U.S. Department
of Transportation

Federal Railroad
Administration

Lightweight Intermodal Flatcar Evaluation Program

Office of Research
and Development

Washington, D.C. 20590

Modal Analysis of Railcars, an Explanation and Applications Guide

FRA/ORD-80/07.I

Volume I

DECEMBER 1979

M. A. Kenworthy
J. I. Smith

Document is available to
the U.S. public through
the National Technical
Information Service,
Springfield, Virginia 22161

03 - Rail Vehicles &
Components

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FRA/ORD-80/07.I		

4. Title and Subtitle	
Lightweight Intermodal Flatcar Evaluation Program Vol. I Modal Analysis of Railcars An Explanation and Application Guide	5. Report Date December 1979
	6. Performing Organization Code
7. Author(s) M. A. Kenworthy, Ph.D. and J. I. Smith	8. Performing Organization Report No. DOT-FR-79-24
9. Performing Organization Name and Address Rail Transportation Engineering Division ENSCO, INC. 2560 Huntington Avenue Alexandria, VA 22303	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. DOT-FR-64113
12. Sponsoring Agency Name and Address U.S. DEPARTMENT OF TRANSPORTATION Federal Railroad Administration (RRD-12) 400 Seventh Street, S.W. Washington, DC 20590	13. Type of Report and Period Covered Technical Report
	14. Sponsoring Agency Code FRA/ORD-12

15. Supplementary Notes

Volume II, Test Results Report (Including Appendix A - Transducer Locations)
Volume III, Data Appendices B through F

16. Abstract

This report describes a methodology referred to as modal analysis which can be a very useful tool in the study of the dynamic behavior of any elastic or rigid body. The concepts of the technique are defined and illustrated with the use of a relatively simple one dimensional uniform beam. Following this, the general case is formulated and the procedure for its application is summarized. Next, practical considerations are discussed including both the analytical treatment, and the instrumentation and test design. The discussion of modal analysis concludes with the presentation of a program which used the methodology discussed in the report to make a comparative evaluation of two distinctly different rail vehicles. Conclusions are drawn and recommendations for future work are included.

Volume I: Modal Analysis of Railcars - An Explanation and Applications Guide
FRA/ORD-80/07.I
Volume II: Test Results Report (Including Appendix A - Transducer Locations)
FRA/ORD-80/07.II
Volume III: Data Appendices - Appendix B - Wear Measurements
FRA/ORD-80/07.III- Appendix C - Axle Power Spectral Densities
- Appendix D - Load Power Spectral Densities
- Appendix E - Carbody Power Spectral Densities
- Appendix F - Octave RMS Histories

17. Key Words	18. Distribution Statement
Railcar Dynamics Railcar Testing Railcar Instrumentation Intermodal Railcars TOFC/COFC Modal Analysis Railcar Lading/Dynamic Environment	Volumes I and II of this report are available to the public through the National Technical Information Service, Springfield, VA 22161. Volume III is available from the Sponsoring Agency on a loan basis.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	202	

Form DOT F 1700.7 (8-72)　Reproduction of completed page authorized

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT)

LIST OF TABLES

LIST OF SYMBOLS

A	modal coefficients (Section 2.0,Equation 12) elements of A-vector
A	amplitude
a	mode shape coefficient usually in first bending mode
a_m	modeled acceleration
B	mode shape function of bending modes
b	mode shape coefficient usually in first torsion mode
c	mode shape coefficient usually in second bending mode
D	probability density function (Section 4.0) or matrix determinate (Section 2.2)
d	mode shape coefficient usually in second torsion mode
d( )	elemental quantity of ( )
F	force
F	Nyquist frequency (Section 4.0)
f	frequency
G	cross and power spectral density function
g	unit of acceleration referred to as gravity (32.2 feet/second^2)
H	histogram
I	moment of inertia
i	general index
J	polar moment of inertia
j	general index
L	car length
L_{95}	ninety-fifth percentile level
L_{99}	ninety-ninth percentile level

M	moment or integral of mass
m	mass
N	number of points or samples
P	probability function
Q	a square matrix resulting from $X^T X$
R	residual
S	a square matrix (See Q)
T	mode shape function of torsion mode
t	time
W	bin width of histogram
X	transducer location matrix (Section 2.2, Equation 13), also data sample set when subscripted
x	principle axis in the direction of motion, or distance
y	principle horizontal axis perpendicular to the x-axis
z	principle vertical axis
x	longitudinal linear acceleration
y	lateral linear acceleration
z	vertical linear acceleration
$\ddot{x}_o$	longitudinal mode acceleration
$\ddot{y}_o$	lateral mode acceleration
$\ddot{z}_o$	vertical mode acceleration
z	general measured acceleration
$\tilde{z}$	general modeled acceleration from modal analysis
β	bending mode
$\Delta(\ )$	finite difference
θ	angular displacement about x-axis
$\ddot{\theta}_0$	roll mode acceleration
μ	mean

ρ mass density

σ standard deviation

τ torsion mode

ϕ angular displacement about the y-axis

$\ddot{\phi}_0$ pitch mode acceleration

ψ angular displacement about the z-axis

$\ddot{\psi}_0$ yaw mode acceleration

Subscripts

H horizontal, x-direction

L lateral, y-direction

i integer index

j integer index

k integer index

l integer index

m integer index

n integer index

s integer index

V vertical, z-direction

Special note of explanation:

Bending mode β_{abi}

a indicates the axis about which bending occurs
b indicates the direction of bending
i indicates the number of the mode within its family

Torsion mode τ_{ai}

a indicates the axis about which torsion occurs
i indicates the number of the mode within its family

ACKNOWLEDGEMENTS

The authors would like to express their appreciation for the invaluable contribution of Dr. Herb Weinstock of the Transportation Systems Center. Dr. Weinstock was solely responsible for the inception of the Lightweight Flatcar Evaluation Program and for the basic formulation of the modal analysis technique. His continued inputs and guidance during the subsequent development were instrumental in the successful implementation of the technique for the comparative evaluation of the three rail vehicles.

1.0 INTRODUCTION

The evaluation of the dynamic performance of newly-developed hardware, such as rail vehicles, plays an important role in the design cycle. This report describes a methodology, referred to as modal analysis, which can be used to make a meaningful comparison between rail vehicles.

The modal analysis technique discussed in this report was adapted and refined as in integral part of the Lightweight Flatcar Evaluation (LWFC) Program conducted by the Federal Railroad Administration (FRA) in cooperation with a number of industry participants. The list of industry participants includes:

- American Steel Foundries
- National Castings Division, Midland-Ross Corporation
- Pullman Standard
- Pullman Transport Leasing
- Santa Fe Railway Company
- Trailer Train Company

The program was designed to evaluate the performance of two prototype skelton flatcars, referred to as lightweight flatcars, as compared to a conventional TTAX flatcar. Because of radical structural differences between the lightweight and conventional flatcars, it was necessary to develop a methodology which would provide the means for making a meaningful comparison between the flatcars. This was accomplished through the use of modal analysis.

Modal analysis is basically the transformation of a set of measured linear accelerations to a set of generalized accelerations called modes. These generalized accelerations or modes offer a number of advantages. First, the modes are individually easy to visualize. For example, a set of modes may include bounce (linear vertical displacement), pitch (angular displacement), and first or simple bending for a quasi-one-dimensional body. Each of these is easy to visualize and can be quite simply related to design parameters such as spring stiffness and mass. In contrast, three independent measurements of linear acceleration on the same body would not readily reveal these relations.

Perhaps the most compelling argument, however, for the use of modal analysis is its ability to provide a clear and objective basis for comparison. For example, in the LWFC Program the primary load carrying structural members of the lightweight flatcars were distributed laterally outboard while the primary structure of the TTAX flatcar was concentrated along the car centerline. Thus, any single measurement of acceleration would favor one car and any set would provide at best a confusing picture. By transforming the set of measured accelerations to modes, it became relatively easy to make a meaningful comparison by modes.

One advantage of the use of modal analysis, which was not explored in the LWFC Program, is its ability to provide accurate estimates of acceleration at points on a vehicle other than those where measurements were made. To do this, all modes which make significant contributions to the acceleration environment must be identified and then used to predict the acceleration field at these other points of interest.

Finally, modal analysis offers the potential to examine elastic mode shapes. Knowledge of mode shape amplitude and frequency obtained through modal analysis may provide valuable information to fatigue life cycle analysis. Modal analysis presents in compact form the results of realistic dynamic tests such as service operation, for the fatigue analysis.

The details of the methodology or application of modal analysis are covered in three steps. First, the general application of modal analysis to an arbitrary body is presented. Here the concept of modes as a generalized or global representation are examined. Following the general discussion, use is made of a quasi-one-dimensional example to illustrate the mechanics of modal analysis. This is followed by a statement of the general three-dimensional case with emphasis on basic considerations and a summary of the procedure as applied to an arbitrary body.

Secondly, practical considerations in the application of modal analysis are discussed. Two primary areas of consideration in practical applications of modal analysis are discussed. First, the concept of an overall system being comprised of simple subsystems is presented. For this purpose use is made of a rail car. Second, an examination of the dynamic or acceleration environment is made in order to define transducer requirements.

Thirdly, the LWFC Program mentioned earlier is used as an actual example of modal analysis. Here specific design parameters are given and sample test results displayed. These sample results illustrate data formats which can be effectively used in displaying results.

Finally, conclusions and recommendations are made based on the experience obtained during the LWFC Program. These include aspects of the application of the modal analysis methodology and such things as hardware design. The conclusions and recommendations are intended for future work in practical application of modal analysis.

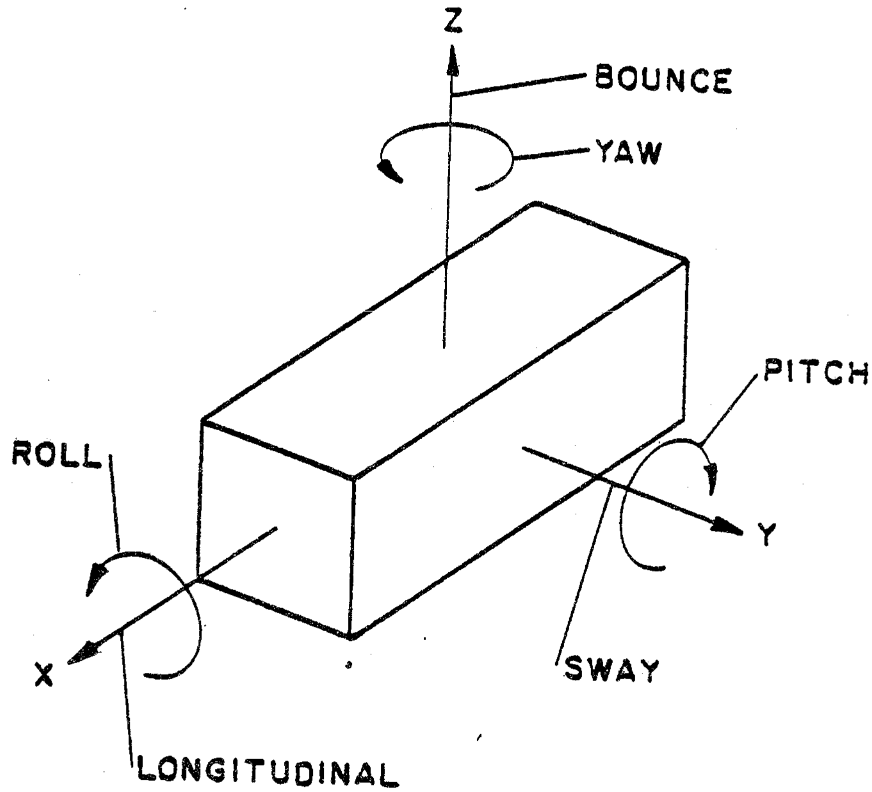
2.0 METHODOLOGY

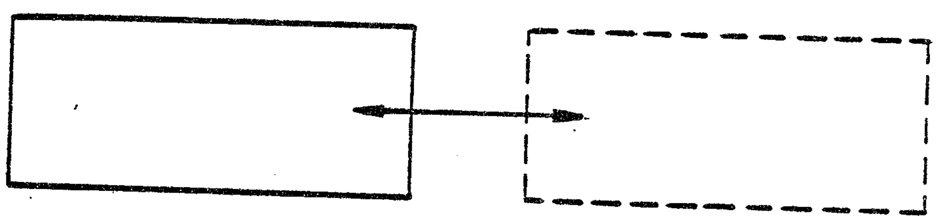
2.1 GENERAL CONCEPTS

Modal analysis is kinematic in nature as opposed to more deterministic types of analyses generally involving the solution of Newtonian equations of motion which are non-linear partial differential equations. That is, modal analysis is simply a transformation of the description of motion from a set of local coordinates to a set of general coordinates which give a global description of motion. This global description has a number of distinct advantages; however, before discussing the advantages of modal analysis, it is necessary to introduce the basic concept, i.e., the mode.

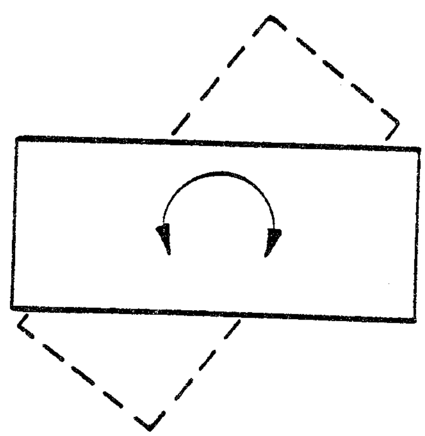
A mode is the description of a specific type of movement of a body. For example, consider a body which has been constrained such that it translates linearly along a single axis. In this case, a single quantity or mode is necessary and sufficient to describe its motion entirely, although any number of measurements can be made on the body. Going one step further, suppose the body is allowed to rotate in the plane perpendicular to the axis of translation. In this instance there are two modes, one linear and one rotational mode, required to completely describe this motion at any point on the body. Again, accelerations can be measured at any point on the body, but it is easily shown that these are simple linear combinations of the two modes. That is, a local measurement of acceleration can be expressed as a linear function of modes. This property is called the principle of superposition which allows the motion of a body to be broken down into modes.

In general, there are three linear modes and three rotational modes necessary to describe the motion of a body in a three-dimensional space (see Figure 2-1). These are referred to as





TRANSLATION



ROTATION

Figure 2-1. Linear and Rotational Modes

The rigid-body modes since it is assumed that every point within the body retains its spatial relationship with respect to every other point in the body. In other words, the body is infinitely rigid. In many applications, the six rigid-body modes are sufficient to describe the required or measured motion.

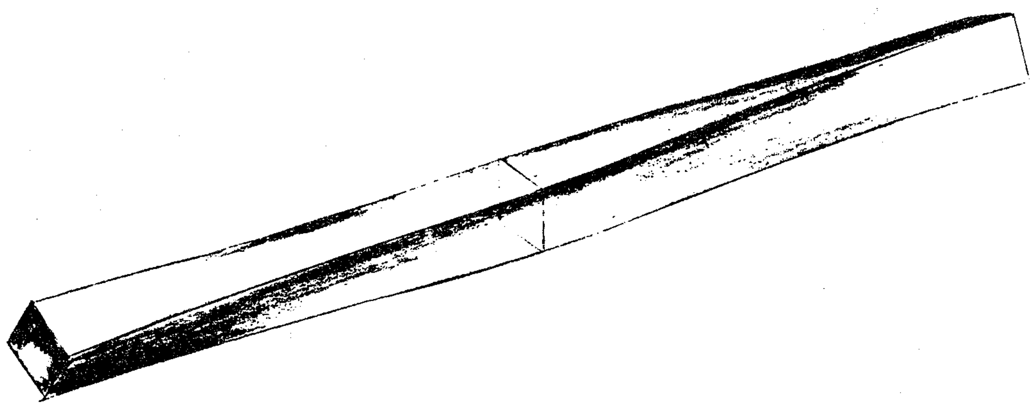
There are, however, a significant number of practical structures which deform elastically, i.e., points within the body move with respect to other points within the body. There are three basic types of elastic-body motion: bending, twisting, and extension (compression). Figure 2-2 illustrates bending and twisting motion. For the purposes of the present discussion, extension/compression will be omitted.

One final concept must be brought out before returning to the discussion of the advantages and applications of modal analysis. Each elastic mode is made up of a family of modes which may be thought of as harmonics. Figure 2-3 shows the first three modes of one of the bending mode families and Figure 2-4 shows the same thing for one of the twist mode families more usually referred to as torsion.

With these basic concepts, attention is turned to the output signal of the linear accelerometer. This transducer possesses a number of virtues which suit it to the study of vehicle dynamics, such as its relatively low cost and simple principle of operation.

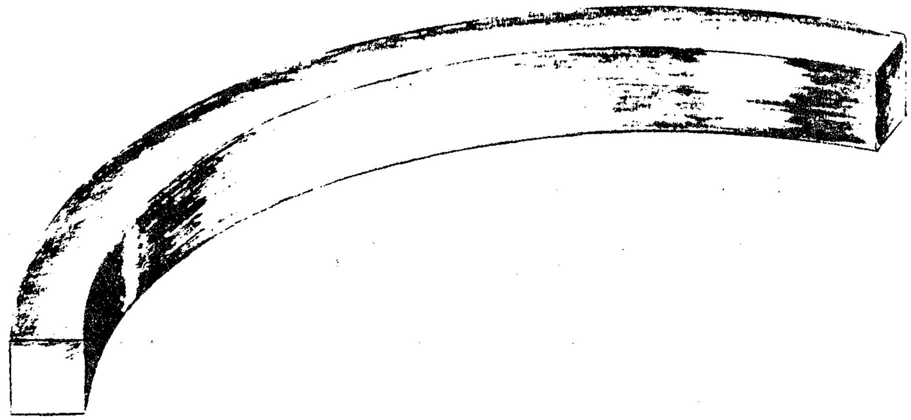
One of the accelerometer's primary advantages is that its measurements are made with respect to inertial space and hence require no mechanical reference. Again, this makes the accelerometer an ideal transducer for the study of vehicle dynamics; however, this does cause some difficulty visualizing the data for the purpose of analysis. Modal analysis has proven to be a useful technique in visualizing the output

BENDING

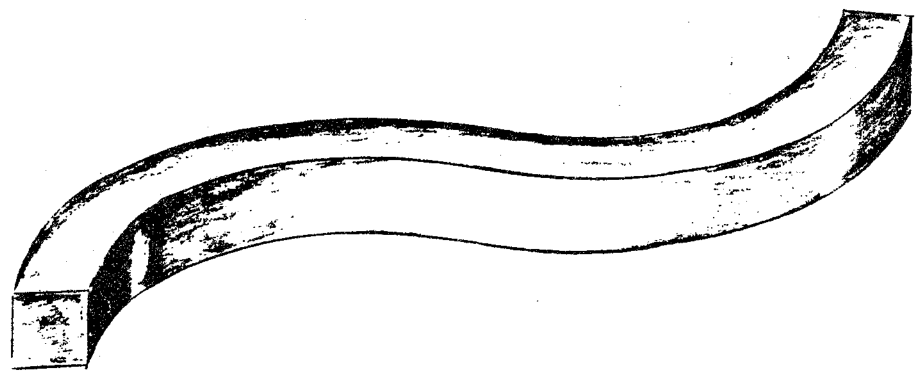


TORSION

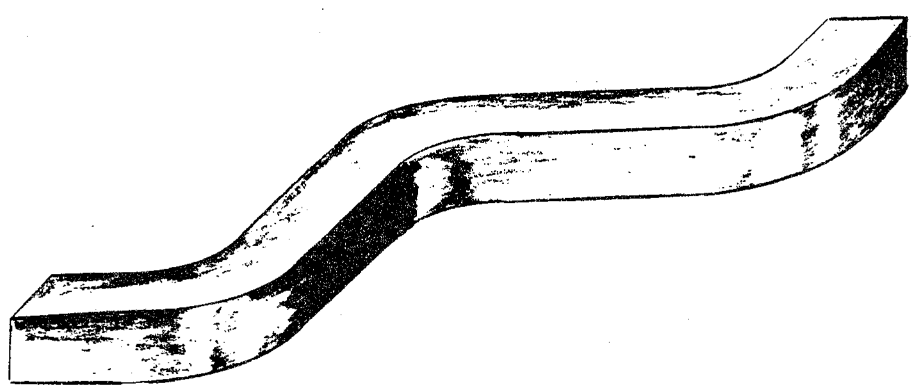
Figure 2-2. Elastic-Body Modes



FIRST BENDING

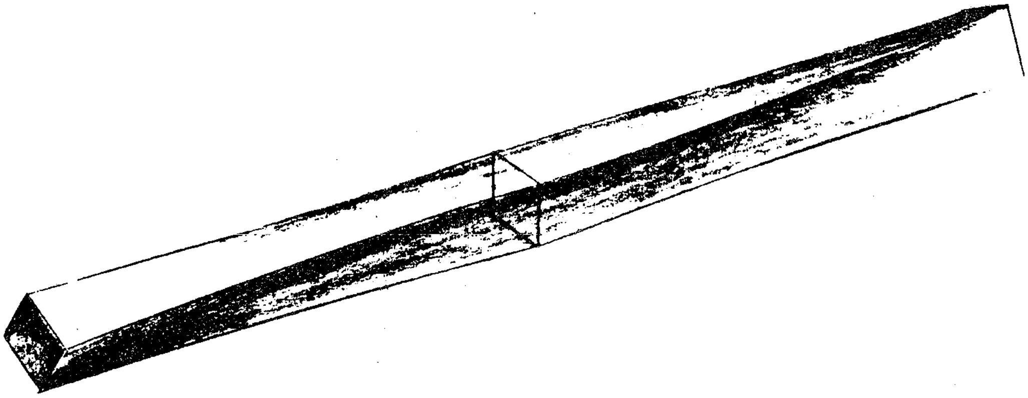


SECOND BENDING

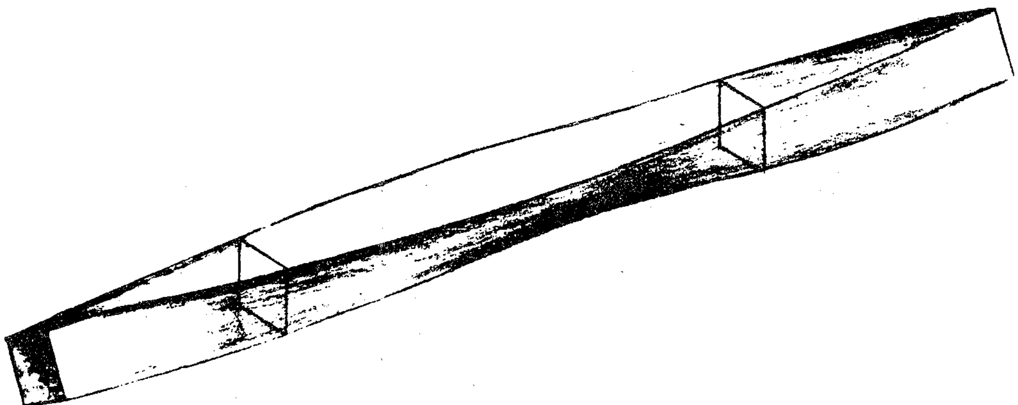


THIRD BENDING

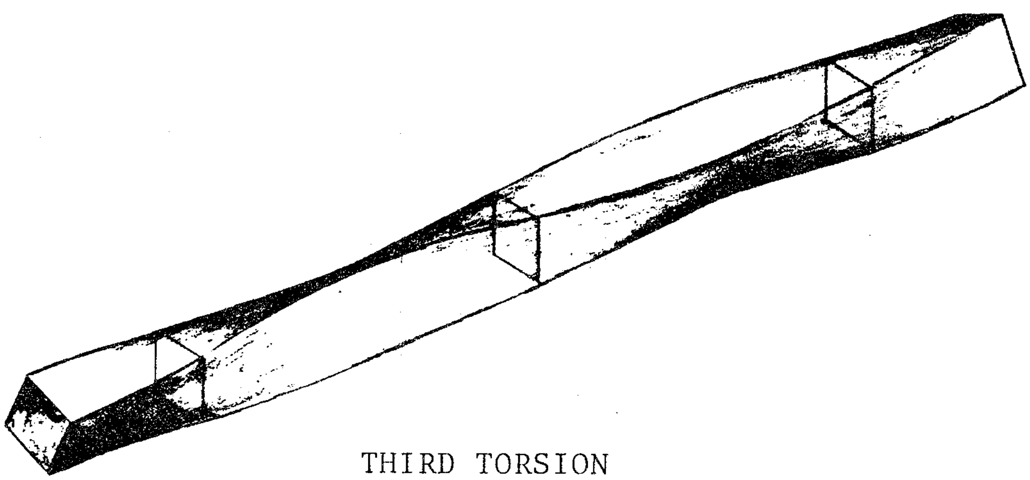
Figure 2-3. Bending Mode Family



FIRST TORSION



SECOND TORSION



THIRD TORSION

Figure 2-4. Torsion Mode Family

of an array of transducers, perhaps several dozen, thereby reducing the difficulty of analyzing the accelerometer output. That is, a set of local measurements of linear acceleration when transformed to modes are readily visualized and understood by a wide range of the technical community.

The use of modes provides two additional advantages. First, in situations where comparative evaluation is desired, modal analysis provides an objective means for making a comparison. For example, in the evaluation of a prototype system, such as a rail vehicle, it would be reasonable to compare the prototype with a predessor or conventional vehicle which has logged extensive service. In this case, it would be unlikely that any single measurement of acceleration on two structurally different vehicles would provide sufficient information for a meaningful comparison. Even an array of accelerometers placed in geometrically similar locations on both vehicles would provide a less than clear basis for comparision. However, upon transformation of these local linear accelerations to modes, the comparison becomes much clearer. For example, the vertical acceleration mode, called bounce, may be compared between vehicles giving some insight into the relative effectiveness of the suspension systems. Other analytical approaches may also be used to provide further insight.

Finally, the use of modal analysis has the virtue of being able to provide a reasonably accurate estimate of acceleration at any point on the body once a finite set of measurements have been made. This is accomplished by transforming the linear accelerations to mode accelerations and making use of the principle of superposition. That is, a general expression for the acceleration at any point on the body may be written as a linear function of the modes and position as stated earlier. This expression can then be evaluated at any of the infinity of points which comprise the body. This approach has numerous applications such as identifying points of low acceleration to provide locations to fasten lading.

Even a purely kinematic treatment of a three-dimensional body using the six rigid-body modes and the numerous (in a sense infinite) elastic-body modes is not a casual undertaking. However, the reader should not despair. The important features of the method are illustrated in what follows through the use of a quasi-one-dimensional example.

2.2 AN ILLUSTRATIVE EXAMPLE

For the purpose of introducing the basic principles and considerations in the application of modal analysis, a quasi-one-dimensional body will be used as an example. The example deals with a body or uniform beam which is characterized by an acceleration field that varies in one direction only, i.e., is one-dimensional. The quantity which is permitted to vary is an acceleration perpendicular to the primary axis, hence the term quasi-one-dimensional.

In this example, only four modes have been selected; bounce, pitch, and first and second vertical bending, as shown in Figure 2-5. That is, based on prior knowledge or analysis, it has been determined or estimated that this set of modes is sufficient to adequately model the acceleration field to be measured. After calculating the modes, this hypothesis will be checked. The first step, however, is the measurement of acceleration at a number of judiciously selected points on the body.

The next step is to write an equation to describe the response of a particular (but unspecified) accelerometer to all of the several varieties of modal components of acceleration which

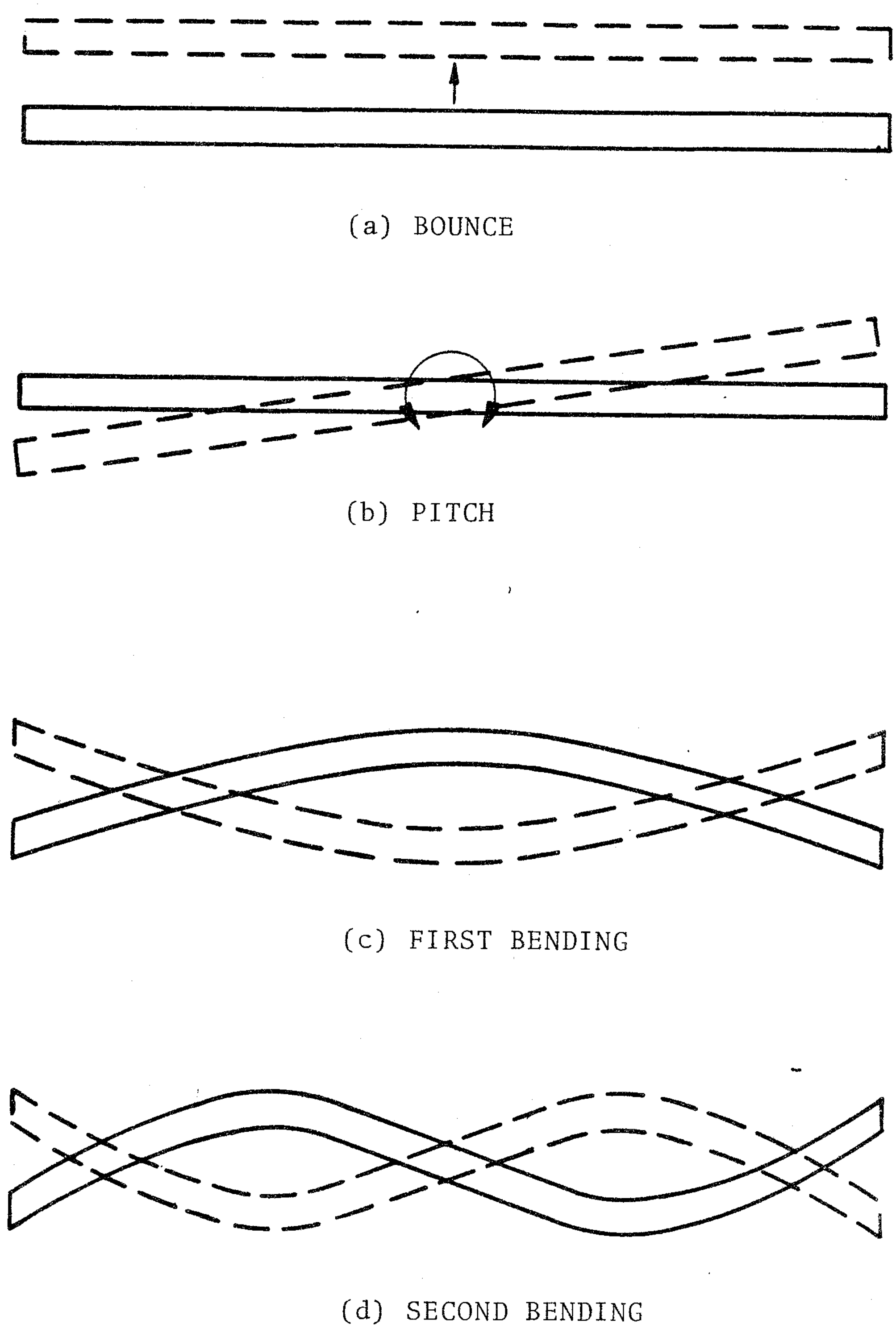


Figure 2-5. Four Modes Used for a Uniform One-Dimensional Beam

are of ultimate interest, in this case four. That is, we proceed as if we knew the rigid-body translation component, rigid-body rotation component and the two components due to first and second modes of bending. It is a fairly simple matter to sum the effects of these modes at the location of a particular accelerometer. Then, since the accelerometer was unspecified, what is obtained is an equation (Equation 1) that can be made to apply to every accelerometer.

$$\ddot{z}(x,t) = \ddot{z}_o(t) - \ddot{\phi}_o(t)x + \beta_{yz1}(t)B_{yz1}(x)$$

$$+ \beta_{yz2}(t)B_{yz2}(x) \qquad (1)$$

Equation (1) represents the measurement of a linear acceleration, $\ddot{z}$, at an arbitrary distance, x, from the center of mass as a function of time, t. The bounce and pitch modes are denoted $\ddot{z}_o(t)$ and $\ddot{\phi}_o(t)$, respectively and are also functions of time. Note that because $\ddot{\phi}_o(t)$ is an angular acceleration, it is necessary to multiply it by a distance, in this case x. The last two terms of Equation (1) represent the contributions of the first and second bending modes to vertical acceleration. Each of these terms is made up of two parts. The first factor of each term, β , represents the mode amplitude. The first subscript indicates the axis about which the bending occurs, in this case the y-axis which passes through the origin perpendicular to the xz-plane. The second subscript indicates the direction of contribution of acceleration, in this case the z-direction. The last subscript represents the elastic mode number within its respective family, in this case the first and second. The second factor, B, is the function which describes the mode shape into which the beam distorts itself when vibrating in the various modes. The function B is referred to as the mode-shape function and the system of subscripts is identical to that of the mode amplitude itself. Figure 2-5, c and d,depicts the mode-shape function, B.

The beam relaxes from the solid curve shape through a continuous set of similar shapes of diminishing amplitude until it becomes a straight line; it then continues past the straight line position through an expanding series of similar but opposite shapes until it reaches the dashed line shape; completing one-half of a cycle of modal vibration.

In Equation (1) the mode-shape functions are expressed as arbitrary functions of x. Although there may exist some more efficient functional expression depending on the body to be modeled, truncated polynomials are used here. In fact, at this point in the discussion the mode shape functions are expressed as binomials (Equations 2 and 3) further simplifying matters. In the event it is necessary to use higher order polynomials, some additional calculations are necessary to evaluate the mode shape. This will be taken up later.

$$B_{yz1} = 1 + b_2x^2 \tag{2}$$

$$B_{yz2} = x + b_3x^3 \tag{3}$$

The lower case b's are called the mode shape coefficients. Note that the leading coefficients b_o and b_1 have been implicitly set equal to unity. This provides a normalized mode shape rather than an exact dimensional displacement function.

Equations (2) and (3) are then substituted into Equation (1) to yield:

$$\begin{aligned} \ddot{z}(x,t) &= \ddot{z}_o(t) - \ddot{\phi}_o(t)x \\ &\quad + \beta_{yz1}(t) + \beta_{yz1}(t)b_2x^2 \\ &\quad + \beta_{yz2}(t)x + \beta_{yz2}(t)b_3x^3 \end{aligned} \tag{4}$$

Collecting terms in like powers of x;

$$\ddot{z}(x,t) = \left[\ddot{z}_o(t) + \beta_{yz1}(t)\right] + \left[\beta_{yz2}(t) - \ddot{\phi}_o(t)\right]x$$
$$+ \left[\beta_{yz1}(t)b_2\right]x^2 + \left[\beta_{yz2}(t)b_3\right]x^3 \qquad (5)$$

The bracketed quantities are defined as follows:

$$A_o \equiv \left[\ddot{z}_o(t) + \beta_{yz1}(t)\right] \qquad (6)$$

$$A_1 \equiv \left[\beta_{yz2}(t) - \ddot{\phi}_o(t)\right] \qquad (7)$$

$$A_2 \equiv \left[\beta_{yz1}(t)b_2\right] \qquad (8)$$

$$A_3 \equiv \left[\beta_{yz2}(t)b_3\right]. \qquad (9)$$

Now Equation (5) can be rewritten more compactly:

$$\ddot{z}(x,t) = A_o + A_1x + A_2x^2 + A_3x^3 \qquad (10)$$

The left hand side of Equation (10), $\ddot{z}(x,t)$, is a measured acceleration, that is, the output of an accelerometer. The quantity x on the right hand side is the known location of that accelerometer. By using four accelerometers the four unknown coefficients; A_o, A_1, A_2, and A_3, can be evaluated. Figure 2-6 represents the four measurement locations on the beam.

Equation (10) applies to each of the accelerometers in Figure 2-6 forming a set of four simultaneous equations in four unknowns as follows:

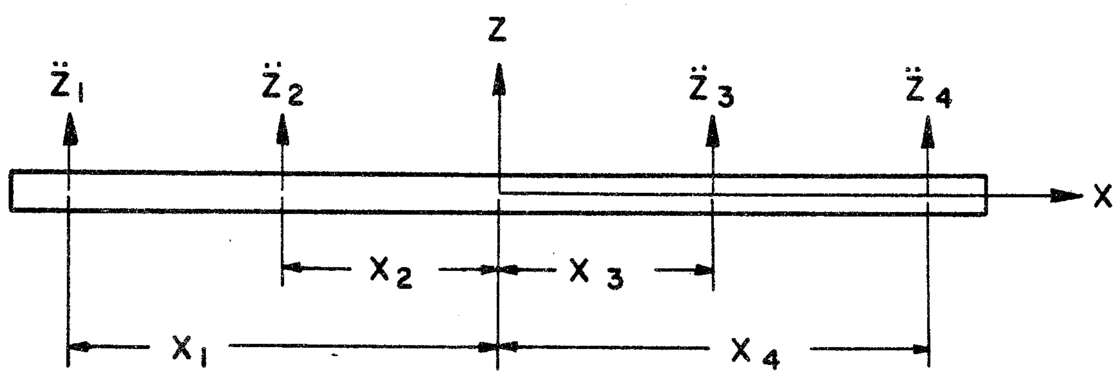


Figure 2-6. Measurement Locations on the Beam

$$\ddot{z}_1(t) = A_o + A_1x_1 + A_2x_1^2 + A_3x_1^3$$

$$\ddot{z}_2(t) = A_o + A_1x_2 + A_2x_2^2 + A_3x_2^3$$

$$\ddot{z}_3(t) = A_o + A_1x_3 + A_2x_3^2 + A_3x_3^3$$

$$\ddot{z}_4(t) = A_o + A_1x_4 + A_2x_4^2 + A_3x_4^3 \quad , \tag{11}$$

which can be solved for the four unknowns; A_o through A_3.

In matrix notation Equation (11) becomes

$$\{Z\} = [X]\{A\}, \tag{12}$$

in which $\{Z\} \equiv \begin{Bmatrix} \ddot{z}_1(t) \\ \ddot{z}_2(t) \\ \ddot{z}_3(t) \\ \ddot{z}_4(t) \end{Bmatrix} \qquad \{A\} \equiv \begin{Bmatrix} A_o \\ A_1 \\ A_2 \\ A_3 \end{Bmatrix}$

$$[X] \equiv \begin{bmatrix} 1, & x_i, & x_i^2, & x_i^3 \end{bmatrix}$$

where i = 1 to 4, resulting in a 4 x 4 matrix as shown in Equation (13)

$$[X] = \begin{bmatrix} 1, & x_1, & x_1^2, & x_1^3 \\ 1, & x_2, & x_2^2, & x_2^3 \\ 1, & x_3, & x_3^2, & x_3^3 \\ 1, & x_4, & x_4^2, & x_4^3 \end{bmatrix} \tag{13}$$

Proceeding in matrix notation, the solution of Equation (11) may be formally indicated as follows:

$$[X]^T[X]\{A\} = [X]^T\{Z\} \tag{14}$$

where $[X]^T$ is the transpose matrix of $[X]$.

Let $$[Q] \equiv [X]^T[X] \tag{15}$$

then Equation (12) becomes

$$[Q]\{A\} = [X]^T\{Z\}. \tag{16}$$

$$[Q]^{-1}[Q]\{A\} = [Q]^{-1}[X]^T\{Z\} \tag{17}$$

$$[I]\{A\} = [Q]^{-1}[X]^T[Z] \tag{18}$$

where $[I]$ is the identity matrix.

$$\{A\} = [Q]^{-1}[X]^T[Z] \tag{19}$$

In any event the A coefficients are now known in terms of the $\ddot{z}$'s and x's. The quantities of ultimate interest, however, are the modal accelerations related to the A coefficients by Equations (6) through (9). Inspection of Equations (6) and (7) quickly reveals that additional relationships are required to decouple the modes in A_o and A_1.

The necessary additional relationships needed to decouple the modal accelerations are derived from equilibrium considerations. That is, the net force due to the elastic accelerations depicted in Figure 2-5 must be zero because the acceleration of the center of mass of the body is zero and the total external force is likewise zero. Newton's second law is used to express this. The sum of the elemental force, dF, over the body must be zero; therefore, the product of the acceleration, $\ddot{z}_{\beta 1}$, and the elemental mass, dm, must be zero.

$$\int_{-L/2}^{L/2} dF = \int_{-L/2}^{L/2} \ddot{z}_{\beta 1}(x,t)dm = 0 \tag{20}$$

where $\ddot{z}_{\beta 1}$ is the acceleration attributed to the first bending mode.

From inspection of Equation (1)

$$\ddot{z}_{\beta 1}(x,t) \equiv \beta_{yz1}(t)B_{yz1}(x). \tag{21}$$

Also $$dm = \rho(x)dx \tag{22}$$

where $\rho(x)$ is the mass density distribution function. Thus, Equation (20) becomes

$$\int_{-L/2}^{L/2} \beta_{yz1}(t)B_{yz1}(x)\rho(x)dx = 0. \tag{23}$$

Because $\beta_{yz1}(t)$ is not a function of x it can be brought outside the integral sign and divided out of the equation to yield:

$$\int_{-L/2}^{L/2} B_{yz1}(x)\rho(x)dx = 0 \tag{24}$$

Using Equation (2) and the fact that for a uniform beam $\rho(x)$ equals constant, Equation (24) becomes:

$$\int_{-L/2}^{L/2} (1 + b_2x^2)dx = 0 \tag{25}$$

Upon integrating Equation (25),

$$\left[x + b_2 \frac{x^3}{3}\right]_{-L/2}^{L/2} = \left(L + b_2 \frac{L^3}{12}\right) = 0 \tag{26}$$

or

$$b_2 = -\frac{12}{L^2} . \tag{27}$$

Substituting this value into Equation (8) gives

$$\beta_{yz1}(t) = -\frac{L^2}{12} A_2 \tag{28}$$

Then substituting Equation (28) into Equation (6) produces,

$$\ddot{z}_o(t) = A_o + \frac{L^2}{12} A_2 . \tag{29}$$

Proceeding in analogous fashion the modal components associated with the second bending elastic mode are uncoupled. Force equilibrium, however, is assured by symmetry; therefore, equilibrium of moments of forces is assumed, and are described by these equations.

$$\int_{-L/2}^{L/2} d\tau = \int_{-L/2}^{L/2} \left[\frac{\ddot{z}_{\beta 2}(x,t)}{x}\right] x^2 \rho(x)dx \equiv 0 \tag{30}$$

Since $d\tau = xdF$, Equation (3) becomes:

$$\int_{-L/2}^{L/2} xdF = \int_{-L/2}^{L/2} \left[\frac{\beta_{yz2}(t)B_{yz2}(x)}{x}\right] x^2 \rho(x)dx = 0 \tag{31}$$

$$\int_{-L/2}^{L/2} \left[B_{yz2}(x)\right] x \ \rho(x)dx = 0 \tag{32}$$

Since $\rho(x)$ is a constant, equation (32) becomes,

$$\int_{-L/2}^{L/2} \left[x + b_3 x^3\right] x dx = 0 \tag{33}$$

$$= \left[\frac{x^3}{3} + \frac{b_3 x^5}{5}\right]_{-L/2}^{L/2} = \frac{L^3}{12} + b_3 \frac{L^5}{80} = 0 \tag{34}$$

$$b_3 = - \frac{20}{3L^2} \tag{35}$$

Upon substituting Equation (35) into Equation (9), Equation (36) is obtained

$$\beta_{yz2}(t) = \frac{-3}{20} L^2 A_3 \tag{36}$$

which, when substituted into Equation (7) yields

$$\ddot{\phi}(t) = - \frac{3}{20} L^2 A_3 - A_1 \tag{37}$$

Thus, all of the modal components of acceleration are reduced to numbers in terms of the already evaluated A coefficients. In addition, the mode shape coefficients have been evaluated.

As mentioned earlier, the use of binomial expressions is a somewhat special case, although, quite often in practical applications, it is sufficient. In those cases where higher order polynomials are required and the mode shape itself is of interest, a slightly different technique is required to evaluate the mode shape coefficients. Upon examining Equation (8) and (9), it is apparent that certain elements of the A-vector are products of the mode and mode shape coefficient. Ideally, the mode shape coefficient is a constant independent of time and, therefore, can be easily obtained by simply dividing the A-element by the mode in question. This particular approach has two minor flaws.

First, because the mode is itself a function of time with a zero mean it must pass through zero quite often. Thus, even though the A-element should simultaneously pass through zero, the ill-defined situation of division by zero arises. This could, of course, be circumvented through the use of conditional logic, but this in turn would pose other problems. Secondly, as explained in Section 4.0, the assumption of independence of time is not entirely satisfied in reality. For these reasons the following technique is used.

The A-element in Equations (8) and (9) is Fourier Transformed to yield

$$A(f) = b\beta(f) \tag{38}$$

Note, the subscripts have been dropped for simplicity. Both sides of Equation (38) are multiplied by the complex conjugate of the mode $\beta^*(f)$,

$$\beta^*(f)A(f) = b\beta^*(f)\beta(f). \tag{39}$$

The complex conjugate multiplication yields the cross and power spectral densities $G_{\beta A}(f)$ and $G_{\beta\beta}(f)$, respectively. Thus, the mode shape coefficient may be written as

$$b = G_{\beta A}(f)/G_{\beta\beta}(f). \tag{40}$$

Equation (40) expresses the mode shape as the ratio of two quantities which are functions of frequency. In the application of this technique it is, therefore, necessary to perform a point-by-point division in the frequency domain, thus expressing b as a single function of frequency. It should be pointed out that the phase angle of any power spectral density is identically zero and, therefore, any phase dependence or conversely imaginary component is introduced through the cross spectral density. This point will be pursued further in the following section which describes an actual application.

