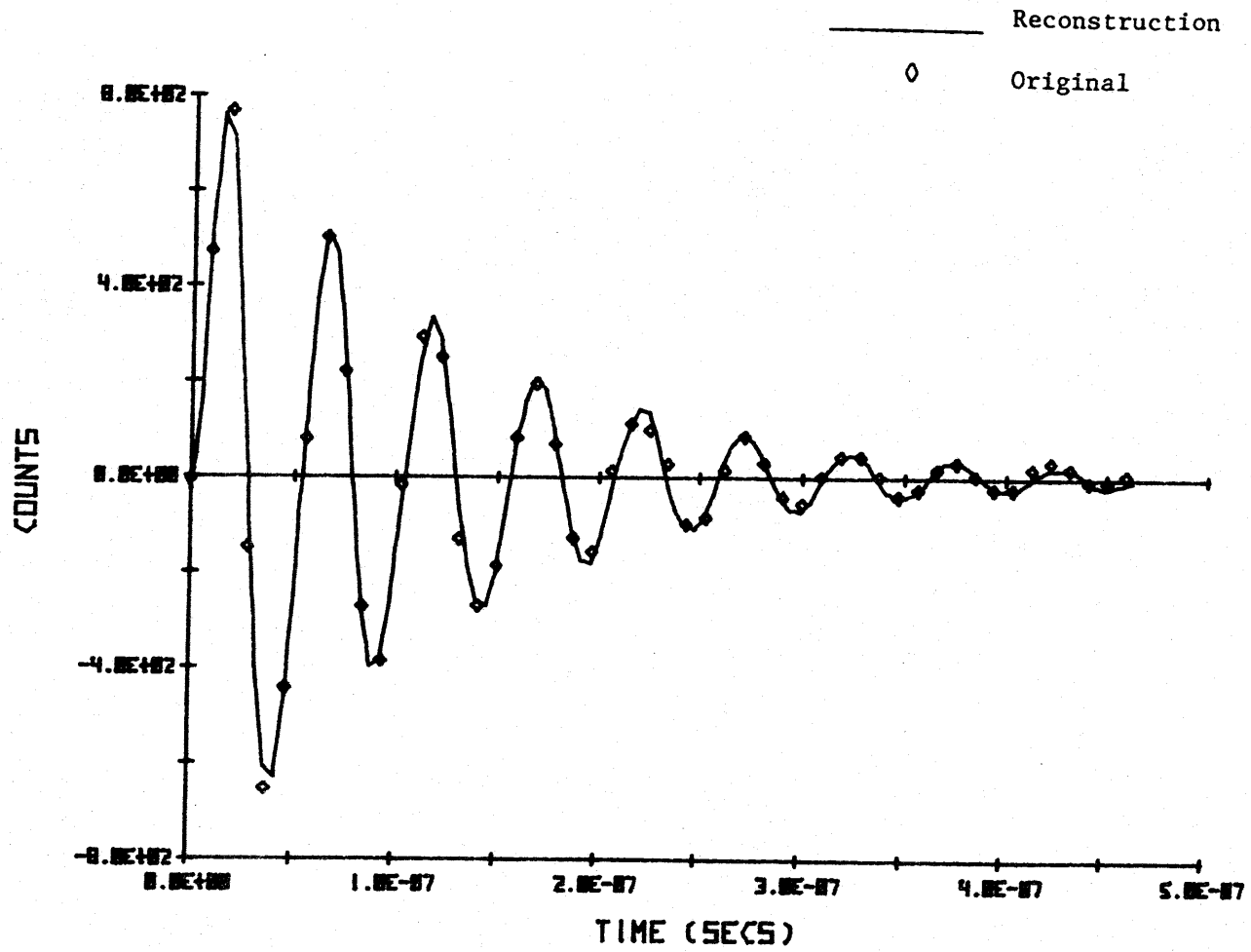


Figure 4. Reconstructed Damped Sine Wave Compared with Original

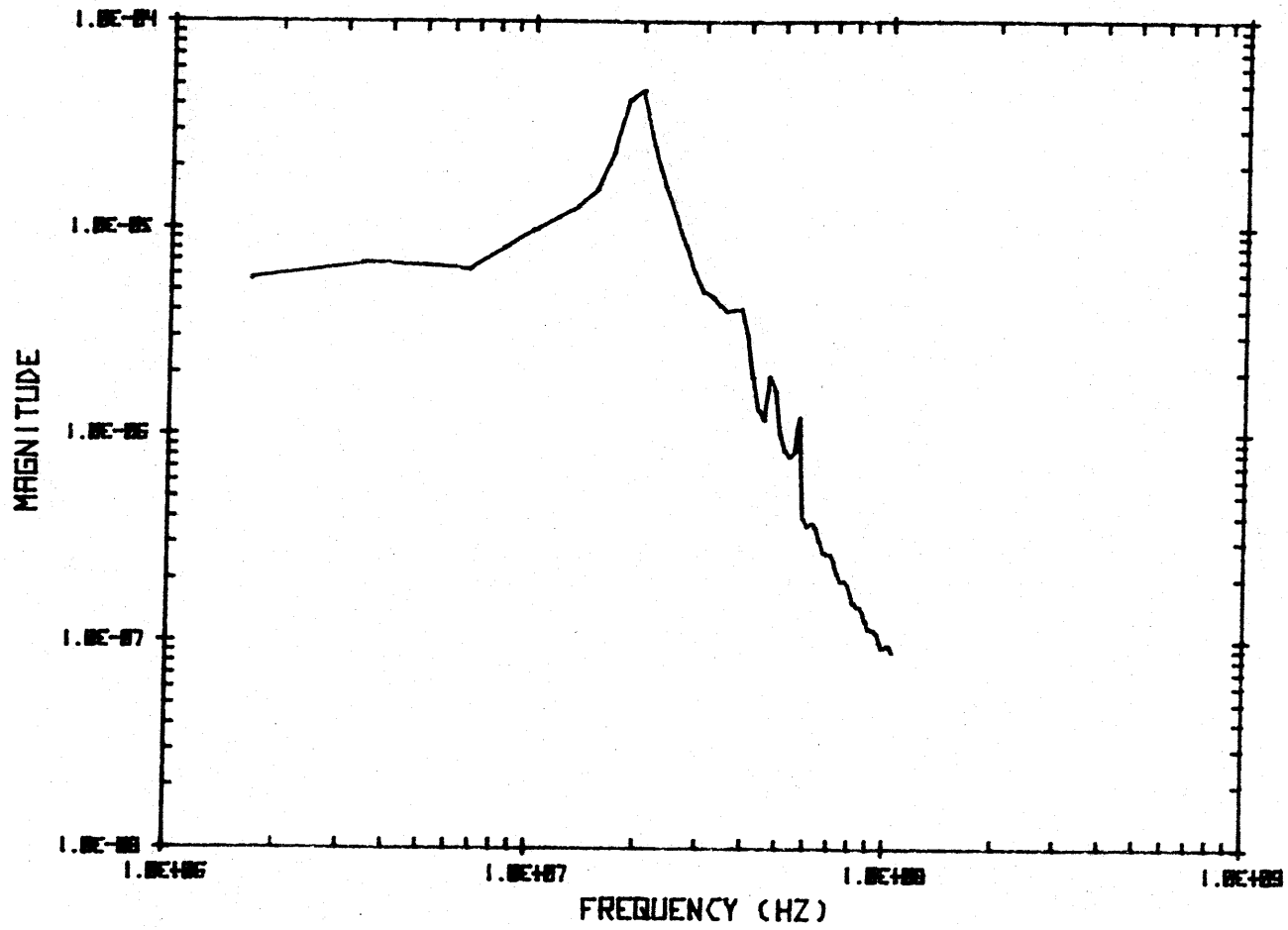


Figure 5. FFT Magnitude of Reconstructed Damped Sine Wave



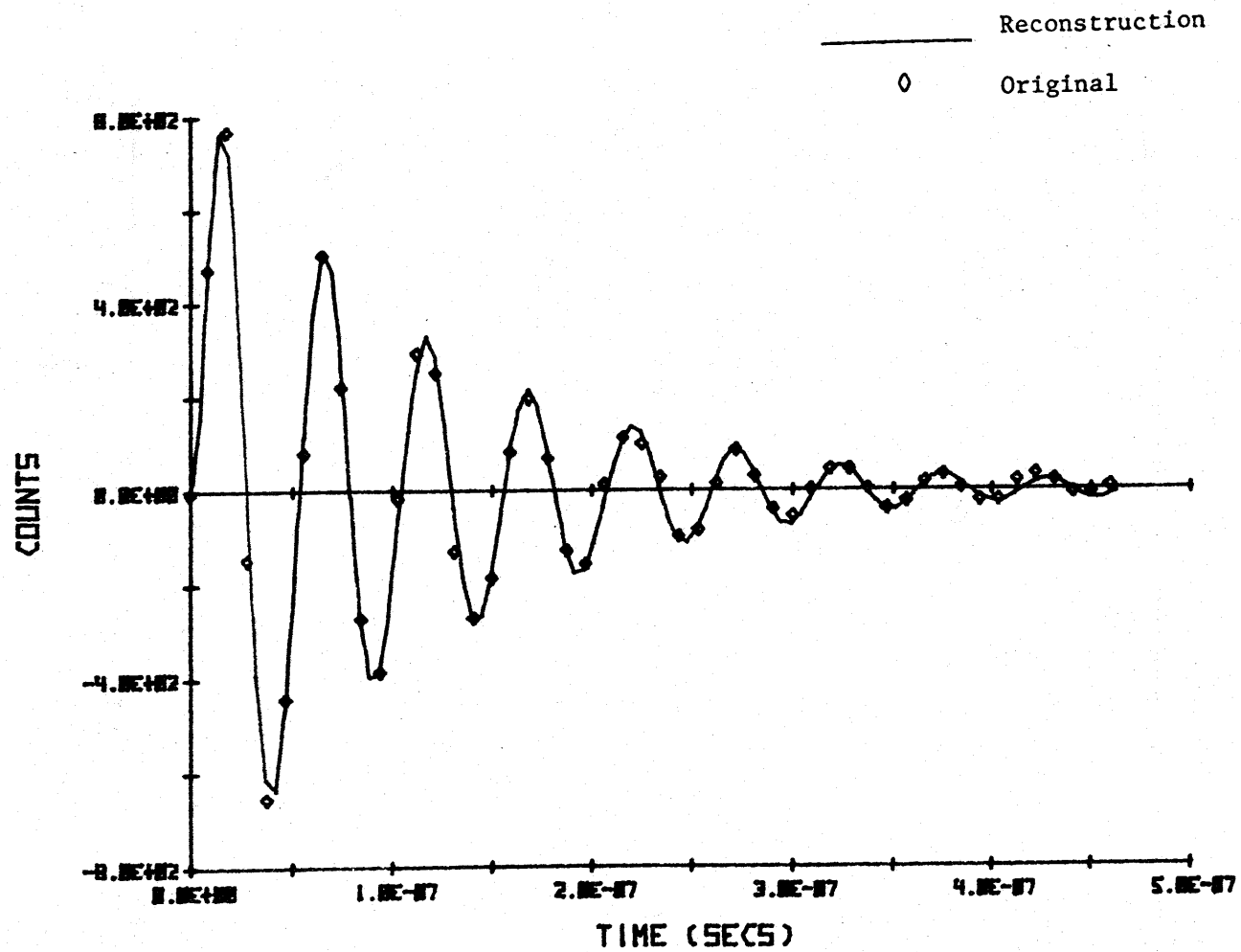


Figure 6. Reconstruction of Filtered Damped Sine Wave

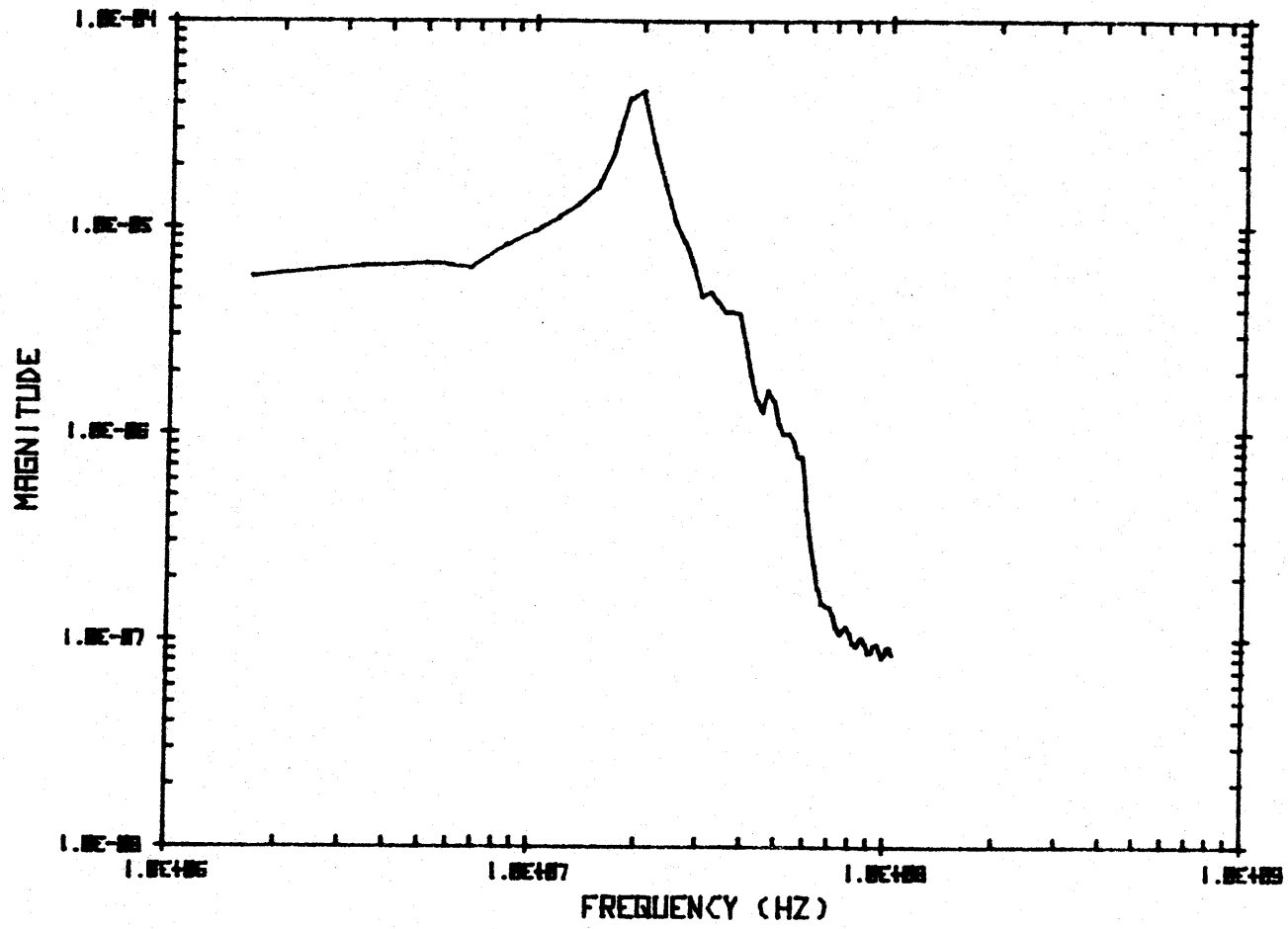


Figure 7. FFT Magnitude of Filtered Reconstruction for the Damped Sine Wave

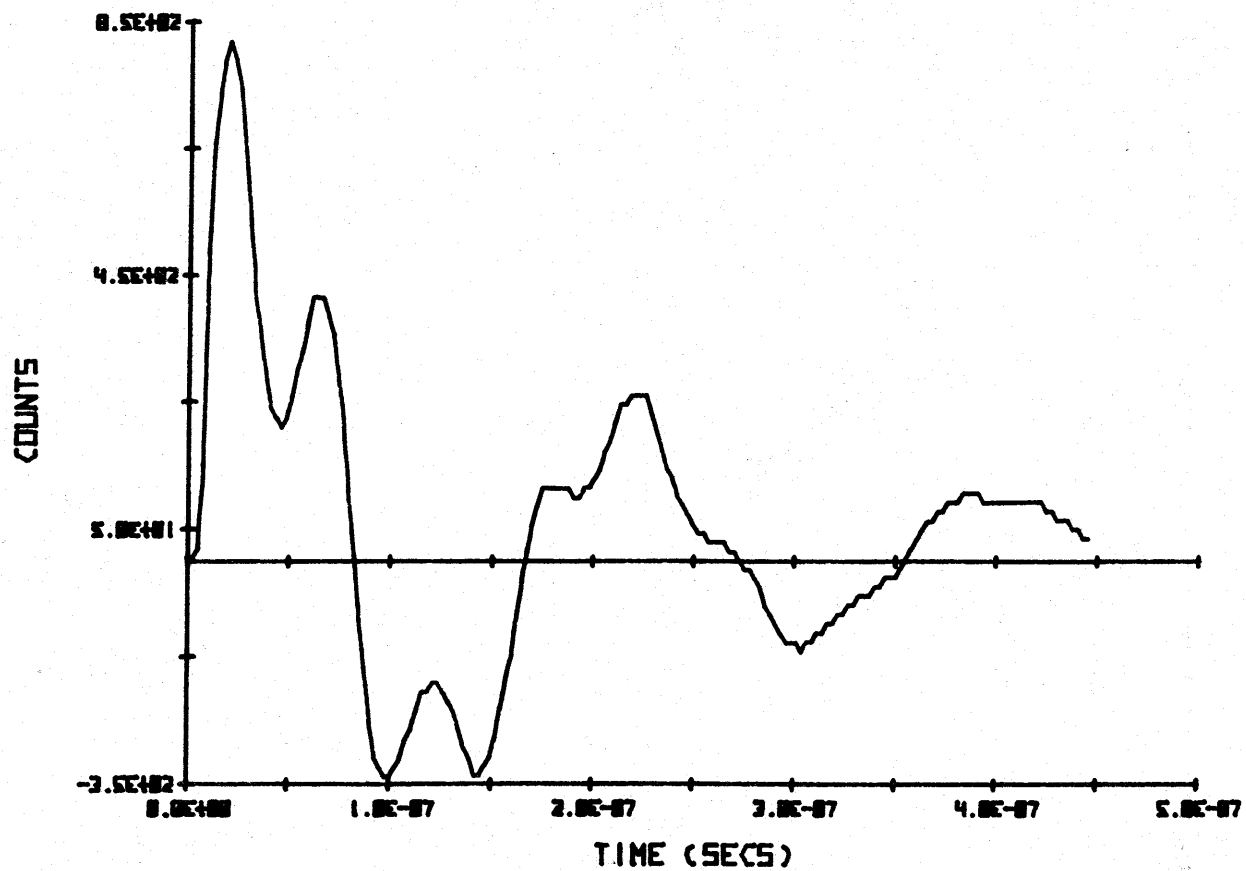


Figure 8. Two Damped Sine Waveforms Used in Example 5

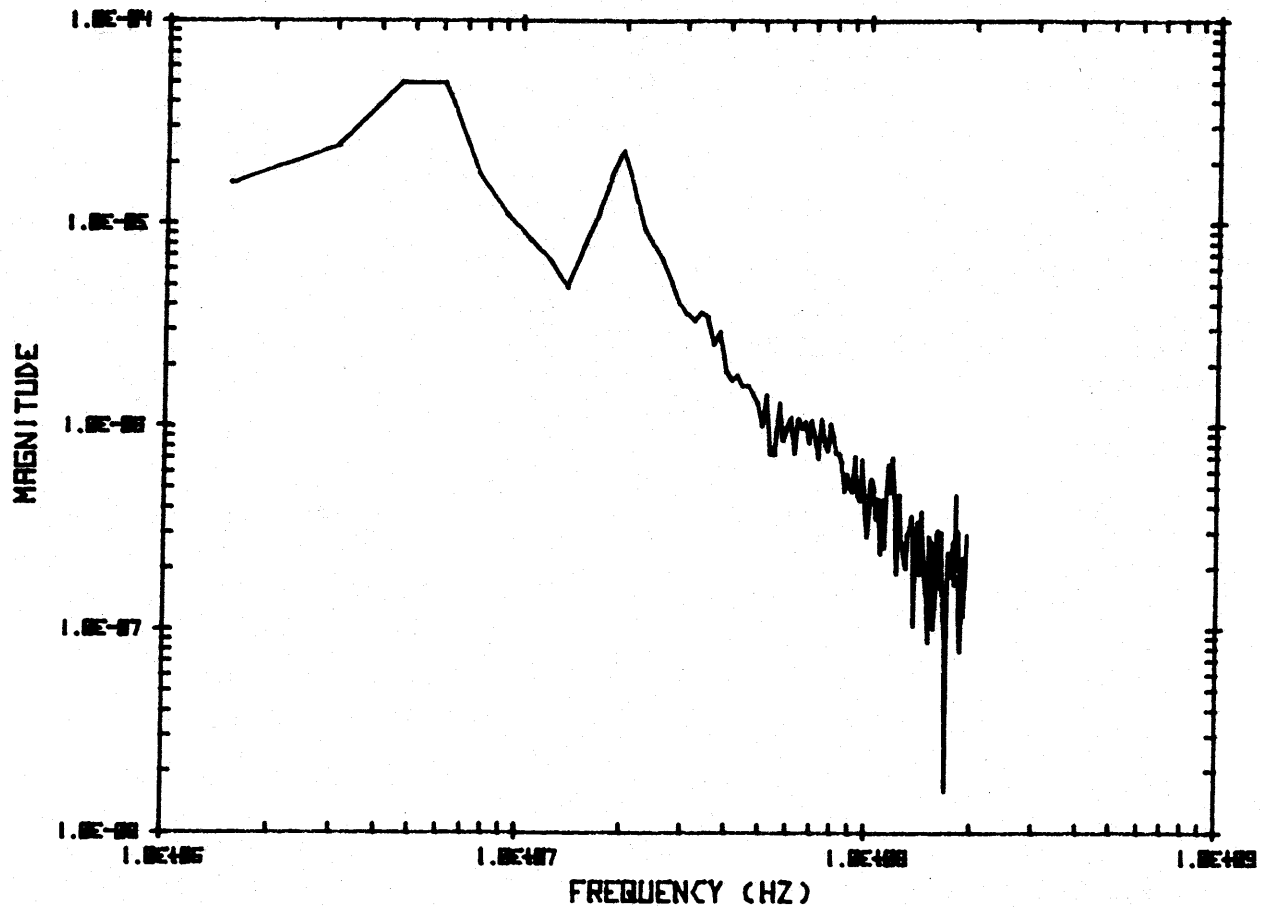


