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
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

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

$\Delta T_i =$ Externally applied tension change $\sim \text{lbs}$.

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

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

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

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

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

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

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

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

$\delta_a =$ Aileron deflection $\sim \text{rad}$.

$\delta_e =$ Elevator deflection $\sim \text{rad}$.

$\delta_r =$ Rudder deflection $\sim \text{rad}$.

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LIST OF SYMBOLS</td>
<td>111</td>
</tr>
<tr>
<td>1.0</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0</td>
<td>ACTIVE FEEDBACK CONTROL SYSTEM LOGIC</td>
<td>2</td>
</tr>
<tr>
<td>3.0</td>
<td>FLYING CABLE SNUBBER SYSTEM</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>LONGITUDINAL AXIS</td>
<td>7</td>
</tr>
<tr>
<td>3.2</td>
<td>LATERAL/DIRECTIONAL AXIS</td>
<td>8</td>
</tr>
<tr>
<td>4.0</td>
<td>ADDITIONAL PROGRAM OPTIONS</td>
<td>9</td>
</tr>
<tr>
<td>4.1</td>
<td>TRANSFER FUNCTIONS</td>
<td>9</td>
</tr>
<tr>
<td>4.1.1</td>
<td>LONGITUDINAL AXIS</td>
<td>9</td>
</tr>
<tr>
<td>4.1.2</td>
<td>LATERAL/DIRECTIONAL AXIS</td>
<td>10</td>
</tr>
<tr>
<td>4.2</td>
<td>FREQUENCY RESPONSE</td>
<td>12</td>
</tr>
<tr>
<td>4.3</td>
<td>WIND OFF CHARACTERISTICS</td>
<td>13</td>
</tr>
<tr>
<td>4.4</td>
<td>CABLELESS MODEL CHARACTERISTICS</td>
<td>13</td>
</tr>
<tr>
<td>5.0</td>
<td>REVISED DATA INPUT</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>APPENDIX A - A Discussion of the Differences in Cable Attachment Points Between the Passive and Active Cable Mount System</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>APPENDIX B - Derivation of Motor and Cable Tension Equations</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>APPENDIX C - Program Listing</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>APPENDIX D - Sample Input</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>APPENDIX E - Sample Output</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>FIGURES</td>
<td>110</td>
</tr>
</tbody>
</table>
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 aerostatically-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 assured 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 \( K_q, K_\theta_m, \) and \( K_\theta \) 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. $Y, \dot{Y}, \phi$ are the perturbation variables. The feedback gains $K_r, K_v$ and $K_m$ 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 $4 \times 4$ 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 $3 \times 3$ 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 is 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 snubbed 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 $3 \times 3$ in $x, z, \theta$ and put in the FXS array. The longitudinal stability is a $3 \times 3$ matrix in $x, z, \psi$. 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 $8 \times 8$ matrix with the form shown in Figure 8B.

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

The lateral-directional stability matrix is a $3 \times 3$ 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 KCODE (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 deter-
mine the numerator roots of the \( \frac{z}{\Delta T_c} \), \( \frac{\theta}{\Delta T_c} \), \( \frac{\Delta T_p}{\Delta T_c} \) and \( \frac{x}{\Delta T_c} \) transfer func-
tions. Setting KODE (14) to zero will determine the denominator roots of 
these transfer functions. Thus the complete transfer function can be deter-
mined. 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, 
\( \frac{\theta}{E_{mo}} \), 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, \( \dot{\alpha} \) defined in Section 4.3, 
can also be determined. The numerators \( \frac{z}{\delta_e} \), \( \frac{\theta}{\delta_e} \) and \( \frac{x}{\delta_e} \) 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 without 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 desires. These are respectively a cable tension change, $\Delta T_c$, a rudder input, $\delta_r$, an aileron input, $\delta_a$, 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, $E_{wo}$, test tension, $\Delta T_t$, rudder input, $\delta_r$, aileron input, $\delta_a$, or a side wind gust, $B_g$.  

11
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 integers 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 KOE (3) to +2. Since a complete transfer function must be generated prior to developing the frequency response data, KOE (14) and KOE (15) or KOE (16) and KOE (17) must be set to non zero values to define the desired transfer function. Two additional parameters, KOE (18) and KOE (19), must be set to define the frequency range and number of points to be computed. KOE (18) set to the order of the lowest frequency to be computed, e.g., KOE (18) set to -1 corresponds to 1 rps and a "+1" corresponds to 10 rps. KOE (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 KOE (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 FREQ where the frequency response data is generated with the aid of subroutine 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 ΔT_p column in figure b. Thus the cableless model option requires a CODE (8) of 3 reflecting a 3 x 3 matrix size.

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

The program will execute this option only if CODE (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.

```plaintext
READ (IR, 150) (TITLE (I), I = 1, 20)          (1)
150 FORMAT (20A4)
READ (IR, 200) (KODE (I), I = 1, 24)           (2)
200 FORMAT (24I3)
```

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

```plaintext
READ (IR, 100) (AERO (I), I = 1, 36)           (3a)
100 FORMAT (6E12.5)
CALL TABIN (1, 36) (See Appendix A, Ref. 1)    (3b)
```

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

```plaintext
READ (IR, 100) (AERO (I), I = 44, 59)          (4)
READ (IR, 100) (AERO (I), I = 66, 130)         (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.

```plaintext
READ (IR, 100) (AERO (I), I = 131, 160)        (6)
```

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

```plaintext
CALL TABIN (1, 2) (See Appendix A, Ref. 1)     (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)  \hspace{1cm} (8)
READ (IR, 200) (KODE (I), I = 1, 24)  \hspace{1cm} (9)
READ (IR, 350) K, VALUE  \hspace{1cm} (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>KODE (1)</td>
<td>-</td>
<td>Run number.</td>
</tr>
<tr>
<td>KODE (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>KODE (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>KODE (4)</td>
<td></td>
<td>Element in &quot;AERO&quot; array to be varied for root locus.</td>
</tr>
<tr>
<td>KODE (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>KODE (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>KODE (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>CODE (8)</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></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 - Closed loop response of Active Cable Mount System (see Section 2.0, 4.1.1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CODE (9)</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></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>
<tr>
<td>NAME</td>
<td>VALUE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>KODE (10)</td>
<td>0</td>
<td>No snubbers.</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Analyze conventional snubbers in unsnubbed 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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, ΔTc. (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, ΔTc. (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 ($\delta$)</td>
</tr>
<tr>
<td>+16</td>
<td></td>
<td>Longitudinal system - numerator and/or frequency characteristics for gust response ($\sigma_0$).</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>+10</td>
<td></td>
<td>Lateral/directional system - numerator and/or frequency characteristics of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inactive cable mount system for tension input ΔTc. (See Section 4.1.2).</td>
</tr>
<tr>
<td>+10</td>
<td></td>
<td>Lateral/directional system - numerator and/or frequency characteristics of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>active cable mount system open loop for tension input, ΔTc. (See Section 4.1.2)</td>
</tr>
<tr>
<td>+11</td>
<td></td>
<td>Lateral/directional system - numerator and/or frequency characteristics of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>active cable mount system close loop for test voltage input E_m0. (See Section 4.1.2)</td>
</tr>
<tr>
<td>+12</td>
<td></td>
<td>Lateral/directional system - numerator and/or frequency characteristics of active cable mount system close loop response for externally applied tension, (ΔT)/ (See Section 4.1.2)</td>
</tr>
<tr>
<td>+14</td>
<td></td>
<td>Lateral/directional system - numerator and/or frequency characteristics for yaw control response (δr).</td>
</tr>
<tr>
<td>+15</td>
<td></td>
<td>Lateral/directional system - numerator and or frequency characteristics for roll control response (δr).</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 ($\dot{\gamma}_g$).</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>CLI</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>CMG</td>
</tr>
<tr>
<td>AERO (13)</td>
<td>1/rad</td>
<td>CDE</td>
</tr>
<tr>
<td>AERO (14)</td>
<td>1/rad</td>
<td>CLE</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>M.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>CMDR</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>AMACH</td>
<td></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>ZIYZ</td>
</tr>
<tr>
<td>AERO (66)</td>
<td>in</td>
<td>WLUF</td>
</tr>
<tr>
<td>AERO (67)</td>
<td>in</td>
<td>WLLF</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>SNUXX***</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>SMLX</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>SNUY</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>SMLZ</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>SNUWL</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>AKSNU</td>
<td>Upper snubber, snubbed spring constant</td>
</tr>
<tr>
<td>AERO (120)</td>
<td>lbs/in</td>
<td>AKSNL</td>
<td>Lower snubber, 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>ADSNL</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>T1SY</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>AKSSET****</td>
<td>Motor torque constant ($K_t$)</td>
</tr>
<tr>
<td>AERO (132)</td>
<td>volts/rad/sec</td>
<td>AKSV</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 ($K_{\theta_m}$).</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>blank</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 TABINL.

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.

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

(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, \( \Delta T_{\text{tot}} \), and the friction in the system. The coulomb and viscous friction can be written as proportional to the pulley rate. (See ref 2.)

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

(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, \( \Delta T_{\text{tot}} \), can be determined.

\[ \Delta T_{\text{tot}} = \frac{1}{r_d} \left\{ \left[ J_M s^2 + Gs + \frac{2K_T s}{K_T r_d} \right] \theta_m - \frac{2K_T E_{m,\text{TOT}}}{R_a + sL_a} \right\} \]

(4)

\( \Delta T_{\text{tot}} \) 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
\[ T_0 - T_1 - (T_0 + \delta T_2) = ma \]  
(5)

For \( ma = 0 \)

\[ -\delta T_1 - \delta T_2 = 0 \]  
(6)

and \( \delta T_2 = -\delta T_1 \)  
(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

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

Substituting results from equation (7) into equation (8)

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

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

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

\[ \delta T_2 = \frac{1}{2r_d} \left\{ \left[ J_{p} s^2 + G s + \frac{2K_v K_m s}{R_a + sL_a} \right] \theta_m - \frac{2K_v E_m s}{R_a + sL_a} \right\} \]  
(10)
APPENDIX C

PROGRAM LISTINGS
C THIS IS THE ACTIVE TRANSMISSION SYSTEM ANALYSIS PROGRAM.
C DEVELOPED JULY 74 TO MAY 75.

