NASA CR-132713

File with N75-29066

ATS-32367

REFINEMENT AND VALIDATION OF TWO DIGITAL MICROWAVE LANDING SYSTEM (MLS) THEORETICAL MODELS

BY

WILLIAM G. DUFF

AND

CHARLES R. GUARINO

Prepared Under Contract No. NAS 1-13683

FOR

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

AUGUST 15, 1975

NASA CR-132713

REFINEMENT AND VALIDATION OF TWO DIGITAL MICROWAVE LANDING SYSTEM (MLS) THEORETICAL MODELS

by

William G. Duff
and
Charles R. Guarino

Prepared under Contract No. NAS 1-13683 by

Atlantic Research Corporation
EMM Department
5390 Cherokee Avenue
Alexandria, Virginia 22314

for

National Aeronautics and Space Administration

August 15, 1975

FOREWORD

This report was prepared for the National Aeronautics and Space Administration, Langley Research Center, by Atlantic Research Corporation under Contract Number NAS1-13683.

The project was performed under the technical direction of William G. Duff at Atlantic Research, and Richard M. Hueschen of NASA Langley. Other contractor personnel who contributed directly to this project include:

Charles R. Guarino David J. Hyduke	Atlantic Research Corporation

Other personnel who participated in this project include:

Thomas M. Walsh	NASA Langley
Nesim Halyo	University of Virginia

ABSTRACT

This report describes the results of an effort to refine and validate two digital microwave landing system theoretical models developed under NASA Contract No. NAS1-1192. The MLS models being considered are generic models for the Doppler and scanning-beam frequency reference versions of the MLS. These models represent errors resulting from both system noise and discrete multipath. The data used for the validation effort were obtained from the Texas Instrument conventional scanning beam and the Hazeltine Doppler feasibility hardware versions of the MLS.

Table of Contents

List of Illustrations

List of Tables

List of Symbols

$X(t)$	Time domain representation of MLS errors
$X_T(t)$	Function $X(t)$ limited to values $-T \leq t \leq T$
$X_T(j\omega)$	Fourier transform of $X_T(t)$
$S(\omega)$	Spectral density of $X(t)$
$U_{T/2}$	Spectral Window
a's & b's	Coefficients resulting from curve fitting
$H(j\omega)$	Transfer Function resulting from curve fitting
$\phi(\tau)$	Autocorrelation function
$\phi(m)$	Discrete form of Autocorrelation function
Γ	Standard deviation of MLS errors
α	Reciprocal of correlation time

1.0 Summary and Suggested Studies

A program is in progress to develop an advanced universal Microwave Landing System (MLS) which is intended to eventually replace the present-day Instrument Landing System. The MLS consists of azimuth and elevation antenna systems, and Distance Measurement Equipment (DME) at the runway and corresponding airborne receiving equipment to provide position and velocity navigation data for terminal area flight operations. Insofar as Conventional Take-off and Landing (CTOL) operations are concerned preliminary design and system integration have been carried out by RTCA[7]. This preliminary design included siting arrangements of the ground equipment, volume of coverage and format of the signals, and accuracy standards for the combined ground-based and airborne equipment. Although an effort has been made to specify the maximum range and angle error, the statistical characteristics of the MLS errors are basically unknown.

Computer simulation models that represent both the system noise and discrete multipath MLS errors were developed and implemented under NASA Contract NAS1-11992. Generic models were developed for frequency-reference scanning-beam and ~~Doppler~~ systems. These models were developed from theoretical considerations concerning the communication channel.

The generic model developed for system noise was represented in terms of an autocorrelation function which was given in discrete form by

$$\phi(m) = \Gamma^2 \; e^{-\alpha T|m|}$$

For the purpose of the modeling effort under NAS1-11992, Γ was taken as the standard deviation of the elevation or azimuth error as given by the RTCA specifications for the MLS, and the correlation time (α^{-1}) was varied to determine the effect on aircraft performance.

The model for discrete multipath errors resulted from applying Green's Theorem to determine the field arising from an electromagnetic wave which is reflected and scattered from an object such as a hangar or aircraft. The resulting discrete multipath errors are a function of the geometry of the multipath configuration as presented in Section 3.7.

The objective of the present contract was to validate and refine these models by comparing and correlating the computer generated MLS signals with MLS signals measured from flight data. The data used in the study were obtained on the Texas Instrument conventional scanning-beam and the Hazeltine Doppler systems. Figure 1 illustrates the test site locations for the Hazeltine and Texas Instrument microwave landing systems at Wallops Island. The test flight path from which the data was obtained was for a straight in approach with a 3° glideslope.

For the purpose of this validation effort, MLS errors were considered to consist of system noise, which could be considered as a stationary random process, and discrete multipath errors, which result in a nonstationary random process. The experimental data for the system noise were used to obtain parameter

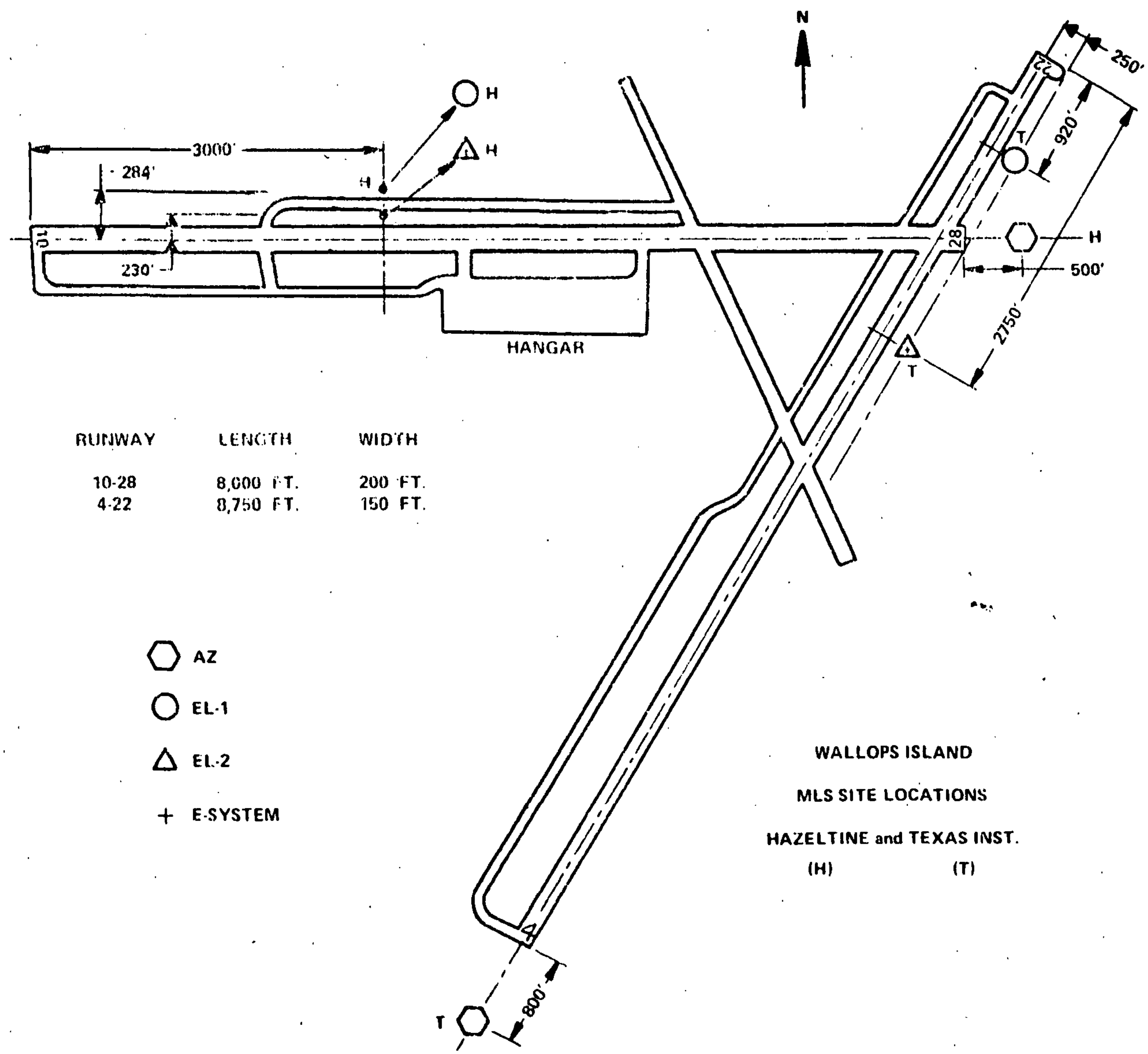

Figure 1. Wallops Station MLS Test Site Locations.

estimates and these were compared with the theoretical model. The best fit models obtained from analyzing the experimental data resulted in the same form for the noise error model as that obtained from theoretical considerations, and this is considered to provide sufficient validation for the form of the generic error model which is

$$\phi(m) = \Gamma^2 e^{-\alpha T|m|}$$

Minor adjustments were required in the values for Γ in order to provide agreement between the theoretical models and those resulting from analyzing the data. In addition, the data analysis resulted in values for α which were not previously available. These results are discussed in Section 3.6. However, it should be cautioned that the results obtained on this effort were limited by the data that was available at the time. As more data is made available, the techniques presented in this report should be used to further refine the models.

In the case of discrete multipath, data were not available on the Hazeltine or Texas Instrument types to obtain a time history of the error characteristics. The only time history plots which could be located for discrete multipath errors were compared with the theory and similar results were obtained. These comparisons are discussed in Section 3.7. Time history plots were used to evaluate the discrete multipath error models and favorable comparisons were obtained.

As a result of the work performed on this contract and other recent developments in the MLS program, it is recommended that the following additional study efforts be performed:

- Mathematical models should be developed and validated for the time-reference-scanning beam system which was selected by the MLS committee.

- The developed models should be tested with conventional aircraft landing systems to determined the sensitivity of the autopilot to correlated noise. In addition, any proposed autopilots developed to take advantage of the MLS accuracy should also be tested.

- Optimal onboard detectors can now be developed based upon the spectrum of the noise statistics. Previously, designers could only make assumptions about the characteristics of the noise. However, since more information is now available, realistic designs are possible.

- Error statistics, generated from additional airports, should be subjected to the same analysis. This will determine how sensitive the parameters of the error model are to site variables. At this point, all the software necessary to perform such an analysis has been developed and additional studies can be made quite easily.

2.0 Approach

The MLS system noise can justifiably be considered a stationary random process, and can best be analyzed in a statistical fashion. Ultimately, an autoregressive moving-average model is desired. It has been shown that a model in that form can represent a random process using less components than either an autoregressive or moving-average process. However, estimating the parameters of such a model is complicated. The major effort expended in this contract was directed toward arriving at a best estimate for these parameters based upon the experimental data and comparing them with the assumed parameters of the implemented model. The parameters were thereby verified by using the experimental data.

In the case of a nonstationary random process, as in discrete multipath reflections, a statistical verification of the implemented models via autoregressive moving-average models is ill-advised. The time histories of the error can be compared, however, to determine if the implemented model is reasonable.

The procedure that was followed is outlined below:

(1) Develop the software to read the data tapes supplied by Texas Instrument and Hazeltine.

(2) Plot the time histories of the errors in range, azimuth and elevation to determine any obvious problems with the data.

(3) Develop the software to find the autocorrelation function and smoothed power spectrum. Plot the results for various values of lag, and determine the optimal lag necessary for analysis.

(4) Develop the software to curve-fit the smoothed power spectrum data with a meromorphic function.

(5) Determine the best model for the power spectrum and thus specify $|H(j\omega)|^2$.

(6) From the estimated $|H(j\omega)|^2$ determine the difference equation to model the system noise.

The remaining sections of this report will elaborate upon each of the six areas and present the results of the analysis.

3.0 Discussion

This section will elaborate upon each of the six areas and present the results of the analysis.

3.1 Tape Read Software

The first major effort performed on the contract was directed toward developing computer programs to read the merged magnetic tapes which contained measured data on the Hazeltine and Texas Instruments feasibility hardware versions of the MLS and radar tracking data. These programs as well as the statistical programs were developed on the LRC Kronas Operation System using standard system subroutines. The tape read programs read the tapes and converted the BCD data to the correct floating point value for printing and plotting. The tape read programs are listed in Appendices A and B.

3.2 Time History Plots

Time histories of the measured errors associated with both the Hazeltine and Texas Instruments MLS were plotted and examined to determine whether there were any obvious problems with the data. Resulting plots of these time histories for the Hazeltine and Texas Instrument systems are shown in Figures 2 and 3, respectively. Referring to Figure 2, it is apparent that for the Hazeltine system, both the elevation and azimuth angle data have relatively large spikes which could indicate a system malfunction or other large error sources in the environment. The elevation error spikes are particularly significant because of their large amplitudes. It was felt that the large error spikes should not be attributed to system noise; and thus, it was considered appropriate to remove them (by limiting the maximum error) before the data were analyzed. The altered time history for the elevation data is shown in Figure 4. Although the spikes shown in Figure 4 are considerably smaller than those shown in Figure 2, they are still present and could affect the analysis results.

3.3 Computation of Power-Spectral Density

The next step is to compute the power spectral density. The spectral density of the function X(t), where X(t) is the MLS data, may be defined in the following manner.

Let $X_T(t) = X(t)$ for $-T \leq t \leq T$

and $X_T(t) = 0$ for all other values of t

Let $X_T(j\omega)$ = the Fourier transform of the function $X_T(t)$, which is defined as follows.

$$X_T(j\omega) = \int_{-\infty}^{\infty} X_T(t)e^{-j\omega t}dt \tag{1}$$

$$= \int_{-T}^{T} X(t)e^{-j\omega t}dt \tag{2}$$

The spectral density $S(\omega)$ of the function X(t) is:

$$S(\omega) = \lim_{T\to\infty} \frac{1}{2T} \left| X_T(j\omega) \right|^2 \tag{3}$$

If the signal X(t) is not considered a deterministic function, but merely one member of the ensemble which comprises a random process, the concept of power spectra must be employed to provide a harmonic representation of the function. Strictly speaking, such a representation is valid only when the random process may be said to be both stationary and ergodic. Briefly, this means that the statistics of the process are independent of time (no change in the mechanism of generation is present) and each sample function is representative of the whole ensemble.

It should be noted that if the random process is not stationary, a power spectral representation is invalid and, in fact, no general representation of reasonable utility exists. If only the ergodicity condition is violated, the power spectrum representation can be used.

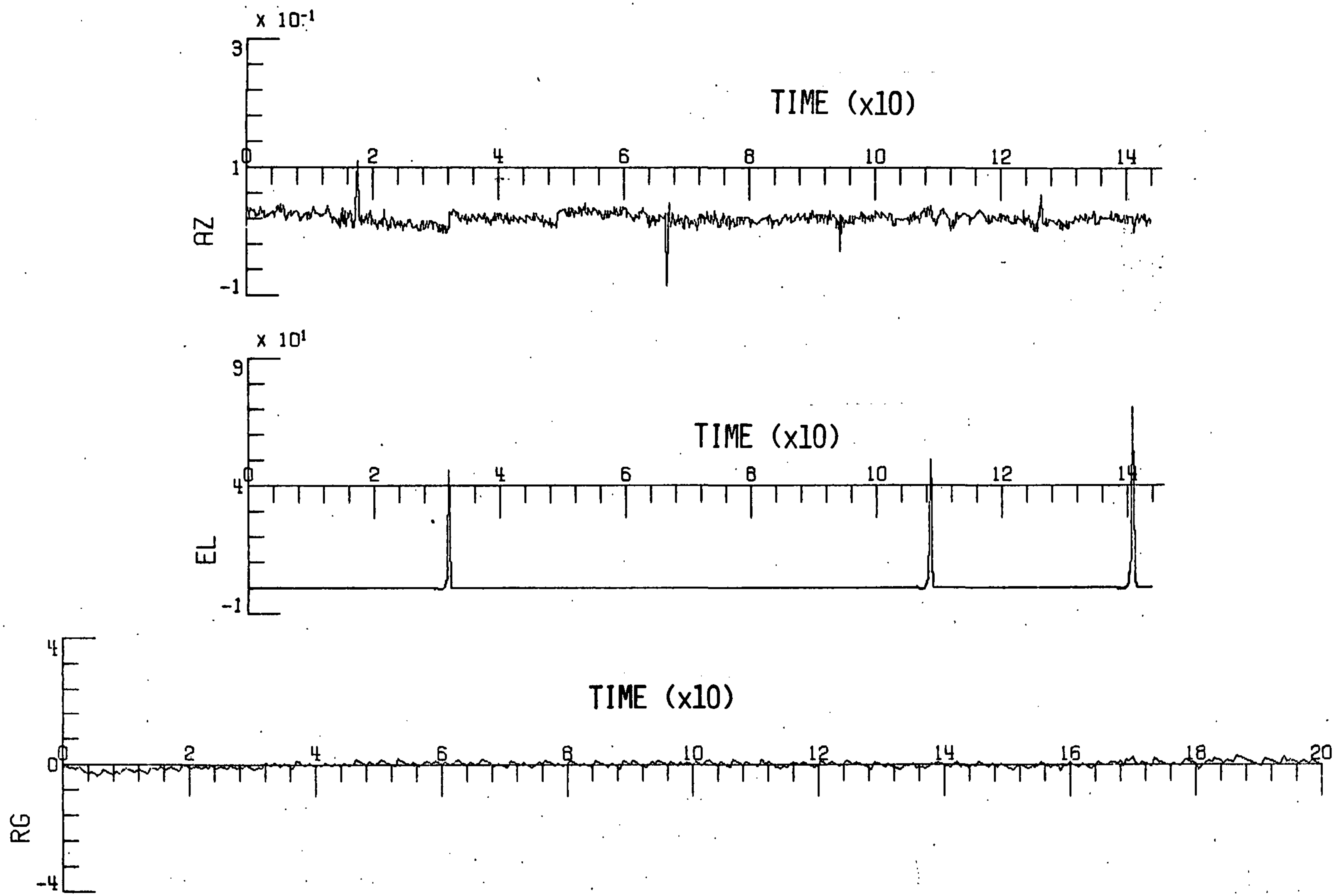

Figure 2. Time History of Errors for Hazeltine Data.

