USER'S GUIDE FOR A REVISED COMPUTER PROGRAM
TO ANALYZE THE LRC 16' TRANSONIC DYNAMICS
TUNNEL ACTIVE CABLE MOUNT SYSTEM

by J. Chin and P. Barbero

Prepared under Contract No. NAS 1-10635-22 by
GRUMMAN AEROSPACE CORPORATION
BETHPAGE, N. Y.
for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
July, 1975
FOREWORD

This report is submitted to the NASA Langley Research Center in partial fulfillment of Master Agreement Contract Number NAS 1-10635-22. Part of this contract involves the revision of an existing digital program to analyze the stability of models mounted on a two-cable mount system used in the LRC 16' transonic dynamics tunnel. The program revisions, discussed in this report, will allow for analysis of an active feedback control system to be used for controlling the free-flying models. This report is considered a supplement to CR-132313 and not a replacement for it.

Mr. R. Herr of the NASA Langley Research Center is the technical monitor. Mr. Frederick Berger of the Grumman Aerospace Corporation is the Master Agreement program manager, and Mr. Paul Barbero is the project engineer.
LIST OF SYMBOLS

\[ F_m = \text{Command voltage from feedback loop} \sim \text{volts} \]
\[ E_{mo} = \text{Externally applied input voltage} \sim \text{volts} \]
\[ E_{mtot} = \text{Total voltage to torque motor} \sim \text{volts} \]
\[ G = \text{Friction in active cable system} \sim \text{in. lbs/rps} \]
\[ I_a = \text{Amperes in motor} \sim \text{amps} \]
\[ J_m = \text{Inertia of active cable system} \sim \text{inches}^4 \]
\[ K_T = \text{Motor torque constant} \sim \text{in. lbs/amp} \]
\[ K_v = \text{Motor velocity constant} \sim \text{volts/rps} \]
\[ K_q = \text{Model pitch rate feedback gain} \sim \text{volts/rps} \]
\[ K_{pm} = \text{Pitch motor rate (tachometer) feedback gain} \sim \text{volts/rps} \]
\[ K_{bm} = \text{Motor position feedback gain} \sim \text{volts/rad} \]
\[ K_r = \text{Model yaw rate feedback gain} \sim \text{volts/rps} \]
\[ K_{ym} = \text{Yaw motor rate feedback gain} \sim \text{volts/rps} \]
\[ K_{pm} = \text{Motor position feedback gain} \sim \text{volts/rad} \]
\[ L_a = \text{Motor armature inductance} \sim \text{henry} \]
\[ \gamma = \text{Rolling moment} \sim \text{ft. lb.} \]
\[ M = \text{Pitching moment} \sim \text{ft. lb.} \]
\[ N = \text{Yawing moment} \sim \text{ft. lb.} \]
\[ Q_o = \text{Output torque from motor} \sim \text{in. lb.} \]
\[ Q_L = \text{Load torque on motor} \sim \text{in. lb.} \]
\[ R_G = \text{Motor armature resistance} \sim \text{ohms} \]
\[ K_d = \text{Torque motor pulley radius} \sim \text{in.} \]
\[ s = \text{Laplace operator} \]
\[ \Delta T = \text{Cable tension change} \sim \text{lbs.} \]
\[ \Delta T_c = \text{One half the total cable tension change due to active cable system} \sim \text{lbs.} \]
\[ \Delta T_F = \text{Front cable tension change due to fixed length contraint} \sim \text{lbs.} \]
\[ \Delta T_{fb} = \text{one half the cable tension change due to feedback} = \frac{1}{2} \Delta T \sim \text{lbs.} \]

\[ \Delta T_i = \text{externally applied tension change} \sim \text{lbs.} \]

\[ \delta T = \text{Tension change on one side of torque motor} \sim \text{lbs.} \]

\[ X = \text{Axial force exerted on model} \sim \text{lbs.} \]

\[ Y = \text{Side force exerted on model} \sim \text{lbs.} \]

\[ Z = \text{Vertical force exerted on model} \sim \text{lbs.} \]

\[ (x, y, z) = \text{Model translational displacement} \sim \text{ft.} \]

\[ (\theta, \psi, \phi) = \text{Model angular displacement in pitch, yaw, and roll resp.} \sim \text{rad.} \]

\[ B_g = \text{Lateral wind gust} \sim \text{rad.} \]

\[ \alpha_g = \text{Vertical wind gust} \sim \text{rad.} \]

\[ \delta_a = \text{Aileron deflection} \sim \text{rad.} \]

\[ \delta_e = \text{Elevator deflection} \sim \text{rad.} \]

\[ \delta_r = \text{Rudder deflection} \sim \text{rad.} \]

\[ \theta_m = \text{Vertical plane torque motor pulley angular displacement} \sim \text{rad.} \]

\[ \psi_m = \text{Lateral plane torque motor pulley angular displacement} \sim \text{rad.} \]
# TABLE OF CONTENTS

LIST OF SYMBOLS ................................................................................. 111

1.0 - INTRODUCTION ........................................................................... 1

2.0 - ACTIVE FEEDBACK CONTROL SYSTEM LOGIC ......................... 2

3.0 - FLYING CABLE SNUBBER SYSTEM ........................................... 7
  3.1 - LONGITUDINAL AXIS ................................................................ 7
  3.2 - LATERAL/DIRECTIONAL AXIS ................................................ 8

4.0 - ADDITIONAL PROGRAM OPTIONS .............................................. 9
  4.1 - TRANSFER FUNCTIONS ............................................................ 9
  4.1.1 - LONGITUDINAL AXIS ......................................................... 9
  4.1.2 - LATERAL/DIRECTIONAL AXIS .......................................... 10
  4.2 - FREQUENCY RESPONSE ......................................................... 12
  4.3 - WIND OFF CHARACTERISTICS .............................................. 13
  4.4 - CABLELESS MODEL CHARACTERISTICS ............................ 13

5.0 - REVISED DATA INPUT ............................................................... 14

REFERENCES .......................................................................................... 31

APPENDIX A - A Discussion of the Differences in Cable Attachment Points
  Between the Passive and Active Cable Mount System ................. 32

APPENDIX B - Derivation of Motor and Cable Tension Equations .... 35

APPENDIX C - Program Listing ............................................................ 37

APPENDIX D - Sample Input ............................................................... 99

APPENDIX E - Sample Output ............................................................. 101

FIGURES .............................................................................................. 110
I. INTRODUCTION

In accordance with the requirements set forth under NASA Master Agreement NAS 1-10635, Development and Implementation of Space Shuttle Structural Dynamics Modeling Technology - Task Order Number 22, the following report is submitted.

Contained in this report is a discussion of the updates to the digital computer program originally written under Task Order Number 9 and described in NASA-CR-132313. The original program modeled the dynamic characteristics of aeroelastically-scaled models "flown" on the two-cable mount system in the Langley Research Center 16' Transonic Dynamics Tunnel. The updated digital program contains the original equations plus the necessary additional equations to model an active feedback system presently being developed. The capability of analyzing a proposed new snubber system is also included. Program options and output have been expanded to include complete transfer function characteristics (numerator and denominator), frequency response data, wind-off and free airframe (w/o cable effects) characteristics.

The discussions in this report will cover only the changes made to the original program. It is assumed that CR-132313 will be used in conjunction with this report to obtain full understanding of the program.
2.0 **ACTIVE FEEDBACK CONTROL SYSTEM LOGIC**

The purpose of the active feedback control system is to artificially augment the stability of the cable mounted model by modulating the cable tension. There are two cables used to suspend the model in the tunnel. The tensions of these cables are controlled independently by two torque motors. Generally one cable lies in the vertical plane and the other in the horizontal plane. The vertically mounted cable is used to control the longitudinal dynamics of the system. The horizontally mounted cable is used to control the lateral-directional dynamics.

The cables are assumed to be attached to hard points on the model rather than to the tunnel wall as it was in the original program. This is necessary to effectively transform the tension change in the cable imparted by the torque motor to stabilizing forces and moments on the model. The differences between this system and the original inactive cable system and the ability of the present program to analyze both setups are discussed in detail in Appendix A.

Figure 1 presents the sign convention used in the derivation of the active cable feedback logic. This figure is generalized to account for both vertical front and rear cables as well as horizontal front and rear cables.

\( \theta_m \) and \( \varphi_m \) are torque motor pulley angular displacements in the vertical and horizontal planes respectively. Note that the sense of rotation is unaltered whether the cable is located in the front or rear. Positive motor rotation corresponds to an increase in cable tension on the sides noted in the figure by "+". Positive \( \Delta T \) is an increase in cable tension and negative \( \Delta T \) is a decrease in cable tension. Positive pulley displacements results in a positive rotational moment imparted by the cable onto the model. The letters "M" and "N" show the direction of the moments induced by the positive motor rotation.
Figures 2 and 3 show block diagrams of the cable mount system with feedback loops for the longitudinal and the lateral-directional modes respectively. These two figures are similar and the discussion of figure 2 applies equally to figure 3.

In figure 2, the block in the forward loop represents the basic inactive cable mount system discussed in reference 1. A change in cable tension, \( \Delta T_c \), will result in a model motion defined by variables \( x, z, \) and \( \theta \).

The multi-feedback loops shown represent the active feedback logic, motor dynamics and system friction. The feedback loop containing the gains \( K_q, K_{\dot{\theta}} \), and \( K_{\theta_m} \) are the active elements controlling the torque motor. They are respectively, the model pitch rate gyro gain, the motor rate or tachometer gain, and the motor pulley position gain. The signals emanating from these elements are summed to give a voltage \( E_m \). This voltage is combined with any externally applied test voltage, \( E_{TD} \), to give a total voltage used to drive the torque motor.

The block containing the notation, "\( \theta_m = f(x, z, \theta) \)" represents the geometric relation between the model motion and the pulley motion. This is derived by determining the movement of the cable, \( \Delta l \), as a function of the model motion. The "\( \Delta l \)" is the length of cable passing over the pulley. This value is divided by the pulley radius to determine the angular displacement of the pulley, \( \theta_m \).

The term, \( \frac{K_m}{R_a + sL_a} \), contained in various blocks represents the torque motor characteristics. \( K_m \) is the motor torque constant, \( R_a \) and \( L_a \) are the motor resistance and inductance respectively, and \( s \) is the Laplace operator. A detailed derivation of the motor dynamics is presented in Appendix B.
The output torque from the motor is reduced by the back EMF of the motor as well as by the motor inertia and system friction. This is reflected in the remaining two feedback loops. The $K_v$ term represents the back EMF. The $J_M$ and $G$ terms are the system inertia and friction.

The friction gain, $G$, is proportional to the pulley rotational rate. Reference 2 shows that for perturbation analysis, the coulomb friction can be replaced by a term proportional to the rotational rate.

The net output torque is divided by the pulley radius, $r_d$, to determine the total tension change in the cable. If the cable mass is assumed negligible, the total tension can be replaced by a $\Delta T$. The magnitude of $\Delta T$ is half the total cable tension. The $\Delta T$ is a positive tension on one side of the pulley and a negative tension on the other side. This accounts for the factor of two in the block containing $2r_d$. A derivation of this concept is shown in Appendix B.

The block diagram is written in the conventional manner in which the cable tension feedback, $\Delta T_{fb}$, is subtracted from the input $\Delta T_i$. The signs are accordingly adjusted. The loop, however, remains consistent with the sign convention of figure 1.

In figure 3, the block diagram differs only in the equations which the block in the forward loop represents. Here, the block represents the lateral-directional perturbation equations of motion. $Y$, $V$, $\phi$ are the perturbation variables. The feedback gains $K_r$, $K_v$, and $K_\phi$ are the model yaw rate gyro gain, the horizontal cable torque motor tachometer gain, and the corresponding pulley displacement gain respectively.

The logic in the two block diagrams are modelled in the program using
an expanded polynomial matrix representation. These matrices are shown in figure 4 and 5. They correspond to expanded versions of the basic matrices shown in figures 6.3 and 7.2 of reference 1. The following discussion of figure 4 applies equally to figure 5.

In the longitudinal mode, the basic cable mount system without feedback is represented by the 4 x 4 matrix in the upper left-hand corner of figure 4. The additional cable tension modulation due to the active feedback logic, including motor and pulley dynamics, is represented by the added $\Delta T_c$ terms in equations 1 through 3. The coefficients of $\Delta T_c$ are derived from equations 5.4-3.3 and 5.4-8 of reference 1.

The motor dynamics are defined by equation 5. Equation 6 defines the geometric relation between pulley displacement and model motion. Equation 7 defines the control law. Equations 9 and 10 represent the summation junctures in the block diagram and equation 8 is an auxiliary equation relating pulley rate to its displacement.

In figure 5, the basic system is represented by the 3 x 3 matrix in the upper left-hand corner. The extension of this basic model to include active feedback is via the $\Delta T_c$ terms in equations 1 through 3. The remaining equations are similar to those of figure 4. The only difference being that these equations represent the lateral-directional mode.

The equations of figures 4 and 5 are implemented in subroutines LONG and LAT respectively. Figures 6 and 7 show the flow charts for these subroutines.

The expanded matrices are activated in the program by KODE (13). When this code is greater than zero, the program will read in additional data to define the active feedback parameters. These parameters are tabulated in Section 5.0.
Open and closed loop characteristic roots as well as numerator roots can be derived from these matrices. The procedure for obtaining this information from the program as discussed in Section 4.0.
3.0 FLYING CABLE SNUBBER SYSTEM

The snubber system used the basic flying cables with a large increase in rear cable tension providing the "snubbing" action. When the snubber system is activated the following sequence of events occurs:

1) the rear cable tension is increased to some predetermined level.
2) Next, disc brakes are applied directly to each of the four flying cables

Following the snubbing sequence the model responds essentially to four pre-stressed dead-ended cables. Consequently the math model for the snubbed dynamics consists of the conventional aerodynamic effects plus cable influence coefficients derived by assuming each cable to be a pre-stressed spring. The direction cosines, cable lengths, and cable tie-down geometry used for the conventional stability analysis are appropriate for the snubbed analysis, since the same cables are being used for snubbing. A schematic of the snub model is shown in Figure 8. The effects of the snubbed flying cables on both longitudinal and lateral/directional stability are modeled similar to the rear flying cables in the conventional analysis (see Sections 5.0 and 6.0 in reference 1). The force and moment contributions for each cable are calculated separately, summed and placed in the characteristic polynomial matrix. These calculations are made within subroutines LONG and LAT.

3.1 LONGITUDINAL AXIS

The general derivation for the longitudinal cable influence coefficients is presented in reference 1 and will not be repeated here. A 7 x 7 matrix with the form shown in Figure 8A is used to model each cable.
The matrix is reduced to a 3 x 3 in x, z, $\theta$ and put in the FXS array. The longitudinal stability is a 3 x 3 matrix in x, z, and $\theta$. The matrix no longer contains $\Delta T_F$ as an independent variable because the front cable constraint equation (no change in total front cable length) is not required in the snubbed condition. Each cable acts as an independent spring restraint.

3.2 LATERAL-DIRECTIONAL AXIS

The general derivation for the lateral-directional cable influence coefficients is also presented in reference 1. The equations describing each cable are set in a 8 x 8 matrix with the form shown in Figure 8B.

The matrix is reduced to a 3 x 3 matrix in Y, $\psi$, and $\phi$, and stored in the FXS array.

The lateral-directional stability matrix is a 3 x 3 matrix, structured exactly the same as the conventional stability matrix.
4.0 ADDITIONAL PROGRAM OPTIONS

Four additional options have been added to the Cable Mount Analysis Program. These are options to compute the numerators and denominators of the transfer function, the determination of the frequency response of any transfer function, the computation of wind-off characteristics and the computation of the wind tunnel model without cable effects (cableless model). The procedure for executing these options are discussed in this section.

4.1 TRANSFER FUNCTION OPTIONS

This option allows the computation of numerators and denominators. A detailed discussion of the procedure is presented in Section 4.1.1 and 4.1.2 for the longitudinal and lateral directional modes respectively.

4.1.1 LONGITUDINAL AXIS

The matrix shown in figure 4 is the complete longitudinal matrix. The size of the matrix to be evaluated determines the system that is being evaluated. KODE (8) is the parameter which sets the size of the matrix from which the roots are to be extracted. KODE (8) is set to either 4, 9, or 10. When KODE (8) is equal to 4, the system being evaluated is the basic inactive cable mount system as defined in reference 1. When KODE (8) is equal to 9, the open-loop roots of the active feedback system are extracted; and when KODE (8) is equal to 10, the closed-loop roots for the active feedback system are extracted.

KODE (14) and KODE (15) are the parameters which indicate to the program whether numerator or denominator roots are to be extracted. If KODE (14) is zero, the characteristic or denominator roots are extracted. If KODE (14) is non-zero, the program assumes that numerator roots are to be extracted. The program will then replace the column defined by KODE (15) by the column defined by KODE (14) in the matrix.
The basic no feedback system transfer function can be evaluated by setting KODE (8) to 4 and KODE (14) to 10. Setting KODE (15) from 1 to 4 will determine the numerator roots of the \( z/\Delta T_c \), \( \theta/\Delta T_c \), \( \Delta T_T/\Delta T_c \) and \( x/\Delta T_c \) transfer functions. Setting KODE (14) to zero will determine the denominator roots of these transfer functions. Thus the complete transfer function can be determined. Transfer function response to either elevator or gust input is possible by setting KODE (14) to 15 or 16 respectively.

The open loop zeros can be determined by setting KODE (8) to 9 and KODE (14) to 10. The variation of KODE (15) from 1 through 9 will determine the zeros for various output parameters. The open loop poles are determined by setting KODE (14) to 0.

In the closed loop numerator computation the forcing function can be either a test voltage input, \( E_{mo} \), an externally applied tension, \( \Delta T_1 \), a model elevator input, \( \delta_e \), or a vertical gust input, \( \sigma_g \). These inputs correspond to a KODE (14) of 11, 12, 15 or 16.

For example, if the closed loop numerator roots of the transfer function, \( \theta/E_{mo} \), are desired, KODE (14) is set to 11 and KODE (15) is set to 2. After the substitution of columns, the roots are extracted from the matrix whose size is set to 10 by KODE (8). By varying KODE (15) from 1 to 10, numerator roots of various output parameters can be obtained.

Since the model pitch rate, \( \dot{\theta} \), is an important parameter and this does not appear explicitly in the matrix, the program is set up to artificially generate the frequency response for this mode. This option is activated by setting KODE (15) to 13.

The transfer function of the cableless model, defined in Section 4.3, can also be determined. The numerators \( z/\delta_e \), \( \theta/\delta_e \) and \( x/\delta_e \) are determined
by setting KODE (8) = 3, KODE (14) = 14, and KODE (15) from 1 through 3. The
denominator roots are determined by setting KODE (14) to zero.

4.1.2 Lateral Directional Axis

KODE (9) is the parameter used in the lateral directional mode to set the
size of the matrix and define the system being evaluated. KODE (9) set to 3
defines the basic cable system without feedback. KODE (9) set to 9 defines
the open loop roots of the active feedback system and KODE (9) set to 10 defines
the closed loop roots of the active feedback system.

The numerator option is determined by KODE (16). KODE (16) set equal to
zero results in the extraction of characteristic roots. KODE (16) non-zero
results in the replacement of the column defined by KODE (17) with the column
defined by KODE (16) in the matrix of figure 5.

Specifically, the numerator characteristics of the basic cable system with-
out feedback are obtained by setting KODE (9) to 3 and KODE (16) to either 10, 14,
15, or 16 depending on the type of forcing function that is desired. These are
respectively a cable tension change, ∆T_c, a rudder input, δ_r, an aileron input,
δ_a, or a side gust, B_g. The dependent variable is determined by KODE (17) which
may vary from 1 through 3. The denominator roots are obtained by setting KODE (16)
to zero.

The open loop zeros of the block diagram shown in figure 5 is determined
by setting KODE (9) to 9, KODE (16) to 10 and KODE (17) from 1 through 9. The
denominator or open loop poles are determined by setting KODE (16) to zero.

The closed loop numerator for the active cable system is determined by
setting KODE (9) to 10. The forcing function is defined by KODE (16). This
code can be 11, 12, 14, 15, or 16. They correspond to a test voltage, E_w0, test
tension, ∆T_i, rudder input, δ_r, aileron input, δ_a, or a side wind gust, B_g.
4.2 **Frequency Response Option**

The frequency response option will compute the complete transfer function according to Section 4.1; and then evaluates for the computed transfer function over a range of frequencies, the amplitude ratio in actual value, db's, and the phase angle. The option will compute up to 60 points over a 3 decade bandwidth with a maximum of 20 ints per decade.

This option will also compute the steady state value of the transfer function to a step input of the forcing function if this value exists.

The frequency response option is activated by setting \texttt{KODE (3)} to +2. Since a complete transfer function must be generated prior to developing the frequency response data, \texttt{KODE (14)} and \texttt{KODE (15)} or \texttt{KODE (16)} and \texttt{KODE (17)} must be set to non-zero values to define the desired transfer function. Two additional parameters, \texttt{KODE (18)} and \texttt{KODE (19)}, must be set to define the frequency range and number of points to be computed. \texttt{KODE (18)} specifies the order of the lowest frequency to be computed, e.g., \texttt{KODE (18) = -1} corresponds to 1 rps and a "+1" corresponds to 10 rps. \texttt{KODE (19)} set to 60 means sixty points are computed for the three decade bandwidth of the frequency response.

The frequency response option is initiated in subroutines LONG and LAT for the longitudinal and lateral directional modes respectively. The program, on sensing \texttt{KODE (3)} equal to 2, will effectively cycle through subroutines LONG or LAT twice, first to compute the numerator and then again to compute the denominator roots.

The information is then passed to subroutine \texttt{FREQ} where the frequency response data is generated with the aid of subroutine \texttt{ANP}.
4.3 Wind-Off Characteristics

This option is used to compute the system response without the aerodynamic effects. The dynamic characteristics reflect the system feedback, and equivalent spring and damping effects.

In this option, the normal trim operation technique is circumvented. Instead, the vehicle attitude is set to zero and the forward cable tension is defined to balance out the rear cable tension.

The program will execute this option if the velocity (AERO (49)) and the MACH number (AERO (48)) are set to zero.

4.4 Cableless Model Characteristics

This option allows the computation of the airframe characteristic roots without the cable effects. The program initially trims the vehicle assuming the cables are attached to the vehicle. After defining the trim attitude, the cable influence coefficients are set to zero.

This option defines the characteristics of a model in the wind tunnel. The equations are different from the conventional airframe analysis equations. The differences are in the relation of angle of attack to model pitch attitude (see equation 5.3-2 of ref 1) and the missing thrust terms.

Prior to extracting roots from the matrix in the longitudinal mode, the X column is shifted to the left one column eliminating the $\Delta T_F$ column in figure 5. Thus the cableless model option requires a $KODE (8)$ of 3 reflecting a $3 \times 3$ matrix size.

The lateral directional mode does not require this extra step of column manipulation and $KODE (9)$ should be set to 3.

The program will execute this option only if $KODE (13)$ is set to -1.
5.0 INPUT DATA

The input format and the description of the elements in the input arrays will be described in this section. This discussion is meant to supersede the description contained in Section 11.0 of Reference 1.

The format for the input data is most easily explained by reproducing the "READ" statements as they appear in the program.

\[
\text{READ (IR, 150) (TITLE (I), I = 1, 20)} \quad (1)
\]

\[
150 \text{ FORMAT (20A4)}
\]

\[
\text{READ (IR, 200) (KODE (I), I = 1, 24)} \quad (2)
\]

\[
200 \text{ FORMAT (24I3)}
\]

Then either 3a or 3b: the value of KODE (7) will determine which "READ" statement will be used.

\[
\text{READ (IR,100) (AERO (I), I = 1,36)} \quad (3a)
\]

\[
100 \text{ FORMAT (6E12.5)}
\]

\[
\text{CALL TABIN (1, 36) (See Appendix A, Ref. 1)} \quad (3b)
\]

Following either (3a) or (3b) the sequence of "READ" statements continues:

\[
\text{READ (IR, 100) (AERO (I), I = 44,59)} \quad (4)
\]

\[
\text{READ (IR, 100) (AERO (I), I = 66, 130)} \quad (5)
\]

Now if KODE (13) is greater than zero the following "READ" statement is encountered. If KODE (13) is less than or equal to zero this "READ" statement is skipped.

\[
\text{READ (IR, 100)(AERO (I), I = 131, 160)} \quad (6)
\]

Now if KODE (12) equals one, the following table read statement is encountered. If KODE (12) equals zero this statement is skipped.

\[
\text{CALL TABIN (1, 2) (See Appendix A, Ref 1)} \quad (7)
\]

This completes the initial sequence of input data. After completion of the first run the following statements initialize another run.
READ (IR, 150) (TITLE (I), I = 1, 20)  (8)
READ (IR, 200) (KODE (I), I = 1, 24)  (9)
READ (IR, 350) K, VALUE  (10)

K = element in "AERO" array to be changed
VALUE = new value of the element

If I = 1 this "READ" statement is repeated
If I = 0 the program begins computation

All succeeding cases follow the same input format starting with

statement (8).

A general description of the input arrays follows:

<table>
<thead>
<tr>
<th>ARRAY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>Alpha-numeric array containing title for each run.</td>
</tr>
<tr>
<td>KODE</td>
<td>Array specifying program options to be exercised.</td>
</tr>
<tr>
<td>AERO</td>
<td>Array containing all the input data pertaining to the model, the mount system, tunnel conditions. etc.</td>
</tr>
</tbody>
</table>