COMMON/INPUT/IN
COMMON/DAT/AERO(175), AERO(50), KODE(26), LL
COMMON/SN3AYS(SN3, 3), SN(33), THUSN, THLSN, SNUP(3, 7)
COMMON Z72(22)
COMMON/MSGZ(300)
COMMON/MSGD(15, 15)
COMMON/NAME/NME(16), NVAL(15)
DIMENSION TITLE(25), SAVF(55), SAVE(150), IKWH(160)

EQUIVALENCE(AERO(1), CDB), (AERO(2), CLU), (AERO(3), CMJ),
1 (AERO(4), CD4), (AERO(5), CLA), (AERO(6), CMA),
2 (AERO(7), CD5), (AERO(8), CLB), (AERO(9), CMA),
3 (AERO(10), CD6), (AERO(12), CLC), (AERO(12), CMH),
4 (AERO(13), CON), (AERO(14), CLO), (AERO(15), CMDE),
5 (AERO(16), CMA), (AERO(17), CLD), (AERO(19), CMAR),
6 (AERO(19), CY3), (AERO(20), CLF), (AERO(21), CMA),
7 (AERO(22), CY9), (AERO(23), CLO), (AERO(24), CMA),
8 (AERO(25), CY7), (AERO(26), CLE), (AERO(27), CMA),
9 (AERO(28), CY8), (AERO(29), CLG), (AERO(30), CMA),
A (AERO(31), CYF), (AERO(32), CHA), (AERO(33), CMA),
B (AERO(34), CY5), (AERO(35), CLO), (AERO(36), CMA),
C (AERO(44), XPEF), (AERO(45), ZPEF), (AERO(46), XCG),
D (AERO(47), XCG)

EQUIVALENCE(AERO(48), AVMG), (AERO(49), VDL), (AERO(50), AM)
EQUIVALENCE(AERO(51), SHA), (AERO(52), WIT), (AERO(53), B)
EQUIVALENCE(AERO(54), CGAS), (AERO(55), SW), (AERO(56), XIXZ)
EQUIVALENCE(AERO(57), XIXY), (AERO(58), YIXY), (AERO(59), ZIXZ)
EQUIVALENCE(AERO(60), CLT), (AERO(61), COT), (AERO(62), CMT).

1 (AERO(63), THE4A)
EQUIVALENCE(AERO(64), NLS), (AERO(67), WLLF), (AERO(68), NLSF)
2 (AERO(69), SPL), (AERO(70), YLS), (AERO(71), YLLF),
3 (AERO(72), STF), (AERO(73), STAB), (AERO(74), YHF),
4 (AERO(75), SLHF), (AERO(76), NLCF), (AERO(77), STAGF),
5 (AERO(78), HLCF), (AERO(79), EF), (AERO(80), F)
6 (AERO(81), AF), (AERO(82), AR), (AERO(83), HUCF), (AERO(84), MLCF),
7 (AERO(85), HCF), (AERO(86), HUCF), (AERO(87), MLCF),
8 (AERO(88), DCF), (AERO(89), DCF), (AERO(90), RVF),
9 (AERO(92), RHE), (AERO(94), RHE), (AERO(95), AKF),
A (AERO(96), CDB), (AERO(97), STLYT), (AERO(98), WLLT)
B (AERO(99), TLCT), (AERO(100), AKLFT),
C (AERO(101), ALTX), (AERO(103), ALTZ), (AERO(104), CVP)
EQUIVALENCE(AERO(105), SN), (AERO(106), SNU), (AERO(107), SNU)
1 (AERO(108), SLLX), (AERO(109), SLLY), (AERO(110), SLLZ)
2 (AERO(111), SMLX), (AERO(112), SMLY), (AERO(113), SMLZ)
3 (AERO(114), SMLT), (AERO(115), SMLW), (AERO(116), SMLR)
4 (AERO(117), TSNO), (AERO(118), TSLNO), (AERO(119), AKSNO)
5 (AERO(120), AKSNL), (AERO(121), AKSNR), (AERO(122), AKSNL),
6 (AERO(123), AKSNR), (AERO(124), AKSNR), (AERO(125), AKSNL),
7 (AERO(126), AKSNR), (AERO(127), AKN), (AERO(128), AKN)
8 (AERO(129), AKSN), (AERO(130), AKSN), (AERO(131), AKSN)
EQUIVALENCE(AERO(1), CUP), (AERO(2), CUP), (AERO(3), CUP), (AERO(4), CUP),
1 (AERO(4), CUP), (AERO(5), CUP), (AERO(6), CUP), (AERO(7), CUP).
FILE CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

2  (AEROP(7), CXOP), (AEROP(8), CZOP), (AEROP(9), CMOP), C3LCC560
3  (AEROP(10), CXOP), (AEROP(11), CZOP), (AEROP(12), CMOP), C3LCC560
4  (AEROP(13), CXOP), (AEROP(14), CZOP), (AEROP(15), CMOP), C3LCC560
5  (AEROP(16), CXOP), (AEROP(17), CXOP), (AEROP(18), CMOP), C3LCC560
6  (AEROP(19), CYDP), (AEROP(20), CLDP), (AEROP(21), CMOP), C3LCC560
7  (AEROP(22), CYDP), (AEROP(23), CLDP), (AEROP(24), CMOP), C3LCC560
8  (AEROP(25), CYDP), (AEROP(26), CLDP), (AEROP(27), CMOP), C3LCC560
9  (AEROP(28), CYDP), (AEROP(29), CLDP), (AEROP(30), CMOP), C3LCC560

EQUIVALENCE (SN(1), GX(1), SN(2), GX(2), SN(3), GX(3), SN(4), GX(4), SN(5), GX(5), SN(6), GX(6), SN(7), GX(7), SN(8), GX(8), SN(9), GX(9), SN(10), GX(10), SN(11), GX(11), SN(12), GX(12), SN(13), GX(13), SN(14), GX(14), SN(15), GX(15), ALU)

CASE=1
IP=5
IY=6
LLE=*
D1 1  J=1,50
11 SAVE(J)=9999
D1 12  I=1,150
12 AE1=I=0.
LLE=*
READ(IF,150)(TITLE(I),I=1,20)
READ(IF,250)(KODE(I),I=1,24)
20 FORMAT(26I3)
WRITE(Iw,170) KODE(I), (TITLE(I), I=1,20)
170 FORMAT(1HI, 3X, CASE NG=*, IJ, 4X, 20A4)
CALL FITE
WRITE(Iw,171)(I, I=1,24), (KODE(I), I=1,24)
READ(IF,190)(AEROP(I), I=1,36)
READ(IF,190)(AEROP(I), I=1,36)
GO TO 20
10 CALL TABIN(I1,36,NG)
WRITE(Iw,320) NG
30 FORMAT(A4, 'ERROR IN READING TABLES 1-36, NG=*12)
GO TO 500
20 READ(IF,130)(AEROP(I), I=44,59)
READ(IF,130)(AEROP(I), I=66,120)
IF(KODE(I), GT, 20) READ(IF,130)(AEROP(I), I=131,160)
100 FORMAT(6E12.5)
IF(AEPO(44), EQ, * AND AEROP(44), EQ, *) WRITE(Iw,1003)
1023 FORMAT(25X, 'WIND TYP CHARACTERISTICS')
IF(KODE(I), LT, 1) GO TO 32
CALL TABIN(I1,2,NG)
IF(NG, EQ, *1) GO TO 32
WRITE(Iw,120) NG
421 FORMAT(A4, 'ERROR IN READING SNOTTEF DATA TABLE, NG=*13)
C3LCC560
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

GO TO 500
100 DO 28 I=1,150
28 AER(I)=SAVE(I)
READ(150,CEND=500)(TITLE(I),I=1,20)
150 FORMAT(20A4)
CASE=1
DO 34 J=1,50
34 SAVE(J)=9999.
READ(151)(KODE(I),I=1,24)
WRITE(140)(KODE(I),TITLE(I),I=1,24)
CALL IEEE
IKM(I)=K
IF(K.LT.1)GO TO 22
IKM=IKM+1
AER(I)=VALUE
22 IF(K.LT.37)SAVE(K)=AER(K)
WRITE(140)(AER(I),I=1,24)
WRITE(140)(KODE(I),I=1,24)
171 FORMAT(/// CODE NOS. FOR THIS CASE.**2415.**2415)
WRITE(140)(352)
352 FORMAT(3X,*DATA CHANGE*)
357 FORMAT(3X,12.5)
IF(K.LT.12)GO TO 24
DO 24 I=1,
K=IKM(I)
VALUE=AER(I)
24 WRITE(140),VALUE
351 FORMAT(3X,12.5)
LL=0
32 IF(KODE(7).EQ.0)GO TO 31
DO 31 I=1,36
CALL STINTI(AMACH,O,0,I,I,AER(I),NG)
IF(NG.NE.0)GO TO 40
30 CONTINUE
DO 36 J=1,36
36 IF(SAVE(J).NE.9999.)AER(J)=SAVE(J)
GO TO 28
40 WRITE(140),NG
400 FORMAT(///,*ERROR IN TABLE NO.*14,*NG=*13)
GO TO 500
360 FORMAT(5E10.3)
31 IF(KASE.EQ.1)GO TO 9
WRITE(140),731
801 FORMAT(5X,*INPUT DATA AS SPECIFIED IN AERMODATORY)
WRITE(140),800(I,AER(I),I=1,150)
800 FORMAT(5X,12.5)
9 DO 25 I=1,150
25 SAVE(I)=AER(I)
IF(KODE(3).NE.0)GO TO 48
WRITE(140),801
48 FORMAT(*FREQUENCY RESPONSE COMPUTATION*)
IF(KODE(3).NE.2)GO TO 48
FILE: CABLE FORTRAN

GROMMANN DATA SYSTEMS

42 DO 27 I=1,150
27 AEOO(I)=SAVE(I)
   CALL PUTLOC
   IF(I-LO.900) GO TO 1000
49 CALL TRAN!
   IF(KODE(5),EQ.0) GO TO 49
   WRITE(IW,832)
802 FORMAT(4X,"AERO DATA IN STAB. AXIS AT EQUAT. REF. CENTER")
   WRITE(IW,860)(1,AEOO(I),I=1,150)
49 CALL TRAN!
   CALL TRANS
   IF(KODE(5),EQ.0) GO TO 50
   WRITE(IW,863)
802 FORMAT(4X,"AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER")
   WRITE(IW,864)(1,AEOO(I),I=1,150)
   WRITE(IW,865)(1,AEOO(I),I-1,150)
   WRITE(IW,866)(1,AEOO(I),I=1,150)
604 FORMAT(5(2X,AERO(I),I=1,150))
57 IF(KODE(2)) 73,30,90
70 WRITE(IW,730)
700 FORMAT(15++++ LONGITUDINAL STABILITY 15++++)
   IF(KODE(16),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
   WRITE(IW,737)
   IF(KODE(15),IS) GO TO 3.*
   WRITE(IW,756)NAME(IDN),NAME(IDX)
702 CALL LONG
   IF(KODE(3),EQ.1) GO TO 42
   GO TO 1020
80 WRITE(IW,759)
750 FORMAT(15++++ LATERAL/DIRECTIONAL STABILITY 15++++)
   IF(KODE(16),EQ.0) GO TO 703
   IDX=KODE(16)
   IDN=KODE(17)
   WRITE(IW,761)NAME(IDN),NAME(IDX)
   CALL LAT
   IF(KODE(3),EQ.1) GO TO 42
   GO TO 1000
90 WRITE(IW,760)
   IF(KODE(14),EQ.0) GO TO 704
   IDX=KODE(14)
   IDN=KODE(15)
   IF(KODE(13),NE.-1.) GO TO 708
   IF(KODE(15),EQ.3.) IDN=4
   IF(KODE(15),LE.3.) GO TO 708
   WRITE(IW,767)
   WRITE(IW,768)
704 CALL LONG
   WRITE(IW,759)