Recall that at the outset of this example, a set of four modes were selected. It is now necessary to evaluate this selection. In Equation (11) the number of equations was chosen exactly equal to the number of unknowns. This is the minimum number of equations needed to obtain a solution. There is another consideration, however, which forms a key ingredient of the modal analysis technique. If the number of accelerometers and hence the number of equations is made greater than the number of unknowns, the redundant information can be used to provide a quantitative check on the adequacy of the selection of the first and second bending modes used in the analysis. That is, when the modal analysis is completed the results are used to predict the acceleration at the exact location of an actual accelerometer. The difference between the predicted and the actually measured acceleration is termed a residual. When the residual is sufficiently small, it can be assumed that the choice of modes and mode shape representation functions are adequate.

A mathematical procedure called variously the method of least squares or linear regression can be used to solve redundant sets of equations. It can be shown that the multiplication of the X-matrix by its transpose as indicated in Equation (14) has the effect of converting the rectangular matrix of a redundant equation set into a smaller square matrix with a unique solution. Thus, the least-squares function fitting procedure is automatically implemented by the mechanics of matrix solution.

The choice of the location of the individual linear accelerometers is arbitrary, although with one primary consideration: the determinant, D of the product of $[X]^T$ and $[X]$, must not be zero or near zero. In matrix notation:

$$D \equiv |X^T X| \neq 0 \tag{38}$$

This is not a difficult condition to avoid; it is merely required that the determinant be evaluated for the particular array configuration chosen. Relocation of one or more accelerometers will correct the computational difficulty caused by a near-zero determinant.

In general, the number of accelerometers must be greater than or equal to the number of A coefficients to be evaluated. That is, there must be at least as many accelerometers as there are terms on the right-hand side of Equation (10). Experience to date suggests increasing this number by 25 to 50 percent to provide the redundancy necessary for the evaluation of the goodness-of-fit by computing the residuals. An analysis of the effects of redundancy has not, as yet, been undertaken.

The measurements of acceleration are then made under prescribed test conditions. The A-matrix is obtained and the decoupling relations are evaluated, which in general requires a knowledge of the mass and moment of inertia distributions. The decoupling relations provide sufficient additional information to enable solving explicitly for the modal components and mode-shape coefficients.

The acceleration field at the specific location of each actual accelerometer is then reconstructed from the derived modal components of acceleration using an equation such as Equation (5). Residuals are then computed and an appraisal is made of the adequacy of the initial choice of modal representation. If necessary, the modal representation is revised and the data recomputed. The process ends when the residuals are acceptably small.

2.3 GENERAL CASE

The foregoing example has illustrated the procedures and important considerations in the application of modal analysis. The remaining task is to extend this to the general case of a three-dimensional body capable of all of the possible modes of acceleration.

As was mentioned earlier, each elastic component in principle involves an infinite number of modes. The first two such modes for the bending case have been illustrated. Fortunately, in most practical considerations of typical structures, only two or three of the lowest-order modes need to be considered. The question of exactly how many to include in the analysis is a matter for a case-by-case determination. The technique of computing residuals enables the confirmation of the choice of modes. In the general case, prior to the mode elimination process, there exist two infinities in bending and one in torsion about each of the three coordinate axes or nine infinities in all.

The equation to describe the local acceleration in the x-direction in terms of the various modal accelerations is derived with the aid of Figures 2-1 and 2-2.

$$\ddot{x}(y,z,t) = \ddot{x}_o(t) + \ddot{\phi}_o(t)z - \ddot{\psi}_o(t)y$$

$$+ \sum_{i=1}^{\infty} \beta_{yxi}(t)B_{yxi}(z) + \sum_{j=1}^{\infty} \beta_{zxj}(t)B_{zxj}(y)$$

$$+ \sum_{k=1}^{\infty} \tau_{zk}(t)T_{zk}(z)y + \sum_{\ell=1}^{\infty} \tau_{y\ell}(t)T_{y\ell}(y)z \tag{41}$$

Proceeding in similar fashion for the y- and z-directions:

$$\ddot{y}(x,z,t) = \ddot{y}_o(t) - \ddot{\theta}_o(t)z + \ddot{\psi}_o(t)x$$

$$+ \sum_{m=1}^{\infty} \beta_{xym}(t)B_{xym}(z) + \sum_{n=1}^{\infty} \beta_{zyn}(t)B_{zyn}(x)$$

$$+ \sum_{k=1}^{\infty} \tau_{zk}(t)T_{zk}(z)x - \sum_{q=1}^{\infty} \tau_{xq}(t)T_{xq}(x)z \tag{42}$$

$$\ddot{z}(x,y,t) = \ddot{z}_o(t) + \ddot{\theta}_o(t)y - \ddot{\phi}_o(t)x$$

$$+ \sum_{r=1}^{\infty} \beta_{xzr}(t)B_{xzr}(y) + \sum_{s=1}^{\infty} \beta_{yzs}(t)B_{yzs}(x)$$

$$+ \sum_{q=1}^{\infty} \tau_{xq}(t)T_{xq}(x)y - \sum_{\ell=1}^{\infty} \tau_{y\ell}(t)T_{y\ell}(y)x \tag{43}$$

Equation (41) through (43) express linear acceleration of a general body which is capable of not only the six rigid-body modes but the nine infinities of elastic-body modes. In reality, however, it is not actually necessary or possible to deal with even one infinity of equations. Practical considerations will, in most cases, reduce the task of solution to a scope more nearly in-line with the capabilities of an engineering analyst. The method of application of the general case and the subsequent solution are then identical to that of the preceeding one-dimensional illustration. The topic of practical considerations is taken up in the following section (3.0) and illustrated in Section 4.0.

However, before concluding the discussion of the general case of modal analysis one final point should be made. Recall that in the one-dimensional example, only two elastic modes from the vertical bending family were considered. These were specifically chosen such that one was an even function (first bending) and one was an odd function (second bending). In the event that it becomes necessary to use more than one even or odd mode of a given elastic-mode family, one further difficulty arises. This difficulty is similar to the coupling that takes place between the rigid-body modes and the elastic-body modes.

In the case of two similar modes within a given family, the mode shape coefficients will couple. These coefficients can be decoupled using the condition of orthogonality since modes are by definition orthogonal. The mathematical statement of this is simply

$$\int_{-L/2}^{L/2} B_n(x) B_m(x) dx = 0 \qquad n \neq m \qquad (44)$$

where B_n and B_m are the mode shape functions in question either both even or both odd.

The solution of this equation, however, involves non-linear algebra as opposed to the relatively simple linear algebra heretofore encountered. Also, the number of equations, in addition to those previously discussed, is equal to (p - 1)! where p is the number of similar modes within the family. It should be noted that the level of complexity will increase dramatically if more than one odd or even mode from a given family is included in the set of modes selected.

2.4 SUMMARY OF METHODOLOGY

The preceeding sections have presented an illustrative example of the application of the procedure to a relatively simple case and a statement of the general case. Basically, the methodology is the same for the general case as it was for the example. Figure 2-7 summarizes the flow of this procedure.

The process of modal analysis begins with a study of the body or system. Through a combination of analysis, previous experience, simulation and sound engineering judgement, a finite set of modes is selected which will be used to model the motion of the body. The next step is to write the expressions for linear acceleration in terms of the chosen set of modes.

This is followed by a decision represented by an elongated diamond in Figure 2-7. If, at this point, no accelerometer locations have been specified and no measurement of acceleration made, the specification of the instrumentation array is made. This conditional branch will be explained a little later. The instrumentation array is a set of triples (x_i, y_i, z_i) which are then used to calculate the X-matrix. The functional form of the X-matrix is determined upon selection of the modes but its numerical value is not established until the location of the instrumentation is

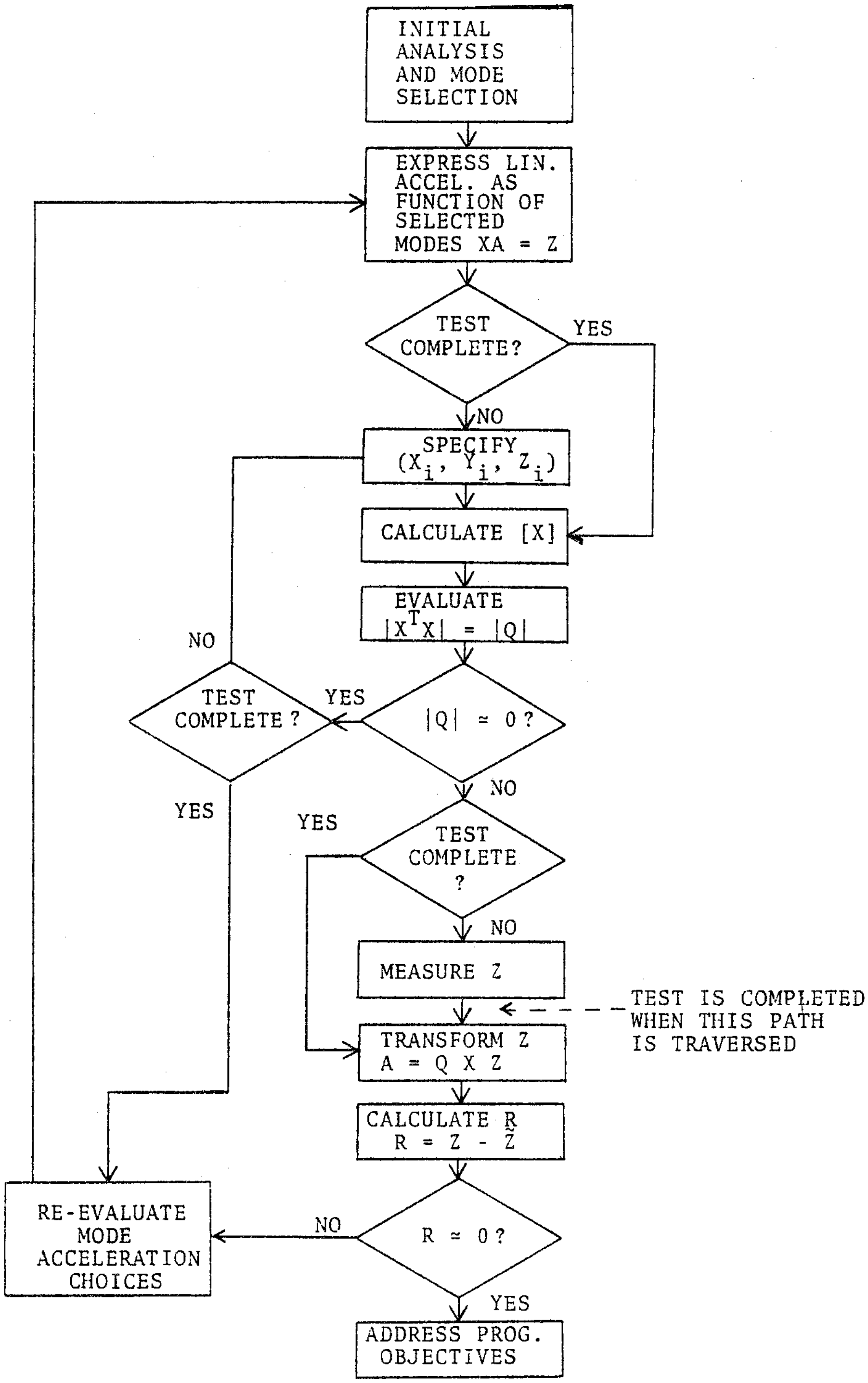


Figure 2-7. Modal Analysis Flow Chart

known. From the X-matrix, the Q-matrix is determined (see Section 2.2) and its determinate, $|Q|$, is evaluated.

A second conditional branch is encountered. If $|Q|$ is zero or near zero, the instrumentation array must be re-evaluated and revised until $|Q|$ is sufficiently large. Again, the condition of completion of testing arises. If the measurement of acceleration has not been made, a test is conducted to obtain this data. For the purposes of this discussion, it is assumed that the conduct of such a test represents a significant portion of the program budget. Therefore, the test is conducted only once; hence, the conditional transfers elsewhere are based on the completion of the test.

The linear acceleration data is transformed to mode accelerations and the residuals are calculated. Based on the magnitude of the residuals and the needs of the program, the flow chart branches, i.e., another decision must be made.

If the residuals are not sufficiently small, the set of modes is re-evaluated and a new $|Q|$ calculated. It should be kept in mind that the X-matrix has a functional dependence on the selection of modes. Thus, a new set of modes may result in a zero or near zero $|Q|$. However, once a set of modes is determined and $|Q|$ is sufficiently different from zero the procedure is the same as before except that no test need be performed. Once sufficiently small residuals are obtained, the proper modes will have been selected and the objectives of the program can be accomplished.

3.0 PRACTICAL CONSIDERATIONS

In the preceeding section the procedures for the general use of modal analysis were presented. This section provides two practical considerations which will aid the user by simplifying the application of the general case already put forth.

First, a method of partitioning a dynamic system into simpler subsystems is detailed to increase the ease and effectiveness of modal analysis. For this purpose, recommendations are made in terms of a hypothetical conventional rail car, with the assumption that is is representative of some 80 to 90 percent of contemporary rolling stock.

The second topic is the instrumentation itself. What is the nature of the accelerations that are to be measured? What are the rigors of the acceleration environment that the accelerometers must survive? Again a representative hypothetical vehicle is used as the basis for the recommendations.

3.1 PARTITIONING THE SYSTEM

Section 2.3 presented the application of modal analysis to a generalized body. In practice,structures, such as a rail vehicle, can be modeled as a compilation of a number of theoretically independent subsystems. That is, the structure can be partitioned at divisions between subassemblies or at interfaces where suspension components are found. For example, a rail vehicle may be said to be comprised of at least four subsystems: the axles, the truck assembly, the vehicle structure and the load.

Apart from computational convenience there are two additional reasons for partitioning the system. The first is that partitioning permits the use of separate Cartesian coordinate systems centered in each subsystem. Since the accelerometers make their measurements with respect to inertial space, an explicit and easily understood description of the accelerations of each subsystem can be obtained.

The second reason for partitioning the system is the conceptual clarity it brings to the dynamic interactions between the subassemblies. For example, the accelerations of the axle, usually measured at the axle journal bearing, are for most purposes a direct measure of the accelerations produced by track perturbations. Hence, transfer functions relating load and car body accelerations to axle accelerations are of direct interest to the structural and suspension designers. In addition, transfer functions relating the load accelerations to the car body accelerations are of interest to the designer of the load container or intermodal highway truck trailer. The acceleration field of the load subsystem is of interest to each designer, and particularly to a prospective shipper.

3.1.1 TERMINOLOGY OF MODAL COMPONENTS

It is convenient to introduce the common names used for most of the rigid-body modal coordinates. Table 3-1 relates the modal components to their common names and to the notation for them adopted in Section 2.0.

The elastic body modes are similarly described in Table 3-2. The bending mode may be conceptually thought of as a sheet of metal draped over a stiff rod (the axis). The result, of course, is the first bending mode about the rod or axis. For example, if the rod coincided with the y-axis this bending mode would then be the vertical bending mode about the y-axis denoted β_{yz} where z denotes the direction of action. In what follows the only vertical mode to be considered is bending about the y-axis and, therefore, will be referred to as simply the vertical bending mode. Similarly, bending about the

TABLE 3-1

COMMON NAMES OF RIGID BODY
MODAL COMPONENTS OF ACCELERATION

Component	Common Name	Symbol
Vertical Translation	Bounce	$\ddot{z}_o(t)$
Lateral Translation	Sway	$\ddot{y}_o(t)$
Longitudinal Translation	--	$\ddot{x}_o(t)$
x-Axis Rotation	Roll	$\ddot{\theta}_o(t)$
y-Axis Rotation	Pitch	$\ddot{\phi}_o(t)$
z-Axis Rotation	Yaw	$\ddot{\psi}_o(t)$

TABLE 3-2

COMMON NAMES OF ELASTIC BODY
COMPONENTS OF ACCELERATION

Component	Common Name	Symbol
Bending Mode (About y-Axis)	Vertical	$\beta_{yz}(t)$
Bending Mode (About z-Axis)	Lateral	$\beta_{zy}(t)$
Torsion Mode (About x-Axis)	Torsion Mode	$\tau_x(t)$

z-axis in the y-direction, β_{zy}, will be referred to as lateral bending. Finally, the only torsional mode considered is about the x-axis, τ_x, and will be referred to as simply the torsion mode.

3.1.2 AXLE SUBSYSTEM

The axle subsystem is very stiff compared to the other subsystems. For this reason attention is restricted to the rigid-body components of axle acceleration. The pitch component is neglected since it corresponds to axle rotation.

Thus, the A-vector defined in Section 2.2 consists of five elements which are identically the five rigid-body accelerations of the axle. Five accelerometers deployed as shown in Figure 3-1 yield five simultaneous equations of axle acceleration. The solution of this system of equations will in turn yield the five modal accelerations of interest.

3.1.3 TRUCK SUBSYSTEM

The truck subsystem is a somewhat more complex system than the axle subsystem and is currently the object of intense investigation. (Reference 1 details some limited results obtained using truck modes.) In contrast to the axle,the truck requires the use of the entire compliment of rigid-body modes since the side frames are indeed capable of pitch. In addition, most trucks behave as a four bar frame; that is, as if the corners were secured by ball joints of limited displacement. A hybrid mode called twist is thus introduced which is directly analogous to the elastic-body torsion mode previously described. This is illustrated in Figure 3-2. Twist plus the six rigid-body modes of the truck acceleration result in an A-vector (refer to Section 2.2) containing seven elements.

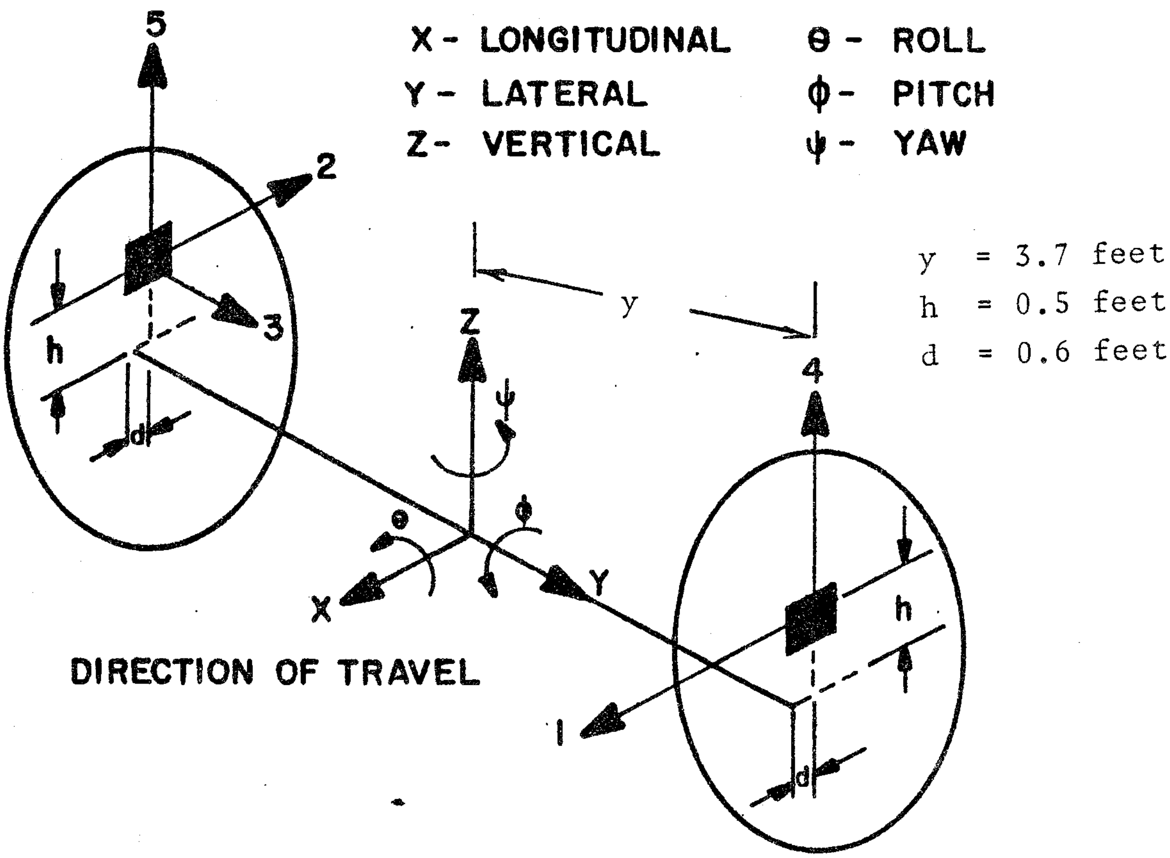
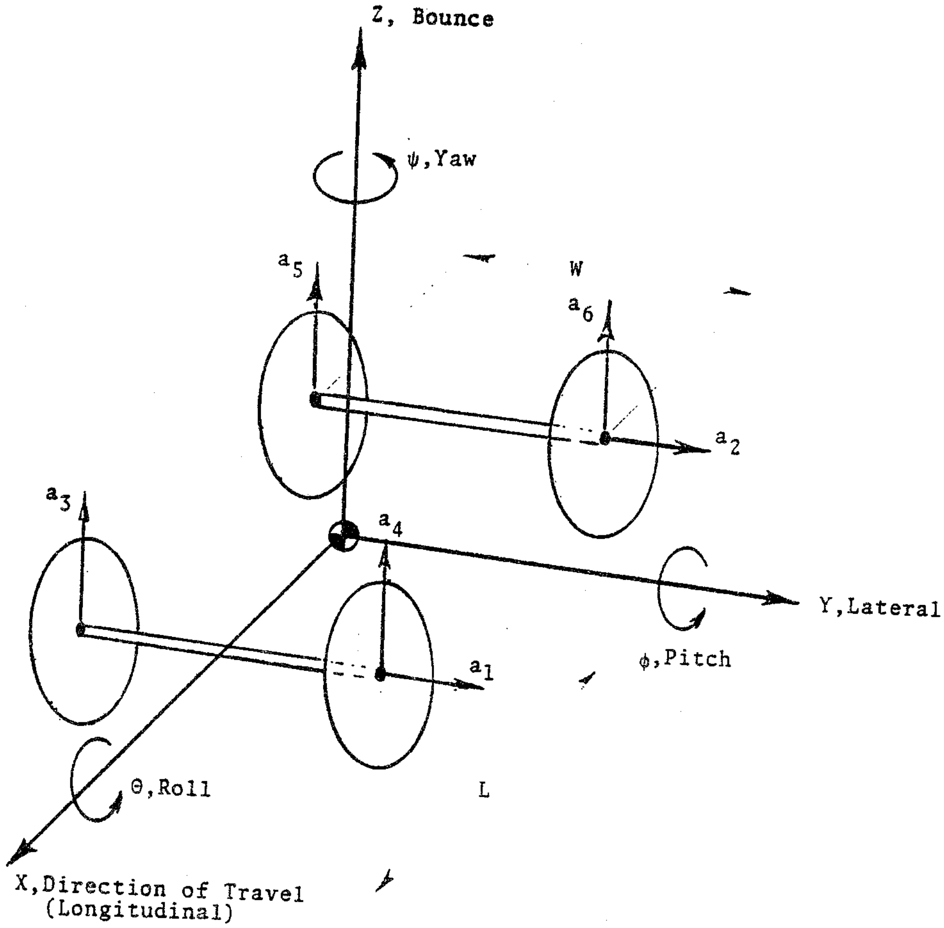
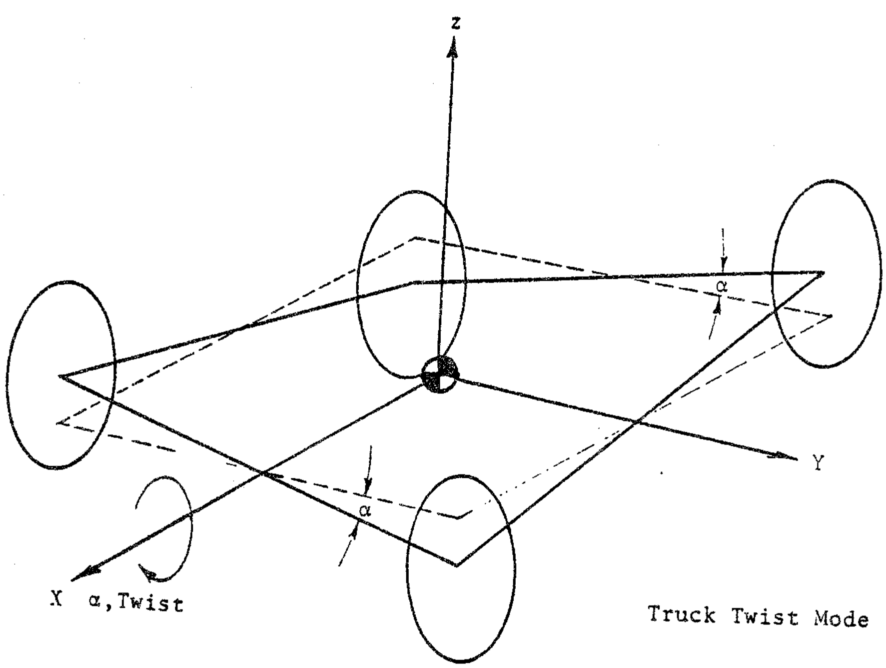


Figure 3-1. Axle Transducer Locations and Axes



Truck Conventions and Transducer Locations



Truck Twist Mode

Figure 3-2. Carbody Modes

An instrumentation array of seven accelerometers shown schematically in Figure 3-2 will provide the necessary set of equations to determine the seven modes of interest. In fact, this set of equations can be written by inspection and solved algebraically (see Reference 1).

3.1.4 CARBODY SUBSYSTEM

Modal analysis of a typical rail vehicle carbody requires careful consideration. That is, in addition to the standard six rigid-body modes there can be several elastic-body modes required to adequately model the motion of the carbody. A boxcar may, for instance, require the first and second modes of both the vertical and lateral bending families to describe the acceleration field while a flatcar may require only the vertical bending family with the addition of the torsion family. Some insight to the structural design of the specific vehicle will shed light on the selection of the elastic-body modes. Recall that the procedure outlined in Section 2.4 depends on an iterative process to determine the modes. It will usually require one or two iterations to identify the necessary set of modes to describe the vehicle motion.

In general, a conservative estimate for the required number of elastic modes would be two modes, one odd and one even (see Section 2.3), from each of the three families for a total of six elastic modes. In addition, the mode shape coefficients will usually add additional elements to the A-vector. A conservative estimate would be an additional six elements due to mode shape coefficients. Thus, the A-vector may contain as many as 18 elements; six due to rigid-body modes, six due to elastic-body modes; and six due to mode shape coefficients. This means that the solution of the resulting set of eighteen equations will involve the inversion of an 18 x 18 matrix. This may be a problem depending on the capacity and type of computer system available.

A practical alternative involves a piecewise solution. That is, the system of equations representing the local measurement of linear acceleration in one of the directions (x, y or z) is first solved for only those modes contributing to acceleration in this direction. Some of these modes will also contribute to acceleration in one of the other directions. For example, roll contributes acceleration to both the y and z directions. These modes then become known quantities in the solution of the equations in a second direction and so on. The piecewise solution offers the advantage of reducing the number of elements in the A-matrix and hence the size of the matrix to be inverted, typically by a factor of two or more. In the most efficient matrix inversion algorithms this means a reduction in capacity and time by more than a factor of four. In practical applications many computers are limited to the inversion of matrices on the order of 10 x 10. Thus, the piecewise solution becomes not only economical, but necessary. On the other hand, the simultaneous solution of all three directions has the theoretical advantage of a more accurate solution. The return on investment of time and energy, however, may be small.

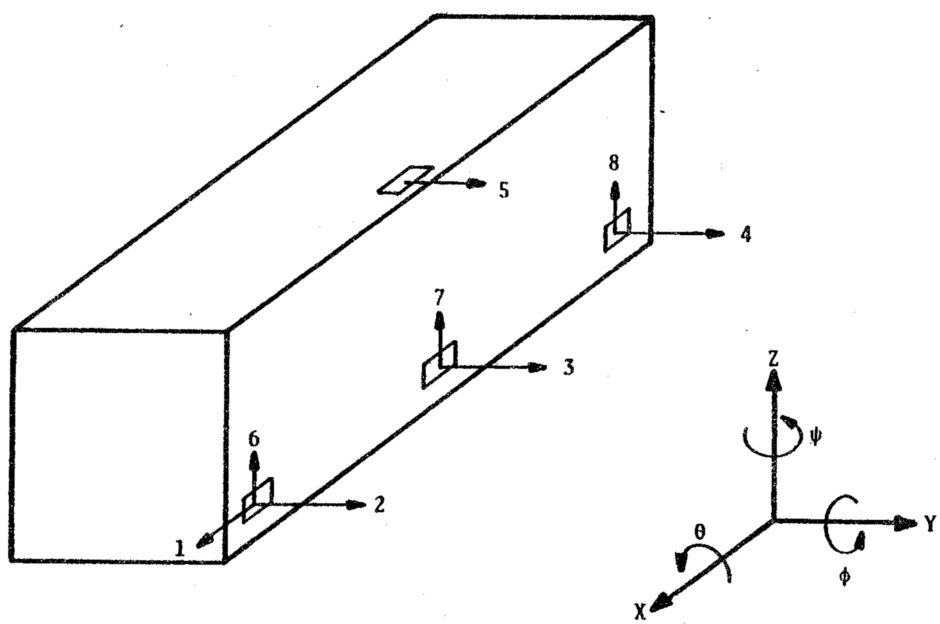
When using the piecewise solution there is a recommended order for the solution. For obvious reasons acceleration in the y- and z-direction are usually of greater interest than in the x-direction. In fact, experience has shown that the ratio x:y:z is approximately 1:2:3 in most rail acceleration environments. Thus, the z-direction is the preferred starting point from which approximately nine of the required 18 A-vector elements can be determined. This leaves nine elements to be calculated. The x- and y-directions may be solved simultaneously to complete the description of carbody dynamic performance.

Once the A-vector has been solved the various terms that are a sum of two or more modes and/or mode shape coefficients must be decoupled. It should be remembered that decoupling of the various modal acceleration components requires the evaluation of definite integrals in which the mass distribution and moment-of-inertia distribution occur. Note further that for any axis where decoupling is required the mass distribution and moment-of-inertia distribution along that axis must be known. It is probable that the longitudinal distribution of mass has previously been computed in the design phase of the vehicle, while the longitudinal distribution of moment-of-inertia is probably not available. It is recommended that the evaluation of functions of this kind be approximated rather than precisely determined. Experience has shown that skillfully made approximations can be acceptably close to the exactly tabulated function. In any event, these integrals need be evaluated only once for a given choice of modes and mode-shape functions.

3.1.5 LOAD SUBSYSTEM

The treatment of loads is similar to the evaluation of the carbody in that some knowledge of the structure is required to develop the necessary set of modes. Of course, in most applications the six rigid-body modes should be included. For the purpose of this discussion, reference is made to conventional trailers and containers presently in use in the intermodal transportation network. Both body types will in most cases require one member from each of the elastic-body families discussed earlier. Depending on the structure and lading,mode-shape coefficients may add elements to the A-vector. Thus, there may be as many as ten elements in the A-vector, six rigid-body, three elastic-body and one mode shape coefficient.

Figure 3-3 represents schematically a practical array of accelerometers which offers a small degree of redundancy which is necessary to ensure sufficient data and also as an alternate solution in the event of an accelerometer failure. That is, a solution is possible as long as the number of properly functioning accelerometers is equal to or greater than the number of elements in the A-vector.



X - longitudinal
Y - lateral
Z - vertical

θ - roll
ϕ - pitch
ψ - yaw

Figure 3-3. Schematic Array of Accelerometers

3.2 INSTRUMENTATION REQUIREMENTS

With the preceeding discussion of subsystem modeling complete, it is now possible to discuss the instrumentation requirements. This breaks down into two basic considerations. First, the magnitude of the accelerations to be measured and the specifications of the accelerometers; second, the actual operating environment the accelerometers will be exposed to.

To begin with, the accelerations on a vehicle, at least on the carbody and load, should not exceed levels that would cause excessive vibration to the lading, whether cargo or people. Most items designed for use by people are not typically designed to withstand accelerations greater than 32.2 ft/sec^2 or one gravity (1 g). In fact, measurements have shown this to be the case with few exceptions, such as, accelerations due to impact which may reach as high as 2 to 3 g. Therefore, in order to provide a sufficient range to measure the accelerations anticipated, a reasonable specification for an accelerometer would be ±5 g providing sufficient resolution can be obtained.

Fortunately, there are a large number of commercially available accelerometers with the required range which offer a resolution of 0.01 g to 0.001 g which should be sufficient for most applications. The acceleration environment encountered at the axle and truck level is somewhat more severe and would be better suited to a ±10 g accelerometer (resolution 0.03 to 0.01 g).

In order to determine the required frequency response of the accelerometer, use is made of simple spring-mass, uniform beam, and torsional pendulum models. For a large percentage of conventional rail cars, the frequencies of interest are relatively low. For example, rigid body linear accemerations, such as bounce, will occur at between 3 Hz and 5 Hz. Rotational accelerations occur somewhat lower, around 2 Hz. The elastic-body mode frequency occurs somewhat higher, but the first mode of both the bending and torsion mode lies below 10 Hz. The second mode of the torsion mode is typically the highest at just below 30 Hz. Thus, the accelerometer selected should be capable of responding to inputs of at least 30 Hz and perhaps as high as 50 Hz. In any case, this range is perfectly feasible with today's instrumentation.

Besides being able to take measurements within the desired frequency band, the accelerometer must be able to operate over extended periods in a harsh environment. It is not uncommon for an acceleration of 1,000 Hz and 100g amplitude to occur due to a phenomenon such as local deformation. For example, a plate of decking may rattle or a bracket resonate at these levels. The local deformation is small, less than 0.001-inch for the conditions cited, but the effect on the accelerometer can be devastating.

To alleviate this problem use is made of a mechanical isolator (one type is described in the following section). A mechanical isolator is designed to pass the acceleration in the frequency band of interest with unity gain while sharply attenuating the higher frequencies. One simple solution is the use of a wood block as a mount. Careful attention, however, must be given to the placement of the wood to avoid creating a resonating bracket. Other isolators involve the use of rubber foam mounts to dissipate the high frequency energy.

In addition to mechanical isolation, some attention should be given to environmental protection. Electronic instrumentation, such as accelerometers, should be protected from rocks, dust and water. Finally, depending on the design of the test and atmospheric conditions, the accelerometer should be maintained at a constant temperature, typically in the range of 70°F to 90°F.

Before closing the section on practical considerations, a few words on data sampling are in order. Before digitizing, the data should be filtered at approximately the highest frequency of interest using a very sharp cut-off filter. This is done to avoid aliasing the information in the pass band with spurious high frequency noise. The sample rate should be two and a half to three times the corner frequency of the anti-aliasing filter or highest frequency of interest. In fact, in some cases, it is advisable to increase this to four times the highest frequency of interest.

Finally, the test procedures must be designed to provide statistically sufficient data. When using Fourier Transform analysis, the normalized statistical error is equal to the inverse of the square root of the number of samples transformed. In the study of vehicle dynamics, the general required resolution of frequency is approximately 1 Hz, occasionally less, thus, requiring samples of 1 second duration or more. One hundred seconds of data would provide a normalized statistical error of 0.1. This means that two thirds of the data lie within 10 percent of the true value. Using this, an estimate of the test requirements can be made based on desired accuracy.

4.0 AN APPLICATION

The Lightweight Flatcar Evaluation (LWFC) Program is presented as an example of the application and utility of Modal Analysis. Modal Analysis was developed and used to interpret the test data of the LWFC Program so that the major objective, a comparative evaluation of lightweight and conventional flatcars could be accomplished. The global nature of the mode accelerations permitted a clear and objective comparison of the dynamic performance of two different vehicle designs.

4.1 PROGRAM AND OBJECTIVES OF THE LIGHTWEIGHT FLATCAR EVALUATION PROGRAM

The Lightweight Flatcar Evaluation Program was designed to compare two prototype skeleton flatcars, referred to as lightweight flatcars, with the conventional TTAX flatcar by quantifying the acceleration environment experienced thereon. The new skeleton flatcars are shown in Figure 4-1. The TLDX62 (shown on the left) is designed to carry standard containers, and the TLDX61 (shown on the right) is designed to carry trailers. These flatcars weigh between 20 and 30 percent less than the widely used TTAX car and they could provide substantial reductions in fuel consumption and equipment wear.

The evaluation was accomplished by analyzing the effect of several factors on the dynamic behavior of the flatcar. These factors include flatcar design, load configuration, track class, vehicle speed and mileage accumulated in service. The dynamic performance of the test vehicles was quantified during a series of three Ride Vibration Tests (RVT) and an extended Over-the-Road Test (OTR). The RVT series was designed to obtain data on the flatcars at a different level of accumulated in-service mileage under controlled conditions. The OTR test was designed to obtain data while the flatcars were being used in revenue service.

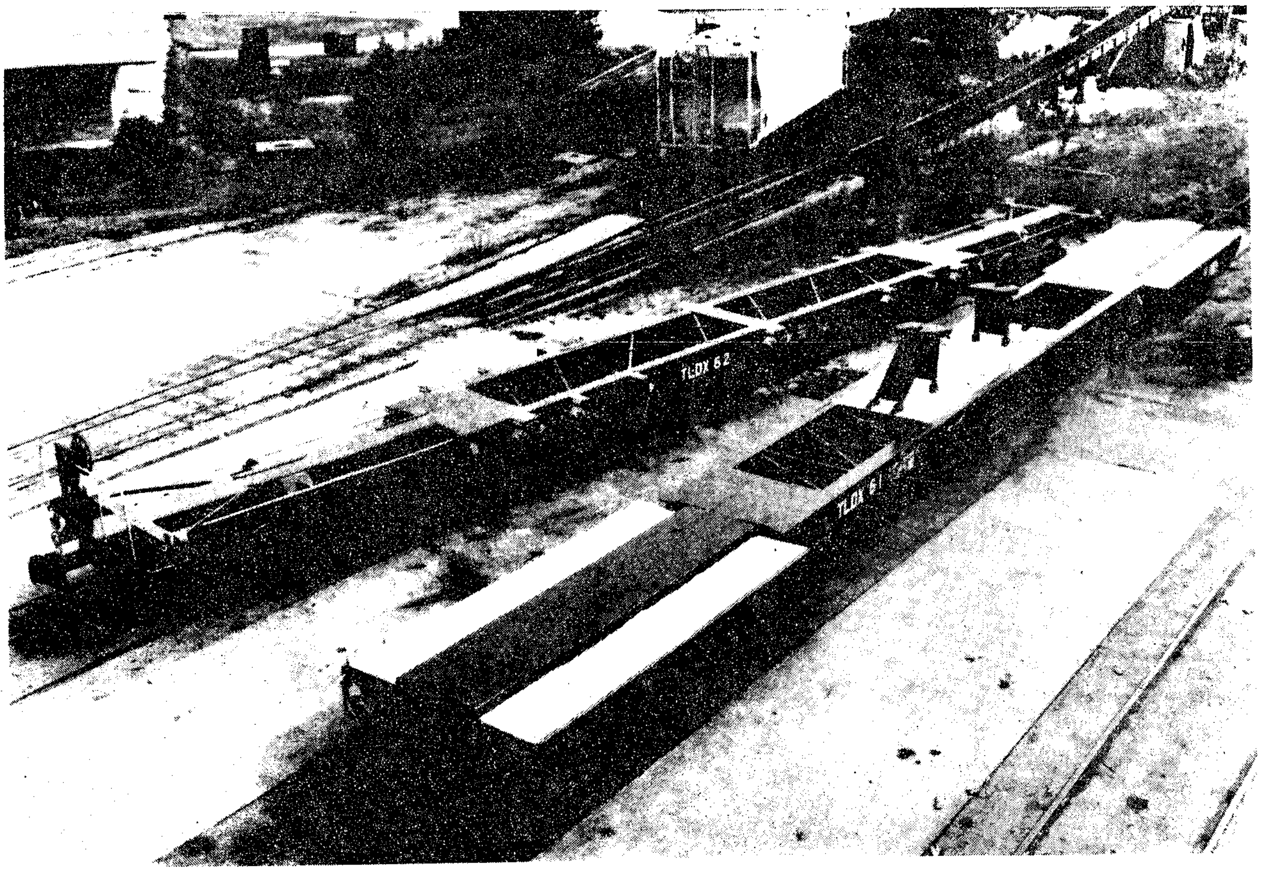


Figure 4-1. Lightweight Flatcars TLDX61 and TLDX62

4.1.1 FLATCARS

Tests were conducted using three different flatcars: one conventional general purpose flatcar and two lightweight prototype flatcars. Accelerations on similar load configurations were found to be comparable for the different types of flatcars.

The conventional flatcar (TTAX 973799) owned and operated by Trailer Train weighs approximately 69,000 pounds including trucks. The car is 90 feet long (over strikers) and 9 feet wide with the deck 2 feet, 5-1/2 inches above the rail. There are two collapsible kingpin pedestals, one at the car center and one at the B-end*. The TTAX is capable of transporting

*The end at which the hand brake is located is referred to as the B-end. The other end is the A-end. All tests were performed with the A-end leading.

two trailers, two 40-foot containers, or one of each. The trucks used in this study were 70-ton American Steel Foundary (ASF) ride control trucks spaced 66 feet center-to-center.

One lightweight flatcar (TLDX61) was configured to transport only trailers and the other (TLDX62) was configured to transport only containers. The TLDX61 flatcar is shown in Figure 4-2 laden with two trailers and the TLDX62 flatcar is shown in Figure 4-3 laden with a single container. These flatcars are referred to as lightweight or skeleton flatcars (see Figure 4-4). These cars weigh approximately 59,000 pounds and 49,000 pounds empty, respectively. Length over strikers is 84 feet, the width is 9 feet, and the deck is 3 feet 5-1/2 inches above the rail. The trucks under these cars were also the 70-ton ASF ride control trucks spaced 65 feet center-to-center.