A description of each element in the "KODE" and "AERO" arrays follows.
<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE (1)</td>
<td>-</td>
<td>Run number.</td>
</tr>
<tr>
<td>CODE (2)</td>
<td>-1</td>
<td>Calculate longitudinal stability.</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Calculate lateral/directional stability.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Calculate both longitudinal and lateral/directional stability.</td>
</tr>
<tr>
<td>CODE (3)</td>
<td>0</td>
<td>No root locus or frequency response.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Do root locus.</td>
</tr>
<tr>
<td></td>
<td>+2</td>
<td>Do frequency response.</td>
</tr>
<tr>
<td>CODE (4)</td>
<td>0</td>
<td>Element in &quot;AERO&quot; array to be varied for root locus.</td>
</tr>
<tr>
<td>CODE (5)</td>
<td>0</td>
<td>Basic printout.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Basic printout plus various test parameters.</td>
</tr>
<tr>
<td>CODE (6)</td>
<td>+1</td>
<td>Front cable vertical-rear cable horizontal.</td>
</tr>
<tr>
<td></td>
<td>+2</td>
<td>Front cable horizontal-rear cable vertical.</td>
</tr>
<tr>
<td></td>
<td>+3</td>
<td>Front and rear cable vertical.</td>
</tr>
<tr>
<td></td>
<td>+4</td>
<td>Front and rear cable horizontal</td>
</tr>
<tr>
<td>CODE (7)</td>
<td>0</td>
<td>Aero data to be input at specific mach number.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Aero data to be input in the form of tables.</td>
</tr>
<tr>
<td>NAME</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CODE (8)</td>
<td>+3</td>
<td>Longitudinal matrix - Cableless Model (see Section 4.3)</td>
</tr>
<tr>
<td></td>
<td>+4</td>
<td>Longitudinal matrix - no stability augmentation.</td>
</tr>
<tr>
<td></td>
<td>+5</td>
<td>Longitudinal matrix - internal stability augmentation (see Section 9.0, Reference 1.)</td>
</tr>
<tr>
<td></td>
<td>+9</td>
<td>Longitudinal matrix - Open loop response of Active Cable Mount System (see Section 2.0, 4.1.1)</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Longitudinal matrix - Close loop response of Active Cable Mount System (see Section 2.0, 4.1.1)</td>
</tr>
<tr>
<td>CODE (9)</td>
<td>+3</td>
<td>Lateral/directional matrix - no stability augmentation or cableless model.</td>
</tr>
<tr>
<td></td>
<td>+4</td>
<td>Lateral/directional matrix - internal yaw stability augmentation, (see Section 9.0, Reference 1.)</td>
</tr>
<tr>
<td></td>
<td>+5</td>
<td>Lateral/directional matrix - internal roll and yaw stability augmentation, (see Section 9.0, Reference 1.)</td>
</tr>
<tr>
<td></td>
<td>+9</td>
<td>Lateral/directional matrix - open loop response of Active Cable Mount System (see Section 2.0, 4.1.2)</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Lateral/directional matrix - Close loop response of Active Cable Mount System (See Section 2.0 4.1.2)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>KODE (10)</td>
<td>0</td>
<td>No snubbers.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Analyze conventional snubbers in un-snubbed condition - see Section 8.1, Reference 1.</td>
</tr>
<tr>
<td></td>
<td>+2</td>
<td>Analyze conventional snubbers in snubbed condition - See Section 8.2, Reference 1.</td>
</tr>
<tr>
<td></td>
<td>+3</td>
<td>Analyze flying cable snubber system.</td>
</tr>
<tr>
<td>KODE (11)</td>
<td>0</td>
<td>No anti-lift cable.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Anti-lift cable in.</td>
</tr>
<tr>
<td>KODE (12)</td>
<td>0</td>
<td>No unsnubbed snubber data input.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Unsnubbed snubber data will be read in.</td>
</tr>
<tr>
<td>KODE (13)</td>
<td>-1</td>
<td>Cableless Airframe Characteristics. (See Section 4.3)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>No active cable stability augmentation.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Active cable stability augmentation in. (See Section 2.0)</td>
</tr>
<tr>
<td>KODE (14)</td>
<td>0</td>
<td>Longitudinal system - compute denominator characteristics only.</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Longitudinal system - numerator and/or frequency characteristics of inactive cable mount system for cable tension input, ( \Delta T_c ). (See Section 4.1.1)</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Longitudinal System - numerator and/or frequency characteristics of active cable mount system open loop for cable tension input, ( \Delta T_c ). (See Section 4.1.1)</td>
</tr>
<tr>
<td>NAME</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>+11</td>
<td></td>
<td>Longitudinal System - numerator and/or frequency characteristics of active cable mount system close loop response for test voltage input $E_{mo}$. (See Section 4.1.1)</td>
</tr>
<tr>
<td>+12</td>
<td></td>
<td>Longitudinal System - numerator and/or frequency characteristics of active cable mount system close loop response for externally applied tension, $\Delta T_i$. (See Section 4.1.1)</td>
</tr>
<tr>
<td>+15</td>
<td></td>
<td>Longitudinal system - numerator and/or frequency characteristics for pitch control response ($\delta_e$)</td>
</tr>
<tr>
<td>+16</td>
<td></td>
<td>Longitudinal system - numerator and/or frequency characteristics for gust response ($\alpha_0$).</td>
</tr>
<tr>
<td>KODE (15)</td>
<td></td>
<td>Longitudinal system - column number of output variable for which numerator and/or frequency data is desired. KODE (15) is set equal to 13 for model pitch rate response. This value must be equal or less than KODE (8). (See Section 4.1.1)</td>
</tr>
<tr>
<td>KODE (16)</td>
<td>0</td>
<td>Lateral/directional system - compute denominator characteristics only</td>
</tr>
<tr>
<td>NAME</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Lateral/directional system - numerator and/or frequency characteristics of inactive cable mount system for tension input ΔTc. (See Section 4.1.2).</td>
</tr>
<tr>
<td></td>
<td>+10</td>
<td>Lateral/directional system - numerator and/or frequency characteristics of active cable mount system open loop for tension input, ΔTc. (See Section 4.1.2)</td>
</tr>
<tr>
<td></td>
<td>+11</td>
<td>Lateral/directional system - numerator and/or frequency characteristics of active cable mount system close loop for test voltage input ( E_{\text{mo}} ). (See Section 4.1.2)</td>
</tr>
<tr>
<td></td>
<td>+12</td>
<td>Lateral/directional system - numerator and/or frequency characteristics of active cable mount system close loop response for externally applied tension, ( (\Delta T)/) (See Section 4.1.2)</td>
</tr>
<tr>
<td></td>
<td>+14</td>
<td>Lateral/directional system - numerator and/or frequency characteristics for yaw control response (( \delta r )).</td>
</tr>
<tr>
<td></td>
<td>+15</td>
<td>Lateral/directional system - numerator and or frequency characteristics for roll control response (( \delta a )).</td>
</tr>
<tr>
<td>NAME</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>+16</td>
<td>Lateral/directional system - numerator and/or frequency characteristics for gust response (\dot{\alpha}_g).</td>
</tr>
<tr>
<td>KODE (17)</td>
<td></td>
<td>Lateral/directional system - column number of independent variable for which numerator and/or frequency data is desired.</td>
</tr>
<tr>
<td>KODE (18)</td>
<td></td>
<td>Order of lowest frequency (RPS) for frequency response data.</td>
</tr>
<tr>
<td>KODE (19)</td>
<td></td>
<td>Number of data points in frequency response (Max of 60.)</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
</tr>
<tr>
<td>----------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>AERO (1)</td>
<td>N.D.</td>
<td>CDU</td>
</tr>
<tr>
<td>AERO (2)</td>
<td>N.D.</td>
<td>CLU</td>
</tr>
<tr>
<td>AERO (3)</td>
<td>N.D.</td>
<td>CMU</td>
</tr>
<tr>
<td>AERO (4)</td>
<td>1/rad</td>
<td>CDA</td>
</tr>
<tr>
<td>AERO (5)</td>
<td>1/rad</td>
<td>CLA</td>
</tr>
<tr>
<td>AERO (6)</td>
<td>1/rad</td>
<td>CMA</td>
</tr>
<tr>
<td>AERO (7)</td>
<td>N.D.</td>
<td>CDQ</td>
</tr>
<tr>
<td>AERO (8)</td>
<td>N.D.</td>
<td>CLQ</td>
</tr>
<tr>
<td>AERO (9)</td>
<td>N.D.</td>
<td>CMQ</td>
</tr>
<tr>
<td>AERO (10)</td>
<td>N.D.</td>
<td>CDO</td>
</tr>
<tr>
<td>AERO (11)</td>
<td>N.D.</td>
<td>CLO</td>
</tr>
<tr>
<td>AERO (12)</td>
<td>N.D.</td>
<td>CMO</td>
</tr>
<tr>
<td>AERO (13)</td>
<td>1/rad</td>
<td>CDDE</td>
</tr>
<tr>
<td>AERO (14)</td>
<td>1/rad</td>
<td>CLDE</td>
</tr>
<tr>
<td>AERO (15)</td>
<td>1/rad</td>
<td>CMDE</td>
</tr>
<tr>
<td>AERO (16)</td>
<td>N.D.</td>
<td>CDAD</td>
</tr>
<tr>
<td>AERO (17)</td>
<td>N.D.</td>
<td>CLAD</td>
</tr>
<tr>
<td>AERO (18)</td>
<td>N.D.</td>
<td>CMAD</td>
</tr>
<tr>
<td>AERO (19)</td>
<td>1/rad</td>
<td>CYB</td>
</tr>
<tr>
<td>AERO (20)</td>
<td>1/rad</td>
<td>CLB</td>
</tr>
<tr>
<td>AERO (21)</td>
<td>1/rad</td>
<td>CNB</td>
</tr>
<tr>
<td>AERO (22)</td>
<td>N.D.</td>
<td>CYP</td>
</tr>
<tr>
<td>AERO (23)</td>
<td>N.D.</td>
<td>CLP</td>
</tr>
<tr>
<td>AERO (24)</td>
<td>N.D.</td>
<td>CNP</td>
</tr>
<tr>
<td>AERO (25)</td>
<td>N.D.</td>
<td>CYR</td>
</tr>
<tr>
<td>AERO (26)</td>
<td>N.D.</td>
<td>CLR</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>AERO (27)</td>
<td>N.D.</td>
<td>CNR</td>
</tr>
<tr>
<td>AERO (28)</td>
<td>1/rad</td>
<td>CYDR</td>
</tr>
<tr>
<td>AERO (29)</td>
<td>1/rad</td>
<td>CLDR</td>
</tr>
<tr>
<td>AERO (30)</td>
<td>1/rad</td>
<td>CNDR</td>
</tr>
<tr>
<td>AERO (31)</td>
<td>1/rad</td>
<td>CYDA</td>
</tr>
<tr>
<td>AERO (32)</td>
<td>1/rad</td>
<td>CLDA</td>
</tr>
<tr>
<td>AERO (33)</td>
<td>1/rad</td>
<td>CNDA</td>
</tr>
<tr>
<td>AERO (34)</td>
<td>1/rad</td>
<td>CYDS</td>
</tr>
<tr>
<td>AERO (35)</td>
<td>1/rad</td>
<td>CLDS</td>
</tr>
<tr>
<td>AERO (36)</td>
<td>1/rad</td>
<td>CNDS</td>
</tr>
<tr>
<td>AERO (44)</td>
<td>in</td>
<td>XREF*</td>
</tr>
<tr>
<td>AERO (45)</td>
<td>in</td>
<td>ZREF</td>
</tr>
<tr>
<td>AERO (46)</td>
<td>in</td>
<td>XCG</td>
</tr>
<tr>
<td>AERO (47)</td>
<td>in</td>
<td>ZCG</td>
</tr>
<tr>
<td>AERO (48)</td>
<td></td>
<td>AMACH</td>
</tr>
<tr>
<td>AERO (49)</td>
<td>ft/sec</td>
<td>VO</td>
</tr>
<tr>
<td>AERO (50)</td>
<td>slugs</td>
<td>AM</td>
</tr>
<tr>
<td>AERO (51)</td>
<td>slug/ft^3</td>
<td>RHO</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>AERO (52)</td>
<td>lbs</td>
<td>WT</td>
</tr>
<tr>
<td>AERO (53)</td>
<td>ft</td>
<td>B</td>
</tr>
<tr>
<td>AERO (54)</td>
<td>ft</td>
<td>CBAR</td>
</tr>
<tr>
<td>AERO (55)</td>
<td>ft²</td>
<td>SW</td>
</tr>
<tr>
<td>AERO (56)</td>
<td>slug·ft²</td>
<td>XIXZ</td>
</tr>
<tr>
<td>AERO (57)</td>
<td>slug·ft²</td>
<td>XIXX</td>
</tr>
<tr>
<td>AERO (58)</td>
<td>slug·ft²</td>
<td>YIYY</td>
</tr>
<tr>
<td>AERO (59)</td>
<td>slug·ft²</td>
<td>ZIZZ</td>
</tr>
<tr>
<td>AERO (66)</td>
<td>in</td>
<td>WLUF</td>
</tr>
<tr>
<td>AERO (67)</td>
<td>in</td>
<td>WLLL</td>
</tr>
<tr>
<td>AERO (68)</td>
<td>in</td>
<td>WLUR</td>
</tr>
<tr>
<td>AERO (69)</td>
<td>in</td>
<td>WLLR</td>
</tr>
<tr>
<td>AERO (70)</td>
<td>in</td>
<td>WLHF</td>
</tr>
<tr>
<td>AERO (71)</td>
<td>in</td>
<td>WLHR</td>
</tr>
<tr>
<td>AERO (72)</td>
<td>in</td>
<td>STAF</td>
</tr>
<tr>
<td>AERO (73)</td>
<td>in</td>
<td>STAR</td>
</tr>
<tr>
<td>AERO (74)</td>
<td>in</td>
<td>BLHF</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>AERO (75)</td>
<td>in</td>
<td>BLHR</td>
</tr>
<tr>
<td>AERO (76)</td>
<td>in</td>
<td>WLCR</td>
</tr>
<tr>
<td>AERO (77)</td>
<td>in</td>
<td>STACR</td>
</tr>
<tr>
<td>AERO (78)</td>
<td>in</td>
<td>BLCR</td>
</tr>
<tr>
<td>AERO (79)</td>
<td>in</td>
<td>EF**</td>
</tr>
<tr>
<td>AERO (80)</td>
<td>in</td>
<td>ER</td>
</tr>
<tr>
<td>AERO (81)</td>
<td>in</td>
<td>AF</td>
</tr>
<tr>
<td>AERO (82)</td>
<td>in</td>
<td>AR</td>
</tr>
<tr>
<td>AERO (83)</td>
<td>in</td>
<td>HUCF</td>
</tr>
<tr>
<td>AERO (84)</td>
<td>in</td>
<td>HLCF</td>
</tr>
<tr>
<td>AERO (85)</td>
<td>in</td>
<td>HUCR</td>
</tr>
<tr>
<td>AERO (86)</td>
<td>in</td>
<td>HLCR</td>
</tr>
<tr>
<td>AERO (87)</td>
<td>in</td>
<td>DCF</td>
</tr>
<tr>
<td>AERO (88)</td>
<td>in</td>
<td>DCR</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>AERO (89)</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>AERO (90)</td>
<td>in</td>
<td>RVF</td>
</tr>
<tr>
<td>AERO (91)</td>
<td>in</td>
<td>RHF</td>
</tr>
<tr>
<td>AERO (92)</td>
<td>in</td>
<td>RVR</td>
</tr>
<tr>
<td>AERO (93)</td>
<td>in</td>
<td>RHR</td>
</tr>
<tr>
<td>AERO (94)</td>
<td>lbs</td>
<td>TRO</td>
</tr>
<tr>
<td>AERO (95)</td>
<td>lbs/in</td>
<td>AKR</td>
</tr>
<tr>
<td>AERO (96)</td>
<td>ft lbs/rad</td>
<td>COU</td>
</tr>
<tr>
<td>AERO (97)</td>
<td>in</td>
<td>STLTT</td>
</tr>
<tr>
<td>AERO (98)</td>
<td>in</td>
<td>WLLTT</td>
</tr>
<tr>
<td>AERO (99)</td>
<td>lbs</td>
<td>TLFTO</td>
</tr>
<tr>
<td>AERO (100)</td>
<td>lbs/in</td>
<td>AKLFT</td>
</tr>
<tr>
<td>AERO (101)</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>AERO (102)</td>
<td>in</td>
<td>ALTX*</td>
</tr>
<tr>
<td>AERO (103)</td>
<td>in</td>
<td>ALTZ</td>
</tr>
<tr>
<td>(1) AERO (104)</td>
<td>ft lbs/rad/sec</td>
<td>CMP</td>
</tr>
<tr>
<td>AERO (105)</td>
<td>in</td>
<td>SNUX***</td>
</tr>
<tr>
<td>AERO (106)</td>
<td>in</td>
<td>SNUY</td>
</tr>
<tr>
<td>AERO (107)</td>
<td>in</td>
<td>SNUZ</td>
</tr>
</tbody>
</table>

(1) AERO (104) through AERO (122) refer to conventional snubbers except where noted.
<table>
<thead>
<tr>
<th>NAME</th>
<th>UNITS</th>
<th>LABEL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AERO (108)</td>
<td>in</td>
<td>SNLX</td>
<td>Distance along X body axis from model lower snubber attachment point to the equation reference center</td>
</tr>
<tr>
<td>AERO (109)</td>
<td>in</td>
<td>SNIY</td>
<td>Distance along Y body axis from model lower snubber attachment point to the equation reference center</td>
</tr>
<tr>
<td>AERO (110)</td>
<td>in</td>
<td>SNLZ</td>
<td>Distance along Z body axis from model lower snubber attachment point to the equation reference center</td>
</tr>
<tr>
<td>AERO (111)</td>
<td>in</td>
<td>SNUST</td>
<td>Station - upper snubber tie-down point</td>
</tr>
<tr>
<td>AERO (112)</td>
<td>in</td>
<td>SNUML</td>
<td>Water line - upper snubber tie-down point</td>
</tr>
<tr>
<td>AERO (113)</td>
<td>in</td>
<td>SNUBL</td>
<td>Butt line - upper snubber tie-down point</td>
</tr>
<tr>
<td>AERO (114)</td>
<td>in</td>
<td>SNLST</td>
<td>Station - lower snubber tie-down point</td>
</tr>
<tr>
<td>AERO (115)</td>
<td>in</td>
<td>SNLWL</td>
<td>Water line - lower snubber tie-down point</td>
</tr>
<tr>
<td>AERO (116)</td>
<td>in</td>
<td>SNLBL</td>
<td>Butt line - lower snubber tie-down point</td>
</tr>
<tr>
<td>AERO (117)</td>
<td>lbs</td>
<td>TUSNO</td>
<td>Upper snubber, snubbed tension or flying cable snubber rear cable tension. Lower snubber, snubber tension</td>
</tr>
<tr>
<td>AERO (118)</td>
<td>lbs</td>
<td>TLSNO</td>
<td></td>
</tr>
<tr>
<td>AERO (119)</td>
<td>lbs/in</td>
<td>AKSNL</td>
<td>Upper snubber, snubbed spring constant</td>
</tr>
<tr>
<td>AERO (120)</td>
<td>lbs/in</td>
<td>AKSNL</td>
<td>Lower snuber, snubbed spring constant flying cable snubber rear cable spring constant</td>
</tr>
<tr>
<td>AERO (121)</td>
<td>lbs/in/sec</td>
<td>ADSNU</td>
<td>Upper snubber, snubbed damping constant or flying cable snubber front cable spring constant</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>AERO (122)</td>
<td>lbs/in/sec</td>
<td>ADSML</td>
<td>Lower snubber, snubbed damping constant.</td>
</tr>
<tr>
<td>AERO (123)</td>
<td>rad/rad/sec</td>
<td>AKSY</td>
<td>Feedback gain - yaw rate to rudder.</td>
</tr>
<tr>
<td>AERO (124)</td>
<td>rad/rad/sec</td>
<td>AKPHI</td>
<td>Feedback gain - roll rate to aileron.</td>
</tr>
<tr>
<td>AERO (125)</td>
<td>rad/rad/sec</td>
<td>AKTHE</td>
<td>Feedback gain - pitch rate to elevator.</td>
</tr>
<tr>
<td>AERO (126)</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO (127)</td>
<td>sec</td>
<td>TPSY</td>
<td>Time constant for lag on yaw rate feedback.</td>
</tr>
<tr>
<td>AERO (128)</td>
<td>sec</td>
<td>T2PHI</td>
<td>Time constant for lag on roll rate feedback.</td>
</tr>
<tr>
<td>AERO (129)</td>
<td>sec</td>
<td>T3THE</td>
<td>Time constant for lag on pitch rate feedback.</td>
</tr>
<tr>
<td>AERO (130)</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO (131)</td>
<td>in-lbs/amp</td>
<td>AKSET **</td>
<td>Motor torque constant ($K_T$)</td>
</tr>
<tr>
<td>AERO (132)</td>
<td>volts/rad/sec</td>
<td>AKSBV</td>
<td>Motor velocity constant ($K_V$)</td>
</tr>
<tr>
<td>AERO (133)</td>
<td>in-lbs/sec^2</td>
<td>AJASM</td>
<td>Motor inertia ($J_M$)</td>
</tr>
<tr>
<td>AERO (134)</td>
<td>ohms</td>
<td>RSBA</td>
<td>Motor armature resistance ($R_a$)</td>
</tr>
<tr>
<td>AERO (135)</td>
<td>henry</td>
<td>ELSBA</td>
<td>Motor armature inductance ($L_a$)</td>
</tr>
<tr>
<td>AERO (136)</td>
<td>in</td>
<td>RSBD</td>
<td>Radius of motor pulley ($r_d$)</td>
</tr>
<tr>
<td>AERO (137)</td>
<td>volts/rad/sec</td>
<td>AKTHD</td>
<td>Pulley rotation rate feedback ($K_{\phi_m}$)</td>
</tr>
<tr>
<td>AERO (138)</td>
<td>volts/rad</td>
<td>AKTH</td>
<td>Pulley rotation displacement feedback ($K_{\phi_m}$)</td>
</tr>
<tr>
<td>AERO (139)</td>
<td>in-lbs/rad/sec</td>
<td>GDMF</td>
<td>Pulley friction ($G$)</td>
</tr>
<tr>
<td>AERO (140)</td>
<td>volts/rad/sec</td>
<td>AKQ</td>
<td>Model pitch rate feedback ($K_q$)</td>
</tr>
<tr>
<td>NAME</td>
<td>UNITS</td>
<td>LABEL</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>AERO (142)</td>
<td>volts/rad/sec</td>
<td>AKPSD</td>
<td>Model yaw rate feedback ($K_Y$)</td>
</tr>
<tr>
<td>AERO (143)</td>
<td>volts/rad</td>
<td>AKY</td>
<td>Pulley rotation displacement feedback ($K_{ym}$)</td>
</tr>
<tr>
<td>AERO (144)</td>
<td>volts/rad/sec</td>
<td>AKYD</td>
<td>Pulley rotation rate feedback ($K_{ym}$)</td>
</tr>
<tr>
<td>AERO (145) to AERO (160)</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*See Figure 9 for pictorial representation of various reference center.

**See Figure 10 for pictorial representation of pulley geometry.

***See Figure 11 for pictorial representation of conventional snubber cable geometry.

****See Figures 2 and 3 for block diagram representations of the active cable control logic. (See appendix B for derivation)
If the aerodynamic data and/or snubber data are to be read in table format, the following discussion applies.

The first 36 tables contain the aerodynamic derivatives in stability axis versus mach number. The order is the same as AERO (1) through AERO (36). The table input format is shown in Appendix A of Reference 1. This data is read in under TABINV.

The unsnubbed snubber data consists of two tables of input. The first table contains cable tension (lbs) versus dynamic pressure (psf) and linear distance (in) between model tie-down point and tunnel side wall. The second table contains cable angle (rad) versus dynamic pressure (psf) and linear distance (in) between model tie-down point and the tunnel side wall. The tensions and angles mentioned here are described in detail in Section 8.0 of Reference 1.
Reference


Appendix A

A Discussion of the Differences in Cable Attachment Points Between the Inactive and Active Cable Mount System

There exists a basic difference in the cable mount system analyzed in the original program and the present active cable system. In the original system, the front cable is attached to hard points on the tunnel wall. The cable wraps around pulleys which are fixed to the model. This cable is assumed to be fixed in length. The rear cable is similarly wrapped around pulleys fixed to the model. There is a spring which is connected in series with the rear cable which allows for play in the system. This system is pictorially represented in figure A-1.

In the present "active cable system," the front cable is attached to hard point on the model. The cable wraps around pulleys fixed to the tunnel. One of the pulleys is connected to a torque motor. The rear cable is similarly routed around pulleys fixed to the tunnel and tied to hard points on the model. The spring on the rear cable is still assumed. This system is pictorially represented in fig. A2.

The present program is capable of handling both cases. The radius of the pulleys fixed to the model must be made very small to reflect the hard attachment point in the new system, i.e. Aero (90) thru (93) inclusively must be set to .01. The pulley radius mounted to the torque motor is important in the new system and is defined by Aero (136). When the program reverts back to the original system, Aero (90) and Aero (93) is significant, and Aero (136) is ignored.

The program is capable of this dual application because of the method utilized in the analysis of the cable forces. The front and rear cables, which are respectively continuous cables, are analyzed as four individual branches. Each branch represents the cable between the model and the tunnel. These branches are numbered in both figures A1 and A2. The force components on the model contributed by each branch of cable is a function of three factors, the tension
in that branch of cable, the orientation of the cable and the exact point of application of the force on the model. The impact of having pulleys fixed to the model is simply to alter the point of application. By reducing the pulley radius, the point of application is analogous to a fixed point on the model.

The other consideration is friction effects of pulleys. There are two different friction definitions, Aero (96) and Aero (104) define the friction in pulleys for the inactive cable system, whereas Aero (139) represents friction effects of the Active Cable System.
**fig A.1** Original Passive Cable System Schematic

**fig A.2** Active 2-Cable Mount System Schematic
APPENDIX B

Derivation of Motor Equations and Cable Tension

The net output torque from the motor is proportional to the current to the motor. The current is related to the voltage and back EMF as shown by equation 1. A list of symbol definition is given on page iii.

\[ Q_o = K_T I_a = K_T \left[ \frac{E_{TOT} - K_V s \theta_m}{R_a + sL_a} \right] \]  

(1)

For two motors in parallel, the output torque is doubled:

\[ Q'_o = 2Q_o \]

The load torque on the motor is due to the total change in cable tension, \( \Delta T_{tot} \) and the friction in the system. The coulomb and viscous friction can be written as proportional to the pulley rate. (See ref 2.)

\[ Q_L = \Delta T_{TOT} \omega_d + Gs\theta_m \]  

(2)

The net torque, output minus load, will cause the motor to rotate.

\[ Q'_o - Q_L = J_M \ddot{\theta}_m = J_M s^2 \theta_m \]  

(3)

Substituting equations (1) and (2) into equation (3) for \( Q'_o \) and \( Q_L \) respectively, the total change in cable tension, \( \Delta T_{tot} \), can be determined.

\[ \Delta T_{tot} = \frac{1}{r_d} \left\{ J_M s^2 + G + \frac{2K_T K_V s}{R_a + sL_a} \theta_m \right\} \left( \frac{2K_T E_{TOT}}{R_a + sL_a} \right) \]  

(4)

\( \Delta T_{tot} \) is positive when the cable is in tension.

Looking at the larger picture shown in figure B-1, the total change in cable tension can be split into two increments \( \delta T_1 \) and \( \delta T_2 \). Writing the equation of motion of the cable

fig B-1
\[ T_0 - T_1 = (T_0 + \delta T_2) = ma \]  \hspace{1cm} (5)

For \( ma = 0 \)

\[ -\delta T_1 - \delta T_2 = 0 \]  \hspace{1cm} (6)

and \( \delta T_2 = -\delta T_1 \)  \hspace{1cm} (7)

This states that if the mass times acceleration of the cable is small and can be neglected, the increase in cable tension on one side of the torque motor is just equal to the decrease cable tension on the other side of the torque motor. This result is ideally suited for the perturbation analysis since the program actually considers the continuous cable in figure B-1 as two separate elements as indicated by the dashed lines. With the change in cable tension having equal magnitude along each element, the mechanization is simplified.

Figure B-2 shows the relation of the change in cable tension on one side of the torque motor, \( \delta T \), to the total change in cable tension \( \Delta T_{TOT} \). Thus

\[ \delta T_2 - \delta T_1 = \Delta T_{TOT} \]  \hspace{1cm} (8)

Substituting results from equation 7 into equation 8

\[ \delta T_2 = \frac{\Delta T_{TOT}}{2} \]  \hspace{1cm} (9)

Replacing \( \Delta T_{TOT} \) in equation (4) with equation (9), \( \delta T_2 \) is determined.

The \( \delta T_2 \) corresponds to \( \Delta T_{rb} \) in figures 4 and 5

\[ \delta T_2 = \frac{1}{2r_d} \left\{ \left[ J_v s^2 + G_s + \frac{2T_k K_d}{R_a + sL_a} \right] \theta_m - \frac{2K_mE_{TOT}}{R_a + sL_a} \right\} \]  \hspace{1cm} (10)
APPENDIX C

PROGRAM LISTINGS
<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>C THIS IS THE ACTIVE TVC CABLE MOUNT SYSTEM ANALYSES PROGRAM</td>
<td>CBLCC010</td>
</tr>
<tr>
<td>C DEVELOPED JULY '74 TO MAY '75</td>
<td>CBLCC020</td>
</tr>
<tr>
<td></td>
<td>CBLCC030</td>
</tr>
<tr>
<td>COMMON/INUT/IN, IF</td>
<td>CBLCC040</td>
</tr>
<tr>
<td>COMMON/DAT/AERO(175), AEROD(50), KCODE(26), ULL</td>
<td>CBLCC050</td>
</tr>
<tr>
<td>COMMON/SNTR/SNU(3, 3), SN(32), THUSN, THLSN, SNUC(3, 7)</td>
<td>CBLCC060</td>
</tr>
<tr>
<td>COMMON 2Z2(22)</td>
<td>CBLCC070</td>
</tr>
<tr>
<td>COMMON/TA27(22)</td>
<td>CBLCC080</td>
</tr>
<tr>
<td>COMMON/TV/SUM(17, 18)</td>
<td>CBLCC090</td>
</tr>
<tr>
<td>COMMON/NAME/NAME(16), NAME1(16)</td>
<td>CBLCC100</td>
</tr>
<tr>
<td>DIMENSION TITLE(25), SAVE(55), SAVE1(150), IKH1(160)</td>
<td>CBLCC110</td>
</tr>
</tbody>
</table>

**EQUIVALENCE**

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AERO(1), C01), (AERO(2), CLU), (AER0(3), CMJ)</td>
<td>CBLCC120</td>
</tr>
<tr>
<td>(AERO(4), C0A), (AERO(5), C0A), (AERO(6), CMJ)</td>
<td>CBLCC130</td>
</tr>
<tr>
<td>(AERO(7), C0D), (AERO(8), C0D), (AERO(9), CMJ)</td>
<td>CBLCC140</td>
</tr>
<tr>
<td>(AERO(10), C0C), (AERO(11), C0C), (AERO(12), CMJ)</td>
<td>CBLCC150</td>
</tr>
<tr>
<td>(AERO(13), C0F), (AERO(14), C0F), (AERO(15), CMJ)</td>
<td>CBLCC160</td>
</tr>
<tr>
<td>(AERO(16), C0A), (AERO(17), C0A), (AERO(18), CMJ)</td>
<td>CBLCC170</td>
</tr>
<tr>
<td>(AERO(19), CY3), (AERO(20), CY3), (AERO(21), CMJ)</td>
<td>CBLCC180</td>
</tr>
<tr>
<td>(AERO(22), CY9), (AERO(23), CY9), (AERO(24), CMJ)</td>
<td>CBLCC190</td>
</tr>
<tr>
<td>(AERO(25), CYF), (AERO(26), CYF), (AERO(27), CMJ)</td>
<td>CBLCC200</td>
</tr>
<tr>
<td>(AERO(28), CYG), (AERO(29), CYG), (AERO(30), CMJ)</td>
<td>CBLCC210</td>
</tr>
<tr>
<td>(AERO(31), CYD), (AERO(32), CYD), (AERO(33), CMJ)</td>
<td>CBLCC220</td>
</tr>
<tr>
<td>(AERO(34), CYE), (AERO(35), CYE), (AERO(36), CMJ)</td>
<td>CBLCC230</td>
</tr>
<tr>
<td>(AERO(44), XE5), (AERO(45), XE5), (AERO(46), XE5)</td>
<td>CBLCC240</td>
</tr>
<tr>
<td>(AERO(47), XE5)</td>
<td>CBLCC250</td>
</tr>
</tbody>
</table>

**EQUIVALENCE**

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AERO(48), XE5), (AERO(49), XE5), (AERO(50), XE5)</td>
<td>CBLCC260</td>
</tr>
<tr>
<td>(AERO(51), XE5), (AERO(52), XE5), (AERO(53), XE5)</td>
<td>CBLCC270</td>
</tr>
<tr>
<td>(AERO(54), XE5), (AERO(55), XE5), (AERO(56), XE5)</td>
<td>CBLCC280</td>
</tr>
<tr>
<td>(AERO(57), XE5), (AERO(58), XE5), (AERO(59), XE5)</td>
<td>CBLCC290</td>
</tr>
<tr>
<td>(AERO(60), XE5), (AERO(61), XE5), (AERO(62), XE5)</td>
<td>CBLCC300</td>
</tr>
<tr>
<td>(AERO(63), XE5)</td>
<td>CBLCC310</td>
</tr>
</tbody>
</table>

**EQUIVALENCE**

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AERO(64), XE5), (AERO(67), XE5), (AERO(68), XE5)</td>
<td>CBLCC320</td>
</tr>
<tr>
<td>(AERO(69), XE5), (AERO(70), XE5), (AERO(71), XE5)</td>
<td>CBLCC330</td>
</tr>
<tr>
<td>(AERO(72), XE5), (AERO(73), XE5), (AERO(74), XE5)</td>
<td>CBLCC340</td>
</tr>
<tr>
<td>(AERO(75), XE5), (AERO(76), XE5), (AERO(77), XE5)</td>
<td>CBLCC350</td>
</tr>
<tr>
<td>(AERO(78), XE5), (AERO(79), XE5), (AERO(80), XE5)</td>
<td>CBLCC360</td>
</tr>
<tr>
<td>(AERO(81), XE5), (AERO(82), XE5), (AERO(83), XE5)</td>
<td>CBLCC370</td>
</tr>
<tr>
<td>(AERO(84), XE5), (AERO(85), XE5), (AERO(86), XE5)</td>
<td>CBLCC380</td>
</tr>
<tr>
<td>(AERO(87), XE5), (AERO(88), XE5), (AERO(89), XE5)</td>
<td>CBLCC390</td>
</tr>
<tr>
<td>(AERO(90), XE5), (AERO(91), XE5), (AERO(92), XE5)</td>
<td>CBLCC400</td>
</tr>
<tr>
<td>(AERO(93), XE5), (AERO(94), XE5), (AERO(95), XE5)</td>
<td>CBLCC410</td>
</tr>
<tr>
<td>(AERO(96), XE5), (AERO(97), XE5), (AERO(98), XE5)</td>
<td>CBLCC420</td>
</tr>
<tr>
<td>(AERO(99), XE5), (AERO(100), XE5), (AERO(101), XE5)</td>
<td>CBLCC430</td>
</tr>
<tr>
<td>(AERO(102), XE5), (AERO(103), XE5), (AERO(104), XE5)</td>
<td>CBLCC440</td>
</tr>
<tr>
<td>(AERO(105), XE5), (AERO(106), XE5), (AERO(107), XE5)</td>
<td>CBLCC450</td>
</tr>
<tr>
<td>(AERO(108), XE5), (AERO(109), XE5), (AERO(110), XE5)</td>
<td>CBLCC460</td>
</tr>
<tr>
<td>(AERO(111), XE5), (AERO(112), XE5), (AERO(113), XE5)</td>
<td>CBLCC470</td>
</tr>
<tr>
<td>(AERO(114), XE5), (AERO(115), XE5), (AERO(116), XE5)</td>
<td>CBLCC480</td>
</tr>
<tr>
<td>(AERO(117), XE5), (AERO(118), XE5), (AERO(119), XE5)</td>
<td>CBLCC490</td>
</tr>
<tr>
<td>(AERO(120), XE5), (AERO(121), XE5), (AERO(122), XE5)</td>
<td>CBLCC500</td>
</tr>
<tr>
<td>(AERO(123), XE5), (AERO(124), XE5), (AERO(125), XE5)</td>
<td>CBLCC510</td>
</tr>
<tr>
<td>(AERO(126), XE5), (AERO(127), XE5), (AERO(128), XE5)</td>
<td>CBLCC520</td>
</tr>
</tbody>
</table>