FILE: CABLE FORTRAN II

GRUMMAN DATA SYSTEMS

IF(KODE(16),CO.,J) GO TO 705
IDX=KODE(16)
IKN=KODE(17)
WRITE(IW,701)NAME1(IDN),NAME1(IDX)

705 CALL LAT
IF(KODE(3),E9.,1) GO TO 12
GO TO 1000

500 STOP
END

SUBROUTINE OUTLOC
COMMON/INDUT/IV,I2
COMMON/DATE/AERDF(175),AERDF(50),KODE(25),LL
IF(LL.GT.0) GO TO 42
II=KODE(4)
IAY= AERDF(II)*.1
ANDF= AERDF(II)

II=1
WRITE(IW,143) II

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

42 L=1+1
II=KODE(4)
AERDF(II)=ANDF-.5,VAR;Y+L=VAR
IF(L.GT.0) GO TO 44
WRITE(IW,143) KODE(4),AERDF(II)

180 FORMAT(* 2X,5H AERDF(*,13,2H)=G12.5)
RETURN

44 AERDF(II)=ANDF

LL=1
RETURN

END

SUBROUTINE BLOC DATA
COMMON/NAME/NAME(16),NAME1(16)
DATA NAME/* Z **THET** X **DTF** TTHM** ENT**
**TMD**) EM** DT** TTHD**
**DELE** A_LG** NAME1*/ Y **PSI** PH** DTF** PS14**
**EM** DT** TMD**) EM** DT** TTHD**
**DELT** PDEL** DFLT** EM** DT** TMD**) EM** DT** TTHD**
END

SUBROUTINE BLOC (ROOTS,K4A,TFG)
COMMON/INDUT/IV,I2
COMMON / DATE/AERDF(175),AERDF(50),KODE(26),LL
COMMON/PLCT/OM(51),XMP(61),ANSL(61),XMP(61),KV
COMMON/NAME/NAME(16),NAME1(16)
COMPLEX ROOTS(1)
COMPLEX CN(29)
DIMENSION DN(21)
DATA D001.112.5,1.7,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,
16.0,6.5,7.2,7.5,8.0,8.5,9.0,9.5,10.0*/

II=0
IN1=KODE(14)
IN2=KODE(15)
IF(KODE(13),NE.,-1) GO TO 32
IF(KODE(15),E0.,7.) IN2=4
IF(KODE(15),L.E.,3.) GO TO 32

42
KODE(15)=3.
WRITE(*,207)

707 FORMAT(3X,KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION,KODE(15) SET TO 3.1)
32. GO TO 91
ENTRY FREQ2(ROOTS,K4A,TFG)
IL=1
IN1=KODE(15)
IN2=KODE(17)
31 CALL ANP(CYU,C,KN,AMPNO,PHSD,ITYPN)
CALL ANP(ROOTS,?,K4A,AMPNO,PHSD,ITYPD)
TAIN=TGN/TFG
SGN=ABS(TGAIN)/TAIN
IF(AMPNO,NE,0.)SGN=TGAIN/AMPNO/AMPDO
ITYPE=ITYPD-ITYPN
IF(KODE(19),LF,10)GO TO 3
IN=20
IK=1
GO TO 4
3 IF(KODE(19),LE,5)GO TO 5
IN=10
IK=2
GO TO 4
5 IN=5
IK=4
4 INIT=KODE(18)
K=IN*3+1
KV=K
IDX=1
DO 1 I=1,K
1 IDX=IDX+1
IF(IDX,LE,IN)GO TO 2
INIT=INIT+1
IDX=1
2 CM=[(IDX-1)=IK+1]*(10.)*INIT
CALL ANP(CYU,CMV(1,KN,AMPNO,PHSD,IDUM))
CALL ANP(ROOTS,?MV(1,K4A,AMPNO,PHSD,IDUM)
ANP(I)=20.*(ALOGE(AMPNO/AMPDO)+ALOGE(ABS(TGAIN))
ANG(I)=TAIN/AMPNO/AMPDO
ANGLE(I)=(PHSN-PHSD)*57.29578
IF(SGN,LE,5)ANGLE(I)=ANGLE(I)+180.
1 CONTINUE
IF(IN=0.0)WRITE(W,10)NAME(IN),NAME(IN)
IF(IN=NE,0)WRITE(W,10)NAME(IN),NAME(IN)
10 FORMAT(IN,1* FREQUENCY RESPONSE OF THE 1,2X,1A4,4/9,1A4,2X.
1 TRANSFER FUNCTION*)
IF(AMPPO,NE,0.)WRITE(W,17)SGN
IF(AMPPO,NE,0.)WRITE(W,18)ITYPN
17 FORMAT(* STEADY STATE GAIN = *,2X,E11.4,/*)
18 FORMAT(* SYSTEM TYPE = *,2X,14)
IF(IN=GE,2C)GO TO 6
WRITE(W,11)
11 FORMAT(*,2X,F9.0(FPS),*,2X,AMP RAT(D9),*,2X,PHASE(DEG),*)
DO 7 I=1,K
7
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEM

7 WRITE (IW,12)CM(I),AMP(I),ANGLE(I),XMP(I)
12 FORMAT (42X,E11.4,5X,4(2X,E11.4))
       GO TO 8
6 WRITE (IW,11)
13 FORMAT (/*2X,FREQ(RPS)*,2X,AMP RAT(DB)*2X,PHASE(DEG)*2X
1   *AMP. VALUE )7X.
2* FREQ(RPS)*,2X,AMP ET(DB)*2X,PHASE(DEG)*2X,AMP. VALUE *)
6 WRITE (IW,11)
15 FORMAT (5X,4(2X,E11.4))
       GO TO 8
9 WRITE (IW,14)
14 FORMAT (1H1)
       RETURN
       ENTRY FREQ2(ROOTS,K4A,TG)
       KN=K4A
6 WRITE (IW,11)
16 FORMAT (/*2X,FREQ(RPS)*,2X,AMP RAT(DB)*2X,PHASE(DEG)*2X
1 *AMP. VALUE )
2* FREQ(RPS)*,2X,AMP ET(DB)*2X,PHASE(DEG)*2X,AMP. VALUE *)
6 WRITE (IW,11)
18 FORMAT (1H1)
       RETURN
       ENTRY CA
       CNUL(I)=ROOTS(I)
       CONTINUE
       RETURN
6 WRITE (IW,11)
2 CONTINUE
       RETURN
6 WRITE (IW,11)
2 CONTINUE
       RETURN
C DEBUG UNIT(3), INIT
       END
SUBROUTINE ANP(CXU,CM,KX,AMP,ANG,ITYPE)
DIMENSION CXU(2,1)
ITYPE=0
ANG=0
AMP=1.0
IF(KX.EQ.0)RETURN
DO 1 I=1,KX
XQL=-CXU(1,I)
Y1M=CM-CXU(2,I)
AMP=SQRT(XQL*XQL+Y1M*Y1M)*AMP
IF(XQL.EQ.0.AND.Y1M.EQ.0)GO TO 2
ANG=ATAN2(Y1M,XQL)*ANG
1 CONTINUE
2 ANG=ANG
ITYPE=ITYPE+1
1 CONTINUE
RETURN
C DEBUG UNIT(3), INIT(ANG,XQL,Y1M)
       END
SUBROUTINE TRAN
C THIS SUBROUTINE CALCULATES BODY AXIS AERO DATA AT CR FROM STAR.
C AXIS AERO DATA AT CR
COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL
EQUIVALENCE(AERO(1),CDU)(AERO(2),CLU)(AERO(3),CMU)
   1 (AERO(4),CDV)(AERO(5),CLO)(AERO(6),CMP)
   2 (AERO(7),CD(0)(AERO(8),CLO)(AERO(9),CM(1))
   3 (AERO(10),CDO)(AERO(11),CLO)(AERO(12),CM(2))

44
<table>
<thead>
<tr>
<th>FILE: CABLE</th>
<th>FORTRAN T1</th>
<th>GRUMMAN DATA SYSTEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(AERO(13),</td>
<td>CDOE), (AERO(14),</td>
</tr>
<tr>
<td></td>
<td>(AERO(15),</td>
<td>CLAE), (AERO(15),</td>
</tr>
<tr>
<td></td>
<td>(AERO(15),</td>
<td>CNDE),</td>
</tr>
<tr>
<td></td>
<td>(AERO(16),</td>
<td>CDOE),</td>
</tr>
<tr>
<td></td>
<td>(AERO(17),</td>
<td>CLAE), (AERO(18),</td>
</tr>
<tr>
<td></td>
<td>(AERO(18),</td>
<td>CNAD),</td>
</tr>
<tr>
<td></td>
<td>(AERO(19),</td>
<td>CYAB), (AERO(20),</td>
</tr>
<tr>
<td></td>
<td>(AERO(21),</td>
<td>CLAE), (AERO(21),</td>
</tr>
<tr>
<td></td>
<td>(CN3),</td>
<td>CABC130)</td>
</tr>
<tr>
<td>7</td>
<td>(AERO(22),</td>
<td>CYAB), (AERO(23),</td>
</tr>
<tr>
<td></td>
<td>(AERO(24),</td>
<td>CLAE), (AERO(24),</td>
</tr>
<tr>
<td></td>
<td>(AERO(25),</td>
<td>CLAE), (AERO(25),</td>
</tr>
<tr>
<td></td>
<td>(AERO(26),</td>
<td>CLAE), (AERO(26),</td>
</tr>
<tr>
<td></td>
<td>(AERO(29),</td>
<td>CYAB), (AERO(29),</td>
</tr>
<tr>
<td></td>
<td>(AERO(30),</td>
<td>CLAE), (AERO(30),</td>
</tr>
<tr>
<td></td>
<td>(AERO(31),</td>
<td>CYAB), (AERO(31),</td>
</tr>
<tr>
<td></td>
<td>(AERO(32),</td>
<td>CLAE), (AERO(32),</td>
</tr>
<tr>
<td></td>
<td>(AERO(33),</td>
<td>CYAB), (AERO(33),</td>
</tr>
<tr>
<td></td>
<td>(AERO(34),</td>
<td>CYAB), (AERO(34),</td>
</tr>
<tr>
<td></td>
<td>(AERO(35),</td>
<td>CYAB), (AERO(35),</td>
</tr>
<tr>
<td></td>
<td>(AERO(36),</td>
<td>CYAB), (AERO(36),</td>
</tr>
<tr>
<td></td>
<td>(AERO(37),</td>
<td>CYAB), (AERO(37),</td>
</tr>
<tr>
<td></td>
<td>(AERO(38),</td>
<td>CYAB), (AERO(38),</td>
</tr>
<tr>
<td></td>
<td>(AERO(39),</td>
<td>CYAB), (AERO(39),</td>
</tr>
<tr>
<td></td>
<td>(AERO(40),</td>
<td>CYAB), (AERO(40),</td>
</tr>
<tr>
<td></td>
<td>(AERO(41),</td>
<td>CYAB), (AERO(41),</td>
</tr>
<tr>
<td></td>
<td>(AERO(42),</td>
<td>CYAB), (AERO(42),</td>
</tr>
<tr>
<td></td>
<td>(AERO(43),</td>
<td>CYAB), (AERO(43),</td>
</tr>
<tr>
<td></td>
<td>(AERO(44),</td>
<td>CYAB), (AERO(44),</td>
</tr>
<tr>
<td></td>
<td>(AERO(45),</td>
<td>CYAB), (AERO(45),</td>
</tr>
<tr>
<td></td>
<td>(AERO(46),</td>
<td>CYAB), (AERO(46),</td>
</tr>
<tr>
<td></td>
<td>(AERO(47),</td>
<td>CYAB), (AERO(47),</td>
</tr>
</tbody>
</table>