Figure 9. FFT Magnitude of Two Damped Sine Waveforms

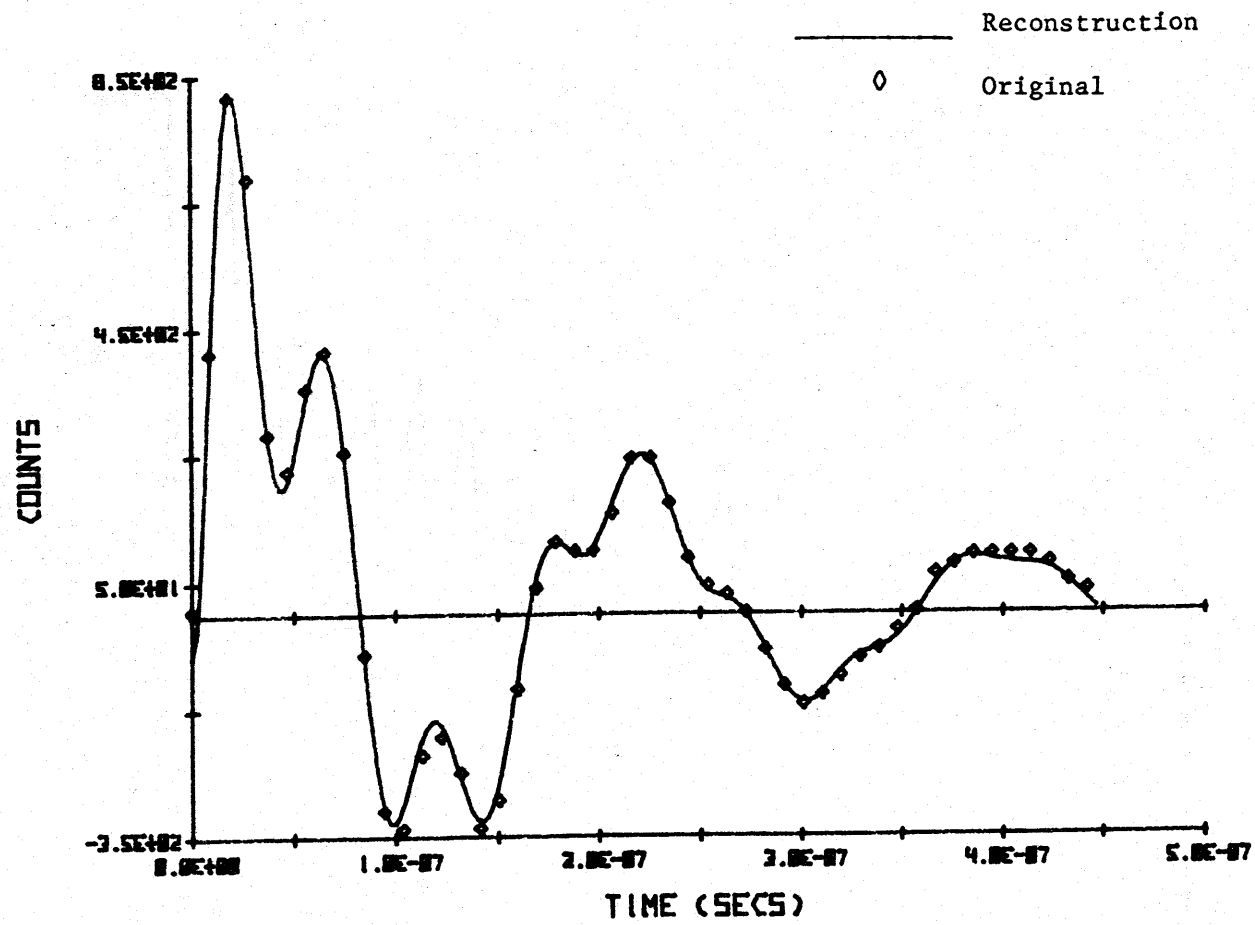


Figure 10. Reconstruction of Two Damped Sine Waveforms

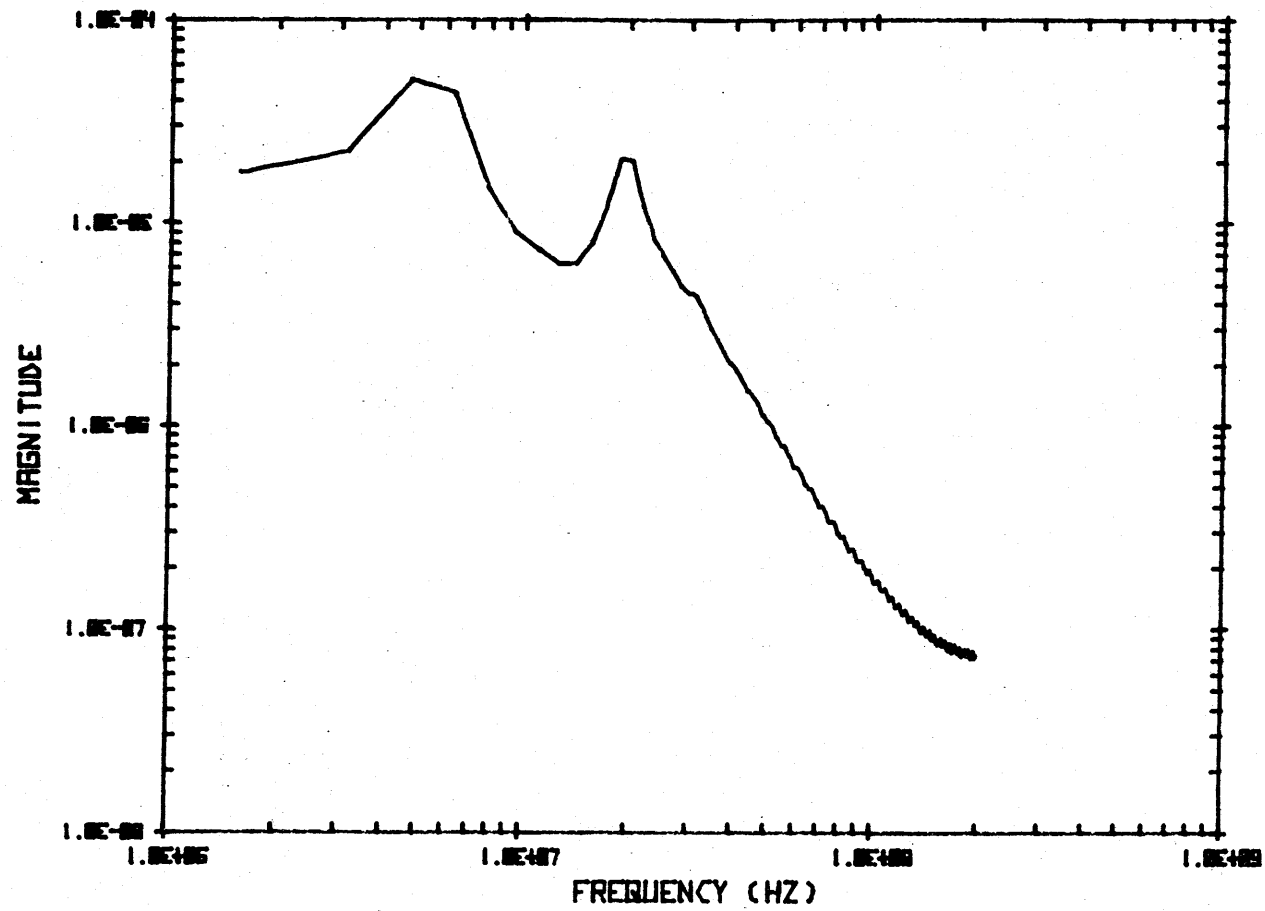


Figure 11. FFT Magnitude of Reconstructed Two Damped Sine Waveforms

squared error was 6.37. This low error probably is due to the fact that the original digitized waveform was very clean.

EXAMPLE 6. This last example uses a photographed "Green Box" response to a double exponential input. The Green Box is a circuit box that has been used to