**EQUIVALENCE**

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AERO(1), CXUP), (AERO(2), CXUP), (AERO(3), CXUP)</td>
<td>CBLCC530</td>
</tr>
<tr>
<td>(AERO(4), CXUP), (AERO(5), CXUP), (AERO(6), CXUP)</td>
<td>CBLCC540</td>
</tr>
<tr>
<td>(AERO(7), CXUP), (AERO(8), CXUP), (AERO(9), CXUP)</td>
<td>CBLCC550</td>
</tr>
</tbody>
</table>
FILE CABLE FORTRAN TI  GRUMMAN DATA SYSTEMS

2 (AEROP( 7), CUDP), (AEROP( 8), CZD), (AEROP( 9), CMOP), C3LOC5460
3 (AEROP(10), CUDP), (AEROP(11), CZD), (AEROP(12), CMOP), C3LOC5467
4 (AEROP(13), CUDP), (AEROP(14), CZD), (AEROP(15), CMOP), C3LOC5493
5 (AEROP(16), CUDP), (AEROP(17), CZD), (AEROP(18), CMOP), C3LOC5496
6 (AEROP(19), CUDP), (AEROP(20), CLDP), (AEROP(21), CMOP), C3LOC6260
7 (AEROP(22), CUDP), (AEROP(23), CLDP), (AEROP(24), CMOP), C3LOC6263
8 (AEROP(25), CUDP), (AEROP(26), CLDP), (AEROP(27), CMOP), C3LOC6269
9 (AEROP(28), CUDP), (AEROP(29), CLDP), (AEROP(30), CMOP), C3LOC6263
A (AEROP(31), CUDP), (AEROP(32), CLDP), (AEROP(33), CMOP), C3LOC6446
B (AEROP(34), CUDP), (AEROP(35), CLDP), (AEROP(36), CMOP), C3LOC6450

EQUIVALENCE (SN( 1), GX1), (SN( 2), GY1), (SN( 3), GZ1), C3LOC650
1 (SN( 4), GX2), (SN( 5), GY2), (SN( 6), GZ2), C3LOC6576
2 (SN( 7), GX3), (SN( 8), GY3), (SN( 9), GZ3), C3LOC6692
3 (SN(10), GX4), (SN(11), GY4), (SN(12), GZ4), C3LOC6549
4 (SN(13), TH1), (SN(14), TH2), (SN(15), ALU), C3LOC7270
5 (SN(16), ALL), C3LOC7130
6 (SN(17), THG1), (SN(18), THG1), (SN(19), THG1), C3LOC7270
7 (SN(20), THG2), (SN(21), THG2), (SN(22), THG2), C3LOC7375
8 (SN(23), THG3), (SN(24), THG3), (SN(25), THG3), C3LOC7841
9 (SN(26), THG4), (SN(27), THG4), (SN(28), THG4), C3LOC7575

KASE=0
IP=5
IW=6
LLL=*

11 SAVE(J)=9999
12 A=3(I)=C.

27 WRITE(IW,170)(KDE(I),I=1,20)
170 WRITE(IW,3X,'CASE NG=',I3,4X,20A4)

10 CALL IFE

20 CALL TABIN1(I3,36,NG)
30 FORMAT(*,'ERROR IN READING TABLES 1-36,NG=',I2)

50 GO TO 500
20 READ(IW,130)(AERD(I),I=44,59)
30 READ(IW,120)(AERD(I),I=66,120)

100 FORMAT(/612,5)

103 FORMAT(*,'ERROR IN READING SNUBUFF DATA TABLE,NG=',I3)

40 FORMAT(/612,5)

42 FORMAT(*,'ERROR IN READING SNUBUFF DATA TABLE,NG=',I3)

39
GO TO 500
100 DO 28 I=1,150
28 AER(I)=SAVE(I)
READ(IF,150,END=500)(TITLE(I),I=1,20)
150 FORMAT(20A4)
CASE=1
DO 34 J=1,50
34 SAVE(J)=9999.
READ(IF,350)(KODE(I),I=1,24)
WRITE(1W,170) KODE(I),(TITLE(I),I=1,20)
CALL RITE
IK=;
DO 24 I=1,160
READ(IF,350)K,VALUE
IKM(I)=K
IF(KLT.1)GO TO 22
IK=IKM+1
AERO(K)=VALUE
22 IF(KLT.37)SAVE(K)=AERO(K)
WRITE(1W,193)(I,1=1,24),(KODE(I),I=1,24)
171 FORMAT(//* CODE NOS. FOR THIS CASE,*//2415/*//2415)
WRITE(1W,352)
352 FORMAT(3X,*DATA CHANGE*)
357 FORMAT(13,E12.5)
IF(1K=12.5)GO TO 24
DO 24 I=1,*
K=IKM(I)
VALUE=AERO(I)
24 WRITE(1W,35) VALUE
351 FORMAT(3X,13,3X,J12.5)
LL=3
32 IF(KODE(7).EQ.0) GO TO 31
DO 30 I=1,36
CALL STINT1(AWCH,0,0,J,I,AERO(I),NG)
IF(NG.NE.0) GO TO 40
30 CONTINUE
DO 36 J=1,36
36 IF(SAVE(J).NE.9999.) AERO(J)=SAVE(J)
GO TO 31
40 WRITE(1W,400) J,NG
400 FORMAT(//,* ERROR IN TABLE NO:*14,*NG=*13)
GO TO 500
360 FORMAT(5E12.3)
31 IF(CASE.EQ.1) GO TO 9
WRITE(1W,971)
801 FORMAT(5X,* INPUT DATA AS SPECIFIED IN AERO ARRAY*)
WRITE(1W,800)(I,AERO(1),I=1,150)
800 FORMAT(5(2X,*AERO(*:13,1)="G10.3))
9 DO 25 I=1,150
25 SAVE(I)=AERO(I)
IF(KODE(3).EQ.0) GO TO 48
IF(KODE(3).EQ.2)WRITE(1W,43)
42 FORMAT(* FREQUENCY RESPONSE COMPUTATION*)
IF(KODE(3).EQ.2)GO TO 49
42 DO 27 I=1,150
27 AERD(I)=SAVE(I)
   CALL OUTLOC
   IF(IALL.EQ.1) GO TO 1000
48 CALL TWIN
   IF(KODE(5).EQ.0) GO TO 49
   WRITE(IW,92)
802 FORMAT(4X,"AERO DATA IN STAB. AXIS AT EQUAT. REF. CENTER")
   WRITE(IW,968)(I,AERD(I),I=1,150)
49 CALL TWIN
   CALL TWIN
   IF(KODE(5).EQ.0) GO TO 50
   WRITE(IW,98)
802 FORMAT(4X,"AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER")
   WRITE(IW,998)(I,AERD(I),I=1,150)
804 FORMAT(4X,"AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER")
   WRITE(IW,1008)(I,AERD(I),I=1,150)
   5 IF(KODE(2)) 70,30,90
70 WRITE(IW,730)
700 FORMAT(3* "**** LATERAL/DIRECTIONAL STABILITY ****")
   IF(KODE(14).EQ.0) GO TO 702
   IDX=KODE(14)
   IDN=KODE(15)
   IF(KODE(13).NE.-1.) GO TO 706
   IF(KODE(15).EQ.3.) IDN=4
   IF(KODE(15).LE.3.) GO TO 706
   KODE(15)=3.
   WRITE(IW,737)
7027 FORMAT(3X,*KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION,KODE(1))
   15) IS SET TO 3.*)
   WRITE(IW,731)NAME(IDN),NAME(IDX)
7513 FORMAT(* COMPUTATION OF */A*/A/ NUMERATOR ROOTS*)
752 CALL LUNG
   IF(KODE(3).EQ.1) GO TO 42
   GO TO 1027
   WRITE(IW,750)
750 FORMAT(3X,* **** LATERAL/DIRECTIONAL STABILITY ****)
   IF(KODE(16).EQ.1) GO TO 703
   IDX=KODE(16)
   IDN=KODE(17)
   WRITE(IW,754)NAME(IDN),NAME(IDX)
703 CALL_LAT
   IF(KODE(3).EQ.1) GO TO 42
   GO TO 1000
90 WRITE(IW,750)
   IF(KODE(14).EQ.0) GO TO 704
   IDX=KODE(14)
   IDN=KODE(15)
   IF(KODE(13).NE.-1.) GO TO 1069
   IF(KODE(15).EQ.3.) IDN=4
   IF(KODE(15).LE.3.) GO TO 1068
   KODE(15)=3.
   WRITE(IW,757)
764 WRITE(IW,751)NAME(IDN),NAME(IDX)
764 CALL_LONG
   WRITE(IW,750)
FILE: CABLE FORTRAN II

GRUMMAN DATA SYSTEMS

105 CALL LAT
IF(KODE(3).EQ.1) GO TO 42
GO TO 1000

500 STOP
END
SUBROUTINE OUTLOC
COMMON/INDUT/W,IE
COMMON/CAM/AERO(175),AERO(50),KODE(25),LL
IF(LL.GT.0) GO TO 42
II=KODE(4)
VARY = ABS(AERO(II)*1)
ANOM = AERO(II)
L=0
LL=1
WRITE(IW,402) II

600 FORMAT(1H1,3X,* FOOT LOCUS VARYING AERO(*,13,*))

42 L=L+1
II=KODE(4)
AERO(II)=ANOM-5.*VARY+L=VARY
IF(LL.GT.0) GO TO 44
WRITE(IW,130) KODE(4),AERO(II)
180 FORMAT(2X,5AERO(II,13,2H)=G12.5)
RETURN
44 AERO(II)=ANOM
L=0
RETURN
END

BLOCK DATA
COMMON/NAME/NAME(16),NAME1(16)
DATA NAME/* Z ** THET ** CT ** X ** DTFB ** THMX ** EM **
** THDO ** EMT ** PSI ** PHI ** DTFB ** PSI **
** DTC ** EM ** DT **
** DEL **
** DEL **
** DEL **

SUBROUTINE FREQ (ROOTS,K4A,TFG)
COMMON/INDUT/W,IE
COMMON/CAM/AERO(175),AERO(50),KODE(26),LL
COMMON/PLCT/O(51),AMP(61),ANGLE(61),XMP(61),KV
COMMON/NAME/NAME(16),NAME1(16)
COMPLEX ROOTS(1)
COMPLEX CN(29)
DIMENSION DOW(21)
DATA DOW/1.1,1.5,1.7,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0,8.5,9.0,9.5,10.0/

II=0
IN1=KODE(14)
IN2=KODE(15)
IF(KODE(13).NE.-1) GO TO 32
IN1=KODE(15)
IN2=IN2+4
IF(KODE(15).LE.3) GO TO 32

CBLZ2210
CBLZ2220
CBLZ2230
CBLZ2240
CBLZ2250
CBLZ2260
CBLZ2270
CBLZ2280
CBLZ2290
CBLZ2300
CBLZ2310
CBLZ2320
CBLZ2330
CBLZ2340
CBLZ2350
CBLZ2360
CBLZ2370
CBLZ2380
CBLZ2390
CBLZ2400
CBLZ2410
CBLZ2420
CBLZ2430
CBLZ2440
CBLZ2450
CBLZ2460
CBLZ2470
CBLZ2480
CBLZ2490
CBLZ2500
CBLZ2510
CBLZ2520
CBLZ2530
CBLZ2540
CBLZ2550
CBLZ2560
CBLZ2570
CBLZ2580
CBLZ2590
CBLZ2600
CBLZ2610
CBLZ2620
CBLZ2630
CBLZ2640
CBLZ2650
CBLZ2660
CBLZ2670
CBLZ2680
CBLZ2690
CBLZ2700
CBLZ2710
CBLZ2720
CBLZ2730
CBLZ2740
CBLZ2750
CBLZ2760

42
FILE 2: CABLE

FORMA3(3XI,KDE(16)) IS INCORRECT FOR CABLELESS MODEL OPTION.KODE(1)=CABLE

1) 3. GO TO 31

32. ENTRY FREQ2(ROUN1, K4A, TFG)

IL=1

IN=KDE(16)

IV=KDE(17)

31 CALL ANP(CVU, C, KN, AMP20, PHSD, ITPN)

CALL ANP(FDTS, C, K4A, AMP20, PHSD, ITPD)

TGA=VG/T2G

SGN=ABS(T2G/TGA)

IF(AMP20.NE.0) SGN=T2G+AMP20/AMP20

ITYP=ITYPD-ITPN

IF(KODE(19).LE.10)GO TO 3

IN=2

IK=1

GO TO 4

3 IF(KODE(19).LE.5)GO TO 5

IN=1

IK=2

GO TO 4

5 IN=5

IK=4

4 INIT=KDE(18)

K=IN+3+1

K=K

IDX=2

DO 1 I=1, K

IDX=IDX+1

1 IF(IDX.LE.IN)GO TO 2

INIT=INIT+1

IDX=1

2 C4)IDM1(I)=DM1((IDX-1)*I+1I+10)*1*INIT

CALL ANP(CVU, DM1 (I), KN, AMP20, PHSD, IDW)

CALL ANP(FDTS, DM1 (I), K4A, AMP20, PHSD, IDW)

ANP(I)=20.0*(ALOGIC(AMP20/AMP20)+ALOGIC(ABS(TGAIV))

ANG(I)=TGAIV+AMP20/AMP20

ANGLE(I)=(PHSH-PSHD)*37.2958

IF(SGN.LT.0.) ANGLE(I)=ANGLE(I)180.

1 CONTINUE

IF(IL.EQ.0) WRITE(IW, 10) NAME(IN2), NAME(IN1)

IF(IN.EQ.0) WRITE(IW, 10) NAME(IN2), NAME(IN1)

10 FORMAT(1X, 10, FREQUENCY RESPONSE OF THE 9.2X, 1AA, 14A, 2X,

1 TRANSFER FUNCTION)

1 IF(AMP20.NE.0) WRITE(IW, 17) SGN

1 IF(AMP20.EQ.0) WRITE(IW, 18) ITYP

17 FORMAT(* STEADY STATE GAIN = 9.2X, 11.

18 FORMAT(* SYSTEM TYPE = * 9.2X, 14)

1 IF(IN.GE.2) GO TO 6

WRITE(IW, 11)


1 9.2X, AMP VALUE)

DO 7 I=1,K
FILE  CABLE
FORTRAN I  
GRUMMAN DATA SYSTEM

7 WRITE (IW,12)CM(I),AMP(I),ANGLE(I),XMP(I)
CBL0331C
12 FORMAT (42X,E11.4,5X,4(2X,E11.4))
CBL0332C
CBL0333C
6 WRITE (IW,11)
CBL0334C
13 FORMAT (/2X,4F3,D3) AMP RAT(DB),2X, PHASE(DEG),2X
CBL0335C
1,AMP.VALUE,7X.
CBL0336C
7 FREQ(PS) *2X,AMP ET(DB),2X, PHASE(DEG),2X,AMP. VALUE *)
CBL0337C
K=K/2+1
CBL0338C
DO 9 I=1,K
CBL0339C
IF (I+NE,K) GO TO 9
CBL0340C
WRITE (IW,15)CM(I+30),AMP(I+30),ANGLE(I+30),XMP(I+30)
CBL0341C
15 FORMAT (5X,4(2X,E11.4))
CBL0342C
GO TO 8
CBL0343C
9 WRITE (IW,12)CM(I),AMP(I),ANGLE(I),XMP(I),CM(3C+I),AMP(3C+I),
1 ANGLE(3C+I),XMP(30+I)
CBL0344C
8 WRITE (IW,14)
CBL0345C
14 FORMAT (1H1)
CBL0346C
RETURN
ENTRY FREQ1(ROO1S,K4A,KFG)
CBL0347C
KN=K4A
CBL0348C
TGN=KFG
CBL0349C
IF (KN.EQ.0) RETURN
CBL0350C
DO 20 I=1,K4A
CBL0351C
CNU(I)=ROOTS(I)
CBL0352C
2 CONTINUE
RETURN
CBL0353C
C DEBUG UNIT(3), INIT
CBL0354C
END
CBL0355C
SUBROUTINE AMP(CXU,CX,XX,AMP,ANG,ITYPE)
CBL0356C
DIMENSION CXU(2,1)
CBL0357C
ITYPE=C
CBL0358C
ANG=0,
CBL0359C
AMP=1,0
CBL0360C
IF (XX.EQ.0) RETURN
CBL0361C
DO 1 I=1,XX
CBL0362C
XQL=-CXU(1,1)
CBL0363C
Y1M=DM-CXU(2,1)
CBL0364C
AMP2=SQRT(XQL*XQL+Y1M*Y1M)*AMP
CBL0365C
IF (XQL.EQ.0.0 AND Y1M.EQ.0.0) GO TO 2
CBL0366C
ANG=ATAN2(Y1M,XQL)*AMP
CBL0367C
GO TO 1
CBL0368C
2 ANG=ANG
CBL0369C
ITYPE=ITYPE+1
CBL0370C
1 CONTINUE
RETURN
CBL0371C
C DEBUG UNIT(3), INIT(ANG,XQL,Y1M)
CBL0372C
END
CBL0373C
SUBROUTINE TRANS
CBL0374C
C THIS ROUTINE CALCULATES BODY AXIS AERO DATA AT CR FROM STAR.
CBL0375C
C AXIS AERO DATA AT CR
CBL0376C
COMMON /DAT/ AERO(175),AEROP(5C),KODE(25),LL
CBL0377C
EQUIVALENCE(AERO(1),CDU),(AERO(2),CLU),(AERO(3),CMU),
CBL0378C
1 (AERO(4),CDA),(AERO(5),CLA),(AERO(6),CMA),
CBL0379C
2 (AERO(7),CDO),(AERO(8),CLO),(AERO(9),CM),
CBL0380C
3 (AERO(10),CJO),(AERO(11),CLO),(AERO(12),CM),
CBL0381C
44
<table>
<thead>
<tr>
<th>FILE: CABLE</th>
<th>FORTRAN T1</th>
<th>GRUMMAN DATA SYSTEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(AERO(13), C00F), (AERO(14), C14F), (AERO(15), C15F)</td>
<td>CAB30030</td>
</tr>
<tr>
<td></td>
<td>(AERO(16), C01D), (AERO(17), CLAD), (AERO(18), CNAD)</td>
<td>CAB30130</td>
</tr>
<tr>
<td></td>
<td>(AERO(19), CLA), (AERO(20), CLA), (AERO(21), CN1)</td>
<td>CABC0110</td>
</tr>
<tr>
<td></td>
<td>(AERO(22), C32P), (AERO(23), CLP), (AERO(24), CAP)</td>
<td>CABC0120</td>
</tr>
<tr>
<td></td>
<td>(AERO(25), C46), (AERO(26), CLF), (AERO(27), CN4)</td>
<td>CABC0130</td>
</tr>
<tr>
<td></td>
<td>(AERO(28), C14F), (AERO(29), C14F), (AERO(30), CN4)</td>
<td>CABC0140</td>
</tr>
<tr>
<td></td>
<td>(AERO(31), CLA), (AERO(32), CLA), (AERO(33), CN1A)</td>
<td>CABC0150</td>
</tr>
<tr>
<td></td>
<td>(AERO(34), CYOS), (AERO(35), CLOS), (AERO(36), CNOS)</td>
<td>CABC0160</td>
</tr>
<tr>
<td></td>
<td>(AERO(44), REF), (AERO(45), REF), (AERO(46), XG)</td>
<td>CABC0170</td>
</tr>
<tr>
<td></td>
<td>(AERO(47), ZCG), (AERO(63), THTA)</td>
<td>CABC0190</td>
</tr>
<tr>
<td></td>
<td>EQUIVALENCES (AERO(1), CXUP), (AERO(2), C2UP), (AERO(3), CMUP)</td>
<td>CABC0120</td>
</tr>
<tr>
<td></td>
<td>(AERO(4), C3AP), (AERO(5), C3AP), (AERO(6), CMAP)</td>
<td>CABC0230</td>
</tr>
<tr>
<td></td>
<td>(AERO(7), CXOP), (AERO(8), C2OP), (AERO(9), CMOP)</td>
<td>CABC0240</td>
</tr>
<tr>
<td></td>
<td>(AERO(10), CXOP), (AERO(11), C2OP), (AERO(12), CMOP)</td>
<td>CABC0250</td>
</tr>
<tr>
<td></td>
<td>(AERO(13), CXOP), (AERO(14), C2OP), (AERO(15), CMOP)</td>
<td>CABC0260</td>
</tr>
<tr>
<td></td>
<td>(AERO(16), CXOP), (AERO(17), C2OP), (AERO(18), CMOP)</td>
<td>CABC0270</td>
</tr>
<tr>
<td></td>
<td>(AERO(19), C43P), (AERO(20), C23P), (AERO(21), CN3P)</td>
<td>CABC0280</td>
</tr>
<tr>
<td></td>
<td>(AERO(22), C46), (AERO(23), CLP), (AERO(24), CN4P)</td>
<td>CABC0290</td>
</tr>
<tr>
<td></td>
<td>(AERO(25), C46), (AERO(26), CLF), (AERO(27), CN4)</td>
<td>CABC0300</td>
</tr>
</tbody>
</table>

ALPHA=THETA  
SNDF= Sin(ALPHA)  
CMALF= Cos(ALPHA)  
SN3Q = SNDF**2  
COSO = C3ALF**2  
SNCD = SNDF*CCDF  
C0U=CCU+2*(CD+CD+*THETA)  
CLU=CLU+2*(CLA+CLA+*THETA)  
CDAD=CDAD+CLA+CLA+*THETA)  
CLA=CLA+CD2+CD2+*THETA  
CXUP=CLA+SN5Q-CD4+C5Q+(CLA+CLU)*SNCO  
C2ZUP=CD4+SN5O+CLA+CLA+CLU+SNCO  
CMUP=-CVA+SNDF+CVU+*C3ALF  
CXAP=CD4+SN5O+CLA+CLA+CLU+SNCO  
C32P=CVU+SNDF+CVU+*C3ALF  
CXOP=CLA+SNDF+CCO+*C3ALF  
C2ZP=CLA+SNDF+CLA+*SN6P  
C13P=CLA+*SN6P+CLA+*SN6P  
C3OP=CVU  
CXADP=-C3AD+*C3ALF+CLA+*SN6P  
C32AP=-C3AD+*C3ALF+CLA+*SN6P  
C32AP=-C3AD+*CLA+*SN6P  
CXOP=CLAD+*C3ALF+CLA+*SN6P  
C32P=CLAD+*C3ALF+CLA+*SN6P  
CMUP=CVA+  
C32P=CL3+SNDF+CVB+*C3ALF  
CLAD=CVB+SNDF+CLA+*C3ALF  
CYPF=(CVF+SNDF+CVF+*C3ALF)  
C2NP=(-CLA+SN5Q+CVF+CD2)+CLA(=CN5)*SNCO)
FILE3 CABLE FORTRAN T1
GRUMMAN DATA SYSTEMS

CLP= ( CN*SN50 + CLP*CSO - CLP + CN0)*SNCO)

CYR= ( CYR*SNALF + CYR*CALF)

CN0=( CLP*SN50 + CN0*CSO + (CLP + CN0)*SNCO)

CLP= (-CN0*SN50 + CLP*CSO + (CLP - CN0)*SNCO)

CYDA= CYDA

CNDA= CLD*SNALF + CNDA*CALF

CLDA= -CNDA*SNALF + CLD*CALF

CYDR= CYDR

CNDR= CLD*SNALF + CNDR*CALF

CLDR= -CNDR*SNALF + CLD*CALF

RETURN

END

SUBROUTINE TRANSFORMS INERTIA DATA & STABILITY AXIS AERO DATA

TO THE EQUATION REFERENCE CENTER

COMMON/DATA/AERO(175),AEP(50),KODE(26),LL

EQUIVALENCE(AERO(1), CDU)*AERO(2), CLU)*AERO(3), CMU), CABC0330,
1 (AERO(4), CD4)*AERO(5), CLA)*AERO(6), CMA), CABC0340

2 (AERO(7), CD5)*AERO(8), CLD)*AERO(9), CM9), CABC03450

3 (AERO(10), CD6)*AERO(11), CLF)*AERO(12), CM3), CABC0350

4 (AERO(13), CD7)*AERO(14), CLH)*AERO(15), CME), CABC03570

5 (AERO(16), CD8)*AERO(17), CJL)*AERO(18), CM4), CABC0360

6 (AERO(19), CY3)*AERO(20), C12)*AERO(21), CM5), CABC03690

7 (AERO(22), CY1)*AERO(23), CLP)*AERO(24), CNI), CABC03700

8 (AERO(25), CY2)*AERO(26), C1D)*AERO(27), CN9), CABC03910

9 (AERO(28), CY3)*AERO(29), CLS)*AERO(30), CN0), CABC03920

A (AERO(31), CY4)*AERO(32), CLA)*AERO(33), CND4), CABC03930

B (AERO(34), CY5)*AERO(35), CLS)*AERO(36), CN5), CABC03940

C (AERO(44), XEF)*AERO(45), ZEF)*AERO(46), XCG3), CABC03950

D (AERO(47), ZCG)*AERO(63), THETA), CABC03950

EQUIVALENCE(AERO (-1), AMACH)*AERO (49), V0 ), (AERO (50), AM), CABC03970

EQUIVALENCE(AERO (51), W0 )*AERO (52), W0 ), (AERO (53), B), CABC03940

EQUIVALENCE(AERO (54), CSO)*AERO (55), SW ), (AERO (56), XI12), CABC03930

EQUIVALENCE(AERO (57), X12)*AERO (58), Y12 ), (AERO (59), Z12), CABC03930

EQUIVALENCE(AERO (60), Y12)*AERO (61), CDT ), (AERO (62), CM1), CABC03930

C INERTIA TRANSFORMATIONS

X=XCG/12

Z=ZCG/12

X12=X12+AM*(Z12)

Y12=Y12+AM*(X12)

Z12=Z12+AM*(X12)

X12=X12+AM*X12

C AERO DATA TRANSFORMATIONS

X=XREF/(12.*CBAF)

Z=XREF/(12.*CBAF)

CMA=CMA-Z*CD0+X*CL0

CAY=CAY-X*CL0+Z*CD0

CL0=CL0-2.*X*CLA+4.*Z*CL0

CD0=CD0-2.*X*CD0+4.*Z*CD0

CMA=CMA-Z*CD0+X*CL0

CAY=CAY-X*CL0+Z*CD0

X=XREF/(12.*0)
<table>
<thead>
<tr>
<th>FILED CABLE FORTRAN TI</th>
<th>GRUMMAN DATA SYSTEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ Z = 2 \cos(12 \cdot \theta) ]</td>
<td>[ \text{CABC1120} ]</td>
</tr>
<tr>
<td>[ C = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1120} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
<tr>
<td>[ \cos \theta = (\cos^2 x + \sin^2 x + \sin^2 y + 2 \cdot x \cdot \cos y) ]</td>
<td>[ \text{CABC1210} ]</td>
</tr>
</tbody>
</table>
FILE: CABLE FORTRAN T1

SPS AND DATA SYSTEMS

DI 1005 I=1,3
DO 1005 J=1,3
SNU(I,J)=0

1005 SNU(J,I)=0
DO 1005 I=1,10
DO 1005 J=1,10

1005 DUM(I,J)=0
IF(KODE(12).EQ.0) GO TO 1002

C TERMS FOR SNUBBER EFFECTS (LAT)
CALL DPCSN(THEETA)
IF(KODE(12)=1) CALL DRCUSN(THEETA)
DUM(1,2) = -TUSN*GXI
DUM(1,3) = TUSN*GZI
DUM(1,5) = -TUSN*SIN(THY1)
DUM(1,7) = GY1
DUM(2,2) = SNUX*TUSNO*GXI/12.+SNUY*TUSNO*GY1/12.
DUM(2,3) = -SNUX*TUSNO*GZI/12.
DUM(2,4) = -SNUY*TUSNO*SIN(THGY1)/12.
DUM(2,5) = SNUX*TUSNO*SIN(THGY1)/12.
DUM(2,7) = (+SNUX*GY1+SNUY*GXI)/12.
DUM(3,2) = -SNUZ*TUSNO*GXI/12.
DUM(3,3) = SNUZ*TUSNO*GZI/12.+SNUY*TUSNO*GY1/12.
DUM(3,5) = -SNUZ*TUSNO*SIN(THGY1)/12.
DUM(3,7) = (+SNUY*GZI+SNUZ*GY1)/12.
DUM(4,1) = GXY(GY1,THGX1,ALU)
DUM(4,2) = GXSY(-SNUY,THGX1,-SNUX,GY1,THGX1,ALU)
DUM(4,3) = GPHI(-SNUZ,GY1,THGX1,-SNUY,GZI,THGX1,ALU)
DUM(4,4) = -1.
DUM(5,1) = GYY(THGY1,ALU)
DUM(5,2) = GYSY(-SNUX,GXI,THGY1,-SNUX,THGY1,ALU)
DUM(5,3) = GYPHI(-SNUZ,GY1,THGX1,-SNUY,GZI,THGX1,ALU)
DUM(5,5) = -1.
DUM(6,1) = GZY(-GY1,THGX1,ALU)
DUM(6,2) = GZSY(-SNUY,GXI,THGX1,-SNUX,GY1,THGX1,ALU)
DUM(6,3) = GZPHI(-SNUZ,GY1,THGX1,-SNUY,THGX1,ALU)
DUM(6,6) = -1.
IF(KODE(12).EQ.2) GO TO 1010
CALL DPCSN(THEETA)

Q=5*RH0*V0*V0
ALU1=ALU1+
CALL STINT(Q,ALU1,0.1,1,TUSNI,NG)
IF(NG*NE.0) GO TO 5900
ALU2=ALU1+
CALL STINT(Q,ALU2,0.1,1,TUSN2,NG)
IF(NG*NE.0) GO TO 5900
GO TO 5001