**ALPHA** = **THETA**

SNALF = SINO(ALPHA)

CDALF = COS(ALPHA)

SNSQ = SNALF**2

COSQ = CDALF**2

SNCO = SNALF*COALF

CQ=CU2+2*(CO+COA**2)*SNCO

CLU=CU2+2*(CLA+CLA**2)*SNCO

CDA=CA+(CLA+CLA**2)*THETA

CLA=CA+CD2+CLA**2*THETA

CXYUP=CLA*SNSQ-CDU+COSQ+(CDU+CLU)*SNCO

CZUP=CD+CN5U+CLUD+CLUD**2)*SNCO

CMUPS=CMV*SNALF+CVU*COALF

CZQ=CD+SNALF+COALF

CMQP=CVQ

CZAD=CLAD*COALF+CD2*SNALF

CZADP=-CDA*COALF-CLA*SNALF

CMUP=CUP

CZXP=CLUD+SNALF+COALF

CZXP=-CLUD+SNALF+COALF

CMQP=CUP

CZQP=CLB*SNALF+CVB*COALF

CZQP=-CVB*SNALF+CLB*COALF

CMQP=-CUP

CZQP=CLUD+SNALF+COALF

CMQP=-CLUD+SNALF+(CLUD+CN5)*SNSQ

45
CLRP = ( CNR*SN50 + CLP*COSO - CLP + CNP)*SNCO)
CYRP = ( CYP*SNALF + CYR*COALF)
CNRP = ( CLP*SN50 + CNP*COSO - CLP + CNP)*SNCO)
CLRP = (-CN0*SN50 + CLP*COSO - CLP - CNR)*SNCO)
CYDP = CYDA
CNDP = CLDA*SNALF + CNDA*COALF
CLDP = -CNDA*SNALF + CLDA*COALF
CYDP = CYDR
CNRP = CLDP*SNALF + CNR*COALF
CLRP = -CNDF*SNALF + CLDF*COALF
CNRP = CYPS = CNDS = CLDS*COALF
CLRP = -CNDS*SNALF + CLDS*COALF
RETURN
END

SUBROUTINE TRANSFORMS INERTIA DATA & STABILITY AXIS AERO DATA
C INERTIA TRANSFORMATIONS
X=XCG/12.
Z=ZCG/12.
XIX=YIX+AM*(Z**2)
YIV=YIV+AM*(X**2)+AM*(Z**2)
ZIZ=ZIZ+AV*(X**2)
XIX=ZIX+AM**2

C AERO DATA TRANSFORMATIONS
X=ZREF/(9.9*CBAP)
Z=ZREF/(9.9*CBAF)
CMO=CMO-Z*CD0+X*CL0
CY0=CY0-V1-(CL0+2.*CM0)+2.**X*CL0-Z*CD0+2.**X*Z*CD0
CL0=CL0+2.**X*CD0+4.**Z*CL0
CD0=CD0+2.**X*CD0+4.**Z*CD0
CM0=CM0-Z*CD0+X*CL0
CM0=CM0-Z*CD0+X*CL0
X=XCF/(12.*0)
FILE CABLE FORTRAN T1 GRUMMAN DATA SYSTEMS

\[ Z=\text{ZREF}(12, * B) \]
\[ C1=CNC*x*(2.*CNR*CYR+2.*X*CY8) \]
\[ CLR=CLR*x*(CLD-Z*CYR)-7*CYR \]
\[ CNP=CNP-2.*Z*(CNR*CYR)+X*CYPR \]
\[ CLP=CLP-Z*(CYP-2.*Z*CY8)-2.*Z*CLB \]
\[ CYR=CYR+2.*X*CY8 \]
\[ CYP=CYP-2.*Z*CYR \]
\[ CN3=CN8-X*CYR \]
\[ CNR=CN8+X*CY8 \]
\[ CND=CND+X*CYDA \]
\[ CND=CN8+X*CY8 \]
\[ CL3=CLR*7*CYR \]
\[ CLD=CLD-Z*CY8 \]
\[ RETURN \]

END

SUBROUTINE LATSN

COMMON /INDUT/1, IR

COMMON /DAT/AERO(175), AEROP(50), KDE(26), LL

COMMON /SNU9B/SNU(3, 3), SN(30), THUSN, THLSN, SNUD(3, 3)

COMMON ZZZ(200)

COMMON /DU/DM(10, 10)

EQUIVALENCE (AERO(1:5), SNUX), (AERO(106), SNUY), (AERO(107), SNUZ)

1(AERO(108), SNLX), (AERO(109), SNLY), (AERO(110), SNLZ)

2(AERO(111), SNUST), (AERO(112), SNWU), (AERO(113), SNWZ)

3(AERO(114), SNUBUY), (AERO(115), SNUW), (AERO(116), SNUBW)

4(AERO(117), TUSND), (AERO(118), TULSND), (AERO(119), AKSN)

5(AERO(120), AKSN), (AERO(49), V0), (AERO(51), SH0)

6(AERO(63), TETA), (AERO(121), APSNU), (AERO(122), AOSNL)

EQUIVALENCE (SN(1), GX1), (SN(2), GY1), (SN(3), GZ1)

1(SN(4), GX2), (SN(5), GY2), (SN(6), GZ2)

2(SN(7), GX3), (SN(8), GY3), (SN(9), GZ3)

3(SN(10), GX4), (SN(11), GY4), (SN(12), GZ4)

4(SN(13), THU), (SN(14), THL), (SN(15), ALU)

5(SN(16), ALL)

6(SN(19), THG01), (SN(20), THG02), (SN(21), THG03)

7(SN(22), THG04), (SN(23), THG05), (SN(24), THG06), (SN(25), THG07)

8(SN(26), THG08), (SN(27), THG09), (SN(28), THG10)

9(SN(29), THG11), (SN(30), THG12)

DIMENSION TOPF(3, 3), T0PL(3, 3), BFSN(3, 3), SFTL(3, 3)

COT(180) = 1/TAN(180)

GXY(A, AA, C) = (AA*COT(0)) / 12.

GXSY(A, AA, C, D, E, F) = (AA*COT(0)) / 12.

GXPHI(A, AA, C, D, E, F, G) = (AA*COT(0)) / 12.

GYY(A, AA) = (AA*COT(0)) / 12.

GYSY(A, AA, C, D, E, F) = (AA*COT(0)) / 12.

GYPHI(A, AA, C, D, E, F) = (AA*COT(0)) / 12.

GZY(A, AA, C) = (AA*COT(0)) / 12.

GZSY(A, AA, C, D, E, F, G) = (AA*COT(0)) / 12.

GZPHI(A, AA, C, D, E, F) = (AA*COT(0)) / 12.

ALY(A) = -A

ALSY(A, AA, C, D) = (AA*COT(0)) / 12.

ALTHI(A, AA, C, D) = (AA*COT(0)) / 12.