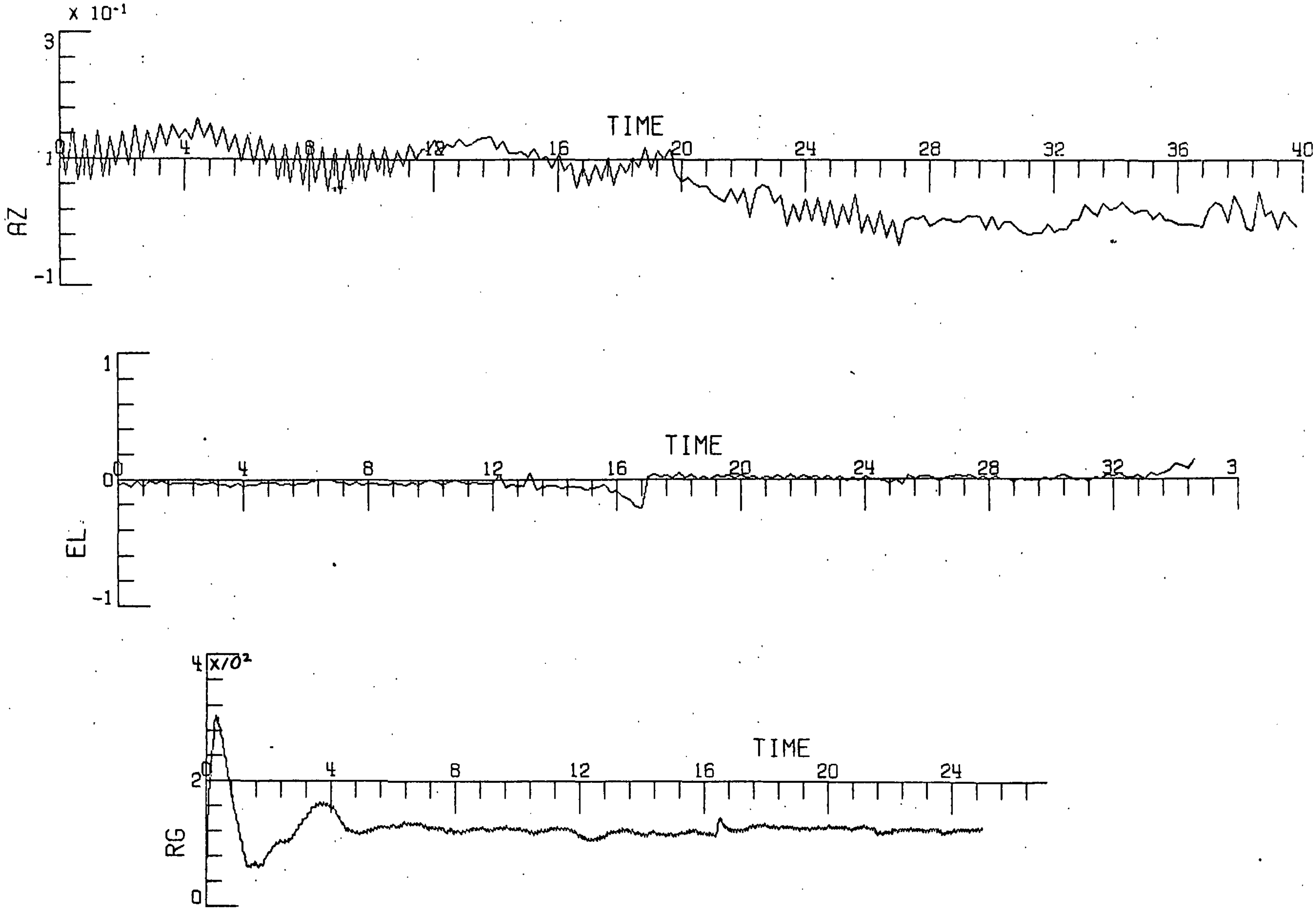

Figure 3. Time History of Errors for Texas Instrument Data.

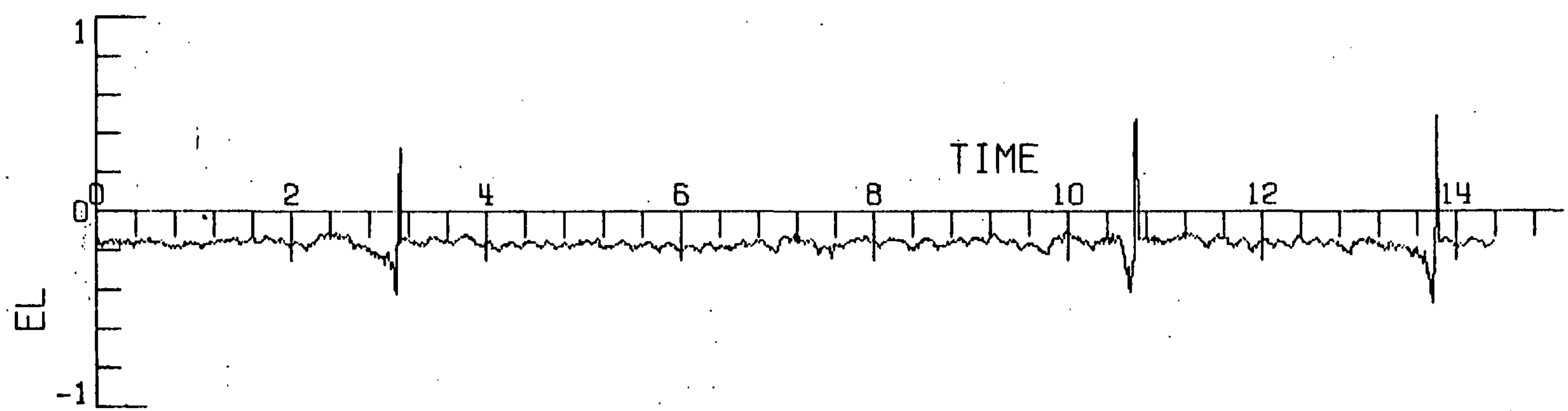

Figure 4. Corrected Time Plot of Hazeltine Data.

A basic limitation of spectral estimation is that data input must always be finite in length. This causes a frequency smearing, or lack of resolution. This can be shown by limiting the data signal, X(t), by using a spectrum window function, $U_{T/2}$, as follows:

$$X_T(j\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U_{T/2}(t)\ X(t)e^{-j\omega t}dt \tag{4}$$

where $U_{T/2}$ can be defined in several ways; the important point being that

$$\begin{aligned} U_{T/2} &= 0 \text{ for } T > T_{MAX} \;\&\; T < -T_{MAX} \\ &= \text{the spectral window otherwise} \end{aligned} \tag{5}$$

From the Fourier frequency convolution theorem, equation (4) may be rewritten as

$$X_T(\omega) = \int_{-\infty}^{\infty} F(\theta)\ U_{T/2}\ (\omega - \theta)\ d\theta \tag{6}$$

where F(θ) is the true transform and the transform of $U_{T/2}(t)$ is given by

$$U_{T/2}(\omega) = \frac{T}{2\pi} \cdot \frac{\sin(\omega T/2)}{\omega T/2}$$

if $U_{T/2}(t)$ is the boxcar data window. Equation (6) is simply the convolution of two functions in the frequency domain which is equivalent to multiplication of the transformed functions in the time domain (equation 5).

From this it can be seen that $X_T(\omega)$ is the weighted average of the values of F(ω) about ω=θ. Also $X_T(\omega)$ is an estimate of X(ω), the true Fourier transform of X(t). Because of the duality of time domain, multiplication and frequency domain convolution, the finite transform $X_T(\omega)$ at ω=θ is an infinite sum of contributions selected from X(ω) by $U_{T/2}(\omega)$. The magnitude of these contributions is dependent upon the lobes of $U_{T/2}$ on either side of the maximum.

It is thus desirable to minimize the size of the sidelobes of $U_{T/2}$ in order that $X_T(\omega)$ may approximate $X(\omega)$. For larger data samples, $(T \to \infty)$, the approximation improves. In Figure 5, the spectrum windows used in this analysis are shown. The program used to calculate the spectrum generates estimates based upon both windows. All analyses that follow, however, are based upon the smooth spectrum using a Hanning window. The program used to calculate the spectrum estimate is called ASA and is available in the NASA Langley Scientific Library.

Various correlation times or lags in the autocorrelation function were evaluated to determine the number of lags to use in generating the spectral density of MLS error. Figure 6 shows the smoothed spectrum for the elevation data, and Figure 7 the azimuth data, from the Hazeltine tapes. From this it can be seen that choosing too small a lag tends to smooth out many of the variations. For this reason it was decided to use 100 lags for all data. As can be seen with the range and angle data, very little is gained in increasing the lags above 100. At 150 lags the range power spectrum is only slightly more detailed than at 100 lags.

Figures 8 and 9 shows the power spectrum for the Texas Instrument elevation and azimuth data. In this case, increasing the number of lags also has little effect upon the spectrum. It should be emphasized that the Texas Instrument tape had far fewer data points than the Hazeltine tapes. The confidence that one can have in the estimated spectrum decreases as the number of data points decreases. It is not wise, therefore, to attempt to achieve great accuracy with large lags because one is greatly limited by the small amount of data available.

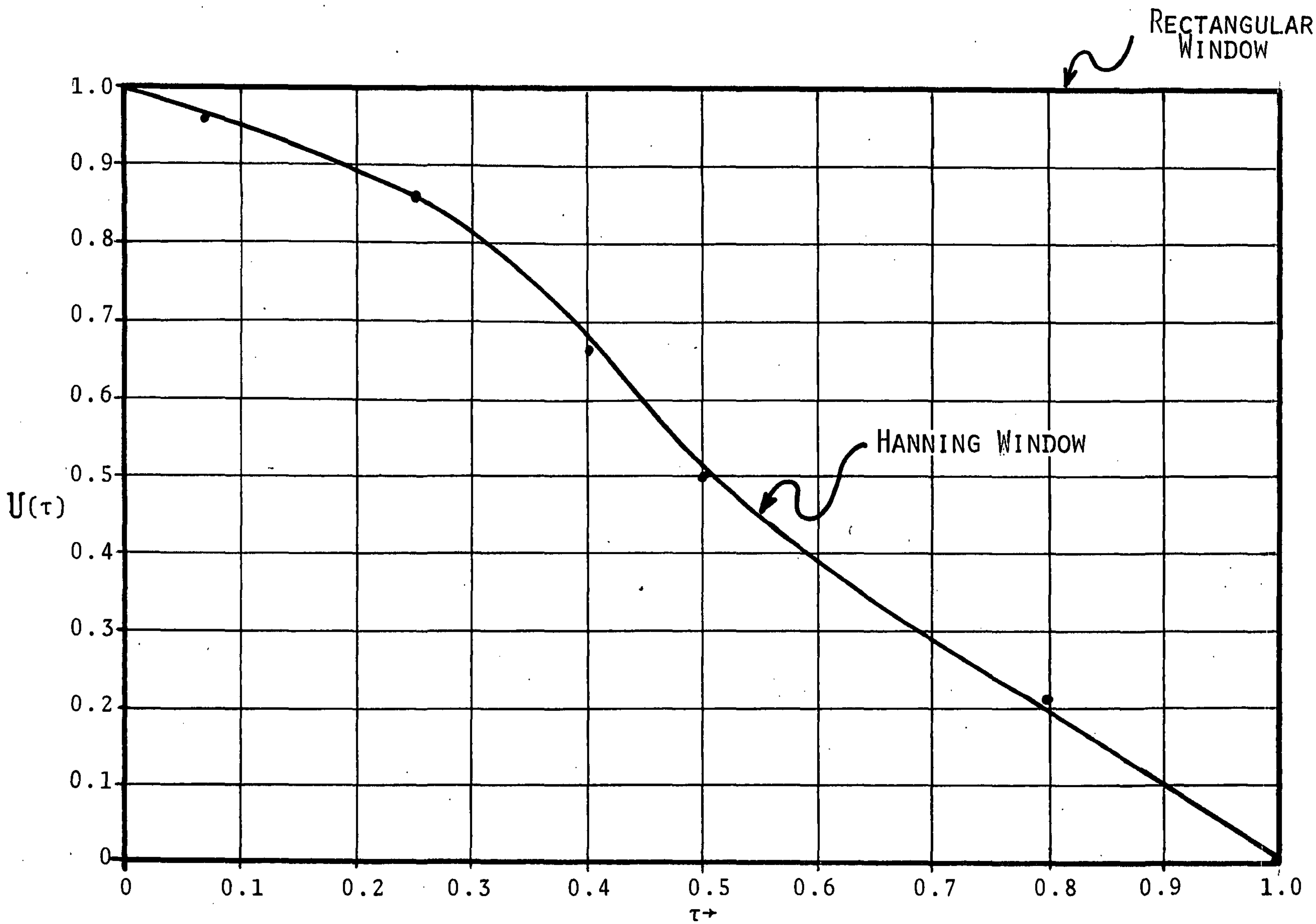

Figure 5. Spectrum Windows Used in Analysis.

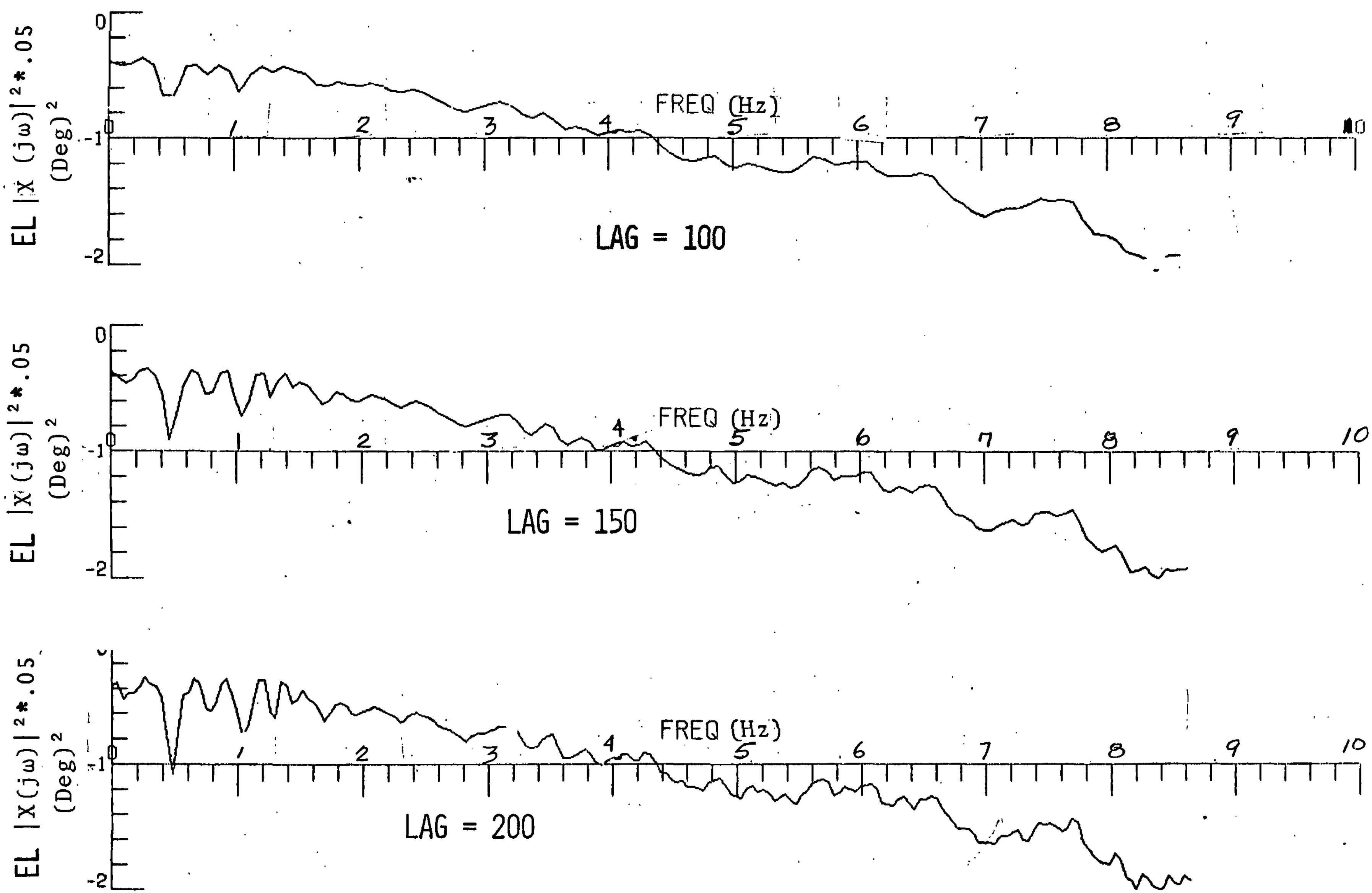

Figure 6. Power Spectrum Density Hazeltine Elevation Error Lag Parameter Plots.

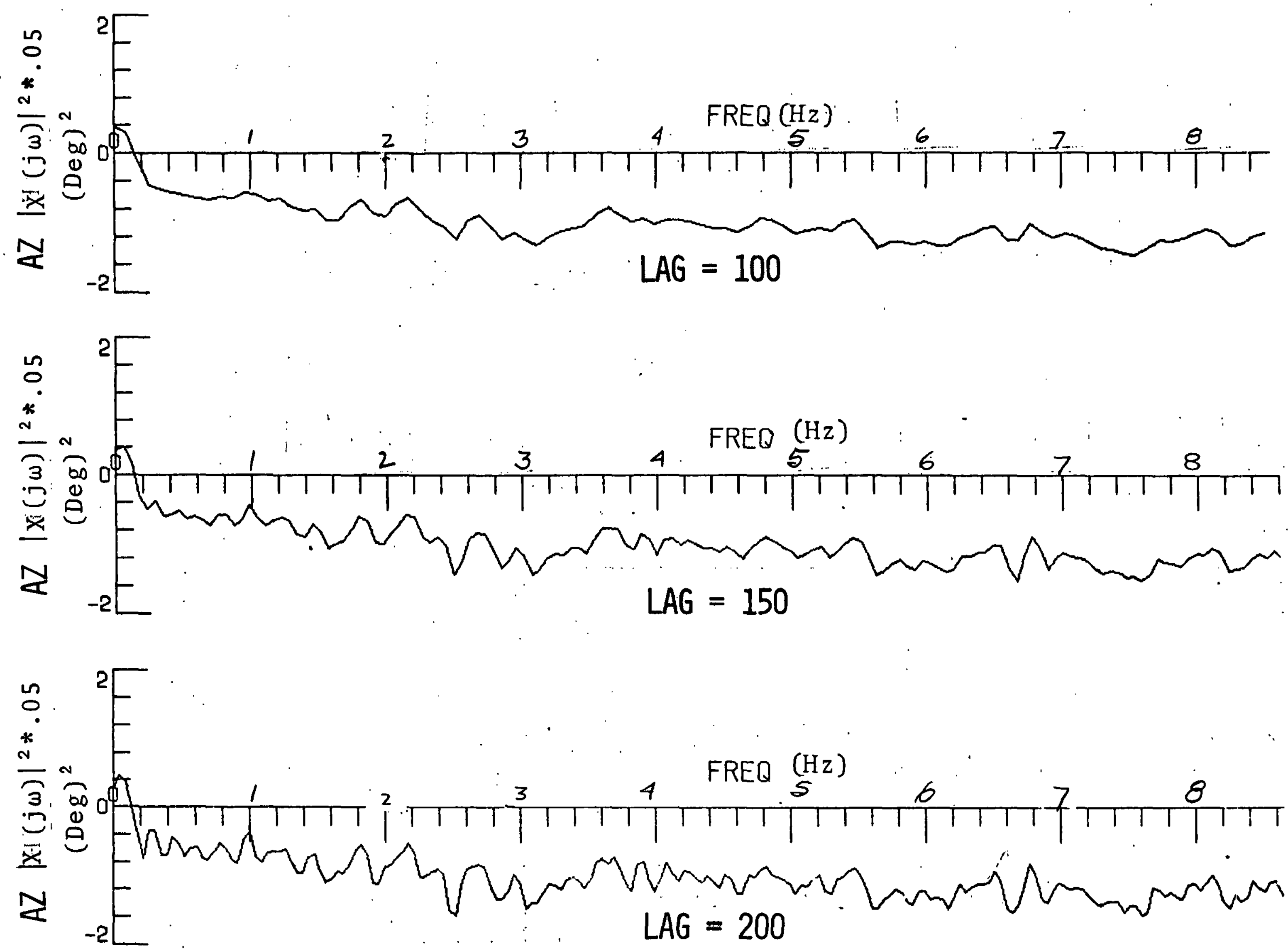

Figure 7. Power Spectrum Density, Hazeltine Azimuth Error Lag Parameter Plots.