5000 FORMAT(1W,5002) NG,ALL,ALU,G
5002 FORMAT(1*ERROR IN SNUBBER TABLE 1*NG=*.12,3*10.3)
RETURN

5001 CONTINUE
AKTU=(TUSN1-TUSN2)/2.
AKSNU=AKTU
1010 CONTINUE
DUM(7,7) = -1.

48
DUM(7, J) = AKSNU*12.
DUM(8, 1) = ALY(GY1)
DUM(8, 2) = ASY(-SNUY, GX1, -SNUX, GZ1)
DUM(8, 3) = ALPHI(-SNUX, GY1, -SNUY, GZ1)
DUM(9, 8) = -1.
IF(KODE(10), EQ, 1) GO TO 1015
DD 101 A I = 1, 3
DD 1016 J = 1, 3
1016 SNVD(1, J) = DUM(1, 7) * ADSNU + DUM(9, J) * 12.
1015 CALL MASH(3, B)
   J = 1, 3
   J = 1, 3
1050 TDRY(J, I) = DUM(I, J)
IF(KODE(10), EQ, 1) CALL DCUSN(THEATA)
DUM(1, 2) = TUSN*GX2
DUM(1, 3) = TUSN*GZ1
DUM(1, 5) = TUSN*SIN(THGY2)
DUM(1, 7) = GY2
DUM(2, 2) = SNUX*TUSNO*GX2/12, -SNUX*TUSNO*GY2/12.
DUM(2, 3) = -SNUX*TUSNO*GZ2/12.
DUM(2, 4) = SNUX*TUSNO*SIN(THGY2)/12.
DUM(2, 5) = SNUX*TUSNO*SIN(THGY2)/12.
DUM(2, 7) = (-SNUX*GY2-SNUY*GX2)/12.
DUM(3, 2) = -SNUX*TUSNO*GX2/12.
DUM(3, 3) = SNUX*TUSNO*GZ2/12, -SNUX*TUSNO*GY2/12.
DUM(3, 5) = -SNUX*TUSNO*SIN(THGY2)/12.
DUM(3, 7) = (SNUX*GZ2-SNUX*GY2)/12.
DUM(4, 1) = GXY(GY2, THX2, ALU)
DUM(4, 2) = GXY(SNUY, THX2, -SNUX, GY2, THGY2, ALU)
DUM(4, 3) = GXY(SNUY, GY2, THX2, SNUY, GZ2, THGY2, ALU)
DUM(4, 4) = -1.
DUM(5, 1) = GY2(GY2, THX2, ALU)
DUM(5, 2) = GY2(SNUY, GX2, THGY2, -SNUX, GY2, ALU)
DUM(5, 3) = GY2(SNUY, GX2, THGY2, SNUY, GZ2, THGY2, ALU)
DUM(5, 5) = -1.
DUM(6, 1) = GXY(GY2, THGZ2, ALU)
DUM(6, 3) = GXY(SNUY, GX2, THGZ2, -SNUX, GY2, THGZ2, ALU)
DUM(6, 6) = -1.
IF(KODE(10), EQ, 2) GO TO 1020
CALL DCUSV(THEATA)
   ALU1 = ALU1 + 1.
   CALL STINT(0, ALU1, 0, 1, 1, TUSN1, NG)
IF(NG, NE, 0) GO TO 5000
ALU2 = ALU1 + 1.
   CALL STINT(0, ALU2, 0, 1, 1, TUSN2, NG)
IF(NG, NE, 0) GO TO 5000
AKTU = (TUSN1 - TUSN2)/2.
AKSNU = AKTU
1020 CONTINUE
DUM(1, 7) = -1.
DUM(7, 1) = AKSNU*12.
DUM(8, 1) = ALY(GY2)
DUM(8, 2) = ASY(SNUY, GX2, -SNUX, GY2)
DUM(8,3) = ALPHI(-SNUZ,GY2,SNUY,GZ2)
DUM(8,4) = -1.
IF(KODE(10) .EQ. 1) GO TO 1025
DO 1026 I = 1,3
DO 26 J = 1,3
1026 SNUD(I,J) = SNUD(I,J) + DUM(I,J) + CALL MASH(3,A)
DO 1060 I = 1,3
DO 1060 J = 1,3
1060 TOLP(I,J) = DUM(I,J)
IF(KODE(10) .EQ. 1) CALL DRCU:N(THE T A)
DUM(1,7) = - TL SN3*GX3
DUM(1,8) = TL SN3*GZ3
DUM(1,5) = - TLSN3*SIN*(THGY3)
DUM(1,7) = GY3
DUM(2,2) = SNLX*TL SN3*GX3/12, - SNLY*TL SN3*GY3/12
DUM(2,3) = - SNLX*TL SN3*GZ3/12
DUM(2,4) = SNLY*TL SN3*SIN*(THGX3)/12
DUM(2,5) = SNLX*TL SN3*SIN*(THGY3)/12
DUM(2,7) = (-SNLX*GY3 - SNLY*GX3)/12
DUM(3,2) = SNLX*TL SN3*GX3/12
DUM(3,3) = SNLX*TL SN3*GZ3/12, - SNLY*TL SN3*GY3/12
DUM(3,5) = SNLX*TL SN3*SIN*(THGX3)/12
DUM(3,6) = - SNLX*TL SN3*SIN*(THG Z3)/12
DUM(3,7) = ( SNLY*GY3 - SNLY*GX3)/12
DUM(4,1) = GX Y(GY3, THGX3, ALL)
DUM(4,2) = GX SY(SNLX, ThGX3, - SNLX*GY3, THGX3, ALL)
DUM(4,3) = GXYI(SNLZ, GY3, THGX3, SNLY, GZ3, THGX3, ALL)
DUM(4,4) = -1.
DUM(5,1) = GYYI(THGY3, ALL)
DUM(5,2) = GYSY( SNLY, GX3, THGY3, - SNLX, THGY3, ALL)
DUM(5,3) = GYPH( SNLZ, THGY3, SNLY, GZ3, THGY3, ALL)
DUM(5,5) = GYYI(SNLZ, THGY3, SNLY, GZ3, THGY3, ALL)
DUM(6,1) = GZY(GY3, THGZ3, ALL)
DUM(6,2) = GZSY( SNLY, GX3, THGZ3, - SNLX, GY3, THGZ3, ALL)
DUM(6,3) = GZPHI(SNLZ, GY3, THGZ3, SNLY, THGZ3, ALL)
DUM(6,4) = -1.
IF(KODE(10) .EQ. 1) GO TO 1030
CALL DRCSUB(THE T A)
ALL = 1
CALL SINT (0, ALL1,C,1,1, T LSN1, NG)
IF(NG, NE, =) GO TO 5000
ALL2 = ALL1
CALL SINT(0, ALL2,C,1,1, T LSN2, NG)
IF(NG, NE, =) GO TO 5000
AKTL = (TL SN1 - T LSN2)/2
AKSNL = AKTL
1030 CONTINUE
DUM(7, 7) = -1.
DUM(7, 8) = AKSNL*12.
DUM(8, 1) = ALY(GY3)
DUM(9, 2) = ALSY( SNLY, GX3, - SNLX, GY3)
DUM(4, 7) = ALPHI( SNL7, GY3, SNLY, GZ3)
DUM(9, 9) = -1.
IF(KODE(10) .EQ. 1) GO TO 1035
DO 1036 I=1,3
DO 1036 J=1,3
1036 SNOD(I,J)=SNOD(I,J)+DUM(I,7)*ADSNL*DUM(B,J)*12.
1035 CALL MASH(3,8)
DO 1070 I=1,3
DO 1070 J=1,3
1075 POTL(I,J)= DUM(I,J)
IF(KODE(10)=EO.1) CALL DRCUSN('META)
DUM(1, 2) = - TLSNO*GX4
DUM(1, 3) = TLSNO*GZ4
DUM(1, 5) = - TLSNO*SIN(THGY4)
DUM(1, 7) = GY4
DUM(2, 1) = SNLX*TLSNO*GX4/12*+SNLY*TLSNO*GY4/12.
DUM(2, 3) = - SNLX*TLSNO*GZ4/12.
DUM(2, 4) = - SNLX*TLSNO*SIN(THGY4)/12.
DUM(2, 5) = SNLX*TLSNO*SIN(THGY4)/12.
DUM(2, 7) = (-SNLX*GY4+SNLY*GY4)/12.
DUM(3, 2) = SNLZ*TLSNO*GX4/12.
DUM(3, 3) = - SNLZ*TLSNO*GZ4/12+SNLY*TLSNO*GY4/12.
DUM(3, 5) = SNLZ*TLSNO*SIN(THGY4)/12.
DUM(3, 6) = SNLZ*TLSNO*SIN(THGY4)/12.
DUM(3, 7) = (-SNLZ*GZ4-SNLZ*GY4)/12.
DUM(4, 1) = GXY(1GY4, THGY4, ALL)
DUM(4, 2) = GXY(-SNLYGX4, SNLXGY4, THGY4, ALL)
DUM(4, 3) = GXYHI(SNLZGY4, THGY4, ALL)
DUM(4, 4) = -1.
DUM(5, 1) = GYY(THGY4, ALL)
DUM(5, 2) = GYY(-SNLYGX4, THGY4, ALL)
DUM(5, 3) = GYYHI(SNLZGY4, ALL)
DUM(5, 5) = -1.
DUM(6, 1) = GZY(1GY4, THGZ4, ALL)
DUM(6, 2) = GZY(-SNLYGX4, THGZ4, ALL)
DUM(6, 3) = GZYHI(SNLZGY4, ALL)
DUM(6, 6) = -1.
IF(KODE(10)=EO.1) CALL DSTN(1,0) GO TO 1040
CALL DCUSN('META)
ALL1=ALL+1.
CALL DSTN(0,0, ALL1, C, 1, 1, TLSN1, NG)
IF(NGNE0) GO TO 1000
ALL2=ALL -1.
CALL DSTN(0,0,ALL2, C, 1, 1, TLSN2, NG)
IF(NGNE0) GO TO 1000
AKTL=(TLSN1-TLSN2)/2.
AKSNL=AKTL
1040 CONTINUE
DUM(7, 7) = -1.
DUM(7, 9) = AKSNL*12.
DUM(8, 1) = AY(1GY4)
DUM(8, 2) = AY(-SNLYGX4, SNLYGX4)
DUM(8, 3) = AYHI(SNLZGY4, -3NLZGZ4)
DUM(8, 9) = -1.
IF(KODE(10)=EO.1) GO TO 1045
DO 1046 I=1,3
DO 1046 J=1,3
1046 SNOD(I,J)=SNOD(I,J)+DUM(I,7)*ADSNL*DUM(B,J)*12.

51
FILE: CABLE FORTRAN T1  GRUMMAN DATA SYSTEMS

1045 CALL 'ASH(1,8)  CABLE2520
DO 1040 I=1,3  CABLE2600
DO 1027 J=1,3  CABLE2610
1080 BTR(I,J)= DUM(I,J)  CABLE2620
DO 1090 I=1,3  CABLE2630
DO 1090 J=1,3  CABLE2640
1090 SNU(I,J)= TDPR(I,J)+TOPL(I,J)+TOTL(I,J)+4OTR(I,J)  CABLE2650
IF(KODE(10),EQ,2) RETURN  CABLE2660
DO 1095 I=1,3  CABLE2670
DO 1095 J=1,3  CABLE2680
1055 SNU(I,J)=0 RETURN  CABLE2690
1055 DO 1094 I=1,3  CABLE2710
DO 1094 J=1,3  CABLE2720
SNU(I,J)=0 RETURN  CABLE2730
1054 SNU(I,J)=0 RETURN  CABLE2740
SUBROUTINE TRIM  CABLE2750
COMMON/INOUT/TW,19  CABLE2760
COMMON /DAT/ AERO(175),AEROP(60),KODE(26),ILL  CABLE2770
COMMON /P,LYCHAI/STE,XMLTH(5),ADC(5,3),ARM(5,3),TR,TRT,TF  CABLE2780
DIMENSION ANG(5,3)  CABLE2790
EQUIVALENCE(AERO(1),CDU),(AERO(2),CLU),(AERO(3),CMU)  CABLE2800
1 (AERO(4),CDA),(AERO(5),CLA),(AERO(6),CMA)  CABLE2810
2 (AERO(7),CDQ),(AERO(8),CLQ),(AERO(9),CMQ)  CABLE2820
3 (AERO(10),CDO),(AERO(11),CLO),(AERO(12),CMO)  CABLE2830
4 (AERO(13),CDE),(AERO(14),CLD),(AERO(15),CMD)  CABLE2840
5 (AERO(16),CDG),(AERO(17),CLG),(AERO(18),CMG)  CABLE2850
6 (AERO(19),CYB),(AERO(20),CLB),(AERO(21),CNB)  CABLE2860
7 (AERO(22),CYP),(AERO(23),CLP),(AERO(24),CNP)  CABLE2870
8 (AERO(25),CPF),(AERO(26),CLF),(AERO(27),CNR)  CABLE2880
9 (AERO(29),CYD),(AERO(29),CLD),(AERO(30),CND)  CABLE2890
A (AERO(31),CDY),(AERO(32),CLO),(AERO(33),CNQ)  CABLE2900
B (AERO(34),CYS),(AERO(35),CLO),(AERO(36),CNS)  CABLE2910
EQUIVALENCE(AERO(46),XG),XG,(AERO(47),ZG)  CABLE2920
EQUIVALENCE(AERO (48),AMACH),(AERO (49),V0),(AERO (50),AM)  CABLE2930
EQUIVALENCE(AERO (51),FHO),(AERO (52),WTS),(AERO (53),R ) CABLE2940
EQUIVALENCE(AERO(54),CABR),(AERO(55),S),(AERO(56),XXZ)  CABLE2950
EQUVALENCE(AERO(57),XXX),(AERO(58),YYY),(AERO(59),ZXX)  CABLE2960
EQUVALENCE(AERO(60),CLT),(AERO(61),COT),(AERO(62),CMT)  CABLE2970
1(AERO(63),THETA)  CABLE2980
EQUVALENCE(AERO(66),WL),(AERO(67),WLF),(AERO(68),WLJ)  CABLE2990
1 (AERO(69),WLL),(AERO(70),WLJ),(AERO(71),WLF)  CABLE3000
2 (AERO(72),STA),(AERO(73),STA),(AERO(74),PLH)  CABLE3010
3 (AERO(75),RL),(AERO(76),WCL),(AERO(77),STA)  CABLE3020
4 (AERO(78),BFC),(AERO(79),E),(AERO(80),E)  CABLE3030
5 (AERO(81),AF),(AERO(82),TF),(AERO(83),HUC)  CABLE3040
6 (AERO(84),HLC),(AERO(85),HUC),(AERO(86),HLC)  CABLE3050
7 (AERO(87),DCF),(AERO(88),DCF),(AERO(89),ALF)  CABLE3060
8 (AERO(90),DV),(AERO(91),RF),(AERO(92),RF)  CABLE3070
9 (AERO(93),RF),(AERO(94),RF),(AERO(95),RF)  CABLE3080
A (AERO(96),AL),(AERO(97),SLT),(AERO(98),MLTT)  CABLE3090
B (AERO(99),TLE),(AERO(100),AKL),(AERO(101),ALLTC)  CABLE3100

52
FILE: CABLE

EQUIVALENCE(AEROP(1), CXUP), (AEROP(2), CZUP), (AEROP(3), CWUP), CABC390
1 (AEROP(4), CXAP), (AEROP(5), CZAP), (AEROP(6), CMAP), CABC400
2 (AEROP(7), CXQP), (AEROP(8), CZQP), (AEROP(9), CMQP), CABC410
3 (AEROP(10), CXDP), (AEROP(11), CZDP), (AEROP(12), CMDP), CABC420
4 (AEROP(13), CXAP), (AEROP(14), CZAP), (AEROP(15), CMAP), CABC430
5 (AEROP(16), CXDP), (AEROP(17), CZDP), (AEROP(18), CMDP), CABC440
6 (AEROP(19), CXQP), (AEROP(20), CL6P), (AEROP(21), CN3P), CABC450
7 (AEROP(22), CYDP), (AEROP(23), CLPP), (AEROP(24), CN4P), CABC460
8 (AEROP(25), CYQP), (AEROP(26), CLRP), (AEROP(27), CN5P), CABC470
9 (AEROP(28), CYDP), (AEROP(29), CLDP), (AEROP(30), CN6P), CABC480
A (AEROP(31), CYDP), (AEROP(32), CLDP), (AEROP(33), CN7P), CABC490
B (AEROP(34), CYDS), (AEROP(35), CLDS), (AEROP(36), CN8P), CABC500

PTO=57.2958 CABC510
THETA=0 CABC520
DELALF=0)C1 CABC530
DFF=+1 CABC540
DALFAM=C0 CABC550
DOFST=C0 CABC560
DTHEST=C0 CABC570
ICNTF=C0 CABC580
FIRST=0 C0 CABC590
THINT=C0 CABC600

ALFINT=THETA CABC610
DELINT=0 CABC620

THST0 THINT CABC630

1 IF(V3-EOE,0) THST0 = TR*(COS(ADC(3,1))+COS(ADC(4,1)))/(COS(ADC(1,1)) CAB0440
1)+COS(ADC(2,1))) CABC650
VAL5=COS(ADC(3,1)) CABC660
VALA=COS(ADC(4,1)) CABC670
VAL7=COS(ADC(1,1)) CABC680
VALR=COS(ADC(2,1)) CABC690
ALFAM=ALFIN CABC700
DELTE=DELINT CABC710

Q=QH*VD+VQ*5*SW CABC720

209 THST1=THST0 DTHEST CABC730
ALFAM=ALFAM+DALFAM CABC740
DELTE1=DELTE+DDELTE CABC750
ICNTF=ICNTF+1 CABC760

IF (ICNTF GT 100) GO TO 520 CABC770

3 VAL1=ALFAM*RTD CABC780
VAL2=DELTE1*RTD CABC790
VAL3=THSTI CABC800

CALL EQU(ALFAM,DELTE1,THSTI,F1,GO,H2,FIRST) CABC810
IF (VQ*NE.3 OR F1 ST NE.0) GO TO 2 CABC820
FIRST=1 CABC830
GO TO 1 CABC840

2 IF(FIRST NE.1) FIRST=1 CABC850

C COMPUTES PARTIALS CABC860
ALFAM=ALFAM+DALFAM*G CABC870
CALL EQU(ALFAM,DELTE1,THST1,F1,G1,H111) CABC880
ALFAM=ALFAM+DALFAM CABC890

CALL EQU(ALFAM,DELTE1,THST1,F2,G2,H211) CABC900
ALFAM=ALFAM+DALFAM*G CABC910

GO TO 1 CABC920
TTH£=951HE®TI.£6E=2C1100 TO A2 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
TTH£=951HE®TI.£6E=2C1100 TO A2 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.

GALFWD=(~11-G21)/DELE®E=2C11AL¥1M1-CC25=SI NA.L1M11)
G0LE®D=051,0L;FI= 0SI AL¥1M1-CC25=SI NA.L1M11)
G0LE®D=051,0L;FI= 0SI AL¥1M1-CC25=SI NA.L1M11)
G0LE®D=051,0L;FI= 0SI AL¥1M1-CC25=SI NA.L1M11)
G0LE®D=051,0L;FI= 0SI AL¥1M1-CC25=SI NA.L1M11)

T105 GO TS 1107 IF (ABS(A1CZ1.1)<1L.€11.€0 TO 1057
T105 GO TS 1107 IF (ABS(A1CZ1.1)<1L.€11.€0 TO 1057
T105 GO TS 1107 IF (ABS(A1CZ1.1)<1L.€11.€0 TO 1057
T105 GO TS 1107 IF (ABS(A1CZ1.1)<1L.€11.€0 TO 1057
T105 GO TS 1107 IF (ABS(A1CZ1.1)<1L.€11.€0 TO 1057

T27 WRITE(I1M.521) 42 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
T27 WRITE(I1M.521) 42 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
T27 WRITE(I1M.521) 42 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
T27 WRITE(I1M.521) 42 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
T27 WRITE(I1M.521) 42 CALL EQU.1AL FAM1.DELTE11.THE1ST1.FC.G0M0.H0.11.
WRITE(IW,525)I27,XLTH(I27),(ANG(I27,IZK),ARM(I27,IZK)),IZK=-1,3)
529 FORMAT(* CABLE GEOMETRY-CABLE NO.*I2.*X.*CABLE LENGTH*E15.6,1*
     1* IN.*,3X,* DIR.* COS=*DEG ARM-IN=*/(3(3X,E15.5/)))/)
524 CONTINUE
I(V Eq.0..)*WRITE(IW,529)
529 FORMAT(* COMPUTATION OF WIND OFF CONDITION,TRIM ROUTINE NOT USED*CAB
     1* WRITE(IW,526)ICHT,ACCZ,ACCC,THEOT
526 FORMAT(* ITERATION PARAMETER =*.15./.2X.*ACCZ =*.E15.6
     1/.2X.*ACCC =*.E15.6.* 2X.*THEOT =*.E15.6.* RAD/SEC)
528 WRITE(IW,527)THEOT,DELTA,TF,TR
527 FORMAT(*,1VEH. ATT.*DELTN,5 CABLE TENSION*/)
2.*FCT CAB. TENSION =*.E15.6.* LRS*/
32X.* TF CAB. TENSION =*.E15.6.* LBS*)
RETURN
C DEbjerg UNIT.(3),INIT(VAL1,VAL2,VAL3,FI,GH,HI)
C IFAL=0,GA=1,EGER=HALF=3,EDELE=MG,HEDELE,DAF
C 2*ST3,GTST3,HST3,DALAE,WDELTE,DTHST3
C 3ACCC,ACCC,THEOT,TF,VALS,VAL5,VAL7,VAL9)
END
SUBROUTINE EQUIV(THETA,DE                                      }
C CABLE SUSPENSION SYSTEM TRIM EQUATIONS
C COMMON /INPUT/WICC
C COMMON /DATA/ AERO(175),AERO(50),KODE(26),LL
C COMMON /F_YCHA,XXM2,XXM3,XXM4,XXM5,REAL,9
C COMMON /ABC/1553,AERO(1),CMU
C EQUIVALENCE(AERO(1),CMU),(AERO(2),CLU),(AERO(3),CMU)
C 1 (AERO(4),CDA),(AERO(5),CLA),(AERO(6),CMA)
C 2 (AERO(7),CDA),(AERO(8),CLA),(AERO(9),CMA)
C 3 (AERO(10),CDA),(AERO(11),CLA),(AERO(12),CMA)
C 4 (AERO(13),CDA),(AERO(14),CLA),(AERO(15),CMA)
C 5 (AERO(16),CDA),(AERO(17),CLA),(AERO(18),CMA)
C 6 (AERO(19),CDA),(AERO(20),CLA),(AERO(21),CMA)
C 7 (AERO(22),CDA),(AERO(23),CLA),(AERO(24),CMA)
C 8 (AERO(25),CDA),(AERO(26),CLA),(AERO(27),CMA)
C 9 (AERO(28),CDA),(AERO(29),CLA),(AERO(30),CMA)
C A (AERO(31),CDA),(AERO(32),CLA),(AERO(33),CMA)
C B ( AERO(34),CDA),(AERO(35),CLA),(AERO(36),CMA)
C EQUIVALENCE(AERO(46),XCG),(AERO(47),ZCG)
C EQUIVALENCE(AERO(43),XACH),(AERO(49),XVO),(AERO(50),AM)
C EQUIVALENCE(AERO(51),PO),(AERO(52),W),(AERO(53),B)
C EQUIVALENCE(AERO(54),CBAR),(AERO(55),SW),(AERO(56),XIX)
C EQUIVALENCE(AERO(57),XIXY),(AERO(58),YIY),(AERO(59),ZIY)
C EQUIVALENCE(AERO(60),CLT),(AERO(61),COT),(AERO(62),CUT)
C EQUIVALENCE(AERO(63),MLU),(AERO(67),MLF),(AERO(68),MLH)
C 1 (AERO(69),WLH),(AERO(70),WLH),(AERO(71),WLH)
C 2 (AERO(72),STAF),(AERO(73),STAF),(AERO(74),HLH)
C 3 (AERO(75),BLLH),(AERO(76),MLCH),(AERO(77),STAC)
C 4 (AERO(78),BDP),(AERO(79),EF),(AERO(80),EF)
C 5 (AERO(81),AF),(AERO(82),AG),(AERO(83),HUCF)
C 6 (AERO(84),HLCF),(AERO(85),HUCF),(AERO(86),HLCH)
C 7 (AERO(87),DFC),(AERO(88),DCR),(AERO(89),ALF)
C 8 (AERO(90),CFC),(AERO(91),CHFC),(AERO(92),CVC)
C 9 (AERO(93),CVC),(AERO(94),CVC),(AERO(95),AKC)
A (AERO(96),ALFC),(AERO(97),STLC),(AERO(98),WLLT)
C 55
13 CONTINUE

508 CALL SNTRY (FXSN, FZSN, EMSN, THETA)
       IF (FIRST. NE. 0) GO TO 510
       IF (KODE. EQ. 4) GO TO 512
       WRITE (IO, 509) YNVI, YNVM

509 FORMAT (* CABLE CONFIGURATION ON MODEL */ /, 
       * FRONT CABLE IS *, A9, *, AND REAR CABLE IS *, A9) 

510 EL = XLSGN (3) * XLSGN (4)
       TR = TAF + AK (* EL - EL0)
       ELRDT = 0. * SW (* CLD + CMA * THETA + CLOF * DE)
       ADRA = 0. * SW (* CMA + CD * THETA + CDSE * DE)
       FXAIF = ADRAG * COS (THETA) + ELIFT * SIN (THETA)
       FZAIF = ADRAG * SIN (THETA) - ELIFT * COS (THETA)
       WGF = 32.2 * AN * SIN (THETA)
       WGT = T2.2 * AN * COS (THETA)
       ARMGF = (ZCG * WGF - XCG * GTZ) / 12.
       FZCIF = TF (* COS (ADC (3.1)) + COS (ADC (4.1)))
       FZCIF = TF (* COS (ADC (3.3)) + COS (ADC (4.3)))
       FZCIF = TF (* COS (ADC (1.1)) + COS (ADC (2.1)))
       FZCIF = TF (* COS (ADC (1.3)) + COS (ADC (2.3)))
       EMOC = *
       DC 511 I = 1, 4
       TENS = TF
       IF (I. EQ. 2) TENS = TP
       EMOC = EMOC + TENS (* (COS (ADC (I.1)) * ARM (I.3) - COS (ADC (I.3)) * ARM (I.1))

511 CONTINUE
       EMOC = EMOC / 12.
       ARMG = EMOC (+ BAR * CM * CO + CM + CM * THETA + CMDE * DE)
       FFP = ZCFH + YFCE + FZL TR + FXLS + WGF + FZAT
       GGE = XCFH + XFCF + FXT3 + FXSN + GTX + FXAIF
       H4 = EMOC + YMLFT + EYSN + EMGT * AERF
       RETURN
       END

SUBROUTINE FLYV (STAV, KLU, WLU, NHU, HHL, EP, PAD, THETA, IF)
COMMON / DAT/ AERO (175), AERO (5), KOSE (26), LL
COMMON / LCHA/ TP, XLSGN (5), TCC (5.3), ARM (5.3), TR, TLET, TF
EQUIVALENCE (AERO (76), XLCF), (AERO (77), STACF), (AERO (78), 3LCF)
PI = 3.14159

33 GAMU = ATAN (NHU / EP)
       T1 = EP + EP + HHU + HNU
       T2 = THETA + GAMU
       IF (IF. EQ. 3) T2 = GAMU - THETA
       WLUC = WLC + SORT (T1) * SIN (T2)
       T3 = WLU - WLUC
       T4 = ABS (STACF - STAV) - SORT (T1) * COS (T2)
       XLP = SORT (T3 + T4 + T4)
       XLU = SORT (XLUP + XLP - RAD * RAD)
       RUP = ATAN (T3 / T4)
       DRTU = ATAN (PAD / XLU)
       T5 = EP + EP + HHL + HHL
       T6 = THETA + GAMU
       IF (IF. EQ. 3) T6 = (THETA + GAMU)
FILE  CABLE  FORTRAN T1  GRUMMAN  DATA  SYSTEMS

WLLC = WLC - SORT(T5) * SIN(T6)
T7 = WLLC - WLL
T9 = ARG(STAC - STAV) - SORT(T5) * COS(T6)
XLLP = SORT(T7 * T7 + T8 * T8)
XLL = SORT(XLLP * XLLP - RAD * RAD)
BLP = ATAN(T7 / T8)
DRL = ATAN(RAD / XLL)
BETAL = (BLP - DRL) * FTD
IF (IF .EQ. 0.1) GO TO 1
XLGTH(3) = XLU
XLGTH(4) = XLL
ADC(3,1) = BETAU / FTD * THETA * PI
ADC(3,2) = D1/2
ADC(3,3) = D1/2 * ADC(3,1)
ADC(4,1) = D1/2 * (3FTAL / RTO - THETA)
ADC(4,2) = D1/2
ADC(4,3) = D1/2 * ADC(4,1)
ARM(3,1) = EP * RAD * SIN(ADC(3,1))
ARM(3,2) = EP
ARM(3,3) = MNU * RAD * COS(ADC(3,1))
ARM(4,1) = EP * RAD * SIN(ADC(4,1))
ARM(4,2) = EP
ARM(4,3) = MNU * RAD * COS(ADC(4,1))
RETURN

1 XLGTH(1) = XLU
XLGTH(2) = XLL
ADC(1,1) = BETAU / FTD * THETA
ADC(1,2) = D1/2
ADC(1,3) = D1/2 * ADC(1,1)
ADC(2,1) = BETAL / FTD * THETA
ADC(2,2) = D1/2
ADC(2,3) = D1/2 * ADC(2,1)
ARM(1,1) = EP * RAD * SIN(ADC(1,1))
ARM(1,2) = EP
ARM(1,3) = MNU * RAD * COS(ADC(1,1))
ARM(2,1) = EP * RAD * SIN(ADC(2,1))
ARM(2,2) = EP
ARM(2,3) = MNU * RAD * COS(ADC(2,1))
RETURN

END
SURFOUTINE RPLYH(STAC, BLK, WLD, XP, YP, ZP, RAD, THETA, IF)
COMMON /DATA/AERO(175), AERCP(50), KDEE(26), LL
COMMON /PLYCHA/PTO, XLGTH(5), ADC(5,3), ARM(5,3), TP, TLEF, TF
EQUIVALENCE (AERO(76), WLCF), (AECO(77), STAC), (AERO(78), BLCF)
PI = 3.14159
XWT = STACP - STAD
ZWT = WLCF - WLD
XP = XWT * COS(THETA) - ZWT * SIN(THETA)
ZP = XWT * SIN(THETA) + ZWT * COS(THETA)
T9 = RAD - YP
TIC = X3 - XP
XLLIP = SORT(T9 * T9 + TIC * TIC)
BHLP = ATAN2(T9, T10)
XLLH = SORT(XLLIP * XLLIP - RAD * RAD)
DBHI = ATAN(RAD / XLLH)
FILE: CABLE FOR TRAN T1

GRUMMAN DATA SYSTEMS

RHI = 8HP - 09HI
T11 = 29-2P
XL = SQRT(XLHI*XLHI + T11*T11)
TH1 = T1C = AD - COS(8HI)

TH1 = T2 - AD * SIN(BHI)

IF (IF <= 60.316) GO TO 3
XLGTH(1) = XL
XLGTH(2) = XL

ADC(1,1) = ARCOS(TH1/XL)
ADC(1,2) = ARCOS(TH2/XL)
ADC(1,3) = ARCOS(TH3/XL)

ADC(2,1) = ADC(1,1)
ADC(2,2) = DI - ADC(1,2)
ADC(2,3) = ADC(1,3)

ARM(1,1) = XP - AD*SIN(BHI)
ARM(1,2) = VP - AD*COS(BHI)

ARM(1,3) = C*

ARM(2,1) = ARM(1,1)
ARM(2,2) = ARM(1,2)
ARM(2,3) = C*

RETURN

3

XLGTH(3) = XL
XLGTH(4) = XL

ADC(3,1) = ARCOS(TH1/XL)
ADC(3,2) = ARCOS(TH2/XL)
ADC(3,3) = ARCOS(TH3/XL)

ADC(4,1) = ADC(3,1)
ADC(4,2) = DPI - ADC(3,2)
ADC(4,3) = ADC(3,3)

ARM(3,1) = XP - AD*SIN(BHI)
ARM(3,2) = VP - AD*COS(BHI)

ARM(3,3) = C*

ARM(4,1) = ARM(3,1)
ARM(4,2) = ARM(3,2)
ARM(4,3) = C*

RETURN

END

SUBROUTINE DLGTH(C1,C2,C3,IC,IDX)

COMMON/PLYCHART,TLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
1 IF (1<=C1) GO TO 2
C1 = ADC(1C,1)
C2 = C3 = ARM(1C,1)*COS(ADC(1C,3)) - ARM(1C,3)*COS(ADC(1C,1)) / 12.
RETURN

2 C1 = COS(ADC(1C,2))
C2 = (ARM(1C,2)*COS(ADC(1C,1)) - ARM(1C,1)*COS(ADC(1C,2)) / 12.
C3 = (ARM(1C,3)*COS(ADC(1C,2)) - ARM(1C,2)*COS(ADC(1C,3)) / 12.
RETURN

END

SUBROUTINE DCSLG(LIC,CX1,CZ1,C1,CT1,CX3,CZ3,CT3)

COMMON/PLYCHART,TLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
C1 = SIN(ARM(1C,1))/XLGTH(1C)*12.