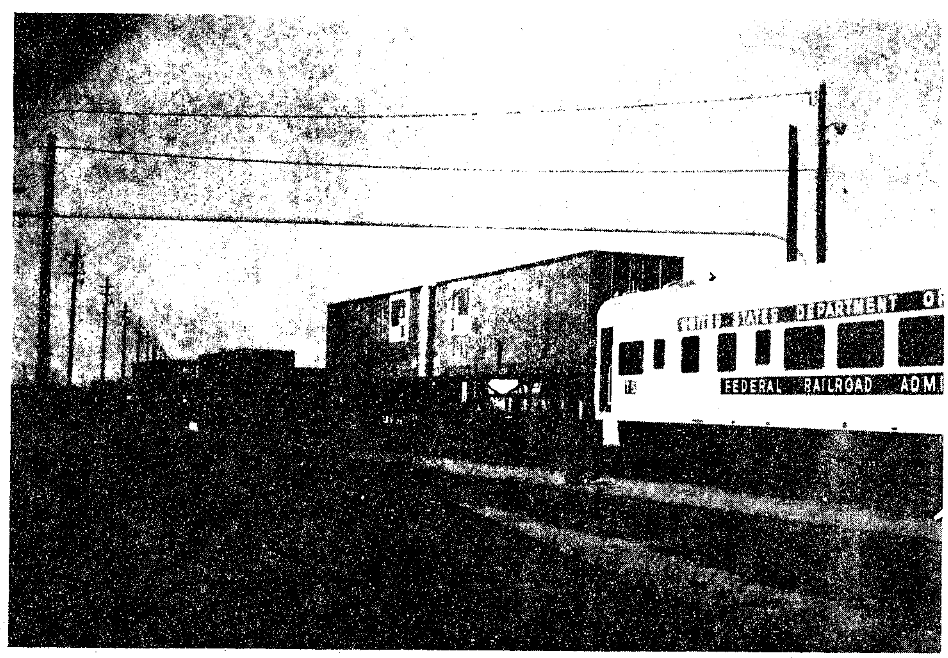


Figure 4-2. TLDX61 Flatcar With Two Trailers

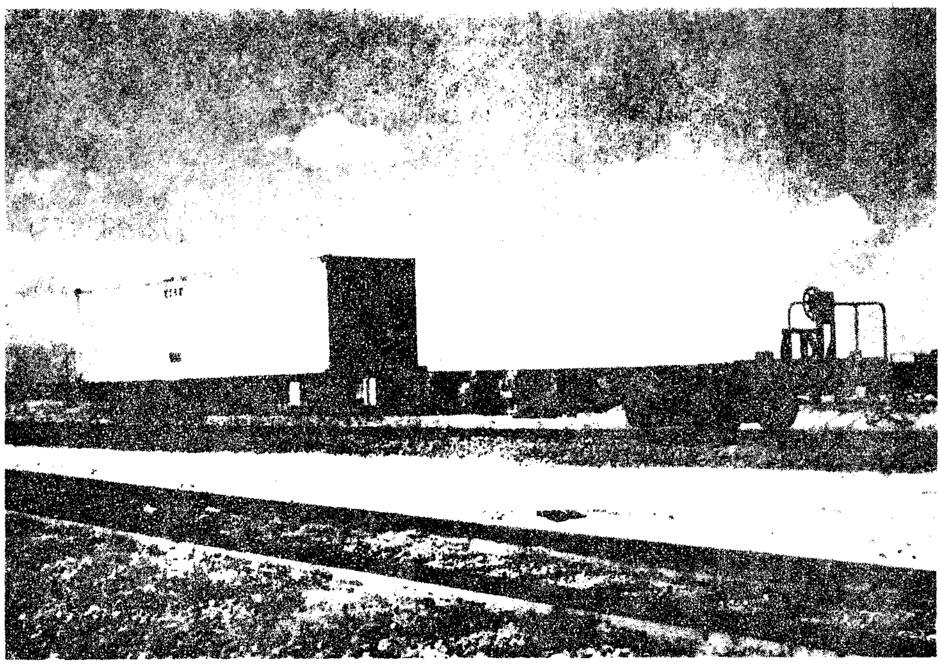


Figure 4-3. Standard 40-Foot Container

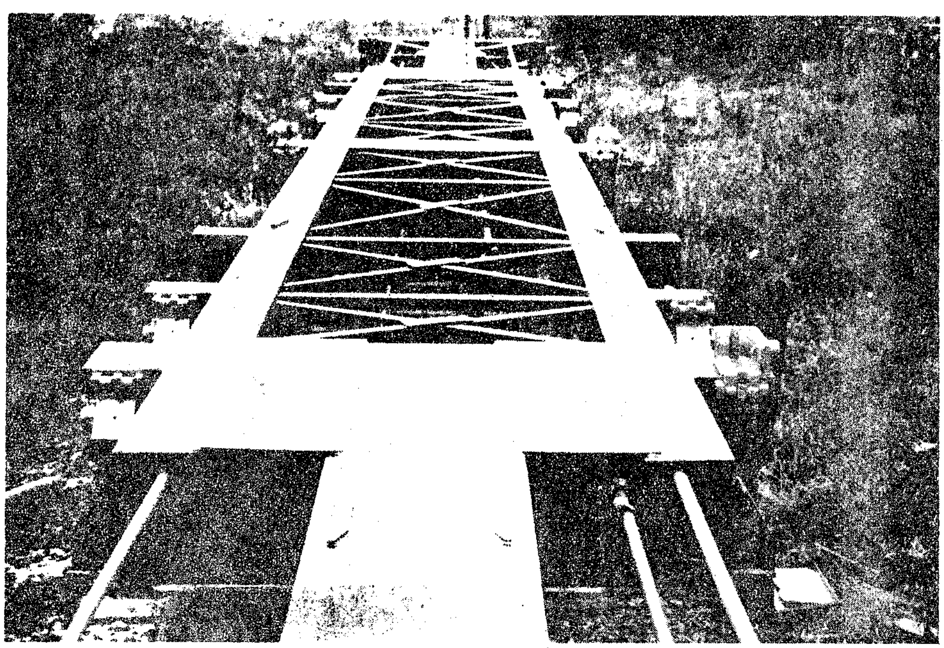


Figure 4-4. Lightweight Flatcar, TLDX62

4.1.2 LOADS

The lading of the flatcars consisted of standard intermodal trailers and containers. During the various phases of the test program, the trailers and containers were laded with a number of paper bales weighing 2,000 pounds each and in some limited cases were left empty.

The trailer used for this study was the Fruehauf Z-Van (Model FBZ9-F2-40) shown in Figure 4-5. This trailer has a length of 40 feet and an overall height of 13 feet 6 inches. The empty weight is 12,500 pounds and the maximum gross weight is rated at 68,000 pounds. During the test program the trailers were loaded to a gross weight of 44,500 pounds. Four of these trailers (SFTZ 202519, 202699, 202710, 202751) were employed during testing. All four trailers werc manufactured during August and September 1974 and had been used in service prior to the LWFC program.

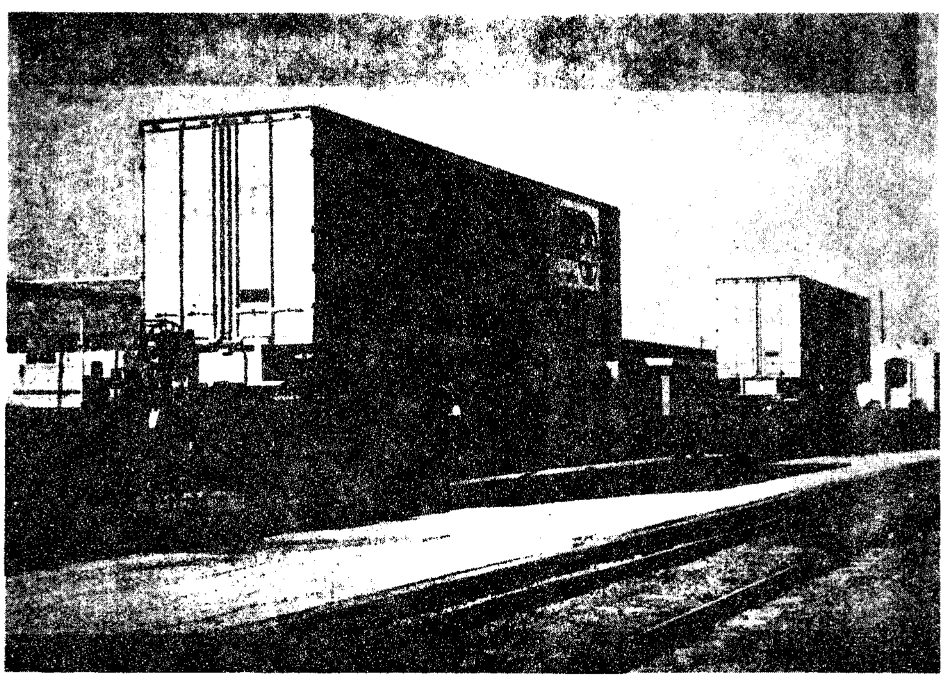


Figure 4-5. Instrumented Trailer

Similarly four Fruehauf general cargo containers (Model KAX-40TRA; Serial Nos. XTRU 871264, 871395, 874756, 878189) were used. A container is shown in Figure 4-3. The container is constructed of a steel frame with sheet aluminum sides and a wooden floor. These containers were 40 feet long, 8.5 feet wide and 8 feet high. Each container weighed 6,450 pounds empty and was rated at a maximum of 64,000 pounds gross weight. During testing the gross weight of each container was 38,450 pounds.

4.1.3 INSTRUMENTATION

In order to obtain high-resolution measurements of acceleration, precision servo-accelerometers, manufactured by Schaevitz, were used. These accelerometers (Model LSBC-5) had a dynamic range of ±5g and a natural frequency of approximately 150 Hz with near critical damping. Resolution at full-scale conditions (5g) was ±0.01g; at 1g resolution was ±0.003g; and at conditions characteristic of the piggyback environment (0.1g) resolution was ±0.001g or better. The 5g accelerometers were used to measure the acceleration environment on the loads and carbody. The axles were instrumented with viscously damped accelerometers having a 30g range (Schaevitz Model LSVCJ-30). These instruments were capable of a resolution of 0.18g in a 30g environment and could resolve to 0.012g in a 1g environment.

As discussed in Section 3.2, all accelerometers were protected using a mechanical isolator designed to attenuate high amplitude or shock accelerations above 150 Hz. The mechanical isolator was basically a cup-in-a-cup design with the inner cup isolated from the outer by a firm, open-cell foam. The mechanical isolator is shown in Figure 4-6 and the complete accelerometer/isolator package is shown in Figure 4-7.

Eight accelerometer/isolator packages were mounted on each trailer and each container. The instrumentation array on each trailer is shown in Figure 4-8. Nominal locations of

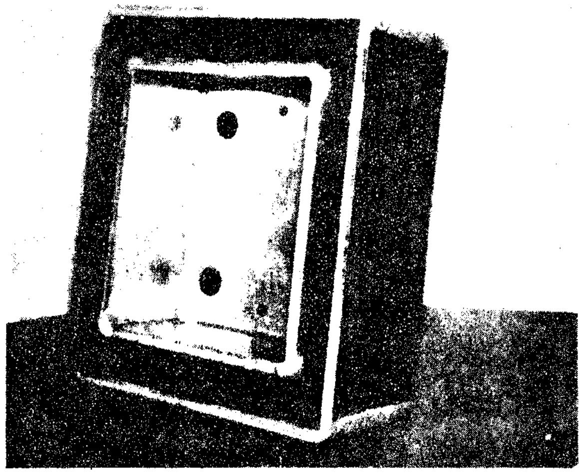


Figure 4-6. Mechanical Isolator

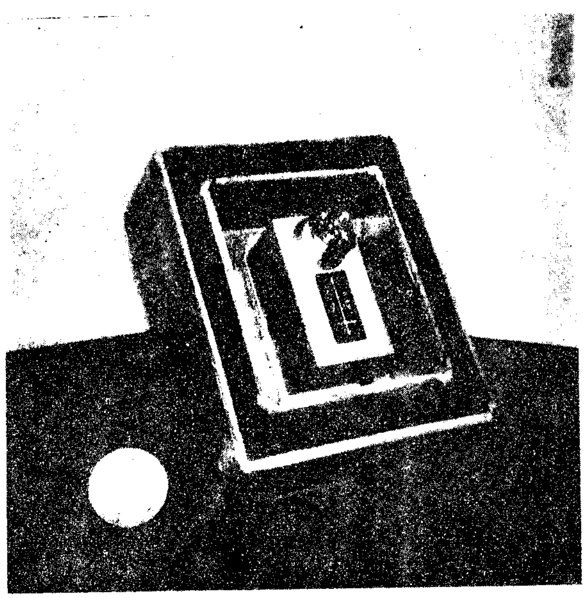


Figure 4-7. Accelerometer/Isolator Package

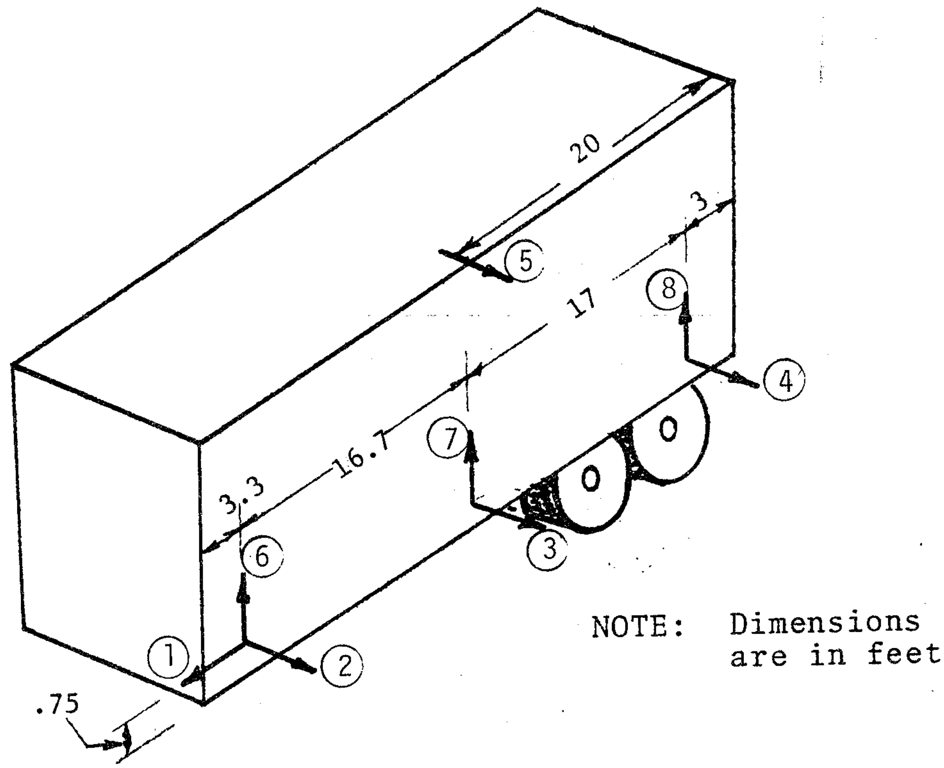


Figure 4-8. Trailer Instrumentation Array

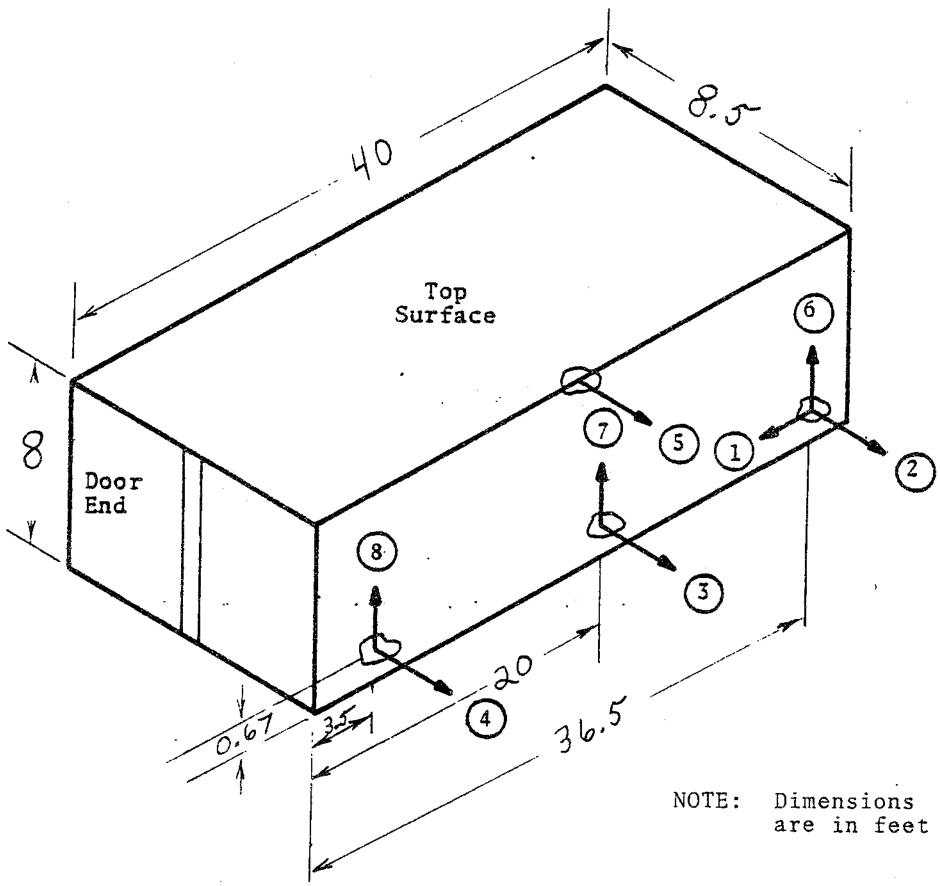
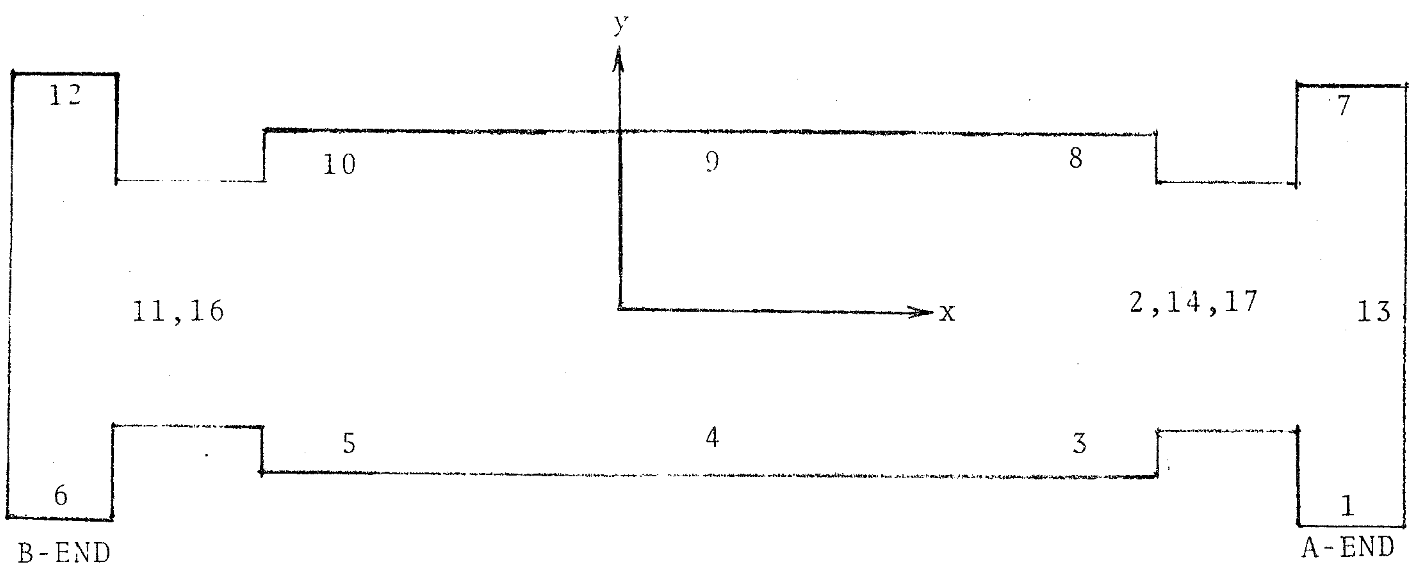


Figure 4-9. Transducer Locations For Containers

the eight measurement stations are shown in feet (±0.25 feet). The accelerometer locations on the containers are shown in Figure 4-9.

Seventeen accelerometers were mounted on the carbody; twelve in the vertical direction, four in the lateral direction, and one in the horizontal (longitudinal) direction. The nominal locations of the instruments were the same for all three flatcars; these locations are indicated schematically in Figure 4-10. The Cartesian coordinates of the accelerometers on the three cars are given in Tables 4-1 through 4-3.



Accelerometer	Direction
1 - 12	Vertical
13 - 16	Lateral
17	Longitudinal

Figure 4-10. Schematic of Carbody Transducer Locations with Identification Numbers

TABLE 4-1

TRANSDUCER COORDINATES FOR TLDX-62 CAR
(L/2 = 41'5")

No.	ID	x	y
		Vertical	
1	43V	39'4-3/4"	-3'5-1/4"
2	41V	31'1-3/8"	10"
3	40V	22'8-5/8"	-2'11-7/8"
4	38V	10-3/4"	-2'10-3/8"
5	37V	-22'9-7/8"	-2'11-7/8"
6	35V	-39'4-1/8"	-3'5-1/2"
7	44V	39'4-3/4"	3'5"
8	45V	22'9"	2'11-5/8"
9	46V	9-5/8"	2'10-7/8"
10	47V	-22'9-3/4"	2'11-5/8"
11	36V	-33'10-1/4"	10-1/2"
12	48V	-39'3-3/4"	3'5-1/4"
		Lateral	
13	42L	39'4"	1'9"
14	41L	31'9"	1'4"
15	39L	1'6-1/8"	2'7"
16	36L	-33'3-5/8"	1'4-1/2"
		Longitudinal	
17	41H	31'1-3/8"	10"

TABLE 4-2

TRANSDUCER LOCATIONS ON TLDX-61 CAR
(L/2 = 41'5")

No.	ID	x	y
		Vertical	
1	27V	40'6-1/2"	-4'2-7/8"
2	25V	31'3"	-10-1/2"
3	24V	23'3-1/2"	-2'11-7/8"
4	22V	3-3/8"	-2'10-3/4"
5	21V	-23'2-3/8"	-2'11-7/8"
6	19V	-40'7-1/2"	-4'3-1/2"
7	52V	40'6-1/8"	4'3-1/4"
8	53V	23'3-7/8"	3'-1/8"
9	54V	3-1/2"	2'10-5/8"
10	55V	-23'3-5/8"	2'11-3/4"
11	20V	-37'7-3/4"	-10-1/2"
12	56V	-40'6-1/4"	3'-1/8"
		Lateral	
13	26L	41'1-3/4"	1'5"
14	25L	30'10-5/8"	1'4-3/8"
15	23L	-1-3/4"	2'7-1/2"
16	20L	-33'9-5/8"	1'4-1/4"
		Longitudinal	
17	25H	30'1-5/8"	10-1/2"

TABLE 4-3

TRANSDUCER LOCATIONS ON TTAX CAR
(L/2 = 45 ft.)

No.	ID	x	y
		Vertical	
1	34V	43'9-1/2"	-4'3/4"
2	32V	31'10-7/8"	7-5/8"
3	31.1V	16'6"	-4'3/4"
4	30V	-8-1/2"	-4'3/4"
5	29.1V	-16'6-1/4"	-4'3/4"
6	28V	-43'9"	-4'3/4"
7	49V	43'9-1/2"	4'3/4"
8	49.1V	16'5-1/4"	4'3/4"
9	50V	7-1/2"	4'3/4"
10	50.1V	-16'6-1/2"	4'3/4"
11	29V	-32'1/4"	7-5/8"
12	51V	-43'6-3/4"	4'3/4"
		Lateral	
13	33L	44'5-5/8"	1'8-1/2"
14	32L	33'9-1/2"	9-1/4"
15	31L	3-1/4"	-1'2-3/8"
16	29L	-32'2-1/2"	-1'4"
		Longitudinal	
17	32H	31'1-3/4"	7-5/8"

Five accelerometers were mounted on the axle. A triaxial accelerometer package was located on top of one bearing housing and a biaxial package was located atop the other bearing housing, as shown in Figure 4-11. The accelerometer signals were processed, collected and recorded by the Data Acquisition Vehicle, T-5. Figure 4-12 shows an overall schematic of the accelerometer signal flow through the data acquisition system.

Each individual accelerometer was cabled to a junction box by means of a shielded cable. A 61-pair shielded cable was used to carry the data signals from the junction box to the data acquisition car. All cables were secured at regular intervals to eliminate cable movement and to protect the cables.

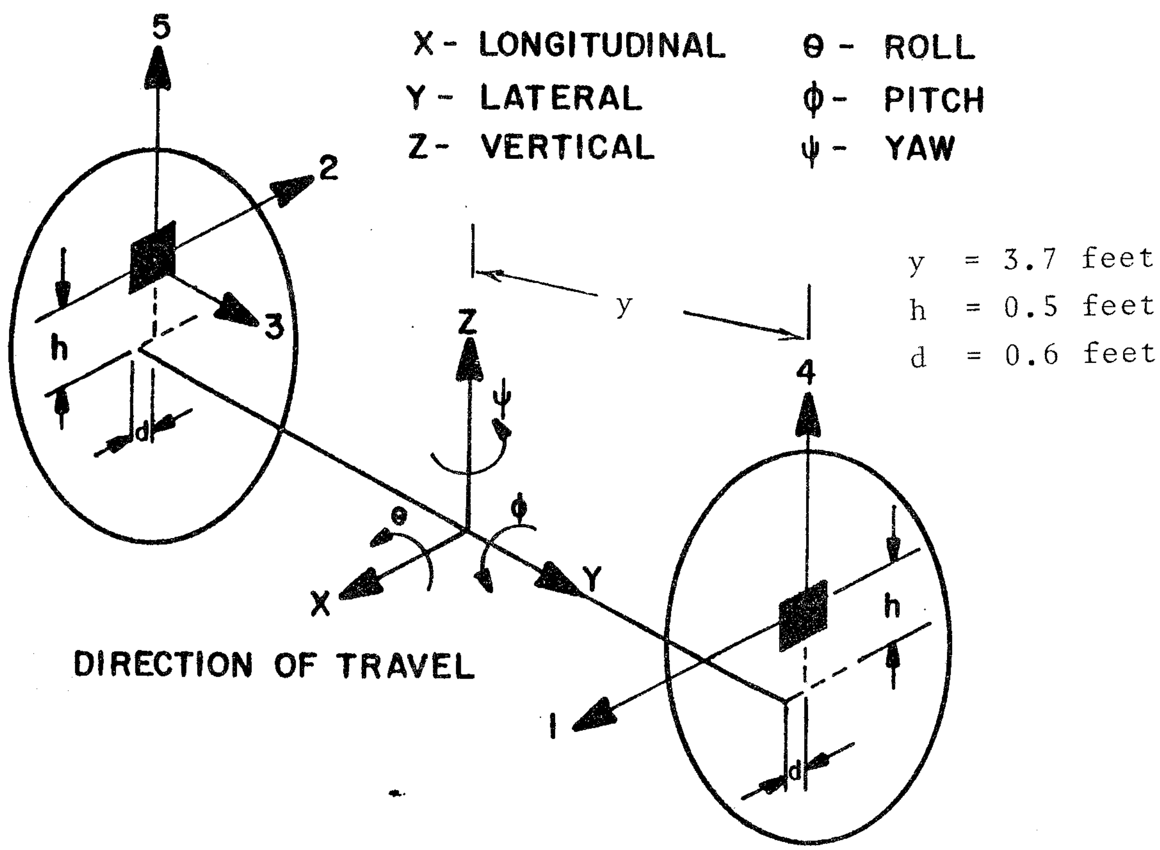


Figure 4-11. Axle Transducer Locations and Axes

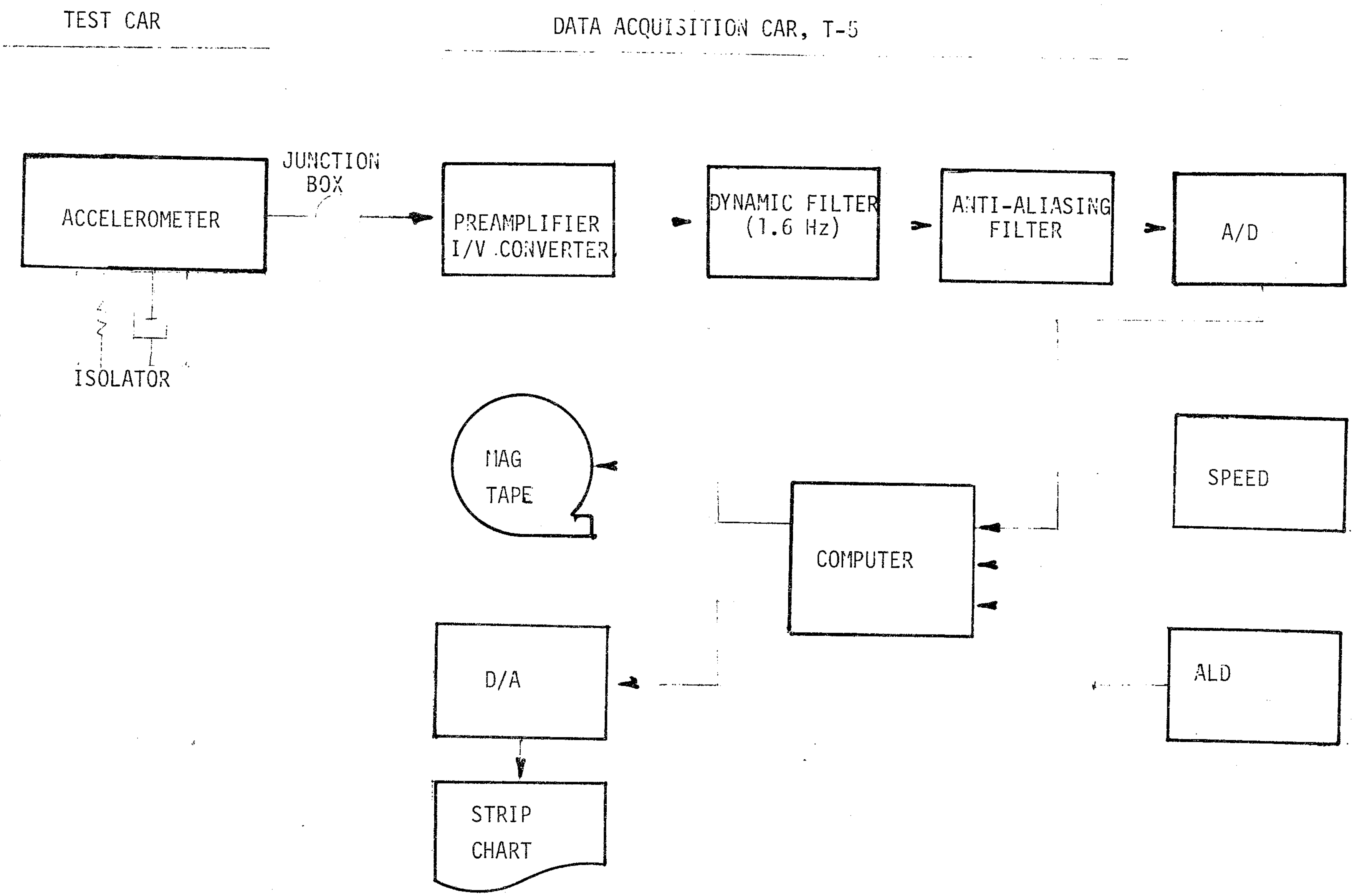


Figure 4-12. Schematic of Instrumentation and Recording System - T-5 Data Acquisition Car

Onboard the data acquisition car the signals were conditioned, amplified and filtered. A four-pole (-24 dB/octave) Bessel filter with a corner frequency of 30 Hz (-3 dB) was used for the purpose of anti-aliasing. This filter provides a nearly linear phase shift in the passband. A second single-pole filter (-6 dB/octave) having a corner frequency of 1.6 Hz was used since the acceleration level was observed to increase with increasing frequency in the rail environment. The acceleration signal was thus low-pass filtered to allow maximum resolution in the frequency band of interest (0-30 Hz) during the process of digitizing. This allowed maximum use of the dynamic range of the analog-to-digital (A/D) converter. The effects of the 1.6-Hz filter were removed during data processing.

The signals, once filtered, were digitized at a rate of 128 samples per second which is slightly more than four times higher than the highest frequency of interest. This process was controlled by an onboard computer which supplied a buffer or temporary storage for the data. When the buffer was full, the data was written on magnetic tape in 12-bit words. Speed and location (milepost) were also recorded with these data. The computer also provided for manually recording a header for each data file to aid retrieval. Six selected channels were re-converted to a quasi-analog signal and were displayed on a strip chart recorder. This display provided the capability for data validation and real time analysis.

4.2 TEST METHODOLOGY

Two types of tests were performed in order to obtain the necessary data. The first test was the Ride Vibration Test and the second was the Over-the-Road Test. The geographic location of these tests are shown in Figure 4-13.

4.2.1 RIDE VIBRATION TEST

A series of three Ride Vibration Tests (RVT) were conducted on the Santa Fe Railroad, First District, Colorado Division,

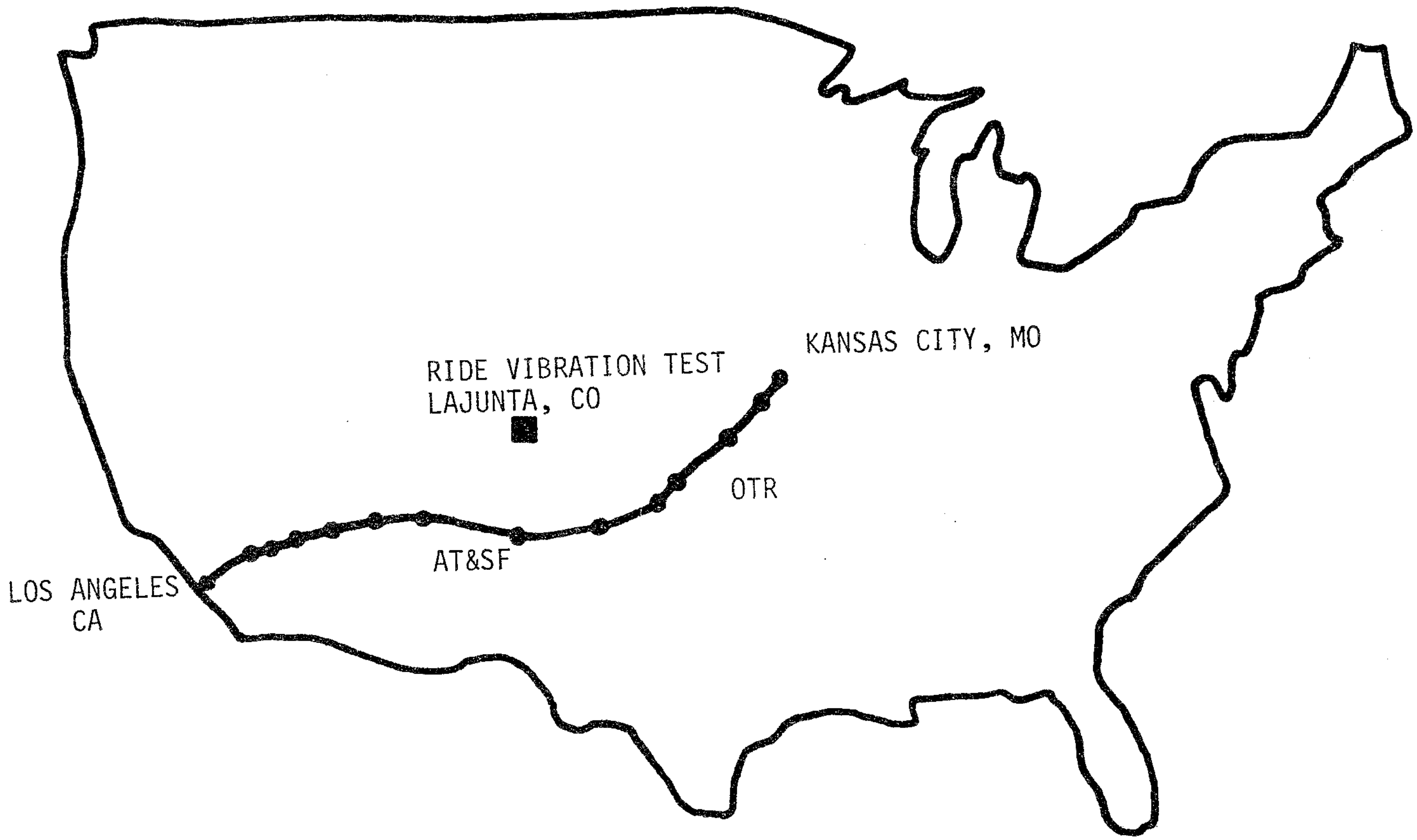


Figure 4-13. Test Zone Locations

near La Junta, CO. The RVT's were run after the test consist had accumulated zero, 50,000 and 125,000 miles. The first was conducted in August 1976; second in August 1977 and the third in November 1978. The three flatcars amassed the mileage while in regular freight revenue consists between RVT tests.

Two test zones were chosen on tangent track. Test Zone 1 consisted of 1.1 miles (5,808 feet) of Class 3 track beginning at milepost (MP) 234 and extended 528 feet south of MP 233 on the Boise City Line. Test Zone 2 consisted of 3.1 miles (16,368 feet) of Class 3 mainline track beginning 4,224 feet west of MP 541 and extended to a point 528 feet west of MP 537.

The entry to both test zones was marked with three automatic location device (ALD) targets and the exit with two ALD targets. Additionally, single ALD targets were placed every 528 feet. A Ride Vibration Test consist is shown in Figure 4-14. The consist was made up of, from left to right, the TLDX62, TTAX, TLDX61, and Data Acquisition Car T-5. Tractive power was supplied by a locomotive coupled to TLDX62. Refer to Table 4-4 for a listing of target speeds in the two test zones.

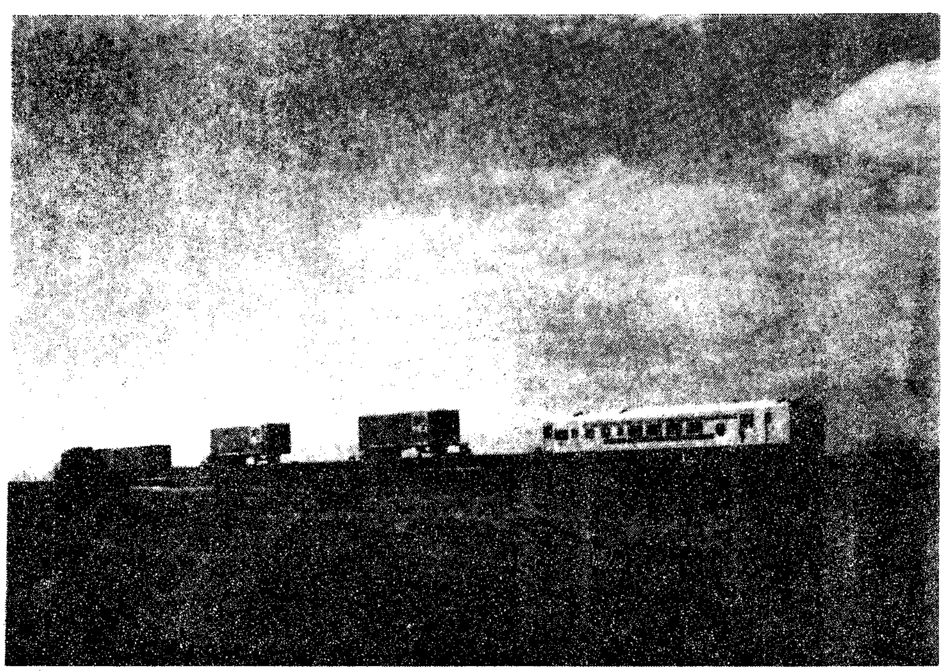


Figure 4-14. Ride Vibration Test Consist

Tests were conducted as follows. The consist traversed each test zone at low speed (10-20 mph) to condition the track and to check the ALD targets. Once this was accomplished, the consist backed through the test zone a sufficient distance to allow the test consist to attain the specified target speed. The target speeds, listed in Table 4-4 were then held constant while data was collected in the test zone. On completion of each data run, the consist was stopped and again backed through the test zone to get ready for the remainder of the tests.

TABLE 4-4

TARGET SPEEDS FOR THE RIDE VIBRATION TEST

Test Zone	Speed (mph)				
1	10	15	20	30	40
2	40	50	60	70	79

4.2.2 OVER-THE-ROAD TEST

The second type of test performed during the LWFC Program was conducted under actual operating conditions in revenue service and is referred to as the Over-the-Road (OTR) test series. During the OTR test series, the three test flatcars were instrumented and cabled to the T-5 data acquisition vehicle, and were coupled to the end of a regularly scheduled freight train on the Atchison, Topeka and Santa Fe (AT&SF) between Kansas City, MO and Los Angeles, CA. A photograph of one such consist is shown in Figure 4-15 taken from the T-5 vehicle. Note the instrumentation on the test cars and the length of the regular train. Refer to Table 4-5 for the lading configuration tested on the three flatcars during the RVT and OTR test series.

TABLE 4-5

LADING CONFIGURATIONS TESTED ON THE THREE FLATCARS DURING THE RVT AND OTR TEST SERIES

TEST CAR	EMPTY	ONE EMPTY TRAILER (A-END)	ONE LOADED TRAILER (A-END)	TWO LOADED TRAILERS	ONE EMPTY CONTAINER (A-END)	ONE LOADED CONTAINER (A-END)	TWO LOADED CONTAINERS
TLDX62	RVT_1 OTR	RVT_1	RVT OTR	RVT_1 OTR			
TTAX	RVT_1 OTR		RVT OTR	RVT_1 OTR		RVT_1 OTR	RVT_1 OTR
TLDX62	RVT_1 OTR				RVT_1	RVT OTR	RVT_1 OTR

RVT_1 - Ride Vibration Test No. 1 only

RVT - Ride Vibration Test Nos. 1, 2 and 3

OTR - Over-the-Road Tests

TABLE 4-6

SPECIAL TEST ZONES FOR OVER-THE-ROAD TESTS (OTR)

Division	MP	Location	Grade	Curvature	Rail	Remarks
1. L.A.	735-725	Newberry	Flat	Tangent	W	Double
2.	716-706	Pisgah	1% E.B.	Reversed 2°	W	Double
3.	656-646	Cadiz	Sag	Mostly Tang.	W	Concrete Ties 651-650 DBL
4. Albq.	527-517	Harris	1.42 EB	Numerous to 7°	W	Double
5.	412-402	Eaglenest	Sag, 10,000' V.C. 1%	Long 1°	W	Double
6.	231-221	Pinta	To .6% EB	Few Short	N.J./S.W.	Double
7. N.M.	888-878	Becker	To .6% EB	One 1°	W	Single
8. Plains	620-610	Black	Rel. Flat	Tangent	W	Single
9.	408-398	Fargo	Rel. Flat	Tangent	W	Single
10.	320-310	Loder	Rel. Flat	Tangent	W	Single
11. Middle	168-158	Aikman	Undulating, E.B.	Few Lt.	W	Single
12.	25-15	Craig	.6% W.B.	Numerous -2°	W 15-20 J 20-25	Double

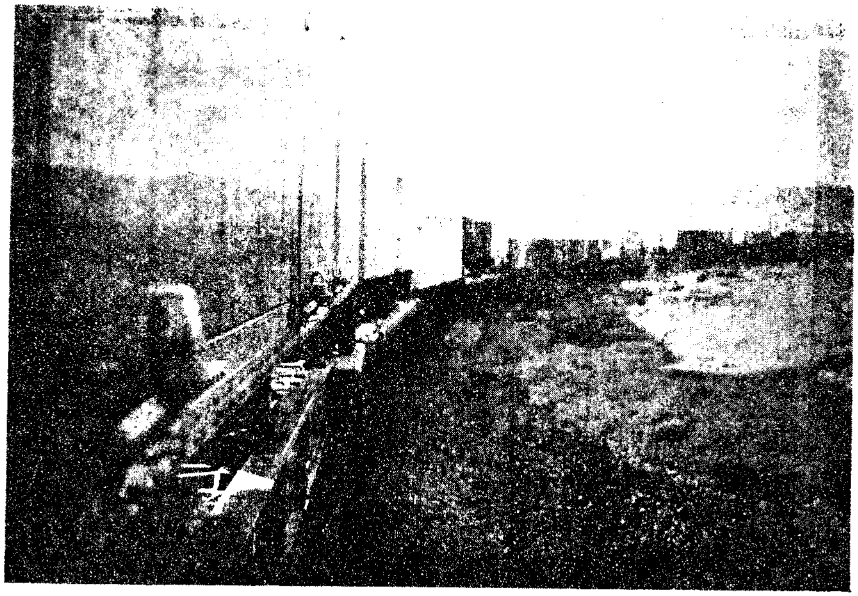


Figure 4-15. Over-the-Road Test Consist

Twelve test zones, listed in Table 4-6, each ten miles long were selected along the AT&SF route which was approximately 1,750 miles long. The selection of the test represents a cross section of track structure and operating conditions typically encountered in intermodal service west of the Mississippi. Six OTR tests were conducted between December 1976 and February 1977 after the test consist had accumulated between 10,000 and approximately 20,000 miles. This was done to collect acceleration data on each flatcar system (carbody, axles and loads) with all possible load configurations.

Although the instrumentation of each OTR consist was identical to that of the RVT consist, the test procedure was modified due to the nature of the operation; i.e., there was absolutely no control over test conditions such as train handling and speed during the OTR test series. Therefore, the data was collected in the following manner. Between five and ten miles before each test zone the data acquisition system was given a final checkout and placed on stand-by or hold. Due to the operating conditions, this procedure was conducted under way and in the event of an instrumentation failure exterior to the data acquisition car no recourse was available. As a result, a channel of data could be lost but with the redundant compliment of accelerometers a solution of the modes was still possible. The impact of one accelerometer loss was, therefore, not catastrophic, but a great deal of attention was given to accelerometer survivability, hence, the use of the mechanical isolators.

An observer in the data acquisition car alerted the crew when the consist was one mile from the test zone and based on speed, displayed directly above the computer console, a count down was initiated. The observer then called out the test zone entry and the data acquisition system was enabled.

The exit of the test zone was similarly noted and the data acquisition system was returned immediately to hold. The data tape was then removed from the tape deck; marked with the date, test zone and OTR run number; and stored for off-line processing.

4.3 DATA ANALYSIS

4.3.1 CARBODY EQUATIONS

The Lightweight Flatcar System was partitioned into three subsystems; the carbody, the load and the axle, to facilitate the use of modal analysis as explained. Six rigid-body and four elastic-body components of acceleration are used in the modal analysis of each subsystem. The three rigid-body translational components along the x,y, and z axes, respectively, are called longitudinal, sway and bounce, respectively. The three rigid-body rotational accelerations around the x,y, and z axes, respectively, are called roll, pitch and yaw.