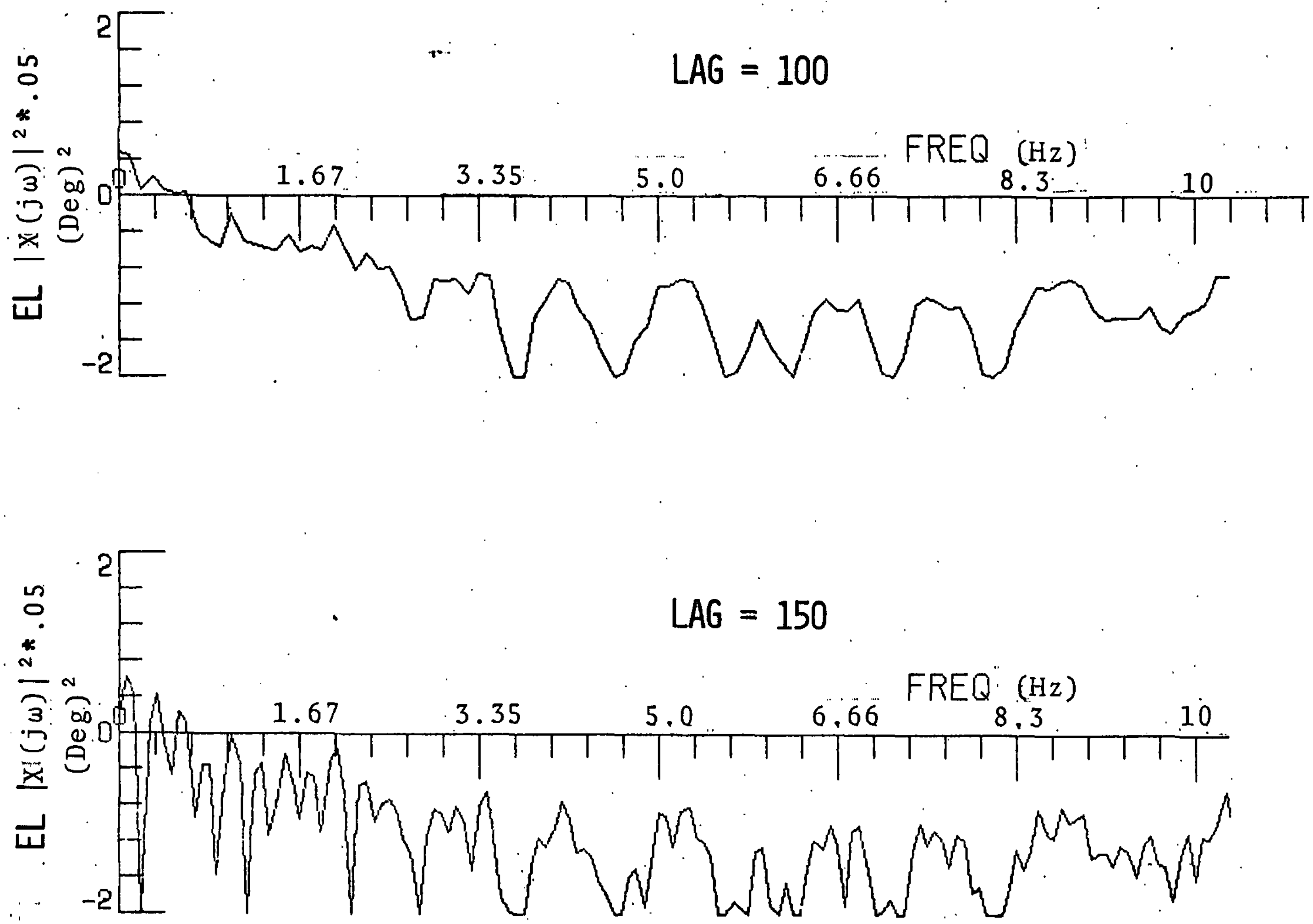

Figure 8. Power Spectrum Desnity Texas Instrument Elevation Error Lag Parameter Plots.

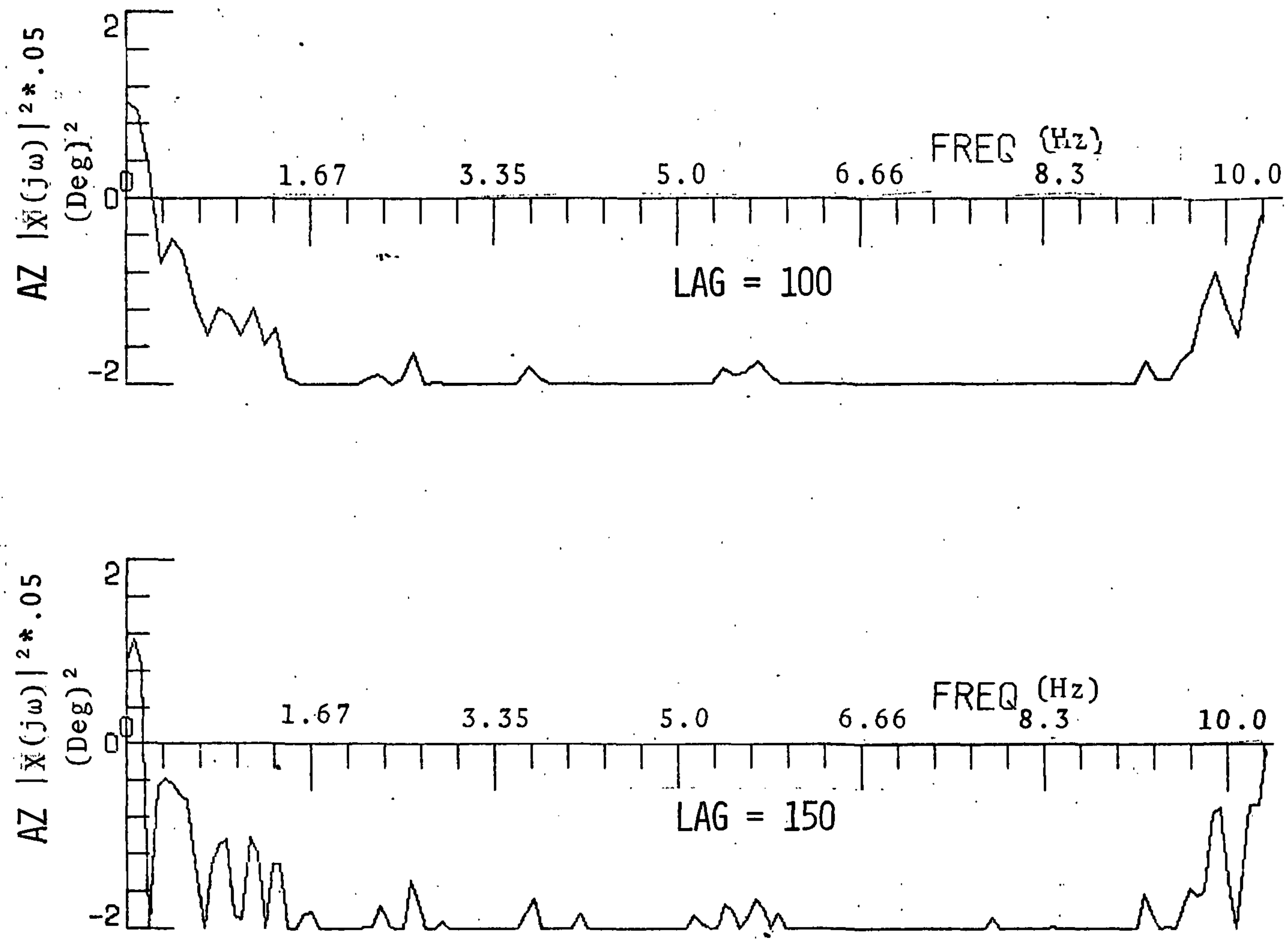

Figure 9. Power Spectrum Density Texas Instrument Azimuth Error Lag Parameter Plots.

3.4 Smoothed Power Spectra

The next step in the analysis is to approximate the spectral density function by meromorphic functions. There are two major reasons for using meromorphic functions.

(1) The mean-squared error can be calculated if the spectral density of the error is given in the form of a meromorphic function of ω.

(2) Many methods exist for determining a transfer function which yields a minimum mean-squared error; these methods also are based on the assumption that the spectral density is a meromorphic function of ω.

Therefore, for the application of both of the above mentioned methods, it is necessary first to approximate the spectral density curves by meromorphic functions. In addition to the above applications, the approximation of a given curve by a meromorphic function is of value in solving many problems involved in the synthesis of linear dynamic systems. Therefore, the following methods that are presented have wide application. Since this aspect of the analysis is in some respects the most difficult and at the same time the most important, a brief theoretical background discussion is presented.

Let us assume that we have a given curve $F(\omega)$ and that we wish to approximate it by a rational fractional function $S(\omega)$ in the form

$$S(\omega) = \frac{b_0 + b_1\omega^2 t \cdots + b_m\omega^{2m}}{a_0 + a_1\omega^2 t \cdots + a_n\omega^{2n}} \tag{7}$$

We can write

$$0 = (b_0 + b_1\omega^2 \cdots + b_m\omega^{2m}) - S(\omega)(a_0 + a_1\omega^2 + \cdots + a_n\omega^{2n}) \tag{8}$$

By selecting a number of arbitrary points on the given curve for (n+m+2) particular values of ω and determining the ordinates of the curve, i.e., the values of the function $S(\omega)$ at these points, we obtain a system of (n+m+2) simultaneous linear equations for n+m+2 unknown coefficients b_0, $b_1 \cdots b_m$; a , a $+ \cdots a_n$ having the form

$$0 = \left(b_0 + b_1\omega_1{}^2 + \cdots + b_m\omega_1{}^{2m}\right) - S(\omega_1)\left(a_0 + a_1\omega_1{}^2 + \cdots + a_n\omega_1{}^{2n}\right) \quad (9)$$

$$0 = \left(b_0 + b_1\omega^2_{n+m+2} + \cdots + b_m\omega^{2m}_{n+m+2}\right) - S(\omega_{n+m+2})\left(a_0 + a_1\omega^2_{n+m+2} + \cdots + a_n\omega^{2m}_{n+m+2}\right) \quad (10)$$

The solution of these equations will provide us with the numerical values of the coefficients b_0, b_m, b_m, a_0, a_n.

Besides being awkward, this method has the further disadvantage that even if the function $S(\omega)$ is continuous and the number of points taken is very large, it is still impossible to verify that the error (Δ) in the approximation, i.e.,

$$\Delta = F(\omega) - S(\omega) \quad (11)$$

is of sufficiently low absolute value at points which do not coincide with the selected points ω_1, $\omega_2 \cdots$.

An alternate approach to solving this problem is using nonlinear least squares. The standard method for solving least squares problems which lead to nonlinear normal equations depends upon a reduction of the residuals to linear form by first-order Taylor approximations taken about an initial or trial solution for the parameters. The nonlinear nature of the problem can best be seen if $S(\omega)$ is written in the following form:

$$S(\omega;a,b) = \frac{a_0 + a_1\omega^2 + \cdots + a_{n-1}\omega^{2(n-1)}}{b_0 + b_1\omega^2 + \cdots + b_n\omega^{2n}} \quad (12)$$

where the a's and b's denote the numerator and denominator coefficients, respectively. The least squares solution to this problem by direct minimization of the approximation error as a function of the filter coefficients requires the solution of a set of nonlinear equations.

Several methods [1-3] are available which have decoupled the solution for the numerator and denominator coefficients. First a linear estimate for the b's is calculated and then the corresponding a's are found. However, these linear estimates are suboptimal, since the b's are solutions to an overdetermined set of linear equations that minimize the linear equation error $\varepsilon(b)$. This $\varepsilon(b)$ is a nonlinear mapping of the approximation error $\varepsilon(a,b)$. In general, the solution for b that minimizes $||\varepsilon(b)||$ is only the same as that which minimizes $||\varepsilon(a,b)||$ when $\varepsilon(a,b)=0$. Therefore it is necessary to minimize with respect to both a and b.

A program has been used which solves for a and b simutaneously. The method is based on the Marquardt technique [4] which is similar to the Levenberg procedure [5].

$$\text{Let } \phi(B) = \sum_{i=i}^{m} [W_i \quad Y_i - F(X_i,B)]^2 \tag{13}$$

The usual linearized Gauss procedure [6] applied to the problem of minimizing $\phi(B)$ generates estimates by the following procedure:

Start with an initial estimate B.

Solve.

$$A^T A\delta = A^T r \text{ for } \delta. \tag{14}$$

The new estimate is calculated as $B_1 = B + t\delta$ where $0 < t \leq 1$. The process is then repeated at the new estimate B_1 until some acceptable convergence criterion is met. In the above,

r_i = the ith residual = $W_i^{\frac{1}{2}}[Y_i - F(X_i,B)]$

A = the matrix defined by

A_{ij} = $W^{\frac{1}{2}}$ [partial of $F(X_i,B)$ with respect to bj]

δ = the correction vector, $B_1 = B + t\delta$.

The Marguardt method differs from the above in that $(A^T A + \lambda I)\ \delta = A^T r$ is solved instead of $A^T A = A^T r$. The λ value is called a damping coefficient. In practice, the method is implemented in the procedure given below.

Start each iteration with a B_0 and a λ_0. B_0 is the current estimate of the minimizing parameter vector and λ_0 is an estimate of the "scaled" damping coefficient.

The A^TA and A^Tr quantities are formed at the value B_0. This system is scaled to form $[SA^TAS + \lambda_0 S^2]S^{-1}\delta = SA^Tr$. Thus, the λ_0 above is an estimate of the damping constant for this scaled system. Here S is a diagonal matrix with $S_{ii} = [A^TA_{ii}]^{-1/2}$. The scaled correction factor, $S^{-1}\delta$, is computed and the new correction is $B_1 = B_0 + \underline{S}(S^{-1}\delta)$. If $\Phi(B_1) < \phi(B_0)$ then a new vector is calculated using $\lambda/10$, this is called B_2. The new λ estimate is chosen to be the one corresponding to minimize $[\Phi(B_1), \Phi(B_2)]$. The corresponding B_k;K=1,2 is taken as the next parameter estimate. If $\phi(B_1) > \phi(B_0)$, the new vector, B_2, corresponding to a damping constant of $\lambda/10$ is still calculated. If $\phi(B_2) < \phi(B_0)$, $\lambda/10$ is accepted. If $\phi(B_2) > \phi(B_1)$, then λ is increased by factors of 10 until one of two conditions occurs. If some $\lambda*10^r$ is found that yields an estimate that decreased ϕ, this $\lambda*10^r$ is taken as the damping constant and the new estimate is the B_2 found from $(SA^TAS + \lambda*10^rS^2)\ S^{-1}\delta = SA^Tr$. However, if the angle between the Marguardt direction vector and the negative gradient direction becomes smaller than 45 degrees before such a $\lambda*10^r$ is found, then the search on λ is terminated and the direction taken to be examined is the negative gradient.

The entire procedure is repeated until the maximum number of iterations is exceeded or convergence is achieved. Convergence is accepted when $|\delta_i| \leq r+\varepsilon|B_i|$ for $i=1,n_0$. Here δ is the correction vector to B and $r<<\varepsilon$.

Since the Marquardt method requires many calculations, a linear least squares method was developed and compared to the Marquardt method.

Let

$$S(\omega) = \frac{a_0 + a_1\omega^2 + \cdots + a_n\omega^{2n}}{1 + b_1\omega^2 + \cdots + b_{n+1}\omega^{2(n+1)}} \qquad (15)$$

This can be written as

$$S(\omega) + b_1 S(\omega)\omega^2 + \cdots + S(\omega) b_{n+1}\omega^{2(n+1)} = a_0 + a_1\omega^2 + \cdots + a_n\omega^{2n}$$

$$S(\omega) = S(\omega) b_1\omega^2 - \cdots - S(\omega) b_{n+1}\omega^{2(n+1)} + a_0 + a_1\omega^2 + \cdots + a_n\omega^{2n}$$

Making a change of variables as follows:

$$X_1 = \omega^2$$
$$X_2 = \omega^4$$
$$\vdots$$
$$X_n = \omega^{2n}$$

and

$$Y_1 = -S(\omega)\omega^2$$
$$Y_2 = -S(\omega)\omega^4$$
$$\vdots$$
$$Y_m = -S(\omega)\omega^{2m}$$

Results in

$$S(\omega) = b_1 Y_1 + b_2 Y_2 + \cdots + b_n Y_n + a_0 + a_1 X_1 + a_2 X_2 + \cdots. \quad (16)$$

The transformed equation is now linear in its coefficient which implies that the "a's" and "b's" can be estimated by linear regression. The nonlinear correlation between ω and $S(\omega)$ will not affect the efficiency of estimation. In this form, the estimation problem has been greatly simplified. A computer study was made to verify the performance of the linear regression as compared to the nonlinear regression. In general, the linear regression tended to give estimates which weighted all data points equally; whereas, the nonlinear program tended to weigh more heavily those data points of high magnitude. It is felt that additional work should be done with the linear regression because theoretically it represents a great simplification of the approximation problem. Indeed, it is possible to estimate the coefficients on-line so that one could get a fast estimate of how quickly the error statistics are changing. This method could also be used in designing digital filters where the impulse response is given and the transform H(Z) is desired which will best approximate the impulse response.

3.5 Best Fit Models

The math models which give the best fit to the spectra, based upon the nonlinear regression program are given below; and are illustrated in Figures 10 through 14.