IF (ADC(1C,1) - 3.14159) GT 001) GO TO 2
FILE CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

FILE: CABLE FORTRAN T1

DIMENSION CMAT(14,14,3),SMAT(14,3)

COMPLEX P0005(14,3)
COMMON/SNUD/SN(3,3),SN(3,3),THUSN,THSN,SNUD(3,3)

COMMON/DEGAUSS(3,3)

DIMENSION F(XS(3,4))

DO 10 J=1,3

XVAL=100.
GO TO 1
2 XVAL=COTAN(ADC(1:3))
1 C1=-COS(ADC(1:3))*COTAN(ADC(1:3))/XLAGH(1:3)*12.
2 XW=COS(1:3)
3 XW=ARM(1:3)
1 C1=(7*T*XW)*COTAN(ADC(1:3))/XLAGH(1:3)
1/XLAGH(1:3)
RETURN

END

C THIS IS A DOUBLE PRECISION VERSION OF CABLE4 TO BE USED
C WITH THE LFC MATRIX REDUCTION AND IBM FOOT
C FINDING ROUTINE

SUBROUTINE LONG

COMMON/NOUT/W,T

COMMON/DA/6,AE00(175),AE00(50),KODE(25),LL

COMMON/P,XYCH/AD,XLAGH(5),ADC(3,3),AR4(3,3),TP,TLFT,TF

COMMON/FOU/XYDUM(10,10)

COMMON/FO2/+C4(30)

EQUIVALENCE(AE00(46),XG),(AE00(47),ZG)

EQUIVALENCE(AE00(117),TUBENO),(AE00(119),AKSNU),(AE00(120),AKSNU)

EQUIVALENCE(AE00(123),AKS),(AE00(124),AKS),(AE00(125),AKS)

EQUIVALENCE(AE00(131),AKS3T),(AE00(132),AKS),(AE00(133),AKS)

1 (AE00(129),T3THE),(AE00(130),T3THE)

2 (AE00(131),T3THE),(AE00(132),T3THE)

EQUIVALENCE(AE00(131),AKS3T),(AE00(132),AKS),(AE00(133),AKS)

1 (AE00(129),T3THE),(AE00(130),T3THE)

2 (AE00(131),T3THE),(AE00(132),T3THE)

EQUIVALENCE(AE00(131),AKS3T),(AE00(132),AKS),(AE00(133),AKS)

1 (AE00(129),T3THE),(AE00(130),T3THE)

2 (AE00(131),T3THE),(AE00(132),T3THE)

EQUIVALENCE(AE00(131),AKS3T),(AE00(132),AKS),(AE00(133),AKS)

1 (AE00(129),T3THE),(AE00(130),T3THE)

2 (AE00(131),T3THE),(AE00(132),T3THE)

EQUIVALENCE(AE00(131),AKS3T),(AE00(132),AKS),(AE00(133),AKS)

1 (AE00(129),T3THE),(AE00(130),T3THE)

2 (AE00(131),T3THE),(AE00(132),T3THE)
FILE: CAL E FORTRAN TI

GRUMMAN DATA SYSTEMS

DO 1C K=1,4
   IF(XS(J,K)=5)
   DO 1 IC=1,5
   DO 3 J=1,10
   DO 3 K=1,10
   3 DUM(J,K)=
   IF(K<OE(11),EQ,3)GO TO 649
   TENS=TF
   IF(IC.GT.2) TENS=TC
   IF(IC.GT.4) TENS=TLFT
   DUM(1,2)= TENS*Cos(ADC(IC,3))
   DUM(1,5)= TENS*SIN(ADC(IC,1))
   DUM(2,2)= TENS*Cos(ADC(IC,1))
   DUM(2,6)= TENS*SIN(ADC(IC,3))
   DUM(3,2)= (ARM(IC,3)*DUM(1,2)-ARM(IC,1)*DUM(2,2))/12.
   DUM(3,5)= ARM(IC,3)*DUM(1,5)/12.
   DUM(2,6)= ARM(IC,1)*DUM(2,6)/12.
   IF(IC.GT.2)GO TO 2
   DUM(1,3)=Cos(ADC(IC,1))
   DUM(2,3)=Cos(ADC(IC,3))
   DUM(3,3)= (ARM(IC,3)*DUM(1,3)-ARM(IC,1)*DUM(2,3))/12.
   CALL DLGTH(CX,CZ,CT,1,0)
   CALL DLGTH(CXP,CZP,CTP,2,0)
   CX= CX + CX
   XZ=(-(CZ+CZP)/CX)
   DUM(4,1)=XZ
   XT=(-(CT+CTP)/CX)
   DUM(4,2)=XT
   DUM(4,4)= 1
   CALL DCOS_G(IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(5,4),DUM(6,4),
   DUM(2,1),DUM(6,2))
   DUM(5,5)= 1
   DUM(6,5)= 1
   CALL MASH(3,6)
   DO 4 J=1,3
   DO 4 K=1,3
   FXS(J,K)=FXS(J,K)+DUM(J,K)
   GO TO 1
   2 IF(IC.GT.4) GO TO 5
   CALL DLGTH(CX,CZ,CT,3,0)
   CALL DLGTH(CXP,CZP,CTP,4,0)
   DUM(7,1)=CZ+CZP
   DUM(7,2)=CT+CTP
   DUM(7,3)=CX+CXP
   DUM(4,7)=AK*12.
   8 DUM(1,4)=Cos(ADC(IC,1))
   DUM(2,4)=Cos(ADC(IC,3))
   DUM(1,4)= (ARM(IC,3)*DUM(1,4)-ARM(IC,1)*DUM(2,4))/12.
   CALL DCOS_G(IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(5,3),DUM(6,3),DUM(6,4),DUM(6,5),
   DUM(4,4)= 1
   DUM(5,5)= 1
   DUM(6,5)= 1
   DUM(7,7)= 1
   CALL MASH(3,7)

61
DO 6 J = 1, 3
DO 6 K = 1, 3
IF (K, NE, 3) FXS(J, K) = FXS(J, K) + DUM(J, K)
6 IF (K, EQ, 3) FXS(J, 4) = FXS(J, 4) + DUM(J, K)
GO TO 1
5 IF (KODE(11), EQ, 0) GO TO 1
CALL DLGTH(DUM(7, 3) + DUM(7, 2) + DUM(7, 1), 0)
DUM(4, 7) = AKLFT + 12.
GO TO 8
1 CONTINUE
C ADD SNUPPER INCREMENTS
CALL LONSN
DO 7 J = 1, 3
FXS(J, 1) = FXS(J, 1) + SNU(J, 2)
7 FXS(J, 4) = FXS(J, 4) + SNU(J, 1)
CALL FICT(C)
C ZERO CABLE EFFECTS FOR CABLELESS MODEL CHAR.
IF (KODE(13), NE, -1.) GO TO 649
DO 84 J = 1, 3
DO 84 K = 1, 4
84 FXS(J, K) = 0.
DO 85 J = 1, 3
DO 85 K = 1, 6
85 FRIC(J, K) = 0.
DO 86 J = 1, 3
DO 86 K = 1, 3
86 SNUD(J, K) = 0.
C THE CABLE FORCES/MOMENTS PARTIALS ARE COMPLETED
C AEDG. DATA IS NOW COMPUTED
649 Q = SQRT*VQ*VQ/2.
QS = Q*S
IF (VQ, NE, *) QSV = QS/VQ
IF (VQ, LE, 0.0) QSV = 0.
XU = CXU% = 0.
ZJ = CZUP% = 0.
SNV = CMUP% = S% = CB%.
X% = CXAP% = 0.
Z% = CZAP% = 0.
EM% = CMAP% = QSV% = CB%.
IF (VQ, NE, *) X0 = CXOF% = QSV% = CB% = (VQ*2.).
IF (VQ, LE, 0.0) X0 = 0.
IF (VQ, NE, *) Z0 = CZOF% = QSV% = CB% = (VQ*2.).
IF (VQ, LE, 0.0) Z0 = 0.
EQ = CMOP% = QSV% = CB% = 2.
XE = CXDEP% = 0.
ZE = CZDEP% = 0.
EM% = CMDEP% = QSV% = CB%.
IF (VQ, NE, *) XAD = CXAD% = QSV% = CB% = (VQ*2.).
IF (VQ, LE, 0.0) XAD = 0.
IF (VQ, NE, *) ZAD = CZAD% = QSV% = CB% = (VQ*2.).
IF (VQ, LE, 0.0) ZAD = 0.
EM% = CMAD% = QSV% = CB% = (VQ*2).
IF (VQ, LE, 0.0) EMAD = 0.
IF (VQ = 14

62
ICOL=14
ICORDER=3
42 DO 20 I=1,1600
DO 20 J=1,ICOL
DO 20 K=1,ICORDER
20 CMAT(I,J,K)=6.0
IF(KCOME(10).EQ.3)GO TO 650
C FX EQUATION
CMAT(1,1,1)=-FXS(1,1)
CMAT(1,1,2)=-ZC*SNUD(1,2)-FRIC(1,5)-FRIC(1,2)
CMAT(1,1,3)=-XAD
CMAT(1,2,1)=-FXS(1,2)+W*T*COS(THETA)-ZAD
CMAT(1,2,2)=-FXS(2,2)-W*T*SIN(THETA)-ZAD
CMAT(1,2,3)=ZCG*A/12.
CMAT(1,3,1)=-FXS(1,3)
CMAT(1,3,2)=FXS(1,4)
CMAT(1,4,2)=XU*SNUD(1,1)-FRIC(1,4)-FRIC(1,1)
CMAT(1,4,3)=AV
CMAT(1,5,1)=-XDE
C FY EQUATION
CMAT(2,1,1)=-FXS(2,1)
CMAT(2,1,2)=-7A*SNUD(2,2)-FRIC(2,5)-FRIC(2,2)
CMAT(2,1,3)=AV-ZAD
CMAT(2,2,1)=-FXS(2,2)+W*T*SIN(THETA)-ZAD
CMAT(2,2,2)=Z0-7A*D*V0*SNUD(2,3)-FRIC(2,6)-FRIC(2,3)
CMAT(2,2,3)=XCG*A/12.
CMAT(2,3,1)=-FXS(2,3)
CMAT(2,4,1)=-FXS(2,4)
CMAT(2,4,2)=ZU*SNUD(2,1)-FRIC(2,4)-FRIC(2,1)
CMAT(2,5,1)=ZCF
C MOMENT EQUATION
CMAT(3,1,1)=-FXS(3,1)
CMAT(3,1,2)=-7A*SNUD(3,2)-FRIC(3,5)-FRIC(3,2)
CMAT(3,1,3)=AV-XCG*A/12.
CMAT(3,2,1)=-FXS(3,2)+W*T*SIN(THETA)\n\n1-XCG*A/W*SIN(THETA)/12.
CMAT(3,3,1)=-FXS(3,3)
CMAT(3,4,1)=-FXS(3,4)
CMAT(3,4,2)=-EMU*SNUD(3,1)-FRIC(3,4)-FRIC(3,1)
CMAT(3,5,1)=EMDF
C ELIMINATION OF DTF COL FOR CABLELESS MODEL CHA.
IF(KCOME(10).NE.-1.)GO TO 81
IF(KCOME(11).NE.-1.)IFTE(1,J,W2)
82 FORMAT(5X,KODE(9)HAS BEEN SET BY PFG. TO 3, FOR CABLELESS MODE)
    1L CHARACTERISTICS)
    KODE(9)=3.
    DO 93 I=1,3
    DO 93 J=1,3
93 CMAT(I,3,K)=CMAT(I,4,K)
GO TO 91
C CONSTRAINT EQUATION
A1 CMAT(4,1,1)=XQZ
FILE: CABLE  FORTRAN T1  

C ACTIVB CABLE CONTROL Eqs.

IF (KODE(13) .LE. 0) GO TO 30
CMAT(1,5,1)=0.0
CMAT(2,5,1)=0.0
CMAT(3,5,1)=0.0
I=(KODE(6),EO.1,DF,KODE(6),EO.3) GC TU 46
IC2=4
IC1=3
GO TO 47
46 IC2=1
IC1=2
47 CMAT(1,1,1)=-(COS(ADC(IIC2,1)) - COS(ADC(IIC1,1)))
CMAT(2,1,1)=-(COS(ADC(IIC2,3)) - COS(ADC(IIC1,3)))
CMAT(3,1,1)=(ADC(IIC2,3)*COS(ADC(IIC2,1)) - ADC(IIC1,3)*COS(ADC(IIC1,3))) /12
C  EQ OF MOTOR DYN.
CMAT(5,5,1)=2.0*E*DF*PSBA
CMAT(5,5,2)=2.0*E*DF*ELSBA
CMAT(5,7,1)=AKSRT*2.
CMAT(5,6,2)=AKSRT*2.*AKSRV-OMP*PSBA
CMAT(5,6,3)=AJASM*FSA-FC*ELSBA
CMAT(5,8,3)=AJASM=ELS16
C  EQ RELATING PULLEY ROTATION TO SYS. GEOM., MOTOR ON TOP
CALL DLGTH(CMAT(6,4,1),CMAT(6,1,1),CMAT(6,2,1),IC1,1)
CMAT(6,6,3)=-C20/12
C  ACTIVB CABLE FEEDBACK Eq.
CMAT(7,2,2)=AK0
CMAT(7,6,1)=AKTH
CMAT(7,6,2)=AKTH
CMAT(7,9,1)=1.0
C  TOTAL VOLTAGE EQ EM + ENC
CMAT(9,7,1)=1.0
CMAT(9,9,1)=1.0
CMAT(9,11,1)=1.0
C  RELATION OF THM TO THMD
CMAT(8,1,1)=1.0
CMAT(8,6,2)=1.0
C  RELATION OF TFOBK TO B1C AND INPUT DT
CMAT(10,5,1)=1.0
CMAT(10,10,1)=1.0
CMAT(12,12,1)=1.0
GO TO 31
C  FEEDBACK LOOP EQUATION
30 CMAT(5,5,2)=AKTHF
CMAT(5,6,2)=THF
CMAT(5,6,1)=1.0
31 IF(IHEC)
IF(KODE(14),EO.4) GO TO 32
C SURST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS
IDX=KODE(14)
IDN=KODE(15)
FILE: CABLE FORTRAN 1

GRUMMAN DATA SYSTEMS

IF (IDN NE 13) GO TO 52
IDN = 2
I*HO = 13

52 IF (IDX .GT. 14) GO TO 38
D0 34 I = 1, 14
D0 34 K = 1, 3
NMAT (1, K) = CMAT (1, IDN, K)

34 CMAT (1, IDN, K) = CMAT (1, IDX, K)
GO TO 32

38 D0 37 I = 1, 14
D0 37 K = 1, 3
NMAT (1, K) = CMAT (1, IDN, K)

37 CMAT (1, IDN, K) = CMAT (1, IDN, K)
IF (IDN .EQ. 16) GO TO 39
CMAT (1, IDN, 1) = XDE
CMAT (2, IDN, 1) = YDE
CMAT (3, IDN, 1) = ZDE
GO TO 32

39 CMAT (1, IDN, 1) = XA
CMAT (2, IDN, 1) = YA
CMAT (3, IDN, 1) = ZA

N = CODE(R)

65 CALL MATRIX (CMAT, N, ROOTS, K4A, IEP)

IF (KODE(14).NE.0) GO TO 35
D0 36 I = 1, 14
D0 36 K = 1, 3

36 CMAT (1, IDN, K) = CMAT (1, K)

C 35 IF (KODE(5) .NE. 0) WRITE (IW, 100) IEP
C此为输出条件，当I=13时，输出I=13, 3x, X见程序说明
C THE ROOTS OF THE CHC, CMAT, ARE IN THE COMPLEX ARRAY *ROOTS*
C AND THE NUMBER OF ROOTS IS *K4A*

35 K4A = K4A -

1 IF (I+HDN .GT. 13) GO TO 70
K4A = K4A + 1

70 CALL PRINTR (IW, ROOTS, K4A)
GO TO 651

650 CONTINUE

C NEW SNOWER EFFECTS
KIDE (14) = 0
D0 600 IC = 1, 4
D0 201 I = 1, 10
D0 201 J = 1, 10

201 DUM (1, J) = 0
TC = T - T + T SNQ
IF (I+GT. 2) TC = T SNQ

DUM (1, 3) = - TC * COS (ADG (1, 3, 1))
DUM (1, 4) = - TC * SIN (ADG (1, 3, 1))
DUM (1, 6) = COS (ADG (1, 3, 1))
DUM (2, 4) = TC * COS (ADG (1, 3, 1))
DUM (2, 5) = TC * SIN (ADG (1, 3, 1))
DUM (3, 5) = TC * COS (ADG (1, 3, 1))
DUM(2,6)=COS(ARC(1C,3))
DUM(3,3)=ARCCOS(DUM(1,3)-ARC(1C,1))=DUM(2,3))/12.
DUM(3,4)=ARCOS(ARC(1C,3))=DUM(2,4)/12.
DUM(3,5)=ARCOS(ARC(1C,1))=DUM(2,5)/12.
DUM(3,6)=ARCOS(ARC(1C,3))=DUM(2,6))/12.
CALL DCOSLS(ARC, DUM(4,1), DUM(4,2), DUM(4,3), DUM(5,1), DUM(5,2).
1 DUM(5,3)
DUM(4,4)=1,
DUM(4,5)=1,
DUM(6,6)=1,
DUM(6,7)=AKSNUM*12.
IF(1C.GT.2) DUM(6,7)=AKSNUM*12.
CALL DLGTH(DUM(7,1), DUM(7,2), DUM(7,3), 1C, 0)
DUM(7,7)=1,
CALL VIAH(3,7)
D0 20 ? J=1, 3
D0 20 ? K=1, 3
200 FXS(J,K)=FXS(J,K)+DUM(J,K)
600 CONTINUE
CMAT(1,2,2)=XX
CMAT(1,2,3)=XAD
CMAT(1,3,1)=XT*COS(THETA)-XX*VO
CMAT(1,3,2)=XO-XAD*VO
CMAT(1,3,3)=XG*AM/12.
CMAT(1,1,2)=XU
CMAT(1,1,3)=AM
CMAT(2,2,2)=ZA
CMAT(2,2,3)=AM-ZAD
CMAT(2,3,1)=XT*SIN(THETA)-ZAD*VO
CMAT(2,3,2)=-ZD-ZAD*VO
CMAT(2,3,3)=-XG*AM/12.
CMAT(2,1,2)=ZU
CMAT(3,2,2)=EMA
CMAT(3,2,3)=-EVAD=CBAR-XG*AM/12.
CMAT(3,3,1)=-EAMA+VO*XG*T*COS(THETA)/12.-XG*XT*SIN(THETA)/12.
CMAT(3,3,2)=-(EVAD+VO)*CBAR
CMAT(3,3,3)=YY
CMAT(3,1,2)=FNU
CMAT(3,1,3)=XG*AM/12.
D0 700 I=1, 3
D0 700 J=1, 3
700 CMAT(1,1,1)=CMAT(I,J,1)-FXS(I,J)
I=W
N=3
GO TO 655
651 CONTINUE
IF(KODE(3).NE.2)RETURN
IF(KODE(14).EQ.0) GO TO 41
WRITE(1W,43)
43 FORMAT(/' COMPUTATION OF THE DENOMINATOR ROOTS'//) LKODE=KODE(14)
KODE(14)=0
CALL FREN(FOO, KAM, CA(KAM+1))
GO TO 42
41 KODE(14)=LKODE
CALL FREQ (ROOTS,K4A,K4A+1))
RETURN
END

SUBROUTINE PRINT (LOUT,ST,NROOT)
COMMON/PRD/C4(30)
DIMENSION ST(2,1)
K4=NROOT+1
WRITE (LOUT,11) (C4(I), I=1,K4)
1 FORMAT (* POLYNOMIAL W/CONST TERM FIRST, */E27.6,A16.6))
COMMENT PRINTS PERTINENT INFORMATION ABOUT CHARACTERISTIC ROOTS
WRITE (LOUT,507)
507 FORMAT (* REAL IMAGINARY T H/D-SEC 1/T H/D, */CABC37)
1 1D PERIOD-SEC DNAT-CPS UNDNAT-CPS DAMP */CABC27
2 *RATIO DECAY RATIO */CABC27
NEXT=1
IF (NROOT.GT.0) GO TO 5
WRITE (LOUT,2)
RETURN
5 IF (STC=I=1,N=00)
IF (NEXT.EQ.2) GO TO 777
SIG=ST(1,11)
ASIG=AS(SIG)
AW=ABS(RT(2,1))
THDI=ASIG*1.442695
THD=99999.
IF (THDI.GT.1.E-5) THD=1./THDI
IF (AW.EQ.0.) GO TO 531
NEXT=2
WD=-AWD
DNAT=AWD * 159155
PER=99999.
IF (DNAT.GT.1.E-6) PER=1./DNAT
UNDNAT=SIG*T (ASIG**2+AWD**2) * 1591550
DAMP=0
IF (AWD - 1.E15 * ASIG ) 503,504,505
503 DAMPS=SIGN ( COS ( ATAN ( AWD/ASIG ) ) ) * -SIG
504 CHDI=THDI*PER
DECR=99999.
APR=SIG * PER
IF (APR.GT.174.6) DEC=EXP (APR)
WRITE (LOUT,529) SIG,WD,THC,THDI,PER,DNAT,UNDNAT,DAMP,DEC
529 FORMAT (E12.4,A2X,1H+,E11.4,A13.4)
GO TO 530
531 WRITE (LOUT,532) SIG,THC,THDI
532 FORMAT (E12.4,A14X,2E13.4)
GO TO 530
777 VFX=1
530 CONTINUE
RETURN
END

SUBROUTINE MASK (NN,N)
COMMON /DUP/DUM(10,10)
C NN = FINAL MATRIX SIZE
C N = ORIGINAL MATRIX SIZE
FILE: CABLE FORTRAN T1  GRUMMAN DATA SYSTEMS

INN=NN
D0 1001 LL=1.INN
L=N+1-LL
II=1
JJ=LL-1
D0 1001 I=1,II
ND 1001 J=1,JJ
1001 SUM(I,J)= SUM(I,J)+DUM(L,J)+SUM(I,II)-SUM(LL,J)
RETURN
END
SUBROUTINE LAT
COMMON/INPUT/IF,T0
COMMON /DATA/ AEFO(175), AEFO(50), KODE(26), LL
COMMON /PLYCHART/MTD, XLGTH(5), ABC(5,3), ARM(5,3), TR, TLFT, TF
COMMON /YOU/DUM(11,15)
COMMON/FRO/C4(30)
EQUIVALENCE(AEFO(46), XCG), (AEFO(47), ZCG)
EQUIVALENCE(AEFO(63), THEETA), (AEFO(49), YD), (AEFO(50), AM)
EQUIVALENCE(AEFO(51), FE), (AEFO(52), WT), (AEFO(53), B)
EQUIVALENCE(AEFO(54), CBAF), (AEFO(55), SW), (AEFO(56), XIXZ)
EQUIVALENCE(AEFO(57), XIYX), (AEFO(58), YIVY), (AEFO(59), ZIZZ)
1
(AEFO(95), AEFO(100), AEFO(100), AEFO(100))
EQUIVALENCE(AEFO(117), USYD), (AEFO(119), AKSN), (AEFO(122), AKSNL)
EQUIVALENCE(AEFO(123), AKSY), (AEFO(124), AKPHI), (AEFO(125), AKTHS)
1
(AEFO(126), AKAZ), (AEFO(127), T1SY), (AEFO(128), YDPMI)
2
(AEFO(129), T1THS), (AEFO(130), T1AZ)
EQUIVALENCE(AEFO(131), AKSAT), (AEFO(132), AKSR), (AEFO(133), AKJSN)
1
(AEFO(134), KSB), (AEFO(135), ELSBA), (AEFO(136), KPSD)
2
(AEFO(137), AKTHD), (AEFO(138), AKTH), (AEFO(139), G2DP)
3
(AEFO(140), AKO), (AEFO(141), AK7), (AEFO(142), AKPS)
4
(AEFO(143), AKY), (AEFO(144), AKV)
EQUIVALENCE(AEFO(1), CKUP), (AEFO(2), CZUP), (AEFO(3), CMUP)
1
(AEFO(4), CKAP), (AEFO(5), CZAP), (AEFO(6), CMAP)
2
(AEFO(7), CKOP), (AEFO(8), CZOP), (AEFO(9), CMOP)
3
(AEFO(10), CKDP), (AEFO(11), CZDP), (AEFO(12), CMDP)
4
(AEFO(13), CKEP), (AEFO(14), CZEP), (AEFO(15), CMEP)
5
(AEFO(16), CKPD), (AEFO(17), CZPD), (AEFO(18), CMPD)
6
(AEFO(19), CKRP), (AEFO(20), CZRP), (AEFO(21), CMRP)
7
(AEFO(22), CKDP), (AEFO(23), CZDP), (AEFO(24), CMDP)
8
(AEFO(25), CKDP), (AEFO(26), CZDP), (AEFO(27), CMDP)
9
(AEFO(28), CYDP), (AEFO(29), CDLP), (AEFO(30), CMRP)
10
(AEFO(31), CYDP), (AEFO(32), CDLP), (AEFO(33), CMRP)
11
(AEFO(34), CYDP), (AEFO(35), CDLP), (AEFO(36), CMRP)
12
DIMENSION CHAT(14,14,3), BMAT(14,3)
COMPLEX SNUR(3,3), SV(30), RUSN, TNLN, SNUR(3,3)
COMMON /FRO/GFIC(3,6)
DIMENSION FXS(3,3)
DO 10 J=1,3
DO 10 K=1,3
10 FXS(J,K)=0.
IF(KODE(11), OE, 3) GO TO 65C
DO 111 I=1,5
IF(KODE(11), OE, 3) AND, IC, EQ, 5) GO TO 1
DO 3 J=1,8
CABCC53C
CABCC54C
CABCC55C
CABCC56C
CABCC57C
CABCC58C
CABCC59C
CABCC60C
CABCC61C
CABCC62C
CABCC63C
CABCC64C
CABCC65C
CABCC66C
CABCC67C
CABCC68C
CABCC69C
CABCC70C
CABCC71C
CABCC72C
CABCC73C
CABCC74C
CABCC75C
CABCC76C
CABCC77C
CABCC78C
CABCC79C
CABCC80C
CABCC81C
CABCC82C
CABCC83C
CABCC84C
CABCC85C
CABCC86C
CABCC87C
CABCC88C
CABCC89C
CABCC90C
CABCC91C
CABCC92C
CABCC93C
CABCC94C
CABCC95C
CABCC96C
CABCC97C
CABCC98C
CABCC99C
CABCC00C
CABCC01C
CABCC02C
CABCC03C
CABCC04C
CABCC05C
CABCC06C
CABCC07C
CABCC08C
CABCC09C
CABCC10C
CABCC11C
CABCC12C
CABCC13C
CABCC14C
CABCC15C
CABCC16C
CABCC17C
68
DO 3 K=1,A
  3 DUM(J,K)=0.
  IF(IC.GT.2)TENS=TD
  IF(IC.GT.4)TENS=TLFT
  CA1=COS(ADC(IC,1))
  CA2=COS(ADC(IC,2))
  CA3=COS(ADC(IC,3))
  IF(ABS(CA1).LT..0.01) CA1=0.
  IF(ABS(CA2).LT..0.01) CA2=0.
  IF(ABS(CA3).LT..0.01) CA3=0.
  DUV(1,2)=-TENS*CA1
  DUV(1,3)=-TENS*CA3
  DUV(1,4)=CA2
  DUV(1,6)=-TENS*SIN(ADC(IC,2))
  DUV(2,2)=1. AEM(IC,1)*SUM(1,2) - AEM(IC,2)*TENS*CA2)/12.
  DUV(2,3)= AEM(IC,1)*SUM(1,3)/12.
  DUV(2,4)= AEM(IC,2)*CA2-AEM(IC,2)*CA1)/12.
  DUV(2,5)= AEM(IC,2)*TENS*SIN(ADC(IC,1))/12.
  DUV(2,6) = AEM(IC,2)*SUM(1,6)/12.
  DUV(4,4)=1.
  DUV(4,9)=0.
  IF(IC.GT.2)DUM(4,9)=AK=-12.
  IF(IC.GT.4)DUM(4,9)=AKLF=-12.
  DUV(3,2)= -AEM(IC,3)*SUM(1,2)/12.
  DUV(3,3)=-AEM(IC,3)*SUM(1,3)-AEM(IC,2)*TENS*CA2)/12.
  DUV(3,4)=AEM(IC,2)*CA3-AEM(IC,3)*CA2)/12.
  DUV(3,7)=AEM(IC,2)*TENS*SIN(ADC(IC,3))/12.
  DUV(3,6) = AEM(IC,3)*SUM(1,6)/12.
  CALL COSD(IC,CUM(5,1),CUM(5,2),CUM(5,3),CUM(6,1),CUM(6,2),CUM(1,1),CUM(7,1),CUM(7,2),CUM(7,3))
  DUV(5,5)=-1.
  DUV(6,6)=-1.
  DUV(7,7)=-1.
  IF(IC.GT.2) GO TO 2
  CALL WASH(3,7)
6  DO 4 J=1,3
5  DO 4 K=1,3
4  FXS(J,K)=FXS(J,K)+SUM(J,K)
   GO TO 1
2  IF(IC.GT.4) GO TO 5
   CALL DLGTH(CY,CSP,CPH,3,1)
   CALL DLGTH(CYP,CSPH,CPH,4,1)
   DUM(8,1)=CY+CYP
   DUM(9,2)=CSP+CPSP
   DUM(9,3)=CPH+CPH
   DUM(8,9)=-1.
   CALL WASH(3,8)
   GO TO 6
5  IF(KDOF(11).EQ.0)GO TO 1
   CALL DLGTH(DUM(9,1),DUM(9,2),DUM(9,3),5,1)
   DUM(9,8)=-1.
   CALL WASH(3,8)
   GO TO 6
1  CONTINUE
111 CONTINUE
C COMPLETE SUMMATION OF CABLE FORCES & MOMENTS
C ADD SNUBEFF INCREMENTS
112 CALL LATSN
DO 3 J=1,3
DO 8 K=1,3
3 FXS(J,K)=FXS(J,K)+SNU(J,K)
8 CALL FFICI(1)
C ZERO CABLE EFFECTS FOR CABLELESS MODEL OPTION
IF(KODE(13).NE.-1) GO TO 620
IF(KODE(9).NE.1) GOTO(1M,22)
22 FORMAT('5X*KODE(1H) HAS BEEN SET BY PROG TO 3 FOR CABLELESS MODEL CCAB174C
1HARACTERISTICS*)
KODE(9)=3
DO 20 C J=1,3
DO 20 K=1,3
20 FXS(J,K)=0.
DO 21 C J=1,3
DO 21 K=1,6
21 FFICI(J,K)=0.
G) T 3 620
650 CONTINUE
KODE(16)=0
DO 610 I=1,3
DO 610 J=1,6
610 FFICI(I,J)=0.
DO 660 IC=1,4
DO 660 I=1,10
DO 660 J=1,10
660 DUM(I,J)=0.
TC=TC-TC+TUSNO
IF(IC.GT.2) TC=TUSNO
CA1=CSS(ADC(CIC,1))
CA2=CSS(ADC(CIC,2))
CA3=CSS(ADC(CIC,3))
IF(ABS(CA1).LT.10CC1) CA1=0.
IF(ABS(CA2).LT.10CC1) CA2=0.
IF(ABS(CA3).LT.10CC1) CA3=0.
DUM(1,2)=-TC*CA1
DUM(1,3)=TC*CA2
DUM(1,4)=CA2
DUM(1,5)=-TC*CSS(CIC,2)
DUM(2,2)=AFM(CIC,1)+DUM(1,2)-AFM(CIC,2)*TC*CA2/12.
DUM(2,7)=AFM(CIC,1)+DUM(1,7)/12.
DUM(2,4)=AFM(CIC,1)*CA2-AFM(CIC,2)*CA1/12.
DUM(2,5)=AFM(CIC,2)*TC*CSS(CIC,1)/12.
DUM(2,6)=AFM(CIC,1)+DUM(1,6)/12.
DUM(3,2)=AFM(CIC,3)+DUM(1,7)/12.
DUM(3,3)=(-AFM(CIC,3)+DUM(1,3)-AFM(CIC,2)*TC*CA2)/12.
DUM(3,4)=(AFM(CIC,2)*CA2-AFM(CIC,3)*CA2)/12.
CAB0140
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

100 DUM(3,7)=-AFM(4,2)*TC*SIN(ADC(4,3))/12.
250 DUM(3,6)=-AFM(4,3)*DUM(1,6)/12.
300 DUM(4,4)=1.
350 DUM(4,5)=<SNU12.
400 IF(IC.GT.2) DUM(4,8)=AKSNL12.
450 CALL DCS(4,1).DUM(5,1).DUM(5,2).DUM(5,3).DUM(6,1).DUM(6,2).