test AFWL transient data recording equipment (ref. 16). The photograph was digitized to 200 points. A graph of the digitized waveform is given in figure 12 and its FFT magnitude in figure 13. The FFT magnitude graph does not extend below about 1 MHz since limitations on the digitizer constrained the frequency sampling interval to this value. The figures show large components at about 1 MHz and 25 MHz and a relatively low amplitude component at about 5 MHz. The remaining peaks probably are due to noise and distortion. As mentioned earlier, this kind of statement is just an educated guess. At any rate, the remaining peaks are outside the instrumentation dynamic range. The peak at 5 MHz is only marginally within range. This last fact made the Green Box response the most difficult of the example waveforms to analyze. However, the results look good. The poles were found by asking for 60 poles. Then poles with frequencies less than 35 MHz (a total of 22 poles) were used to calculate the residues. The resulting squared error was 59.6. The reconstructed function is shown in figure 14 and its FFT magnitude in figure 15. Figures 13 and 15 must be compared carefully as the vertical scales are different. When all 60 poles were used in reconstruction, the squared error was reduced to 16.4.

In analyzing these examples, using all the available data, RLS found no poles with positive real parts. The poles and residues for the last three examples are not printed here since no theoretical values are available for comparison. The comparisons between original and reconstructed waveforms in figures 2 through 15 show that the method works if used carefully.