<u>Hazeltine Data</u>:

Elevation

$$|H(j\omega)|^2 = \frac{B_0}{(B_1 + \omega^2)} \qquad (17)$$

B_0 = 7.024 Standard Error = 0.4288

B_1 = 366.37 Standard Error = 28.02

Azimuth

$$|H(j\omega)|^2 = \frac{B_0}{(B_1 + \omega^2)} \qquad (18)$$

B_0 = 0.1156 Standard Error = 0.01314

B_1 = 0.938 Standard Error = 0.128

Range

$$|H(j\omega)|^2 = \frac{B_0}{(B_1 + \omega^2)} \qquad (19)$$

B_0 = 1.4998 Standard Error = 2.62

B_1 = 1.02389 Standard Error = 0.1092

<u>Texas Instrument</u>:

Azimuth

$$|H(j\omega)|^2 = \frac{B_0}{(B_1 + \omega^2)} \qquad (20)$$

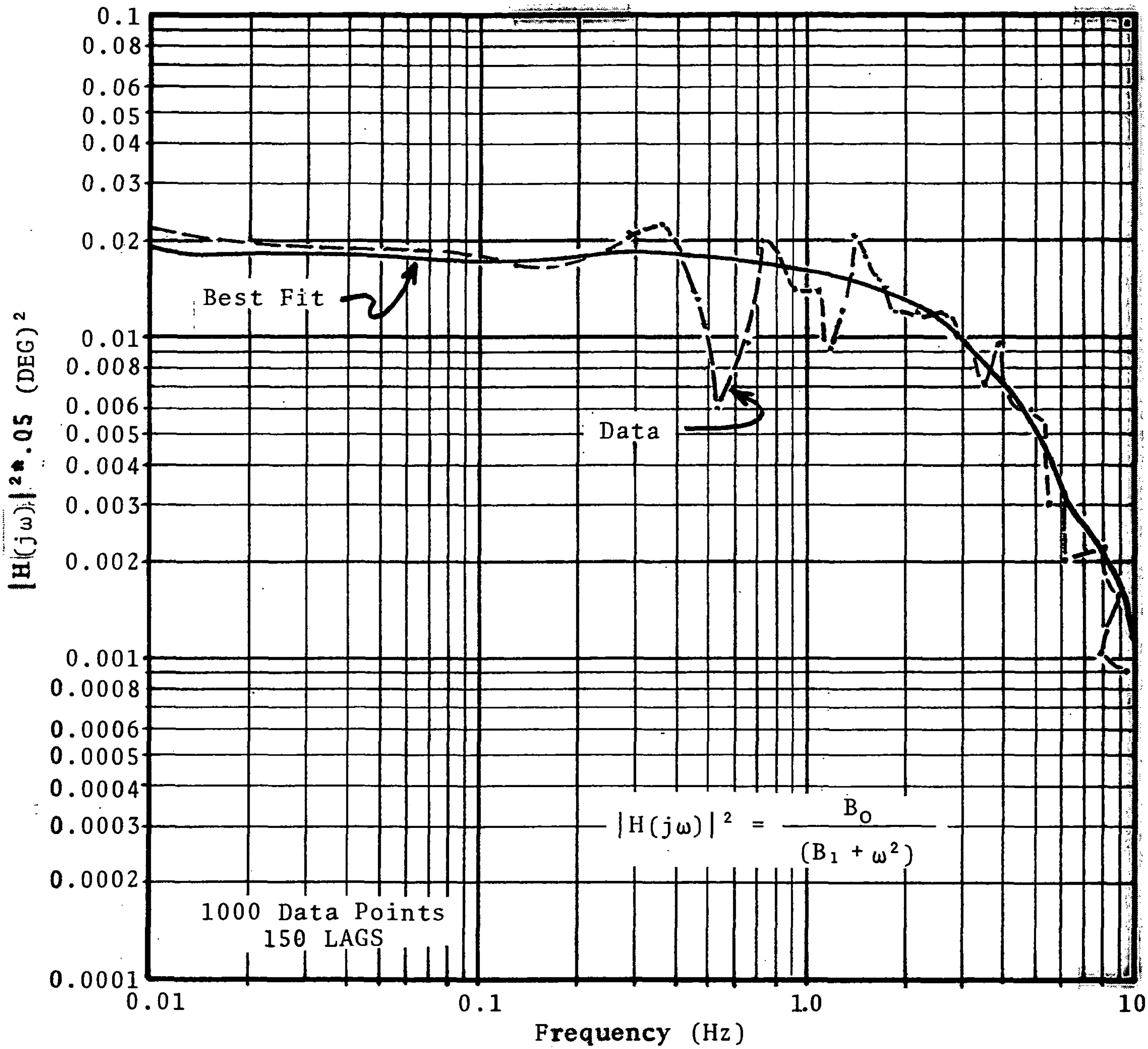

Figure 10. Hazeltine Elevation Data.

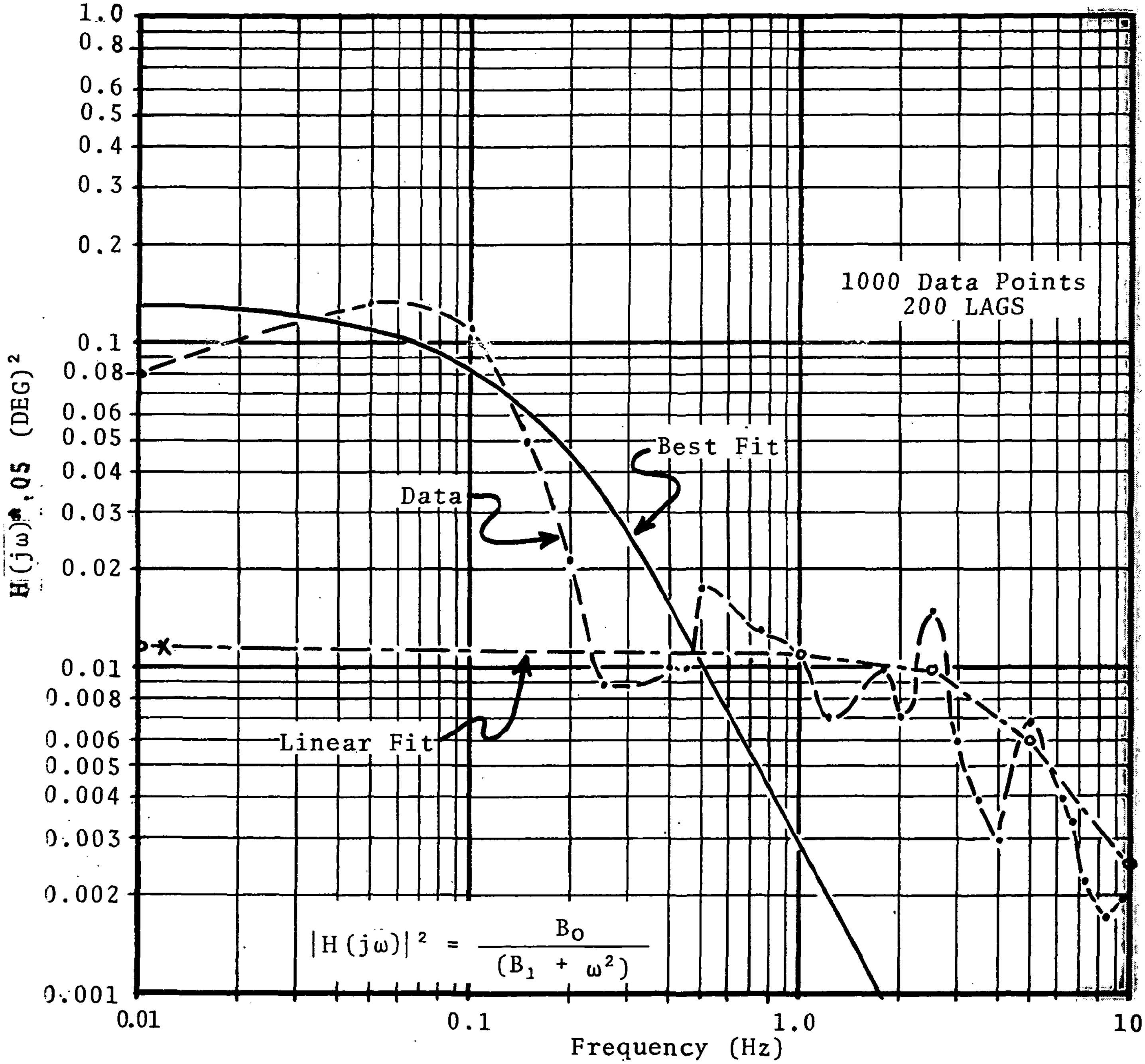

Figure 11. Hazeltine Azimuth Data.

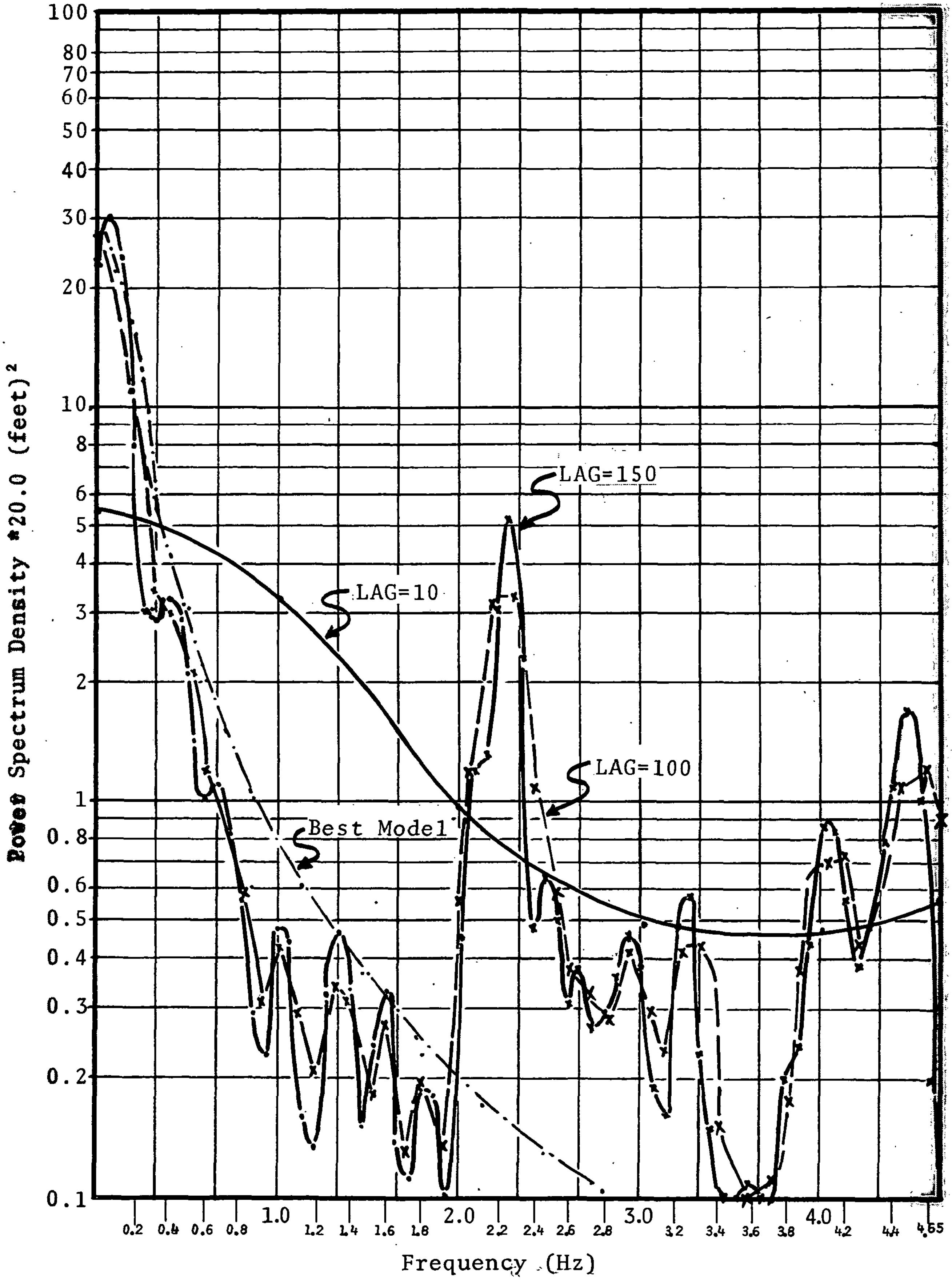

Figure 12. Hazeltine Range, Power Spectrum Density.

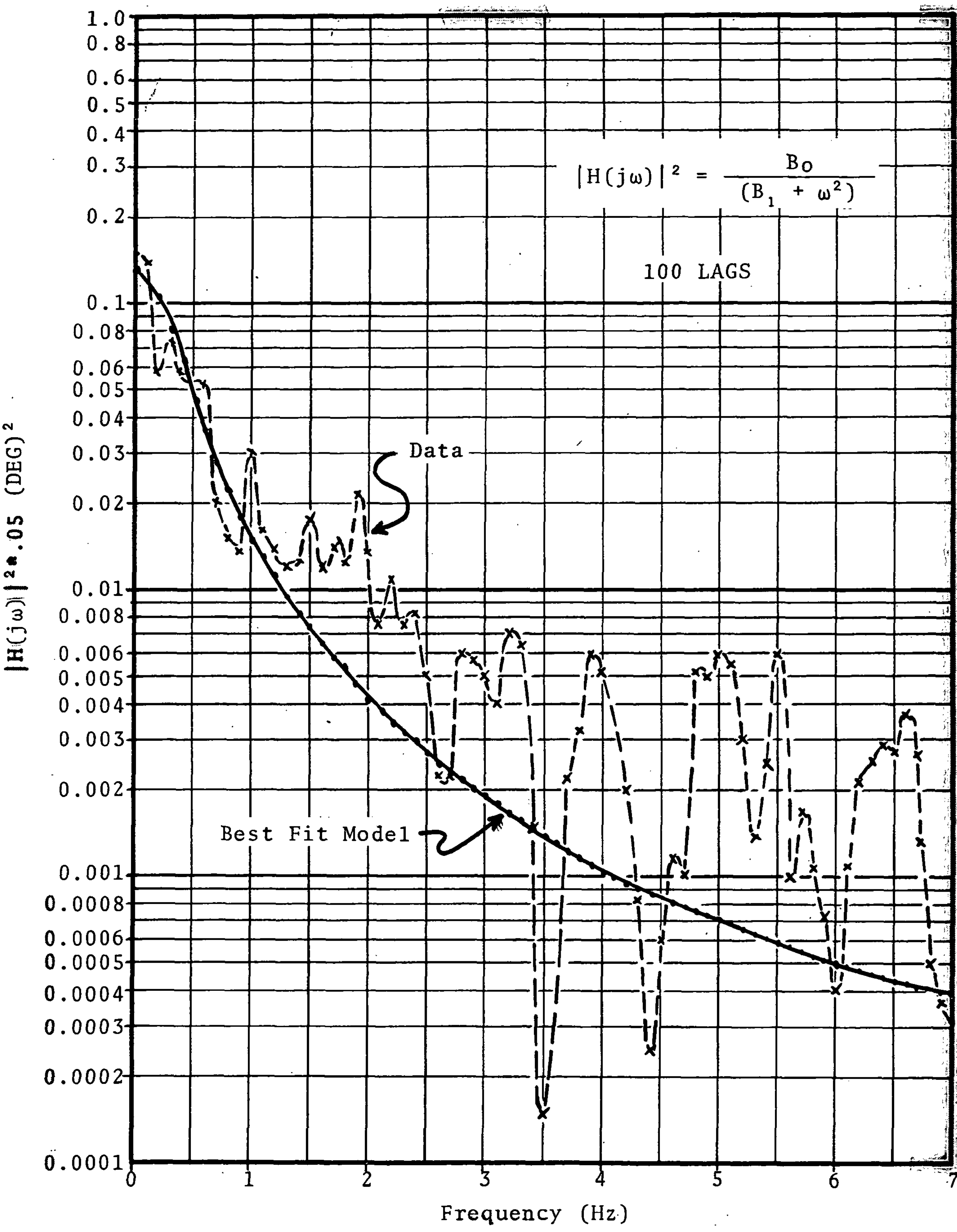

Figure 13. Texas Instrument Elevation Data.

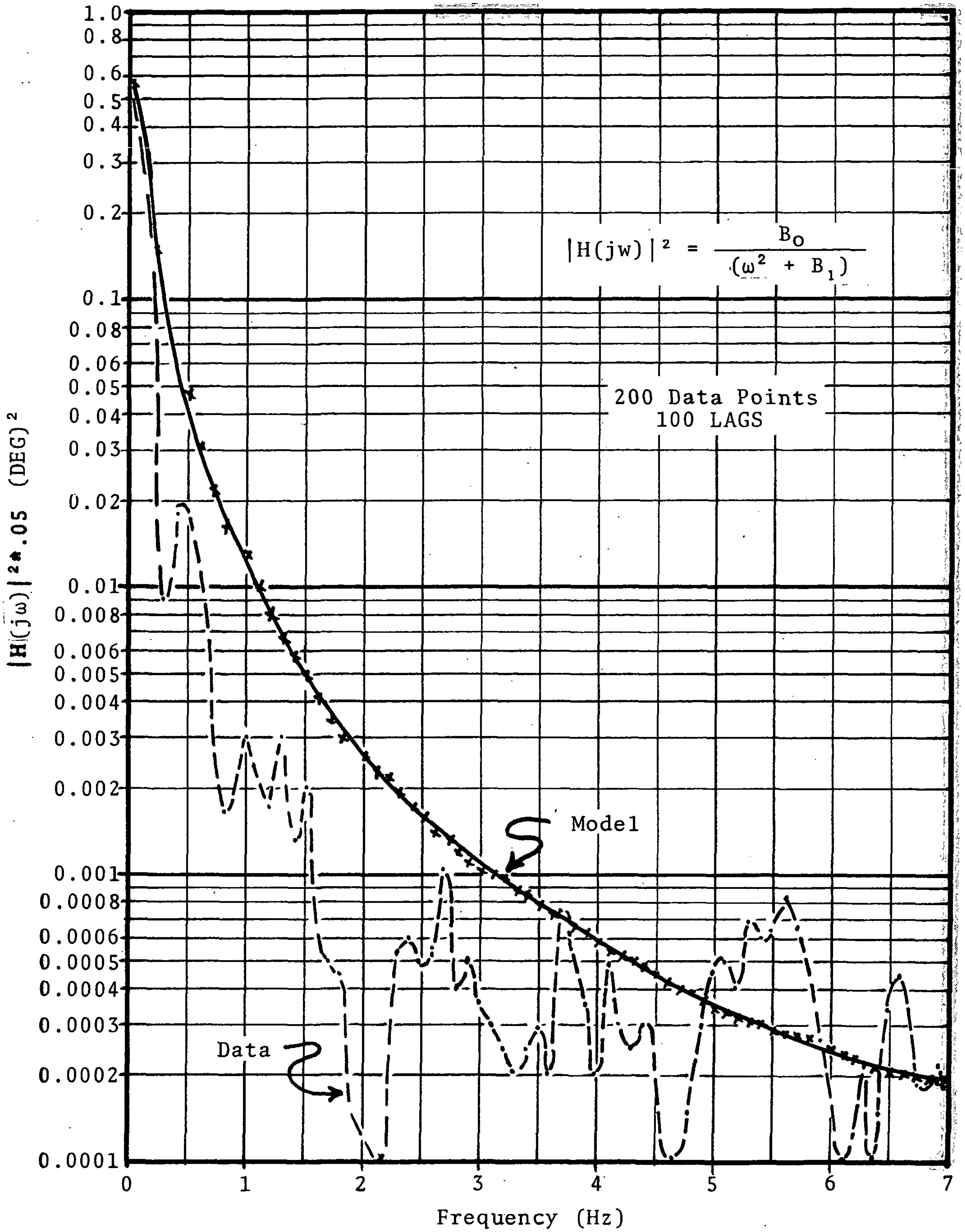

Figure 14. Texas Instrument Azimuth Data.