Four elastic-body modes are also included: first and second vertical bending about the y-axis; and first and second torsion about the x-axis. (See Figure 4-16). The mode shape functions are:

$$B_{yz1}(x) = 1 + b_2x^2 + b_4x^4 \tag{1}$$

$$B_{yz2}(x) = x + b_3x^3 \tag{2}$$

$$T_{x1}(x) = x + c_3x^3 \tag{3}$$

$$T_{x2}(x) = 1 + c_2x^2 \tag{4}$$

The resulting equations relating local accelerations to modal components of acceleration are:

$$\ddot{x}(y,z,t) = \ddot{x}_o(t) + \ddot{\phi}_o(t)z - \ddot{\psi}_o(t)y \tag{5}$$

$$\ddot{y}(x,z,t) = \ddot{y}_o(t) - \ddot{\theta}_o(t)z + \ddot{\psi}_o(t)x - \tau_{x1}(t)(x + c_3x^3) - \tau_{x2}(t)(1 + c_2x^2) \tag{6}$$

$$\ddot{z}(x,y,t) = \ddot{z}_o(t) + \ddot{\theta}_o(t)y - \ddot{\phi}_o(t)x + \beta_{yz1}(t)(1 + b_2x^2 + b_4x^4) + \beta_{yz2}(t)(x + b_3x^3) + \tau_{x1}(t)(x + c_3x^3)y + \tau_{x2}(t)(1 + c_2x^2)y \tag{7}$$

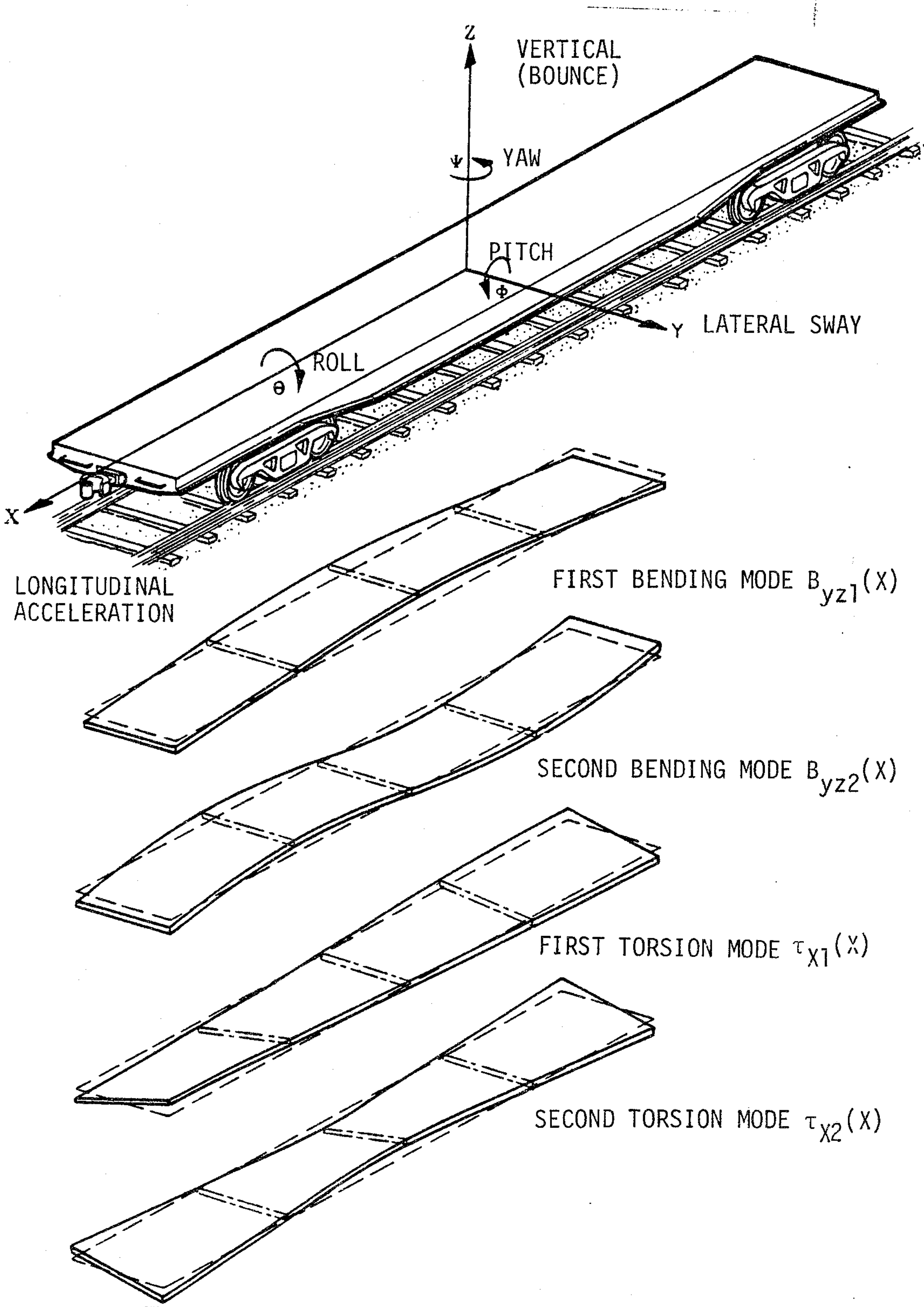


Figure 4-16. Rigid and Elastic Modes

The final equations, in the order in which they were solved, are:

$$\ddot{z}(x,y,t) = A_o(t) + A_1(t)x + A_2(t)x^2 + A_3(t)x^3$$

$$+ A_4(t)x^4 + A_5(t)xy + A_6(t)x^2y$$

$$+ A_7(t)x^3y + A_8(t)y \tag{8}$$

in which $A_o(t) \equiv \ddot{z}_o(t) + \beta_{yz1}(t)$

$$A_1(t) \equiv \beta_{yz2}(t) - \ddot{\phi}_o(t)$$

$$A_2(t) \equiv \beta_{yz1}(t)b_2$$

$$A_3(t) \equiv \beta_{yz2}(t)b_3$$

$$A_4(t) \equiv \beta_{yz1}(t)b_4$$

$$A_5(t) \equiv \tau_{x1}(t)$$

$$A_6(t) \equiv \tau_{x2}(t)c_2$$

$$A_7(t) \equiv \tau_{x1}(t)c_3$$

$$A_8(t) \equiv \ddot{\theta}_o(t) + \tau_{x2}(t)$$

$$\ddot{y}(x,z,t) = A_{10}(t) + A_{11}(t)x + A_{12}(t)x^2$$

$$+ A_{13}(t)x^3 + A_{14}(t)z \tag{9}$$

in which $A_{10}(t) \equiv \ddot{y}_o(t) - \tau_{x2}(t)$

$A_{11}(t) \equiv \ddot{\psi}_o(t) - \tau_{x1}(t)$

$A_{12}(t) \equiv -\tau_{x2}c_2$

$A_{13}(t) \equiv -\tau_{x1}(t)c_3$

$A_{14}(t) \equiv -\ddot{\theta}_o(t)$

$$\ddot{x}(y,z,t) = A_{20}(t) + A_{21}(t)z + A_{22}(t)y \tag{10}$$

in which $A_{20}(t) \equiv \ddot{x}_o(t)$

$A_{21}(t) \equiv \ddot{\phi}_o(t)$

$A_{22}(t) \equiv -\ddot{\psi}_o(t)$

Table 4-7 lists the time series coefficients (A_i) and their appropriate spatial coordinates.

It is necessary to decouple the rigid and elastic components of the A coefficients in only the xy-plane since a piecewise solution approach is being used. The coupled components are:

$$A_o(t) = \ddot{z}_o(t) + \beta_{yz1}(t) \tag{11}$$

$$A_1(t) = \beta_{yz2}(t) - \ddot{\phi}_o(t) \tag{12}$$

$$A_8(t) = \ddot{\theta}_o(t) + \tau_{x2}(t) \tag{13}$$

To do the uncoupling, set the net work due to the elastic components equal to zero. This results in the expressions:

TABLE 4-7

NOMENCLATURE

Time Series Coefficients	Powers of Spatial Coordinate
$A_o(t)$	$x^o = 1$
$A_1(t)$	x^1
$A_2(t)$	x^2
$A_3(t)$	x^3
$A_4(t)$	x^4
$A_5(t)$	x_y
$A_6(t)$	x^2y
$A_7(t)$	$x^3 y$
$A_8(t)$	y
$A_{10}(t)$	$x^o = 1$
$A_{11}(t)$	x
$A_{12}(t)$	x^2
$A_{13}(t)$	x^3
$A_{14}(t)$	z
$A_{20}(t)$	$x^o = 1$
$A_{21}(t)$	z
$A_{22}(t)$	y

$$\ddot{z}_o(t) = \Lambda_o(t) + \frac{M_1}{M_o} \Lambda_2(t) + \frac{M_2}{M_o} \Lambda_4(t) \qquad (14)$$

$$\ddot{\phi}_o(t) = A_1(t) - \frac{M_2}{M_1} A_3(t) \qquad (15)$$

$$\ddot{\theta}_o(t) = A_8(t) + \frac{H_1}{H_o} A_6(t) \qquad (16)$$

where

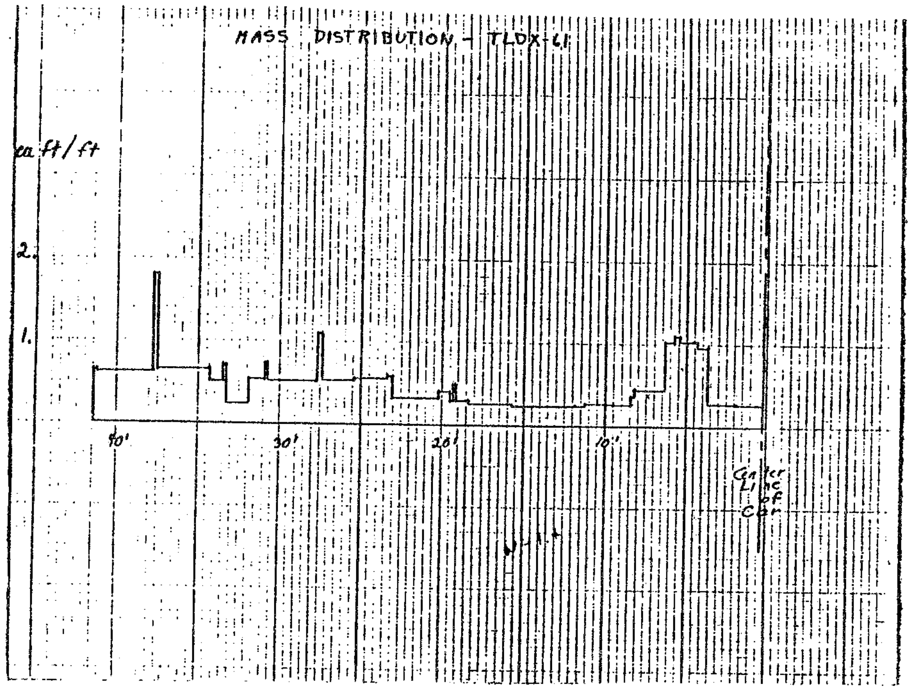
$$M_n = \int_{-L/2}^{L/2} x^{2n} \rho(x)\,dx \qquad n = 0,1,2 \qquad (17)$$

$$H_s = \int_{-L/2}^{L/2} x^{2s} J(x)\,dx \qquad s = 0,1 \qquad (18)$$

The evaluation of Equations (17) and (18) requires knowledge of the mass distribution, $\rho(x)$, and of the polar moment of inertia distribution, $J(x)$.

Exact tabulations of the mass distribution, $\rho(x)$, were made from blue prints. Each car was divided into segments of constant density as small as three inches resulting in the detailed distribution shown in Figure 4-17.

The moments, M_n, are not radically affected by the gross approximation of a homogeneous weight distribution. That is, if $\rho(x)$ is approximated as a constant, ρ_o, (the weight of the car divided by its length) one obtains the results summarized in Table 4-28.



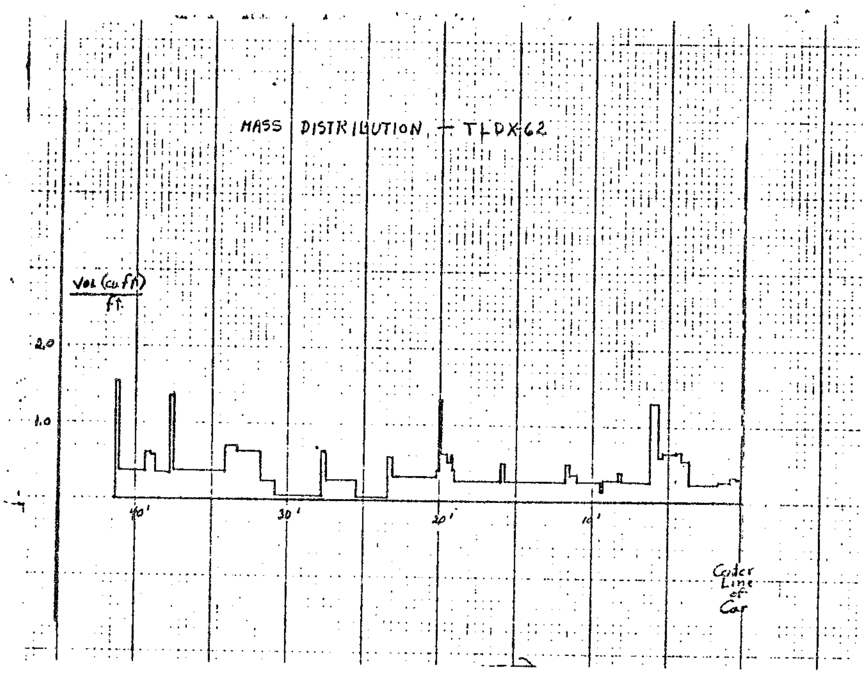


Figure 4-17. Detailed Distributions

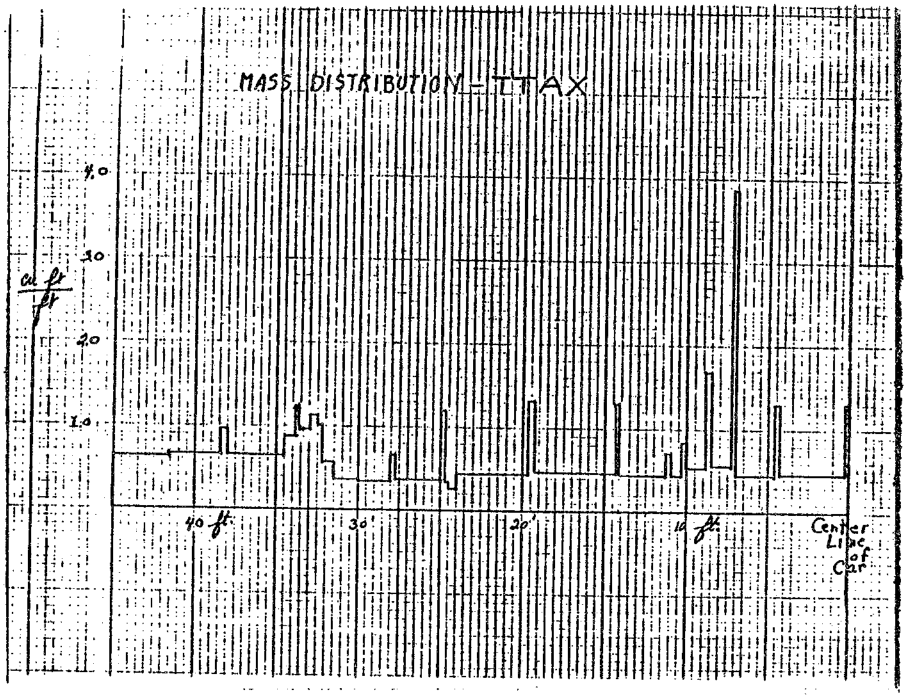


Figure 4-17. Detailed Distribution (cont)

TABLE 4-8

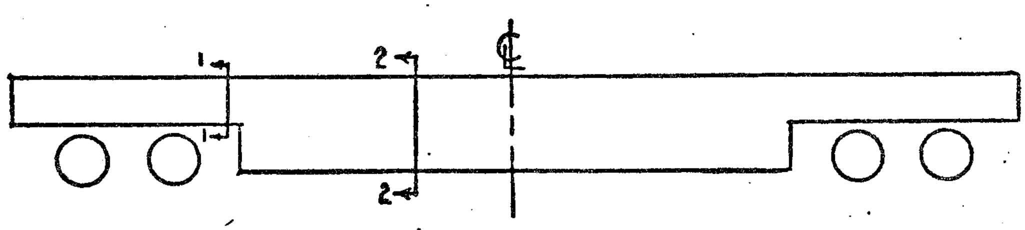
APPROXIMATION OF WEIGHT DISTRIBUTION

	$\rho(x)$*	ρ_o*	Difference
M_o	19.86	19.86	0
M_1	6.52	6.62	1.5%
M_2	3.21	3.97	24%
M_3	2.21	2.84	28%

*$\rho(x)$ and ρ_o are in units of cubic feet/unit length where one unit length is half car body, L/2. This unit was chosen to keep these parameters within a reasonable range (<10) and are readily converted to units of weight/foot by multiplying by the weight density of steel (490 lb/ft^3). Note also M_o as given above is for a quarter car.

Considering the effort to obtain $\rho(x)$, which was considerable, the above results show only marginal justification. The work required to obtain a detailed distribution of the polar mass moment of inertia, J(x), would be even greater. Therefore, an approximation was made, based on blue prints for all three cars under study. The cars are comprised basically of two portions of constant moment of inertia (see Figure 4-18). The portion over the bolster is characterized by a moment of inertia significantly smaller than that of the mid-car section (approximately 40 percent in the case of the TLDX-61 flatcar).

The corresponding step moment of inertia distribution for the TLDX-61 car is shown in Figure 4-19 with the abscissa non-dimensionalized by the carbody half length.



$$J(x)_{1-1} = \text{Constant}$$

$$J(x)_{2-2} = \text{Constant}$$

$$J_{1-1} < J_{2-2}$$

Figure 4-18. Separation of the Polar Moment of Inertia

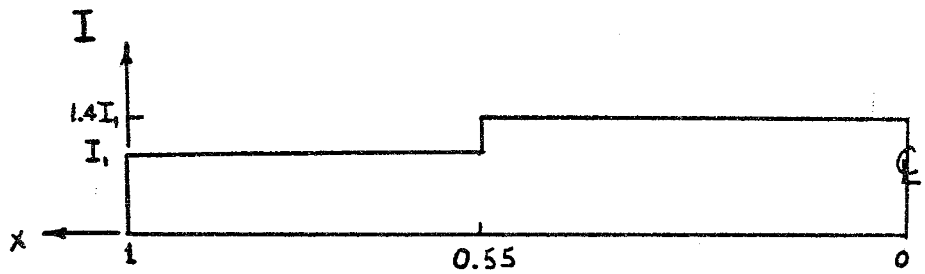


Figure 4-19. Polar Moment of Inertia Distribution

Integrating the moment of inertia distribution (Equation (18)), the value of H_1/H_0 is found to be 0.30. In order to ascertain the effect of detail in the moment distribution on this ratio one may make use of Table 4-8. The ratio of M_1/M_0 is 0.33 or only nine percent larger than H_1/H_0. Although $\rho(x)$ and $J(x)$ are most likely dissimilar, the difference illustrated above indicates the relative insensitivity of the ratio of the first two integrals to detail in distribution. Therefore, one may conclude that the step distribution for the polar mass moment of inertia is an acceptable approximation.

As explained in Section 2.0, the mode shape coefficients are calculated using frequency domain techniques.

$$b_2 = \frac{G_{A_2(t)\beta_{yz1}(t)}}{G_{\beta_{yz1}(t)\beta_{yz1}(t)}} \tag{19}$$

$$b_3 = \frac{G_{A_3(t)\beta_{yz2}(t)}}{G_{\beta_{yz2}(t)\beta_{yz2}(t)}} \tag{20}$$

$$b_4 = \frac{G_{A_4(t)\beta_{yz1}(t)}}{G_{\beta_{yz1}(t)\beta_{yz1}(t)}} \tag{21}$$

$$c_2 = \frac{G_{A_6(t)\tau_{x2}(t)}}{G_{\tau_{x2}(t)\tau_{x2}(t)}} \tag{22}$$

$$c_3 = \frac{G_{A_7(t)\tau_{x1}(t)}}{G_{\tau_{x1}(t)\tau_{x1}(t)}} \tag{23}$$

where G_{pq} is the cross spectral density (CSD) of p with respect to q, and G_{pp} is the power spectral density (PSD) of p.

The lateral acceleration equation, Equation (9), is solved by matrix inversion as described in Section 2.0. Values of the A-coefficients are obtained point-by-point in the time domain using the measured local accelerations as input data. No decoupling is required because the values of $\tau_{x1}(t)$ and $\tau_{x2}(t)$ are already known, having been established above in the vertical axis solution.

The longitudinal equation, Equation (10), is solved directly on a point-by-point basis since the coefficients A_{21} and A_{22} are already known, thus, completing the solution of the modal coordinates of the carbody.

4.3.2 LOAD EQUATIONS

The local-to-modal transformation equations for the load are:

$$\ddot{x}(y,z,t) = \ddot{x}_o(t) + \ddot{\phi}_o(t)z - \ddot{\psi}_o(t)y \tag{24}$$

$$\ddot{y}(x,z,t) = \ddot{y}_o(t) - \ddot{\theta}_o(t)z + \ddot{\psi}(t)x$$

$$+ \beta_{zy1}(t)B_{zy1}(x) \tag{25}$$

$$\ddot{z}(x,y,t) = \ddot{z}_o(t) - \ddot{\phi}_o(t)x + \ddot{\theta}_o(t)y \tag{26}$$

Lateral bending about the z-axis is the one elastic component included in the modal analysis. The associated mode shape function is

$$B_{zy1}(x) = 1 + d_2x^2 \tag{27}$$

The decoupling relation, based on equilibrium of inertia forces is

$$\int_{-L/2}^{L/2} \rho(x)B_{zy1}(x)dx = 0 \; . \tag{28}$$

A bulk load may reasonably be approximated by a uniform mass distribution ($\rho(x) \equiv 1$). Therefore, Equation (28) becomes

$$\int_{-L/2}^{L/2} B_{zy1}(x)dx = \int_{-L/2}^{L/2} (1 + d_2x^2)dx = 0 \tag{29}$$

which has the solution

$$d_2 = -\frac{12}{L^2} \tag{30}$$

Accordingly,

$$B_{zy1} \equiv 1 - \frac{12}{L^2} x^2 \tag{31}$$

and can be used as an independent spatial coordinate.

One linear acceleration in the longitudinal direction, four accelerations in the lateral direction, and three in the vertical direction were measured. The set of equations required to model this system is, therefore, comprised of one Equation (24), four Equations (25) and three Equations (26). The resulting set of equations may be written in matrix form as

$$\{a_m\} = [X]\{A\} \tag{32}$$

where

$$\{a_m\} \equiv \begin{Bmatrix} \ddot{x}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \\ \ddot{y}_4 \\ \ddot{y}_5 \\ \ddot{z}_6 \\ \ddot{z}_7 \\ \ddot{z}_8 \end{Bmatrix} \qquad \{A\} \equiv \begin{Bmatrix} \ddot{x}_o \\ \ddot{y}_o \\ \ddot{z}_o \\ \ddot{\theta}_o \\ \ddot{\phi}_o \\ \ddot{\psi}_o \\ \beta_{zy1}(t) \end{Bmatrix}$$

$$[X] \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & z_1 & -y_1 & 0 \\ 0 & 1 & 0 & -z_2 & 0 & x_2 & B_{zy1}(x_2) \\ 0 & 1 & 0 & -z_3 & 0 & x_3 & B_{zy1}(x_3) \\ 0 & 1 & 0 & -z_4 & 0 & x_4 & B_{zy1}(x_4) \\ 0 & 1 & 0 & -z_5 & 0 & x_5 & B_{zy1}(x_5) \\ 0 & 0 & 1 & y_6 & -x_6 & 0 & 0 \\ 0 & 0 & 1 & y_7 & -x_7 & 0 & 0 \\ 0 & 0 & 1 & y_8 & -x_8 & 0 & 0 \end{bmatrix}$$

Note that the elements of the A-vector are subscripted consecutively regardless of orientation. This is done to identify the spatial coordinate of a particular accelerometer with its respective measurement. This is illustrated schematically for both the trailer and container in Figure 4-20.

Equation (32) represents a system of eight equations with seven unknowns, the elements of the A-vector. These unknowns are fitted to the measured data in exactly the same way as the vertical carbody modal coordinates were. Thus:

$$[X]^T\{a_m\} = [X]^T\{a\} \ . \tag{33}$$

Next, Equation (32) is substituted into this expression to obtain:

$$[X]^T[X]\{A\} = [X]^T\{a\} \ ; \tag{34}$$

defining a square matrix S as

$$[S] \equiv [X]^T[X] \ ,$$

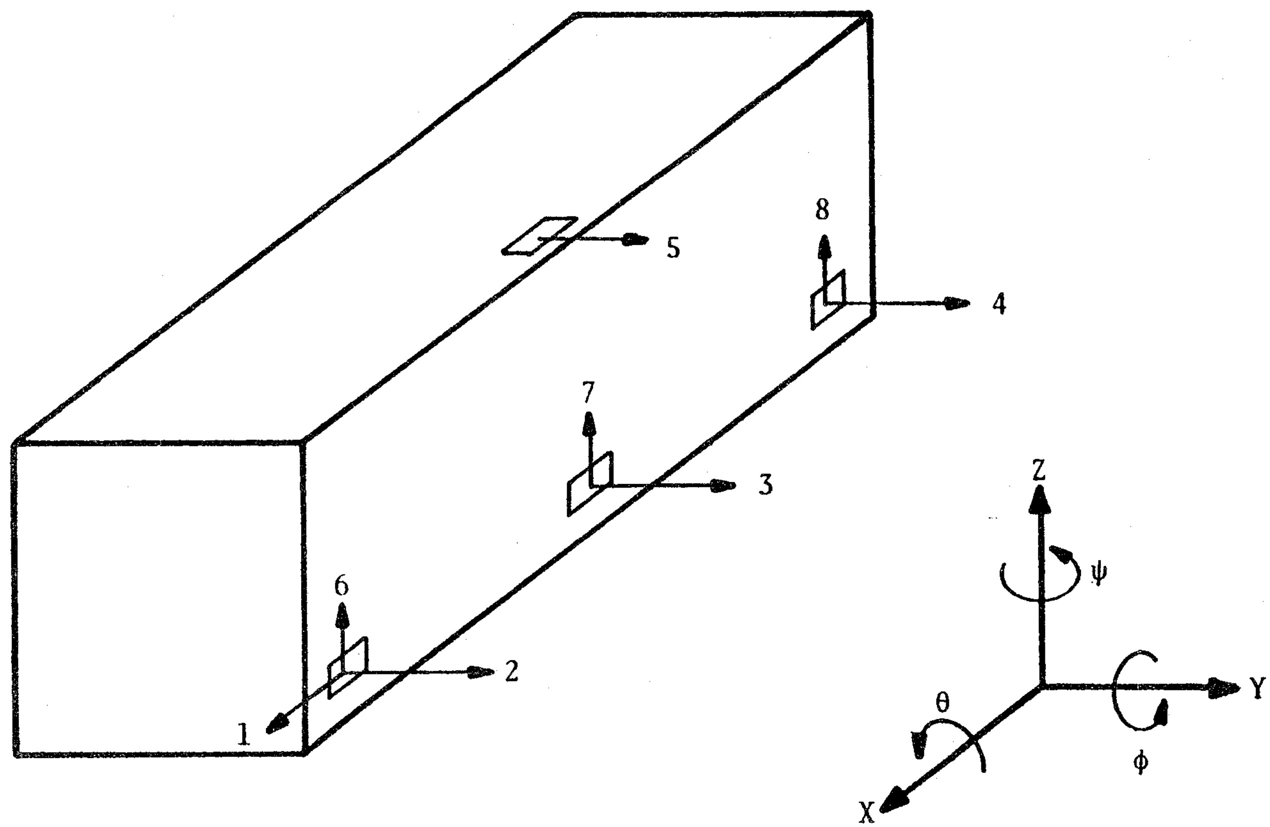
Equation (34) can be written as:

$$[S]\ \{A\} = [X]^T\{a\} \ . \tag{35}$$

Multiplying each side of Equation (33) by the inverse of the S-matrix, Equation (33) becomes

$$\{A\} = [S]^{-1}[X]^T\{a\} \ . \tag{36}$$

Thus, the A-vector is the best fit of the modal coordinates to the measured accelerations on the loads. This completes the solution of the load equations.



X - longitudinal

Y - lateral

Z - vertical

θ - roll

ϕ - pitch

ψ - yaw

Figure 4-20. Load Axes and Accelerometer Locations

4.3.3 AXLE EQUATIONS

The axle modal components are the same as the load components with the exception of $\ddot{\phi}_o$. Obviously rotation about the y-axis will not enter into accelerations on the axle, and therefore, only the five modal accelerations $\ddot{x}_o(t)$, $\ddot{y}_o(t)$, $\ddot{z}_o(t)$, $\ddot{\theta}_o(t)$ and $\ddot{\psi}_o(t)$ must be determined.

Five accelerometers were mounted over the journal bearings to obtain the five observations required to solve for the five unknown modal accelerations. A tri-axial package was mounted at one end and a bi-axial at the other. Tri-axial means that acceleration was measured along the three axes of translation and the bi-axial package only two, longitudinal and vertical (see Figure 4-11).

These accelerations can be written in terms of modal accelerations with the origin located at the axle center. Thus,

$$\ddot{x}_1(t) = \ddot{x}_o(t) - \ddot{\psi}_o(t)y_H \tag{37}$$

$$\ddot{x}_2(t) = -\ddot{x}_o(t) - \ddot{\psi}_o(t)y_H \tag{38}$$

$$\ddot{y}_3(t) = \ddot{y}_o(t) - \ddot{\theta}_o(t)z_L - \ddot{\psi}_o(t)x_L \tag{39}$$

$$\ddot{z}_4(t) = \ddot{z}_o(t) + \ddot{\theta}_o(t)y_V \tag{40}$$

$$\ddot{z}_5(t) = \ddot{z}_o(t) - \ddot{\theta}_o(t)y_V \tag{41}$$

where y_H = y distance to c.g. longitudinal accelerometers = 3.7 ft.

z_L = z distance to c.g. of lateral accelerometers = 0.5 ft.

x_L = x distance to c.g. of lateral accelerometers = 0.6 ft.

y_V = y distance to c.g. of vertical accelerometers = 3.7 ft.

$\ddot{x}_1$ and $\ddot{x}_2$ = longitudinal accelerations at positions 1 and 2.

$\ddot{y}_3$ = lateral accelerations at Position 3.

$\ddot{z}_4$ and $\ddot{z}_5$ = vertical accelerations at Positions 4 and 5.

Note that due to symmetry the coordinates required are the x and z-coordinates of the lateral accelerometer, the y-coordinate of the longitudinal accelerometer and the y-coordinate of the vertical accelerometer.

These five equations (Equations 37 through 41) containing five unknowns are solved exactly to obtain the modal accelerations as follows:

$$\ddot{x}_o(t) = (\ddot{x}_1 - \ddot{x}_2)/2 \tag{42}$$

$$\ddot{y}_o(t) = \ddot{y}_3 + \ddot{\theta}_o z_L + \ddot{\psi}_o x_L \tag{43}$$

$$= \ddot{y}_3 + [(\ddot{z}_4 - \ddot{z}_5)z_L/y_V$$

$$- (\ddot{x}_1 + \ddot{x}_2)x_L/y_H]/2 \tag{43}$$

$$\ddot{z}_o(t) = (\ddot{z}_4 + \ddot{z}_5)/2 \tag{44}$$

$$\ddot{\theta}_o(t) = (\ddot{z}_4 - \ddot{z}_5)/2y_V \tag{45}$$

$$\ddot{\psi}_o(t) = -(\ddot{x}_1 + \ddot{x}_2)/2y_H \tag{46}$$

This completes the solution of the axle modal components.

4.4 DATA PROCESSING

Data processing involves two basic steps. First,the data is transformed from measured linear accelerations to mode accelerations. This simply converts one set of time series to another. Second, the time series must be presented. This involves the selection and implementation of suitable analytical techniques and the choice of graphic output formats.

Prior to any actual processing, however, it is always advisable to reproduce the data and do some preliminary analysis and check its validity. This was done in the LWFC Program by reconstructing an analog reproduction of each measured acceleration and comparing it with the strip chart recordings made during the test. Also, some basic calculations were performed to determine if the magnitude and frequency content of the signals which were compared with physically realizable and expected values. That is, some judgement was made as to whether a given structure could actually undergo accelerations of a certain magnitude at the indicated frequency.

Once the data was validated, the modal transformation was made. First the data from the raw data tapes were reformatted to a 16-bit floating point format, scaled to dimensional units, inverse filtered to remove the effects of the 1.6-Hz filter discussed in Section 4.3, and debiased (i.e., the mean was removed). The transformation was made using the algorithms outlined in the previous section, 4.3. The software used to perform this transformation is flow charted in Appendix A followed by a listing of the program and all routines.

The first and perhaps most powerful technique to summarize and display the results of the mode transformation was the Power Spectral Density (PSD). This is a common and widely used method to examine the frequency content of a signal and it has the additional virtue of providing a clear-cut graphic display.

For the purposes of the LWFC Program the PSD was calculated in blocks of four seconds which at a rate of 128 samples per second corresponds to 512 samples. Each set of 512 samples were Fourier transformed to produce a set of 256 complex values which represent the amplitude at 256 equally spaced frequencies. The frequency increment (Δf) is

$$\Delta f = \frac{1}{N\Delta t} \tag{47}$$

where N is the number of samples (512) and Δt is the sample interval (1/128 second). Since Δf is 1/4 Hz, the highest frequency, (F), (the Nyquist frequency), is:

$$F = \left(\frac{N}{2}\right)\Delta f = 64 \text{ Hz} \tag{48}$$

The Fourier series for each record is multiplied by its complex conjugate and scaled by Δf to yield the PSD. The PSD gives the mean square acceleration per Δf which is defined as G_i ($i = 1, 2, \ldots, 256$) for each Δf. An average PSD is calculated by adding the calculated G_i for each second data record on a bin-by-bin basis and dividing each bin by the total number of records stacked. The PSD estimate is then plotted in a linear-linear graph as shown in Figure 4-2. Note that the abscissa has a maximum value of only 30 Hz as compared to the maximum realizable frequency of 64 Hz. For this test program, only the frequency band between 0 and 30 Hz was of interest and the data was heavily filtered above 30 Hz. The scale for the ordinate is given by the value YMAX which is in units of g^2/Hz, and is the maximum value of the ordinate (the minimum is zero).

Integration of the PSD yields the mean square value, and the square root of this quantity is the root mean square (rms). This integration is accomplished by numerical means and since Δf is constant, the value of rms acceleration z_{rms} may be expressed as

$$z_{rms} = (\Delta f \sum_i G_i)^{\frac{1}{2}} \tag{49}$$

This summation is performed only over the frequency band of interest (0 to 30 Hz).

A simple example, used to verify the software, illustrates the utility of the PSD. For this purpose, a sine wave was artificially generated with a peak amplitude (A) of 1.0g and a frequency of 1.0 Hz. The PSD of this signal is shown in Figure 4-21.

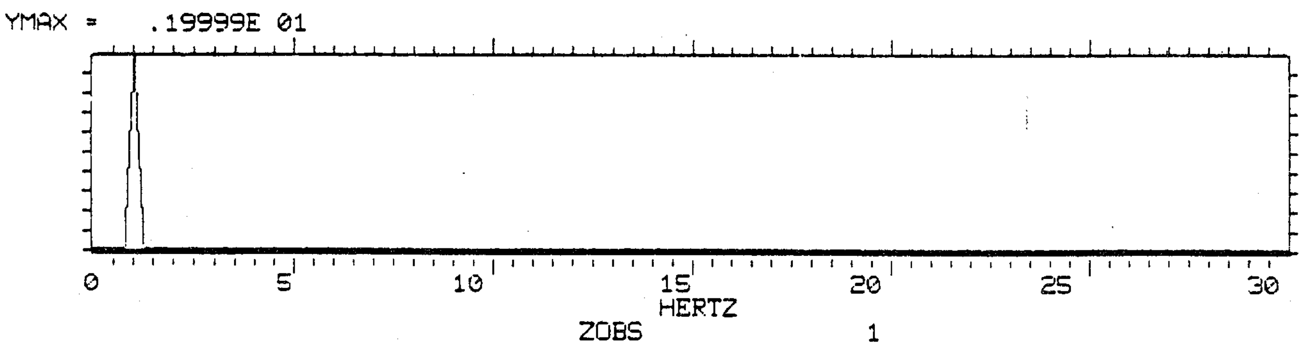


Figure 4-21. PSD of Sine Wave

The rms value of a sine wave is the peak amplitude over the square root of two. Furthermore, because there is only one non-zero G_i in the PSD of a sine wave (that G_i which is associated with the Δf or bin in which the frequency of the wave falls), the above equation yields

$$G = A^2/(2\Delta f) \tag{50}$$

The values of A(1.0g) and Δf(¼Hz) result in a value of G which in this case corresponds to YMAX in Figure 4-21.

$$G = YMAX = 2 \tag{51}$$

This value agrees to four places with Figure 4-21. This demonstrates how a PSD may be interpreted to provide an estimate of the predominate frequency and amplitude of an arbitrary time series such as a mode acceleration history.

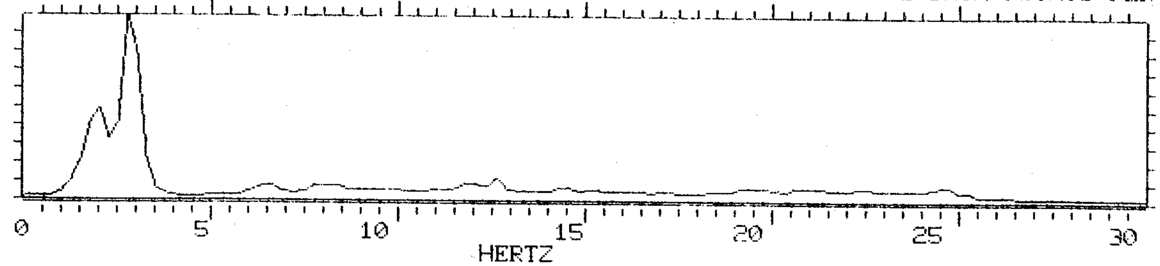
An example of a typical result obtained during the LWFC Program is shown in Figure 4-22, where three modes of a flat-car are shown. The bounce mode is characterized by a narrow band signal and two peaks. The predominant peak at 2.75 Hz coincides with the natural frequency of a spring-mass system with mass and spring constants equivlent to those of the car. The first bending mode has a narrow band of activity at 3 Hz. This coincides with the natural frequency of a uniform beam equal to that of the car. The bending energy around 20 Hz is due to a number of contributing causes. Among them is the method of sampling which did not provide for simultaneous sampling of all channels.

This may be due to a third bending mode since the first and third modes are odd functions; also, the third bending mode would theoretically lie in the region of 20 Hz. The pitch mode shows the widest band response but does possess a predominant frequency in agreement with a spring-mass model.

The preceeding has dealt with a PSD which is the average of data over an entire test zone. This encompasses between 100 and 600 seconds of data depending on speed and test zone length. A logical question arises: Is this a valid average? One way to address this question is through a time history of the rms value. For this purpose, a graphical display of

CAR BODY
TTAX ZONE 8 LWFC 8 2LC

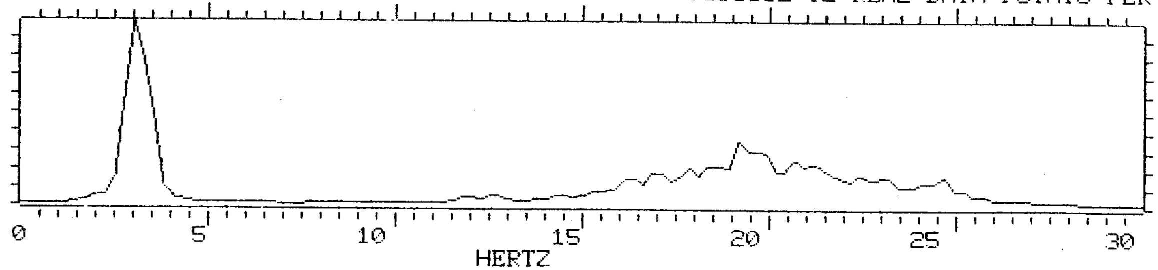
YMIN = .00000E 00 YMAX = .13418E-02 X-AXIS SCALE .19866E 02 REAL DATA POINTS PER INCH



BOUNCE MODE

TOTAL POWER IN 30 HERTZ BAND .271240E-02 G'S MEAN SQUARE
.520807E-01 RMS

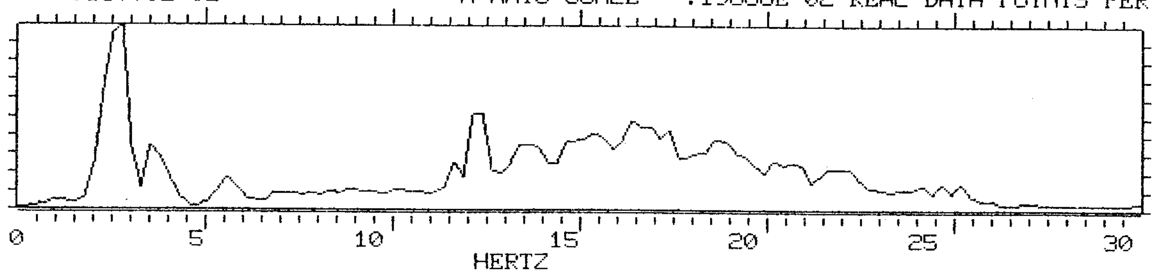
YMIN = .57846E-05 YMAX = .29419E-02 X-AXIS SCALE .19866E 02 REAL DATA POINTS PER INCH



FIRST BENDING MODE

TOTAL POWER IN 30 HERTZ BAND .813697E-02 G'S MEAN SQUARE
.902051E-01 RMS

YMIN = .00000E 00 YMAX = .16770E-02 X-AXIS SCALE .19866E 02 REAL DATA POINTS PER INCH



PITCH MODE

TOTAL POWER IN 30 HERTZ BAND .878457E-02 (RAD/SEC/SEC) MEAN SQUARE
.937260E-01 RMS

Figure 4-22. Typical PSD Results

of rms acceleration history was developed which showed not only the rms value of the entire frequency band of interest, 0-30 Hz, but also the four octaves in the 0-30 Hz band. The four octaves are centered at 2, 4, 8, and 16 Hz. Table 4-8 summarizes the lower, f_l, center, f_c, and upper frequency, f_u, of each octave.

TABLE 4-9

OCTAVE FREQUENCY SUMMARY

OCTAVE	f_l (Hz)	f_c (Hz)	f_u (Hz)
1	2.33	2	2.67
2	2.67	4	5.33
3	5.33	8	10.67
4	10.67	16	21.33

NOTE: Octaves were selected at random to cover as much of the 0 to 30 Hz band as possible.

The software was designed to calculate the rms value of acceleration over time periods of multiples of the basic 4-second data sample block (512 samples at 128 Hz). This provides the ability to filter or smooth the histories to provide a clear account of events within a test zone.

An example of an octave rms history obtained during the OTR test series is presented in Figure 4-23. In this case, the interval of time over which the rms acceleration is calculated is 20 seconds or a sub-average of five 4-second PSD's. For this reason the curves begin at 20 seconds rather than zero. Note also that there are five curves each labeled with an integer between 1 and 5. The first four correspond to the octave numbers given in Table 4-9. The curve labeled 5 is the rms history of the entire frequency band of interest, 0-30 Hz. Immediately below the rms history plot a similar history of speed is also given.

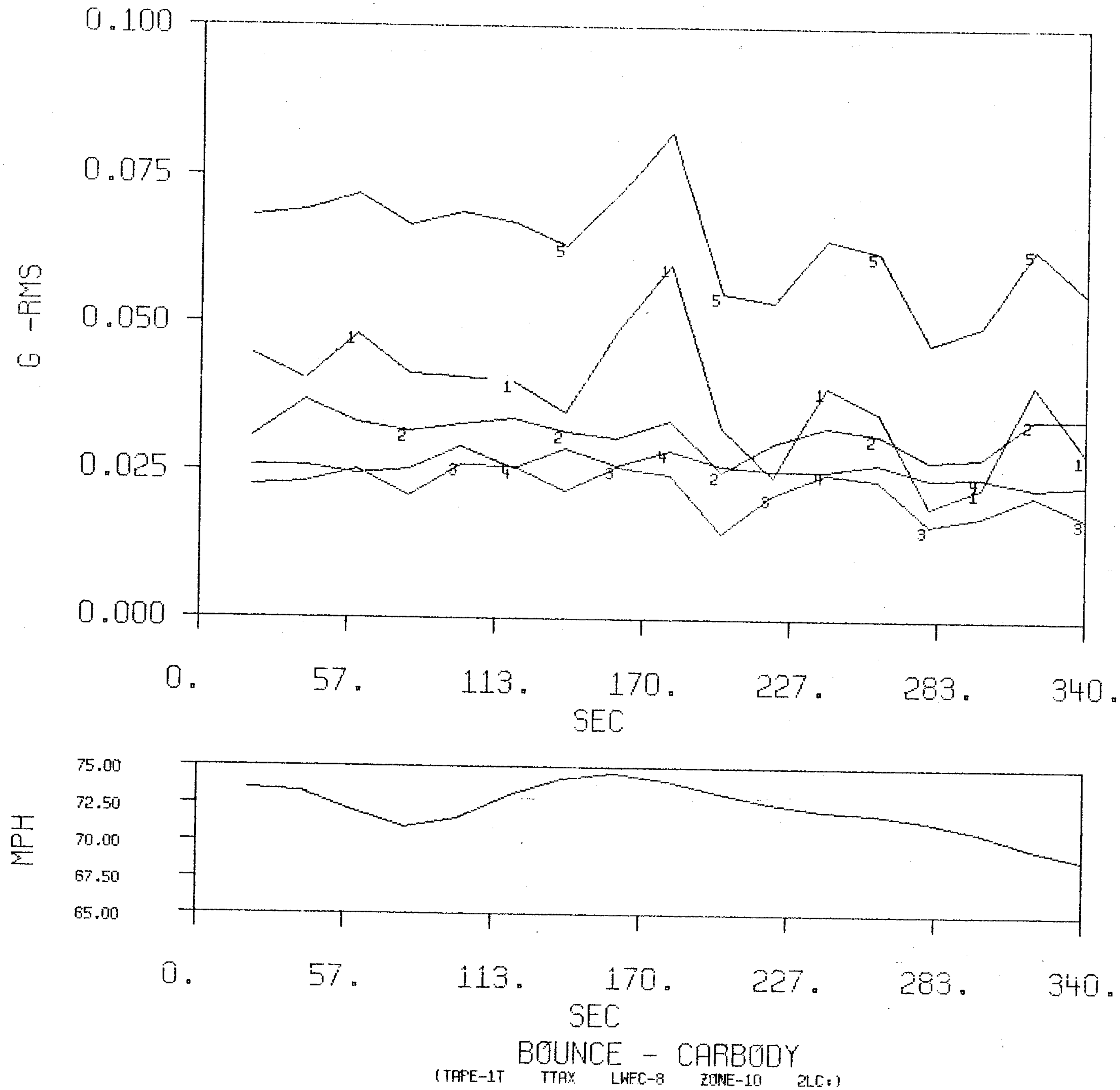


Figure 4-23. Octave RMS History

This particular example illustrates several points determined from the LWFC Program. First, the instantaneous PSD does exhibit some variation with time, a feature which is not evident in the average PSD. Second, the signature or characteristic profile of this PSD does, for the most part, remain constant with time. This is illustrated in Figure 4-23 where the relative importance of each octave is generally the same throughout the test zone. Octave 1 in particular remains dominent and its history follows very closely with that of the wide band rms acceleration. Third, the magnitude of acceleration can be shown to be directly dependent on speed. For example, at 170 seconds, speed is a maximum; the acceleration is also a maximum. This relation was observed throughout the LWFC data.