47
FILE: CABLE FORTRAN T1

CALL DPCS(N(THETA))
IF(KODE(12).EQ.1) CALL DRCUSN(THETA)
DUM(1.2) = TUSN**GXI
DUM(1.3) = TUSN**GZI
DUM(1.5) = -TUSN**SIN(THGY)
DUM(1.7) = GY1
DUM(2.2) = SNUX*TUSNO*GX1/12 + SNUY*TUSNO*GY1/12.
DUM(2.3) = -SNUX*TUSNO*GZ1/12.
DUM(2.4) = -SNUY*TUSNO*SIN(THGX1)/12.
DUM(2.5) = SNUX*TUSNO*SIN(THGY1)/12.
DUM(2.7) = (-SNUX*GY1 + SNUY*GX1)/12.
DUM(3.2) = -SNUZ*TUSNO*GX1/12.
DUM(3.3) = SNUZ*TUSNO*GZ1/12 + SNUY*TUSNO*GY1/12.
DUM(3.5) = -SNUZ*TUSNO*SIN(THGX1)/12.
DUM(3.6) = SNUZ*TUSNO*SIN(THGY1)/12.
DUM(3.7) = (-SNUZ*GZ1 + SNUY*GY1)/12.
DUM(4.1) = GXY(GY1,THGY1,ALU)
DUM(4.2) = GXY(-SNUV,THGX1,-SNUX,GY1,THGX1,ALU)
DUM(4.3) = GXY(-SNUZ,GY1,THGX1,-SNUY,GZ1,THGY1,ALU)
DUM(4.4) = -1.
DUM(5.1) = GYY(THGY1,ALU)
DUM(5.2) = GYY(-SNUV,GX1,THGY1,-SNUX,THGY1,ALU)
DUM(5.7) = GYPH(-SNUZ,THGY1,-SNUY,GZ1,THGY1,ALU)
DUM(5.5) = -1.
DUM(6.1) = GZY(THGY1,ALU)
DUM(6.2) = GZY(-SNUV,GX1,T-ZG1,-SNUX,THGY1,ALU)
DUM(6.7) = GZP(-SNUZ,GY1,THGY1,-SNUY,THGY1,ALU)
DUM(6.6) = -1.
IF(KODE(12).EQ.2) GO TO 1010
CALL DPCS(N(THETA))
Q = 5RHO*Q*0
ALU1 = ALU + 1.
CALL STINT(0,ALU1,0,1,1,TUSN1,NG)
IF(NG+NEC) GO TO 5900
ALU2 = ALU-1.
CALL STINT(0,ALU2,0,1,1,TUSN2,NG)
IF(NG+NEC) GO TO 5900
GO TO 5001
5000 FORMAT(*ITE(1W,5002) NG,ALU,G)
5002 FORMAT(*ERROR IN SNUBER TABLE 1*NG=1,1,12,3X*10,3)
RETURN
5001 CONTINUE
AKTU = (TUSN1-TUSN2)/2.
AKSNU = AKTU
1010 CONTINUE
DUM(1,7) = -1.
DUM(7, 9) = AKSNU*12.
DUM(8, 1) = A1Y(GY1).
DUM(8, 2) = ALSY(-SNUY, Gx2, -SNUX, GY1).
DUM(8, 3) = ALPHI(-SNUZ, GY1, -SNUY, GZ1).
DUM(9, 9) = -1.
IF(KODE(10).EQ.1) GO TO 1015.
D0 1016 I=1, 3.
D0 1016 J=1, 3.
1016 SNUO(I,J)=DUM(1,7)* ACSNU* DUM(9, J)*12.
1015 CALL MASH(3, 9).
D0 1050 I=1, 3.
D0 1050 J=1, 3.
1050 TOTR(I,J) = DUM(I, J).
1050 IF(KODE(10).EQ.1) CALL DCSUSN(THETA).
DUM(1, 2) = -TUSY*Gx2.
DUM(1, 3) = TUSN*GZ1.
DUM(1, 5) = -TUSN*GZ1.
DUM(1, 7) = GY2.
DUM(2, 2) = SNUX*TUSN*GZ2/12, -SNUY*TUSN*GY2/12.
DUM(2, 3) = -SNUX*TUSN*GZ2/12.
DUM(2, 4) = SNUY*TUSN*GZ2/12.
DUM(2, 5) = -SNUY*GZ2/12.
DUM(2, 7) = (-SNUX*GY2- SNUY*Gx2)/12.
DUM(3, 2) = -SNUX*TUSN*Gx2/12.
DUM(3, 3) = SNUX*TUSN*GZ2/12, -SNUY*TUSN*GY2/12.
DUM(3, 4) = -SNUY*TUSN*GZ2/12.
DUM(3, 6) = SNUY*TUSN*GZ2/12.
DUM(3, 7) = (SNUY*GZ2+SNUZ*GY2)/12.
DUM(4, 1) = GXY(GY2, THGX2, ALU).
DUM(4, 2) = GXY(SNUY, THGX2, -SNUX, GY2, THGY2, ALU).
DUM(4, 3) = GXY(-SNUZ, GY2, THGX2, SNUY, GZ2, THGX2, ALU).
DUM(4, 4) = -1.
DUM(5, 1) = GYY(THGY2, ALU).
DUM(5, 2) = GYSY(SNUY, Gx2, THGY2, -SNUX, THGY2, ALU).
DUM(5, 3) = GYSY(-SNUZ, THGY2, SNUY, GZ2, THGY2, ALU).
DUM(5, 5) = -1.
DUM(6, 1) = GZY(GY2, THZG2, ALU).
DUM(6, 2) = GZY(SNUY, Gx2, THZG2, -SNUX, GY2, THZG2, ALU).
DUM(6, 3) = GZPHI(-SNUZ, GY2, THZG2, SNUY, THZG2, ALU).
DUM(6, 4) = -1.
IF(KODE(10).EQ.2) GO TO 1020.
CALL DCSUSN(THETA).
ALU1=ALU+1.
CALL STINT(O, ALU1, 0, 1, 1, TUSN1, NG).
IF(NG, NE, 0) GO TO 5000.
ALU2=ALU-1.
CALL STINT(O, ALU2, 0, 1, 1, TUSN2, NG).
IF(NG, NE, 0) GO TO 5000.
AKTU=(TUSN1-TUSN2)/2.
AKSWU=AKTU.
1020 CONTINUE.
DUM(7, 7) = -1.
DUM(7, 9) = AKSNU*12.
DUM(9, 9) = A1Y(GY2).
DUM(9, 2) = ALSY(SNUY, Gx2, -SNUX, GY2).
FILE: CABLE FORTRAN TI

GRUMAN DATA SYSTEMS

DUM(8,3) = ALPHI(-SNUZ,GY2,SNUY,G22)
DUM(8,4) = -1.
IF(KODE(12) .EQ. 1) GO TO 1025
DO 1026 I=1,3
   1026 SNUD(I,J) = SNUD(I,J) + DUM(I,7) * ALPHI * DUM(3,7) * DUM(3,8) * DUM(8,8) * DUM(8,8) * DUM(8,8)
   DO 1060 J=1,3
      1060 TOPL(I,J) = DUM(I,J)
      IF(KODE(12) .EQ. 1) CALL DRCVX(THETA)
      DUM(1,7) = -TSLNO*G3
      DUM(1,1) = GY3
      DUM(2,2) = SNLX*TLSNO*G3/12. - SNLY*TLSNO*GY3/12.
      DUM(3,3) = -SNLY*TLSNO*G3/12. - SNLY*TLSNO*GY3/12.
      DUM(5,5) = -SNLY*TLSNO*G3/12. - SNLY*TLSNO*GY3/12.
      DUM(7,7) = (-SNLX*GY3-SNLY*GY3)*12.
      DUM(3,2) = SNLX*TLSNO*G3/12.
      DUM(3,3) = SNLX*SNLY*TLSN*G3/12.
      DUM(3,5) = SNLX*SNLY*TLSN*G3/12.
      DUM(7,5) = SNLX*TLSNO* Sin(THEGA3)/12.
      DUM(7,7) = SNLX*TLSNO* Sin(THEGA3)/12.
      DUM(4,4) = -1.
      DUM(5,1) = GY3(THEGA3,FALL)
      DUM(5,2) = GY3(THEGA3,FALL)
      DUM(5,3) = GY3(THEGA3,FALL)
      DUM(5,5) = -1.
      DUM(6,1) = GYZ(GY3,THGZ3,FALL)
      DUM(6,2) = GYZ(GY3,THGZ3,FALL)
      DUM(6,3) = GYZ(GY3,THGZ3,FALL)
      DUM(6,5) = -1.
      IF(KODE(12) .EQ. 1) GO TO 103C
      CALL DRCVX(THETA)
      ALL = ALL + 1.
      CALL STINT(0,ALL,1,C1,1,TLSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALL = ALL - 1.
      CALL STINT(0,ALL,2,C1,1,TLSN2,NG)
      IF(NG.NE.0) GO TO 5000
      AKTL = (TLSN1-TLSN2)/2.
      AKSNL = AKTL

103C CONTINUE

DUM(7,7) = -1.
DUM(7,8) = AKSNL*12.
DUM(8,1) = ALPHI(GY3)
DUM(8,2) = ALPHI(SNLY,GX3,SNLX,GY3)
DUM(8,7) = ALPHI(SNL7,GY3,SNLY,G72)
DUM(9,8) = -1.
IF(KODE(12) .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) * ADSWL * DUM(R, J) * 12.
1035 CALL MASH(3, 8)
DO 1070 I = 1, 3
DO 1070 J = 1, 3
1070 PTL(I, J) = DUM(I, J)
IF(KODE(10) = 0, 1) CALL DRCUSN(HETA)
DUM(I, 1, 2) = TLSNO * GX4
DUM(I, 3) = TLSN0 * GZ4
DUM(I, 1, 5) = TLSNO * SIN(TCY4).
DUM(I, 1, 7) = GY.
DUM(I, 2, 7) = SNLX * TLSNO * GX4/12 * SNLY * TLSNO * GY4/12.
DUM(I, 2, 4) = SNLX * TLSNO * SIN(TCY4)/12.
DUM(I, 2, 5) = SNLX * TLSNO * SIN(TCY4)/12.
DUM(I, 2, 7) = (-SNLY * GY3 + SNLY * GY4)/12.
DUM(I, 3, 2) = SNLX * TLSNO * GX4/12.
DUM(I, 3, 4) = SNLX * TLSNO * GZ4/12.
DUM(I, 3, 5) = SNLX * TLSNO * SIN(TCY4)/12.
DUM(I, 3, 6) = SNLX * TLSNO * SIN(TCY4)/12.
DUM(I, 3, 7) = (-SNLY * GZ3 - SNLZ * GY4)/12.
DUM(I, 4, 1) = GXY(GY4, THG4, ALL).
DUM(I, 4, 2) = GXY(-SNLY, TXG4, SNLX, TXG4, ALL).
DUM(I, 4, 3) = GXY(3NLZ, GY4, TXG4, -SNLY, TXG4, ALL).
DUM(I, 4, 4) = 1.
DUM(I, 5, 1) = GYY(TCY4, ALL).
DUM(I, 5, 2) = GYY(-SNLY, GX4, -SNLY, THG4, ALL).
DUM(I, 5, 3) = GYY(3NLZ, GY4, -SNLY, TXG4, ALL).
DUM(I, 5, 5) = 1.
DUM(I, 6, 1) = GZG(GY4, THG4, ALL).
DUM(I, 6, 2) = GZG(-SNLY, GX4, -SNLY, THG4, ALL).
DUM(I, 6, 3) = GZG(3NLZ, GY4, -SNLY, TXG4, ALL).
DUM(I, 6, 6) = 1.
IF(KODE(10) = 0, 1) GO TO 1046
CALL DCSV1(HETA)
ALL1 = ALL + 1.
CALL STINT(Q, ALL1, C, 1, 1, TLSN1, NG)
IF(NG, NE, C) GO TO 5000
ALL2 = ALL - 1.
CALL STINT(Q, ALL2, C, 1, 1, TLSN2, NG)
IF(NG, NE, C) GO TO 5000
AKTL = (TLSN1 - TLSN2)/2.
AKSNL = AKTL
1040 CONTINUE
DUM(7, 7) = 1.
DUM(8, 1) = AKSNL * 12.
DUM(8, 2) = ALY(GY).
DUM(8, 3) = ALPHI(3NLZ, GY4, -3NLX, GZ4).
DUM(8, 9) = 1.
IF(KODE(10) = 0, 1) GO TO 1045
DO 1046 I = 1, 3
DO 1046 J = 1, 3
1046 SNOD(I, J) = SNOD(I, J) + DUM(I, 7) * ADSWL * DUM(R, J) * 12.
FILED CABLE FORTRAN TI GRUMMAN DATA SYSTEMS