B_0 = 0.33025 Standard Error = 0.026

B_1 = 0.567 Standard Error = 0.051

Elevation

$$|H(j\omega)|^2 = \frac{B_0}{(B_1+\omega^2)} \tag{21}$$

B_0 = 0.687 Standard Error = 0.0634

B_1 = 5.066 Standard Error = 0.5666

Several additional models were fit to the data but the results in general were poorer. Table 1 shows a listing of the models with the standard errors of the coefficients. With several models, the coefficients did not converge to a final value; this indicates that the hypothesized model is an inadequate representation of the data. It should also be pointed out that the models that did the best for angle data, for both the Hazeltine and Texas Instrument data, are also the simplest with which to work (no zeros and just a single pole). The specification of the difference equation becomes quite straightforward and the results are more easily understood. For instance, it is possible for the angle data to specify the errors by the correlation time, a variable that is derived simply from the "B" coefficients. This simplicity is a very desirable property, one which cannot be too strongly emphasized.

An important fact to observe from the time plots of the Texas Instrument data (Figure 3) is the small time period of the available data. The plot of the azimuth angle error data indicates that its statistics are not stationary. This is also true for the range data where the first four or five seconds of data are obviously nonstationary. Therefore, a large amount of confidence cannot be placed on the power spectral estimates for the Texas Instrument data. As mentioned earlier, power spectral estimates are not valid for a nonstationary process. In addition, it was extremely difficult finding transfer function coefficients that would represent the data and the number of data points available for angle analysis both for azimuth and elevation data was less than 200. This number is far too small for adequate analysis. For these reasons, it was felt that to carry the Texas Instrument

Table 1. Standard Errors of Coefficients.

$$|H(j\omega)|^2 = \frac{B_0(B_1 + \omega^2)}{(B_2 + \omega^2)(B_3 + \omega^2)}$$

	HAZELTINE				TEXAS INSTRUMENT			
	AZIMUTH		ELEVATION		AZIMUTH		ELEVATION	
		Standard Error		Standard Error		Standard Error		Standard Error
B_0 =	0.11040	00.02606	1.3877	2.013	Convergence Not Possible		01.62879	00.2827
B_1 =	1.27560	03.80400	12760.0000	21690.000			03.34141	01.6420
B_2 =	1.05183	56.02000	976.4200	17740.000			34.20000	12.4000
B_3 =	1.05182	56.15000	976.4170	17750.000			01.03198	00.3221

$$|H(j\omega)|^2 = \frac{B_0(B_1 + \omega^2)(B_2 + \omega^2)}{(B_3 + \omega^2)(B_4 + \omega^2)(B_5 + \omega^2)}$$

	HAZELTINE				TEXAS INSTRUMENT			
	AZIMUTH		ELEVATION		AZIMUTH		ELEVATION	
		Standard Error		Standard Error		Standard Error		Standard Error
B_0 =	00.0894619	0.002656	Convergence Not Possible		Convergence Not Possible		1.63194	0.3190
B_1 =	00.2454520	0.051840					3.45148	370.8000
B_2 =	00.1416080	0.122400					3.45025	410.3000
B_3 =	-3.5308500	0.332200					34.34100	19.3200
B_4 =	14.0772000	1.028000					3.54700	168.1000
B_5 =	00.0182540	0.003520					1.03450	0.7432

angle data analysis any further would be misleading. Basically, the data does not lend itself to the type of analysis being performed, and to continue with the analysis would be misleading.

The range data, excluding the first five seconds, are also open to question. Unlike the Hazeltine data, there is clearly a bias on the range error data (compare Figures 2 and 3). An attempt was made to continue the analysis with the first five seconds removed from consideration. Even then, however, it must be pointed out that the results obtained may be misleading.

3.6 Derivation of Difference Equations for Digital Simulation

A detailed derivation of the Hazeltine elevation difference equation is given. Since the derivation for the other models is similar, only the results are presented for them.

Hazeltine Elevation:

$$|H(j\omega)|^2 = \frac{B_0}{(B_1 + \omega^2)} \qquad B_0 = 7.024 \quad B_1 = 366.37 \qquad (22)$$

$$|H(j\omega)|^2 = \frac{\gamma^2 2\alpha}{\alpha^2 + \omega^2} \qquad \alpha = 19.1 \quad \gamma = 0.184 \qquad (23)$$

$\Phi(\tau)$ = Autocorrelation function

$\Phi(\tau)$ = Scale Factor e

The scale factor results from the normalization that the subroutine ASA performs. In general $\Phi(\tau)_{\tau=0}$ = mean-squared error

$$\Phi(\tau)\Big|_{\tau=0} = E[X(f)^2] = [E(X)] + \text{Variance} \qquad (24)$$

Therefore, in order to properly scale the autocovariance, E(X) and σ^2 must be known. It came to our attention at the conclusion of this project that the subroutine "ASA" which calculates the power spectrum density calculates the following parameter.

$$Z = \frac{X - E[X]}{\sigma}$$

and calculates the spectrum for Z and not X. This in itself is of no major concern as long as E(X) and σ are both known. However, σ is not printed by ASA; in fact, the entire normalization procedure is never mentioned in the documentation.

The normalization procedure does not affect the shape of the spectrum, but it does affect the scale factor. Therefore, all results for the mean-squared error are conservative (i.e., the results do not include the variance which is unknown). Continuing in light of the above, the autocorrelation function is given as follows:

$$\Phi(\tau) = (.184)(.166)^2 e^{-19.1|\tau|}$$

$$\Phi(\tau) \doteq 50.703 \times 10^{-4} e^{-19.1|\tau|}$$

$$\Phi(\tau) = \Gamma^2 e^{-\alpha|\tau|} \qquad \Gamma = 7.01 \times 10^{-2} \text{ deg}, \quad \alpha = 19.1 \text{ sec}^{-1}$$

In discrete form,

$$\Phi(m) = \Gamma^2 e^{-\alpha T|m|} \tag{25}$$

The Z transform of $\Phi(m)$ is obtained by taking the sum of the individual Z transforms of the parts for $m > 0$ and $m < 0$. Letting $A = e^{-\alpha T}$ we have

$$\Phi(Z) = \left\{ \frac{1}{1-AZ^{-1}} + \frac{1}{1-AZ} - 1 \right\} \Gamma^2 \tag{26}$$

$$\Phi(Z) = \left\{ \frac{\Gamma\sqrt{1-A^2}}{1-AZ^{-1}} \right\} \left\{ \frac{\Gamma\sqrt{1-A^2}}{1-AZ} \right\} \tag{27}$$

$$H(Z) = \frac{\Gamma\sqrt{1-A^2}}{1-AZ^{-1}} \tag{28}$$

$$H(Z) = \Gamma\sqrt{1-A^2}\,[1 + AZ^{-1} + A^2 Z^{-2} + \cdots] \tag{29}$$

then

$$h(0) = \Gamma(\sqrt{1-A^2}) \tag{30}$$

$$C_{00} = \sqrt{\Phi(0) - h^2(0)} = \Gamma A \tag{31}$$

$$\xi_0 = C_{00}V(0) = \Gamma AV(0) \tag{32}$$

$$y(0) = \Gamma\sqrt{1-A^2}\ u(0) + \Gamma AV(0) \tag{33}$$

Since u(0) and v(0) are independent and their values do not enter the expression for y(n) for $n \geq 1$, y(0) can be generated more simply from a single random variable having the appropriate variance, by taking

$$y(0) = u(0) \tag{34}$$

Finally for $n \geq 1$

$$y(n) = \Gamma\sqrt{1-A^2}\ u(n) + Ay(n-1) \tag{35}$$

Since Γ represents the mean-squared error, this variable can be compared to the total error that is specified by RTCA. A copy of this is reproduced in Table 2. As can be seen the Hazeltine system can be used for configuration k, Cat III, based on azimuth data. The remaining results for the Hazeltine data are shown in Table 3.

Several significant conclusions can be made about the results presented in Table 3. First, the correlation time $(\frac{1}{\alpha})$ is considerably different for azimuth and elevation data, varying by a factor of twenty to one. Also, the correlation time for range is similar to the azimuth. The mean-squared error for range is just barely up to the requirements, and in the case of elevation, the requirement has not been met.

Table 2.
RTCA MLS Specifications.

Configuration Opeartional Use	D Cat. I	F Cat. II	K Cat. III
DME			
Bias	91.4 m (300 ft.)	30.5 m (100 ft.)	6.1 m (20 ft.)
Random	*	*	*
Total	91.4 m	30.5 m	6.1 m
AZ			
Bias	.125 degrees	.090 degrees	.036 degrees
Random	.065 degrees	.033 degrees	.024 degrees
Total	.141 degrees	.096 degrees	.042 degrees
EL			
Bias	0.050 degrees	.050 degrees	.050 degrees
Random	.058 degrees	.035 degrees	.035 degrees
Total	.077 degrees	.061 degrees	.061 degrees

* Random error negligible compared to bias.

Table 3.
Hazeltine MLS Error Models.

Function	Model	A	$\alpha(\frac{1}{sec})$	Γ	u(n)
Elevation	$y(n) = \Gamma\sqrt{1 - A^2}\, u(n) + Ay(n-1)$	$e^{-\alpha T}$	19.100	7.01×10^{-2} Degrees	IGRS
Azimuth	$y(n) = \Gamma\sqrt{1 - A^2}\, u(n) + Ay(n-1)$	$e^{-\alpha T}$	0.971	5.10×10^{-3} Degrees	IGRS
Range	$y(n) = \Gamma\sqrt{1 - A^2}\, u(n) + Ay(n-1)$	$e^{-\alpha T}$	1.013	21.1 Feet	IGRS

Note: IGRS = Independent Gaussian Random Sample

Initialization is achieved by setting $y(0) = u(0)$

3.7 Discrete Multipath Errors

The preceding discussion has described the validation results for the system noise component of the MLS error. The other error component included in the MLS model results from discrete multipath reflections. Data were not available on the Hazeltine or Texas Instrument tapes to obtain a time history of the multipath characteristics; however, Reference 8 contains the results of a study of discrete multipath effects. Figure 15 shows this effect for the elevation beam of a time reference MLS. Figure 16 displays the error for the localizer of a frequency reference MLS based on the simulation models described in Reference 9.

As can be seen for both error models the magnitude of the error builds up over a relatively long period of time, reaches a maximum and then decays. This effect can be explained by the model used for the discrete multipath reflections. As the aircraft approaches the specular reflection point the error due to reflections increases. Outside the specular reflection point the error is less but somewhat periodic. It would be highly desirable to develop simulation models for discrete multipath errors for the time reference MLS and to obtain empirical data to better verify this type of performance on both channels.

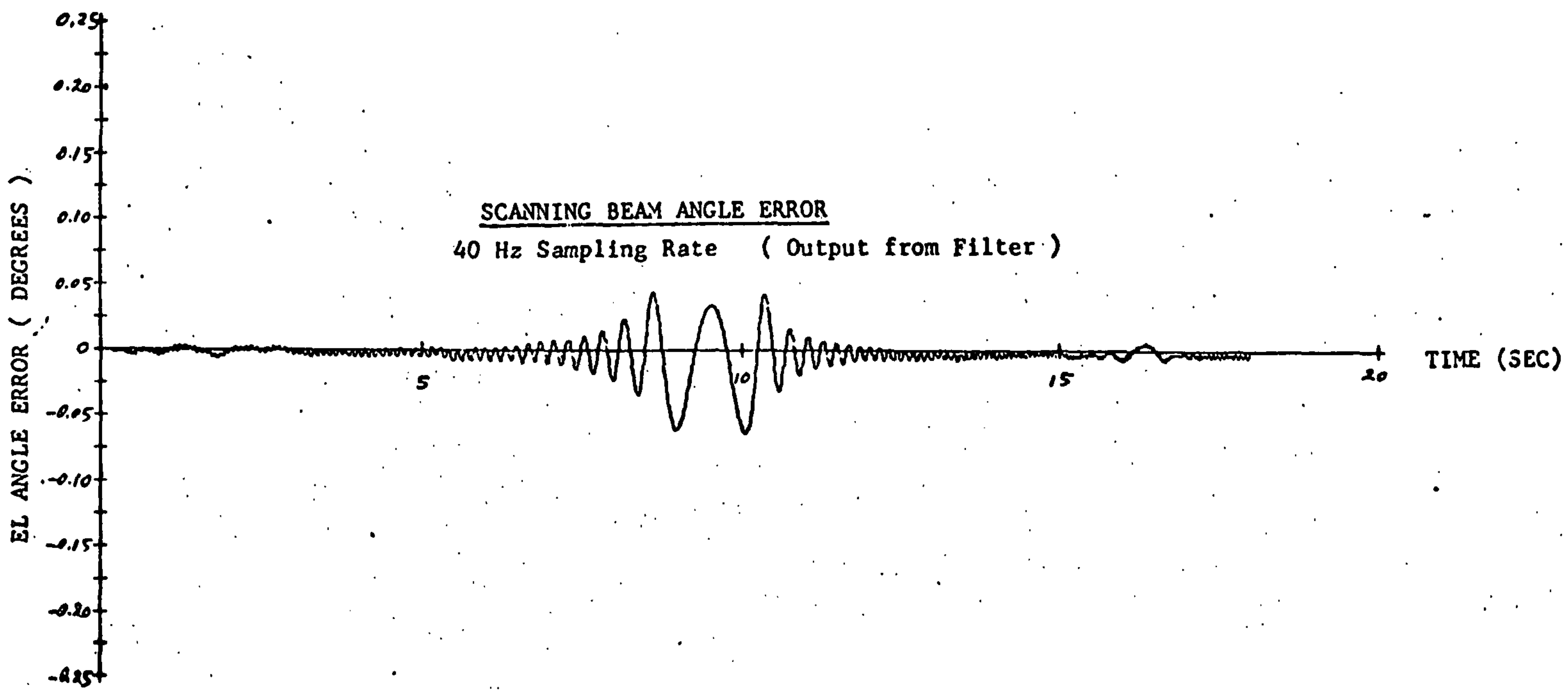

Figure 15. Scanning-Beam Angle Error, 40-Hz Sampling Rate (output from Filter).

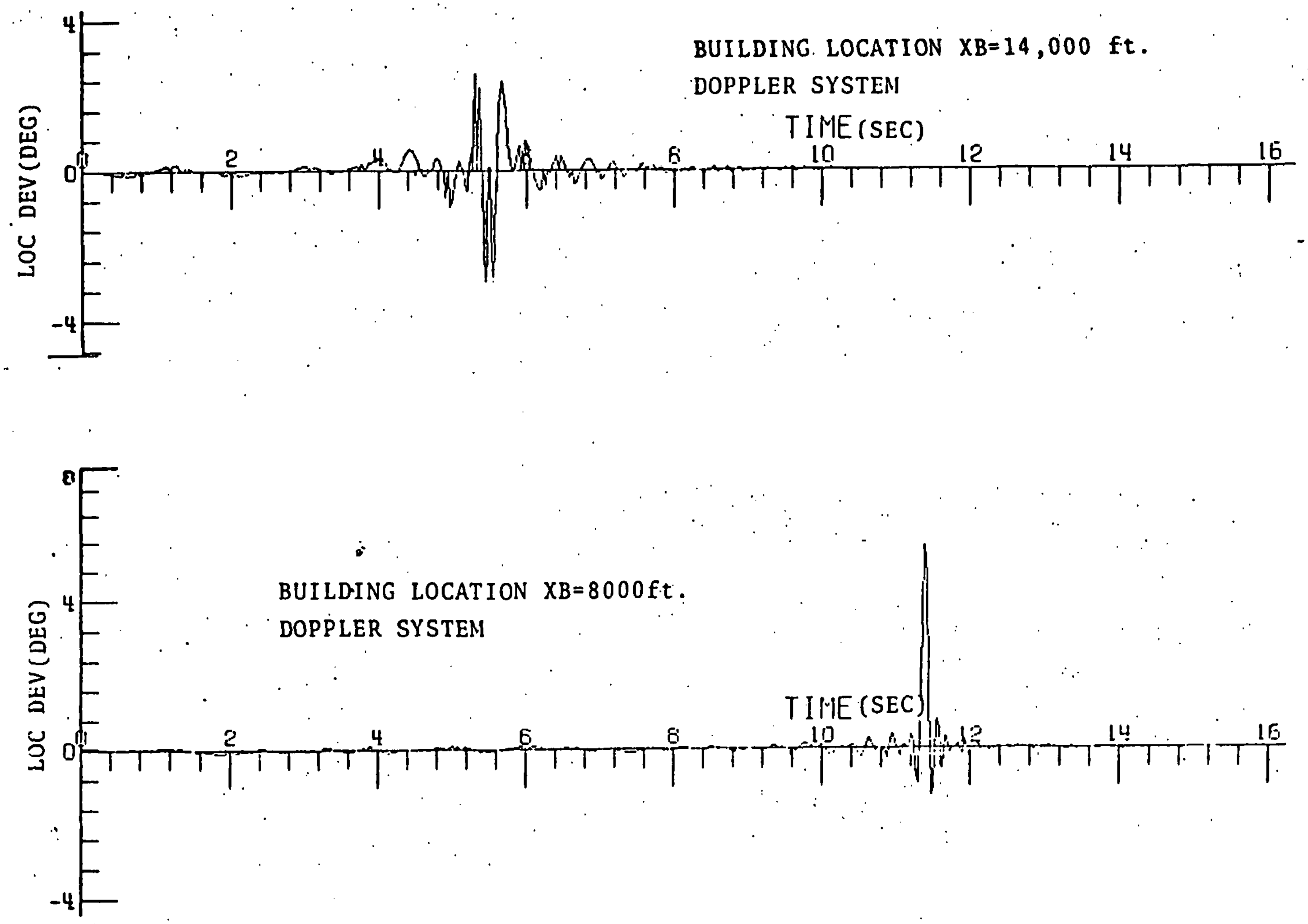

Figure 16. Time Plot of Discrete Multipath Errors.