One additional presentation of the data, which was useful during the LWFC Program, was the Probability Density Function (PDF). Although the PDF is perhaps not as powerful as the PSD in terms of providing analytical insight, it does aid in the assessment of the meaningfulness or validity of the data.

The first step in the calculation of the PDF which strictly speaking is an estimation, is the sorting or classifying of each discrete value of the mode acceleration time series, $\ddot{X}_j$, by amplitude. The result is a histogram which may be thought of as a vector, H_i, with each element representing the number of occurrences falling within an assigned incremental range. The incremental range of each element is referred to as the bin width, W.

$$W = (A_{max} - A_{min})/N \qquad (52)$$

where N is the number of uniform bins and A_{max} and A_{min} are the expected (or known) upper and lower limits respectively of the entire set of data.

Thus, H_1 is the number of times that a mode acceleration sample, X_j satisfies the condition

$$A_{min} \leq X_j \leq A_{min} + W \tag{53}$$

In general, H_i is the number of occurrences for which X_j satisfies the condition

$$A_{min} + (i - 1) W \leq X_j \leq A_{min} + iW \tag{54}$$

Following the completion of the filling of the vector H_i, it is divided by the total number of data samples observed

$$N = \Sigma H_i \tag{55}$$

The probability vector, P_i is obtained from the expression

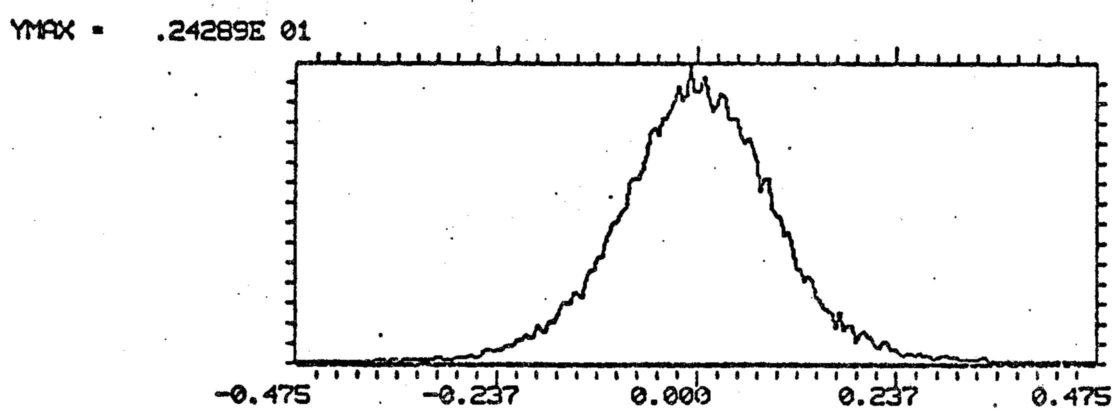
$$P_i = H_i/N \tag{56}$$

The probability density vector, D_i, is obtained by dividing each probability P_i by the bin width W.

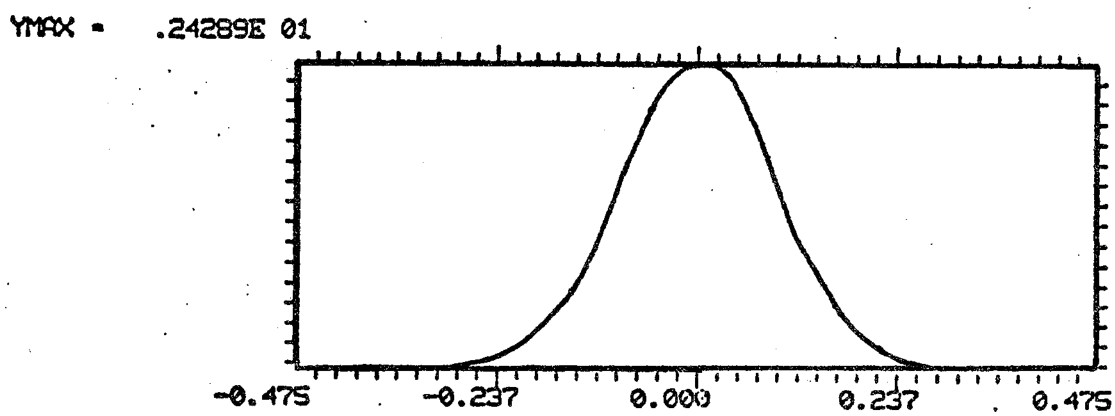
$$D_i = P_i/W \tag{57}$$

Thus, the probability density estimate indicates the percent probability of the acceleration amplitude lying in a given incremental range. Note that by the above definitions, the area under the PDF is equal to one which simply means that all data in a given set have been considered.

The PDF estimate, an example of which is shown in Figure 4-24 is useful in determining absolute maximum acceleration and relative severity. The PDF can also be used to ascertain



(a) Estimated Distribution



(b) Ideal Normal Distribution

Figure 4-24. Comparison of Ideal and Estimated PDF Distributions the validity of the sample length. That is, from the Central Limit Theorem it is known that the PDF of a set of independent random variables will be asymptotically Gaussian or normal. Inspection of the shape of the PDF estimates can be qualitatively used to determine if this condition has been met. Figure 4-24 shows a PDF estimate obtained from actual data and an ideal normal distribution calculated with the same mean and standard deviation.

The standard deviation, σ, is calculated from the expression

$$\sigma^2 = \left(\sum_{i=1}^{N} (X_i - \mu)^2 \right) / (N-1)$$

where $$\mu = \sum_{i=1}^{N} X_i / N$$

The 95 and 99 percentile levels, denoted L_{95} and L_{99} respectively, are determined from the probability vector as follows:

$$0.95 = \sum_{i=1}^{L_{95}} P_i \qquad \text{and } 0.99 = \sum_{i=1}^{L_{99}} P_i \ .$$

$$\sigma^2 = \frac{N\left(\sum X_i^2\right) - \left(\sum X_i\right)^2}{N(N-1)}$$

These percentile levels are useful in estimating the extreme of the mode acceleration history. This information can be used to assess the effect of impacts occuring during operation.

5.0 CONCULSIONS AND RECOMMENDATIONS

5.1 GENERAL CONCLUSIONS

The modal analysis of the output of an accelerometer array has proven to be a valuable technique for the study of elastic structures under actual operating conditions (in particular rail vehicles). Modal analysis, by its very nature, indicates the natural modes of the structure (the six rigid-body modes and the elastic-body modes). Locally measured accelerations are difficult to interpret, but when transformed into kinematically equivalent modal components, the difficulty in interpretation is greatly simplified. Modal accelerations are easy to visualize and as such engage the dynamic analysts power of conception and intuition. Modal analysis may well provide the most direct method for understanding the dynamic response of an elastic structure.

Another advantage of modal analysis is that it provides an objective and complete comparison of the dynamic response of structures which are markedly different is design. The Lightweight Flatcar Program summarized in Section 4.0 is an example of the use of modal analysis. A major objective of that program was the comparative evaluation of a radical new flatcar design, in which the major longitudinal members were outboard, with a conventional and time-proven flatcar design having a single beam on the centerline. Even though the new cars were up to 28 percent lighter, they proved comparable in most respects. In particular, the accelerations imposed on the lading were comparable. The local-to-global transformation of modal analysis suppressed the effects of accelerometer placement and yielded a unified assessment of dynamic response independent of structural configuration.

A third feature of modal analysis is that it permits the construction of an acceleration field for the structure. If this predictive property is used to prepare contour maps or three-dimensional pictorials of the acceleration field, the results are accessible to a broad range of the technical community and its customers. Questions such as the optimal locations of tie downs for lading are easily and directly answered.

It is of interest to note that by straightforward integration of acceleration, velocity and displacement fields for the structure may also be mapped. These variables, along with acceleration are of general interest and are not easily obtained by other means. Simultaneous acceleration, velocity, and displacement data are of central importance to the designer of such things as protective packaging.

Certain technical skills are prerequisites to the application of modal analysis. But such conceptual and mathematical skills are well within the grasp of structural designers and also of civil and mechanical engineers. The need for skill of a broader kind will be presented in the following recommendations. Studies have shown that it is useful to partition a structural system into subsystems each having its own coordinate system and its own set of modal accelerations. Important mathematical and conceptual benefits can be thus derived in the case of applying linear system analysis to explore the interdependence of the subsystems.

The general approach to modal analysis has been formulated in Section 2.3. The notation will be helpful in writing descriptive equations for various applications of modal analysis. However, the user may see fit to simplify it in special cases; he will not have the need to enlarge upon it.

Finally, it may be concluded that the choice and design of the output format is as important as the application of modal analysis methodology in the successful accomplishment of program objectives. This is true because any investigative project, such as the LWFC Program, must provide information in a form that provides insight to the analyst. The output format or any other machine/analyst interface must enable the analyst to come to grips with and extract meaning from rather complex and sometimes voluminous data.

5.2 RECOMMENDATIONS ON PROJECT DESIGN

Modal analysis is very useful in any experiment that involves many independent parameters. In the LWFC Program, for example, independent parameters included vehicle type, lading, speed, track condition, and accumulated mileage. As a result there were 330 possible combinations of these parameters in the RVT series alone. The OTR test series adds over 200 additional combinations for a total of well over 500 situations from which the effects on vehicle dynamic response must be deduced. It should be obvious that some care must be exercised in the design of the experiment to avoid burying the facts in data. Therefore, during the design of a project, such as LWFC, it should always be kept in mind that too much information, even information of exactly the right kind, can be a hinderance.

In the design of a project care must be taken to ensure that the questions posed can be reasonably answered within the scope and framework of the project. As a corollary, non-productive questions must be avoided. These will not only absorb limited resources but can, in fact, confuse important issues. It should be recognized, therefore, that modal analysis can serve as a useful tool in dynamic analysis but does have inherent limitations as do all analytical techniques.

As was mentioned on numerous occasions,modal analysis is most effective when sufficient preliminary groundwork has been laid. This may include analysis, computer simulation or even scale modeling. One potential useful approach would be a preliminary test on a similar structure. In the analysis of rail vehicles, it is recommended that the companion volume of this report (Reference 1) be consulted.

The output format is of singular importance. In the LWFC Program, the average Power Spectral Density (PSD) was found to be quite useful. It might be helpful in future efforts to develop formats that would allow inspection of these results by superimposing the PSD plots. This has been done on at least one other study of dynamics and was found to be very helpful in analysis (Reference 2). A second format which was a great help in evaluating the data was the octave rms history. This data display allows an assessment of the stationarity of the data while at the same time providing an estimate of the amplitude and frequency content of the signal. Thirdly, but to a lesser extent, the Probability Density Function can be useful in data analysis.

In order to achieve an optimum machine/analyst interface on output, the output format and even the entire software package should be designed, in effect, backwards. The question to be posed is: What does the analyst need and what is the best means of displaying it? In addition to this, there is one thing the analyst needs and that is processed information as soon as possible in order to provide the feedback required in the modal analysis methodology (see Section 2.4). This, of course, means that the software must be developed and verified early in the project. The use of sinusoidal pseudo-data was found to be useful for software check-out. In conclusion, the development of software and data display should at least keep pace with instrumentation design and testing to make optimum use of the modal analysis technique.

5.3 RECOMMENDATIONS FOR INSTRUMENTATION DESIGN

Before designing and implementing an instrumentation and data acquisition system, it is advisable to make a complete error analysis. That is, the analysis should be complete in that every component in the data stream should be included, but a relatively simple analysis (such as rms error) will suffice. The result of this analysis will direct attention to those areas or components which degrade system accuracy and may suggest the special calibration procedures or component matching techniques which should be employed.

From the LWFC Program, two specific areas which require a great deal of attention were identified with respect to the application of modal analysis to the data. First, modal analysis is a procedure that depends inherently on the differencing of data signals. Phase and amplitude matching become very important. Although the matching of gain characteristics is often noted, phase matching often is as neglected. It can be shown that the percent error (E) due to a phase shift (ϕ) over a single period of a sinusoid is given by the expression

$$E = 2 \sin (\phi/2).$$

From this expression, it is found that the relatively small phase angle five degrees, which represents only a one percent error in the 360-degree period, results in an error of 8.7 percent. This is a significant contribution to the overall error of the system. Phase shift also results from time delays incurred when sampling several data channels in series. For example, a time delay between samples of one millisecond (1kHz) causes an error of 18.8 percent in a 30-Hz signal.

It should, therefore, be apparent that phase as well as amplitude matching is necessary in the measurement of linear acceleration data which is to be transformed to mode acceleration. It should be equally apparent that sampling should be done simultaneously on all data channels. This is easily accomplished using multiple analog to digital converters which are becoming very cost effective or by using sample and hold amplifiers.

The second finding of the LWFC Program which will benefit other investigations into the dynamics of rail vehicles is that the acceleration levels in the 0-30-Hz frequency band do not increase dramatically with frequency. Thus, above the carbody primary suspension system, the use of the low-frequency, 1.6 Hz, filter is not required. In a similar study (Reference 2) this was, in fact, found to be true for frequencies up to 128 Hz.

The acceleration environment seen at the axle journal bearing may require some further low-pass filtering but probably above 10 Hz. Again it is recommended that phase matching be carefully considered since it may be of interest to estimate the relationship between axle and carbody accelerations. The use of mechanical isolators will, of course, enter into this since they were found to be almost an absolute necessity during the LWFC Program.

5.4 RECOMMENDATIONS FOR FUTURE WORK

During the course of the Lightweight Flatcar Program, certain of the details were dealt with pragmatically in order to accomplish program objectives in the most timely and cost effective manner. There are a limited number of areas for potential improvement in the application of the modal analysis methodology presented in this report.

The first area for improvement would be in the expression of the mode shape function. It was found that two members from each of the nine elastic families, one odd function and one even function, could be accommodated with little difficulty. That is, the use of higher order polynomials did not require further analytical groundwork only computing power. Therefore, depending on the nature of the body or structure of interest, some additional work could be done to express the mode shape functions more efficiently to reduce the required computations. For example, the use of Fourier series may reduce the residual error without increasing the need for computations. There are indeed other functional expressions which may be used to advantage depending on the structure. For example, the Tchebichef polynomial possesses the unique property of evenly distributing the error thus reducing the required number of terms to achieve a given accuracy. Additionally, it appears that the choice of limited polynomials, binomials and trinomials may offer the possibility for the inclusion of four members from each family with a bare minimum of analytical difficulty.

Finally, the specification for redundancy should be more carefully considered. At some point, diminishing returns will be realized with increased instrumentation. Therefore, cost benefits in terms of achieved accuracy should be evaluated. This may require an in-depth analysis using information theory as well as other more economic considerations.

Summarizing, the use of modal analysis provides a powerful tool in the study of the dynamics of an arbitrary body. By careful planning and design as well as sound engineering judgement, a great deal of insight can be obtained which is potentially beneficial to a wide range of people including designers, operators and owners of vehicles.

6.0 REFERENCES

1. Test Results Report Lightweight Flatcar Evaluation Program, Volume II, Ride Vibration and Over-the-Road Testing and Analysis, ENSCO Report No. DOT-FR-79-24, April 1979

2. Test Results Report Piggyback Evaluation Project, Volume II, Full-Scale Test on B&M Railroad, FRA Report No. FRA/ORD 79/05, September 1978

APPENDIX A

LWFC DATA PROCESSING SOFTWARE

APPENDIX A

LWFC DATA PROCESSING SOFTWARE

A.1 DESCRIPTION OF OTR/RVT1/RVT3 PROCESSING SOFTWARE (PROGRAM MODULES)

A.1.1 MAIN DRIVER AND SETUP ROUTINES

LIP - Main driver

MXN - Reads and unpacks data tape.

DUMP - Removes bias, saves data on disk.

TOTLE - Prints title.

A.1.2 CARBODY ROUTINES

TOQS - Forms preliminary X-matrix.

CAR - Reads cards, controls carbody processing.

QINV - Prints dead channel information.

QAZ - Controls computations leading to determination of modal coefficients for vertical acceleration.

SET1 - Forms X-matrix and determines $(X^TX)^{-1}X^T$.

MINV - Computes matrix inverse.

ACF3 - Applies inverse filter (f_c = 1.6 Hz).

FNDA - Solves for the modal coefficients for vertical acceleration.

RES1 - Computes residuals (vertical acceleration).

HELP - Plots observed data and residuals (vertical acceleration).

A2C - Converts modal coefficients to coordinates (vertical acceleration).

MCOR - Computes PSD's and power levels of modal coefficients (vertical acceleration).

MODL - Plots PSD's, print power levels, computes mode shape coefficients.

TITE - Determines plot titles.

LLAT - Does longitudinal and lateral processing, plots PSD's of coordinates.

RES2 - Computes residuals (longitudinal and lateral).

ELP2 - Plots residuals (longitudinal and lateral).

DOE - Discards observed data.

A.1.3 LOADS ROUTINES

SDS - Reads cards, begins load processing.
LDCH - Prints dead channel information.
LD2 - Forms X-matrix and determines $(X^TX)^{-1}X^T$.
SD1 - Controls further computations, computes residuals.
ACF - Applies inverse filter (1.6 Hz).
SPWR1 - Computes PSD's of observed data or residuals.
P2B - Plots observed data and residuals.
CKM - Performs coordinate computations and prints results.
SPWR - Computes PSD's of coordinates.

A.1.4 AXLE ROUTINES

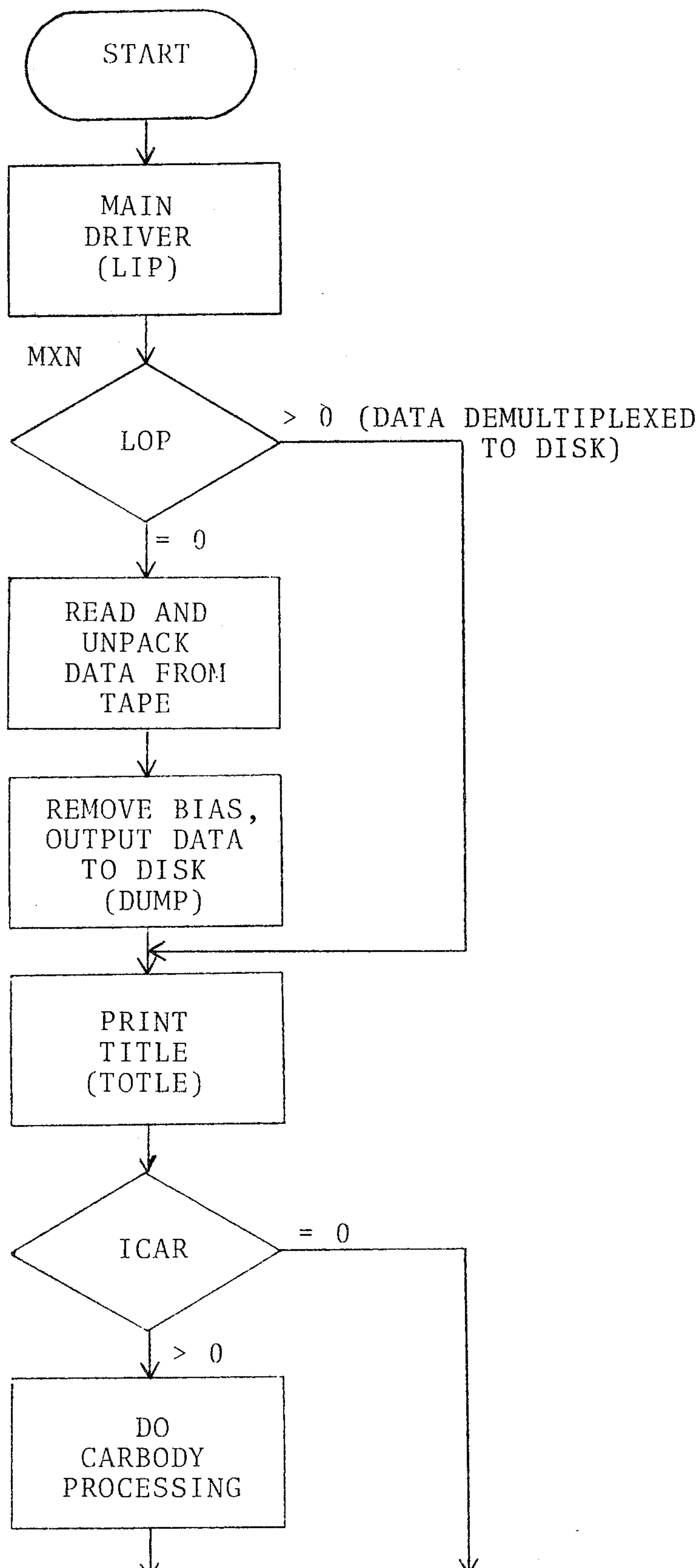
AXL - Performs axle computations and prints results.
SPEC - Computes PSD's of coordinates.
ACF4 - Applies inverse filter (1.6 Hz).

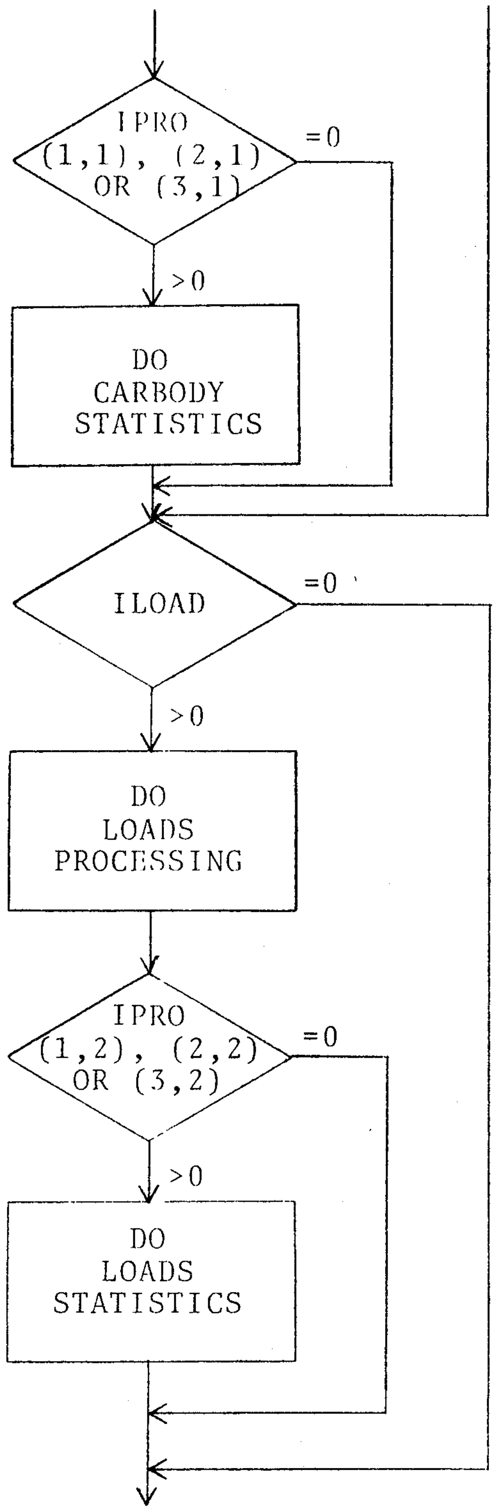
A.1.5 STATISTICS ROUTINES

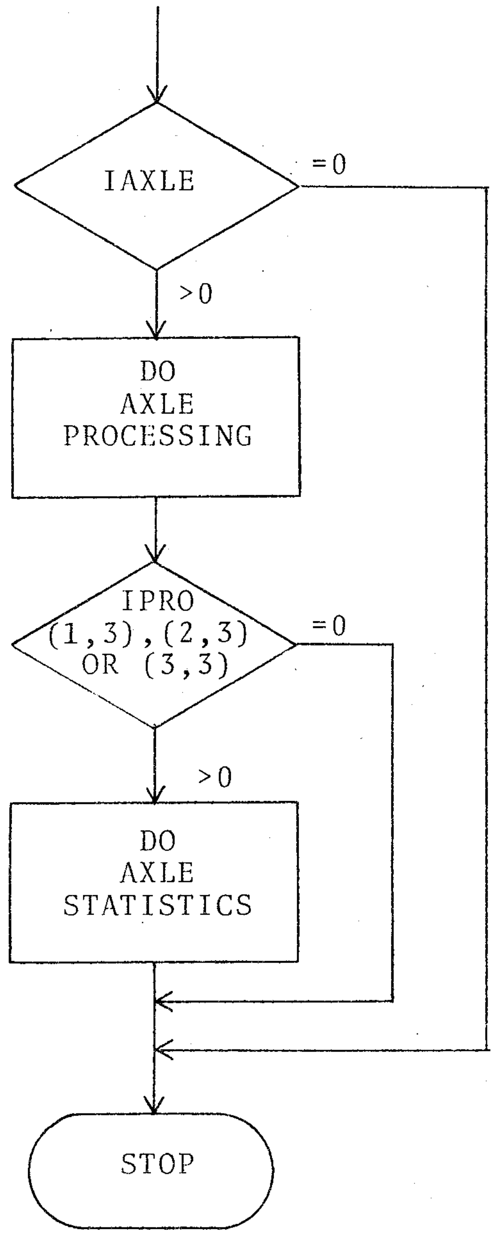
PROG - Controls statistics routines.
ZXS - Performs zero crossings statistics.
ZTITL - Determines zero crossings page header.
HGDMP - Prints zero crossings histogram.
OMS - Creates octave band RMS plot tape.
STAT - Controls general statistics processing.
SFLT - Determines maximum and minimum array values.
STTD - Computes histogram values.
SSCL - Determines histogram bins.
STD1 - Computes statistics.
SPLT - Plots histogram.
SPUM - Prints statistical summary page.

A.2 MAIN DRIVER AND SETUP SOFTWARE

A.2.1 FLOW CHART




IPRO
(1,1), (2,1)
OR (3,1)
=0
>0
DO
CARBODY
STATISTICS
ILOAD
=0
>0
DO
LOADS
PROCESSING
IPRO
(1,2), (2,2)
OR (3,2)
=0
>0
DO
LOADS
STATISTICS


IAXLE
=0
>0
DO
AXLE
PROCESSING
IPRO
(1,3),(2,3)
OR (3,3)
=0
>0
DO
AXLE
STATISTICS
STOP

A.2.2 SYMBOLS FOR MAIN DRIVER AND SETUP ROUTINES

ICAR - Flag to do carbody processing.

ILOAD - Flag to do loads processing.

IAXLE - Flag to do axles processing.

IPRO - Flags to do zero crossing, octave band, RMS and statistics processing.

LOP - Equals zero if data is on tape. Greater than zero if data is demultiplexed to disk.

TITLES - Titles for printout.

IFILE - Beginning file number.

IREC - Beginning record number.

NOREC - Number of input data records.

NREC - Number of output data records.

NCHAN - Number of channels on tape.

MOVE - Channels to use.

ICDHD - Header card for output.

IBUF - Buffer to receive tape data.

KAREA - Buffer where tape data is unpacked.

IWORK - Buffer for unpacked data arranged by channel.

```
:F
      PROGRAM LIP
C
C        MAIN DRIVER PROGRAM FOR LWFC PROCESSING
C
      DOUBLE PRECISION TITLES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      INTEGER RES
      COMMON/AZBLCK/WORK1(514),WORK2(514),VXC(512,8),SCALED(16),
     1IBLK(7168),IVEC(512,16),LOOD(2,4),SPACE(7,8,4),IABC(2),LB,LE,IACF,
     2NTOC,NSTK,MREC,ILOAD,ICHDD(8,2),XH(8,7,4),ICHAN,NCHAN,IGNOR(1101)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYTITL(24),IXTITL(48)
     1,ICDHD(40),IHD(45),IDUM(90)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
   10 FORMAT(13I5)
   20 FORMAT(40A2)
      READ(2,10,END=1000)ICAR,ILOAD,IAXLE,IPRO,LOP
      CALL MXN(LOP)
      READ(2,20) ICDHD
      CALL TOTLE
      IF(ICAR)2,4,2
    2 CALL IORA(7,30,0)
      CALL IORA(9,200,0)
      CALL TOQS
      CALL CAR
      IF(IPRO(1,1)+IPRO(2,1)+IPRO(3,1).NE.0 ) CALL PROG(1)
    4 IF(ILOAD)5,6,5
    5 CONTINUE
      CALL IORA(10,30,0)
      CALL IORA(9,200,0)
      CALL SDS(NTCC,JRES)
   25 CALL IORA(6,198,0)
      CALL IORA(7,199,0)
      CALL IORA(8,201,0)
      DO 30 IK=LB,LE
      CALL SD1(IK,JRES)
      IF(JRES) 30,26,30
   26 CONTINUE
      CALL P2B(NTCC)
   30 CONTINUE
   32 CALL IORA(10,200,0)
      CALL IORA(7,30,0)
      CALL CKM(NTCC)
      IF(IPRO(1,2)+IPRO(2,2)+IPRO(3,2).NE.0)CALL PROG(2)
      NOREC=0
    6 IF(IAXLE)7,8,7
    7 CONTINUE
      CALL IORA(10,30,0)
      CALL IORA(9,200,0)
      CALL AXL
      IF(IPRO(1,3)+IPRO(2,3)+IPRO(3,3).NE.0) CALL PROG(3)
    8 CONTINUE
      STOP
 1000 CONTINUE
      STOP 7
      END
```

```
:F
      SUBROUTINE MXN(LOP)
C
C        M BY N
C        UNPACKS TAPE - INPUT DATA PACKED EVERY 12 BITS
C        CAN READ TAPES UP TO 128 CHANNELS
C
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3,
     1          LSEC0,LSEC1,LSEC2,LSEC3
      DIMENSION IHDR(39)
      COMMON/AZBLCK/IWORK(27672)
      COMMON/BZBLCK/IBUF(4896),MOVE(49),IAX(9),KAREA(3840)
      COMMON/CZBLCK/KCAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      DATA IA,IB,IC,ID,IE,IFF,IS/5*0,1,10/
   10 FORMAT(20I4)
   15 FORMAT(1H1,/,6X,4HFILE,4X,5HREC 1,4X,5HN.REC,4X,5HN.OUT,/,
     14(5X,I5))
    5 FORMAT(5X,5HSTART,2X,4(I7,2X),/,7X,3HEND,2X,4(I7,2X))
20202 FORMAT(1H ,10X,'NREC = ',I5)
   21 FORMAT(5X,5HN.REC,I5,5X,5HN.OUT,I5)
   17 FORMAT(16(2X,I4))
   55 FORMAT(45X,11HTAPE HEADER,/,5X,39A2)
      READ(2,10,END=1000)IFILE,IREC,NOREC,NREC,NCHAN
      PRINT 15,IFILE,IREC,NOREC ,NREC,NCHAN
      IF (NCHAN.LE.0) NCHAN = 124
      ISEC0=1
      CALL DKINT(0,8704,ISEC0,LSEC0,NREC)
      PRINT 20202,NREC
      ISEC1=LSEC0+1
      CALL DKINT(1,8192,ISEC1,LSEC1,NREC)
      PRINT 20202,NREC
      ISEC2=LSEC1+1
      CALL DKINT(2,7680,ISEC2,LSEC2,NREC)
      PRINT 20202,NREC
      ISEC3=LSEC2+1
      CALL DKINT(3,512,ISEC3,LSEC3,NREC)
      PRINT 20202,NREC
      PRINT 5,ISEC0,ISEC1,ISEC2,ISEC3,
     1        LSEC0,LSEC1,LSEC2,LSEC3
      MOREC=FLOAT(NREC)*512./30.
      IF(NOREC.GT.MOREC+1)NOREC=MOREC+1
      MOREC=FLOAT(NOREC)*30./512.
      PRINT 21,NOREC,MOREC
      IF(LOP.NE.0) RETURN
      CALL POPEN(IA,IC,ID,IE,IFF,IS,IB)
      READ(2,10,END=1000)MOVE
      PRINT 17,MOVE
      CALL MTOPEN(8,19)
C     INPUT FILE CONTROL
      IF(IFILE.LT.2) GO TO 30
      JF=IFILE-1
      DO 20 I=1,JF
      CALL MTOPEN(8,21)
   20 CONTINUE
   30 CALL MTOPEN(8,9)
      CALL MTIO(8,IHDR,39)
      IF(IEOF(8).NE.0)GO TO 1100
```

```
C      INPUT RECORD CONTROL
       IF(IREC.LT.2)GO TO 50
       JR=IREC-1
       DO 40 I=1,JR
       CALL MTIO(8,IBUF,2973)
       IF(IEOF(8).NE.0)GO TO 1100
   40  CONTINUE
   50  CONTINUE
       PRINT 55,IHDR
       MREC=0
       LOC=1
  100  CONTINUE
       CALL MTIO(8,IBUF,2973)
       IF(IEOF(8).NE.0)GO TO 900
       MREC=MREC+1
       INDINC = (NCHAN*3)/4 + 3
       IF (NCHAN.NE.4*(NCHAN/4)) INDINC = INDINC + 1
       INDX = -INDINC + 7
       INTO = -NCHAN + 1
       DO 200 K=1,30
       INDX = INDX + INDINC
       INTO = INTO + NCHAN
       CALL PIOR(IBUF(INDX),0,12,0,0,0,KAREA(INTO),0,16,12,NCHAN)
  200  CONTINUE
       I = 30*NCHAN
       CALL PSRA(KAREA,1,KAREA,1,4,I,0)
       NDO=30
       MV=0
       IF(LOC+29.GT.512)NDO=513-LOC
  210  CONTINUE
       INK=1
       DO 300 I=1,49
       INDX=MOVE(I)+MV
       IROW=LOC+(I-1)*512
       CALL PREL(KAREA(INDX),NCHAN,IWORK(IROW),INK,NDO,0)
  300  CONTINUE
       IF(LOC+29.LT.512) GO TO 310
       CALL DUMP(LOC,NDO)
       MV=(30-NDO)*NCHAN
       IF(NDO.NE.30) GO TO 210
  310  CONTINUE
       LOC=LOC+NDO
       IF(MREC.LT.NOREC)GO TO 100
       CALL PCLOS
       RETURN
  900  CONTINUE
       CALL PCLOS
       STOP
 1000  CONTINUE
       CALL PCLOS
       STOP 1
 1100  CONTINUE
       CALL PCLOS
       STOP 2
       END
```

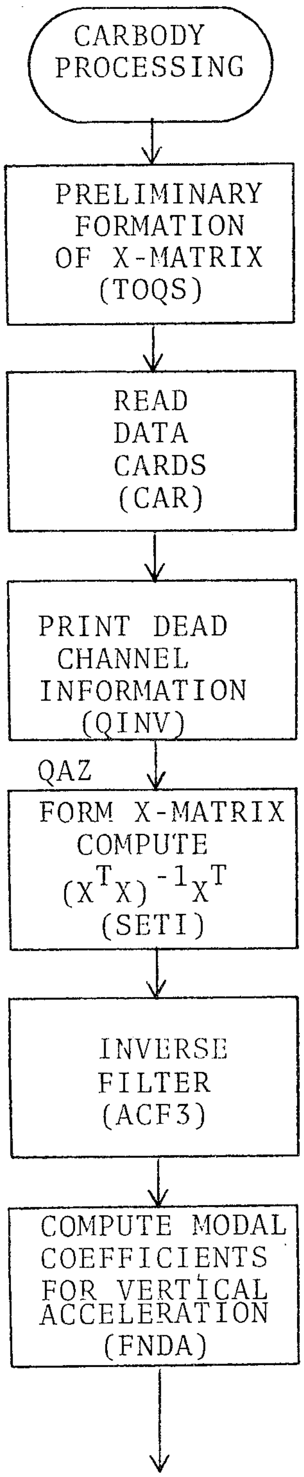
:F

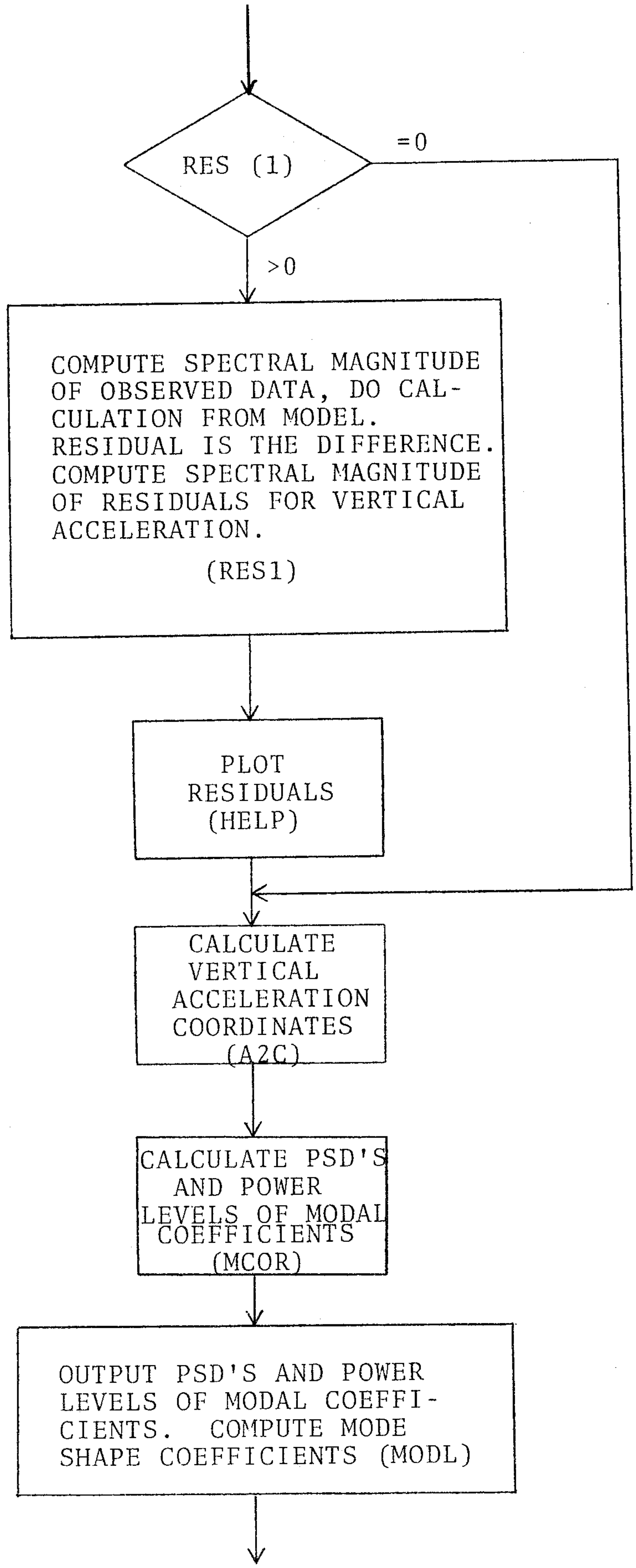
```
      SUBROUTINE DUMP(LOC,NDO)
      DIMENSION Z(2)
      DIMENSION POJNT(512)
      COMMON/AZBLCK/IWORK(27672)
      COMMON/BZBLCK/IBUF(8481),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      EQUIVALENCE (IBUF,POJNT)
      DATA JREC0/0/
      X=1./512.
      DO 400 I=1,48
      J=(I-1)*512+1
      CALL PCNFL(IWORK(J),1,POJNT,2,15,512,0)
      CALL PCONR(POJNT,2,X,0,Z,2,512,2,15)
      CALL PSUB(Z,0,POJNT,2,POJNT,2,512,15)
      CALL PCNFX(POJNT,2,IWORK(J),1,15,512,0)
  400 CONTINUE
      JREC0=JREC0+1
      CALL DKWR(1,IWORK(8705),8192,JREC0,0)
      CALL DKWT(ISTAT)
      CALL DKWR(2,IWORK(16897),7680,JREC0,0)
      CALL DKWT(ISTAT)
      CALL DKWR(3,IWORK(24577),512,JREC0,0)
      CALL DKWT(ISTAT)
      CALL PREL(IWORK(1),1,IWORK(8800),1,8704,0)
      DO 410 I=1,512
      J=(I-1)*17+1
      II=I+8799
      CALL PREL(IWORK(II),512,IWORK(J),1,17,0)
  410 CONTINUE
      CALL DKWR(0,IWORK(1),8704,JREC0,0)
      CALL DKWT(ISTAT)
      LOC=1
      IF(NDO.EQ.30)LOC=-29
      IF(NDO.NE.30)NDO=30-NDO
      CALL PCLR(IWORK(1),1,25600,0)
      RETURN
      END
```

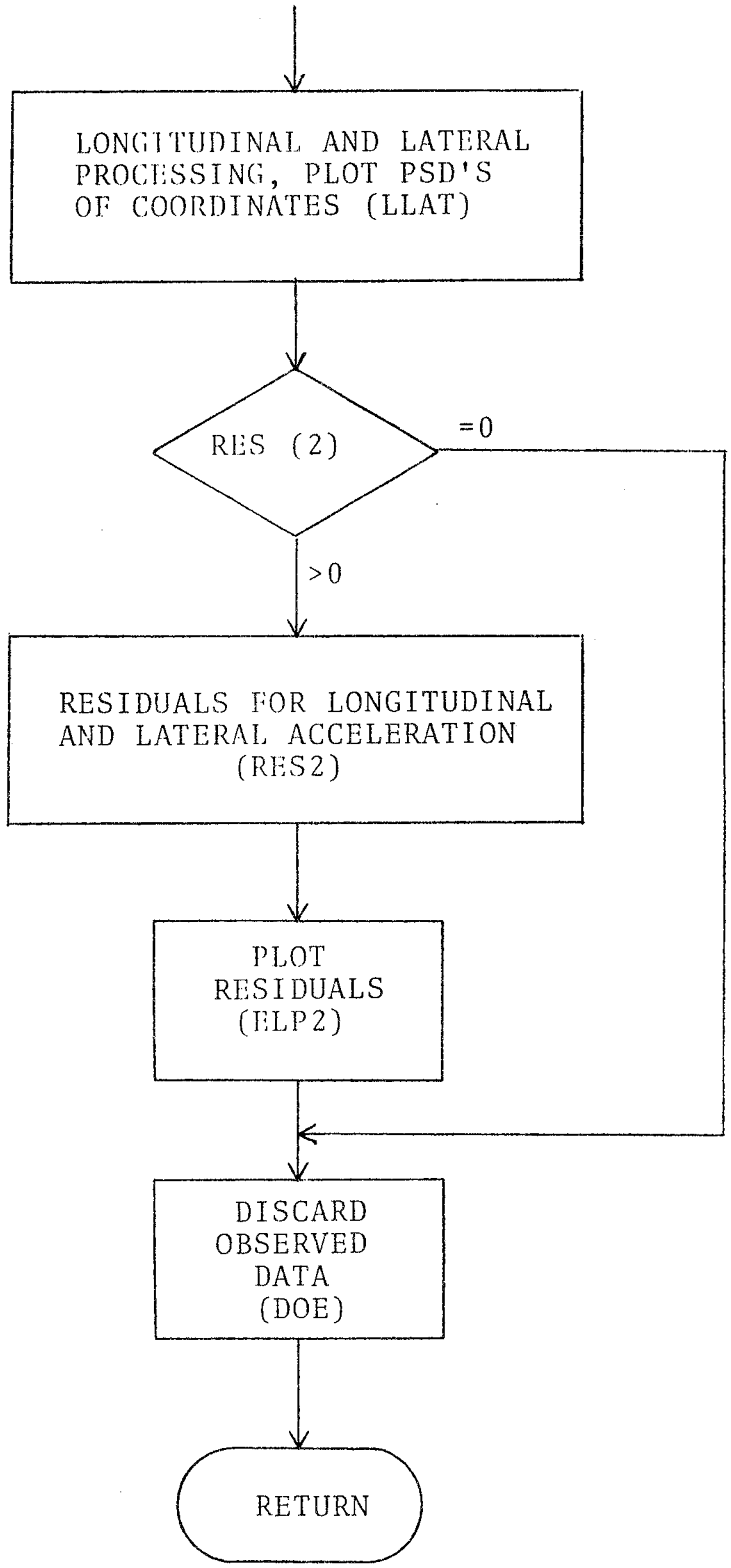
```
:F
      SUBROUTINE TOTLE
      DIMENSION IZTIT1(66),IYTIT1(24),IXTIT1(48)
      DIMENSION ISTR1(39),ISTR2(39),ISTR3(39)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYTITL(24),IXTITL(48)
     1,ICDHD(40),IHD(45)
      DATA ISTR1/1,7,13,19,25,31,37,43,49,55,1,7,13,19,25,31,61,1,7,13,
     119,25,31,61,1,7,13,19,31,1,7,13,19,31,1,7,13,19,31/
      DATA ISTR2/10*1,7*5,7*9,5*13,5*17,5*21/
      DATA ISTR3/3*1,3*13,2*25,2*37,3*1,3*13,25,3*1,3*13,25,3*1,2*13,
     13*1,2*13,3*1,2*13/
      DATA IZTIT1/2HLO,2HNG,2HIT,2HUD,2HIN,2HAL,4*2H  ,2HSW,2HAY,3*2H  ,
     12HBO,2HUN,2HCE,4*2H  ,2HRO,2HLL,3*2H  ,2H P,2HIT,2HCH,4*2H  ,2H Y,
     22HAW,2H 1,2HST,2H B,2HEN,2HDI,2HNG,2H 2,2HND,2H B,2HEN,2HDI,2HNG,
     32H 1,2HST,2H T,2HOR,2HSI,2HON,2H 2,2HND,2H T,2HOR,2HSI,2HON,2H L,
     42HAT,2H B,2HEN,2HDI,2HNG/
      DATA IYTIT1/2HCA,2HR ,2HBO,2HDY,
     12HA ,2H- ,2HLO,2HAD,
     12HB ,2H- ,2HLO,2HAD,
     12HAA,2H- ,2HAX,2HLE,
     12HA ,2H- ,2HAX,2HLE,
     12HBB,2H- ,2HAX,2HLE/
      DATA IXTIT1/5*2H  ,2H  ,2HG ,5*2H  ,3*2H  ,2HRA,2HD/,2HSE,2HC/,
     12HSE,2HC ,3*2H  ,3*2H  ,2H G,2H /,2HUN,2HIT,2H D,2HIS,2HP ,2*2H  ,
     42H  ,2H G,2H /,2HUN,2HIT,2H D,2HIS,2HP/,2HUN,2HIT,2H L,2HEN/
      DO 10 I=1,66
      IZTITL(I)=IZTIT1(I)
   10 CONTINUE
      DO 20 I=1,24
      IYTITL(I)=IYTIT1(I)
   20 CONTINUE
      DO 30 I=1,48
      IXTITL(I)=IXTIT1(I)
   30 CONTINUE
      PRINT 100
      DO 40 I=1,39
      IST1=ISTR2(I)
      IEN1=IST1+3
      IST2=ISTR1(I)
      IEN2=IST2+5
      IST3=ISTR3(I)
      IEN3=IST3+11
      PRINT 50,I,(IYTITL(J),J=IST1,IEN1),(IZTITL(J),J=IST2,IEN2),
     1(IXTITL(J),J=IST3,IEN3)
      IF(I.EQ.10.OR.I.EQ.24) PRINT 10101
      IF(I.EQ.17.OR.I.EQ.29.OR.I.EQ.34) PRINT 20202
   40 CONTINUE
      PRINT 60
  100 FORMAT(1H1,61X,'CHANNEL DESCRIPTION',// 20X,'NUMBER',11X,'TYPE',
     110X,'MODE',15X,'ENGINEERING UNITS',/)
   50 FORMAT(22X,I2,11X,4A2,4X,6A2,5X,12A2)
   60 FORMAT(1H1)
10101 FORMAT(1H0,/)
20202 FORMAT(1H0)
      RETURN
      END
```

A.3 CARBODY SOFTWARE

A.3.1 FLOW CHART




RES (1)
=0
>0
COMPUTE SPECTRAL MAGNITUDE OF OBSERVED DATA, DO CALCULATION FROM MODEL. RESIDUAL IS THE DIFFERENCE. COMPUTE SPECTRAL MAGNITUDE OF RESIDUALS FOR VERTICAL ACCELERATION.
(RES1)
PLOT RESIDUALS (HELP)
CALCULATE VERTICAL ACCELERATION COORDINATES (A2C)
CALCULATE PSD'S AND POWER LEVELS OF MODAL COEFFICIENTS (MCOR)
OUTPUT PSD'S AND POWER LEVELS OF MODAL COEFFICIENTS. COMPUTE MODE SHAPE COEFFICIENTS (MODL)


LONGITUDINAL AND LATERAL PROCESSING, PLOT PSD'S OF COORDINATES (LLAT)
RES (2)
=0
>0
RESIDUALS FOR LONGITUDINAL AND LATERAL ACCELERATION (RES2)
PLOT RESIDUALS (ELP2)
DISCARD OBSERVED DATA (DOE)
RETURN

A.3.2 SYMBOLS FOR CARBODY PROGRAMS

EL2 - Mass distribution integral.