1045 CALL 'BASH(1,8)
DO 1090 I=1,3
DO 1077 J=1,3
1050 BTRG(I,J)= DUM(I,J)
DO 1090 I=1,3
DO 1090 J=1,3
1090 SNU(I,J)= TDPR(I,J)+TOPL(I,J)+GTL(I,J)+GTR(I,J)
IF(KODE(10)*EQ.2) RETURN
DO 1095 I=1,3
DO 1095 J=1,3
1095 SNU(I,J)=0
RETURN
1092 DO 1077 I=1,3
DO 1077 J=1,3
SNU(I,J)=0
RETURN
1094 SNU(I,J)=0
RETURN
SUBROUTINE TRIM
C CABLE SUSPENSION SYSTEM TRIM ROUTINE
COMMON/DOUT/DU,10
COMMON /DAT/ AEROP(175),AEROP(50),KODE(26),LL
COMMON /PLCH/TR,TR,TR,TR,TR,TR
DIMENSION ANG(5,3)
EQUIVALENCE(AEROP(1),CDU),(AEROP(2),CLU),(AEROP(3),CMU)
1 (AEROP(4),CDA),(AEROP(5),CLA),(AEROP(6),CMA)
2 (AEROP(7),CDQ),(AEROP(8),CLQ),(AEROP(9),CMQ)
3 (AEROP(10),CDD),(AEROP(11),CLD),(AEROP(12),CMD)
4 (AEROP(13),CODE),(AEROP(14),CLDE),(AEROP(15),CMDE)
5 (AEROP(16),CDA),(AEROP(17),CLDA),(AEROP(18),CMDA)
6 (AEROP(19),CYF),(AEROP(20),CLF),(AEROP(21),CMF)
7 (AEROP(22),CYP),(AEROP(23),CLP),(AEROP(24),CMF)
8 (AEROP(25),CYP),(AEROP(26),CLP),(AEROP(27),CMF)
9 (AEROP(29),CYDF),(AEROP(29),CLDF),(AEROP(30),CMDF)
A (AEROP(31),CYDA),(AEROP(32),CLDA),(AEROP(33),CMDF)
B (AEROP(34),CYDS),(AEROP(35),CLDS),(AEROP(36),CMDS)
EQUIVALENCE(AEROP(46),XCG),(AEROP(47),ZCG)
EQUIVALENCE(AEROP(48),AMACH),(AEROP(49),VO),(AEROP(50),AM)
EQUIVALENCE(AEROP(51),FHO),(AEROP(52),WU),(AEROP(53),R )
EQUIVALENCE(AEROP(54),CBAR),(AEROP(55),SW),(AEROP(56),XIXZ)
EQUIVALENCE(AEROP(57),XIXX),(AEROP(58),YIYY),(AEROP(59),ZIZZ)
EQUIVALENCE(AEROP(60),CLT),(AEROP(61),COT),(AEROP(62),CUM)
EQUIVALENCE(AEROP(63),THETA)
1 (AEROP(66),WLF),(AEROP(67),WLF),(AEROP(68),WLF)
1 (AEROP(69),WLF),(AEROP(70),WLF),(AEROP(71),WLF)
2 (AEROP(72),STAF),(AEROP(73),STAF),(AEROP(74),STAF)
2 (AEROP(75),RHLF),(AEROP(76),RHLF),(AEROP(77),RHLF)
4 (AEROP(78),SCLF),(AEROP(79),SCLF),(AEROP(80),SCLF)
5 (AEROP(81),AF),(AEROP(82),AF),(AEROP(83),AF)
5 (AEROP(84),HLCF),(AEROP(85),HLCF),(AEROP(86),HLCF)
7 (AEROP(87),DF),(AEROP(88),DF),(AEROP(89),DF)
8 (AEROP(90),RF),(AEROP(91),RF),(AEROP(92),RF)
9 (AEROP(93),RF),(AEROP(94),RF),(AEROP(95),RF)
A (AEROP(96),RF),(AEROP(97),RF),(AEROP(98),RF)
B (AEROP(99),TLET),(AEROP(100),TLET),(AEROP(101),TLET)
FILED: CARD  FORTRAN 77

GRUMM: N  DATA  SYSTEMS

C

(AERO(1), 2), ALTX),  (AERO(103), ALTZ)

EQUIVALENCE(AERO(1), CXUP), (AERO(2), CZUP), (AERO(3), CWUP),

CABCC 390

1

(AERO(4), CXAP), (AERO(5), CZAP), (AERO(5), CMAP),

CABCC 400

2

(AERO(7), CXQP), (AERO(8), CZQP), (AERO(9), CMQP),

CABCC 410

3

(AERO(10), CXQP), (AERO(12), CZQP), (AERO(12), CWQP),

CABCC 420

4

(AERO(13), CXQP), (AERO(14), CZQP), (AERO(14), CWQP),

CABCC 430

5

(AERO(16), CXQP), (AERO(17), CZQP), (AERO(16), CMQP,

CABCC 440

6

(AERO(19), CYP), (AERO(20), CL6P), (AERO(21), CNP),

CABCC 450

7

(AERO(22), CYQP), (AERO(23), CL6P), (AERO(24), CNP),

CABCC 460

8

(AERO(25), CYP), (AERO(26), CL6P), (AERO(27), CNP),

CABCC 470

9

(AERO(28), CYQP), (AERO(29), CL6P), (AERO(31), CNP),

CABCC 480

A

(AERO(31), CYP), (AERO(32), CL6P), (AERO(33), CNP),

CABCC 490

B

(AERO(34), CYQP), (AERO(35), CL6P), (AERO(36), CNP),

CABCC 500

PTD=57.2568

THETA=0.

DELALF=0.1

DFP=1

DARFAM=0.1

DDELTE=0.1

DTHST=0.1

ICNTF=2

FIRST=0.

THINT=C.

ALFIN=THETA

DELINT=F.

THST=THINT

1

IF(V3.EQ.0.0) THST=THST-TR*(COS(ADC(3,1)))*COS(ADC(4,1))/(COS(ADC(1,1))

1)+COS(ADC(2,1))

CABCC 650

1)

VAL5=COS(ADC(3,1))

CABCC 660

VAL6=COS(ADC(4,1))

CABCC 670

VAL7=COS(ADC(1,1))

CABCC 680

VAL8=COS(ADC(2,1))

CABCC 690

ALFAM=ALFIN

CABCC 700

DELTE=DELINT

CABCC 710

Q5=F92*0*5*5

CABCC 720

209

THST1=THST+600

CABCC 730

ALFAM=ALFAM+DALFAM

CABCC 740

DELTE1=DELTE+DELTE

CABCC 750

ICNTF=ICNTF+1

CABCC 760

IF(1.CAGT(100)) GO TO 520

CABCC 770

VAL1=ALFAM*5TD

CABCC 780

VAL2=DELTE*5TD

CABCC 790

VAL3=THST1

CABCC 800

CALL EQUI(ALFAM, DELTE, THST1, F1, G1, H1, I1)

CABCC 810

IF(V3 .NE. 0.0) OR FIRST .NE. 0.0) GO TO 2

CABCC 820

FIRST=1.

GO TO 1

CABCC 830

F1=FIRST .NE. 1.0) FIRST=1.

C

COMPUTES  PARTIALS

ALFAM=ALFAM+DELALF*0.1

CABCC 840

CALL EQUI(ALFAM, DELTE, THST1, F1, G1, H1, I1)

CABCC 850

ALFAM=ALFAM+DELALF

CABCC 860

CALL EQUI(ALFAM, DELTE1, THST1, F2, G2, H2, I1)

CABCC 870

ALFAM=ALFAM+DELALF*0.1

CABCC 880

F1=W0*(F1-F2)/DELALF

CABCC 8920
GALFWD = (G1-G2)/DELALF
HALFWD = (H1-H2)/DELALF
DELE = 0.5*(CLDF+CCS(ALFAWI) + CDDE*SIN(ALFAWI))
GODELO = 0.5*(CLDF+SIN(ALFAWI) - CDDE*COS(ALFAWI))
HOELEOD = OS*CB4*CMDE
THSTI = THSTI+DTF
CALL EQU(ALFAWI, DELTEI, THSTI, F1, G1, H1, I)
THSTI = THSTI+DTF
CALL EQU(ALFAWI, DELTEI, THSTI, F2, G2, H2, I)
GTFST = (G1-G2)/(DTF**2)
HTST = (H1-H2)/(DTF**2)

SET UP ITERATION EQUATIONS

FI = F0 + GALFWD*GALFA + GDELE + DELTE + THST*O
G1 = G0 + GALFWD*GALFA + GDELE + DELTE + GTHST + O
HI = H0 + GALFWD*GALFA + GDELE + DELTE + HTHST + O

AC7 = FI/A
ACCC = G1/A
THEO = H1/V
IF (V0.EQ.0.0) GO TO 42
IF (ABS(ACCT) .LT .CT) GO TO 1005
GO TO 1100

1005 IF (ABS(ACCT) .LT .CT) GO TO 1007
GO TO 1100

1007 IF (ABS(THEO) .LT .CT) GO TO 42

C NOW COMPUTE PARAMETER INCREMENTS FROM MATRIX EQUATIONS

1100 DETRM = GALFWD*THST + GALFA*DELE + GALFWD*GTHST + GALFWD + GALFA*DELE + GTHST
2*HTST
DFA = -(GDELE*GTHST + THST*DELE + GALFWD*GDELE + GALFWD*GTHST + GALFWD + GALFA*DELE + GTHST)
1*DELE + GALFWD*DELE + GALFWD + GALFA*DELE + GTHST
1*HTST

THST = THST + (1*GALFWD*DELE + GALFA*GTHST + GALFA*DELE + GALFA*GTHST + GALFWD*DELE + GALFWD*GTHST + GALFWD + GALFA*DELE + GTHST)
THST = THST + (1*GALFWD*DELE + GALFA*GTHST + GALFA*DELE + GALFA*GTHST + GALFWD*DELE + GALFWD*GTHST + GALFWD + GALFA*DELE + GTHST)