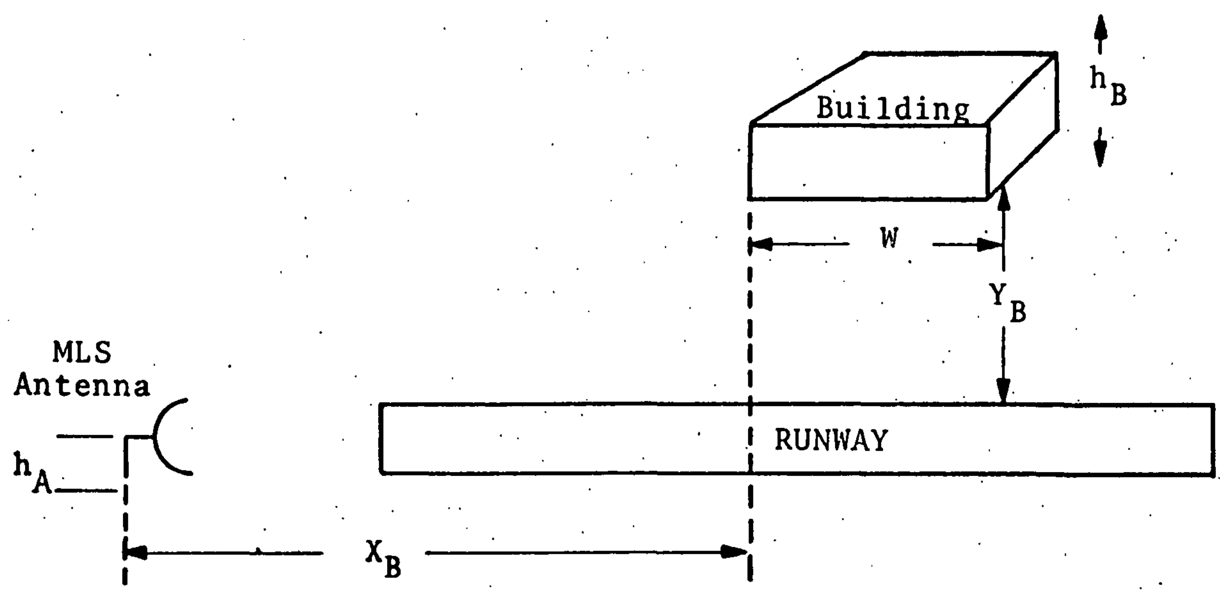

Figure 17. Airport Geometry.

4.0 REFERENCES

1. C.S. Burrus and T.W. Parks, "Time Domain Design of Recursive Digital Filters," IEEE Transactions Audio Electroacoustics, Volume AU-18, pp. 137-141, June 1970.

2. J.L. Shanks, "Recursion Filters for Digital Processing," Geophysics, Volume XXXII, No. 1, pp. 33-51, February 1967.

3. W.C. Yengst, "Approximation to a Specified Time Response," IRE Transactions Circuit Theory, Volume CT-9, pp. 152-162, June 1962.

4. Marquardt, C.W. (1963) "An Algorithm for Least-Squares Estimation of Non-Linear Parameters," J. Siam, Volume 2, #2, p. 431.

5. Levenberg, K. (1944) "A Method for the Solution of Certain Non-Linear Problems in Least Squares," Quarterly Appl. Math., Volume 2, pp. 164.

6. Bard, Y. (1970), "Comparison of Gradient Methods for the Solution of Non-Linear Parameter Estimation Problems," SIAM Journal of Numerical Analysis, Volume 7, No. 1, pp. 157.

7. Anon: "A New Guidance System for Approach and Landing," Volume 2, p. 3 (SC-117 Committee Reports) Radar Technical Commission for Aeronautics, 17.7 H Street, N.W., Washington, D.C., Document DO-148, 18 December 1970.

8. Anon: "Final Report of the MLS Scanning Beam Working Group; MLS Technique Assessment," Logistic Services, LG-340, Washington, D.C.; December 1974, pp. 2-116.

9. NASA CR-132562 "Development and Modification of a Digital Program for Final Approach to Landing."

APPENDIX A

TEXAS INSTRUMENT TAPE READ

```
      PROGRAM RDTI(INPUT,OUTPUT,TAPE5,TAPE11,TAPE12,TAPE13)
      DIMENSION IDS(10),JFG(2),AZ(100,5),EL(100,5),RANGE(100,6),
     1IAZ(100),IEL(100),IRNG(100)
      REAL MR,MF
      INTEGER AZT(100,4),ELT(100,1),RNGT(100,4)
        DATA(IDS(J),J=1,10)/3H AH,3H KA,3H K1,3H K2,3H DA,3H DE,3H EA,
     1 3H EE,3H RM,3H AT/
        DATA(JFG(N),N=1,2)/4H   S,4H   N/
      IFILE = 1
      IFT = 10
      ICOUNT = 0
      MEL=MAZ=MRNG=0
      JAZ=JEL=JRNG=0
      ISET=0
5       DO 6  I=1,12
6      READ(5,400)
400   FORMAT(102X)
10    READ(5,1000)ID,IH,IM,IS,IMS,PT,DIFR,XR,ZR,ITR1,TR2
1000  FORMAT(5X,I1,35X,3I2,1X,I3,F8.3,F8.3,F9.1,9X,F7.1,I6,F4.3)
      IF(EOF(5))999,15
15    CONTINUE
C TI VARIABLES NA UNDER HAZ FORMAT
      DRR=0.
      MR=0.
      MF=0.
      DIFF=0.
C CONVERT TO HAZ FORMAT
      IFG=4H   N
      IPT=1000.*PT
C     ZERO ID ON FIRST RECORD
      ID=ID+1
      IF(ID.LT.1)ID=1
      IF(ID.GT.6)ID=6
      GO TO (10,22,23,10,10,10),ID
C
C       AZIMUTH  DATA
C
C
22     CONTINUE
C AVOID TRACKER OFF RECORDS
      IF(XR.EQ.0 .AND. ZR.EQ.0) GO TO 10
C AVOID S RECORDS
```

Appendix A. Texas Instrument Fortran Extended Read Program for Data.

```
      IF(IFG.EQ.JFG(1))GO TO 10
      IF(ABS(DIFR).GT.ABS(PT))GO TO 10
      JAZ = JAZ + 1
      MAZ = MAZ + 1
      AZT(JAZ,1) = IH
      AZT(JAZ,2) = IM
      AZT(JAZ,3) = IS
      AZT(JAZ,4) = IMS
      AZ(JAZ,1) = PT
      AZ(JAZ,2) = MR/100.
      AZ(JAZ,3) = MF/100.
C NO FILTERING ON TI
      AZ(JAZ,4) = DIFR
      AZ(JAZ,5) = DIFR
      IAZ(JAZ) = IFG
      IF(JAZ.EQ.100)300,10
C
C       ELEVATION      DATA
C
23         CONTINUE
      IF(IFG.EQ.JFG(1))GO TO 10
      IF(XR.EQ.0 .AND. ZR.EQ.0) GO TO 10
C AVOID 130E70
      IF(ABS(DIFR).GT.ABS(PT))GO TO 10
      JEL = JEL + 1
      MEL = MEL + 1
      ELT(JEL,1) = IH
      ELT(JEL,2) = IM
      ELT(JEL,3) = IS
      ELT(JEL,4) = IMS
      EL(JEL,1) = PT
      EL(JEL,2) = MR/100.
      EL(JEL,3) = MF/100.
      EL(JEL,4) = DIFR
      EL(JEL,5) = DIFR
      IEL(JEL) = IFG
      IF(JEL.EQ.100)300,10
C
C       RANGE         DATA
C
24         CONTINUE
```

Appendix A. Texas Instrument Fortran Extended Read Program for Data. (Continued)

```
      IF(XR.EQ.0 .AND. ZR.EQ.0) GO TO 10
      IF(IFG.EQ.JFG(1))GO TO 10
      JRNG = JRNG + 1
      MRNG = MRNG + 1
      RNGT(JRNG,1) = IH
      RNGT(JRNG,2) = IM
      RNGT(JRNG,3) = IS
      RNGT(JRNG,4) = IMS
      RANGE(JRNG,1) =PT
      RANGE(JRNG,2) = MR
      RANGE(JRNG,3) = MF
      RANGE(JRNG,4) = DRR
      RANGE(JRNG,5) = DIFR
      RANGE(JRNG,6) = DIFR
      IRNG(JRNG) = IFG
      IF(JRNG.EQ.100)300,10
  101 CONTINUE
      ISET = 1
C
C     PRINT DATA
C
  300 CONTINUE
         PRINT 900
  900 FORMAT(//5X*K  AZ  DATA*/5X*TIME*11X*AZ(T)*10X*MR*13X*MF*13X*DIFR*
     111X*DIFF*11X*FLAG*10X*RECORD NO.*//)
      DO 500 I =1,JAZ
      WRITE(12)  (AZT(I,J),J=1,4),(AZ(I,J),J=1,5)
  500 PRINT 1001,(AZT(I,J),J=1,4),(AZ(I,J),J=1,5),IAZ(I),I
 1001 FORMAT(2X,I2,*:*I2,*:*I2,*.*I3,5(F10.3,5X),A4,10X,I4)
         PRINT 901
  901 FORMAT(//5X*K EL DATA*/ 6X*TIME*11X*EL(T)*10X*MR*13X*MF*13X*DIFR*
     111X*DIFF*11X*FLAG*10X*RECORD NO.*//)
      DO 501 I=1,JEL
      WRITE(13)  (ELT(I,J),J=1,4),(EL(I,J),J=1,5)
  501 PRINT 1001,(ELT(I,J),J=1,4),(EL(I,J),J=1,5),IEL(I),I
         PRINT 902
  902 FORMAT(//5X*RANGE DATA*/5X*TIME*11X*SRANGE(T)*6X*MR*13X*MF*13X
     1*RNG RATE*7X*DIFR*11X*DIFF* 6X*FLAG* 4X*REC NO.*//)
      DO 502 I=1,JRNG
      WRITE(11)  (RNGT(I,J),J=1,4),(RANGE(I,J),J=1,6)
  502 PRINT 1002,(RNGT(I,J),J=1,4),(RANGE(I,J),J=1,6),IRNG(I),I
 1002 FORMAT(2XI2,*:*I2,*:*I2,*.*I3,6(F10.3,5X),A4,10X,I4)
```

Appendix A. Texas Instrument Data. (Continued)

```
      JEL=JAZ=JRNG=0
      GO TO 10
  999 PRINT 1003,MEL,MAZ,MRNG
 1003 FORMAT(//5X*TOTAL EL MEASURENENTS =*I8,/5X*TOTAL AZ MEASUREMENTS =
     1*I8/5X*TOTAL RANGE MEASUREMENTS =*I8/)
      PRINT 903,IFILE
  903 FORMAT(//10X*END OF FILE NO. - *I3//)
      IFILE = IFILE + 1
      IF(IFILE.EQ.IFT)40,45
   40 CONTINUE
      STOP
   45 MEL=MAZ=MRNG=0
      ISET = 0
      GO TO 5
          END
```

```
RITY  DETAILS    DIAGNOSIS OF PROBLEM

   63 CD 19  FIELD WIDTH OF A CONVERSION DESCRIPTOR SHOULD BE AS LARGE AS THE MINIMUM SPECIFIED FOR TH


LIC REFERENCE MAP (R=1)
```

Appendix A. Texas Instrument Data. (Continued)

APPENDIX B

HAZELTINE TAPE READ

```
      PROGRAM ASAHAZ(INPUT,OUTPUT,TAPE5=INPUT,TAPE7=OUTPUT,TAPE11,
     +  TAPE12,TAPE13,TAPE8)
      DIMENSION DUM4(4),IDUM4(4)
      DIMENSION DUM8(8),DUM3(3),SAVE(1002)
      COMMON TC,HEADER,NRECS,LAG,NLAGS,IFILT,
     + IPRTI,IPRTO,IPLTI,IPLTO,T(10,1002),ERDATA(1002,2),
     + YIO,YIST,YISI,YIINC,YTO,YTST,YTSI,YTINC,
     + XB,CKSUM,TINC,DAXIS(1002),XAXIS(1002),YIN(1002),ITSEL,IPLT
      XB=0.
      CKSUM=0.
      IPLT=1
      YIO=-1.
      YIST=-40.
      YISI=2.
       YIINC=40.
      YTO=-1.
      YTST=-50.
      YTSI=2.
      YTINC=50.
      CALL PSEUDO
      READ(5,900)HEADER
      IF(EOF,5) 1050,2
    2 CONTINUE
900   FORMAT(A10)
1     READ(5,1000)TC,NRECS,LAG,TINC,IFILT,IPRTI,IPRTO,IPLTI,IPLTO,ITSEL
      WRITE(7,1000)TC,NRECS,LAG,TINC,IFILT,IPRTI,IPRTO,IPLTI,IPLTO,ITSEL
      IF(EOF,5) 1050,5
    5  CONTINUE
      NLAGS = LAG + 1
1000  FORMAT(A5,2I5,F5.0,5I1,I2)
      DO 10 I=1,500
      DO 10 J=1,10
10    T(J,I)=0.
      FMAX=1./(2.*TINC)
      DFLF=FMAX/LAG
      DO 20 I=1,NRECS
20    DAXIS(I)=(I-1)*TINC
      IF(TC.EQ.2HRG)GO TO 11
      IF(TC.EQ.2HAZ)GO TO 12
      IF(TC.EQ.2HFL)GO TO 13
      WRITE(7,1020)TC
      STOP
1020  FORMAT(* TAPE CODE ERROR*,A5)
11    DO 21 I=1,NRECS
```

Appendix B. Hazeltine Fortran Extended Read Program for Data.

```
12     DO 22 I=1,NRECS
       READ(12)IDUM4,DUM3,ERDATA(I,1),ERDATA(I,2)
       IF(EOF,12 ) 16,22
   16 NRECS = I-1
       IF(NRECS.GT.1) GO TO 200
       GO TO 1021
   22 CONTINUE
       GO TO 200
13     DO 23 I=1,NRECS
       READ(13)IDUM4,DUM3,ERDATA(I,1),ERDATA(I,2)
       IF(EOF,13 ) 17,23
   17 NRECS = I-1
       IF(NRECS.GT.1) GO TO 200
       GO TO 1021
   23 CONTINUE
200    CONTINUE
       DO 29 I=1,NRECS
29     SAVE(I)=ERDATA(I,IFILT)
       CALL ASA(NRECS,LAG,SAVE,XB,T,CKSUM)
C T ARRAY IS BACKWARDS
C ITSEL=1 TO 10
       DO 30 I=1,NLAGS
       XAXIS(I) = (I-1) * DELF
       X=T(ITSEL,I)*TINC
       T(ITSEL,I)=X
       IF(X.LT..01)X=.01
30     YIN(I)=ALOG10(X)
       WRITE(8)LAG,(T(ITSEL,I),XAXIS(I),I=1,LAG)
       CALL SPOUT
       CALL SPLOT
       GO TO 1040
1021   WRITE(7,1030)TC
1030   FORMAT(* EOF ON*,A5)
1040   REWIND 13
       REWIND 12
       REWIND 11
       GO TO 1

 1050 WRITE(7,1060) TC
 1060 FORMAT(* ALL INPUT DATA PROCESSED*/,
     + * LAST DATA WAS ON*,A5)
       STOP
```

Appendix B. Hazeltine Data. (Continued)

```
1050 WRITE(7,1060) TC
1060 FORMAT(* ALL INPUT DATA PROCESSED*/,
    + * LAST DATA WAS ON*,A5)
     STOP
     END
```

Appendix B. Hazeltine Data. (Continued)