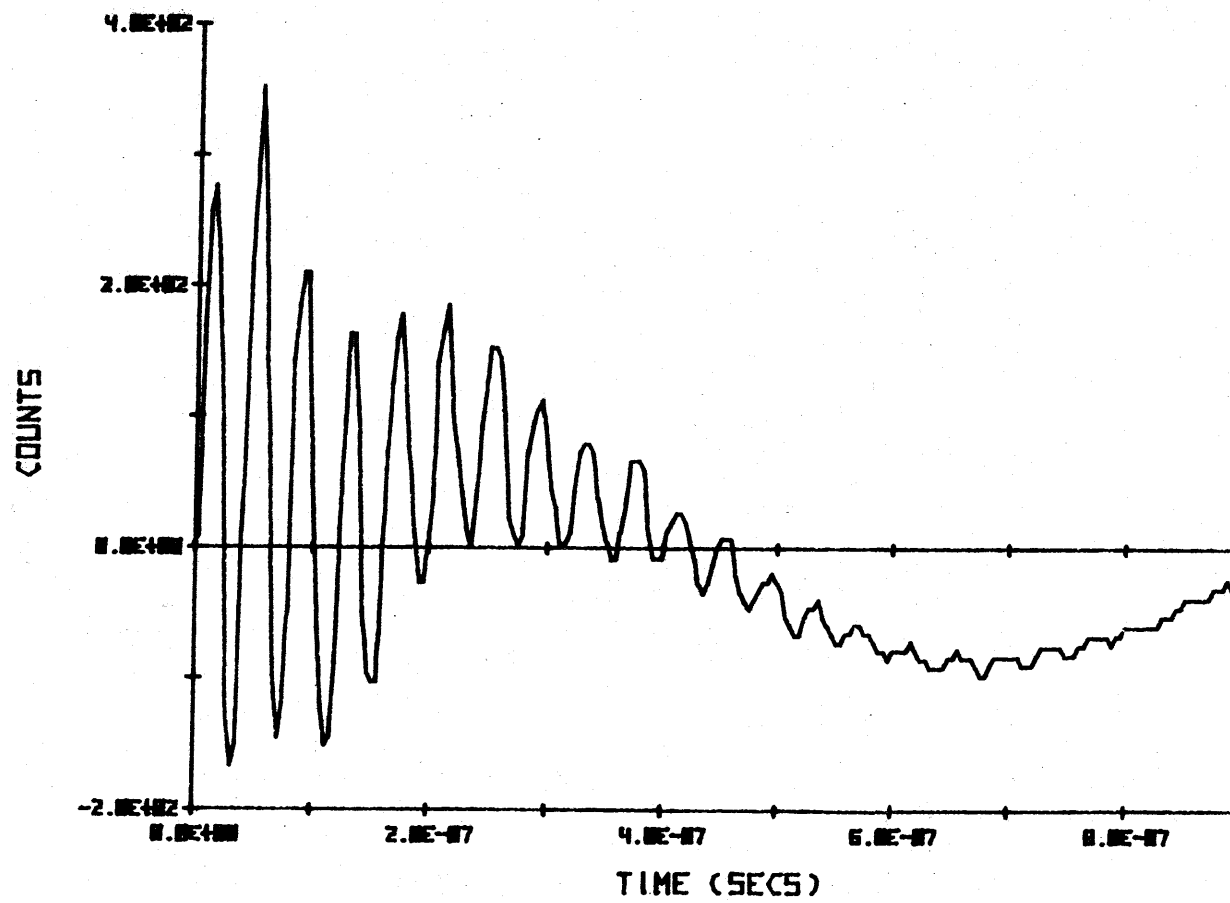


Figure 12. Green Box Response



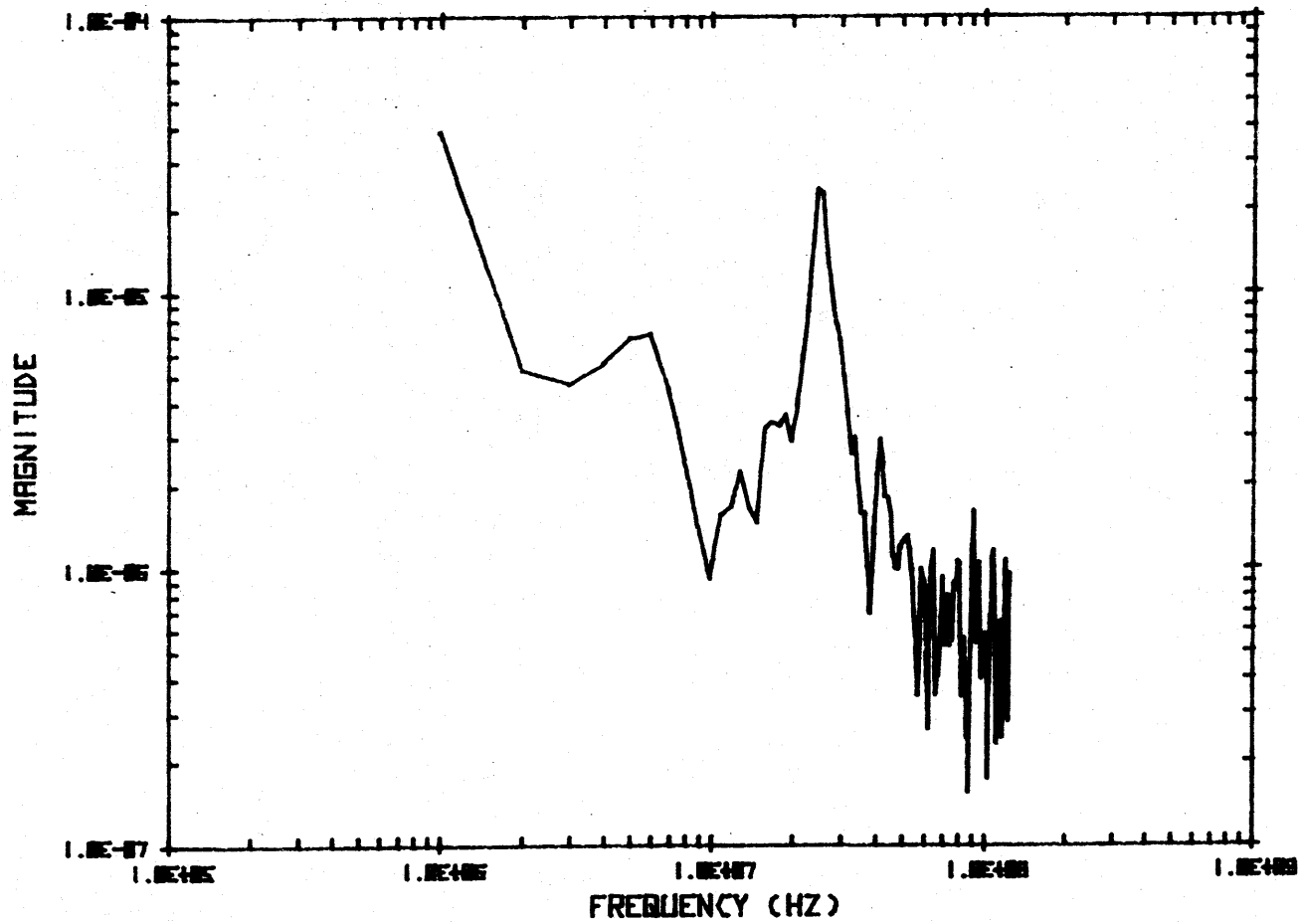


Figure 13. FFT Magnitude of Green Box Response

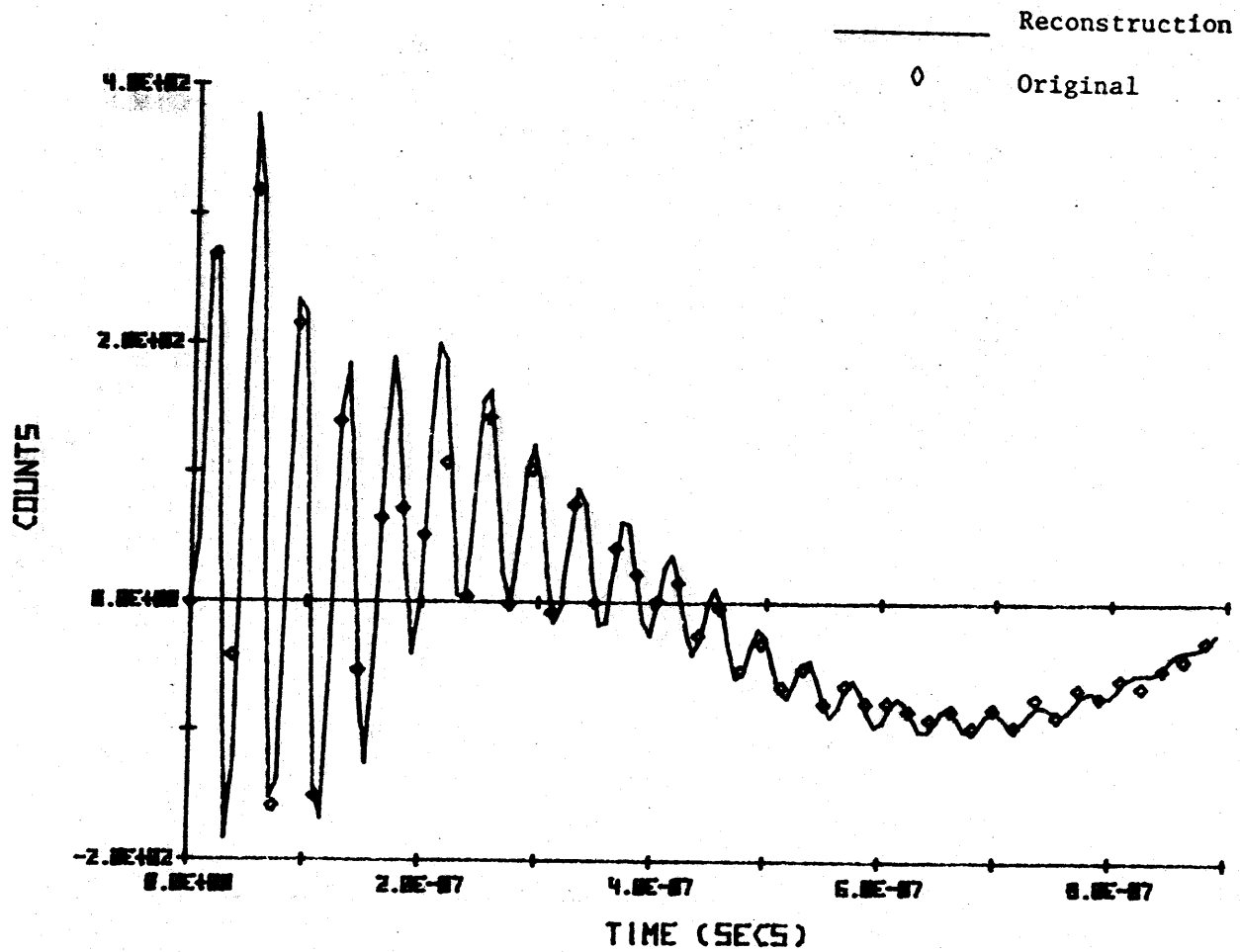


Figure 14. Reconstructed Green Box Response

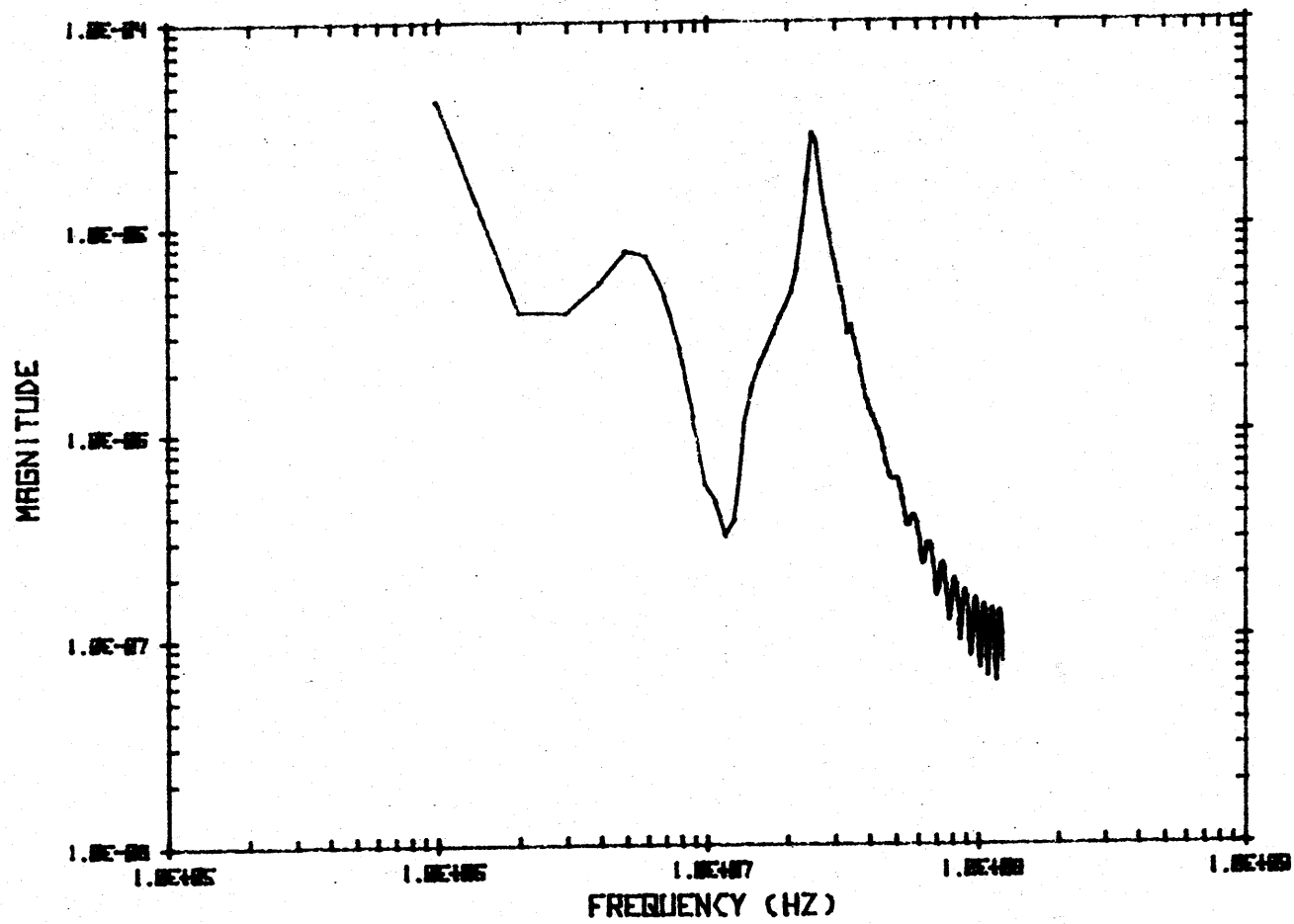


Figure 15. FFT Magnitude of Reconstructed Green Box Response

## SECTION IV

### CONCLUSIONS

The RLS method has worked well on all the examples tested. Of the various pole extraction techniques discussed, RLS is the easiest to program and probably the fastest to run.

Some general conclusions can be made about its use. The examples show that, unless the observed waveform is known very accurately, better results are obtained by using all available data ( $N > 2m$ ) rather than the minimum amount of data ( $N = 2m$ ). This allows for least squares solutions for the difference equation coefficients and for the residues. The examples also show that the order of the difference equation,  $m$ , must be taken larger than the number of system poles. A simple explanation for this is that the effect of noise is to put extra peaks in the spectrum of the observed waveform. These peaks must be fitted with extra poles. Deciding which poles correspond to the system and which to noise still requires some judgment. In practice, the decision can be made by looking at the observed waveform and its FFT.

Low-pass filtering the observed waveform to remove high frequency noise produces mixed results. For the same number of poles, the error between a filtered waveform and its reconstruction is smaller than the error between the same waveform unfiltered and its reconstruction. This does not mean that the poles or residues are more accurate. Unless the filter is used very carefully, it can cause errors in the real part of the poles. Apparently this is due to the filter attenuating part of the high frequency components of a damped