PM(1-2) - Mass distribution integrals.

F2T - Mass distribution integral.

RES(1-2) - Residual flags for vertical accelerometers and lateral and longitudinal accelerometers.

MCV, MCL are flags for plotting PSD's of modal coordinates.

ZSCALE are scale factors for the raw data input channels.

In working with the equations for vertical acceleration:

$$\ddot{z}_m(x,y,t) = A_o(t) + A_1(t)x^2 + A_2(t)x^4 + B_o(t)x + B_1(t)x^3 + D_o(t)y + D_1(t)x^2y + C_o(t)xy + C_1(t)x^3y$$

FX contains the X-matrix (dimension 12 measurements by 9 coordinates.

YZOBS is the observed data (in QAZ).

A(*,1) - A(*,9) are the modal coefficients A_o, A_1, A_2, B_o, B_1, D_o, D_1, C_o, C_1 (in QAZ).

X2X is X^TX (in SET1).

QIXT is $(X^TX)^{-1}X$ (in SET1).

ZOBS is the observed data (in RES1).

ZSTAK are the stacked spectral magnitudes of the observed data (in RES1, HELP).

AA(*,1) - AA(*,9) are the modal coefficients (in RES1).

WORK1 is AX (later) are the spectral magnitudes of the residuals (in RES1).

WORK2 are the residuals (in RES1).

RSTAK are the stacked spectral magnitudes of the residuals (in RES1, HELP).

A9(*,1) - A9(*,9) are the modal coefficients (in A2C).

AOUT(*,1) - AOUT(*,9) are the modal coefficients (in A2C).

AOUT(*,10) = $\ddot{z}_o(t)$ bounce

AOUT(*,11) = α_1 first bending

AOUT(*,12) = $\ddot{\phi}_o(t)$ pitch

AOUT(*,13) = α_2 second bending

AOUT(*,14) = $\ddot{\theta}_o(t)$ roll

AOUT(*,15) = β_2 second torsion

AOUT(*,8) = β_1 first torsion

PM(1) = M_1/M_o

PM(2) = M_2/M_o

F2B = $-M_2/M_1$

F2T = H_1/H_o

AIN (in MCOR) = AOUT (in A2C).

ACPOW(*,1) - ACPOW(*,6) are the PSD's of AOUT(*,10) - AOUT(*,15).

CORFET(*,1) - CORFET(*,4) are the spectral magnitudes of α_1, α_2, β_2, β_1.

XPOW(*,1) - XPOW(*,5) are the cross-spectral densities for A_1, α_1 A_2, α_1 B_1, α_2 D_1, β_2 C_1, β_1.

The mode shape function for the second bending mode is $f_2(x) = x + b_1x^3$ (in MODL).

The torsion mode shape functions are $q_1(x) = x + c_1x^3$ and $q_2(x) = 1 + d_1x^2$.

C1 = VEC(1) = c_1

D1 = VEC(2) = d_1

FFX is the X-matrix (in LLAT).

YOBS contains the observed data in Channels 13 through 17.

AA9 are the modal coefficients: AA9(*,8) = $\ddot{\theta}_o$

AA9(*,7) now contains β_1

AA9(*,9) = β_2

Solving $B = (X^TX)^{-1}X^T$,

EF(*,1) = $\ddot{y}_o$ sway

EF(*,2) = $\ddot{\psi}_o$ yaw

EF(*,3) = $\ddot{x}_o$ longitudinal acceleration

```
:F
      SUBROUTINE CAR
      INTEGER RES
      DOUBLE PRECISION TITLES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      COMMON/AZBLCK/ILARG(27672)
      COMMON/BZBLCK/ISMAL(8464),IDEDCH(17),IZL(66),IYL(24),IXL(48),
     1ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     . IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
      CALL IORA(7,30,0)
C
C   READ PSD(RAW) FLAGS AND PLOT DELETE FLAGS
C
      READ(2,15) EL2,(PM(I),I=1,2),F2T
      READ(2,11) (RES(I),I=1,2)
      READ(2,11) (MCV(I),I=1,7),(MCL(I),I=1,3)
   11 FORMAT(10I2)
      READ(2,15)(ZSCALE(I),I=1,17)
   15 FORMAT(8F10.0)
      CALL IORA(9,200,0)
      CALL QINV
      CALL QAZ
      IF(RES(1)) 1,3,1
    1 CONTINUE
      CALL RES1
    3 CONTINUE
      CALL IORA(10,30,0)
      CALL IORA(8,200,0)
      CALL IORA(9,201,0)
      CALL A2C
      MLT=MCL(1)+MCL(2)+MCL(3)
      CALL IORA(8,201,0)
      CALL IORA(9,200,0)
      CALL MCOR(MLT)
      IF(MLT) 14,20,14
   14 CONTINUE
      CALL IORA(9,202,0)
      CALL IORA(8,200,0)
      CALL LLAT
      CALL DOE
   20 RETURN
      END
```

:F

```
      SUBROUTINE TOQS
      DIMENSION X1(51),Y1(51),Z1(51)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      DATA X1/.9512072,.7512575,.5485412,.02162978,-.5510563,-.9499497, TLDX62
     1       .95120724,.5492958,.019366197,-.5508048,-.8170443,-.9491952,
     2 .934,.754,.036,-.791,.739,
     3    .9731481,.70902778,.3666667,-.01574074,-.3671296,-.9722222,    TTAX
     4   .97314815,.3652778,.01388889,-.3675926,-.7115741,-.9680556,
     5 .988,.751,.006,-.716,.692,
     6   .9788732,.7545272,.5623742,.006790745,-.5601103,-.9808853,      TLDX61
     7   .9781187,.5631288,.007042254,-.5626258,-.9089537,-.9783702,
     8 .977,.734,-.0035,-.803,.716/
      DATA Y1/-.08299799,.02012072,-.07218310,-.06916499,-.07218309,     TLDX62
     2-.08350101,.08249497,.07168008,.07017103,.07168008,.02112676,
     3.08299799,.042,.032,.061,.033,.020,
     4  -.09027778,.01412037,-.09027778,-.09027778,-.09027778,-.09027778 TTAX
     5,.09027778,.09027778,.09027778,.09027778,.01412037,.09027778,
     6  .038,.017,-.027,-.03,.014,
     7  -.1023642,-.02112676,-.07218310,-.06991952,-.07218310,-.1036217, TLDX61
     8  .1031187,.07268612,.06966800,.07193159,-.02112676,.07268612,
     9  .034,.032,.062,.032,.021/
      DATA Z1/12*0.,.0045,-.00416,.0156,-.0044,-.0117,                   TLDX62
     2  12*0.,-.0022,.0036,-.0028,.0027,-.0013,                          TTAX
     3  12*0.,.0047,.0032,.0057,.005,-.0186/                             TLDX61
      DO 5 I=1,17
      IP=I+17
      IPP=IP+17
      X(I,1)=X1(I)
      X(I,2)=X1(IP)
      X(I,3)=X1(IPP)
      Y(I,1)=Y1(I)
      Y(I,2)=Y1(IP)
      Y(I,3)=Y1(IPP)
      Z(I,1)=Z1(I)
      Z(I,2)=Z1(IP)
      Z(I,3)=Z1(IPP)
    5 CONTINUE
      RETURN
      END
```

:F

```
      SUBROUTINE QINV
      COMMON/BZBLCK/ISMAL(8464),IDEDCH(17),IZL(66),IYL(24),IXL(48),
     1ICDHD(40),IHD(45)
   15 FORMAT(1H0,10X,'DEAD CHANNELS',/)
   45 FORMAT(1H0,15X,'HORIZONTAL CHANNEL BAD, WILL NOT PROCESS LONG-LAT
     1FOR CARBODY')
   20 FORMAT(15X,'CH  ',I2)
    5 FORMAT(17I4)
      READ(2,5) IDEDCH
      ICNT=0
      PRINT 15
      DO 10 I=1,17
      IF(IDEDCH(I).EQ.0) GO TO 10
      PRINT 20,I
      ICNT=ICNT+1
   10 CONTINUE
      IF(IDEDCH(17).NE.0) GO TO 40
      IF(ICNT.EQ.0) PRINT 25
   25 FORMAT(15X,'NONE')
      PRINT 30
   30 FORMAT(1H0,//)
   27 CONTINUE
      RETURN
   40 PRINT 45
      GO TO 27
      END
```

```
:F
      SUBROUTINE QAZ
      INTEGER DUM(17)
      INTEGER RES
      DOUBLE INTEGER ISEC0,LSEC0,ISEC1,ISEC2,ISEC3
      DOUBLE PRECISION DNCAR(3)
      COMMON/AZBLCK/A(512,10),YZOBS(17,512),INEG(24)
      COMMON/BZBLCK/FX(12,10),QIXT(12,10),S,IGEN(7982),IDEDCH(17),
     1IZL(66),
     1IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      EQUIVALENCE(A(1,1),DUM(1) )
      DATA ITAPE2/9/,IPRINT/3/,ICARDS/2/,NQI/9/,KDEL/1/,IOW/9216/
      DATA IOFX/216/
      DATA IEND/Z0010/
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,7,0/
      DATA DNCAR/6HTLDX62,6HTTAX  ,6HTLDX61/
      N=0
      PRINT 5000
      WRITE(3,6002)DNCAR(ICAR),(ZSCALE(I),I=1,17)
      CALL SET1(KDEL)
      MREC=NREC
      CALL DKINT(0,8704,ISEC0,LSEC0,MREC)
      PRINT 41,LSEC0,MREC
   41 FORMAT(5X,7HLSEC0 =,2X,I10,10H     MREC =,I5,//)
      CALL MTOPEN(ITAPE2,19)
      CALL MTOPEN(ITAPE2,3)
C**** READ NEW HEADER FROM CARDS
C**** WRITE NEW HEADER ON OUTPUT TAPE
      CALL MTIO(ITAPE2,ICDHD,85)
      CALL MTWAIT(ITAPE2,ISTAT,IWDS)
C**** PRINT NEW HEADER
      WRITE(IPRINT,5003) ICDHD
C
C**** WRITE FX ARRAY ON OUTPUT TAPE
C
      CALL MTIO(ITAPE2,FX,IOFX)
      CALL MTWAIT(ITAPE2,ISTAT,NWDS)
C
C**** OPEN THE APOLLO ARRAY PROCESSOR
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
   10 CONTINUE
      DO 33 IK=1,NREC
C     READ A BLOCK OF Z-DATA FROM DISK
      CALL DKRD(0,A,8704,IK,0)
      CALL DKWT(ISTAT)
      IF(ISTAT)9002,20,9002
   20 CONTINUE
C**** FLOAT THE Z DATA FROM THE A BUFFER TO THE Z AREA
      DO 25 I=1,17
   25 CALL PCNFL(DUM(I),17,YZOBS(I,1),34,15,512,0)
      IF(N.GT.0) GO TO 30
      CALL ACF3(17,-512)
      GO TO 31
   30 CONTINUE
      CALL ACF3(17,512)
   31 CONTINUE
```

```
      DO 32 I=1,17
      IF(I.GT.12) GO TO 44
      IF(IDEDCH(I).EQ.0) GO TO 44
      CALL PMPY(YZOBS(I,1),34,0.0,0,YZOBS(I,1),34,512,15)
   44 CONTINUE
      CALL PCNFX(YZOBS(I,1),34,DUM(I),17,15,512,0)
   32 CONTINUE
C**** WRITE THE Z ARRAY (INPUT)
      CALL MTIO(ITAPE2,A,8704)
      CALL MTWAIT(ITAPE2,ISTAT,IWDS)
      CALL FNDA(NQI)
C**** WRITE THE A ARRAY (RESULT)
      CALL MTIO(ITAPE2,A,IOW)
      CALL MTWAIT(ITAPE2,ISTAT,IWDS)
      N=N+1
   33 CONTINUE
C
 9000 CONTINUE
      CALL MTOPEN(ITAPE2,17)
      CALL MTOPEN(ITAPE2,19)
      WRITE(IPRINT,6001) N
      GO TO 43
      CALL PCLOS
 9002 PRINT 5002
      STOP 77
   43 CONTINUE
      CALL PCLOS
      RETURN
 5000 FORMAT(I1)
 5002 FORMAT(1X,10HISTAT.NE.0)
 5003 FORMAT(10X,23HOUTPUT TAPE DESCRIPTION/10X,40A2/)
 5005 FORMAT(8F10.0)
 6001 FORMAT(20X,I5,2X,19HSOLUTIONS PERFORMED)
 6002 FORMAT(1X,14HCAR SELECTED   ,A6,3X,21HACCELEROMETER SCALING/
     . 3(5X,8F15.5))
      END
```

```
:F
      SUBROUTINE SET1(KDEL)
      INTEGER RES
      DOUBLE PRECISION X2X,D
      DOUBLE INTEGER ISEC0,LSEC0,ISEC1,ISEC2,ISEC3
      DIMENSION FX1(12,9)
      DIMENSION L(10),M(10)
      COMMON/BZBLCK/FX(12,10),QIXT(12,10),S,X2X1(9,9),X2X(9,9),
     1IGEN(7577),IDEDCH(17),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      EQUIVALENCE (FX,FX1)
  505 FORMAT(1H0,//)
 4004 FORMAT(1X,9D14.6)
      DO 5 M=1,12
      XI=X(M,ICAR)
      YI=Y(M,ICAR)
      FX(M,1)=1.0
      FX(M,2)=XI*XI
      FX(M,3)=FX(M,2)*FX(M,2)
      FX(M,4)=FX(M,2)*FX(M,3)
      FX(M,5)=XI
      FX(M,6)=FX(M,2)*XI
      FX(M,7)=YI
      FX(M,8)=FX(M,2)*YI
      FX(M,9)=XI*YI
      FX(M,10)=FX(M,6)*YI
    5 CONTINUE
      DO 10 M=1,12
      IF(IDEDCH(M).EQ.0) GO TO 10
      DO 8 MM=1,10
      FX(M,MM)=0.0
    8 CONTINUE
   10 CONTINUE
      NQI=9
      DO 12 M=1,12
      DO 12 I=1,6
      IP=I+3
      IPP=IP+1
      FX1(M,IP)=FX(M,IPP)
   12 CONTINUE
      DO 13 I=1,12
      PRINT 4004,(FX1(I,J),J=1,9)
   13 CONTINUE
      PRINT 505
      JKL=ICAR
      DO 909 I=1,9
      DO 909 K=1,9
      X2X(I,K)=0.0D0
      DO 909 J=1,12
      X2X(I,K)=X2X(I,K)+FX(J,I)*FX(J,K)
  909 CONTINUE
      DO 15 I=1,9
      PRINT 4004,(X2X(I,J),J=1,9)
   15 CONTINUE
      PRINT 505
      CALL MINV(X2X,9,D,L,M)
      DO 808 I=1,9
```

```
      DO 808 J=1,9
      X2X1(I,J)=X2X(I,J)
  808 CONTINUE
      DO 14 I=1,9
      PRINT 4004,(X2X1(I,J),J=1,9)
   14 CONTINUE
      DO 11 I=1,9
      DO 11 K=1,12
      QIXT(K,I)=0.0
      DO 11 J=1,9
      QIXT(K,I)=QIXT(K,I)+X2X1(I,J)*FX(K,J)
   11 CONTINUE
      PRINT 505
      DO 10101 I=1,12
      PRINT 4004,(QIXT(I,K),K=1,9)
10101 CONTINUE
      RETURN
      END
```

```
:F
      SUBROUTINE MINV(A,N,D,L,M)
C     A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C         RESULTANT INVERSE
C     N - ORDER OF MATRIX A
C     D - RESULTANT DETERMINANT
C     L - WORK VECTOR OF LENGTH N
C     M - WORK VECTOR OF LENGTH N
C
      DOUBLE PRECISION A,D,BIGA,HOLD                                   C
      DIMENSION A(1),L(1),M(1)
C          SEARCH FOR LARGEST ELEMENT
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA).GE.DABS(A(IJ))) GO TO 20                           C
C                                    02 OT OG )))JI(A(SBA.EG.)AGIB(SBA(FI
      BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C         INTERCHANGE ROWS
      J=L(K)
      IF (J-K)35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
C         INTERCHANGE COLUMNS
   35 I=M(K)
      IF (I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI)=HOLD
C         DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C         CONTAINED IN BIGA)
C                                      84 OT OG )02-E0.1.TG.)AGIB(SBA(FI 54
   45 IF(DABS(BIGA).GE.1.0D-20) GO TO 48                               C
      D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K)50,55,50
   50 IK=NK+I
```

```
      A(IK)=A(IK)/(-BIGA)
 55   CONTINUE
C         REDUCE MATRIX
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K)60,65,60
 60   IF(J-K)62,65,62
 62   KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
 65   CONTINUE
C        DIVIDE ROW BY PIVOT
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K)70,75,70
 70   A(KJ)=A(KJ)/BIGA
 75   CONTINUE
C        PRODUCTS OF PIVOTS
      D=D*BIGA
C        REPLACE PIVOT BY RECIPROCAL
      A(KK)=1.0/BIGA
 80   CONTINUE
C        FINAL ROW AND COLUMN AND INTERCHANGE
      K=N
 100  K=(K-1)
      IF (K) 150,150,105
 105  I=L(K)
      IF(I-K) 120,120,108
 108  JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
 110  A(JI)=HOLD
 120  J=M(K)
      IF(J-K) 100,100,125
 125  KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
 130  A(JI)=HOLD
      GO TO 100
 150  RETURN
      END
```

:F

```
      SUBROUTINE ACF3(M,N)
      DOUBLE PRECISION A1,A2,XCHN,X1,X2
      DIMENSION XM1(20)
      COMMON/AZBLCK/A(512,10),YZOBS(17,512),INEG(24)
      DATA A1,A2,XM1/13.3189768,0.92446525,20*0./
      IN=IABS(N)
      IM=M
      IF(N.GT.0) GO TO 20
      DO 10 I=1,IM
      XM1(I)=YZOBS(I,1)+YZOBS(I,1)-YZOBS(I,2)
   10 CONTINUE
   20 CONTINUE
      DO 40 I=1,IM
      XCHN=XM1(I)
      DO 30 J=1,IN
      X1=YZOBS(I,J)
      X2=A1*(X1-A2*XCHN)
      YZOBS(I,J)=SNGL(X2)
      XCHN=X1
   30 CONTINUE
      XM1(I)=XCHN
   40 CONTINUE
      RETURN
      END
```

:F

```
      SUBROUTINE FNDA(NQI)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DIMENSION TQI(12,10)
      COMMON/AZBLCK/A(512,10),YZOBS(17,512),IHP(24)
      COMMON/BZBLCK/FX(12,10),QIXT(12,10),S,IGE(7982),IDEDCH(17),
     1IZL(66),
     1IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      CALL PCLR(A,2,5120,15)
      DO 70 I=1,NQI
      DO 70 J=1,12
      TQI(J,I)=ZSCALE(J)*QIXT(J,I)
   70 CONTINUE
      DO 10 I=1,NQI
      DO 10 K=1,512
      DO 10 J=1,12
      A(K,I)=A(K,I)+TQI(J,I)*YZOBS(J,K)
   10 CONTINUE
   51 RETURN
      END
```

```
:F
      SUBROUTINE RES1
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      INTEGER DUM
      DIMENSION DUM(12)
      COMMON/AZBLCK/AA(512,9),ZOBS(12,512),RSTAK(12,257)
      COMMON/BZBLCK/FX(12,10),WORK1(514),WORK2(514),ZSTAK(12,257),
     1IPLIS(17),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      EQUIVALENCE(AA,DUM)
      DATA IEND/Z0010/
      DATA N/0/
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,7,0/
      SCALE=.005524272
      CALL MTOPEN(9,19)
      NQI=9
      KOT=9216
      KOT1=216
      CALL MTOPEN(9,9)
      CALL MTIO(9,ICDHD,85)
      CALL MTWAIT(9,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 1,1,9000
    1 CONTINUE
      CALL MTIO(9,FX,KOT1)
      CALL MTWAIT(9,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 9,9,9000
    9 CONTINUE
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
      CALL PCLR(RSTAK,2,3084,15)
      CALL PCLR(ZSTAK,2,3084,15)
    5 CONTINUE
C**** READ ZOBS ARRAY
      CALL MTIO(9,DUM,8704)
      CALL MTWAIT(9,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 11,11,1001
   11 CONTINUE
      DO 211 I=1,12
      CALL PCNFL(DUM(I),17,ZOBS(I,1),24,15,512,0)
      CALL PMPY(ZOBS(I,1),24,ZSCALE(I),0,ZOBS(I,1),24,512,15)
  211 CONTINUE
      CALL MTIO(9,AA,KOT)
      CALL MTWAIT(9,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 10,10,1000
   10 CONTINUE
      DO 100 I=1,12
C**** COMPUTE SPECTRAL MAGNITUDE OF ZOBS
      CALL PREL(ZOBS(I,1),24,WORK2,2,512,1)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)*2.0
      WORK1(2)=0.0
      WORK1(514)=0.0
      WORK1(1)=WORK1(1)*2.
      CALL PMPY(WORK1,2,SCALE,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.0
      CALL PADD(ZSTAK(I,1),24,WORK2,2,ZSTAK(I,1),24,257,15)
  100 CONTINUE
```

```
      DO 111 I=1,12
C**** COMPUTE ZCALC
      CALL PREL(AA(1,1),2,WORK1,2,512,1)
      DO 50 J=2,NQI
      CALL PMPY(AA(1,J),2,FX(I,J),0,WORK2,2,512,15)
      CALL PADD(WORK1,2,WORK2,2,WORK1,2,512,15)
   50 CONTINUE
C**** COMPUTE   RESID=ZOBS-ZCALC
      CALL PSUB(WORK1,2,ZOBS(I,1),24,WORK2,2,512,15)
C**** COMPUTE SPECTRAL MAGNITUDE OF RESIDUALS
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)*2.0
      WORK1(2)=0.0
      WORK1(514)=0.0
      WORK1(1)=WORK1(1)*2.
      CALL PMPY(WORK1,2,SCALE,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.0
C**** STACK SPECTRAL MAGNITUDE ARRAYS
      CALL PADD(RSTAK(I,1),24,WORK2,2,RSTAK(I,1),24,257,15)
C**** INCREMENT STACKING COUNTER
  111 CONTINUE
      N=N+1
      GO TO 5
 1001 CONTINUE
 1000 CONTINUE
      CALL MTOPEN(9,19)
      CALL HELP(N)
      GO TO 6
 9000 CONTINUE
      PRINT 3001
    6 CONTINUE
      CALL PCLOS
      RETURN
 2000 FORMAT(2I5)
 2001 FORMAT(8F10.0)
 3000 FORMAT(10X,30HVERTICLE ACCELEROMETER SCALING,E15.6/)
 3001 FORMAT(10X,55HEND OF FILE ENCOUNTERED WHILE ATTEMPTING TO READ HEA
     *DER)
 3002 FORMAT(10X,    22HINPUT TAPE DESCRIPTION/10X,40A2/10X,45A2/)
 3003 FORMAT(1H1,9X,21HACCELEROMETER SCALING,10E10.4/30X,7F10.4)
      END
```

```
:F
      SUBROUTINE A2C
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DIMENSION A9(512,9)
      COMMON/AZBLCK/AOUT(512,15),WK1(512),WK2(512),IHD(85),LDUM(216),IGU
     . (9963)
      COMMON/BZBLCK/IABC(8464),IDEDCH(17),IZL(66),IYL(24),IXL(48),
     1ICDHD(40),JHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      EQUIVALENCE(AOUT(1,1),A9(1,1))
      EQUIVALENCE(A9,KDUM)
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,10,0/
      DATA N,IT1,IT2,IEND/0,8,9,Z0010/
      CALL MTOPEN(IT1,19)
      CALL MTOPEN(IT2,19)
      CALL MTOPEN(IT1,9)
      CALL MTOPEN(IT2,3)
      CALL MTIO(IT1,ICDHD,85)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 5,5,9000
    5 CONTINUE
      CALL MTIO(IT2,ICDHD,85)
      CALL MTWAIT(IT2,ISTAT,NWDS)
      F2B=-PM(2)/PM(1)
      PRINT 3013,PM,F2T,F2B,EL2
      ELG=32.174/EL2
C
C  READ PAST FX ON INPUT TAPE
C
      CALL MTIO(IT1,LDUM,216)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 6,6,9000
    6 CONTINUE
C
C  OPEN APOLLO
C
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
    1 CONTINUE
C
C  READ ZOBS
C
      CALL MTIO(IT1,A9,8704)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 490,490,499
  490 CONTINUE
      CALL MTIO(IT2,A9,8704)
      CALL MTWAIT(IT2,ISTAT,IWDS)
C
C  READ A(512,9)
C
      CALL MTIO(IT1,A9,9216)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IWDS) 7,7,500
    7 CONTINUE
      CALL PCLR(WK2,2,512,15)
      DO 10 I=2,3
      IM1=I-1
```

```
      CALL PMPY(A9(1,I),2,PM(IM1),0,WK1,2,512,15)
      CALL PSUB(WK1,2,WK2,2,WK2,2,512,15)
   10 CONTINUE
C
C  FORM Z-BOUNCE IN WK1=AZERO-ALPHA(1)
C
      CALL PSUB(WK2,2,A9(1,1),2,WK1,2,512,15)
C
C  WRITE Z-BOUNCE AND ALPHA(1) TO OUTPUT ARRAY
C
      CALL PREL(WK1,2,AOUT(1,10),2,512,1)
      CALL PREL(WK2,2,AOUT(1,11),2,512,1)
      CALL PREL(A9(1,4),2,WK1,2,512,1)
      CALL PCLR(WK2,2,512,15)
      CALL PSUB(WK1,2,WK2,2,WK2,2,512,15)
      CALL PMPY(A9(1,5),2,F2B,0,WK1,2,512,15)
      CALL PADD(WK1,2,WK2,2,WK2,2,512,15)
      CALL PMPY(WK2,2,ELG,0,WK2,2,512,15)
      CALL PREL(WK2,2,AOUT(1,12),2,512,1)
      CALL PREL(WK1,2,AOUT(1,13),2,512,1)
      CALL PMPY(A9(1,7),2,F2T,0,WK1,2,512,15)
      CALL PADD(A9(1,6),2,WK1,2,WK2,2,512,15)
      CALL PREL(WK2,2,AOUT(1,14),2,512,1)
      CALL PMPY(AOUT(1,14),2,ELG,0,AOUT(1,14),2,512,15)              C
      CALL PSUB(WK2,2,A9(1,6),2,WK1,2,512,15)
      CALL PREL(WK1,2,AOUT(1,15),2,512,1)
      CALL MTIO(IT2,AOUT,15360)
      CALL MTWAIT(IT2,ISTAT,NWDS)
      N=N+1
      GO TO 1
  499 CONTINUE
  500 CONTINUE
      CALL MTOPEN(IT1,19)
      CALL MTOPEN(IT2,17)
      CALL MTOPEN(IT2,19)
      CALL PCLOS
      RETURN
 9000 CONTINUE
      WRITE(3,3012)
      STOP
 2000 FORMAT(4E15.9)
 2001 FORMAT(40A2,40A2,5A2)
 3000 FORMAT(10X,    22HINPUT TAPE DESCRIPTION/10X,40A2/10X,45A2/)
 3002 FORMAT(10X,23HOUTPUT TAPE DESCRIPTION/10X,40A2/10X,45A2/)
 3012 FORMAT(10X,19HEOF ON HEADER OR FX)
 3013 FORMAT(1X,31HTHE MASS DISTRIBUTION INTEGRALS,5E15.6//)
      END
```

```
:F
      SUBROUTINE MCOR(MLT)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DOUBLE PRECISION TITLES
      COMMON/AZBLCK/AIN(512,15),FTW1(514),FTW2(514),XPOW(514,5),CORFFT(5
     .14,4),IEX(1004)
      COMMON/BZBLCK/ACPOW(257,7),IEP(4866),IDEDCH(17),IZL(66),IYL(24),
     1IXL(48),
     1ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,10,0/
      DATA N,IT1,IDSK,IPRINT,ICDR,IEND/0,8,9,3,2,Z0010/
      DATA IBIT/Z0020/
      SCALE=1.4142136/256.
      CALL MTOPEN(IT1,19)
      IF(MLT) 13,48,13
   13 CONTINUE
      CALL MTOPEN(IDSK,19)
   48 MAXF=30
      CALL MTOPEN(IT1,9)
      IF(MLT) 8,47,8
    8 CONTINUE
      CALL MTOPEN(IDSK,3)
   47 CONTINUE
      CALL MTIO(IT1,ICDHD,85)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 5,5,9000
    5 CONTINUE
      IF(MLT) 17,3,17
   17 CONTINUE
      CALL MTIO(IDSK,ICDHD,85)
      CALL MTWAIT(IDSK,ISTAT,IWDS)
      IF(ISTAT.AND.IBIT) 3,3,9006
    3 CONTINUE
C**** OPEN APOLLO PROCESSOR
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
C****
C**** CLEAR STACKING ARRAYS
      CALL PCLR(ACPOW,2,1799,15)
      CALL PCLR(XPOW,2,2570,15)
    1 CONTINUE
C
C  READ AND WRITE Z
C
      CALL MTIO(IT1,AIN,8704)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEND) 24,24,7000
24    CONTINUE
      IF(MLT) 11,44,11
   11 CONTINUE
      CALL MTIO(IDSK,AIN,8704)
      CALL MTWAIT(IDSK,ISTAT,IWDS)
   44 CONTINUE
      CALL MTIO(IT1,AIN,15360)
      CALL MTWAIT(IT1,ISTAT,NWDS)
      IF(ISTAT.AND.IEND) 2,2,7000
```

```
    2 CONTINUE
      IF(MLT) 12,46,12
   12 CONTINUE
      CALL MTIO(IDSK,AIN(1,7),9216)
      CALL MTWAIT(IDSK,ISTAT,IWDS)
C
C  GET PSD S OF MODAL COOR S - SAVING FFT S FOR XPOW
C
   46 CONTINUE
      DO 50 I=1,6
      IM=I/2
      IP=I+9
      CALL PREL(AIN(1,IP),2,FTW1,2,512,1)
      CALL PFFT(FTW1,FTW2,0,256,13)
      FTW2(513)=FTW2(2)*2.
      FTW2(2)=0.
      FTW2(514)=0.
      FTW2(1)=FTW2(1)*2.
      CALL PMPY(FTW2,2,SCALE,0,FTW2,2,514,15)
      IF((IP/2)*2.0.EQ.IP) GO TO 36
      CALL PREL(FTW2,2,CORFFT(1,IM),2,514,1)
   36 CONTINUE
      CALL PCSM(FTW2,4,FTW1,2,257,15)
      NI=2*IM
      IF(NI-I) 4,7,4
    4 FTW1(1)=0.
    7 CONTINUE
      CALL PADD(ACPOW(1,I),2,FTW1,2,ACPOW(1,I),2,257,15)
   50 CONTINUE
C
C  GET BETA-1 PSD - SAVING FFT
      CALL PREL(AIN(1,8),2,FTW1,2,512,1)
      CALL PFFT(FTW1,FTW2,0,256,13)
      FTW2(513)=FTW2(2)*2.
      FTW2(2)=0.
      FTW2(514)=0.
      FTW2(1)=FTW2(1)*2.
      CALL PMPY(FTW2,2,SCALE,0,FTW2,2,514,15)
      CALL PREL(FTW2,2,CORFFT(1,4),2,514,1)
      CALL PCSM(FTW2,4,FTW1,2,257,15)
      CALL PADD(ACPOW(1,7),2,FTW1,2,ACPOW(1,7),2,257,15)
C
C  NOW GET 5 XPOW'S
C
      DO 75 I=1,4
      IPM=I+1
      IP=2*I+1
      CALL PREL(AIN(1,IP),2,FTW1,2,512,1)
      CALL PFFT(FTW1,FTW2,0,256,13)
      FTW2(513)=FTW2(2)*2.
      FTW2(2)=0.
      FTW2(514)=0.
      FTW2(1)=FTW2(1)*2.
      CALL PMPY(FTW2,2,SCALE,0,FTW2,2,514,15)
      CALL PCCON(FTW2,4,CORFFT(1,I),4,FTW1,4,257,15)
      CALL PADD(XPOW(1,IPM),2,FTW1,2,XPOW(1,IPM),2,514,15)
   75 CONTINUE
C
C  NOW GET XPOW FOR A-1 ALPHA(1)
C
```

```
      CALL PREL(AIN(1,2),2,FTW1,2,512,1)
      CALL PFFT(FTW1,FTW2,0,256,13)
      FTW2(513)=FTW2(2)*2.
      FTW2(2)=0.
      FTW2(514)=0.
      FTW2(1)=FTW2(1)*2.
      CALL PMPY(FTW2,2,SCALE,0,FTW2,2,514,15)
      CALL PCCON(FTW2,4,CORFFT(1,1),4,FTW1,4,257,15)
      CALL PADD(XPOW(1,1),2,FTW1,2,XPOW(1,1),2,514,15)
      N=N+1
      GO TO 1
 7000 CONTINUE
C
C  NORMALIZE POW AND XPOW BY N
C
      DN=1.0/N
      CALL PMPY(XPOW(1,1),2,DN,0,XPOW(1,1),2,2570,15)
      CALL PMPY(ACPOW(1,1),2,DN,0,ACPOW(1,1),2,1799,15)
      CALL MODL(MAXF,MLT)
      IF(MLT) 60,9006,60
   60 CONTINUE
      GO TO 9005
 9000 CONTINUE
      WRITE(IPRINT,3012)
 9005 CONTINUE
      CALL MTOPEN(IDSK,17)
 9006 RETURN
 2001 FORMAT(40A2/40A2/5A2)
 2002 FORMAT(3I5)
 3000 FORMAT(10X,22HINPUT TAPE DESCRIPTION/10X,40A2/10X,45A2/)
 3002 FORMAT(10X,23HOUTPUT TAPE DESCRIPTION/10X,40A2/10X,45A2/)
 3012 FORMAT(10X,13HEOF ON HEADER)
      END
```

:F

```
      SUBROUTINE MODL(MAXF,MLT)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DOUBLE PRECISION TITLES
      DIMENSION IDAT(3)
      DIMENSION R(2),LBL(6)
      DIMENSION WK1(514),WK2(514)
      COMMON/AZBLCK/AIN(512,15),FTW1(514),FTW2(514),XPOW(514,5),CORFFT(5
     .14,4),IEX(1004)
      COMMON/BZBLCK/ACPOW(257,7),IEP(4866),IDEDCH(17),IZL(66),IYL(24),
     1IXL(48),
     1ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
      EQUIVALENCE (IRAST,AIN(1,1))
      EQUIVALENCE (FTW1,WK1),(FTW2,WK2)
      DATA IDAT/2HB1,2HD1,2HC1/
      CALL TITE
      NPTP=MAXF*4
      DO 501 I=1,6
      LBL(I)=I*MAXF/6+.5
  501 CONTINUE
      DM=1.0
      DO 510 I=1,7
      IF(MCV(I)) 8,510,8
    8 CONTINUE
      IF((I.EQ.1).OR.(I.EQ.4).OR.(I.EQ.7)) PRINT 3001,ICDHD
      F=0.
      G=0.
      CALL HPLOT(ACPOW(1,I),NPTP,1,IRAST,66,100,228,F,G)
      PRINT 3003, LBL
      PRINT 3002,(TITLES(J,I),J=1,4)
      CALL PCONR(ACPOW(1,I),2,DM,0,R,2,NPTP,2,15)
      R(1)=R(1)*.25
      RUT=SQRT(R(1))
      IF((I.EQ.3).OR.(I.EQ.5))GO TO 502
      PRINT 3005,R(1),RUT
      GO TO 510
  502 PRINT 3006,R(1),RUT
  510 CONTINUE
      IF(MLT) 1,5,1
    1 CONTINUE
      LMAX=0
      TMAX=-32000.
      DO 6 I=1,120
      IF(ACPOW(I,7).LT.TMAX) GO TO 6
      TMAX=ACPOW(I,7)
      LMAX=I
    6 CONTINUE
      IND=2*LMAX-1
      DO 7 I=1,3
      IP=I+1
      IPT=2*IP
      IF(I.EQ.3) IPT=IPT-1
      IPP=I+2
      CALL PDIV(ACPOW(1,IPT),2,XPOW(1,IPP),4,WK1(1),4,257,15)
      CALL PDIV(ACPOW(1,IPT),2,XPOW(2,IPP),4,WK1(2),4,257,15)
```

```
      IF(I.EQ.1) B1=WK1(1)
      IF(I.EQ.2) D1=WK1(1)
      IF(I.EQ.3) C1=WK1(IND)
    7 CONTINUE
      VEC(1)=C1
      VEC(2)=D1
 3001 FORMAT(1H1,/50X,8HCAR BODY,/50X,40A2)
 3002 FORMAT(54X,4A6,/)
 3003 FORMAT(29X,1H0,2(10X,I2,10X,I2,11X,I2)/60X,5HHERTZ/)
 3005 FORMAT(20X,28HTOTAL POWER IN 30 HERTZ BAND,E15.6,5X,15HG'S MEAN SQ
     .UARE/48X,E15.6,5X,3HRMS/)
 3006 FORMAT(20X,28HTOTAL POWER IN 30 HERTZ BAND,E15.6,5X,25H(RAD/SEC/SE
     .C) MEAN SQUARE,/48X,E15.6,5X,3HRMS/)
 3009 FORMAT(35X,34HPHASE OF MODE SHAPE COEFFICIENT A-,I1,//)
 3013 FORMAT(35X,38HMAGNITUDE OF MODE SHAPE COEFFICIENT A-,I1,//)
 3014 FORMAT(35X,38HREAL PART OF MODE SHAPE COEFFICIENT A-,I1//)
 3015 FORMAT(35X,43HIMAGINARY PART OF MODE SHAPE COEFFICIENT A-,I1//)
 3030 FORMAT(1X,15(1X,10E12.4/)/)
 3040 FORMAT(35X,37HMAGNITUDE OF MODE SHAPE COEFFICIENT  ,A2/)
 3041 FORMAT(35X,37HREAL PART OF MODE SHAPE COEFFICIENT  ,A2/)
 3042 FORMAT(35X,42HIMAGINARY PART OF MODE SHAPE COEFFICIENT  ,A2/)
 3043 FORMAT(35X,33HPHASE OF MODE SHAPE COEFFICIENT  ,A2/)
    5 CONTINUE
      CALL PCLOS
      RETURN
      END
```