APPENDIX C

NONLINEAR LEAST SQUARES PROGRAM

Note: This program was adapted from "The Computing Technology Center Numerical Analysis Library" CTC-39 G.W. Westley, Oct. 1970.

```
                SUBROUTINE NONLS2(NPAR,NPTS,IW,FAIL,MAXITR,IFXE,NEXT,LINMIN,IE,
               1ID,X,Y,WTS,ATA,B,BDB,EP,LAM,PHI,DAC)
          C
          C NONLINEAR LEAST SQUARES  ESTIMATOR
          C
          C REFERENCES :
          C
          C   MARQUAARDT "ALGORITHM FOR LEAST-SQUARES ESTIMATION OF NONLINEAR
          C
          C                  PARAMETERS" J.SIAM, VOL. 11,=2   P431
          C
          C   LEVENBERG  "METHOD FOR SOLUTION OF CERTAIN NONLINEAR PROBLEMS IN LS"
          C                  QUART. APPL.MATH.,VOL.2,P164
          C
          C
000027          REAL ATF(20),ANS(20),ANS1(20),EX(20),NORM,FV,SS,PHI1,PHI2,PHI3,
               1     GAMO,TOLR,         LD0,L10,GAMA
000027          REAL X(IE,1),Y(1),B(1),BDB(1),WTS(1),EP,DAC,LAM,PHI,ATA(ID,1),
               1     FVALUE,PARVAL
000027          INTEGER IFXE(1),INX(20),FAIL
000027          EXTERNAL FVALUE,PARVAL
000027          TOLR = 1.0E-08
000030          GAMAO = 45.
000032          ITER   = 0
000033          IR     = 20
          C
          C   DETERMINE  THE  PARAMETERS TOBE  VARIED
          C
          C
          C
000034          NVAR = 0
000035          DO 10  I=1,NPAR
000036             BDB(I)  = B(I)
000042             ANS(I)  = 0.0
000043             ANS1(I) = 0.0
000044             IF(IFXE(I).LE.0) GO TO 10
000046          NVAR = NVAR + 1
000047          INX(NVAR) = I
000051     10   CONTINUE
000054          IF(IW.GE.0) WRITE(6,11700) NPAR,NVAR,NPTS,MAXITR,IW,LINMIN,EP,DAC
          C
          C     EXAMINE THE WTS ARRAY
          C
000105          NORM = 0.
```

```
000106          DO 40 I = 1,NPTS
000110             IF (WTS(I)) 20,30,40
000113    20       WTS(I) = 1.0/  WTS(I)**2
000117             GO TO 40
000117    30       WTS(I) = 1.0
000122    40       NORM = NORM + WTS(I)
000130          NORM = FLOAT(NPTS)/ NORM
000131          IF (NORM.EQ.1.) GO TO 60
000133          DO 50 I=1,NPTS
000134    50       WTS(I) = WTS(I)*NORM
         C
         C      CALCULATE THE INITIAL SUM OF SQUARES
         C
000141    60    PHI = 0.0
000142          IF(IW.GT.0) WRITE(6,10200)
000153          DO 70 I=1,NPTS
000155             CALL   FVALUE(X,B,IE,FV,I)
000161             SS = Y(I)-FV
000170             IF (IW.GT.0) WRITE(6,10000) I,Y(I),FV,SS
000216    70    PHI = PHI + SS**2*WTS(I)
         C
         C   START THE DAMPED GAUSSIAN PROCEDURE
         C
000226    80    ITER = ITER + 1
000230          IF(IW.GT.0) WRITE(6,11800)
000241          IF(ITER.GT.MAXITR)  GO TO  380
000244          DO 90  I=1,NPAR
000245    90    BDB(I) = B(I)
000253          DO 100 I=1,NVAR
000254             ATF(I)  = 0.0
000255             DO 100 J=I,NVAR
000257    100         ATA(I,J) = 0.0
         C
         C   GENERATE THE ATA AND ATF  ARRAYS
         C
000273          DO 110  I=1,NPTS
000274              CALL FVALUE(X,B,IE,FV,I)
000300              NORM = WTS(I) * (Y(I)-FV)
000311              CALL PARVAL(X,B,IE,I,EX)
000316              DO 110 L=1,NVAR
000323                 J=INX(L)
000325                 ATF(L) = ATF(L) + NORM* EX(J)
000331                 DO 110 M =L,NVAR
000332                    K = INX(M)
000334    110             ATA(L,M)=  ATA(L,M)+ EX(J)* EX(K) * WTS(I)
```

```
000356         IF(IW.GT.0) WRITE(6,11500) ITER,PHI,(B(I),I=1,NPAR)
             C
             C  PERFORM A LOCAL SCALING ON THE ATA MATRIX TO AID CALCULATIONS
             C
000407         DO 120 I=1,NVAR
000411            IF(ATA(I,I).EQ.0.) GO TO 410
000416    120  EX(I) = SQRT(ATA(I,I))
000434         DO 140 I=1,NVAR
000436            ATF(I) = ATF(I)/EX(I)
000440            DO 140 J=I,NVAR
000442               IF(I.EQ.J) GO TO 130
000443               ATA(I,J) = ATA(I,J)/(EX(I)*EX(J))
000453               GO TO 140
000453    130  ATA(I,I) = 1.0
000461    140  CONTINUE
             C  DETERMINE A VALID LAMDA FOR THE SCALED PARTIAL MATRIX
             C
             C
000466         FAC = 1.0
000467         CALL NEWLAM(ATA,LAM,BDB,ATF,ANS,EX,GAMA,FVALUE,PHI1,X,Y,WTS,B,
             >  ID,IE,NVAR,NPTS,NPAR,IFAL,INX,IR )
000521         IF (IW.GT.0) WRITE(6,10600) LAM,PHI1,GAMA
000544         IF (IFAL-1) 150,390,400
000547    150  DO 160 I=1,NVAR
000551            KDUMMY = INX(I)
000553            IF (ABS(ANS(I)).GT.(TOLR+EP*ABS(B(KDUMMY)))) GO TO 180
000566    160  CONTINUE
000570         IF (PHI.LT.PHI1) GO TO 440
000573         DO 170 I=1,NPAR
000574    170     B(I) = BDB(I)
000602         PHI = PHI1
000603         GO TO 440
000604    180  IF (PHI1.GE.PHI) GO TO 220
000607         IF (LAM.LE.TOLR) GO TO 320
000612         LDO = LAM/10.0
000614         CALL NEWLAM(ATA,LDO,BDB,ATF,ANS1,EX,GAMA,FVALUE,PHI2,X,Y,WTS,B,
             >  ID,IE,NVAR,NPTS,NPAR,IFAL,INX,IR )
000645         IF (IW.GT.0) WRITE(6,10700) LDO,PHI2,GAMA
000670         IF (IFAL-1) 190,390,400
000673    190  IF (PHI2.GE.PHI1) GO TO 320
000676    200  LAM = LDO
000700         DO 210 I=1,NVAR
000702    210     ANS(I) = ANS1(I)
000706         PHI1 = PHI2
000710         GO TO 320
```

```
000710     220  LDO = LAM/10.0
000713          CALL NEWLAM(ATA,LDO,BDB,ATF,ANS1,EX,GAMA,FVALUE,PHI2,X,Y,WTS,B,
               >  ID,IE,NVAR,NPTS,NPAR,IFAL,INX,IR)
000744          IF (IW.GT.0) WRITE(6,10700) LDO,PHI2,GAMA
000767          IF (IFAL-1) 230,390,400
000772     230  IF (PHI2.LT.PHI) GO TO 200
000775          L10 = LAM
000777     240  L10 = L10*10.0
001001          CALL NEWLAM(ATA,L10,BDB,ATF,ANS ,EX,GAMA,FVALUE,PHI3,X,Y,WTS,B,
               >  ID,IE,NVAR,NPTS,NPAR,IFAL,INX,IR )
001032          IF (IW.GT.0) WRITE(6,10900) L10,PHI3,GAMA
001055          IF (IFAL-1) 250,390,400
001060     250  IF (PHI3.GE.PHI) GO TO 260
001063          PHI1 = PHI3
001064          LAM = L10
001066          GO TO 320
001067     260  IF (GAMA.GE.GAMAO) GO TO 240
001072          FAC = FAC/2.0
001073          DO 270 I=1,NVAR
001075     270     ANS(I) = ANS(I)/2.0
001101          DO 280 I=1,NVAR
001103             KDUMMY = INX(I)
001105             IF (ABS(ANS(I)).GT.(TOLR+EP*ABS(B(KDUMMY)))) GO TO 290
001120     280  CONTINUE
001122          GO TO 430
001122     290  DO 300 I=1,NVAR
001124             KDUMMY = INX(I)
001126     300     BDB(KDUMMY) = B(KDUMMY) + ANS(I)
001135          PHI3 = 0.0
001136          DO 310 I=1,NPTS
001137             CALL FVALUE(X,BDB,IE,FV,I)
001143     310     PHI3 = PHI3 + (Y(I)-FV)**2*WTS(I)
001157          IF (IW.GT.0) WRITE(6,10300) FAC,PHI3
001174          GO TO 250
001175     320  CONTINUE
              C
              C    AT THE POINT OF THE LINEAR MINIMIZATION PROCEDURE THE CORRECTION
              C    VECTOR IS IN ANS AND THE LAMDA THAT PRODUCED THIS VECTOR IS IN
              C    LAM.
              C
001175          IF (IW.LE.1) GO TO 360
001177          IF (NVAR.EQ.NPAR) GO TO 340
001201          DO 330 I=1,NPAR
001202     330     EX(I) = 0.0
001205     340  DO 350 I=1,NVAR
```

```
001207              KDUMMY = INX(I)
001211      350     EX(KDUMMY) = ANS(I)
001215          WRITE(6,11600) (EX(I),I=1,NPAR)
001234      360 CONTINUE
001234          CALL PARLIN(ANS,B,BDB,FVALUE,PARVAL,X,Y,WTS,PHI,PHI1,IE,NPTS,
               > LINMIN,NVAR,INX,DAC,IW,EX)
001266          IF (IW.GT.1) WRITE(6,10800) PHI,(B(I),I=1,NPAR)
001322          DO 370 I=1,NVAR
001324              KDUMMY = INX(I)
001326            IF (ABS(ANS(I)).GT.(TOLR+EP*ABS(BDB(KDUMMY)))) GO TO 80
001341      370 CONTINUE
001343          GO TO 440
001343      380 IF (IW.GE.0) WRITE(6,12300)
001355          FAIL = 2
001356          GO TO 450
001357      390 IF (IW.GE.0) WRITE(6,12000)
001371          FAIL = 3
001372          RETURN
001373      400 IF (IW.GE.0) WRITE(6,11900) GAMA,LAM
001411          FAIL = 4
001412          RETURN
001413      410 IF (IW.GE.0) WRITE(6,11400) I,(B(K),K=1,NPAR)
001443          FAIL = 6
001444          RETURN
001445      420 IF (IW.GE.0) WRITE(6,10500)
001457          FAIL = 5
001460          RETURN
001461      430 IF (IW.GT.0) WRITE(6,12400)
001472          FAIL = 1
001473          GO TO 450
001474      440 IF (ITER.EQ.1) GO TO 80
          C
          C      THE PROCEDURE HAS CONVERGED
          C
001476          FAIL = 0
001477          IF (IW.GE.0) WRITE(6,11802)
001510          IF (IW.GE.0) WRITE(6,12200) ITER
001524      450 IF (IW.GE.0) WRITE(6,12100) PHI,(B(I),I=1,NPAR)
          C
          C
          C      CALCULATE THE STANDARD ERROR.  USE NPTS - NVAR - NEXT AS THE DEF.
          C
001554          NORM = SQRT(PHI/FLOAT(NPTS - NVAR - NEXT))
          C
          C      REMAKE THE PARTIAL MATRIX INSTEAD OF USING THE RE-SCALED MATRIX.
```

```
              C      THIS COULD BE EASILY CHANGED.
              C
001566                 DO 460 I=1,NPAR
001570         460        BDR(I) = 0.0
001574                 DO 470 I=1,NVAR
001575                    DO 470 J=1,NVAR
001576         470           ATA(I,J) = 0.0
001612                 DO 480 I=1,NPTS
001613                    CALL PARVAL(X,B,IE,I,EX)
001617                    DO 480 L=1,NVAR
001624                       J = INX(L)
001626                       DO 480 M=L,NVAR
001627                          K = INX(M)
001631         480              ATA(L,M) = ATA(L,M) + EX(J)*EX(K)*WTS(I)
001653                 CALL INVRTA(NVAR,ATA,ID,IFU)
001657                 IF (IFU.EQ.1) GO TO 420
001665                 DO 490 I=1,NVAR
001666                 EX(I) = SQRT(ATA(I+1,I))
001702                       KDUMMY = INX(I)
001703         490     BDB(KDUMMY) = NORM*EX(I)
              C
              C      CALCULATE THE CORRELATION MATRIX.
              C
001711                 DO 500 I=1,NVAR
001712                    DO 500 J=I,NVAR
001713         500           ATA(I,J) = ATA(J+1,I)/(EX(I)*EX(J))
001741                 IF (IW.LT.0) RETURN
              C
              C      PRINT THE PARAMETERS AND THE STANDARD ERRORS ASSOCIATED TO THEM.
              C
001743                      DO 555  I =1,NVAR
001745                       KDUMMY = INX(I)
001747                 WRITE(6,11300)  INX(I),B(KDUMMY),BDB
001772         555          CONTINUE
002000                 WRITE(6,11802)
002004                 IF (NVAR.EQ.1) GO TO 520
002012                 IF (FAIL.EQ.2) GO TO 540
              C
              C      PRINT THE CORRELATION MATRIX
              C
002014                 WRITE(6,11200)
002017                 DO 510 I=1,NVAR
002024         510        WRITE(6,11000)I,(ATA(J,I),J=1,I     )
              C
              C      PRINT THE INVERSE MATRIX
```

```
           C
002055           WRITE(6,11802)
002061     520   WRITE(6,11100)
002065           M = NVAR + 1
002067           DO 530 I=2,M
002074              K=I-1
002076                   DO  530  J=1,K
002077     530      WRITE(6,10400) I, ATA(I,J)
002132     540   CONTINUE
002132           WRITE(6,11802)
002136           WRITE(6,10100)
002142           DO 550 I=1,NPTS
002147              CALL FVALUE(X,B,IE,FV,I)
002153              SS = Y(I)-FV
002162     550      WRITE(6,10000) I,Y(I),FV,SS
002214           RETURN
002214   10000 FORMAT(1H ,I5,3F16.7)
002214   10100 FORMAT(1H0,*FINAL DEVIATIONS*/1H ,13X,*OBS*,13X,*CALC*,12X,*O-C*)
002214   10200 FORMAT(1H0,*INITIAL DEVIATIONS*/1H ,13X,*OBS*,13X,*CALC*,12X,
                > *O-C*)
002214   10300 FORMAT(1H0,*FAC = *,F15.6,6X,*  PHI(FAC) = *,F16.7)
002214   10400 FORMAT(1H ,*ROW *,I2/(1H ,8E15.6))
002214   10500 FORMAT(1H0,*FINAL A TRANSPOSE A IS NOT POSITIVE DEFINATE*)
002214   10600 FORMAT(1H0,* IN-L = *,F16.8,*      PHI(IN-L) = *,F20.10,
                > *       GAMA = *,F10.4)
002214   10700 FORMAT(1H0,* L/10 = *,F16.8,*      PHI(L/10) = *,F20.10,
                > *       GAMA = *,F10.4)
002214   10800 FORMAT(3H0L ,7F16.7/(3H   ,16X,6F16.7))
002214   10900 FORMAT(1H0,* L*10 = *,F16.8,*      PHI(L*10) = *,F20.10,
                > *       GAMA = *,F10.4)
002214   11000 FORMAT(1H ,*ROW *,I2/(1H ,8E15.6))
002214   11100 FORMAT(1H0,* INVERSE MATRIX - LOWER TRIANGULAR PORTION*)
002214   11200 FORMAT(1H0,*CORRELATION MATRIX LOWER TRIANGULAR PORTION ROW BY ROW
                > PRINT*)
002214   11300 FORMAT(1H0,*  VARIBLE*,6X,*PARAMETER VALUE*,5X,*STANDARD ERROR*/
                > (1H ,4X,I2,9X,E15.6,7X,E10.4))
002214   11400 FORMAT(1H0,*THE DIAGONAL ELEMENT RESULTING FROM THE PARTIAL WRT
                >B(*,I2,*) IS = 0.0*/1H ,*THE POINT AT WHICH THE FAILURE OCCURRED
                >IS */(1H ,7F16.7))
002214   11500 FORMAT(1H0,*ITERATION  *,I3/1H ,7F16.7/(1H ,16X,6F16.7))
002214   11600 FORMAT(1H0,*DIR-VEC*,5X,6F16.7/(1H0,12X,6F16.7))
002214   11700 FORMAT(1H1,* NPAR*4X*NVAR*4X*NPTS*4X*MAXITR*4X*IW*3X*LINMIN*8X
                1*EP*13X*DAC*/1H ,2XI2,6XI2,7XI2 ,6XI2,5XI2,5XI2,7XE11.4,5XE11.4)
002214   11800 FORMAT(1H0/)
002214   11802 FORMAT(1H0,4(/))
```

```
002214   11900 FOPMAT(1H0,*GAMA = *,F20.10,* WHEN LAM = * ,F20.10/
              >   1H ,*THERE PROBABLY EXISTS EXCESSIVELY HIGH CORRELATIONS BETWEE
              >N THE PARAMETERS*)
002214   12000 FORMAT(1H0,*THE (ATA + LAM*I) MATRIX FAILED TO BE POS. DEF.*)
002214   12100 FORMAT(1H0,7F16.8/1H ,1LX,6F16.8))
002214   12200 FORMAT(1H0,*OPTIMAL POINT REACHED IN  *,I5,*  ITERATIONS*)
002214   12300 FORMAT(1H0,*MAXIMUM NUMBER OF ITERATIONS REACHED - BEST POINT PRIN
              >TED*//)
002214   12400 FORMAT(1H0,*THE DELTA-B VECTOR REDUCED TO CONVERGENCE LEVEL WHILE
              >GAMA LESS THAN GAMAO.*/1H ,*THE POINT IS PROBABLY OPTIMAL WITHIN
              >ROUNDING ERRORS.*)
002214         END
```

NONLS2

SUBPROGRAM LENGTH
002762

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

10	-	000051	20	-	000113	30	-	000117	40	-	000122
50	-	000134	60	-	000141	70	-	000216	80	-	000226
90	-	000245	100	-	000257	130	-	000453	140	-	000461
150	-	000547	170	-	000574	180	-	000604	190	-	000673
200	-	000676	210	-	000702	220	-	000710	230	-	000772
240	-	000777	250	-	001060	260	-	001067	270	-	001075
290	-	001122	320	-	001175	330	-	001202	340	-	001205
360	-	001234	380	-	001343	390	-	001357	400	-	001373
410	-	001413	420	-	001445	430	-	001461	440	-	001474
450	-	001524	460	-	001570	470	-	001576	500	-	001713
520	-	002061	540	-	002132	10000	-	002227	10100	-	002232
10200	-	002242	10300	-	002252	10400	-	002260	10500	-	002264
10600	-	002273	10700	-	002305	10800	-	002317	10900	-	002324
11000	-	002336	11100	-	002342	11200	-	002351	11300	-	002361
11400	-	002374	11500	-	002414	11600	-	002423	11700	-	002431
11800	-	002447	11802	-	002451	11900	-	002454	12000	-	002474
12100	-	002503	12200	-	002507	12300	-	002516	12400	-	002526

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

ANS	-	002610	ANS1	-	002634	ATA	-	000007	ATF	-	002564
B	-	000010	BDB	-	000011	DAC	-	000015	EP	-	000012
EX	-	002660	FAC	-	002756	FV	-	002705	GAMA	-	002716
GAMAO	-	002745	GAMO	-	002712	H.	-	002717	I	-	002751
ID	-	000003	IE	-	000002	IFAL	-	002757	IFU	-	002761
INX	-	002721	IR	-	002747	ITER	-	002746	J	-	002752
K	-	002755	KDUMMY	-	002760	L	-	002753	LAM	-	000013
LDO	-	002714	LINMIN	-	000001	L10	-	002715	M	-	002754
NEXT	-	000000	NORM	-	002704	NVAR	-	002750	PHI	-	000014
PHI1	-	002707	PHI2	-	002710	PHI3	-	002711	SS	-	002706
TOLR	-	002713	WTS	-	000006	X	-	000004	Y	-	000005