1 DUM(4,3).DUM(7,1).DUM(7,2).DUM(7,3)

600 CONTINUE

620 QS=Q*S

IF(VDNEQ.7) Q=Q/VD

IF(VDNEQ.4) QS=0.

IF(VDNEQ.7) Q=Q/(2.*VD)

IF(VDNEQ.7) Q=Q/2.

YV=CYPQ*2EV

ELV=CLAPDSVNY

ENPV=CNAPPBOSV

YV=CYPQ*2OSV

ELAP=CLAPD87VOSV

ENP=CNAPPB7VOSV

YV=CYPQ*2OSV

FLP=CLAPDSVNY

CNP=CNAPPBOSV

YV=CYPQ*2OSV

YDR=CYPQ*25

ENDR=ENDP*Q5P

ELDP=ELDP*Q5P

YD=CYPPQ*OS

ENDA=ENDP*Q5P

ELDA=ELDA*Q5P

YDS=CYPQ*25

ENDS=ENDP*Q5P

ELDS=ELDS*Q5P

42 DO 113 I=1,4

45 DO 113 J=1,4

113 CVAT(J,4,K)=0.

C... FORCE EQUATION

400 CVAT(1,1,1)=FXS(1,1)

100 CVAT(1,1,2)=YY-SNU(1,1)-FRIC(1,4)-FRIC(1,1)

160 CVAT(1,1,3)=YY

210 CVAT(1,1,4)=FXS(1,2)+YY*VC-W*TSIN(THETA)

260 CVAT(1,2,1)=YY-SNU(1,2)-FRIC(1,5)-FRIC(1,2)

310 CVAT(1,2,2)=AM*CG/12.

360 CVAT(1,2,3)=FXS(1,3)-W*COS(THETA)

410 CVAT(1,2,4)=YY-SNU(1,3)-FRIC(1,6)-FRIC(1,3)
C MAT(1,3,1) = -AM*7CG/12.

C YAW EQUATION
C MAT(2,1,1) = -FXS(2,1)
C MAT(2,1,2) = ENV*SNU(2,1) - FRIC(2,1)
C MAT(2,1,3) = AM*XCG/12.
C MAT(2,2,1) = FXS(2,2) + ENV*V0 - XCG*WT*SN(I(THEA))/12.
C MAT(2,2,2) = ENV*SNU(2,2) - FRIC(2,2)
C MAT(2,2,3) = 2IZZ
C MAT(2,3,1) = FXS(2,3) - XCG*WT*COS(I(THEA))/12.
C MAT(2,3,2) = ENV*SNU(2,3) - FRIC(2,3)
C MAT(2,3,3) = -XIXZ

C ROLL EQUATION
C MAT(3,1,1) = -FXS(3,1)
C MAT(3,1,2) = ELV*SNU(3,1) - FRIC(3,1)
C MAT(3,1,3) = AM*7CG/12.
C MAT(3,2,1) = FXS(3,2) + ELV*V0 + XCG*WT*SN(I(THEA))/12.
C MAT(3,2,2) = ELV*SNU(3,2) - FRIC(3,2)
C MAT(3,2,3) = -XIXZ
C MAT(3,3,1) = FXS(3,3) + XCG*WT*COS(I(THEA))/12.
C MAT(3,3,2) = ELV*SNU(3,3) - FRIC(3,3)
C MAT(3,3,3) = XIXZ

C ACTIVE CABLE CONTROL EQUATIONS
IF (KODE(13) .NE. 1) GO TO 30
IF (KODE(6) .EQ. 3) = KODE(6), EQ. 4) GO TO 46
IC2 = 2
ICl = 1
GO TO 47

46 IC2 = 4
Ic1 = 7

47 CMAT(1,1,1) = (COS(ADG(1C2,2)) - COS(ADG(1C1,2)))/12
C MAT(3,1,1) = (COS(ADG(1C2,2)) - COS(ADG(1C1,2)))/12
C MAT(3,2,1) = (COS(ADG(1C2,2)) - COS(ADG(1C1,2)))/12
C MAT(3,3,1) = (COS(ADG(1C2,2)) - COS(ADG(1C1,2)))/12

C EQ. DF WHTO DYN.
C MAT(4,4,1) = +2.5*CS3*CSA
C MAT(4,4,2) = +2.5*SP0*ELSBA
C MAT(4,6,1) = +AKS3*2.
C MAT(4,5,2) = -AKS3*2.

C CALL DLGTH(CMAT(5,1,1), CMAT(5,2,1), CMAT(5,3,1), IC1, IC2)
C MAT(5,5,1) = +230/12.

C EQ. FOR TOTAL VOLTAGE = ACTIVE SYSTEM*INPUT VOLTAGE, END
C MAT(9,6,1) = 1.
C MAT(9,9,1) = 1.
C MAT(9,11,1) = 1.

C FEEDBACK CONTROL EQ.
C MAT(6,2,2) = AKP0
C MAT(6,5,1) = AXY
C MAT(6,7,1) = AXY
C MAT(6,9,1) = 1.

C RELATE ANGULAR RATES TO ANGULAR DISPLACEMENTS
CMAT(9,2,7)=1.
CMAT(8,4,1)=-1.
CMAT(7,5,2)=1.
CMAT(7,7,1)=1.

C RELATION OF DTC TO DT AND DTFB
CMAT(10,4,1)=1.
CMAT(10,10,1)=1.
CMAT(10,12,1)=1.
GO TO 31

C RUDDER FEEDBACK LOOP
30 CMAT(4,2,2)=4KSY
CMAT(4,4,2)=-T3SY
CMAT(4,4,1)=-1.

C AILERON FEEDBACK LOOP
CMAT(5,3,2)=AKPHI
CMAT(5,5,2)=T2PHI
CMAT(5,5,1)=-1.
CMAT(1,4,1)=0S*CYDEP
CMAT(1,5,1)=0S*CYDEP
CMAT(2,4,1)=2E*3*CNDEP
CMAT(2,5,1)=2E*3*CNDEP
CMAT(3,4,1)=2E*3*CNDEP
CMAT(3,5,1)=2E*3*CNDEP

31 IF(KODE(16).EQ.0)GO TO 32

C SURRTG COL IDX INTO COL IDN TO GET NUMERATOR ROOTS
IDX=KODE(16)
IDN=KODE(17)
IF(IDX.GT.13)GO TO 38
DO 34 I=1,14
DO 34 K=1,3
50 CMAT(I,K)=CMAT(I,IDN,K)
34 CMAT(I,IDN,K)=-CMAT(I,IDX,K)
GO TO 32

38 DO 37 I=1,14
DO 37 K=1,3
50 CMAT(I,K)=CMAT(I,IDN,K)
37 CMAT(I,IDN,K)=1.0
50 IF(IDX.EQ.15)GO TO 39
50 IF(IDX.EQ.16)GO TO 41
CMAT(1,IDN,1)=YDR
CMAT(2,IDN,1)=ENDF
CMAT(3,IDN,1)=LDR
GO TO 32
39 CMAT(1,IDN,1)=YD
CMAT(2,IDN,1)=ENDA
CMAT(3,IDN,1)=ELDA
GO TO 32
41 CMAT(1,IDN,1)=YV
CMAT(2,IDN,2)=ENV
CMAT(3,IDN,3)=FLV

42 IF(KODE(9))
CALL MATRIX(CMAT,N,PHOTG,KAA,IFP)
50 IF(KODE(16).EQ.0)GO TO 35
DO 36 I=1,14
DO 36 K=1,3
FILE: CABLE
FORTRAN T1

GRUMMAN DATA SYSTEMS

36 C4AT(I, IQN, K) = CMAT(I, K)
C 35 IF (KODE(5), NE, 0) WRITE(IW, I03) IEF
C 10 FORMAT(2X, F22.13, 3X, 6F15.0) FORF AND PRIM FOR ERROR CODES
C THE ROOTS OF THE CHARACTERISTIC EQUAT. ARE IN THE COMPLEX ARRAY
C ROOTS* AND THE NUMBER OF ROOTS IS *K4A*

35 K4A = K4A + 1
CALL PRINT(IW, SYLTS, K4A)
IF(KODE(3), NE, 2) RETURN
IF(KODE(16), EQ, 0) GO TO 44
WRITE(IW, 43)

43 FORMAT(/// COMPUTATION OF THE DENOMINATOR ROOTS///)
LZKE = KODE(16)
KODE(16) = C0
CALL FREQ1(FOOTS, K4A, C4(K4A + 1))
GO TO 42

44 KODE(16) = LZKK
CALL FREQ2(FOOTS, K4A, C4(K4A + 1))
RETURN
END

SUBROUTINE DCOSD(A, CY1, CPS11, CPSI1, CPSI2, CPSI1, CPSI3, CPSI3)

COMMON /PLYCHA/GT, XMT, TAF, TFC, VTfc, TFC, TFC
1 = ABS( ADC(I,I) - 3.14159) / GT * COS(1) GO TO 2
XVAL = I*IC
GO TO 1

2 XVAL = COTAN( ADC(I, 3))
1 XWT = ARM(1,C)
YWT = ARM(1, C)
ZWT = ARM(1, C)
CY1 = COS(ADC(I, 2)) * COTAN(ADC(I, 1)) / XMT / TAF / TAF
CPSI1 = -YWT * SIN(ADC(I, 1)) * YWT * COS(ADC(I, 2)) * COTAN(ADC(I, 1))
XMT / TAF / TAF
CPH1 = (XWT * COS(ADC(I, 2)) * COTAN(ADC(I, 1)) - YWT * COS(ADC(I, 3)) * COTAN(ADC(I, 2)))

COMMON /PLYCHA/GT, XMT, TAF, TFC, VTfc, TFC, TFC
1 = ABS( ADC(I, I) - 3.14159) / GT * COS(1) GO TO 2
XVAL = I*IC
GO TO 1

2 XVAL = COTAN( ADC(I, 3))
1 XWT = ARM(1,C)
YWT = ARM(1, C)
ZWT = ARM(1, C)
CY1 = COS(ADC(I, 2)) * COTAN(ADC(I, 1)) / XMT / TAF / TAF
CPSI1 = -YWT * SIN(ADC(I, 1)) * YWT * COS(ADC(I, 2)) * COTAN(ADC(I, 1))
XMT / TAF / TAF
CPSI2 = (YWT * COS(ADC(I, 1)) * COTAN(ADC(I, 2)) * XWT * SIN(ADC(I, 2)) / XMT / TAF / TAF

COMMON /PLYCHA/GT, XMT, TAF, TFC, VTfc, TFC, TFC
1 = ABS( ADC(I, I) - 3.14159) / GT * COS(1) GO TO 2
XVAL = I*IC
GO TO 1

2 XVAL = COTAN( ADC(I, 3))
1 XWT = ARM(1,C)
YWT = ARM(1, C)
ZWT = ARM(1, C)
CY1 = COS(ADC(I, 2)) * COTAN(ADC(I, 1)) / XMT / TAF / TAF
CPSI1 = -YWT * SIN(ADC(I, 1)) * YWT * COS(ADC(I, 2)) * COTAN(ADC(I, 1))
XMT / TAF / TAF
CPSI2 = (YWT * COS(ADC(I, 1)) * COTAN(ADC(I, 2)) * XWT * SIN(ADC(I, 2)) / XMT / TAF / TAF

RETURN
END

SUBROUTINE SNTFM (FXSN, FZSN, AHSN, THETA)

COMMON /INUT/IW, IP
COMMON /DATA/AERO (175), AEROP (50), KODE (26), LL
COMMON 227 (2C)
COMMON /TAH/x (175)

COMMON /SNU/ (3, 3), SN (3C), THUSN, THLSN, SNU (3, 3)

EQUIVALENCE (AEROP (10), SNU) , (AERO (104), SNU ()), (AERO (107), SNU ()), (AERO (113), SNU ())
1 (AERO (101), SNU ()), (AERO (104), SNU ()), (AERO (107), SNU ()), (AERO (113), SNU ())
FILE: CABLE FORTRAN TI

GRUMMAN DATA SYSTEMS

3 (AERO(114), SNLST), (AERO(115), SNLW), (AERO(116), SNLNL), CABO110C
4 (AERO(117), TAS), (AERO(118), TSLN), (AERO(119), AKSNJ), CABO1110
5 (AERO(120), AKNL), (AERO(49), V0), (AERO(51), RHO), CABO120E
6 (AERO(76), WLCR), (AERO(77), STACR), CABO130C
7 (AERO(78), BLCF), CABO140C

EQUIVALENCE (SN( 1), GX1), (SN( 2), GY1), (SN( 3), GZ1), CABO150C
1 (SN( 4), GX2), (SN( 5), GY2), (SN( 6), GZ2), CABO160C
2 (SN( 7), GX3), (SN( 8), GY3), (SN( 9), GZ3), CABO170C
3 (SN(10), GX4), (SN(11), GY4), (SN(12), GZ4), CABO180C
4 (SN(13), THU), (SN(14), TML), (SN(15), ALU), CABO190C
5 (SN(16), ALJ), CABO200C
6 (SN(19), THK), (SN(20), TMH), (SN(21), THG21), CABO210C
7 (SN(22), THK), (SN(23), TMH), (SN(24), THG22), CABO220C
8 (SN(25), THK), (SN(26), TMH), (SN(27), THG23), CABO230C
9 (SN(28), THK), (SN(29), TMH), (SN(30), THG24), CABO240C

IF(KODE(10), EQ.0) GO TO 5005
IF(KODE(10), EQ.3) GO TO 5005
CALL DCSV(N, THETA)
IF(KODE(10), NE.1) GO TO 5003

C TERMS TO MODEL SNUBBER EFFECTS (MODEL UNSNUBBED)

Q=5*PHD*V=W0
CALL STINT(N, ALU, 1, 1, TUSN, NG)
IF(NG, NE.0) GO TO 5000
CALL STINT(N, ALU, 2, 2, TUSN, NG)
IF(NG, NE.0) GO TO 5000

CALL STINT(N, ALU, 1, 1, TSLN, NG)
CALL STINT(N, ALU, 2, 2, TSLN, NG)

500 CONTINUE

C CALCULATING FORCE AND MOMENT EFFECTS

CALL DCSV(N, THETA)
FXSN = 2*TUSN*GX1
FYSN = 2*TUSN*GY1
AMSN = FXSN*SUZ*SUZ+FZUSN
FLSN = 2*TSLN*GX3
FZSN = FZUSN+FZLSN
ANSN = (AMSN+AMSN)/12.
RETURN

5003 CONTINUE

C TERMS TO MODEL SNUBBER EFFECTS (MODEL SNUBBED)

FXSN = 2*TUSN*GX1
FYSN = 2*TUSN*GY1
AMSN = FXSN*SUZ*SUZ+FZUSN
FLSN = 2*TSLN*GX3
FZSN = FZUSN+FZLSN
ANSN = (AMSN+AMSN)/12.
RETURN
FILED: C ABLE
FORTRAN T1
GRUMMAN DATA SYSTEMS

AMSN = (AMUS + AMLSN) / 12.
RETURN

5000 IF (SN>3) THEN
    IF (SN>2) THEN
        AMSN = 0
    END IF
RETURN
END

SUBROUTINE LONGSN
COMMON/INDUT/1W,1P
COMMON/DAT/AER0(175), AEPO(50), KODE(24), LL
COMMON/SNJD/SNU(3,3), SN(30), THUSN, THL, SN, SNUD(3,3)
COMMON Z77(200)
COMMON TAHIZZ(300)
COMMON/YOUNG(10,10)

EQUIVALENCE (AER0(105), SNUX), (AEPO(106), SNUY), (AEPO(107), SNJ)
1 (AEP0(108), SNLX), (AER0(109), SLY), (AEP0(110), SNL), (AEC0(111), SLC)
2 (AEP0(112), SNUL), (AEPC(113), SNUJL), (AEC0(114), SLNBL), (AEC0(115), SCL)
3 (AEPT(117), TUSD), (AEPT(118), TLSN), (AEPT(119), AKSNJ), (AEC0(120), TUSD)
4 (AEPT(122), AKSL), (AEPT(49), V), (AEP0(51), PHD), (AEP0(54), PSNL)
5 (AEPT(63), THE)), (AEP0(121), AOSNU), (AEP0(122), AOSNL)
6 (AEC0(36), FTCP), (3,3), FCT0(3,3)

DIMENSION FTCP(3,3), FCT0(3,3)
C0T(A)=1./TAN(A)
DO 1000 I=1,3
DO 1000 J=1,3
SNU(1,1)=
1001 SNU(1,J)=
    DO 1002 I=1,10
1002 DUM(1,J)=
    IF (KODE(1)<0.1) GO TO 1000
C TERMS FOR UNSNAPPED SNUBED EFFECTS (LONG)
DO 1000 I=1,7
    DO 1000 J=1,7
1003 DUM(1,J)=
    CALL ORCSN(THETA)
    DUM(1,1) = -2.*TUSN0 *CZ1
    DUM(1,4) = -2. * TUSN0 * SIN(TUSGX)
    DUM(1,6) = 2.*GX1
    DUM(2,3) = 2. * TUSN0 * GXL
    DUM(2,5) = -2.*TUSN0 * SIN(TUSGX)
    DUM(2,6) = 2.*DZ1
    DUM(3,1) = (-SNUX*DUM(1,1) + SNUX*DUM(2,3))/12.
    DUM(3,4) = -SNUX*DUM(1,4)/12.
    DUM(3,6) = SNUX*DUM(2,6)/12.
FILE CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

DUM(3,6) = (-SNUIZ*DUM(1,6)+SNUX*DUM(2,6))/12.
DUM(4,1) = (SIN(THGX1)/ALU)*12.
DUM(4,2) = (-GZ1*COT(THGX1)/ALU)*12.
DUM(4,3) = -SNUX*SIN(THGX1)/ALU-SNUZ*GZ1*COT(THGX1)/ALU
DUM(4,4) = -1.
DUM(5,1) = (-GZ1*COT(THGX1)/ALU)*12.
DUM(5,2) = (SIN(THGX1)/ALU)*12.
DUM(5,3) = SNUX*GZ1*COT(THGX1)/ALU + SNUX*SIN(THGX1)/ALU
DUM(5,5) = -1.

CALL DFCSN(THETA)

Q=9*GZ3*V*V

ALU=ALU-1.

CALL STINT(Q,ALU-1,1,TUSN1,NG)

IF(NG*GE*7) G0 TO 500 C

ALU=ALU-1.

CALL STINT(Q,ALU-1,1,TUSN2,NG)

IF(NG*LE*9) G0 TO 500 C

5000 WRITE(IW,5502) NG,ALL,ALU,Q

5502 ENDWA(TUP=123 IN TABLE 1-2,NG=4,12,3X1C3)

RETURN

5001 CONTINUE

AKT=(TUSN1-TUSN2)/2.

DUM(6,6) = -1.

DUM(6,7) = AKT*12.

DUM(7,1) = -GZ1

DUM(7,2) = -GZ1

DUM(7,3) = (-SNUX+ALU*GZ1)*G71=1*(-SNUX+ALU*GZ1)=G1)*12.

DUM(7,7) = -1.

CALL DFASH(3,7)

DO 1025 I=1,3

DO 1025 J=1,3

1025 CONTINUE

DUM(TJ,1)=DUM(1,1)

CALL DFASU(N(TUP)

DUM(1,3) = -2*TL9C*GZ3

DUM(1,4) = -2*TL9C=SIN(THGX3)

DUM(1,6) = 2*GZ3

DUM(2,7) = -2*TL9C*GZ3

DUM(2,5) = -2*TL9C=SIN(THGX3)

DUM(2,6) = 2*GZ3

DUM(3,3) = (SINL7*SUM(1,7)+SNLX*DUM(2,3))/12.

DUM(3,4) = SNLZ*DUM(1,4)/12.

DUM(3,5) = SNLX*DUM(2,5)/12.

DUM(7,6) = (SINL7*SUM(1,6)+SNLX*DUM(2,6))/12.

DUM(4,1) = (SIN(THGX3)/ALL)*12.

DUM(4,2) = (-GZ3*COT(THGX3)/ALL)*12.

DUM(4,3) = SNLZ*SIN(THGX3)/ALL - SNLX*GZ3*COT(THGX3)/ALL

DUM(4,4) = -1.

DUM(5,1) = (-GZ3*COT(THGX3)/ALL)*12.

DUM(5,2) = (SIN(THGX3)/ALL)*12.

DUM(5,3) = -SNLZ*GZ3*COT(THGX3)/ALL + SNLX*SIN(THGX3)/ALL

DUM(5,6) = -1.

CALL DFCSN(THETA)

ALL=ALL+1.

CALL STINT(Q,ALL-1,1,1,TLSN1,NG)

IF(NG*LE*6) G0 TO 1023
FILED CABLE

Fortran T1

GRUMMAN DATA SYSTEMS

ALL2=ALL1.
CALL 5*INT(2.*ALL2+1.+TLSN2*NG)
IF(NG.EQ.0.) GO TO 5004

L003 I=ITE(IW5002) NG,ALL,ALU,Q
RETURN

5004 CONTINUE
AKTL=(TLSN1-TLSN2)/2.
DUM(6,6)= -1.
DUM(6,7)= AKTL*12.
DUM(7,1)= -GX3
DUM(7,2)= -GZ3
DUM(7,3)=((-SNLX*ALL*GX3)*GZ3-(SNL7*ALL*G73)*GX3)/12.
DUM(7,7)= -1.
CALL HASH(3,7)
D0 1008 I=1,5
D7 1008 J=1,7

1008 FJOT(I,J)=DUM(I,J)
D7 1009 I=1,5
D7 1009 J=1,7
SNUD(I,J)=0

1009 SNUI(I,J)= FTDL(I,J)+JBT(I,J)
RETURN

100 C IF(KODC(1),EQ,') GO TO 1002
C TERMS FOR SNUBBED SNUBBED EFFECTS(5X)
CALL OF(CSN(THEETA))
D0 1006 I=1,7
D7 1006 J=1,7

1006 DUM(I,J)=
DUM(1,1)= -2.*TUSNC*GZ1
DUM(1,4)= -2.*TUSNC*SIN(THGX1)
DUM(1,6)= 2.*GX1
DUM(2,7)= 2.*TUSNC*GX1
DUM(2,8)= -2.*TUSNC*SIN(THGU1)
DUM(2,9)= 2.*GX1
DUM(3,7)= (-SNUZ*DUM(1,3)+SNUX*DUM(2,3))/12.
DUM(3,4)= -SNUZ*DUM(1,4)/12.
DUM(3,5)= SNUX*DUM(2,5)/12.
DUM(3,6)= (-SNUZ*DUM(1,6)+SNUX*DUM(2,6))/12.
DUM(4,1)= (SIN(THGDX1)/ALU)*12.
DUM(4,2)= (-GZ1*COT(THGDX1)/ALU)*12.
DUM(4,3)= -SNUX*SIN(THGDX1)/ALU-SNUZ*GZ1*COT.THGDX1)/ALU
DUM(4,4)= -1.
DUM(5,1)= (-GX1*COT(THGDX1)/ALU)*12.
DUM(5,2)= (SIN(THGDX1)/ALU)*12.
DUM(5,3)= SNUZ*GX1*COT(THGDX1)/ALU+SNUZ*SIN(THGDX1)/ALU
DUM(5,5)= -1.
DUM(6,6)= -1.
DUM(6,7)= AKNUC*12.
DUM(7,1)= -GX1
DUM(7,2)= -GZ1
DUM(7,7)= (-SNUZ+ALU*GX1)*GZ1-(-SNUX+ALU*GZ1)*GX1)/12.
DUM(7,8)= -1.
D7 10 I=1,7
D7 10 J=1,7

10 SNUI(I,J)=DUM(I,6)*AKSNU*DUM(7,J)*12.

78
CALL WASH(3,7)
DO 1007 I=1,3
DO 1019 J=1,3

1007 FDOT(I,J)=SUM(I,J)
DUM(1,1) = -2.*TLSN*GZ3
DUM(1,4) = -2.*TLSN*SIN(THG3)
DUM(1,6) = 2.*GX3
DUM(2,3) = 2.*TLSN*GZ3
DUM(2,5) = -2.*TLSN*SIN(THG3)
DUM(2,6) = 2.*GZ3
DUM(3,3) = (SNL7*SUM(1,3)+SNLX*SUM(2,3))/12.
DUM(3,4) = SNL7*SUM(1,4)/12.
DUM(3,5) = SNLX*SUM(2,5)/12.
DUM(3,6) = (SNL7*SUM(1,6)+SNLX*SUM(2,6))/12.
DUM(4,1) = (SIN(THG3)/ALL)*12.
DUM(4,2) = (-GZ3*COT(THG3)/ALL)*12.
DUM(4,3) = SNLX*SIN(THG3)/ALL - SNLX*GZ3*COT(THG3)/ALL
DUM(4,4) = -1.
DUM(5,1) = (-GZ3*COT(THG3)/ALL)*12.
DUM(5,2) = (SIN(THG3)/ALL)*12.
DUM(5,3) = -SNLX*GZ3*COT(THG3)/ALL + SNLX*SIN(THG3)/ALL
DUM(5,4) = -1.
DUM(6,5) = -1.
DUM(6,7) = AKSN * 12.
DUM(7,1) = -GZ
DUM(7,2) = -GZ
DUM(7,3) = ((-SNLX+ALL*GZ3)*GZ3 - (SNLX+ALL*GZ3)*GZ3)/12.
DUM(7,4) = -1.
DO 20 I=1,3
DO 20 J=1,3

20 SNUI(I,J) = JD(I,J)+DUM(I,6)*3+SNUL*SUM(7,4)*12.
CALL WASH(3,7)
DO 1010 I=1,3
DO 1019 J=1,3

1010 FDOT(I,J)=SU(I,J)
DO 1011 I=1,3
DO 1019 J=1,3

1011 SNUI(I,J) = FDOT(I,J)+FCT(I,J)
RETURN

1002 DO 1007 I=1,3
DO 1003 J=1,3
SNUL(I,J)=0

1003 SNUI(I,J)=
RETURN
END
SUBROUTINE DEC9N(THETA)
COMMON/DAT/AERG(175),AERG(53)*KODE(25)*ALL
COMMON/SNUI(3)/SNUI(3),SN(3),THSN,THLSN,SNUI(3,3)
EQUIVALENCE(AERG(105),SNUX),(AERG(106),SNUY),(AERG(107),SNUS),(AERG(108),SNUL)
1 (AERG(109),SNLX),(AERG(109),SNLY),(AERG(110),SNLZ),(AERG(110),SNLU)
2 (AERG(111),SNUST),(AERG(112),SNUWL),(AERG(111),SNUL),(AERG(112),SNUST)
3 (AERG(114),SNLST),(AERG(115),SNLW),(AERG(115),SNLWL),(AERG(114),SNLST)
4 (AERG(117),TUSNC),(AERG(118),TLSND),(AERG(119),AKSN),(AERG(119),AKSN)
5 (AERG(120),AKSN),(AERG(120),AKSN)
6 (AERG(76),WLC4),(AERG(77),STACF),(AERG(78),BLCF)
CAMO1.00
FILE: CABLE FORTAN II  GRUMMAN DATA SYSTEMS

EQUIVALENCE (SN(1), GX1), (SN(2), GY1), (SN(3), GZ1), CABCC11C
1 (SN(4), GX2), (SN(5), GY2), (SN(6), GZ2), CABCC120
2 (SN(7), GX3), (SN(8), GY3), (SN(9), GZ3), CABCC130
3 (SN(10), GX4), (SN(11), GY4), (SN(12), GZ4), CABCC140
4 (SN(13), THU1), (SN(14), THL1), (SN(15), ALU), CABCC15C
5 (SN(16), ALL), CABCC16C
6 (SN(19), THGX1), (SN(20), THGY1), (SN(21), TGZ1), CABCC170
7 (SN(22), THGX2), (SN(23), THGY2), (SN(24), TGZ2), CABCC180
8 (SN(25), THGX3), (SN(26), THGY3), (SN(27), TGZ3), CABCC190
9 (SN(28), THGX4), (SN(29), THGY4), (SN(30), TGZ4), CABCC20C

C CALCULATION OF SNUBSIC CABLE DIRECTION COSINES
X31: (STAC-SNUST)*COS(THETA)-(WLCF-SNUWL)*SIN(THETA)
Z81: (WLCF-SNUWL)*COS(THETA)+(STAC-SNUST)*SIN(THETA)
X82: X81
Z82: Z81
X83: (STAC-SNUTL)COS(THETA)-(WLCF-SNUL)SIN(THETA)
Z83: (WLCF-SNUL)*COS(THETA)+(STAC-SNUTL)*SIN(THETA)
X84: X83
Z84: Z83
DX1 = X81+SNUX
DY1 = -SNUBL+SNUY
DZ1 = Z81+SNUZ
DX2 = DX1
DY2 = SNUBL-SNUY
DZ2 = D71
DX3 = X83+SNLX
DY3 = SNUL-SNLY
DZ3 = Z83-SNLZ
DX4 = DX3
DY4 = -SNUBL+SNLY
DZ4 = DX7
ALJSO = DX1**2 + DY1**2 + DZ1**2
ALU = SQRT(ALJSO)
ALLJSO = DX3**2 + DY3**2 + DZ3**2
ALL = SQRT(ALLJSO)
GX1 = DX1/ALU
GY1 = DY1/ALU
GZ1 = DZ1/ALU
G42 = DX2/ALU
GY2 = DY2/ALU
G22 = D22/ALU
G33 = D33/ALL
GY3 = DY3/ALL
G23 = D23/ALL
G44 = DX4/ALL
GY4 = DY4/ALL
G43 = D43/ALL
DG1 = I=19,3
J=1-19
1 SN(J) = ACOS(SN(J))
RETURN
END