sinusoid. Another point is that although filtering will remove noise poles, it will put in other poles to account for the sharp cutoff produced in the spectrum. At this time we have to conclude that there is no clear advantage to filtering. However, this point is worth more study. Some type of spectrum smoothing may be better than sharp cutoff low-pass filtering.

One objective of this work has been to decide if pole extraction can be done on ADSET's PDP-11/40. Most of the RLS computation involves solving the least squares normal equations twice, plus finding the roots of an  $m$ th order polynomial. These calculations can be segmented. Several improvements should be made in RLS.

The normal equations can be solved more efficiently using Cholesky's decomposition (ref. 4, appendix A) rather than the general purpose algorithm in DECOMP; DECOMP was chosen because it was available as an AFWL library program. The sequential regression algorithm is easy to program but slower than DECOMP. Our conclusion is that pole extraction using RLS can be done on a minicomputer. But a lot of work remains in deciding how the technique should be programmed efficiently.

Another objective has been to evaluate the usefulness of pole extraction methods for actual EMP test analysis. The present work has to be regarded as only a preliminary study of this problem. We have shown that RLS gives good results when the data actually consist of damped sinusoids plus noise. Problems encountered with EMP test data such as pulser delay and the possibility of non-exponential components have not been studied.

The author benefited from reading a preliminary draft of the users manual for the SEMPEX computer code (ref. 17). The general approach used with SEMPEX is similar to that described here, but the calculations are done differently.

## REFERENCES

1. Baum, C. E., "On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems," Interaction Note 88, 11 December 1971.