:F

```
      SUBROUTINE TITE
      DOUBLE PRECISION RIT,TITLES
      DIMENSION TITLES(4,7)
      COMMON/BKTIT/RIT(4,7),VEC(2)
      DATA((TITLES(I,J),I=1,4),J=1,1)/6HBOUNCE,6H MODE ,2*6H      /
      DATA((TITLES(I,J),I=1,4),J=2,2)/6HFIRST ,6HBENDIN,6HG MODE,6H
     . /
      DATA((TITLES(I,J),I=1,4),J=3,3)/6HPITCH ,6HMODE  ,2*6H      /
      DATA((TITLES(I,J),I=1,4),J=4,4)/6HSECOND,6H BENDI,6HNG MOD,
     .6HE     /
      DATA((TITLES(I,J),I=1,4),J=5,5)/6HROLL M,6HODE   ,2*6H      /
      DATA((TITLES(I,J),I=1,4),J=6,6)/6HSECOND,6H TORSI,6HON MOD,
     .6HE     /
      DATA((TITLES(I,J),I=1,4),J=7,7)/6HFIRST ,6HTORSIO,6HN MODE,6H
     . /
      DO 5 I=1,4
      DO 5 J=1,7
      RIT(I,J)=TITLES(I,J)
    5 CONTINUE
      RETURN
      END
```

```
:F
      SUBROUTINE LLAT
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DOUBLE PRECISION TITLES
      DIMENSION IRY(17,512)
      DIMENSION TEMP(512,4)
      COMMON/AZBLCK/AA9(512,9),YOBS(5,512),WORK1(514),WORK2(514),
     . RSTK(257,4),YSTK(257,4),YP(512,4),EF(512,3)
      COMMON/BZBLCK/SA(514,3),EFP(257,3),G1(4),G2(4),R(2),FFX(4),LBL(6),
     1IAX(3804),IDEDCH(17),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
      EQUIVALENCE(AA9,IRY)
      EQUIVALENCE(AA9(1,1),TEMP(1,1))
      EQUIVALENCE(IRAST,AA9)
      EQUIVALENCE (IDEDCH(17),ID17)
      DATA N,IT1,IT2,IEOF/0,8,9,Z0010/
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,10,0/
      DATA IFL/0/
C
      B=SQRT(2.0)/256.
      NPTP=120
      ELG=32.174/EL2
      DO 150 I=1,6
  150 LBL(I)=5*I
      IF(ID17.NE.0) GO TO 16
C
C   FORM LITQ INVERSE
C
      SUM1=0.0
      CNT=0.0
      SUM2=0.0
      DO 12 J=1,4
      JP=J+12
      IF(IDEDCH(JP).EQ.0) GO TO 15
      X(JP,ICAR)=0.0
      CNT=CNT+1.0
   15 CONTINUE
      FFX(J)=X(JP,ICAR)
      XJPI=X(JP,ICAR)
      X2=XJPI*XJPI
      G1(J)=XJPI*(1.0+VEC(1)*X2)
      G2(J)=1.0+VEC(2)*X2
      SUM1=SUM1+XJPI
      SUM2=SUM2+XJPI*XJPI
   12 CONTINUE
      COEF=4.-CNT
      IF(COEF.LT.2.0) GO TO 16
      GO TO 17
   16 PRINT 18
      RETURN
   18 FORMAT(1H0,10X,'INSUFFICIENT DATA TO CALCULATE MODAL COORDS. FOR
     1LONG-LAT',/)
   17 DEN=COEF*SUM2-SUM1*SUM1
      QI11=SUM2/DEN
      QI12=-SUM1/DEN
```

```
      QI22=COEF/DEN
C
C   INPUT
C
      CALL MTOPEN(IT1,19)
      CALL MTOPEN(IT1,9)
      CALL MTOPEN(IT2,19)
      CALL MTOPEN(IT2,3)
C
      CALL MTIO(IT1,ICDHD,85)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      CALL MTIO(IT2,ICDHD,85)
      CALL MTWAIT(IT2,ISTAT,NWDS)
C
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
      CALL PCLR(EFP,2,771,15)
    1 CONTINUE
C
C   READ Z-DATA INTO A-ARRAY
C
      CALL MTIO(IT1,AA9,8704)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      IF(ISTAT.AND.IEOF) 98,98,7000
   98 CONTINUE
C
C   FLOAT Z-DATA INTO Z-ARRAY
C
      DO 3 I=1,5
      IP=I+12
      CALL PCNEL(IRY(IP,1),17,YOBS(I,1),10,15,512,0)
      CALL PMPY(YOBS(I,1),10,ZSCALE(IP),0,YOBS(I,1),10,512,15)
    3 CONTINUE
      CALL MTIO(IT1,AA9,9216)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      CALL PREL(AA9(1,2),2,AA9(1,7),2,512,1)
C
C   GET NEW Y-PRIME TIME SERIES
C
      DO 11 I=1,512
      DO 9 J=1,4
      JP=J+12
      ZJPI=Z(JP,ICAR)
      TEMP(I,J)=ZJPI*(AA9(I,8)+AA9(I,7)*G1(J)+AA9(I,9)*G2(J))
      YP(I,J)=YOBS(J,I)+TEMP(I,J)
    9 CONTINUE
   11 CONTINUE
C
C   SOLVE FOR Y0-DD AND PSI0-DD
C
      DO 14 J=1,512
      RHS1=0.0
      RHS2=0.0
      DO 4 I=1,4
      RHS1=RHS1+YP(J,I)
      RHS2=RHS2+YP(J,I)*FFX(I)
    4 CONTINUE
      EF(J,1)=QI11*RHS1+QI12*RHS2
      EF(J,2)=QI12*RHS1+QI22*RHS2
      EF(J,3)=YOBS(5,J)+EF(J,2)*Y(17,ICAR)-AA9(J,6)*Z(17,ICAR)
   14 CONTINUE
```

```
      IF(IFL-1) 6,31,31
    6 CONTINUE
      CALL MTIO(IT2,EF,3072)
      CALL MTWAIT(IT2,ISTAT,NWDS)
C
C  OUTPUT
C
C
      CALL PMPY(EF(1,2),2,ELG,0,EF(1,2),2,512,15)
      DO 30 J=1,3
      CALL PFFT(EF(1,J),SA(1,J),0,256,13)
      SA(513,J)=SA(2,J)*2.0
      SA(2,J)=0.0
      SA(514,J)=0.0
      SA(1,J)=SA(1,J)*2.
      CALL PMPY(SA(1,J),2,B,0,SA(1,J),2,514,15)
      CALL PCSM(SA(1,J),4,WORK1,2,257,15)
      WORK1(1)=0.
      CALL PADD(EFP(1,J),2,WORK1,2,EFP(1,J),2,257,15)
   30 CONTINUE
      GO TO 32
   31 CONTINUE
      CALL RES2
C
   32 N=N+1
      GO TO 1
C
 7000 CONTINUE
      IF(IFL-1) 7001,7002,7002
 7001 CONTINUE
      DN=1.0/N
      CALL PMPY(EFP(1,1),2,DN,0,EFP(1,1),2,771,15)
      PRINT 3013,ICDHD
      DM=1.0
      DO 99 J=1,3
      F=0.0
      G=0.0
      CALL HPLOT(EFP(1,J),NPTP,1,IRAST,66,100,228,F,G)
      PRINT 3004,LBL
      IF(J.EQ.3) PRINT 3003
      IF(J.EQ.2) PRINT 3002
      IF(J.EQ.1) PRINT 3001
      CALL PCONR(EFP(1,J),2,DM,0,R,2,NPTP,2,15)
      R(1)=R(1)*.25
      RUT=SQRT(R(1))
      IF(J.EQ.2) PRINT 3006,R(1),RUT
      IF((J.EQ.1).OR.(J.EQ.3)) PRINT 3005,R(1),RUT
   99 CONTINUE
      IF(RES(2)) 34,43,34
   34 CONTINUE
      N=0
      IFL=1
      CALL MTOPEN(IT1,19)
      CALL MTOPEN(IT1,9)
      CALL MTIO(IT1,IHD,85)
      CALL MTWAIT(IT1,ISTAT,IWDS)
      CALL PCLR(YSTK,2,1028,15)
      CALL PCLR(RSTK,2,1028,15)
      GO TO 1
 7002 DN=1.0/N
```

```
      CALL PMPY(YSTK,2,DN,0,YSTK,2,1028,15)
      CALL PMPY(RSTK,2,DN,0,RSTK,2,1028,15)
      DN=100.
      DM=1.
      CALL ELP2(DN,DM)
 3001 FORMAT(57X,9HSWAY MODE,/)
 3002 FORMAT(57X,8HYAW MODE,/)
 3003 FORMAT(57X,17HLONGITUDINAL MODE,/)
 3004 FORMAT(29X,1H0,2(10X,I2,10X,I2,11X,I2)/60X,5HHERTZ/)
 3005 FORMAT(20X,28HTOTAL POWER IN 30 HERTZ BAND,E15.6,5X,15HG'S MEAN SQ
     .UARE/48X,E15.6,5X,3HRMS/)
 3006 FORMAT(20X,28HTOTAL POWER IN 30 HERTZ BAND,E15.6,5X,25H(RAD/SEC/SE
     .C) MEAN SQUARE/48X,E15.6,5X,3HRMS/)
 3013 FORMAT(1H1,/50X,8HCAR BODY,/50X,40A2)
   43 CONTINUE
      CALL MTIO(IT2,17)
      CALL MTIO(IT2,19)
      CALL PCLOS
      RETURN
      END
```

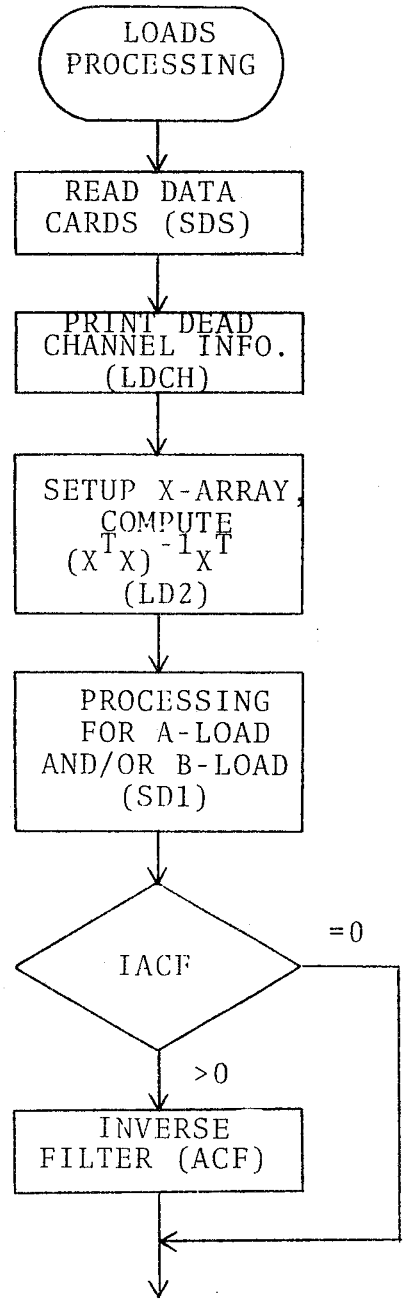
```
:F
      SUBROUTINE RES2
      DOUBLE PRECISION TITLES
      DIMENSION TEMP(512,4)
      COMMON/AZBLCK/AA9(512,9),YOBS(5,512),WORK1(514),WORK2(514),
     . RSTK(257,4),YSTK(257,4),YP(512,4),EF(512,3)
      COMMON/BZBLCK/SA(514,3),EFP(257,3),G1(4),G2(4),R(2),FFX(4),LBL(6),
     1IAX(3804),IDEDCH(17),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/DZBLCK/X(17,3),Y(17,3),Z(17,3)
      COMMON/BKTIT/TITLES(4,7),VEC(2)
      EQUIVALENCE(AA9(1,1),TEMP(1,1))
      B=SQRT(2.)/256.
C
C  COMPUTE Y-CALC. + GAMMA
C
      DO 111 I=1,4
      CALL PREL(EF(1,1),2,WORK1,2,512,1)
      CALL PMPY(EF(1,2),2,FFX(I),0,WORK2,2,512,15)
      CALL PADD(WORK1,2,WORK2,2,WORK1,2,512,15)
      CALL PSUB(TEMP(1,I),2,WORK1,2,WORK1,2,512,15)
C
C  COMPUTE RESIDUALS = YP-Y-CALC. AND GET PSD'S
C
      CALL PSUB(WORK1,2,YOBS(I,1),10,WORK2,2,512,15)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)*2.0
      WORK1(2)=0.
      WORK1(514)=0.
      WORK1(1)=WORK1(1)*2.0
      CALL PMPY(WORK1,2,B,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.
      CALL PADD(RSTK(1,I),2,WORK2,2,RSTK(1,I),2,257,15)
  111 CONTINUE
      DO 100 I=1,4
      CALL PREL(YOBS(I,1),10,WORK2,2,512,1)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)*2.0
      WORK1(2)=0.
      WORK1(514)=0.
      WORK1(1)=WORK1(1)*2.0
      CALL PMPY(WORK1,2,B,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.
      CALL PADD(YSTK(1,I),2,WORK2,2,YSTK(1,I),2,257,15)
  100 CONTINUE
      RETURN
      END
```

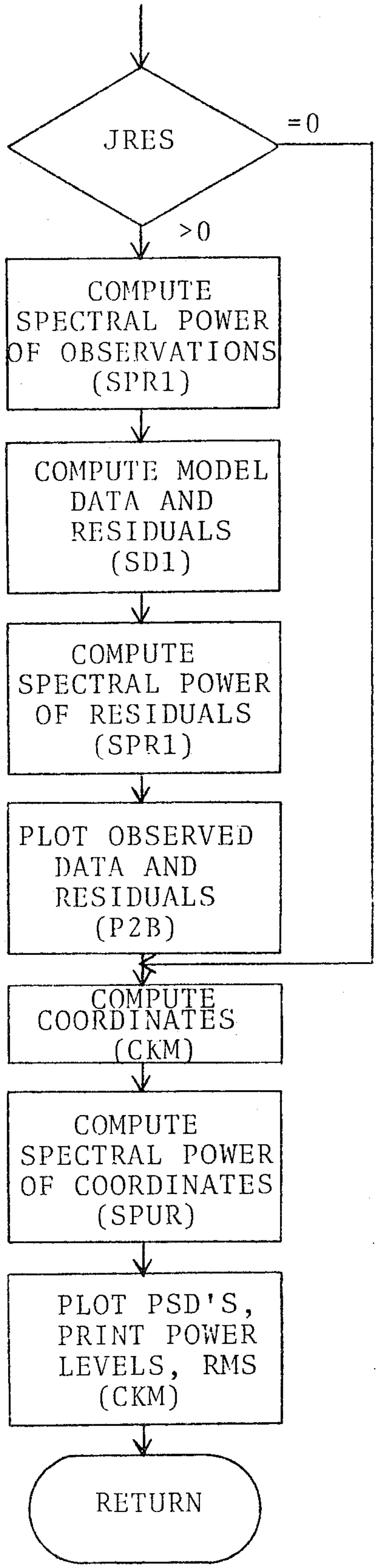
:F

```
      SUBROUTINE DOE
      DOUBLE INTEGER SEC0,SEC1,SEC2,SEC3
      INTEGER RES
      COMMON/AZBLCK/A(512,10),B(512,5),IHD(85),IEX(12227)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),IPRO(3,3
     .),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,7,0/
      CALL IORA(8,200)
      CALL IORA(9,201)
      CALL IORA(10,202)
      N=0
      CALL MTOPEN(8,19)
      CALL MTOPEN(9,19)
      CALL MTOPEN(10,19)
      CALL MTOPEN(8,9)
      CALL MTOPEN(10,9)
      CALL MTOPEN(9,3)
      CALL MTIO(8,IHD,85)
      CALL MTWAIT(8,IS,NW)
      CALL MTIO(10,IHD,85)
      CALL MTWAIT(10,IS,NW)
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
    1 CONTINUE
C
C    READ AND DISCARD ZOBS
C
      CALL MTIO(8,A,8704)
      CALL MTWAIT(8,IS,IW)
      CALL MTIO(8,A,9216)
      CALL MTWAIT(8,JST,JWD)
      IF(JST) 4,3,4
    3 CALL MTIO(10,B,3072)
      CALL MTWAIT(10,IST,IWDS)
      CALL PREL(A(1,9),2,A(1,10),2,512,1)
      CALL PREL(A(1,2),2,A(1,09),2,512,1)
      CALL PREL(B(1,3),2,A(1,01),2,512,1)
      CALL PREL(B(1,1),2,A(1,02),2,512,1)
      CALL PREL(A(1,4),2,A(1,03),2,512,1)
      CALL PREL(A(1,8),2,A(1,04),2,512,1)
      CALL PREL(A(1,5),2,A(1,08),2,512,1)
      CALL PREL(A(1,6),2,A(1,05),2,512,1)
      CALL PREL(B(1,2),2,A(1,06),2,512,1)
      CALL PREL(A(1,7),2,B(1,02),2,512,1)
      CALL PREL(A(1,8),2,A(1,07),2,512,1)
      CALL PREL(B(1,2),2,A(1,08),2,512,1)
      CALL MTIO(9,A,10240)
      CALL MTWAIT(9,KSTS,KWDS)
      N=N+1
      GO TO 1
    4 CALL MTOPEN(9,17)
      CALL MTOPEN(9,19)
      CALL PCLOS
      RETURN
      END
```

A.4 LOADS SOFTWARE

A.4.1 FLOW CHART




JRES
=0
>0
COMPUTE SPECTRAL POWER OF OBSERVATIONS (SPR1)
COMPUTE MODEL DATA AND RESIDUALS (SD1)
COMPUTE SPECTRAL POWER OF RESIDUALS (SPR1)
PLOT OBSERVED DATA AND RESIDUALS (P2B)
COMPUTE COORDINATES (CKM)
COMPUTE SPECTRAL POWER OF COORDINATES (SPUR)
PLOT PSD'S, PRINT POWER LEVELS, RMS (CKM)
RETURN

A.4.2 SYMBOLS FOR LOADS PROCESSING

NTOC is trailer configuration [in SUS].

IACF is non-zero to perform inverse filter.

SCALED are scale factors for input channels.

MREC is number of records.

ILOAD is load configuration.

JRES is residual processing flag.

XT is the X-matrix [in LD2].

SAV = XTX = $X^T X$.

XTS (later) is $(X^T X)^{-1}$.

PRO = $(X^T X)(X^T X)^{-1}$; should be unity matrix.

HOLD = SPACE is $(X^T X)^{-1} X^T$.

XH is X-matrix.

IVEC is the observed data [in SD1].

VEC is the spectral power of the observed data.

WORK1 is the modeled data.

WORK1 (later) are the residuals.

WRRK1 is the spectral power of the residuals.

WORK2 is the spectral power of the observed values [in P2B].

WORK1 is the spectral power of the residuals.

VEC is the percentage of residuals to observed values.

COOR(*,1) is $\ddot{x}_o$ longitudinal acceleration [in CKM].

COOR(*,2) is $\ddot{y}_o$ sway.

COOR(*,3) is $\ddot{z}_o$ bounce.

COOR(*,4) is $\ddot{\theta}_o$ roll.

COOR(*,5) is $\ddot{\phi}_o$ pitch.

COOR(*,6) is $\ddot{\psi}_o$ yaw.

COOR(*,7) is β_{zy} lateral bending.

:F

```
      SUBROUTINE SDS(INDEX,JRES)
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3,LSEC1
      INTEGER RES
      COMMON/AZBLCK/WORK1(514),WORK2(514),VEC(512,8),SCALED(16),
     1IBLK(7168),IVEC(512,16),LOOD(2,4),SPACE(7,8,4),IABC(2),LB,LE,IACF,
     2NTOC,NSTK,MREC,ILOAD,ICHDD(8,2),XH(8,7,4),ICHAN,NCHAN,IGNOR(1101)
      COMMON/BZBLCK/ISMAL(8481),IZL(66),IYL(24),IXL(48),ICDHD(40),
     1IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
   17 FORMAT(3I5)
    1 FORMAT(1H1)
   15 FORMAT(5X,
     14HNO.R,1X,I5,2X,4HNTOC,1X,I5,2X,4HNSTK,1X,I5,2X,3HACF,2X,I5)
   30 FORMAT(8F10.5)
   35 FORMAT(8(2X,F10.5))
      NOREC=0
      NSTK=NREC
      MREC=NREC
      LOOD(1,1)=2
      LOOD(2,1)=1
      LOOD(1,2)=1
      LOOD(2,2)=2
      LOOD(1,3)=4
      LOOD(2,3)=4
      LOOD(1,4)=4
      LOOD(2,4)=3
      READ(2,17) NTOC,IACF
      INDEX=NTOC
      NSTK=NREC
      PRINT 1
      PRINT 15,NREC,NTOC,NSTK,IACF
      READ(2,30)SCALED
      PRINT 35,SCALED
      READ(2,17)MREC,ILOAD,JRES
      LB=1
      IF(NTOC.EQ.2) LB=2
      LE=2
      IF(NTOC.EQ.1) LE=1
      NTOC=(NTOC+1)/2
      DO 600 IK=LB,LE
      II=LOOD(IK,ILOAD)
      IABC(IK)=0
      CALL LDCH(IABC(IK),IK,ISAV,ICHDD(1,IK))
      IXABC=IABC(IK)+1
   16 CALL LD2(II,ISAV,ICHDD(1,IK))
  600 CONTINUE
      RETURN
      END
```

:F

```
      SUBROUTINE LDCH(IABC,IK,ISAV,ICHDD)
      DIMENSION ICHDD(8)
    5 FORMAT(8I5)
    3 FORMAT(//15X,'DEAD CHANNELS',//)
    7 FORMAT(20X,'CH  ',I2)
   16 FORMAT(20X,'INSUFFICIENT INFORMATION TO PROCESS LOAD ',I2,', WILL
     1NOT PROCESS LOAD ',I2)
      READ(2,5) ICHDD
      PRINT 3
      ISAV=0
      ICNT=0
      IABC=0
      DO 10 I=1,8
      IF(ICHDD(I))6,10,6
    6 PRINT 7,I
      IF(I.EQ.1) GO TO 8
      ICNT=ICNT+1
      IF(ICNT-1) 8,8,9
    8 ISAV=I
      IF(IABC.NE.3) IABC=1
      IF(I.EQ.1) IABC=3
      GO TO 10
    9 PRINT 16,IK,IK
      IABC=2
      GO TO 50
   10 CONTINUE
   50 RETURN
      END
```

:F

```
      SUBROUTINE LD2(II,IOUT,ICHDD)
      DOUBLE PRECISION D,L,M,CO,XT,X,XTX,SAV,PRO,HOLD,XCO
      DOUBLE PRECISION XTX6
      DIMENSION XTX6(6,6)
      DIMENSION XCO(64),ICHDD(8)
      COMMON/AZBLCK/WORK1(514),WORK2(514),VEC(512,8),SCALED(16),
     1IBLK(7168),IVEC(512,16),LOOD(2,4),SPACE(7,8,4),IABC(2),LB,LE,IACF,
     2NTOC,NSTK,MREC,ILOAD,ICDDD(8,2),XH(8,7,4),ICHAN,NCHAN,IGNOR(1101)
      COMMON/BZBLCK/XT(7,8),XTX(7,7),X(8,7),PRO(7,7),SAV(7,7),HOLD(7,8),
     1CO(8,2),L(7),M(7),ISMAL(7446),IZL(66),IYL(24),IXL(48),ICDHD(40),
     2IHD(45)
      DATA XCO/4.17D0,-3.33D0,-16.56D0,-2.90D0,0.0D0,-2.90D0,16.54D0,
     1-2.90D0,0.0D0,4.54D0,-16.56D0,4.17D0,0.0D0,4.17D0,16.54D0,
     14.17D0,
     2-4.17D0,-3.33D0,16.56D0,-2.90D0,0.0D0,-2.9D0,-16.54D0,-2.9D0,
     30.0D0,4.54D0,16.56D0,-4.17D0,0.0D0,-4.17D0,-16.54D0,-4.17D0,
     44.0D0,-4.12D0,-16.5D0,-3.83D0,0.0D0,-3.83D0,16.7D0,-3.83D0,
     50.0D0,4.75D0,-16.5D0,4.0D0,0.0D0,4.0D0,16.5D0,4.0D0,
     6-4.0D0,-4.12D0,16.5D0,-3.83D0,0.0D0,-3.83D0,-16.7D0,-3.83D0,
     80.0D0,4.75D0,16.5D0,-4.0D0,0.0D0,-4.0D0,-16.5D0,-4.0D0/
      AFUN(ZZ)=1.0-0.0075*ZZ*ZZ
   45 FORMAT(1X,7D16.8)
   55 FORMAT(1X,8D16.8)
      IXINDEX=16*(II-1)
      DO 4 I=1,8
      DO 2 J=1,2
      IXINDEX=IXINDEX+1
      CO(I,J)=XCO(IXINDEX)
    2 CONTINUE
      DO 3 J=1,7
      XT(J,I)=0.0D0
    3 CONTINUE
    4 CONTINUE
      XT(1,1)=1.0D0
      XT(2,2)=1.0D0
      XT(2,3)=1.0D0
      XT(2,4)=1.0D0
      XT(2,5)=1.0D0
      XT(3,6)=1.0D0
      XT(3,7)=1.0D0
      XT(3,8)=1.0D0
      XT(4,2)=-CO(2,2)
      XT(4,3)=-CO(3,2)
      XT(4,4)=-CO(4,2)
      XT(4,5)=-CO(5,2)
      XT(4,6)= CO(6,2)
      XT(4,7)= CO(7,2)
      XT(4,8)= CO(8,2)
      XT(5,1)= CO(1,2)
      XT(6,1)=-CO(1,1)
      XT(7,2)=AFUN(CO(2,1))
      XT(7,3)=AFUN(CO(3,1))
      XT(7,4)=AFUN(CO(4,1))
      XT(7,5)=AFUN(CO(5,1))
      DO 8 I=1,3
      IP=I+1
      IPP=I+5
      XT(6,IP)=CO(IP,1)
```

```
      XT(5,IPP)=-CO(IPP,1)
    8 CONTINUE
      XT(6,5)=CO(5,1)
      DO 9 I=1,8
      IF(ICHDD(I).EQ.0) GO TO 9
      DO 29 J=1,7
      XT(J,I)=0.0D0
   29 CONTINUE
    9 CONTINUE
      INI=7
      INJ=8
      IMLIM=7
      IF(ICHDD(1).EQ.0) GO TO 10
      INI=6
      INJ=7
      IMLIM=6
      DO 11 I=2,8
      DO 11 J=2,7
      XT(J-1,I-1)=XT(J,I)
   11 CONTINUE
   10 CONTINUE
      DO 12 I=1,INI
      DO 12 J=1,INJ
      X(J,I)=XT(I,J)
   12 CONTINUE
      DO 1 I=1,INI
      PRINT 55,(XT(I,J),J=1,INJ)
    1 CONTINUE
      DO 14 I=1,INI
      DO 14 K=1,INI
      XTX(I,K)=0.0D0
      DO 14 J=1,INJ
      XTX(I,K)=XTX(I,K)+XT(I,J)*X(J,K)
   14 CONTINUE
      DO 15 I=1,INI
      PRINT 55,(XTX(I,J),J=1,INI)
   15 CONTINUE
      IF(ICHDD(1).EQ.0) GO TO 21
      DO 19 I=1,INI
      DO 19 K=1,INI
      XTX6(I,K)=XTX(I,K)
   19 CONTINUE
   21 CONTINUE
      DO 16 I=1,INI
      DO 16 J=1,INI
      SAV(I,J)=XTX(I,J)
   16 CONTINUE
      IF(ICHDD(1).EQ.0) GO TO 23
      CALL MINV(XTX6,IMLIM,D,L,M)
      DO 25 I=1,INI
      DO 25 J=1,INI
      XTX(I,J)=XTX6(I,J)
   25 CONTINUE
      GO TO 27
   23 CONTINUE
      CALL MINV(XTX,IMLIM,D,L,M)
   27 CONTINUE
      PRINT 45,D
      DO 18 I=1,INI
      DO 18 J=1,INI
```

```
      PRO(I,J)=0.0D0
      DO 18 K=1,INI
      PRO(I,J)=PRO(I,J)+SAV(I,K)*XTX(K,J)
   18 CONTINUE
      DO 22 I=1,INI
      PRINT 45,(PRO(I,J),J=1,INI)
   22 CONTINUE
      DO 24 I=1,INI
      DO 24 K=1,INJ
      HOLD(I,K)=0.0D0
      DO 24 J=1,INI
      HOLD(I,K)=HOLD(I,K)+XTX(I,J)*XT(J,K)
   24 CONTINUE
      DO 26 I=1,INI
      PRINT 55,(HOLD(I,J),J=1,INJ)
   26 CONTINUE
      DO 1010 I=1,INI
      DO 1010 J=1,INJ
      SPACE(I,J,II)=SNGL(HOLD(I,J))
      XH(J,I,II)=SNGL(X(J,I))
 1010 CONTINUE
      RETURN
      END
```

```
:F
      SUBROUTINE SD1(IK,JRES)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3,LSEC1
      DIMENSION OVEC(512,7)
      COMMON/AZBLCK/WORK1(514),WORK2(514),VEC(512,8),SCALED(16),
     1IBLK(7168),IVEC(512,16),LOOD(2,4),SPACE(7,8,4),IABC(2),LB,LE,IACF,
     2NTDC,NSTK,MREC,ILOAD,ICHDD(8,2),XH(8,7,4),ICHAN,NCHAN,IGNOR(1101)
      COMMON/BZBLCK/WRRK1(257),WRRK2(257),XWORK1(514),XWORK2(514),
     1IFILL(5397),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      EQUIVALENCE (IBLK       ,OVEC)
      DATA IA,IB,IC,ID,IE,IFF,IS/5*0,1,10/
      DATA JSTAT/Z0010/
  458 FORMAT(5X,'RECORD ',I2,' AFTER 200')
  459 FORMAT(5X,'RECORD ',I2,' AFTER 394')
  460 FORMAT(5X,'RECORD ',I2,' AFTER 416')
      CALL POPEN(IA,IC,ID,IE,IFF,IS,IB)
      IF(JRES) 2,1,2
    1 KKK1=6
      GO TO 3
    2 KKK1=8
    3 DO 16 KKK=KKK1,9
      CALL MTOPEN(KKK,19)
      CALL MTOPEN(KKK,3)
   16 CONTINUE
      IF(JRES) 6,4,6
    4 CONTINUE
      CALL PCLR(WRRK1,2,257,15)
      DO 5 I=1,16
      CALL MTIO(6,WRRK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
    5 CONTINUE
      CALL MTOPEN(6,17)
      CALL MTOPEN(6,19)
      CALL MTOPEN(6,9)
    6 CONTINUE
      IXYZ=IABC(IK)
      IML=-1
      IF(LB.NE.LE) NOREC=1
      ILUN=10-IK
      IF(LB.EQ.LE)ILUN=9
      IK1=IK-1
      ICHAN=IK1*8+1
      NCHAN=ICHAN+7
      ICOUNT=0
      II=LOOD(IK,ILOAD)
      CALL DKINT(1,8192,ISEC1,LSEC1,MREC)
C     AQUIRE A RECORD OF DATA
  150 CONTINUE
      JREC=ICOUNT+1
      CALL DKRD(1,IVEC,8192,JREC,0)
      CALL DKWT(IDSTAT)
      ICOUNT=ICOUNT+1
C     CONVERT TO FLOATING POINT ,SCALE, AND STACK POWER
      K=0
      DO 200 I=ICHAN,NCHAN
      IF(JRES) 152,151,152
```

```
  151 CONTINUE
      CALL MTIO(6,WRRK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
      IF((ISTAT.AND.JSTAT).EQ.JSTAT) PRINT 458,I
  152 CONTINUE
      K=K+1
      CALL PCNFL(IVEC(1,I), 1,VEC(1,K),2,15,512,0)
      CALL PMPY(VEC(1,K),2,SCALED(I),0,VEC(1,K),2,512,15)
      IF(IACF.EQ.0) GO TO 160
      NCH=I*IML
      CALL ACF(VEC(1,K),512,NCH)
  160 CONTINUE
      IF(JRES) 200,161,200
  161 CALL SPWR1(VEC(1,K))
      CALL MTIO(7,WRRK1,514)
      CALL MTWAIT(7,ISTAT,NWDS)
  200 CONTINUE
      DO 300 KK=1,7
      K=KK
      CALL PCLR(WORK2,2,512,15)
      IF(IXYZ-3) 286,285,286
 285  K=KK-1
      IF(KK.EQ.1) GO TO 294
  286 CONTINUE
      DO 290 LL=1,8
      L=LL
      IF(IXYZ-3) 288,287,288
  287 L=LL-1
      IF(LL.EQ.1) GO TO 290
  288 CONTINUE
      RTX1=SPACE(K,L,II)
      CALL PMPY(VEC(1,LL),2,RTX1,0,WORK1,2,512,15)
      CALL PADD(WORK1,2,WORK2,2,WORK2,2,512,15)
  290 CONTINUE
  294 CALL PREL(WORK2,2,OVEC(1,KK),2,512,1)
  300 CONTINUE
      CALL MTIO(ILUN,OVEC(1,1),7168)
      CALL MTWAIT(ILUN,ISTAT,NWDS)
      IF(JRES) 417,384,417
  384 CONTINUE
      DO 400 KK=1,8
      K=KK
      CALL PCLR(WORK1,2,512,15)
      IF(IXYZ-3) 386,385,386
  385 K=KK-1
      IF(KK.EQ.1) GO TO 394
  386 CONTINUE
      DO 390 LL=1,7
      L=LL
      IF(IXYZ-3) 388,387,388
  387 L=LL-1
      IF(LL.EQ.1) GO TO 390
  388 CONTINUE
      XH1=XH(K,L,II)
      CALL PMPY(OVEC(1,LL),2,XH1,0,WORK2,2,512,15)
      CALL PADD(WORK2,2,WORK1,2,WORK1,2,512,15)
  390 CONTINUE
  394 CALL PSUB(WORK1,2,VEC(1,KK),2,WORK1,2,512,15)
      CALL MTIO(6,WRRK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
```

```
      IF((ISTAT.AND.JSTAT).EQ.JSTAT) PRINT 459,KK
  399 CALL SPWR1(WORK1)
      CALL MTIO(7,WRRK1,514)
      CALL MTWAIT(7,ISTAT,NWDS)
  400 CONTINUE
      CALL MTOPEN(7,17)
      CALL MTOPEN(7,19)
      CALL MTOPEN(7,9)
      CALL MTOPEN(6,19)
      CALL MTOPEN(6,3)
      DO 416 IJK=1,16
      CALL MTIO(7,WRRK1,514)
      CALL MTWAIT(7,ISTAT,NWDS)
      IF((ISTAT.AND.JSTAT).EQ.JSTAT) PRINT 460,IJK
      CALL MTIO(6,WRRK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
  416 CONTINUE
      CALL MTOPEN(6,17)
      CALL MTOPEN(6,19)
      CALL MTOPEN(7,19)
  417 CONTINUE
      IF(ICOUNT-NREC) 420,600,600
  420 CONTINUE
      CALL MTOPEN(6,9)
      CALL MTOPEN(7,3)
      IML=1
      GO TO 150
  600 CONTINUE
      CALL MTOPEN(ILUN,17)
      CALL MTOPEN(ILUN,19)
      CALL PCLOS
      RETURN
      END
```

:F

```
      SUBROUTINE SPWR1(X)
      DIMENSION X(1)
      COMMON/BZBLCK/WRRK1(257),SX(257),WORK1(514),WORK2(514),
     1IFILL(5397),IZL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      DATA SCALE/.005524272/
      CALL PREL(X,2,WORK2,2,512,1)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)+WORK1(2)
      WORK1(2)=0.
      WORK1(514)=0.
      WORK1(1)=WORK1(1)+WORK1(1)
      CALL PMPY(WORK1,2,SCALE,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.
      CALL PREL(WRRK1,2,SX,2,257,1)
      CALL PADD(SX,2,WORK2,2,SX,2,257,15)
      CALL PREL(SX,2,WRRK1,2,257,1)
      RETURN
      END
```

:F

```
      SUBROUTINE P2B(NTCC)
      DIMENSION IRAS(6600)
      DIMENSION IBLK(4)
      COMMON/AZBLCK/WRRK1(514),WRRK2(514),VEC(512,8),SCALED(16),
     1JBLK(7168),IVEC(512,16),LOOD(2,4),SPACE(7,8,4),IABC(2),LB,LE,IACF,
     2NTOC,NSTK,MREC,ILOAD,ICHDD(8,2),XH(8,7,4),ICHAN,NCHAN,
     2WORK1(257),WORK2(257),IGNOR(73)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYL(24),IXL(48),ICDHD(40),
     1IHD(45)
      EQUIVALENCE(IVEC,IRAS)
      DATA JSTAT/Z0010/
      DATA IBLK/2H A,2H B,2H A,2H B/
      DATA NTHRU/-1/
      DATA IA,IB,IC,ID,IE,IFF,IS/5*0,1,10/
      DATA ACON/100./
 1001 FORMAT(1H1,/50X,A2,6H- LOAD,/50X,40A2)
   30 FORMAT(60X,4HZOBS,10X,I5)
   20 FORMAT(65X,5HHERTZ)
   10 FORMAT(29X,I1,2(10X,I2,10X,I2,11X,I2))
   40 FORMAT(58X,9HRESIDUALS,5X,I5)
   50 FORMAT(40X,50HPERCENT ERROR, RESIDUAL*100/OBSERVATION FOR SENSOR,
     1I5)
   60 FORMAT(10X,28HTOTAL POWER IN 30 HERTZ BAND)
   70 FORMAT(10X,11HOBSERVATION,2X,E12.5,5X,8HRESIDUAL,2X,
     1E12.5,5X,13HPERCENT ERROR,2X,E12.5)
    7 FORMAT(5X,'EOF ',I2,' AFTER 5')
    8 FORMAT(5X,'EOF ',I2,' 7 READ')
    9 FORMAT(5X,'EOF ',I2,' 6 READ')
      CALL POPEN(IA,IC,ID,IE,IFF,IS,IB)
      NTHRU=NTHRU+1
      IXINDEX=NTCC
      IF(IXINDEX.LT.3) GO TO 1
      IXINDEX=IXINDEX+NTHRU
    1 CONTINUE
      CALL MTOPEN(6,9)
      CALL MTOPEN(7,9)
      DO 5 I=1,8
      CALL MTIO(6,WORK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
      IF((ISTAT.AND.JSTAT).EQ.JSTAT)  PRINT 7,I
    5 CONTINUE
      AMUL=4.*FLOAT(NSTK)
      FAC=1./AMUL
      PRINT 1001,IBLK(IXINDEX),ICDHD
      DO 1000 L=ICHAN,NCHAN
      CALL MTIO(7,WORK2,514)
      CALL MTWAIT(7,ISTAT,NWDS)
      IF((ISTAT.AND.JSTAT).EQ.JSTAT)  PRINT 8,L
      CALL MTIO(6,WORK1,514)
      CALL MTWAIT(6,ISTAT,NWDS)
      IF((ISTAT.AND.JSTAT).EQ.JSTAT)  PRINT 9,L
      CALL PMPY(WORK2,2,FAC,0,WORK2,2,257,15)
      CALL PMPY(WORK1,2,FAC,0,WORK1,2,257,15)
      GMX=0.
      GMN=0.
      IM=0
      CALL HPLOT(WORK2,120,1,IRAS,66,100,228,GMX,GMN)
      PRINT 10,IM,(I,I=5,30,5)
```

```
      PRINT 20
      PRINT 30,L
      GMX=0.
      GMN=0.
      CALL HPLOT(WORK1,120,1,IRAS,66,100,228,GMX,GMN)
      PRINT 10,IM,(I,I=5,30,5)
      PRINT 20
      PRINT 40,L
      SUM1=0.
      SUM2=0.
      SUM3=0.
      DO 45 I=1,120
      ST=WORK2(I)
      WO=WORK1(I)
      SUM1=SUM1+ST
      SUM2=SUM2+WO
      PER=1.
      IF(ST.NE.0.) PER=WO/ST
      PER=PER*100.
      VEC(I,1)=PER
      SUM3=SUM3+PER
   45 CONTINUE
      POW=SUM2/SUM1*ACON
      SUM1=4.*SUM1
      SUM2=4.*SUM2
      GMN=0.
      GMX=ACON
      CALL HPLOT(VEC(1,1),120,1,IRAS,66,100,228,GMX,GMN)
      PRINT 10,IM,(I,I=5,30,5)
      PRINT 20
      PRINT 50,L
      PRINT 60
      PRINT 70,SUM1,SUM2,POW
      IF(L.EQ.NCHAN) GO TO 1000
      PRINT 1001,IBLK(IXINDEX),ICDHD
 1000 CONTINUE
      CALL PCLOS
      RETURN
      END
```

:F

```
      SUBROUTINE CKM(NTOC)
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      INTEGER RES
      DIMENSION IBLK(4)
      DIMENSION ISTRT1(7)
      COMMON/AZBLCK/COOR(512,7),STACK(257,14),IRAS(6600),CIT(7),
     1             LHD(85),WORK1(514),WORK2(514),IFILL(4553)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYL(24),IXL(48),ICDHD(40),
     1IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      DATA ISTRT1/1,7,13,19,25,31,61/
      DATA IA,IB,IC,ID,IE,IFF,IS/5*0,1,7/
      DATA IBLK/2H A,2H B,2H A,2H B/
      IXINDEX=NTOC
      BCON=1035.069756
      CALL POPEN(IA,IC,ID,IE,IFF,IS,IB)
      MREC=0
      ISVREC=0
      IM=0
    5 CONTINUE
      READ(2,10,END=1000) NSET,DF,IFILE
      CALL PCLR(STACK,2,3598,15)
      DO 35 JJ=1,NSET
      IF(JJ.EQ.2) CALL IORA(10,201,0)
      CALL MTOPEN(10,19)
      CALL MTOPEN(10,9)
      DO 30 II=1,NREC
      JK=(JJ-1)*7
      CALL MTIO(10,COOR,7168)
      IF(IEOF(10).NE.0) GO TO 100
      DO 20 I=1,7
      IJ=I+JK
      CALL SPWR(COOR(1,I),STACK(1,IJ))
   20 CONTINUE
      IF(JJ-1) 26,26,27
   26 MREC=MREC+1
      GO TO 28
   27 ISVREC=ISVREC+1
   28 CONTINUE
   30 CONTINUE
      CALL MTOPEN(10,19)
   35 CONTINUE
  100 CONTINUE
      IF(ISVREC.GT.MREC) MREC=ISVREC
      PRINT 1,IBLK(IXINDEX),ICDHD
      CON=1./FLOAT(MREC)
      CALL PMPY(STACK,2,CON,0,STACK,2,3598,15)
      MG=NSET*7
      DO 110 I=1,MG
      J=I
      IF(J.GT.7)J=J-7
      IF(J.GT.3.AND.J.LT.7)
     1            CALL PMPY(STACK(1,I),2,BCON,0,STACK(1,I),2,120,15)
      GMX=0.
      GMN=0.
      CALL HPLOT(STACK(1,I), 120,1,IRAS,66,100,228,GMX,GMN)
      PRINT 120,IM,(IL,IL=5,30,5)
```

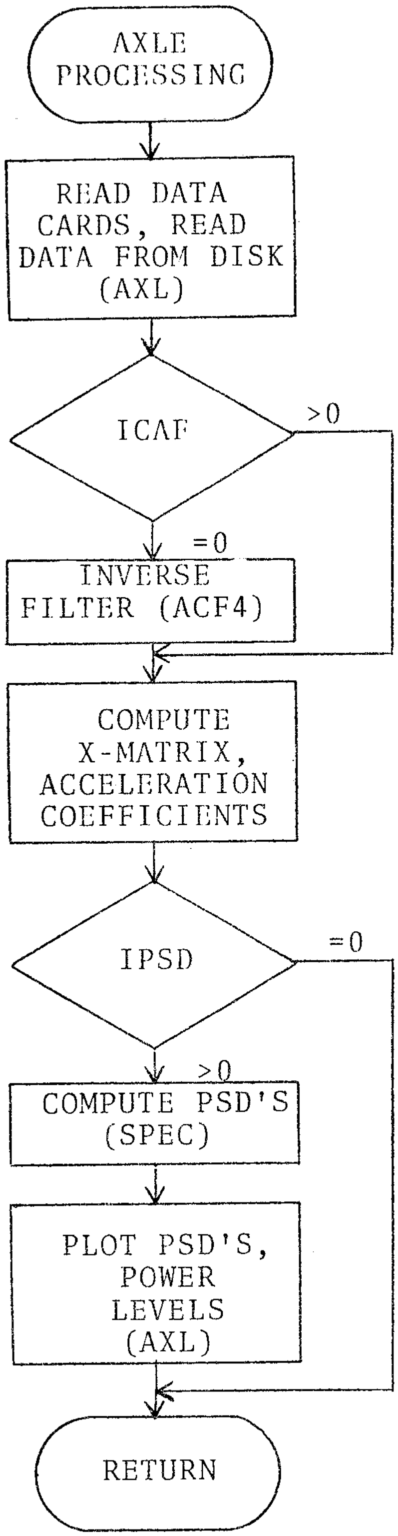
```
      PRINT 130
      IS1=ISTRT1(J)
      IE1=IS1+5
      PRINT 140,(IZTITL(KKK),KKK=IS1,IE1)
      CALL PCONR(STACK(1,I),2,DF,0,COOR(1,1),2,120,2,15)
      PRINT 145,COOR(1,1)
      IF(J.GT.3.AND.J.LT.7) GO TO 148
      PRINT 147
      GO TO 149
  148 CONTINUE
      PRINT 150
  149 CONTINUE
      COOR(1,1)=SQRT(COOR(1,1))
      PRINT 146,COOR(1,1)
      IF(J.EQ.3.OR.J.EQ.6) PRINT 1,IBLK(IXINDEX),ICDHD
      IF(J.NE.7) GO TO 110
      IF (MOD(MG,14)) 112,111,112
  111 IF(I.EQ.14) GO TO 112
      IXINDEX=4
      PRINT 1,IBLK(IXINDEX),ICDHD
      GO TO 110
  112 PRINT 2
  110 CONTINUE
      GO TO 5
 1000 CONTINUE
      CALL PCLOS
      RETURN
   10 FORMAT( I5,F10.5,I5)
    1 FORMAT(1H1,/50X,A2,6H- LOAD,/50X,40A2)
  120 FORMAT(29X,I1,2(10X,I2,10X,I2,11X,I2))
  130 FORMAT(65X,5HHERTZ)
  140 FORMAT(51X,6A2,1X,4HMODE)
  145 FORMAT(42X,30HTOTAL POWER IN THE 0-30HZ BAND,1X,G10.4)
  147 FORMAT(55X,15HG'S MEAN SQUARE)
  150 FORMAT(48X,24H(RAD/SEC/SEC)MEAN SQUARE)
  146 FORMAT(49X,16HROOT MEAN SQUARE,1X,G10.4)
    2 FORMAT(1H1)
 2000 FORMAT(1H0,5X,'NTOC = ',I5)
      END
```

:F

```
      SUBROUTINE SPWR(X,SX)
      DIMENSION X(1),SX(1)
      COMMON/BZBLCK/WORK1(514),WORK2(514),IFILL(6425),IZL(66),IYL(24),
     1IXL(48),ICDHD(40),IHD(45)
      DATA SCALE/.005524272/
      CALL PREL(X,2,WORK2,2,512,1)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)+WORK1(2)
      WORK1(2)=0.
      WORK1(514)=0.
      WORK1(1)=WORK1(1)+WORK1(1)
      CALL PMPY(WORK1,2,SCALE,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.
      CALL PADD(SX,2,WORK2,2,SX,2,257,15)
      RETURN
      END
```

A.5 AXLE SOFTWARE

A.5.1 FLOW CHART



A.5.2 SYMBOLS FOR AXLE SOFTWARE

IPSD is PSD plot flag [in AXL].