START OF CONSTANTS
002216

```
START OF TEMPORARIES
002546

START OF INDIRECTS
002556

UNUSED COMPILER SPACE
032700
```

```
               SUBROUTINE NEWLAM(ATA,LAM,B1,ATF,ANS,EX,GAMA,FVALUE,PHI,X,Y,WTS,
              > B,ID,IE,NVAR,NPTS,NPAR,IFAL,INX,IR)
000030         REAL ATA(ID,1),LAM,B1(1),ATF(IR),ANS(IR),EX(1),GAMA,FVALUE,PHI,
              >  Y(1),X(IE,1),WTS(1),FV,B(1),SUM1,SUM2,SUM3,COSGAM,CON,DET
000030         INTEGER INX(1)
          C
          C
          C     THIS ROUTINE PERFORMS THE FOLLOWING CALCULATIONS:
          C
          C        1. SOLVES THE SYSTEM (ATA + LAM*I) * ANS = ATF FOR A GIVEN LAMD
          C        2. INCREMENTS THE VECTOR B1 = B + CORRECTION VECTOR.
          C        3. COMPUTES PHI FOR THIS NEW VECTOR.
          C        4. IFAL = 0  ALL IS OK.
          C           IFAL = 1  NOT POSITIVE DEFINATE MATRIX
          C           IFAL = 2  LAMDA - GAMMA TEST FAILURE
          C
          C
000030         CON = 57.2957795
000031         IFAL = 0
000032         DO 10 I=1,NVAR
000034            ATA(I,I) = 1.0
000041      10    ATA(I,I) = ATA(I,I) + LAM
          C
          C     SOLVE THE SYSTEM (ATA + LAM*I) * ANS = ATF
          C
000047         CALL SOLVER(ATA,ATF,ANS,ID,IR,NVAR,1,IFAIL,DET,IDET)
000060         IF (IFAIL.EQ.0) GO TO 20
000065         IFAL =1
000066         RETURN
          C
          C
          C     THIS PORTION PERFORMS THE FOLLOWING CALCULATIONS:
          C
          C        1. ADJUST THE ANS ARRAY TO THE FAC COEFFICIENT
          C        2. COMPUTE THE COSINE(GAMMA) AND THE TERM GAMMA.
          C             REQUIRES THAT THE SYSTEM HAVE AN ARC-COSINE ROUTINE.
          C             IF NO SUCH ROUTINE IS PRESENT USE THE FOLLOWING:
          C              SS = ABS(COSGAM)
          C              GAMA = CON*(1.5707288+SS*(-.2121144+SS*(.074261-SS*
          C                     .0187293)))*SQRT(1.0/SS)
          C              IF(COSGAM.LT.0.0) GAMA = 180.0 - GAMA
          C        3.   PERFORM THE LANDA-GAMMA TEST FOR FORCED CONVERGENCE.
          C
          C
```

```
000067        20 DO 30 I=1,NVAR
000071        30    ANS(I) = ANS(I)/EX(I)
000077           IFAIL = 0
000100           IF (NVAR.NE.1) GO TO 40
000102           GAMA = 0.0
000103           GO TO 70
000104        40 SUM1 = 0.0
000105           SUM2 = 0.0
000106           SUM3 = 0.0
000107           DO 50 I=1,NVAR
000110              SUM1 = SUM1 + ANS(I)*ATF(I)
000115              SUM2 = SUM2 + ATF(I)**2
000117        50    SUM3 = SUM3 + ANS(I)**2
000124           COSGAM = SUM1/SQRT(SUM2*SUM3)
000131           GAMA =  ACOS(COSGAM) * CON
000134           IF (COSGAM.GT.0.0) GO TO 60
000142           GAMA = 180.0 - GAMA
000144           IF (LAM.LT.1.0) GO TO 60
000146           IFAIL = 1
000147        60 IF (IFAIL.EQ.0) GO TO 70
000150           IFAL = 2
000152           RETURN
000152        70 DO 80 I=1,NVAR
000154              K = INX(I)
000157        80    B1(K) = B(K) + ANS(I)
000166           PHI = 0.0
000167           DO 90 I=1,NPTS
000171              CALL FVALUE(X,B1,IE,FV,I)
000204        90    PHI = PHI + (Y(I) - FV)**2 * WTS(I)
000216           RETURN
000216           END
```

```
NEWLAM

SUBPROGRAM LENGTH
000255

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
20        -  000067     30        -  000071     40        -  000104     60        -  000147
70        -  000152

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
B         -  000006     CON       -  000247     COSGAM    -  000246     DET       -  000250
FV        -  000242     FVALUE    -  000001     GAMA      -  000000     I         -  000251
ID        -  000007     IDET      -  000253     IE        -  000010     IFAIL     -  000252
IFAL      -  000014     INX       -  000015     IR        -  000016     K         -  000254
NPAR      -  000013     NPTS      -  000012     NVAR      -  000011     PHI       -  000002
SUM1      -  000243     SUM2      -  000244     SUM3      -  000245     WTS       -  000005
X         -  000003     Y         -  000004

START OF CONSTANTS
000220

START OF TEMPORARIES
000225

START OF INDIRECTS
000236

UNUSED COMPILER SPACE
041500
```

```
                 SUBROUTINE PARLIN(ANS,B,BDB,FVAL,PARVAL,X,Y,WTS,PHI,PHI1,IE,NPTS,
                > LINMIN,NVAR,INX,ACC,IW,EX)
000025           REAL ANS(1),B(1),BDB(1),FVAL,PARVAL,X(IE,1),Y(1),WTS(1),PHI,
                > PHI1,ACC,FV,EX(1),NORM,CON,R0,F1,L0,L1,F0,R1,L2,F2,R2,R3,R4,DA
000025           INTEGER INX(1)
000025           NLN = 0
000026           DO 10 I=1,NVAR
000027        10    BDB(I) = 0.0
000033           DO 20 I=1,NPTS
000034              CALL FVALUE(X,B,IE,FV,I)
000037              NORM = WTS(I)*(Y(I)-FV)
000050              CALL PARVAL(X,B,IE,I,EX)
000063              DO 20 L=1,NVAR
000064              KDUMMY = INX(L)
000067        20       BDB(L) = BDB(L) + NORM*EX(KDUMMY)
000102           NORM = 0.0
000102           DO 30 I=1,NVAR
000104        30    NORM = NORM + ANS(I)*BDB(I)
000113           CON = NORM
000114           NORM = ABS(PHI/NORM)
000116           R0 = AMIN1(NORM,1.0)
000121           IF (R0.EQ.1.0) GO TO 60
000122           DO 40 I=1,NVAR
000124              KDUMMY = INX(I)
000127        40    BDB(KDUMMY) = B(KDUMMY) + R0*ANS(I)
000137           F1 = 0.0
000140           DO 50 I=1,NPTS
000141              CALL FVALUE(X,BDB,IE,FV,I)
000144        50    F1 = F1 + (Y(I)-FV)**2*WTS(I)
000161           IF (IW.GT.2)  WRITE(6,10100) F1,R0
000177           NLN = NLN + 1
000201           GO TO 70
000201        60 F1 = PHI1
000203        70 L0 = 0.0
000204           L1 = R0
000206           F0 = PHI
000207        80 R1 = CON*R0*R0/(F1-F0+2.0*CON*R0)
000216           IF (R1.GT.0.0) GO TO 130
000220        90 L2 = L1 + L1
000222       100 F2 = 0.0
000223           DO 110 I=1,NVAR
000225              KDUMMY = INX(I)
000230       110    BDB(KDUMMY) = B(KDUMMY) + L2*ANS(I)
000240           DO 120 I=1,NPTS
```

```
000241            CALL FVALUE(X,BDB,IE,FV,I)
000244      120   F2 = F2 + (Y(I)-FV)**2*WTS(I)
000261         IF (IW.GT.2) WRITE(6,10100) F2,L2
000277         NLN = NLN + 1
000301         IF (NLN.GT.LINMIN) GO TO 320
000304         IF (F2.GE.F1) GO TO 200
000306         L0 = L1
000310         F0 = F1
000311         L1 = L2
000312         F1 = F2
000312         GO TO 90
000313      130 IF (R1-L1) 150,90,140
000316      140 L2 = R1
000320         GO TO 100
000320      150 R2 = AMAX1(.25*R0,AMIN1(.75*R0,R1) )
000330         DO 160 I=1,NVAR
000332            KDUMMY = INX(I)
000335      160   BDB(KDUMMY) = B(KDUMMY) + R2*ANS(I)
000345         NORM = 0.0
000346         DO 170 I=1,NPTS
000347            CALL FVALUE(X,BDB,IE,FV,I)
000352      170   NORM = NORM + (Y(I)-FV)**2*WTS(I)
000367         IF (IW.GT.2) WRITE(6,10100) NORM,R2
000405         NLN = NLN + 1
000407         IF (NLN.GT.LINMIN) GO TO 320
000412         IF (NORM.LT.F0) GO TO 180
000414         L1 = R2
000416         F1 = NORM
000417         R0 = R2
000420         GO TO 80
000420      180 IF (NORM.LE.F1) GO TO 190
000423         L0 = R2
000424         F0 = NORM
000425         GO TO 90
000425      190 L2 = L1
000427         F2 = F1
000430         L1 = R2
000431         F1 = NORM
000432      200 K = 1
000433         R3 = .5*(F0*(L1**2-L2**2) + F1*(L2**2-L0**2) + F2*(L0**2-L1**2))/
              >   (F0*(L1-L2) + F1*(L2-L0) + F2*(L0-L1))
000465         IF (ABS(R3-L1).LE.ACC*L1) GO TO 290
000472         R4 = AMAX1(L0+.1*(L2-L0),AMIN1(L0+.9*(L2-L0),R3))
000506      210 NORM = 0.0
000507         DO 220 I=1,NVAR
```

```
000511          KDUMMY = INX(I)
000514   220    BDB(KDUMMY) = B(KDUMMY) + R4*ANS(I)
000524        DO 230 I=1,NPTS
000525          CALL FVALUE(X,BDB,IE,FV,I)
000530   230    NORM = NORM + (Y(I)-FV)**2*WTS(I)
000545        IF (IW.GT.2) WRITE(6,10000) NORM,R4
000563        NLN = NLN + 1
000565        IF (NLN.GT.LINMIN) GO TO 320
000570        IF (R4 .EQ. L1) GO TO 290
000572        IF (R4 .GT. L1) GO TO 270
000575        IF (NORM.LT.F1) GO TO 250
000577        L0 = R4
000600        F0 = NORM
000601        IF (K.EQ.2) GO TO 200
000603        R4 = (L1 + L2)/2.0
000606   240  K = 2
000607        GO TO 210
000610   250  L2 = L1
000612        F2 = F1
000613   260  L1 = R4
000615        F1 = NORM
000616        GO TO 200
000617   270  IF (NORM.GE.F1) GO TO 280
000622        L0 = L1
000623        F0 = F1
000624        GO TO 260
000624   280  L2 = R4
000626        F2 = NORM
000627        IF (K.EQ.2) GO TO 200
000631        R4 = (L1 + L2) /2.0
000634        GO TO 240
000635   290  DA = 1.0
000637        PHI = PHI1
000641        IF (PHI1.LE.F1) GO TO 300
000643        DA = L1
000644        PHI = F1
000645   300  DO 310 I=1,NVAR
000647          K = INX(I)
000652          BDB(K) = B(K)
000655          ANS(I) = DA*ANS(I)
000657   310    B(K) = B(K) + ANS(I)
000665        GO TO 330
000665   320  DA = 1
000667        PHI = PHI1
000671        CON = AMIN1(F0,F1,F2)
```

```
000676              IF (CON.GE.PHI1) GO TO 300
000700              IF (CON.EQ.F0) DA = L0
000703              IF (CON.EQ.F1) DA = L1
000707              IF (CON.EQ.F2) DA = L2
000713              PHI = CON
000715              GO TO 300
000715          330 RETURN
000716        10000 FORMAT (3H0B ,F25.12,5X,F15.6)
000716        10100 FORMAT (3H0S ,F25.12,5X,F15.6)
000716              END
```

```
PARLIN

SUBPROGRAM LENGTH
001033

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10     - 000027   30     - 000104   60     - 000201   70     - 000203
80     - 000207   90     - 000220   100    - 000222   130    - 000313
140    - 000316   150    - 000320   180    - 000420   190    - 000425
200    - 000432   210    - 000506   240    - 000606   250    - 000610
260    - 000613   270    - 000617   280    - 000624   290    - 000635
300    - 000645   320    - 000665   330    - 000715   10000  - 000732
10100  - 000736

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
ACC    - 000011   CON    - 001011   DA     - 001025   EX     - 000013
FV     - 001007   F0     - 001016   F1     - 001013   F2     - 001021
I      - 001027   IF     - 000004   INX    - 000010   IW     - 000012
K      - 001032   KDUMMY - 001031   L      - 001030   LINMIN - 000006
L0     - 001014   L1     - 001015   L2     - 001020   NLN    - 001026
NORM   - 001010   NPTS   - 000005   NVAR   - 000007   PHI    - 000002
PHI1   - 000003   R0     - 001012   R1     - 001017   R2     - 001022
R3     - 001023   R4     - 001024   WTS    - 000001   Y      - 000000

START OF CONSTANTS
000720

START OF TEMPORARIES
000742

START OF INDIRECTS
001001

UNUSED COMPILER SPACE
037600
```

```
                  SUBROUTINE SOLVER(A,B,DB,IE,ID,N,NR,IFL,D1,D2)
            C
            C      SOLVER IS A ROUTINE TO SOLVE A*DB = B  :  A AND B ARE SYM.
            C                                                B IS POSITIVE DEFINITE.
            C
000015            REAL A(IE,1),B(ID,1),DB(ID,1),X,D1
000015            INTEGER D2
000015            IFL = 0
000016            NP1 = N + 1
000020            D1 = 1.0
000021            D2 = 0
000022            DO 70 I=1,N
000024               DO 70 J=I,N
000025                  X = A(I,J)
000031                  K = I-1
000033        10        IF (K.LT.1) GO TO 20
000036                  X = X-A(I,K)*A(J,K)
000046                  K = K-1
000050                  GO TO 10
000050        20        CONTINUE
000050                  IF (I.NE.J) GO TO 60
000052                  D1 = D1*X
000054                  IF (X.EQ.0.0) GO TO 140
000055        30        IF (ABS(D1).LT.1.0) GO TO 40
000061                  D1 = D1*.0625
000062                  D2 = D2+4
000064                  GO TO 30
000065        40        IF (ABS(D1).GE..0625)GO TO 50
000071                  D1 = D1*16.0
000072                  D2 = D2-4
000074                  GO TO 40
000075        50        IF (X.LT.0.0) GO TO 150
000077                  A(NP1,I) = 1.0/SQRT(X)
000111                  GO TO 70
000112        60        A(J,I) = X*A(NP1,I)
000122        70  CONTINUE
000127            DO 130 J=1,NR
000130               DO 90 I=1,N
000131                  X = B(I,J)
000136                  K = I-1
000140        80        IF (K.LT.1) GO TO 90
000143                  X = X-A(I,K)*DB(K,J)
000154                  K=K-1
000155                  GO TO 80
```

```
000156        90        DB(I,J) = X*A(NP1,I)
000171              I = N
000171       100    IF (I.LT.1) GO TO 130
000174              X = DB(I,J)
000200              K = I+1
000202       110    IF (K.GT.N) GO TO 120
000205              X = X-A(K,I)*DB(K,J)
000215              K = K+1
000216              GO TO 110
000217       120    DB(I,J) = X*A(NP1,I)
000230              I = I-1
000231              GO TO 100
000231       130 CONTINUE
000234           RETURN
000234       140 D1 = 0.0
000235       150 IFL = 1
000237           RETURN
000237           END
```

```
SOLVER

SUBPROGRAM LENGTH
000261

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10    -  000033   20   -  000050   30   -  000055   40   -  000065
50    -  000075   60   -  000112   70   -  000122   80   -  000140
90    -  000156   100  -  000171   110  -  000202   120  -  000217
130   -  000231   140  -  000234   150  -  000235

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
D1    -  000002   D2   -  000003   I    -  000256   IFL  -  000001
J     -  000257   K    -  000260   NP1  -  000255   NR   -  000000
X     -  000254

START OF CONSTANTS
000241

START OF TEMPORARIES
000246

START OF INDIRECTS
000252

UNUSED COMPILER SPACE
041500
```

```
                        SUBROUTINE INVRTA(N,A,IE,IFL)
                  C
                  C      ON RETURN A HAS THE LOWER TRIANGULAR PORTION OF  I=A-INVERSE STOR
                  C      AS FOLLOWS  (FOR N = 3) :
                  C               A A A
                  C               I A A
                  C               I I A
                  C               I I I
                  C
                  C
000007                  REAL A(IE,1),X,Y,Z
000007                  IFL = 0
000007                  DO 40 I=1,N
000011                     I1 = I+1
000013                     DO 40 J=I,N
000014                        J1 = J+1
000016                        X = A(I,J)
000022                        K = I-1
000024            10          IF (K.LT.1) GO TO 20
000027                        X = X-A(J1,K)*A(I1,K)
000037                        K = K-1
000040                        GO TO 10
000041            20          IF (J.NE.I) GO TO 30
000043                        IF (X.LE.0.0) GO TO 90
000045                        Y = 1.0/SQRT(X)
000050                        A(I1,I) = Y
000057                        GO TO 40
000057            30          A(J1,I) = X*Y
000065            40    CONTINUE
000072                  NL = N-1
000074                  DO 60 I=1,NL
000075                     KL = I+1
000077                     DO 60 J=KL,N
000101                        Z = 0.0
000102                        J1 = J+1
000104                        K = J-1
000105            50          IF (K.LT.I) GO TO 60
000110                        Z = Z-A(J1,K)*A(K+1,I)
000120                        K = K-1
000122                        GO TO 50
000122            60          A(J1,I) = Z*A(J1,J)
000137                  J1 = N+1
000141                  DO 80 I=1,N
000142                     DO 80 J=I,N
```

```
000143                  Z = 0.0
000144                  KL = J+1
000146                  DO 70 K=KL,J1
000147       70            Z = Z+A(K,J)*A(K,I)
000163       80         A(KL,I) = Z
000174            RETURN
000174       90   IFL = 1
000175            RETURN
000176            END
```

```
INVPTA

SUBPROGRAM LENGTH
000224

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10      -  000024   20     -  000041   30     -  000057   40     -  000065
50      -  000105   60     -  000122   70     -  000147   90     -  000174

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
I       -  000215   I1     -  000216   J      -  000217   J1     -  000220
K       -  000221   KL     -  000223   NL     -  000222   X      -  000212
Y       -  000213   Z      -  000214

START OF CONSTANTS
000200

START OF TEMPORARIES
000203

START OF INDIRECTS
000207

UNUSED COMPILER SPACE
041700
```

```
NPAR    NVAR    NPTS    MAXITR    IW    LINMIN        EP             DAC
 11      11      *0       10       0      10      1.0000E-05     5.0000E-02

OPTIMAL POINT REACHED IN       8  ITERATIONS

      .15515072        .98462950       6.43276013       6.97716836      -6.16625657        .43606629

*ERROR DATA OUTPUT* SPECIFIED FIELD WIDTH ZERO*  FORMAT NO.12100
ERROR NUMBER   0071 DETECTED BY KODER   AT ADDRESS 020131
CALLED FROM OUTPTC  AT 020331
CALLED FROM         AT 012012

ERROR SUMMARY
    ERROR     TIMES
     0071      0001
```

NASA CR-132713

DISTRIBUTION LIST

NAS1-13683

	No. Copies
NASA Langley Research Center Hampton, VA 23665	
Attn: Report & Manuscript Control Office, Mail Stop 180A	1
Electronics Directorate, Mail Stop 117	1
Flight Instrumentation Division, Mail Stop 476	2
Measurements Research Branch, Mail Stop 473	1
John P. Reeder, Mail Stop 265	5
Robert T. Taylor, Mail Stop 246A	5
Dr. Roland L. Bowles, Mail Stop 125B	5
William D. Mace, Mail Stop 476	1
Edwin C. Loudriat, Mail Stop 152	3
Robert L. Kenimer, Mail Stop 473	1
Dr. Thomas M. Walsh, Mail Stop 473	1
John E. Hogge, Mail Stop 125	3
Elmer L. Goyetts, Jr., Mail Stop 257	3
Richard M. Hueschen, Mail Stop 473	25
Ames Research Center Moffett Field, CA 94035 Attn: Library, Mail Stop 202-3	1
NASA Flight Research Center Edwards, CA 93523 Attn: Library	1
NASA Goddard Space Flight Center Greenbelt, M 20771 Attn: Library	1
NASA Lyndon B. Johnson Space Center Houston, TX 77058 Attn: JM6/Library	1
Jet Propulsion Laboratory 4800 Oak Grove Drive Pasadena, CA 91103 Attn: Library, Mail 111-113	1
NASA John F. Kennedy Space Center Kennedy Space Center, FL 32899 Attn: Library, IS-DOC-I1	1
NASA Lewis Research Center 21000 Brookpark Road Cleveland, OH 44135 Attn: Library, Mail Stop 60-3	1

NASA CR-132713

DISTRIBUTION LIST

NAS1-13683

	No. Copies
NASA Marshall Space Flight Center Marshall Space Flight Center, AL 35812 Attn: Library, AS61L	1
National Aeronautics & Space Administration Washington, DC 20546 Attn: KSS-10/Library	1
MK/NASA Headquarters	1
NASA Scientific & Technical Information Facility 6571 Elkridge Landing Road Linthicum Heights, MD 21090	33 plus reproducible