2. McDonough, R. N. and W. H. Huggins, "Best Least-Squares Representation of Signals by Exponentials," IEEE Trans. on Automatic Control, Vol. AC-13, No. 4, August 1968.
3. McBride, L. E., Jr., H. W. Schaeffgen and K. Steiglitz, "Time-Domain Approximation by Iterative Methods," IEEE Trans. on Circuit Theory, Vol. CT-13, No. 4, December 1966.
4. Bard, Y., Nonlinear Parameter Estimation, Academic Press, 1974.
5. Fischl, R., "Optimal  $l_p$ -Approximation of Prescribed Impulse Response Function on A Finite Point Set," Proc. IEEE Int. Symp. Circuit Theory, 1970.
6. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw Hill, 1956.
7. VanBlaricum, M. L. and R. Mitra, "A Technique for Extracting the Poles and Residues of a System Directly from Its Transient Response," Interaction Notes, Note 245, February 1975.
8. Astrom, K. J. and P. Eykhoff, "System Identification - A Survey," Automatica, Vol. 7, 1971.
9. Eykhoff, P., System Identification, Wiley, 1974.
10. Graupe, D., Identification of Systems, Van Nostrand Reinhold, 1973.
11. Astrom, K. J. and T. Bohlin, "Numerical Identification of Linear Dynamic Systems from Normal Operating Records," Theory of Self-Adaptive Control Systems, Plenum, 1966.
12. Hsia, T. C., "On Least Squares Algorithms for Systems Parameter Identification," IEEE Trans. Auto. Control, February 1976.
13. Makhoul, J., "Linear Prediction: A Tutorial Review," Proc. of the IEEE Trans. Auto. Control, February 1976.
14. Jenkins, G. and D. Watts, Spectral Analysis and Its Applications, Holden-Day, 1968.
15. S. Stearns, Digital Signal Analysis, Hayden, 1975.
16. Siege 1.2 Data Acquisition Support, final report, EG&G, Inc., Albuquerque Division Technical Support Group, 24 October 1969.
17. Lager, D. L. and H. G. Hudson, Users Manual for SEMPEX - A Computer Code for Extracting A Sum of Complex Exponentials from a Time Waveform, Mathematics Note 45, March 1977.