IF (ABS(THEO) .LT .CT) GO TO 1007
GO TO 209

520 WRITE (1, 521)
521 FORAY = IF (TRIM ITERATION EXCEEDS LIMITS)
GO TO 522

42 CALL EQU(ALFAWI, DELTEI, THSTI, FC, GO, HO, I)
522 DO 523 IZ2 = 1, 4
DO 524 IZ2 = 1, 4
523 CONTINUE

524 ANG = 2*PI*IC, 524.5 GO TO 529
STOP
WRITE(IW, 525) IZF, XGLTH(IZF), (ANG(IZF, IZK), ARM(IZF, IZK), IZK = 1.3, 5) CABC149C
525 FORMAT(* CABLE GEOMETRY-CABLE NO., 12, X, CABLE LENGTH-E15.6, 1* IN.**/3, X, DIR, COS, DEG ARM-IN.**, (3(X, E15.6)), */)
CABC149C
524 CONTINUE CABC151C
529 FORMAT(* COMPUTATION OF WIND OFF CONDITION, TRIM ROUTINE NOT USED) CABC155C
WRITE(IW, 526) ICHT, ACCX, ACCX, THEDOT CABC154C
526 FORMAT(* ITERATION PARAMETER =.15*/2X, ACCX =.E15.8, 1/2X, ACCX =.E15.8, 2*F**=15.6, 2X, T) CABC154C
528 WRITE(IW, 527) THEOT, DEFT, TT CABC157C
527 FORMAT(*, VEH: ATT-DELTN.G CABLE TENSION*', R) CABC154C
2, *F**=15.6, LRS*', /
32X, CABC160C
RETURN CABC162C
C DEBG UNIT(3), INIT(VAL1, VAL2, VAL3, FJ, GI, HI)
C 1FALE, XW, HALF=, GDEL0, GDELE, GDELE0, CABC164C
C 2FF, ST, GTHS, T=HST, DAFW, GDEFT, DTHST, CABC165C
C 3ACCX, ACCX, THEDOT, TT, VALS, VAL, VAL7, VAL8) CABC166C
END CABC167C
SUBROUTINE EQUS (THETA, DEFT, FF, GG, HHT, FIRST) CABC201C
C CABLE SUSPENSION SYSTEM TRIM EQUATIONS CABC202C
COMMON /INPUT/IW, IC CABC203C
COMMON /AER(I3(175), AERO(50), KODE(26), LL CABC204C
COMM /P_XCH/TST, XGLTH(5), ACC(5, 3), ARM(5, 3), TR, TLFT, GMY, RALE) XNMX, XNMY, YNMX, YNMY CABC205C
REAL A, CABC206C
EQUIVALENCE(AERO(1), CCM), (AERO(2), CLU), (AERO(3), CMU), CABC207C
1 (AERO( 4), CDA), (AERO(5), CLA), (AERO(6), CIA), CABC208C
2 (AERO( 7), CC0), (AERO(8), CLO), (AERO(9), CM0), CABC209C
3 (AERO(10), CHT), (AERO(11), CLT), (AERO(12), CM1), CABC210C
4 (AERO(13), CDE), (AERO(14), CLE), (AERO(15), CM2), CABC211C
5 (AERO(16), CDA), (AERO(17), CLA), (AERO(18), CM3), CABC212C
6 (AERO(19), CYA), (AERO(20), CLB), (AERO(21), CN4), CABC213C
7 (AERO(22), CYD), (AERO(23), CLP), (AERO(24), CNP), CABC214C
8 (AERO(25), CYF), (AERO(26), CLF), (AERO(27), CN5), CABC215C
9 (AERO(28), CYG), (AERO(29), CLQ), (AERO(30), CNP), CABC216C
A (AERO(31), CYA), (AERO(32), CLA), (AERO(33), CN4), CABC217C
B (AERO(34), CYD), (AERO(35), CLD), (AERO(36), CN5), CABC218C
EQUIVALENCE(AERO(46), XCG), (AERO(47), ZCG) CABC219C
EQUIVALENCE(AERO(48), (AERO (48), YQ), (AERO (49), Y9), (AERO (50), AM) CABC220C
EQUIVALENCE(AERO(51), CM0), (AERO(52), WHT), (AERO(53), B) CABC221C
EQUIVALENCE(AERO(54), YCER), (AERO(55), SW), (AERO(56), XIE), CABC222C
EQUIVALENCE(AERO(57), YXY), (AERO(58), YYY), (AERO(59), ZIFZ) CABC223C
EQUIVALENCE(AERO(60), YOT), (AERO(61), Coy), (AERO(62), CMT) CABC224C
EQUIVALENCE(AERO(63), LFF), (AERO(67), WLLF), (AERO(68), WLLF), CABC225C
1 (AERO(69), WLLF), (AERO(70), WLLF), (AERO(71), WLLF), CABC226C
2 (AERO(72), STAF), (AERO(73), STAF), (AERO(74), STAF), CABC227C
3 (AERO(75), RLHFK), (AERO(76), WLF), (AERO(77), STAC), CABC228C
4 (AERO(78), RLCB), (AERO(79), SE), (AERO(80), SE), CABC229C
5 (AERO(81), AF), (AERO(82), AG), (AERO(83), HUCF), CABC230C
6 (AERO(84), HLCF), (AERO(85), HUCF), (AERO(86), HLCF), CABC231C
7 (AERO(87), DC), (AERO(88), DCR), (AERO(89), ALF), CABC232C
8 (AERO(90), BGF)), (AERO(91), SHF), (AERO(92), RUC), CABC233C
9 (AERO(93), RPH), (AERO(94), TPP), (AERO(95), AKE), CABC234C
A (AERO(96), ALFC), (AERO(97), STORT), (AERO(98), WLLT), CABC235C
FILE: CABLE FORTRAN TI  

GRUMMAN DATA SYSTEMS

B (AERO(99), TLFT0), (AERO(100), AKLFT), (AERO(101), ALLTO)
C (AERO(102), ALTX), (AERO(103), ALTZ)
DATA XNN1, XNN2 /* VERTICAL */ H=SIZNL */
RTD=T7.259
VAL1=HETA
Q = QH0*V0*V0/2.0
64 IND=CODE(6)
GO TO (501, 502, 503, 504).IND
501 XNN1=XNN1
XNN2=XNN2
CALL FPLYH(STAF, WLUF, WLLF, HUCF, HLCF, EF, TVE, THETA, 1)
CALL FPLYH(STAF, BLHF, WLHF, -AF, DCR, CF, HRF, THETA, 3)
GO TO S5
502 XNN1=XNN1
XNN2=XNN2
CALL FPLYH(STAF, BLHF, WLHF, AF, DCF, CF, HRF, THETA, 1)
CALL FPLYH(STAF, WLUF, WLLF, HUCF, HLCF, EF, TVE, THETA, 3)
GO TO S5
503 XNN1=XNN1
XNN2=XNN2
CALL FPLYH(STAF, WLUF, WLLF, HUCF, HLCF, EF, TVE, THETA, 1)
CALL FPLYH(STAF, WLUF, WLLF, HUCF, HLCF, EF, TVE, THETA, 3)
GO TO S5
504 XNN1=XNN1
XNN2=XNN2
CALL FPLYH(STAF, BLHF, WLHF, AF, DCF, CF, HRF, THETA, 1)
CALL FPLYH(STAF, BLHF, WLHF, -AF, DCR, CF, HRF, THETA, 3)
GO TO S5
505 IF(KODE(111), 506, 507, 508)
506 WLLT = WLCL + ALTX*SM(THETA) - ALTZ*CS(THETA)
STLC = STAC - ALTX*CS(THETA) - ALTZ*SM(THETA)
XLGTH(5) = SQRT((WLLTT - WLLT)**2 + (STLTT - STALT)**2)
IF(FIRST, NE, 4) GO TO 12
ELL=XLGTH(5)
12 ELL=XLGTH(5)
TLEF = TLFT0+AKLFT*(ELL-ELLO)
ARM(5, 1)=ALTX
ARM(5, 2)=AF
ARM(5, 3)=ALTZ
FXLT = (TLEF*(STALT - STLTT))/XLGTH(5)
FZLTT = (TLEF*(WLLT - WLLTT))/XLGTH(5)
FZLTA = FZLTT*CS(THETA) - FZLTT*SM(THETA)
FZLTS = FZLTT*SM(THETA) + FZLTT*CS(THETA)
YMLT = (FZLTA*ALTZ - FZLTS*ALTX)/12.
ADD(5, 1)=ARCOS(FZLTA/TLEF)
ADD(5, 2)=3.14159/2.
ADD(5, 3)=ARCOS(FZLTS/TLEF)
GO TO 609
507 FZLTR=C
FZLTB=C
YMLTT=C
XLGTH(5)=C
TLEF=C
DE 13 IA=1, 3
ARM(5, IA)=C.
ADD(5, IA)=C.
13 CONTINUE
508 CALL SNTRIV(FXSN,FZSN,EMS,THETA)
   IF (FIRST .NE. 0) GO TO 510
   IF(KODE.EQ.5) GO TO 512
   WRITE(9,509)YNV1,YNV2
509 FORMAT(* CABLE CONFIGURATION IN MCC0L* /)
   PRINT*, "1* FRONT CABLE IS *",A9,* AND REAR CABLE IS *",A9)
512 EL=XLGTH(3)+XLGTH(4)
510 EL=XLGTH(3)+XLGTH(4)
   TR=TX+AK*R*(EL-ELC)
   ELFT=0*SW*(CLD+CL*THETA+CLD*DE)
   ADRG=0*SW*(COS*THETA+COS*DE)
   FXAIF=AD*AG*COS(THETA)+EILFT*SIN(THETA)
   FZAIF=AD*AG*SIN(THETA)-EILFT*COS(THETA)
   WGTX=-3*2*A*SIN(THETA)
   WGTZ=12*2*A*COS(THETA)
   E+GT=ZCG*WGTX-XCG*GTV/12*
   FPGA=FIF*(COS(ADC(1,1)))*COS(ADC(4,1))
   FZCA=FIF*(COS(ADC(1,3)))*COS(ADC(4,3))
   FZCFH=FIF*(COS(ADC(1,1)))*COS(ADC(2,1))
   FZCFH=FIF*(COS(ADC(1,3)))*COS(ADC(2,3))
   EMOC=.*
   DC 511 I=1,*4
   TENS=TF
   IF(I.GT.2)TENS=TP
   EMOC=EMOC+TENS*(COS(ACC(I,1)))+ARM(I,3)-COS(ACC(I,3)))*ARM(I,1)
511 CONTINUE
   EMOC=EMOC/12.*
   AERD=Q*SW*(BAR*CM+CM*THETA+CMDE*DE)
   FF=FZCFH+FZCF+FZL*TR+FZSN+WGTZ+FZATD
   GGE=XCFH*XCF+FZL*TF+FZSN*GTV*FXAIF:
   H=EMOC*YMLFT*EMS+EMGT*AEF3h
   RETURN
END
SUBROUTINE POLYV(STAV,KLU,WLL,HHU,HHL,EP,PAO,THETA,IF)
COMMON /DAT/AERO(175),AERD(5),KDE(26),LL
COMMON /POLCHA*7D,XLGTH(5),*CC(5,3),ARM(5,3),TR,TLET,TF
EQUVALENCE (AERO(76),WLC),(AERO(77),STACF),(AERO(78),3LC)
RI=3.14159
33 GAMU= ATAN(HHU/EP)
   T1= EP*EP +HHU*HHU
   T2= THETA *GAMU
   IF(IF.EQ.3) T2=GAMU-THETA
   WLU= WLC* SQRT(T1)*SIN(T2)
   T3= WLU -WLU
   T4= ABS(STACF-STAV) - SQRT(T1)*COS(T2)
   XLUP= SQRT(T1*T3+T4*T4)
   XLU= SQRT(XLUP*XLUP -RAD*RAD)
   RUP= ATAN(T3/T4)
   DRU= ATAN(PAD/XLU)
   RETAU=(RUP -DRU)*R TO
   GAML= ATAN(HHL/EP)
   T5= EP*EP +HHL*HHL
   T6= THETA -GAML
   IF(IF.EQ.3) T6=(THETA+GAML)
WLLC = WLC + SORT(T5) * SIN(T6)
T7 = WLLC - WLL
T9 = ARS(STACR - STAV) - SORT(T5) * COS(T6)
XLNP = SORT(T7 + T7 + T8 + T8)
XLN = SORT(XLP + XLNP + RAD + RAD)
BLP = ATAN(T7 / T9)
DLB = ATAN(RAD / XLI)
BETAL = (BLP - DBL) * RTO
IF(IF(0.1)) GO TO 1
XLGTH(3) = XLU
XLGTH(4) = XLL
ADC(1, 1) = BETAU / RTO + THETA * PI
ADC(1, 2) = PI / 2
ADC(1, 3) = I / 2 * ADC(1, 1)
ADC(1, 4) = 1 - (3 * TAL / RTO - THETA)
ADC(2, 1) = PI / 2
ADC(2, 3) = I / 2 * ADC(2, 1)
ARM(1, 1) = EP * RAD * SIN(ADC(1, 1))
ARM(1, 2) = 0
ARM(1, 3) = HNU + RAD * COS(ADC(1, 1))
ARM(1, 4) = EP * RAD * SIN(ADC(1, 1))
ARM(2, 2) = 0
ARM(2, 3) = HML - RAD * COS(ADC(2, 1))
RETURN
END