IACF is zero to perform inverse filter (1.6 Hz).

SCALED are scale factors for input channels.

A(*,15) contains input data.

XDD = $\ddot{x}_o$ longitudinal acceleration.

ZDD = $\ddot{z}_o$ bounce.

PSIDD = $\ddot{\psi}_o$ yaw.

THETADD = $\ddot{\theta}_o$ roll.

YDD = $\ddot{y}_o$ sway.

B(*,1) - B(*,15) are the coordinates.

PSDA are the PSD's of the coordinates.

```
:F
      SUBROUTINE AXL
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3,LREC
      INTEGER RES
      DIMENSION ISTRT1(5)
      DIMENSION IAX(3)
      DIMENSION Z1(512,5),XDD(512),YDD(512),ZDD(512),
     1          THETADD(512),PSIDD(512),A(512,5),B(512,15)
      DIMENSION WORK1(514),WORK2(514)
      DIMENSION PSDA(257,15),IRAS(6600)
      COMMON/AZBLCK/IWORK(27672)
      COMMON/BZBLCK/INBUF(512,15),INO(514),SCALED(15),JNO(257),
     1IZTITL(66),IYL(24),IXL(48),ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      EQUIVALENCE (IWORK(1),B(1,1)),(IWORK(15361),Z1(1,1)),
     1 (IWORK(20481),A(1,1))
      EQUIVALENCE (IWORK(15361),PSDA(1,1)),(IWORK(1),IRAS(1))
      EQUIVALENCE (IWORK(15361),XDD),(IWORK(16385),YDD),
     1  (IWORK(17409),ZDD),(IWORK(18433),THETADD),(IWORK(19457),PSIDD)
      EQUIVALENCE (IWORK(23543),WORK1(1)),(IWORK(24572),WORK2(1))
      DATA ISTRT1/1,7,13,19,31/
      DATA IA,IB,IC,ID,IE,IFF,IS/5*0,1,10/
      DATA IAX/2HAA,2H A,2HBB/
      DATA XL/-7.5/,YV/44.25/,YH/44.75/,SCL/386.4/,H/6.5/,DEF/0.25/
      CALL POPEN(IA,IC,ID,IE,IFF,IS,IB)
      CALL MTOPEN(9,19)
      CALL MTOPEN(9,3)
      NCH=-1
      ONE=1.
      LREC=0
      READ(2,10,END=2000) IPSD,IACF
      READ(2,12)SCALED
      AF1=.5
      AF2=-1./(YH+YH)
      AF3=1./(YV+YV)
      PRINT 13,SCALED
      PRINT 14,H,XL,SCL,YV,YH,DEF
      CALL DKINT(2,7680,ISEC2,LREC,NREC)
      PRINT 515,NREC,ISEC2
      DO 1000 I=1,NREC
      CALL DKRD(2,INBUF,7680,I    ,0)
      CALL DKWT(ISTAT)
      IF(ISTAT.NE.0) PAUSE 1
      DO 400 LV=0,10,5
      DO 100 IM=1,5
      MA=LV+IM
      CALL PCNFL (INBUF(1,MA),1,A(1,IM),2,15,512,0)
      SC=SCALED(MA)
      CALL PMPY(A(1,IM),2,SC,0,A(1,IM),2,512,15)
      IF(IACF.GT.0) GO TO 50
      MCH=NCH*MA
      CALL ACF4(A(1,IM),512,MCH)
   50 CONTINUE
  100 CONTINUE
      CALL PSUB(A(1,2),2,A(1,1),2,XDD,2,512,15)
      CALL PMPY(XDD,2,AF1,0,XDD,2,512,15)
      CALL PADD(A(1,4),2,A(1,5),2,ZDD,2,512,15)
      CALL PMPY(ZDD,2,AF1,0,ZDD,2,512,15)
```

```
      CALL PADD(A(1,1),2,A(1,2),2,PSIDD,2,512,15)
      CALL PMPY(PSIDD,2,AF2,0,PSIDD,2,512,15)
      CALL PSUB(A(1,5),2,A(1,4),2,THETADD,2,512,15)
      CALL PMPY(THETADD,2,AF3,0,THETADD,2,512,15)
      DO 120 JX=1,512
      YDD(JX)=H*THETADD(JX)-XL*PSIDD(JX)
  120 CONTINUE
      CALL PADD(A(1,3),2,YDD,2,YDD,2,512,15)
      CALL PMPY(PSIDD,2,SCL,0,PSIDD,2,512,15)
      CALL PMPY(THETADD,2,SCL,0,THETADD,2,512,15)
      MA=LV+1
      CALL PREL(Z1(1,1),2,B(1,MA),2,2560,1)
  400 CONTINUE
      NCH=1
      CALL MTIO(9,B,15360)
      CALL MTWAIT(9,ISTAT,NWDS)
 1000 CONTINUE
      CALL MTOPEN(9,17)
      CALL MTOPEN(9,19)
      IF(IPSD.EQ.0) GO TO 2000
      CALL MTOPEN(9,9)
      JM=0
      CALL PCLR(PSDA,2,3855,15)
 1010 CONTINUE
      CALL MTIO(9,B,15360)
      CALL MTWAIT(9,ISTAT,NWDS)
      IF(ISTAT.NE.0) GO TO 1100
      IF(NWDS.NE.15360) PAUSE 7
      DO 1020 I=1,15
      CALL SPEC(B(1,I),PSDA(1,I),WORK1,WORK2)
 1020 CONTINUE
      JM=JM+1
      GO TO 1010
 1100 CONTINUE
      IF(JM.EQ.0) GO TO 2000
      AM=1./FLOAT(JM)
      CALL PMPY(PSDA(1,1),2,AM,0,PSDA(1,1),2,3855,15)
      JM=0
      DO 1300 JJ=1,3
      PRINT 1202,IAX(JJ),ICDHD
      DO 1200 II=1,5
      I=(JJ-1)*5+II
      F=0.
      G=0.
      CALL HPLOT(PSDA(1,I),120,1,IRAS,66,100,228,F,G)
      PRINT 1015,JM,(J,J=5,30,5)
      IS1=ISTRT1(II)
      IE1=IS1+5
      PRINT 1025,(IZTITL(KKK),KKK=IS1,IE1)
      SCALED(1)=0.
      CALL PCONR(PSDA(1,I),2,ONE,0,SCALED,2,120,2,15)
      SCALED(1)=SCALED(1)*DEF
      RMX=SQRT(SCALED(1))
      PRINT 1030,SCALED(1)
      PRINT 1040,RMX
      IF (I.EQ.15) GO TO 1200
      IF(II.EQ.3) PRINT 1202,IAX(JJ),ICDHD
 1200 CONTINUE
 1300 CONTINUE
 2000 CONTINUE
```

```
      CALL MTOPEN(9,19)
      CALL PCLOS
      RETURN
   10 FORMAT(6I5)
   12 FORMAT(8F10.5)
   13 FORMAT(1H1,/5X,13HSCALE FACTORS,/,2(5X,8F10.5,/,))
   14 FORMAT(5X,1HH,2X,F10.5,2X,2HXL,2X,F10.5,2X,3HSCL,2X,F10.5,/,
     1   5X,2HYV,2X,F10.5,2X,2HYH,2X,F10.5,2X,2HDF,2X,F5.3)
  515 FORMAT(5X,2I10)
 1015 FORMAT(29X,I1,2(10X,I2,10X,I2,11X,I2))
 1025 FORMAT(56X,6A2,1X,4HMODE)
 1030 FORMAT(1X,'POWER IN BAND (0 - 30 ) =',F10.5,1X,'(GS-MS)')
 1040 FORMAT(23X,'RMS = ',F10.5)
 1202 FORMAT(1H1,/50X,A2,6H- AXLE,/50X,40A2)
      END
```

:F

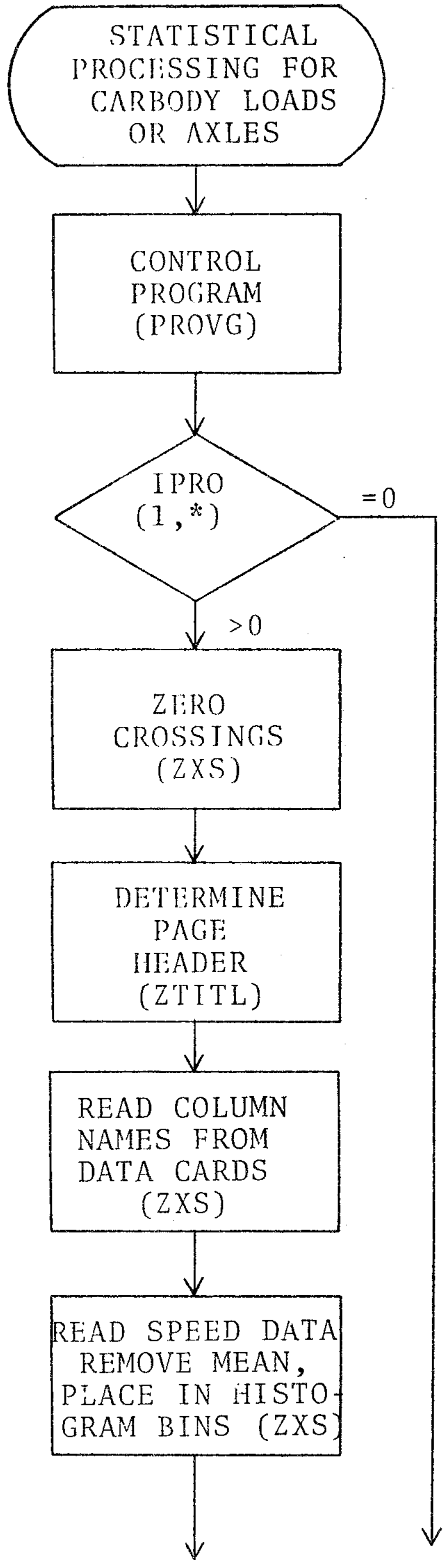
```
      SUBROUTINE SPEC(X,SX,WORK1,WORK2)
      DIMENSION X(1),SX(1)
      DIMENSION WORK1(514),WORK2(514)
      DATA SCALE/.005524272/
      CALL PREL(X,2,WORK2,2,512,1)
      CALL PFFT(WORK2,WORK1,0,256,13)
      WORK1(513)=WORK1(2)+WORK1(2)
      WORK1(2)=0.
      WORK1(514)=0.
      WORK1(1)=WORK1(1)+WORK1(1)
      CALL PMPY(WORK1,2,SCALE,0,WORK1,2,514,15)
      CALL PCSM(WORK1,4,WORK2,2,257,15)
      WORK2(1)=0.
      CALL PADD(SX,2,WORK2,2,SX,2,257,15)
      RETURN
      END
```

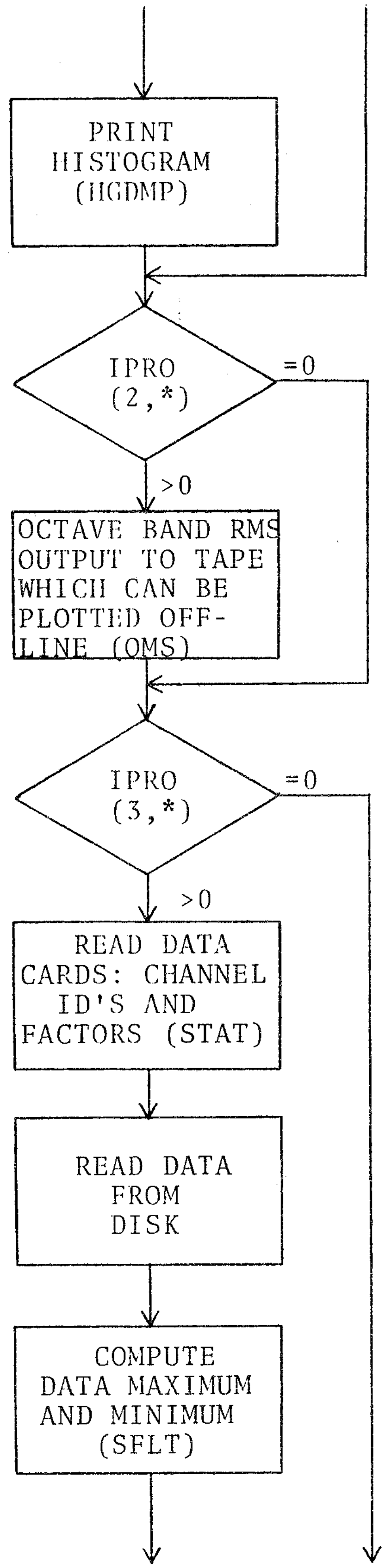
:F

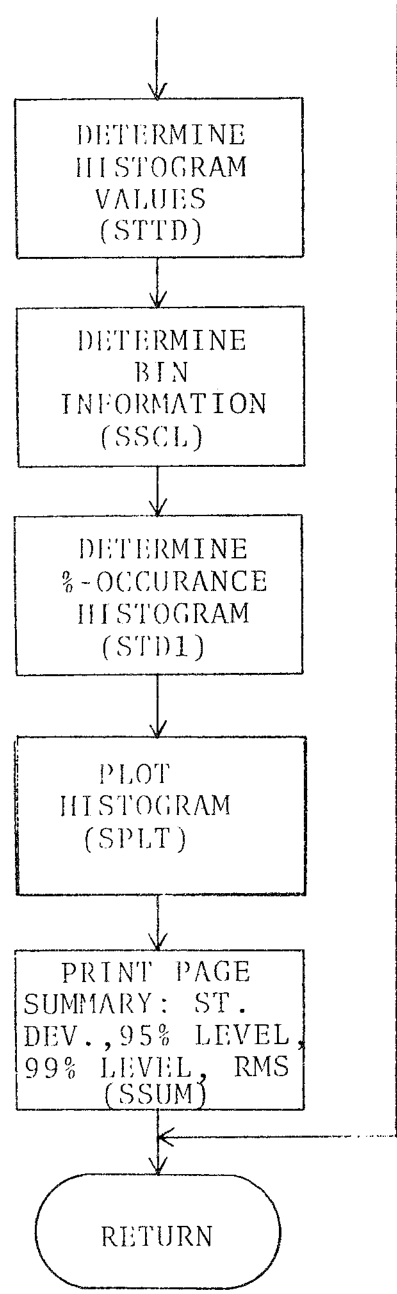
```
      SUBROUTINE ACF4(X,N,ICHNL)
      DOUBLE PRECISION A1,A2,B2,XCHN,X1,X2
      DIMENSION XM1(20),X(1)
      DATA A1,A2,XM1     /13.3189768,.92446525,20*0./
      ICHAN=IABS(ICHNL)
      IF(ICHNL.GT.0) GO TO 10
      XM1(ICHAN)=X(1)+X(1)-X(2)
   10 CONTINUE
      XCHN=XM1(ICHAN)
      DO 100 I=1,N
      X1=X(I)
      X2=A1*(X1-A2*XCHN)
      X(I)=SNGL(X2)
      XCHN=X1
  100 CONTINUE
      XM1(ICHAN)=SNGL(XCHN)
      RETURN
       END
```

A.6 STATISTICAL SOFTWARE

A.6.1 FLOW CHART




PRINT HISTOGRAM (HGDMP)
IPRO (2,*)
=0
>0
OCTAVE BAND RMS OUTPUT TO TAPE WHICH CAN BE PLOTTED OFF-LINE (OMS)
IPRO (3,*)
=0
>0
READ DATA CARDS: CHANNEL ID'S AND FACTORS (STAT)
READ DATA FROM DISK
COMPUTE DATA MAXIMUM AND MINIMUM (SFLT)


DETERMINE HISTOGRAM VALUES (STTD)
DETERMINE BIN INFORMATION (SSCL)
DETERMINE %-OCCURANCE HISTOGRAM (STD1)
PLOT HISTOGRAM (SPLT)
PRINT PAGE SUMMARY: ST. DEV.,95% LEVEL, 99% LEVEL, RMS (SSUM)
RETURN

A.6.2 SYMBOLS USED IN STATISTICAL PROCESSING

HSTOGM is zero-crossing histogram data [in ZXS].

RAW is data read from disk [in STAT].

HIST are the histogram values [in STTD].

ABINC is the bin increment [in SSCL].

BINWIDTH is the bin width [in SSCL].

FACTOR are scale factors [in SSCL].

```
:F
      SUBROUTINE PROG(ID)
      INTEGER CHANID
      DOUBLE PRECISION TOT
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      INTEGER RES
      COMMON/AZBLCK/RAW(512,15),RMS(513,5),AC(513,5),XMEAN(5),
     1STDEV(5),BINWIDTH(5),TEMPS(5),FACTOR(5),FLINE(5),ABSC(5),
     2SCHANG(5),RMSSUM(5),SUM,TOT(5),SF(5),      ABINC(5),Y95(5),
     3Y99(5),P95(5),P99(5),CHANID(5),ICARD(40),SCNCNT,IAX(1834),
     1ICBB(4)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),IPRO(3,3
     .),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
C
    6 IF(IPRO(1,ID)) 7,8,7
    7 CALL IORA(10,30)
      CALL ZXS(ID)
    8 IF(IPRO(2,ID)) 9,11,9
    9 CALL OMS(ID)
   11 IF(IPRO(3,ID)) 16,17,16
   16 CALL IORA(9,201)
      CALL STAT(ID)
   17 RETURN
      END
```

```
:F
      SUBROUTINE ZXS(ID)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3,LSEC3
      INTEGER HSTOGM
      DIMENSION IHDRS(12)
      DIMENSION KCOL(6)
      DIMENSION LCOL(4),MCOL(12)
      DIMENSION HSTOGM(15,51)
      COMMON/AZBLCK/REC(512,16),JSTOGM(15,51),NZERO(15),DNAME(5,16),
     1 DLIM(16),INBUF(512),SUM(15),PEAK(15),IPSIEN(15),RESOLN(15),
     2 SON(2),JAIL(9691),ICBB(4)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYTITL(24),IXTITL(48),
     1ICDHD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      EQUIVALENCE(JSTOGM,HSTOGM)
      DATA IHDRS/2HCA,2HR ,2HBO,2HDY,2H  ,2HLO,2HAD,2H  ,2H  ,2HAX,2HLE,
     12H  /
      DATA DELTAT/5.683E-04/
      DATA LUN/9/
      DATA IA,IC,ID,IE,IF,ISUN,IB/4*0,1,10,0/
      DATA AMILE/5280./
      DO 50505 I=1,4
      IJ=(ID-1)*4+I
      ICBB(I)=IHDRS(IJ)
50505 CONTINUE
C
C    CAR BODY
      NTIMES=1
      GO TO (6,7,8),ID
    6 ICOL=10
      CALL IORA(9,201,0)
      GO TO 9
    7 ICOL=7
      CALL IORA(9,200,0)
      GO TO 9
    8 ICOL=15
      CALL IORA(9,200,0)
    9 CONTINUE
      TDIST=0.0
      DIST=0.0
      ODIST=-1.0
      NRED=0
      IP=2
      NCOL=ICOL
      NCOL1=NCOL+1
      IJREC=0
      MREC=NREC
      DNSN=1./512.
      IWDS=1024*NCOL
      CALL POPEN(IA,IC,ID,IE,IF,ISUN,IB)
      CALL DKINT(3,512,ISEC3,LSEC3,MREC)
      CALL DKWT(ISTAT)
    4 CONTINUE
      KREC=NREC
      WRITE(3,3002) ICBB,ICDHD
      CALL MTOPEN(LUN,19)
      CALL MTOPEN(LUN,9)
```

```
      JCOL=50
      CALL ZTITL(KCOL,JCOL,LCOL,MCOL,NCOL,IWDS)
      DO 1 I=1,NCOL
      READ(2,2001) JCOL,(DNAME(J,I),J=1,3)
      CALL ZTITL(KCOL,JCOL,LCOL,MCOL,NCOL,IWDS)
      WRITE(3,3001) (DNAME(J,I),J=1,3),KCOL,LCOL,DLIM(I),MCOL
      KTMP=MOD(I,5)
      IF(ID.EQ.3.AND.KTMP.EQ.0) PRINT 3004
    1 CONTINUE
      CALL MTOPEN(LUN,19)
      CALL MTOPEN(LUN,9)
    5 CONTINUE
C     THIS ROUTINE READS DATA FROM THE DISC LOGICAL AND WORK FILES
      IJREC=IJREC+1
      CALL MTIO(LUN,REC,IWDS)
      CALL MTWAIT(LUN,ISTAT,NWDS)
      CALL DKRD(3,INBUF,512,IJREC,0)
      CALL DKWT(IDWT)
      CALL PCNFL(INBUF,1,REC(1,16),2,15,512,0)
C  NEED TO AQUIRE AND CONDITION THE SPEED DATA
      CALL PCLR(SUM,2,15,15)
      DO 20 I=1,NCOL
      SON(1)=0.
      CALL PCONR(REC(1,I),2,DNSN,0,SON,2,512,2,15)
      SUM(I)=SON(1)
   20 CONTINUE
      DO 30 I=1,NCOL
      CALL PSUB(SUM(I),0,REC(1,I),2,REC(1,I),2,512,15)
   30 CONTINUE
      NRED=NRED+1
      IF(NRED.GT.1) GO TO 40
      DO 35 I=1,NCOL
      PEAK(I)=REC(1,I)
      IPSIEN(I)=(3.+ SIGN(1.,REC(1,I)))/2.
   35 RESOLN(I)=DLIM(I)/25.
      CALL PCLR(HSTOGM,1,765,0)
      CALL PCLR(NZERO,1,15,0)
      DIST=REC(1,16)*DELTAT
   40 CONTINUE
      DO 100 I=IP,512
      DO 99 J=1,NCOL
      ISIEN=(3.+ SIGN(1.,REC(I,J)))/2.
      IF(ISIEN.EQ.IPSIEN(J)) GO TO (60,70),ISIEN
      IBIN=26-IFIX(PEAK(J)/RESOLN(J))
      IF(IBIN.GT.51)IBIN=51
      IF(IBIN.LT.1)IBIN=1
      HSTOGM(J,IBIN)=HSTOGM(J,IBIN)+1
      PEAK(J)=REC(I,J)
      IPSIEN(J)=ISIEN
      NZERO(J)=NZERO(J)+1
      GO TO 80
   60 CONTINUE
      IF(REC(I,J).LT.PEAK(J)) PEAK(J)=REC(I,J)
      GO TO 80
   70 CONTINUE
      IF(REC(I,J).GT.PEAK(J)) PEAK(J)=REC(I,J)
   80 CONTINUE
   99 CONTINUE
      DIST=DIST+REC(I,16)*DELTAT
      IF(DIST.LT.AMILE) GO TO 100
```

```
      SMILE=DIST/AMILE
      TDIST=TDIST+1.0
      CALL HGDMP(HSTOGM,NZERO,DLIM,SMILE,DNAME,NCOL,ODIST,
     1ICBB,ICDHD)
      DIST=0.0
      CALL PCLR(HSTOGM,1,765,0)
      CALL PCLR(NZERO,1,15,0)
  100 CONTINUE
      IF(NRED.GE.KREC) GO TO 901
      IP=1
      GO TO 5
  901 CONTINUE
      SMILE=DIST/AMILE
      CALL HGDMP(HSTOGM,NZERO,DLIM,SMILE,DNAME,NCOL,ODIST,ICBB,ICDHD)
      TDIST=TDIST+SMILE
      CALL MTOPEN(LUN,19)
      WRITE(3,3003) TDIST
      IF(NTIMES.EQ.2) GO TO 905
      IF(NOREC-1) 905,902,905
  902 CALL IORA(9,201,0)
      NTIMES=NTIMES+1
      GO TO 9
  905 RETURN
 2001 FORMAT(I2,A4,4X,A4,1X,A4,1X,E15.9)
 3001 FORMAT(18X,3A4,5X,6A2,5X,4A2, 9X,E9.2,7X,12A2)
 3002 FORMAT(1H1,/62X,4A2,/50X,40A2,/////56X,'CHANNEL DESCRIPTION',
     1/21X,6HNUMBER,9X,4HMODE,13X,4HTYPE,10X,14HDYNAMIC LIMITS,8X,
     217HENGINEERING UNITS,//)
 3003 FORMAT(10X,26HTOTAL DISTANCE REPRESENTED,F10.3,2X,5HMILES)
 3004 FORMAT(1H )
      END
```

:F

```
      SUBROUTINE ZTITL(KCOL,JCOL,LCOL,MCOL,NCOL,IWDS)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DIMENSION KCOL(6),LCOL(4),MCOL(12),ISTRT1(39),ISTRT2(39),
     1ISTRT3(39)
      COMMON/BZBLCK/ISMAL(8481),IZTITL(66),IYTITL(24),IXTITL(48),
     1ICDHD(40),IHD(45)
      COMMON/AZBLCK/REC(512,16),JSTOGM(15,51),NZERO(15),DNAME(5,16),
     1 DLIM(16),INBUF(512),SUM(15),PEAK(15),IPSIEN(15),RESOLN(15),
     2 SON(2),AMN(2),JAIL(9687),ICBB(4)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      DATA ISTRT1/1,7,13,19,25,31,37,43,49,55,1,7,13,19,25,31,61,1,7,13,
     119,25,31,61,1,7,13,19,31,1,7,13,19,31,1,7,13,19,31/
      DATA ISTRT2/10*1,7*5,7*9,5*13,5*17,5*21/
      DATA ISTRT3/3*1,3*13,2*25,2*37,3*1,3*13,25,3*1,3*13,25,3*1,2*13,
     13*1,2*13,3*1,2*13/
      IF(JCOL.EQ.50) GO TO 40
      IS1=ISTRT1(JCOL)
      IE1=IS1+5
      KK=0
      DO 10 I=IS1,IE1
      KK=KK+1
      KCOL(KK)=IZTITL(I)
   10 CONTINUE
      IS1=ISTRT2(JCOL)
      IE1=IS1+3
      KK=0
      DO 20 I=IS1,IE1
      KK=KK+1
      LCOL(KK)=IYTITL(I)
   20 CONTINUE
      IS1=ISTRT3(JCOL)
      IE1=IS1+11
      KK=0
      DO 30 I=IS1,IE1
      KK=KK+1
      MCOL(KK)=IXTITL(I)
   30 CONTINUE
      GO TO 100
   40 DO 42 I=1,NCOL
      SUM(I)=10.0E10
      PEAK(I)=-10.0E10
   42 CONTINUE
      DO 50 IJK=1,NREC
      CALL MTIO(9,REC,IWDS)
      CALL MTWAIT(9,ISTAT,NWDS)
      DO 45 I=1,NCOL
      CALL PMAX(REC(1,I),2,SON,0,512,15)
      CALL PMIN(REC(1,I),2,AMN,0,512,15)
      IF(PEAK(I).LT.SON) PEAK(I)=SON
      IF(SUM(I).GT.AMN) SUM(I)=AMN
   45 CONTINUE
   50 CONTINUE
      DO 80 I=1,NCOL
      SUM(I)=ABS(SUM(I))
      XTEMP=AMAX1(PEAK(I),SUM(I))
      IF(XTEMP.LE.0.0) GO TO 80
```

```
      NEXP=0
   60 IF(XTEMP.GT.1) GO TO 65
      NEXP=NEXP-1
      XTEMP=XTEMP*10.0
      GO TO 60
   65 ILTMP=IFIX(XTEMP+1.0)
      DLIM(I)=FLOAT(ILTMP)*10.0**NEXP
   80 CONTINUE
  100 RETURN
      END
```

:F

```
      SUBROUTINE HGDMP(HSTOGM,NZERO,DLIM,DIST,DNAME,NCOL,ODIST,
     1ICBB,ICDHD)
      DIMENSION ICBB(4),ICDHD(40)
      DOUBLE PRECISION DCHAR,DBLANK,DLABL(11)
      INTEGER HSTOGM
      DIMENSION HSTOGM(15,51),NZERO(15),DLIM(16),ILABL(17)
      DIMENSION DNAME(5,16)
      DATA DBLANK/6H      /,IBLANK/2H  /
      DATA IDASH/2H--/
      DATA DLABL/6H  1.00,6H  0.80,6H  0.60,6H  0.40,6H  0.20,6H  0.00,
     *   6H -0.20,6H -0.40,6H -0.60,6H -0.80,6H -1.00/
      DATA ILABL/2H P,2H E,2H A,2H K,2H  ,2H V,2H A,2H L,2H U,2H E,2H  ,
     *   2H I,2H N,2H  ,2H G,ZA0A7,2H S/
      DATA ODIST/0.0/
      PRINT 4000,ICBB,ICDHD,(DNAME(1,I),I=1,NCOL)
      PRINT 3999,(DNAME(3,I),I=1,NCOL)
      PRINT 4001,(DLIM(I),I=1,NCOL)
      ICHAR=IBLANK
      DO 100 I=1,51
      IF(MOD(I-1,5).NE.0) GO TO 25
      J=I/5+1
      DCHAR=DLABL(J)
      JCHAR=IDASH
   25 CONTINUE
      IF(I.LT.18.OR.I.GT.34.) GO TO 35
      ICHAR=ILABL(I-17)
   35 CONTINUE
      WRITE(3,4002) ICHAR,DCHAR,JCHAR,(HSTOGM(J,I),J=1,NCOL)
      ICHAR=IBLANK
      JCHAR=IBLANK
      DCHAR=DBLANK
  100 CONTINUE
      DO 110 MI=1,NCOL
      NZERO(MI)=(NZERO(MI)-1)/2
  110 CONTINUE
      WRITE(3,4003) NZERO
      IF(DIST.LT.1.0) GO TO 120
      ODIST=ODIST+DIST
      PDIST=ODIST+1.0
  115 WRITE(3,4004) ODIST
      WRITE(3,4005) PDIST
      GO TO 125
  120 ODIST=ODIST+1.0
      PDIST=ODIST+DIST
      GO TO 115
  125 RETURN
 3999 FORMAT(5X,6HNUMBER,15(2X,A4,2X))
 4000 FORMAT(1H1,50X,4A2,/50X,40A2,
     1           /4X,7HCHANNEL,1X,15(A4,4X))
 4001 FORMAT(5X,7HDYNAMIC,15(8H  +/-   )/5X,6HLIMITS,15E8.2)
 4002 FORMAT(A2,1X,A6,1X,A2,1X,15(I5,3X))
 4003 FORMAT(4X,6HNUMBER,3X,118(1H-)/4X,9HOF CYCLES,15(I5,3X))
C                    ))X3,5I(51,SGNISSORCH9,X4/)-H1(811,X5,OREZH4,X4(TAMROF 3004
 4004 FORMAT(4X,14HDATA FROM MILE,F10.3)
C                                 )SELIMH5,X3,3.01F,DETNESERPER ECNATSIDH02,X4(TAMROF 4004
 4005 FORMAT(6X,12HDATA TO MILE,F10.3)
      END
```

:F
:F

```
      SUBROUTINE OMS(ICOL)
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      DOUBLE INTEGER LSEC3
      INTEGER RES
      DIMENSION HDR(3)
      COMMON/AZBLCK/VEC(512,15),IJOKE(12312)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
      DATA LUN/9/,LUN1/7/
      DATA HDR/4HCARB,4HLOAD,4HAXLE/
      NTIMES=1
      CALL DKINT(3,512,ISEC3,LSEC3,NREC)
      CALL DKWT(LSTAT)
      GO TO (1,2,3),ICOL
    1 CALL IORA(9,201)
      IWDS=10240
      GO TO 5
    2 CALL IORA(9,200)
      IWDS=7168
      GO TO 5
    3 CALL IORA(9,200)
      IWDS=15360
    5 CALL IORA(7,16)
      CALL MTOPEN(LUN,19)
      CALL MTOPEN(LUN,9)
      CALL MTOPEN(LUN1,3)
      CALL MTIO(LUN1,HDR(ICOL),2)
      CALL MTWAIT(LUN1,ISTAT,NWDS)
      DO 300 I=1,NREC
      IJ=I
      CALL MTIO(LUN,VEC,IWDS)
      CALL MTWAIT(LUN,ISTAT,NWDS)
      CALL MTIO(LUN1,VEC,IWDS)
      CALL MTWAIT(LUN1,ISTAT,NWDS)
      CALL DKRD(3,IJOKE,512,IJ,0)
      CALL DKWT(ISTAT)
      CALL MTIO(LUN1,IJOKE,512)
      CALL MTWAIT(LUN1,ISTAT,NWDS)
  300 CONTINUE
      CALL MTOPEN(LUN1,17)
      IF(NTIMES.EQ.2) GO TO 500
      IF(NOREC-1) 500,400,500
  400 NTIMES=NTIMES+1
      CALL IORA(9,201)
      GO TO 5
  500 RETURN
      END
```

```
:F
      SUBROUTINE SFLT(QX,QN,RAW)
      DIMENSION QX(5),QN(5),RAW(512,15)
      DIMENSION AMX(2),AMN(2)
      DO 10 I=1,5
      CALL PMAX(RAW(1,I),2,AMX,0,512,15)
      CALL PMIN(RAW(1,I),2,AMN,0,512,15)
      IF(QX(I).LT.AMX) QX(I)=AMX
      IF(QN(I).GT.AMN) QN(I)=AMN
   10 CONTINUE
      RETURN
      END
```

:F

```
      SUBROUTINE STTD
      INTEGER CHANID
      DOUBLE PRECISION TOT
      COMMON/AZBLCK/RAW(512,15),RMS(513,5),AC(513,5),XMEAN(5),
     1STDEV(5),BINWIDTH(5),TEMPS(5),FACTOR(5),FLINE(5),ABSC(5),
     2SCHANG(5),RMSSUM(5),SUM,TOT(5),SF(5),        ABINC(5),Y95(5),
     3Y99(5),P95(5),P99(5),CHANID(5),IJARD(40),SCNCNT,IAX(1834),
     3ICBB(4)
      COMMON/BZBLCK/ITEMP(512,5),HIST(5,200),IPX(3921),IZL(66),IYL(24),
     1IXL(48),ICARD(40),IHD(45)
      DO 56 K=1,5
      CALL PADD(RAW(1,K),2,SCHANG(K),0,RAW(1,K),2,512,15)
      CALL PMPY(RAW(1,K),2,FACTOR(K),0,RAW(1,K),2,512,15)
      CALL PADD(RAW(1,K),2,1.0,0,RAW(1,K),2,512,15)
      CALL PCOMG(RAW(1,K),2,1.0,0,RAW(1,K),2,512,15)
      CALL PCOML(RAW(1,K),2,200.0,0,RAW(1,K),2,512,15)
      CALL PCNFX(RAW(1,K),2,ITEMP(1,K),1,15,512,0)
      DO 52 KK=1,512
      HIST(K,ITEMP(KK,K))=HIST(K,ITEMP(KK,K))+1.0
      TOT(K)=TOT(K)+1.0
   52 CONTINUE
   56 CONTINUE
      RETURN
      END
```

:F

```
      SUBROUTINE SSCL
      DOUBLE PRECISION TOT
      INTEGER CHANID
      COMMON/AZBLCK/RAW(512,15),RMS(513,5),AC(513,5),XMEAN(5),
     1STDEV(5),BINWIDTH(5),TEMPS(5),FACTOR(5),FLINE(5),ABSC(5),
     2SCHANG(5),RMSSUM(5),SUM,TOT(5),SF(5),        ABINC(5),Y95(5),
     3Y99(5),P95(5),P99(5),CHANID(5),ICARD(40),SCNCNT,IAX(1834),
     2ICBB(4)
      DO 10 I=1,5
      FACTOR(I)=100./SCHANG(I)
   10 CONTINUE
      CALL PREL(SCHANG,2,SF,2,5,1)
      CALL PMPY(SF,2,0.01,0,ABINC,2,5,15)
      CALL PMPY(SF,2,0.01,0,BINWIDTH,2,5,15)
      RETURN
      END
```

```
:F
      SUBROUTINE STD1(NTIMES)
      INTEGER RES
      DOUBLE INTEGER ISEC0,ISEC1,ISEC2,ISEC3
      INTEGER CHANID
      DOUBLE PRECISION TOT
      COMMON/AZBLCK/RAW(512,15),RMS(513,5),AC(513,5),XMEAN(5),
     1STDEV(5),BINWIDTH(5),TEMPS(5),FACTOR(5),FLINE(5),ABSC(5),
     2SCHANG(5),RMSSUM(5),SUM,TOT(5),SF(5),       ABINC(5),Y95(5),
     3Y99(5),P95(5),P99(5),CHANID(5),IJARD(40),SCNCNT,FAC(5),IAX(1824),
     3ICBB(4)
      COMMON/BZBLCK/ITEMP(512,5),HIST(5,200),IPX(3921),IZL(66),IYL(24),
     1IXL(48),ICARD(40),IHD(45)
      COMMON/CZBLCK/ICAR,RES(2),MCL(3),EL2,PM(2),F2T,ZSCALE(17),
     1 IPRO(3,3),IREC,NOREC,NREC,ISEC0,ISEC1,ISEC2,ISEC3,MCV(7)
  100 FORMAT(5(12X,'CH ',I2,3X),12X,F6.0,6H SCANS)
  101 FORMAT(2(12X,'CH ',I2,3X),72X,F6.0,6H SCANS)
 1000 FORMAT(1X,5(F9.3,F11.5))
 1001 FORMAT(1H1,26X,'P E R C E N T   O C C U R R E N C E   H I S T O G
     1R A M  ',/26X,40A2)
 1003 FORMAT(/,' ST. DEV. ',5(F11.4,9X))
C
C  COMPUTE MEAN + DEVIATIONS IN TIME DOMAIN
C
      IJK=5
      IF(NTIMES.EQ.2) IJK=2
      XREC=NREC
      SCNCNT=XREC*512.
      CALL PCLR(XMEAN,2,5,15)
      CALL PCLR(STDEV,2,5,15)
      DO 15 I=1,5
      TEMPS(I)=100./TOT(I)
      RINC=-SF(I)
      RINC2=RINC+(ABINC(I)/2.)
      DO 5 J=1,200
      XTEMP=HIST(I,J)/TOT(I)
      XMEAN(I)=XMEAN(I)+XTEMP*RINC2
      STDEV(I)=STDEV(I)+XTEMP*(RINC*RINC+RINC*ABINC(I)+ABINC(I)*
     1ABINC(I)/3.)
      RINC=RINC+ABINC(I)
      RINC2=RINC2+ABINC(I)
      HIST(I,J)=HIST(I,J)*TEMPS(I)
    5 CONTINUE
      RMSSUM(I)=SQRT(STDEV(I))*FAC(I)
      STDEV(I)=STDEV(I)-XMEAN(I)*XMEAN(I)
      STDEV(I)=SQRT(STDEV(I))
   15 CONTINUE
C
C  PRINT + PLOT HISTOGRAMS
C  SET PRINT PAGE + PLOT HEADINGS
      PRINT 1001,ICARD
C           PRINT PAGE SUBHEADINGS
    2 IF(IJK.EQ.5) GO TO 3
      PRINT 101,(CHANID(JJ),JJ=1,IJK),TOT(1)
      GO TO 4
    3 PRINT 100,CHANID,TOT(1)
    4 CONTINUE
      LINE=4
      DO 200 I=1,5
```

```
      ABSC(I)=-SF(I)
      FLINE(I)=0.0
200   CONTINUE
      KK=1
      N=0
      DO 285 K=1,6
      GO TO (210,220,230,230,220,210),K
210   JMAX=10
      JLUMP=4
      GO TO 240
220   JMAX=10
      JLUMP=2
      GO TO 240
230   JMAX=40
      JLUMP=1
240   DO 282 J=1,JMAX
      DO 254 I=1,IJK
245   FLINE(I)=0.0
247   DO 253 JJ=1,JLUMP
      KSUB=KK+JJ-1
      ABSC(I)=ABSC(I)+ABINC(I)
  250 FLINE(I)=FLINE(I)+HIST(I,KSUB)
253   CONTINUE
2532  FLINE(I)=FLINE(I)/JLUMP
254   CONTINUE
      N=N+1
      KK=KK+JLUMP
      IF(LINE.GT.64) GO TO 260
  255 PRINT 1000,(ABSC(I),FLINE(I),I=1,IJK)
      LINE=LINE+1
      GO TO 282
  260 PRINT 1001
  270 LINE=1
      GO TO 255
282   CONTINUE
285   CONTINUE
C
C   CALL HPLOT
C
      PRINT 1003,(STDEV(JJ),JJ=1,IJK)
      RETURN
      END
```

```
      DO 90 K=1,5
      RMSSUM(K)=0.
      TOT(K)=0.0
90    CONTINUE
      IPAGE=IPAGE+1
98    RETURN
100   FORMAT(1H1,30X,'V I B R A T I O N   A N A L Y S I S   S U M M A R Y'
     */,30X,40A2,//)
  101 FORMAT(37X,5('CH ',I2,11X),/)
 1001 FORMAT(37X,2('CH ',I2,11X),/)
  102 FORMAT(4X,'STANDARD DEVIATION',6X,5(5X,F9.3,2X),/)
  103 FORMAT(4X,A2,1X,7A2,7X,5(5X,F9.3,2X),/)
  106 FORMAT(4X,'RMS',21X,5(5X,F9.3,2X),/)
105   FORMAT(////////)
110   FORMAT(31X,'T I M E   P R O C E S S E D',F10.2,4X,'S E C O N D S')
      END
```

:F

```
      SUBROUTINE SSUM(NTIMES)
      DIMENSION IPRCNT(7)
      INTEGER CHANID
      DOUBLE PRECISION TOT
      COMMON/AZBLCK/RAW(512,15),RMS(513,5),AC(513,5),XMEAN(5),
     1STDEV(5),BINWIDTH(5),TEMPS(5),FACTOR(5),FLINE(5),ABSC(5),
     2SCHAN6(5),RMSSUM(5),SUM,TOT(5),SF(5),      ABINC(5),Y95(5),
     3Y99(5),P95(5),P99(5),CHANID(5),IJARD(40),SCNCNT,IAX(1834),
     1ICBB(4)
      COMMON/BZBLCK/ITEMP(512,5),HIST(5,200),IPX(3921),IZL(66),IYL(24),
     1IXL(48),ICARD(40),IHD(45)
      DATA IPRCNT/2HPE,2HR ,2HCE,2HNT,2H L,2HEV,2HEL/
      DATA I95,I99/2H95,2H99/
C        COMPUTE 95 + 99 PER CENT LEVELS
      IJK=5
      IF(NTIMES.EQ.2) IJK=2
   20 DO 40 I=1,IJK
      ISUB=I
      SUM2=0.
      OPCT=0.
      THRESH=.95
      DO 15 J=1,200
   15 HIST(I,J)=HIST(I,J)/TEMPS(I)
      ASSIGN 50 TO N1
      DO 35 J=1,100
      J1=100-(J-1)
      SUM2=SUM2+HIST(ISUB,J1)+HIST(ISUB,J+100)
      PCT=SUM2/TOT(1)
30    IF(PCT.GT.THRESH) GO TO N1
      OPCT=PCT
35    CONTINUE
40    CONTINUE
      GO TO 75
50    Y95(I)=(.95-OPCT)*BINWIDTH(I)/(PCT-OPCT)
      P95(I)=(FLOAT(J-1)*BINWIDTH(I)+Y95(I))
      THRESH=.99
      ASSIGN 60 TO N1
      GO TO 30
60    Y99(I)=(.99-OPCT)*BINWIDTH(I)/(PCT-OPCT)
      P99(I)=(FLOAT(J-1)*BINWIDTH(I)+Y99(I))
      GO TO 40
C        PRINT SUMMARY PAGE
   75 CONTINUE
      PRINT 100,ICARD
      PRINT 105
      IF(IJK.EQ.5) GO TO 10
      PRINT 1001,(CHANID(JJ),JJ=1,IJK)
      GO TO 14
   10 PRINT 101,CHANID
   14 CONTINUE
      PRINT 102,(STDEV(JJ),JJ=1,IJK)
      PRINT 103,I95,IPRCNT,(P95(JJ),JJ=1,IJK)
      PRINT 103,I99,IPRCNT,(P99(JJ),JJ=1,IJK)
      PRINT 106,(RMSSUM(JJ),JJ=1,IJK)
      PRINT 105
      S=SCNCNT/128.
      PRINT 110,S
      CALL PCLR(HIST,2,1200,13)
```