SUBROUTINE DECUSN(THETA)
COMMON/DAT/A=0.075, A=2PC(5), KODE(26), LL
COMMON/SNUS/SNU(3, 3), X(30), THUSN, THLSN, SNU(3, 3)
COMMON/CABCC/CABCC
EQUIVALENCE(AERO(1)C), SNUX, (AERO(1)C), SNUX, (AERO(1)C), SNUX), CABC6400
1 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6730
2 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6400
3 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6400
4 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6400
5 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6400
6 (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), (AERO(1)C), CABC6400

EQUIVALENCE
(SN U1) (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
1 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
2 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
3 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
4 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
5 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
6 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
7 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
8 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)
9 (SN P1) (SN P2) (SN P3) (SN P4) (SN P5) (SN P6) (SN P7) (SN P8) (SN P9) (SN P10)

C CALCULATION FOR EFFECTIVE DIRECTION COSINES FOR UNSNABBED CASE
AYL = SNNL - (SNNL + SNNL)
AZL = SNNL - (SNNL + SNNL)
AZU = SNNL - (SNNL + SNNL)
THU = ATAN(THU/AYL)
THL = ATAN(THL/AYL)
ALU = ALU/SIN(THU)*COS(THL)
GZ1 = -COS(THL)/AYL
GY1 = -GY1
GZ2 = -GZ2

GZ1 = GZ1 + COS(THL) - GZ1*SIN(THL)
GY1 = GY1
GZ2 = GZ2

GZ2 = GZ2 + COS(THL) - GZ2*SIN(THL)
GZ3 = GZ3 + COS(THL) - GZ3*SIN(THL)
GY3 = GY3
GZ2 = GZ2

GZ3 = GZ3 + COS(THL) - GZ3*SIN(THL)
GZ4 = GZ4
GY4 = GY4
GZ4 = GZ4

DO 1 I=19, 30
J=I-18
SN(J) = COS(SN(J))
RETURN
END

SUBROUTINE FILE
COMMON/IND/....
COMMON/ALT/A, (175), AER0P(50), KOR(26), FL
IF(KOR(7) .GT. 1) GO TO 1
WRITE(W,*)
100 PRINT(25, "FRONT CABLE VERTICAL", "Rear CABLE HORIZONTAL")

81
GO TO 4
1 IF (KODE(6) .GT. 2) GO TO 2
   WRITE (IW,200)
200 FORMAT (25X, 'FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL*')
   GO TO 4
2 IF (KODE(6) .GT. 3) GO TO 3
   WRITE (IW,300)
300 FORMAT (25X, 'BOTH CABLES VERTICAL*')
   GO TO 4
3 WRITE (IW,400)
400 FORMAT (25X, 'BOTH CABLES HORIZONTAL*')
   CONTINUE
   IF (KODE(11).EQ.0) GO TO 5
   IF (KODE(11).EQ.1) GO TO 6
   WRITE (IW,500)
500 FORMAT (25X, 'SNUBBES SNUBBED*')
   GO TO 7
5 WRITE (IW,600)
600 FORMAT (25X, 'NO SNUBBES*')
   GO TO 7
6 WRITE (IW,700)
700 FORMAT (25X, 'SNUBBES UNSNUBBED*')
   CONTINUE
   IF (KODE(11).EQ.0) GO TO 9
   WRITE (IW,800)
800 FORMAT (25X, 'LIFT/ANTI-LIFT CABLE IN*')
   GO TO 9
7 WRITE (IW,900)
900 FORMAT (25X, 'NO LIFT/ANTI-LIFT CABLE*')
   CONTINUE
   IF (KODE(13).LE.0) WRITE (IW,1000)
   IF (KODE(13).GT.0) WRITE (IW,1001)
   WRITE (IW,1002)
1000 FORMAT (25X, 'FEEDBACK LOGIC NOT IN*')
1001 FORMAT (25X, 'FEEDBACK LOGIC IN*')
1002 FORMAT (25X, 'CABLELESS MODEL CHARACTERISTICS*')
   RETURN
END
SUBROUTINE STINT(A1,A2,A3,WINTAL,MAXTBL,FCT,NG)
 EQUVALENCE (X(1), NUMPTS(1))
 COMMON NUMPTS(1)
 DIMENSION X(1)
 IF = NUMPTS(1) / 3
70 IF (WINTAL-MAXTAL)/1.1.112
71 DO 73 II=WINTAL, MAXTAL
    NJ = NUMPTS(II) + 1
    IF (II .GT. WINTAL) II = WINTAL + 1
72 CONTINUE
73 CONTINUE
 GO TO 112
75 IF = II - 1
 IL = 2
 NJ = NJ
10 CONTINUE
FILE CABLE FORTRAN TI

NI = 1Z+11
ID = NUMPTS(NI)
IP = 10+NJ
DC 77 IO=1,11
NN = NJ+10
IF (A1-X(NN))76,79,77
76 IF (12-1) 112,112,79
77 CONTINUE
G0 TO 112
78 IG = -1
G0 TO 96
79 IG = +1
80 NI=NI+12
IS = NUMPTS(NI)
DO 92 IA=1,10
NS=IF+1A
IF (A2-X(NS))81,63,82
81 IF (IA-1) 110,112,94
82 CONTINUE
G0 TO 112
83 IH = -1
G0 TO 85
84 IH = +1
85 NE=IP+IP+IO+IO+IA-ID
NE=NE-ID
IF(IG+IH)96,88,91
86 IF (X(NE)-99998.5E9)97,113,113
87 FCT = X(NE)
G0 TO 95
88 IF(IG)89,91,93
89 IF(A MAX1(X(NE),X(NE-1))-99998.5E9)90,113,113
90 FCT =X(NE)-(X(NE)-A2)*(X(NE)-X(NF))/(X(NS)-X(NS-1))
G0 TO 95
91 IF(A MAX1(X(NE),X(NE-1),X(NE-1),X(NR-1))-99998.5E9)92,113,113
92 FCT = (((X(NS)-A2)*((X(NE)-A1)*X(NE-1)-(X(NE-1)-A1)*X(NF))
1-(X(NS-1)-A2)*((X(NE)-A1)*X(NE-1)-(X(NE-1)-A1)*X(NF))
2/((X(NS)-X(NS-1))*X(NN)-X(NN-1)))
G0 TO 95
93 IF(A MAX11 X(NE), X(NE-1))-99998.5E9)94,113,113
94 FCT = X(NE)-(X(NE)-A1)*X(NE-1))/(X(NE)-X(NE-1))
95 GO TO (96,99,99)
96 DUMSTG =FCT
97 IL = I-1
98 FCT = DUMSTG-(X(NM)-A3)*(DUMSTG-FCT)/(X(NM)-X(NJ))
99 RETURN
74 IK = 3
IL = 3
G0 TO 101
110 NG = 2
G0 TO 99
112 NG = 3
G0 TO 99
113 NG = 4
GO TO 00
END
FILE: CABLE FORTRAN T1

SUBROUTINE TAUINV(NUNIT, NZ, NG)
COMMON NUMPTS(1)
COMMON /INOUT/ I1U, I19
COMMON /TABOUT/ NUNITBL, ISOQ
DIMENSION NUMPTS(1)
INTEGER I, I1, LABEL(27)
EQUIVALENCE (NUMPTS(1), NUMPTS(1)), (DUMMY(1), DUMMY)
DIMENSION DUMMY(10)
VC=0
10 IZ=IABS(NZ)

NUNIT=5
IF(NZ.LT.0) NUNIT=8
NUNITBL = NUNITBL
NG=2
NUMPTS(I) = IZ+I7+I2
20 FORMAT(NUNIT, 57) K, L1N, L2N, LABEL, ISOQ
IF(NKPO.EQ.0) GO TO 3
4 WRITE(IW*1) K, LIN, L2N, LABEL, ISOQ
1 FORMAT(115, 10X, 27A2, 146)
57 FORMAT(8X14, 212, 27A2:12)
3 IF(ISOQ) 69, 58, 69
58 IF(K) 99, 99, 99
59 4 = IZ + NUNITBL
NUMPTS(W) = LIN
M = 4 + IZ
NUMPTS(W) = L2N
IF(NUMTBL-NUNITBL) 17, 73, 17
17 NUMPTS(NUNITBL) = MVMWY
70 N1 = (LIN-1) / 9 + 1
DO 68 IS = 1, N1
L3 = (IS-1) * 9 + 1
IF((IS-N1) 6C, 61, 60
60 L4 = L3 + 8
GO TO 62
61 L4 = LIN
62 LS = NUMPTS(NUNITBL) + 1
L6 = L5 + L7
L7 = L5 + L4
JJ = 0
L4 = L5 + LIN
LN = LM + L2N
63 READ(NUNIT, 64) (DUMNY(K), K=1, 10), ISOQ
64 FORMAT(11EF7.0, 12)
IF(NKPO.EQ.0) GO TO 5
6 WRITE(IW*2) DUMMY, ISOQ
2 FORMAT(10C12.4, 15)
5 NUMPTS(L5) = DUMNY(1)
K = 2
DO 65 J = L6+L7
NUMPTS(J) = DUMNY(K)
65 K = K+1
ISOQ = (IS-1) * (L2N+1) + JJ+1
IF((ISOQ-I970) 69, 66, 69
66 L6 = LN + L3
L7 = LN + L4
FILEC CARLE FORTRAN TI

GRUMMAN DATA SYSTEMS

LS = LM + 1 + JJ
IF (JJ-L?N) 67, 63, 69
67 JJ = JJ + 1
LN = LN + LIN
GO TO 63
68 CONTINUE
100 NUNNY = NUMPTS(NIMTL) + (LIN+1) * (L2N+1)
108 NIMTL = NIMTL + 1
GO TO 102
69 NG = 1
99 RETURN
END
SUBROUTINE STINTI(A1,A2,A3,NIMTL,MAXTBL,FCT,NG)
EQUIVALENCE (X(1),NUMPTS(1))
COMMON/TAIL/NUMPTS(1)
DIMENSION X(1)
IZ = NUMPTS(1)/3
70 IF(NIMTL-MAXTBL)71,71,11C
71 DO 73 I=1,NIMTL,MXTBL
NJ = NUMPTS(I)+1
72 IF(A1-X(NJ))72,74,73
73 CONTINUE
GO TO 112
75 I1 = 1
IL = 2
NM = NJ
101 DO 97 IF=IK,IL
NJ = NUMPTS(I1)+1
74 I1 = IZ+II
75 J = NUMPTS(N1)
IZ = IZ+NJ
DO 77 IO=1,1J
VN = NJ+IO
76 IF(I1-O1)112,112,79
77 CONTINUE
GO TO 112
78 IG = -1
GO TO 72 80
79 IG = +1
80 NI = NI+12
7A I1 = NUMPTS(N1)
DO 82 IA=1,19
83 IA = IA+1A
85 NS=IP+1A
86 IF(A2-X(NS))81,83,92
81 IF(I1-O1)112,112,94
82 CONTINUE
GO TO 112
83 IH = -1
GO TO 72 85
85 IH = +1
86 IA = IA+1A+10+1A*1A+10
-10
I(W) 96.88.91

85
FILE: CABLE  FORTRAN T1  
GRUMMAN DATA SYSTEMS

86 IF (X(NF)-99998.5E9) 187, 113, 113  
87 FCT = X(NF)  
GO TO 95  
88 IF (IG) 89, 110, 93  
89 IF (MAXI (X(NF), X(NF), X(NF)-99998.5E9)) 113, 113, 113  
90 FCT = (X(NF)-A2)*(X(NF)-A1)*(X(NF)-A1)  
GO TO 95  
91 IF (MAXI (X(NF), X(NF), X(NF)-99998.5E9)) 92, 113, 113  
92 FCT = ((X(NF)-A2)*(X(NF)-A1)*(X(NF)-A1)*(X(NF)-A1))  
GO TO 95  
93 IF (MAXI (X(NF), X(NF)-99998.5E9)) 94, 113, 113  
94 FCT = (X(NF)-A2)*(X(NF)-A2)  
GO TO (96, 99, 99) IF  
95 DO105 G = A1  
96 DUMSTG = FCT  
97 II = II-1  
98 FCT = DUMSTG + (X(NF)-A3)*(DUMSTG-FCT)  
GO TO 99  
99 RETURN  
74 IK = 3  
IL = 3  
GO TO 101  
110 YG = 2  
GO TO 99  
112 NG = 3  
GO TO 99  
113 NG = 4  
END  
SUBROUTINE TABINIT (NUMTTL, NZ, NG)  
COMMON/INPUT/W, IF  
COMMON/TABLENUMPTS(1)  
COMMON/TABR/NUMTTL, ISO0  
DIMENSION NUMPTS(1)  
INTEGER*2 LABEL(27)  
EQUIVALENCE (NUMPTS(1), NUMPTS(1)), (NUMMY(1), NUMMY)  
DIMENSION DUMMY(1)  
ICF = 1  
IZ = IABS(NZ)  
NUMTTL = E  
IF (NZ.LT.0) NUMTTL = 8  
NG = NUMTTL  
NUMPTS(1) = IZ+IZ+I1/2  
102 READ(NUMTTL, 57) K, N, L2N, LABEL, ISF0  
IF (WCF.EQ.0) GO TO 3  
4 WRITE((W+1)K, L2N, LABEL, ISF0  
1 READ (315, 10Y, 27A2, 146)  
57 IF (K.EQ.0) A9, 6A, 60  
1 IF (K) G9, 99, 59  
59 M = 17 + NUMTTL  
NUMPTS(M) = L2N  
W = W + I7  
NUMPTS(M) = L2N
FILE CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

IF(NIMTBL-NIMTBL)17,70,17
17 NUMPTS(NIMTBL) = NUMY
70 N = (LIN-1) / 9 + 1
DO 68 IS = 1, N
   L3 = (IS-1) * 9 + 1
   IF (IS-N!) 60, 61, 60
60 L4 = L3 + 8
   GO TO 52
61 L4 = LIN
62 LS = NUMPTS(NIMTBL) + 1
   LA = LR + L3
   L7 = LS + L4
   JJ = 0
   LM = LS + LIN
   LN = LM + L2N
63 READ(NUNIT,64) (DUMMY(K),K=1,1C), ISEO
64 FORMAT (1ICE7.2,12) IF(NCF,EQ,'C') GO TO 5
6 WITE(IW,2)DUMMY,ISEQ
2 FORMAT(1ICE12.4,15)
5 XNUMPTS(L5) = DUMMY(1)
   K = 2
   DO 65 J = L6,L7
   XNUMPTS(J) = DUMMY(K)
65 K = K+1
   ISEO=(IS-1)*(L2N+1)+JJ+1
   IF(ISEO-ISEO) 69,66,69
66 LK = LN + L3
   L7 = LN + L4
   LS = LM + 1 + JJ
   IF (JJ-L2N) 67, 68, 69
67 JJ = JJ + 1
   LN = LN + LIN
   GO TO 63
68 CONTINUE
109 NUMY = NUMPTS(NIMTBL) + (LIN+1) * (L2N+1)
10B NIMTBL = NIMTBL + 1
   GO TO 102
69 NG = 1
99 RETURN
END

SUBROUTINE FRICT(IDX)
COMMON/LAT/ATC(175),AERO(50),KODE(16)
COMMON/DOUGH/LOCIC(3,6)
EQUIVALENCF (AERO(96),COU),(AERO(104),CMP)
DO 1 I=1,3
   DO 1 J=1,6
1 FRICT(I,J)=C.
   IF(CMP.FEQ,'COU.EQ.G')RETURN
   INO=KODE(16)
   IF(IDX.NEQ.C1)GO TO 2
C LONGITUDINAL PULLEY FRICTION COMPUTATION
   GO TO(1*,11,12,13),INO
10 CALL FRTV(1)
RETURN
11 CALL FERT(3)
RETURN
12 CALL FERT(1)
CALL FERT(3)
13 RETURN

C LATERAL DIRECTIONAL FRICTION COMPUTATION
2 GO TO(21,22,23),IND
20 CALL FEHZ(3)
RETURN
21 CALL FEHZ(1)
22 RETURN
23 CALL FEHZ(1)
CALL FEHZ(3)
RETURN

SUBROUTINE FERT(IC)
C COMPUTES THE FRICT. EFFECT OF THE VERT PULLEYS ON THE LONG. DYN.
COMM=DAT/AERO(175),AERO(50),KODE(20)
COMM=PL/YCHA,EFS,XLTH(5),ADC(5,3),ARM(5,3),TF,*LET,TF
COMM=ROUGH/FLC(3,5)
EQUVALENCE (AERO(92),SVF), (AERO(92),SVF) (AERO(95),CQU).
1(AERO(141),CMR)
DIMENSION DT1(3),DT2(3)
IF(IC.EQ.3)GO TO 1
TENS=TF
RAD=SVF/12.
AVX=(ADC(2,1)-ADC(1,1))/2.
CAX=COS(AVX)
CAZ=SIGN(AVX)
GO TO 2
1 TENS=TF
RAD=SVF/12.
AVX=3.14157+(ADC(4,1)-ADC(3,1))/2.
CAX=COS(AVX)
CAZ=SIGN(AVX)
2 ARMX=(ARM(1C+1)+ARM(1C+1))/2.
ARMZ=(ARM(1C+1)+ARM(1C+1))/2.
ENDX=TENS*COS(ADC(1C+1))
ENDZ=TENS*(ADC(1C+1))
ENDW=SIGN(1+SIGN(ADC(1C+1)))
CMPP=CMPP/ENDW
FACU=CMPP*ENOR*FAD**2
ENORX=SIGN(ADC(1C+1))
ENORY=SIGN(1+SIGN(ADC(1C+1)))
ENORW=SIGN(1+SIGN(1+SIGN(ADC(1C+1)))
CMPP=CMPP/ENDW
FACL=CMPP*ENOR*FAD**2
FACT=4.0*ENOR/3.14159*FAD**2
CALL DLTH(CXY,CZ,CY,IC,C)
CALL DLTH(CY,CZ,CY+1,IC+1,0)
DT1(1)=FACT *(CXY-CX)
DT1(2)=FACT *(CZ-CT)
DT1(3)=FACT *(CT-CX)
DT2(1)=FACL*CXY-FACU*CX
DT2(2)=FACL*CZ-FACU*CZ

88
C COMPUTES THE FRIC* EFFECT OF THE MOVE PULLEYS ON THE LAT. DIR. DYN.

COMMON/DAT/AER (175), AERO(50), KOE(L26)
COMMON/FLYCAR/TS, XLSTH(5), ADC(5, 3), ARM(5, 3), TR, TLF, TF
COMMON/ROUGH/FRICT(3, 6)

EQUIVENC (AERO(91), EFF), (AERO(93), PHP), (AERO(96), COU),

1(AERO(114), CMP)
DIRECTION DT(1), DT(2)

IF((IC, EQ, 3) GO TO 1

TENS=TF
RAD=RHF/12.

GO TO 2

1 TENS=TF
RAD=RHF/12.

2 ENERGY=TENS*COS(ADC(IC, 1))
ENDY=TENS*(1+COS(ADC(IC, 2)))
ENORM=SQR(ENERGY*ENERY*ENFX*ENFY)

C4M=CMP/ENERY*
FACL=CMP/ENERY/RAD**2
FACT=C/M(CP1/1, 14556/RAD**2)
CALL DLSTH(ICY, CPSI, CPM, IC, 1)
CALL DLSTH(CPY, CPS1, CPHP, IC, 1, 1)

DT(1)=FACT*(ICY-CPY)

DT(2)=FACT*(CPSI-CPS1)

D-T(3)=FACT*(CPHM-CPHIP)

D-T(4)=FACT*(CPHM-CPHIP)

D-T(5)=FACT*(CPHM-CPHIP)

D-T(6)=FACT*(CPHM-CPHIP)

D-T(7)=FACT*(CPHM-CPHIP)

D-T(8)=FACT*(CPHM-CPHIP)

D-T(9)=FACT*(CPHM-CPHIP)

D-T(10)=FACT*(CPHM-CPHIP)

D-T(11)=FACT*(CPHM-CPHIP)

D-T(12)=FACT*(CPHM-CPHIP)

D-T(13)=FACT*(CPHM-CPHIP)

D-T(14)=FACT*(CPHM-CPHIP)

D-T(15)=FACT*(CPHM-CPHIP)

D-T(16)=FACT*(CPHM-CPHIP)

D-T(17)=FACT*(CPHM-CPHIP)

D-T(18)=FACT*(CPHM-CPHIP)

D-T(19)=FACT*(CPHM-CPHIP)

D-T(20)=FACT*(CPHM-CPHIP)

CONTINUE

RETURN

END

SUBROUTINE MATRIX(CMAT, N, FOOT, KAA, IEF)

COMMON/DAT/AERO(175), AERO(50), KOE(L26), LL
C DET CONTAINS VALUE OF DETERMINANT OF SMAT WITH G=1
1291 WRITE (IWK,1292)DET
1292 FORMAT (12H DETERMINANT=1.0F15.7)
128 NC = 0
C COUNT NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN COLUMN JS
DO 126 IS=1,NC
IF (MAT(IS,JS)) 99,125,121
121 NC = NC + 1
IS = I
120 CONTINUE
IF (NC-1) 17,126,130
17 FORMAT (I4,16)
16 FORMAT (*) MATRIX IS SINGULAR*
GO TO 257
125 IF (IS-JS) 99,140,123
C ONE INTER CHANGE TRIANGULARIZES THE COLUMN
DO 126 JS=1,NCOL
K1 = MAXI(MAT(IS,JS), MAT(JS,JS))
MA = MAT(IS,JS)
MAT(IS,JS) = MAT(JS,JS)
MAT(JS,JS) = MA
JS = JS + 1
SA = SMAT(IS,JS,K)
SMAT(IS,JS,K) = SMAT(JS,JS,K)
126 SMAT(JS,JS,K) = SA
GO TO 141
130 IS = JS+1
C LOOP 137 REDUCES ALL ELEMENTS BELOW DIAGONAL IN COLUMN JS BY
C AT LEAST ONE DEGREE
14 = IS
137 IF (MAT(IS,JS)) 99,137,129
129 IF (MAT(IS,JS)) 99,133,132
132 IF (MAT(IS,JS) = MAT(JS,JS)) 133,134,134
133 DJ 131 IJS = JS,NCOL
KI = MAXI(MAT(JS,JS), MAT(I,J))
MA = MAT(JS,JS)
MAT(JS,JS) = MAT(I,J)
MAT(I,J) = MA
JS = JS + 1
SA = SMAT(I,J,K)
SMAT(I,J,K) = SMAT(JS,JS,K)
131 SMAT(JS,JS,K) = SA
GO TO 139
134 KI = MAT(I,JS)
KJS = MAT(JS,JS)
KD = KI - KJS
F = SMAT(I,JS,K) / SMAT(JS,JS,K)
IF (DABS(F) = 4.0) 1052,1051,1751
1051 IF (KD) 99,133,1052
1052 DJ 235 IJS = JS,NCOL
KI = MAT(JS,JS)
IF (KJS,ENDS) GO TO 235
JS = JS + 1
KJS = KJS + KD
SMAT = F * SMAT(JS,JS,K)
FILED CABLE FORTRAN II ........................................ GRUMMAN DATA SYSTEMS

IF(K1-N) 141,141,2
2 WRITE(14,31)                     C9L01130
3 FORMAT(79H0,DEGREE OF POLYNOMIAL FORMED WHILE TRIANGULARIZING ORIG- C9L01140
1NAL MATRIX IS TOO HIGH)
G0 TO 257                              C9L01150

141 IF ( DABS (FIRMAI - BMAT(I,J,KI)) .LE. 2.D-6 * DABS (FIRMAI)) C9L01160
1 GO TO 134                              C9L01170
3 BMAT(I,J,KI)= BMAT(I,J,KI)- FIRMAT                  C9L01200
G0 TO 135                              C9L01210
136 BMAT(I,J,KI) = 0.*0.0                  C9L01220
135 CONTINUE                              C9L01230
235 CONTINUE                              C9L01240

C
142 CONTINUE                              C9L01250
J=JS                              C9L01260

147 CONTINUE                              C9L01270
K1=BMAT(JS,J)+KD                     C9L01280
KJ=BMAT(I,J)                        C9L01290
IF(K1.LT.KJ) K1=KJ
IF(MAT(I,J) = 0) DO 140 K=1,K1
138 MAT(I,J) = K                                   C9L01330
140 CONTINUE                              C9L01340
J=J+1                              C9L01350
IF(.....FNCOL) GO TO 142               C9L01360

137 IF(1.LE.NROW) GO TO 139               C9L01370
139(NP)= 128,128,1105
140 JS = JS +1                             C9L01380
IF(JS=NCOL)129,150,151
150 IF(NP.LT.C) GO TO 153              C9L01390
WRITE(16,13)                          C9L01400

13 FORMAT (1H20(1H213H FINAL MATRIX))   C9L01410
DO 151 J=1,NCOL                        C9L01420
DO 151 I=1,NROW                        C9L01430
K1 = MAT(I,J)                        C9L01440
IF(K1)99,151,152
132 WRITE(16,12) I, J, (BMAT(I,J,K), K=1,K1) C9L01450
5 CONTINUE                              C9L01460
153 LK=1                              C9L01470

C LOOP 150=ROOTS OF POLYNOMIALS ON DIAGONAL OF TRIANGULARIZED MATRIX ARE C9L01480
C FOUND AND STORED IN ARRAY ROOTS.
C COEFFICIENTS OF THE POLYNOMIAL EQUIVALENT OF THE DETERMINANT C9L01500
C OF THE MATRIX ARE COMPUTED AND STORED IN ARRAY CA WITH C9L01510
C CA(1) THE CONSTANT TERM.
C
DO 160 J=1,NCOL                      C9L01520
K1 = MAT(J,J)                        C9L01530
M4M=K1+1                            C9L01540
K2=K1-1                             C9L01550
DO 163 K=1,K1                          C9L01560
M4M=M4M-K                           C9L01570
163 N(M4M,4)= BMAT(J,J,K)            C9L01580
IF(K1.EQ.1) GO TO 1620               C9L01590
16: CALL POLYRT(0(14),ROOTS(2*LK-1),KOUNT(LK),K2,J(1,1),D(1,2),D(1,3)) C9L01600

92
C - 2
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

COMPLEX RT(1)
REAL*8 CR
DATA CF//7FFFEFEFEFEFEFE/
DIMENSION KOUNT(1)
K=1
I=1
1 IF(KOUNT(I).GE.0)GO TO 3
RT(K) = CMPLX(SNGL(ROOTS(2*I-1)),SNGL(ROOTS(2*I)))
RT(K+1)=CONJG(RT(K))
K=K+2
GO TO 5
3 RT(K)=CMPLX(SNGL(ROOTS(2*I-1)),0.)
K=K+1
IF(ROOTS(2*I).EQ.CF)GO TO 5
RT(K)=CMPLX(SNGL(ROOTS(2*I)),0.)
K=K+1
5 I=I+1
IF(K.GE.K4A)RETURN
GO TO 1
END
SUBROUTINE POLYRT(AC,ROOT,KTOUNT,MM,0,A,T)
DIMENSION KOUNT(3)
DOUBLE PRECISION A(5),ROOT(5),Q(5),T(3),A(5)
DOUBLE PRECISION D2 , DABS , TOL , S1 , S2
COMMON /HARK/ D1,D2,K*NIX
M = MM
90 IF (A(M+1)) 100,95,100
95 ROOT(M) = 0.DC
KOUNT((M+1)/2) = 0
M = M - 1
GO TO 90
TOL = 1.D5
100 IF (M - 1) 44C,103,106
103 ROOT(1) = -AO(2)/AO(1)
KOUNT(1) = 0
GO TO 460
106 KODE = -1
N = M
N1 = N + 1
K = 0
DO 110 I = 1,N1
110 A(I) = AO(I)
IF(A(N-I))115,112,115
112 B1(I) = 1.D-5
B2(I) = 1.D-8
GO TO 120
115 R2(I) = A(N+1)/A(N-1)
B1(I) = -R2(I)*(A(N-2)/A(N-1)) - A(N)/A(N-1)
120 IF (N - 2) 121,122,130
121 KOUNT(K+1) = 0
A(2) = -A(2)/A(1)
GO TO 317
122 KOUNT(K+1) = 0
A(2) = -A(2)/A(1)
95
ILEO Cable

Fortran T1

Grunman Data Systems

A(3) = -A(3) / A(1)

GO TO 310

130 CALL GFDWL(T(N-2),Q(N))

ITERB = 0

KEY = 30

INK = 15

MURDER = 20

LVE = 4

220 ITERB = ITERB + 1

230 Q(1) = A(1)

Q(2) = A(2) + B1(1)*Q(1)

DO 240 J = 3,N1

240 Q(J) = A(J) + B1(1)*Q(J-1) + B2(1)*Q(J-2)

T(1) = Q(1)

T(2) = Q(2) + B1(1)*T(1)

DO 250 J = 5,N1

250 T(J-2) = Q(J-2) + B1(1)*T(J-3) + B2(1)*T(J-4)

X = R1(1)*T(N-1) + B2(1)*T(N-2)

CALL PUFF (T,0)

B1(1) = B1(1) + D1

B2(1) = B2(1) + D2

IF (KODE) 260,260,260

260 IF (TOL* DABS(D1) - DABS(B1(1))) 261,261,270

261 IF (TOL* DABS(D2) - DABS(B2(1))) 262,262,270

262 IF (KODE) 263,263,460

263 KODE = 1

264 S1 = DABS(D1)

S2 = DABS(D2)

GO TO 220

265 LVE = LVE - 1

IF (LVE) 220,297,220

270 IF (ITERB - KEY) 220,271,271

271 MURDER = MURDER - 1

IF (MURDER) 479,285,272

272 KEY = KEY + INK

B2(1) = B2(1) + 50*C*B1(1)**2

GO TO 220

280 IF (4.C0*DABS(D1) - S1) 281,410,410

281 IF (4.C0*DABS(D2) - S2) 282,410,410

285 ITERB = 997

290 K = K + 1

KOUNT(K) = ITERB + 10

A(N) = R1(1)

A(N1) = B2(1)

N = N - 2

N1 = N1 - 2

DO 300 I = 1,N1

300 A(I) = 0(I)

IF (DABS(B1(1)) > LT*100*DOSQ(T(DABS(B2(1)))) 310,341,300

181(1) = 100*DOSQ(T(DABS(B2(1))))

GO TO 120

310 DO 320 I = 1,N

320 A(I+1) = X

CBL0337C

CBL0341C

CBL0342C

CBL0343C

CBL0344C

CBL0345C

CBL0346C

CBL0347C

CBL0348C

CBL0349C

CBL0350C

CBL0351C

CBL0352C

CBL0353C

CBL0354C

CBL0355C

CBL0356C

CBL0357C

CBL0358C

CBL0359C

CBL0360C

CBL0361C

CBL0362C

CBL0363C

CBL0364C

CBL0365C

CBL0366C

CBL0367C

CBL0368C

CBL0369C

CBL0370C

CBL0371C

CBL0372C

CBL0373C

CBL0374C

CBL0375C

CBL0376C

CBL0377C

CBL0378C

CBL0379C

CBL0380C

CBL0381C

CBL0382C

CBL0383C

CBL0384C

CBL0385C

CBL0386C

CBL0387C
MURDER = -1

N = N

N1 = N + 1

L = N

K = 0

CALL GROWL(T(N-2),Q(N))

330 IF (L = 1) 440, 340, 400

ITER8 = 0

Q(1) = A(1)

B1(1) = A(2)

390 ITER8 = ITER8 + 1

DO 340 J = 2, N1

340 Q(J) = A(J) + B1(J)* Q(J-1)

T(1) = Q(1)

DO 370 J = 1, N1

370 T(J-1) = Q(J-1) + B1(J) * T(J-2)

D1 = Q(N1) / T(N)

B1(J) = B1(J-1) + D1

IF (DABS(P(J)) > TCL* DABS(D1)) 330, 380, 390

380 IF (ITER8 = 8) 350, 385, 350

385 ITER8 = 0

390 KOUNT(K+1) = ITER8

A0(2) = B1(1)

GO TO 440

400 K = K + 1

KODE = C

B1(J) = A0(J)

B2(J) = A0(J+1)

ITER8 = KOUNT(K)

KEY = ITER8 + 8

IF (D(J-1)) 220, 409, 220

409 ITER9 = ITER9 + 1

410 X = B1(J)* X + 4.00* B2(J)

IF (X) 420, 430, 430

420 A0(J) = .500* DSQRT(-X)

A0(J+1) = .500* 91(1)

KOUNT(K) = -ITER8

L = L - 2

GO TO 330

430 X = DSQRT(X)

IF (B1(J)) 432, 431, 431

432 X = -X

431 A0(J) = .500* (B1(J)+ X)

A0(J+1) = -B2(J)/ A0(J)

433 KOUNT(K) = ITER8

L = L - 2

GO TO 330

440 J = N1

DO 450 I = 1, N

ROOT(I) = A0(J)

A0(J) = A(J)

450 J = J - 1

460 RETURN

END

SUBROUTINE GROWL(A,Y)
DOUBLE PRECISION A(2), B, X(2), Y(2), T
COMMON /BARK/ X, B, NIX
RETURN
ENTRY RUFF
NIX = 0
IF (ABS(SNGL(9)) - ABS(SNGL(A(2)))) 100, 120, 110
100 T = B / A(2)
   X(2) = (T*Y(1) - Y(2)) / (A(2) - T*A(1))
   X(1) = -((A(1)*X(2) + Y(1)) / A(2))
RETURN
110 T = A(2) / B
   X(2) = (T*Y(2) - Y(1)) / (A(1) - T*A(2))
   X(1) = -((A(2)*X(2) + Y(2)) / B)
RETURN
120 IF (SNGL(B)) 110, 130, 110
130 NIX = 1
RETURN
END
Appendix D

Sample Input
<table>
<thead>
<tr>
<th>FILE: INPUT DATA</th>
<th>GRUMMAN DATA SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE INPUT FOR ACTIVE CABLE PPOG-BASIC LONG CHAP. W COMP PBMT OUT</td>
<td></td>
</tr>
<tr>
<td>1 0 0 1 0 2 3 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>10.95</td>
</tr>
<tr>
<td>-1.062</td>
<td>-1.1109</td>
</tr>
<tr>
<td>3780</td>
<td>2380</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>0.72</td>
<td>0.000805</td>
</tr>
<tr>
<td>.8</td>
<td>3.60</td>
</tr>
<tr>
<td>5.0</td>
<td>263.0</td>
</tr>
<tr>
<td>0.0</td>
<td>6.0</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>1.9</td>
<td>2.0</td>
</tr>
<tr>
<td>1.9</td>
<td>2.0</td>
</tr>
<tr>
<td>50.0</td>
<td>5.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>13.8</td>
<td>1.53</td>
</tr>
<tr>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SAMPLE DATA-LONG CHAP OF THE ETA/EMO TRANS FUNC W FEEDBACK &amp; FREQUENCY RESPONSE</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0 2 0 1 0 3 0 0 0 1 1 2 0 0 0 -1 60</td>
<td></td>
</tr>
<tr>
<td>137.7.5</td>
<td>138.100.</td>
</tr>
<tr>
<td>140.100.</td>
<td></td>
</tr>
<tr>
<td>SAMPLE INPUT OF VEL=C. W LIFT CABLE-CHAP. ROOTS OPTION</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0 2 0 9 3 0 0 1 0 1 0 8 0 0</td>
<td></td>
</tr>
<tr>
<td>49.0</td>
<td></td>
</tr>
<tr>
<td>49.0</td>
<td></td>
</tr>
<tr>
<td>SAMPLE INPUT FOR CABLELESS MODEL W TRANSFER FUNCTION OPTION</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0 0 2 0 3 1 0 0 0 0 -1 15 3</td>
<td></td>
</tr>
<tr>
<td>48.865</td>
<td>49.445</td>
</tr>
<tr>
<td>SAMPLE OF ACTIVE CABLE SYSTEM-LAT DIP MODE W TRANS. FUNC. OP.</td>
<td></td>
</tr>
<tr>
<td>1 0 0 0 0 2 0 10 10 0 0 0 1 1 1 2</td>
<td></td>
</tr>
</tbody>
</table>
Appendix E

Sample Output
CABLE GEOMETRY-CABLE NO. 3  CABLE LENGTH = 0.123383E 03 IN

DIP, COS-# DEG  ARM-IN
-0.233372E 03  -0.605809E 01
-0.989999E 02  0.0
-0.143272E 03  -0.909894E 00

CABLE GEOMETRY-CABLE NO. 4  CABLE LENGTH = 0.115261E 03 IN

DIP, COS-# DEG  ARM-IN
-0.129705E 03  -0.605642E 01
-0.809999E 02  0.0
-0.307035E 02  0.906588E 00

ITERATION PARAMETER = 4
ACCZ = -1.0954200E-33
ACCD = 0.97964283E-03
THEDOT = 0.24912750E-03 EAD/SEC

Euler Angle DEFLECTION CABLE TENSION
THETA = 1.02 DEG
DELTA = -1.37 DEG
FPY CAI TENSION = 0.127571E 03 LBS
CPY CAR. TENSION = 0.100211E 03 LBS
ARPM DATA IN BODY AXIS AT EQUI=0.0, CENTER
AFROP( 1) = 0.265E-01 AFROP( 4) = 0.29E-01 AFROP( 5) = -5.8E-5
AFROP( 2) = 0.77E-01 AFROP( 7) = 0.16E-01 AFROP( 8) = -1.3E-5
AFROP( 3) = 0.34E-01 AFROP( 12) = 0.16E-01 AFROP( 14) = 0.93E-5
AFROP( 1) = 0.20E-01 AFROP( 17) = 0.0 AFROP( 19) = -1.0E-5
AFROP( 2) = 0.45E-01 AFROP( 22) = 0.41E-01 AFROP( 23) = 0.21E-5
AFROP( 3) = 0.18E-01 AFROP( 27) = 0.10E-01 AFROP( 28) = 0.10E-01
AFROP( 3) = 0.23E-01 AFROP( 32) = 0.11E-01 AFROP( 33) = 0.50E-03
AFROP( 3) = 0.10E-03 AFROP( 3)

POSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX
1  1  1.994769D 01  -5.993150D 00
2  1  3.565982D 01  1.277751D 01  4.719999D 00
3  1  -3.565982D 01  4.277751D 00  -1.538085D 00
4  1  1.994769D 01  2.095982D 00
5  1  -1.092780D 02  -5.943831D 03
6  2  2.546479D 03  3.331009D 02  -1.573320D 00
7  2  2.546479D 03  4.347667D 01  2.192422D 01
8  2  5.556793D 02
9  1  -1.332134D 00
10 2  1.322979D 00
11 2  7.867465D 00
12 4  7.467465D 00  2.033440D 01  4.719999D 00
13 7  1.914799D 00  5.576770D 01
14 7  1.627135D 00  -7.701374D 02
15 8  1.300000D 00

DEterminant AFROP( 4) = 5.2237613E 05  0.0
DEterminant AFROP( 4) = 5.2237613E 05  0.0
DEterminant AFROP( 4) = 5.2237613E 05  0.0
DEterminant AFROP( 4) = 5.2237613E 05  0.0
DEterminant AFROP( 4) = 5.2237613E 05  0.0
DEterminant AFROP( 4) = 5.2237613E 05  0.0
**Dormant**

**Determinant**

-5.223761E+05 0.0
-5.223761E+05 0.0
-5.223761E+05 0.0
-5.223761E+05 0.0

**Final Matrix**

| 1 1 | -1.494176E 01          |
| 1 2 | 1.014281E+02          |
| 1 3 | 1.325213E 00          |
| 2 1 | -7.160784E 00          |
| 2 2 | 2.080230E+01          |
| 2 3 | 3.205325E 00          |
| 3 1 | -7.477466E 02          |
| 3 2 | -2.507620E 01          |
| 3 3 | 4.183917E+02          |
| 4 1 | -1.985230E+03          |
| 4 2 | -9.679156E 01          |
| 4 3 | -1.477630E+03          |
| 4 4 | -2.855462E+04          |

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6.39+4.77E-03</td>
<td>4.329021E 00</td>
<td>-2.922645E+00</td>
</tr>
<tr>
<td>-8.305477E-03</td>
<td>-4.329021E 00</td>
<td>-2.922648E+00</td>
</tr>
<tr>
<td>-2.351342E 00</td>
<td>1.359896E 01</td>
<td>-9.905350E+00</td>
</tr>
<tr>
<td>-2.351342E 00</td>
<td>-1.359896E 01</td>
<td>-9.905350E+00</td>
</tr>
</tbody>
</table>

**Polynomial**

-4.910945E 05 -1.230982E 04 -2.820763E 04 -5.361116E 02 -1.347842E 02

**Polynomial w Const Term First**

-0.410945E 06 -0.123098E 05 -0.282076E 05 -0.636112E 03 -0.134784E 03

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
<th>T H/D-SEC</th>
<th>1/T H/D</th>
<th>PERIOD-SEC</th>
<th>DNAT-CPS</th>
<th>UNDNAT-CPS</th>
<th>DAMP RATIO</th>
<th>DECAY RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.94993E-02</td>
<td>0.43253E 01</td>
<td>0.62532 02</td>
<td>0.1212E-01</td>
<td>0.1451E 01</td>
<td>0.6990E 00</td>
<td>0.6990E 00</td>
<td>0.1941E-02</td>
<td>0.9879E 00</td>
</tr>
<tr>
<td>-0.35134E 02</td>
<td>0.1360E 02</td>
<td>0.2948E 00</td>
<td>0.3392E 01</td>
<td>0.4620E 00</td>
<td>0.2156E 01</td>
<td>0.2196E 01</td>
<td>0.1704E 00</td>
<td>0.3374E 00</td>
</tr>
</tbody>
</table>
CASE NO. 2 SAMPLE DATA-LONG CHA OF THETA/EMO TRANS FNC W FEEDBACK & FREQ RESP.
FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL
NO SHUBBERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

CODE NO. FOR THIS CASE:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
1 -1 2 0 0 2 0 10 -3 0 0 0 1 71 2 0 0 -1 60 0 0 0 0 0 0

DATA CHANGE
137 7 x 5000
138 100 0 0
140 0 0 0

FREQUENCY RESPONSE COMPUTATION

EM. ATT. CABLE TENSION CABLE TENSION
THETA = 1.03 DEG
DELTA = -1.33 DEG
FRONT CABLE TENSION = 0.127591E 03 LBS
REAR CABLE TENSION = 0.160219E 03 LBS
### LONGITUDINAL STABILITY ###
COMPUTATION OF THE EMD NUMERATOR ROOTS
POLYNOMIAL W. CONST TERM FIRST
0.069837E 63 0.153241E 02 0.523025E 32
REAL IMAGINARY 1/ T M/D-SEC PERIOD-SEC DNATP-CPS UNDNAT-CPS DAMP RATIO DELAY RATIO
-0.146E 00 +0.4304E 01 0.7373E 01 0.2113E 00 0.1460E 01 0.6853E 00 0.602E-01 0.607E 00

COMPUTATION OF THE DENOMINATOR ROOTS

POLYNOMIAL W. CONST TERM FIRST
0.069799E 07 0.391416E 06 0.399395E 06 0.718007E 04 0.150616E 04
0.658129E 01
REAL IMAGINARY 1/ T M/D-SEC PERIOD-SEC DNATP-CPS UNDNAT-CPS DAMP RATIO DECAY RATIO
-0.396E 00 +0.4017E 01 0.173E 01 0.5751E 00 0.1564E 01 0.6394E 00 0.64E 00 0.9875E-01 0.536E 00
-0.1596E 01 +0.1569E 02 0.434E 01 0.2302E 01 0.403E 00 0.249E 01 0.251E 01 0.101E 00 0.527E 00
-0.3194E 03 +0.407E-02 0.217E-02 0.467E 03
<table>
<thead>
<tr>
<th>FREQUENCY (F)</th>
<th>AMP. RATIO (DB)</th>
<th>AMP. VALUE</th>
<th>FREQ. (F)</th>
<th>AMP. PT. (DB)</th>
<th>PHASE (DEG)</th>
<th>AMP. VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.16 C7E-01</td>
<td>-0.794E-02</td>
<td>-0.241E-02</td>
<td>0.169E-01</td>
<td>-0.793E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.17 C7E-01</td>
<td>-0.774E-02</td>
<td>-0.231E-02</td>
<td>0.176E-01</td>
<td>-0.787E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.18 C7E-01</td>
<td>-0.751E-02</td>
<td>-0.221E-02</td>
<td>0.184E-01</td>
<td>-0.779E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.19 C7E-01</td>
<td>-0.727E-02</td>
<td>-0.211E-02</td>
<td>0.192E-01</td>
<td>-0.771E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.20 C7E-01</td>
<td>-0.702E-02</td>
<td>-0.201E-02</td>
<td>0.199E-01</td>
<td>-0.763E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.21 C7E-01</td>
<td>-0.676E-02</td>
<td>-0.191E-02</td>
<td>0.207E-01</td>
<td>-0.755E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.22 C7E-01</td>
<td>-0.649E-02</td>
<td>-0.181E-02</td>
<td>0.215E-01</td>
<td>-0.748E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.23 C7E-01</td>
<td>-0.621E-02</td>
<td>-0.171E-02</td>
<td>0.223E-01</td>
<td>-0.741E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.24 C7E-01</td>
<td>-0.592E-02</td>
<td>-0.161E-02</td>
<td>0.231E-01</td>
<td>-0.733E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.25 C7E-01</td>
<td>-0.562E-02</td>
<td>-0.151E-02</td>
<td>0.239E-01</td>
<td>-0.726E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.26 C7E-01</td>
<td>-0.531E-02</td>
<td>-0.141E-02</td>
<td>0.247E-01</td>
<td>-0.719E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.27 C7E-01</td>
<td>-0.500E-02</td>
<td>-0.131E-02</td>
<td>0.255E-01</td>
<td>-0.712E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.28 C7E-01</td>
<td>-0.469E-02</td>
<td>-0.121E-02</td>
<td>0.263E-01</td>
<td>-0.705E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.29 C7E-01</td>
<td>-0.437E-02</td>
<td>-0.111E-02</td>
<td>0.271E-01</td>
<td>-0.698E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.30 C7E-01</td>
<td>-0.405E-02</td>
<td>-0.101E-02</td>
<td>0.279E-01</td>
<td>-0.692E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.31 C7E-01</td>
<td>-0.373E-02</td>
<td>-0.091E-02</td>
<td>0.287E-01</td>
<td>-0.685E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.32 C7E-01</td>
<td>-0.341E-02</td>
<td>-0.081E-02</td>
<td>0.295E-01</td>
<td>-0.679E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.33 C7E-01</td>
<td>-0.309E-02</td>
<td>-0.071E-02</td>
<td>0.303E-01</td>
<td>-0.673E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.34 C7E-01</td>
<td>-0.277E-02</td>
<td>-0.061E-02</td>
<td>0.311E-01</td>
<td>-0.667E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.35 C7E-01</td>
<td>-0.245E-02</td>
<td>-0.051E-02</td>
<td>0.319E-01</td>
<td>-0.661E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.36 C7E-01</td>
<td>-0.213E-02</td>
<td>-0.041E-02</td>
<td>0.327E-01</td>
<td>-0.655E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.37 C7E-01</td>
<td>-0.181E-02</td>
<td>-0.031E-02</td>
<td>0.335E-01</td>
<td>-0.649E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.38 C7E-01</td>
<td>-0.149E-02</td>
<td>-0.021E-02</td>
<td>0.343E-01</td>
<td>-0.643E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.39 C7E-01</td>
<td>-0.117E-02</td>
<td>-0.011E-02</td>
<td>0.351E-01</td>
<td>-0.637E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.40 C7E-01</td>
<td>-0.085E-02</td>
<td>-0.001E-02</td>
<td>0.359E-01</td>
<td>-0.631E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.41 C7E-01</td>
<td>-0.053E-02</td>
<td>-0.001E-02</td>
<td>0.367E-01</td>
<td>-0.625E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
<tr>
<td>0.42 C7E-01</td>
<td>-0.021E-02</td>
<td>-0.001E-02</td>
<td>0.375E-01</td>
<td>-0.619E-02</td>
<td>-0.657E-02</td>
<td>0.120E-03</td>
</tr>
</tbody>
</table>

**FREQUENCY RESPONSE OF THE THIRD-ORDER TRANSFER FUNCTION**

**STEADY STATE GAIN = 0.1607E-03**
CASE NO. 3
SAMPLE INPUT OF VELHOV.LIFT.CABLE -CHARC. ROOTS OPTION
FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL
NO SNUBBERS
LIFT/ANTI-LIFT CABLE IN
FEEDBACK LOGIC IN
WIND OFF CHARACTERISTICS

CODE NO'S. FOR THIS CASE:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
1 -1 0 0 0 2 0 9 3 0 1 0 1 0 8 0 0 0 0 0 0 0 0 0

DATA CHANGE
4A 7.0 49 1.0

EH ATT, DEFLTN, 8 CABLE TENSION
THETA = -6.50 DEG
DELTA = 0.0 DEG
FR. CAP. TENSION = 0.966780E 02 LBS
FR. CAP. TENSION = 0.100000E 03 LBS
LONGITUDINAL STABILITY ****
POLYNOMIAL W CONST TERM FIRST
0.736854E 06 0.231553E 04 0.692365E 05 0.204398E 03 0.141327E 04
0.448472E 01

imaginary T M/S-SEC 1/T M/S PERIOD-SEC ONATF-CPS Undonat-CPS DAMP RATIO DECAY RATIO
-0.3192E 03 0.2176E-02 0.4690E 03 0.1399E 01 0.0 0.7149E 00 0.7149E 00 0.0 0.1000E 01
0.0 0.4492E 01 0.1000E 06 0.0 0.1399E 01 0.0 0.7149E 00 0.7149E 00 0.0 0.1000E 01
0.0 0.5394E 01 0.1000E 06 0.0 0.1236E 01 0.0 0.8091E 00 0.8091E 00 0.0 0.1000E 01
<table>
<thead>
<tr>
<th>CASE NO. 4</th>
<th>SAMPLE INPUT FOR CABLELESS MODEL W. TRANSFER FUNCTION OPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL</td>
</tr>
<tr>
<td></td>
<td>NO SHURBERS</td>
</tr>
<tr>
<td></td>
<td>NO LIPT/ANTI-LIFT CABLE</td>
</tr>
<tr>
<td></td>
<td>FEEDBACK LOGIC NOT IN</td>
</tr>
<tr>
<td></td>
<td>CABLELESS MODEL CHARACTERISTICS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CODE NOS. FOR THIS CASE:</th>
<th>DATA CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 7 6 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24</td>
<td>2,46500</td>
</tr>
</tbody>
</table>

EH, ATT, DEFLTN, & CABLE TENSION

\[ \theta = 1.03 \text{ deg} \]
\[ \delta = -1.33 \text{ deg} \]

PET CAB. TENSION = 0.127591E 03 LBS
RR CAB. TENSION = 0.160214E 03 LBS

**LONGITUDINAL STABILITY ++**

**COMPUTATION OF X/Y DELE NUMERATOR ROOTS**

POLYNOMIAL w CONST TERM FIRST

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.319421E 07</td>
<td>-0.999774E 06</td>
</tr>
<tr>
<td>0.319421E 07</td>
<td>-0.582850E 04</td>
</tr>
<tr>
<td>0.156821E 04</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**POLYNOMIAL w CONST TERM FIRST**

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27092E 01</td>
<td>0.3692E 02</td>
</tr>
</tbody>
</table>

**LATERAL/DIRECTIONAL STABILITY ++**

**POLYNOMIAL w CONST TERM FIRST**

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.677340E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

**ERROR**

\[ 7.8947341E-02 \]

**REAL**

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.255240E 03</td>
<td></td>
</tr>
<tr>
<td>0.955461E 03</td>
<td></td>
</tr>
<tr>
<td>0.266519E 03</td>
<td></td>
</tr>
</tbody>
</table>

**IMAGINARY**

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.255240E 03</td>
<td></td>
</tr>
<tr>
<td>0.255240E 03</td>
<td></td>
</tr>
<tr>
<td>0.255240E 03</td>
<td></td>
</tr>
</tbody>
</table>

**PERIOD-SEC**

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.1050E 08</td>
<td></td>
</tr>
<tr>
<td>0.1050E 08</td>
<td></td>
</tr>
<tr>
<td>0.1050E 08</td>
<td></td>
</tr>
</tbody>
</table>

**DAMP RATIO**

\[ \text{DECAY RATIO} \]

**THE FOLLOWING EXTRACTED ROOTS HAVE POOR ACCURACY**
SAMPLE OF ACTIVE CABLE SYSTEM - LAT DIP MODE WITH TRANS. FUNC. OP.
FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL
NO SNIPPERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

CODE NO$: FOR THIS CASE:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
DATA CHANGE
0 0

EFLTM.CABLE TENSION
THETA = 1.03 DEG
DELTA = -1.33 DEG
FRONT CABLE TENSION = 0.127591E 03 LBS
REAR CABLE TENSION = 0.100214E 03 LBS
LATERAL/DIRECTIONAL STABILITY
COMPUTATION OF PSI/END NUMBERATOR ROOTS
POLYNOMIAL WITH CONSTANT TERM FIRST

REAL IMAGINARY Y M/0-SEC 1/T M/0 PERIOD-SEC DNATF-CPS UNDNAT-CPS DAMP RATIO DECAY RATIO
-0.1270E 01 -0.2390E 01 0.5680E 00 -0.1760E 01 0.2620E 01 0.3804E 00 0.4271E 00 0.4045E 00 0.8045E-01
-0.1445E 01 -0.3750E 01 0.4640E 00 0.2143E 01 0.1876E 01 0.9968E 00 0.6419E 00 0.3682E 00 0.6303E-01
ACTIVE CABLE MOUNT SYSTEM

DEFINITION OF PULLEY MOTION, $\theta_m, \psi_m$

a) LONGITUDINAL CABLE CONTROL

b) DIRECTIONAL CABLE CONTROL
ACTIVE CABLE MOUNT SYSTEM
LATERAL DIRECTIONAL BLOCK DIAGRAM

$\Delta T_i - \Delta T_o$

$\frac{1}{L}$

$\frac{K_T}{R_d + L_d}$

$E_m$

$E_m + E_{m1}$

$K_r y$

$K + K_r \frac{1}{R_d + L_d}$

$J_m + G_d$

$\frac{1}{L}$

$\frac{1}{L}$

$\mu$

$\psi$

$\phi$

$\phi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$

$\psi$
# Active Cable Mount System

## Extended Longitudinal Matrix

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>ΔTc</td>
<td>κ</td>
<td>ΔTfb</td>
<td>θm</td>
<td>Em</td>
<td>˙θm</td>
<td>Em</td>
<td>ΔTc</td>
<td>Eno</td>
<td>ΔTc</td>
</tr>
</tbody>
</table>

### Basic Matrix

#### Inactive Cable Model Sys (see ref. i)

### Equations

1. \[ m \ddot{x} - \nabla \nabla_x - \frac{\partial F}{\partial \Delta T_c} \Delta T_c = 0 \]
2. \[ m \ddot{z} - \nabla \nabla_z - \frac{\partial F}{\partial \Delta T_c} \Delta T_c = 0 \]
3. \[ I_{yy} \ddot{\theta} - \nabla \nabla \theta - \frac{\partial M_y}{\partial \Delta T_c} \Delta T_c = 0 \]
4. \[ x - \frac{\partial x}{\partial z} z - \frac{\partial x}{\partial \theta} \theta = 0 \]
5. \[ \Delta T_{fb} (2r_d) (R_a + sL_a) - (J_m > 2 + G_s) (R_a + L_a) + 2K_m \theta_m + \theta K_T E_{m_{TOT}} = 0 \]
6. \[ \theta_m \ddot{x} = \left[ \frac{\partial \Delta T_c}{\partial x} x + \frac{\partial \Delta T_c}{\partial z} z + \frac{\partial \Delta T_c}{\partial \theta} \theta \right] = 0 \]
7. \[ E_m = K \theta_m + K \dot{\theta}_m + \theta + K \dot{q} \quad \text{where} \quad q = \dot{\theta} \]
8. \[ \dot{\theta}_m = \theta_{ms} \]
9. \[ E_{m_{TOT}} = E_m + W_{m_{o}} \]
10. \[ \Delta T_c = \Delta T_{f_1} - \Delta T_{f_2} \]

114
**ACTIVE CABLE MOUNT SYSTEM**

**EXTENDED LATERAL-DIRECTIONAL MATRIX**

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>ψ</th>
<th>φ</th>
<th>ΔT fb</th>
<th>y m</th>
<th>E m</th>
<th>E m</th>
<th>ΔT c</th>
<th>E m</th>
<th>ΔT i</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Basic Matrix of Inactive Cable-Mtd. Sys (see ref 1.)**

<table>
<thead>
<tr>
<th></th>
<th>E y</th>
<th>E c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

**Equations**

- **Eq. 1.**
  \[ \ddot{y} - \dot{y}R_y - \frac{\partial V}{\partial \Delta T_c} \Delta T_c = 0 \]

- **Eq. 2.**
  \[ I_x \ddot{\phi} - I_{xx} \dot{\phi} - \Sigma N_o - \frac{\partial N}{\partial \Delta T_c} \Delta T_c = 0 \]

- **Eq. 3.**
  \[ I_{xx} \ddot{\phi} - I_{xx} \dot{\phi} - \Sigma T_o - \frac{\partial J}{\partial \Delta T_c} \Delta T_c = 0 \]

- **Eq. 4.**
  \[ \Delta T_{fb}(2r_d) (R_a + sL_a) - (J_M s^2 + G_s) (R_a + sL_a) + 2K_T k_s v_m + 2K_T E_m \text{TOT} = 0 \]

- **Eq. 5.**
  \[ \dot{y}_m + \left[ \frac{\partial \Delta T}{\partial y} \dot{y} + \frac{\partial \Delta T}{\partial v} \dot{v} + \frac{\partial \Delta T}{\partial \phi} \dot{\phi} \right] = 0 \]

- **Eq. 6.**
  \[ E_m = K_y \dot{y}_m + K_v \dot{v}_m + K_r \dot{\phi} \]

- **Eq. 7.**
  \[ \dot{v}_m - ay_m = 0 \]

- **Eq. 8.**
  \[ \dot{y} - ay = 0 \]

- **Eq. 9.**
  \[ E_{m_{TOT}} = E_m + E_{m_o} \]

- **Eq. 10.**
  \[ \Delta T_c = \Delta T_i \Delta T_{fb} \]
FIGURE 7: SUBROUTINE LAT FLOW CHART
Figure 8. Schematic for Snubbed Model
Figure 8.4: Longitudinal Cable Influence Matrix
<table>
<thead>
<tr>
<th></th>
<th>$Y$</th>
<th>$\psi$</th>
<th>$\phi$</th>
<th>$T$</th>
<th>$\Delta \alpha_x$</th>
<th>$\Delta \alpha_y$</th>
<th>$\Delta \alpha_z$</th>
<th>$\Delta \alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-T \cos \alpha_x$</td>
<td>$T \cos \alpha_z$</td>
<td>$\cos \alpha_y$</td>
<td>$-T \sin \alpha_z$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$-l_x T \cos \alpha_x$</td>
<td>$l_y T \cos \alpha_x$</td>
<td>$l_x \cos \alpha_y$</td>
<td>$l_y \sin \alpha_x$</td>
<td>$-l_x T \sin \alpha_y$</td>
<td></td>
<td></td>
<td>$l_y T \sin \alpha_z$</td>
</tr>
<tr>
<td>3</td>
<td>$-l_y T \cos \alpha_y$</td>
<td>$-l_y T \cos \alpha_y$</td>
<td>$-l_y \cos \alpha_y$</td>
<td>$-l_y \cos \alpha_y$</td>
<td>$-l_y T \sin \alpha_y$</td>
<td></td>
<td></td>
<td>$-l_y T \sin \alpha_z$</td>
</tr>
<tr>
<td>4</td>
<td>$-l_x T \cos \alpha_x$</td>
<td>$-l_x T \cos \alpha_x$</td>
<td>$l_x \cos \alpha_y$</td>
<td>$-l_x \sin \alpha_x$</td>
<td>$l_x T \sin \alpha_y$</td>
<td></td>
<td></td>
<td>$-l_x T \sin \alpha_z$</td>
</tr>
<tr>
<td>5</td>
<td>$-\cos \alpha \cot \alpha$</td>
<td>$l_y \sin \alpha \cot \alpha$</td>
<td>$l_x \cos \alpha \cot \alpha$</td>
<td>$l_y \cos \alpha \cot \alpha$</td>
<td>$-l_x \sin \alpha \cot \alpha$</td>
<td>$l_x \cos \alpha \cot \alpha$</td>
<td>$l_y \cos \alpha \cot \alpha$</td>
<td>$-l_y \sin \alpha \cot \alpha$</td>
</tr>
<tr>
<td>6</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
<td>$-l_x \cos \alpha \cot \alpha$</td>
</tr>
</tbody>
</table>

**Figure 3B: Lateral/Directional Cable Influence Matrix**
Fig. 9 - Reference Center and Lift Cable

Side View of Model

Input Data

Notes: All variables are positive as shown.
FRONT VERTICAL - REAR VERTICAL

FRONT HORIZONTAL - REAR HORIZONTAL

FRONT VERTICAL - REAR HORIZONTAL

FRONT HORIZONTAL - REAR VERTICAL

FIG. 10 - CIRCUIT GROUND - RV
FIG. II - SNUBBER CABLE ARRANGEMENT

NOTE: ALL DISTANCES ARE POSITIVE AS SHOWN.