SURROUNTE RPLYH(STAC, BLG, WLD, XP, YP, ZP, RAD, THETA, IF)
COMMON /DATA/ AERO(175), AERCP(51), KODE(26), LLL
COMMON /LYCHA/PD, XLGTH(5), ADC(5, 3), ARM(5, 3), TR, TLFP, TF
SCHEDULE(AERO(76), WLC), (AERO(77), STAC), (AERO(78), BLCP)
PI = 3, 14159
XWT = STAC - STAD
ZWT = WLC - WLD
X3 = XWT * COS(THETA) - ZWT * SIN(THETA)
Z3 = XWT * SIN(THETA) + ZWT * COS(THETA)
T9 = BLG - YP
TIC = X3 * XP
XLHP = SORT(T9 + T9 + TIC + TIC)
BHIP = ATAN(T9, T10)
XLHI = SORT(XLHP + XLHP - RAD + RAD)
DHI = ATAN(RAD / XLHI)
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

RH1 = B11P - 0B11
T11 = Z3.2P
XL = SORT(XL11 * XL1 + T1 * T11)
TH1 = TIC = AD = COS(B11)
TH9 = T2 = AD = SIN(B11)
IF IF = E0.3167 TO 3
XLGTH(1) = XL
XLGTH(2) = XL
ADC(1,1) = ARCOS(TH1/XL)
ADC(1,2) = ARCOS(TH9/XL)
ADC(1,3) = ARCOS(-11/XL)
ADC(2,1) = -ADC(1,1)
ADC(2,2) = 1-ADC(1,2)
ADC(2,3) = ADC(1,3)
ARM(1,1) = XP = RAD = SIN(B11)
ARM(1,2) = VP = RAD = COS(B11)
ARM(1,3) = 0.
ARM(2,1) = ARM(1,1)
ARM(2,2) = ARM(1,2)
ARM(2,3) = 0.
RETURN
3 XLGTH(3) = XL
XLGTH(4) = XL
ADC(3,1) = ARCOS(TH1/XL)
ADC(3,2) = ARCOS(TH9/XL)
ADC(3,3) = ARCOS(-11/XL)
ADC(4,1) = -ADC(3,1)
ADC(4,2) = 1-ADC(3,2)
ADC(4,3) = ADC(3,3)
ARM(3,1) = XP = RAD = SIN(B11)
ARM(3,2) = VP = RAD = COS(B11)
ARM(3,3) = 0.
ARM(4,1) = ARM(3,1)
ARM(4,2) = ARM(3,2)
ARM(4,3) = 0.
RETURN
END

SUBROUTINE DLGTH(C1, C2, C3, IC, IDX)
C COMMON X1, X2, Y1, Y2, Y3, Y4, X5, X6, Y5, Y6
C COMMON Y1, Y2, Y3, Y4, X1, X2, X3, X4, X5, X6
C COMMON ART, YR, TR, TLFT, TF
C IF (C1 < C1) GO TO 1
C1 = C1 = ADC(1C, 1)
C2 = C2 = ADC(1C, 2)
C3 = (ARM(1C, 1) * COS(ADC(1C, 3)) - ARM(1C, 3) * COS(ADC(1C, 1))) / 12.
RETURN
1 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 DCSGLG(I1, CX1, CZ1, CT, CX2, CZ2, CT2)
C COMMON X1, X2, Y1, Y2, Y3, Y4, X5, X6, Y5, Y6
C COMMON ART, YR, TR, TLFT, TF
CX1 = SIN(ARC(1C, 1)) / XLGTH(1C) * 12.
IF (ADC(1C, 3) = 7.14159) GT 001 GO TO 2

59
FILE: CABLE
FORTRAN 71

GRUMMAN DATA SYSTEMS

XVAL = 1.000
GO TO 1
2 XVAL = COTAN (ADC (IC, 3))
1 C71 = -COS (ADC (IC, 3)) * COTAN (ADC (IC, 1)) / XLGTH (IC) * 12.
ZW = ARM (IC, 1)
C71 = (ZW * SIN (ADC (IC, 1)) * XWT * COS (ADC (IC, 3)) * COTAN (ADC (IC, 1))) / XLGTH (IC)
CX3 = -COS (ADC (IC, 3)) * XVAL / XLGTH (IC) * 12.
C73 = SIN (ADC (IC, 3)) / XLGTH (IC) * 12.
C73 = -(ZW * COS (ADC (IC, 1)) * XVAL + XWT * SIN (ADC (IC, 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 / INOUT / W, I8
COMMON / DAT / AID (175), AERD (50), KODE (26), LL
COMMON / PLYCHAR / XLGTH (5), ADC (5, 3), ARM (5, 3), TR, TLFT, TF
COMMON / JOVDUM / (10, IC)
COMMON / E27 / C4 (30)
EQUIVALENCE (AERD (46), XCG), (AERD (47), ZCG)
EQUIVALENCE (AERD (44), THE), (AERD (49), VO ) (AERD (50), AN)
EQUIVALENCE (AERD (51), W ), (AERD (52), WT) (AERD (53), B )
EQUIVALENCE (AERD (54), CBAR ) (AERD (55), SXZ )
EQUIVALENCE (AERD (57), XIXX ) (AERD (58), YYY ) (AERD (59), ZZ7 )
1 (AERD (95), AKP ) (AERD (100), AKLFT )
EQUIVALENCE (AERD (117), TUCNG ) (AERD (119), AKSW ) (AERD (120), AKSNL )
EQUIVALENCE (AERD (123), AKSY ) (AERD (124), AKPHI ) (AERD (125), AKTHE )
1 (AERD (126), AAZ ) (AERD (127), TISY ) (AERD (128), T3PH )
2 (AERD (129), T3THE ) (AERD (130), TS2AZ )
EQUIVALENCE (AERD (131), AKST ) (AERD (132), AKSIV ) (AERD (133), AJASW )
1 (AERD (134), T3R ) (AERD (135), ELSBA ) (AERD (136), AERD )
2 (AERD (137), AKTHD ) (AERD (138), AKTH ) (AERD (139), CP1P )
7 (AERD (140), AKO )
EQUIVALENCE (AERD (1), CXUP ) (AERD (2), CZUP ) (AERD (3), CWUP )
1 (AERD (4), CXAR ) (AERD (5), CZAR ) (AERD (6), CWAR )
2 (AERD (7), CXDP ) (AERD (8), CZDP ) (AERD (9), CWDP )
3 (AERD (10), CXDP ) (AERD (11), CZDP ) (AERD (12), CWDP )
4 (AERD (13), CXDEF ) (AERD (14), CZDEF ) (AERD (15), CWDEF )
5 (AERD (16), CXADP ) (AERD (17), CZADP ) (AERD (18), CWADP )
6 (AERD (19), CYWP ) (AERD (20), CLDP ) (AERD (21), CNBP )
7 (AERD (22), CYDP ) (AERD (23), CLDP ) (AERD (24), CNBP )
8 (AERD (25), CYDP ) (AERD (26), CLDP ) (AERD (27), CNBP )
9 (AERD (28), CYDF ) (AERD (29), CLDP ) (AERD (30), C1DP )
A (AERD (31), CYWP ) (AERD (32), CLDP ) (AERD (33), CNBP )
B (AERD (34), CYDP ) (AERD (35), CLDP ) (AERD (36), CNBP )

DIMENSION CMAT (14, 14, 3), CMAT (14, 3)
COMMON SNI8/3, SNI8/3, SNI8/3, SNL, SNLS, SNI8/3
COMMON / DOUTH / I1, I2, I3
DIMENSION FXS (3, 4)
DO 10 I=1, 3
FILE: CARLE FORTRAN I1

GRUMMAN DATA SYSTEMS

DO 10 IC K=1,4
10 FXS(J,K)=.5.
DO 1 IC=1,5
DO 3 J=1,10
DO 3 K=1,10
3 DUM(J,K)=.
IF(K<=51)GO TO 649
TENS=TF
IF(TIC.*GT.*2) TENS=TF
IF(TIC.*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(3,6)= -ARM(IC,1)*DUM(2,6)/12.
IF(TIC.*GT.*2) GO TO 2
DUM(1,3)= COS (ADC(IC,1))
DJM(2,3)= COS (ADC(IC,3))
DUM(3,3)= (ARM(IC,3)*DUM(1,3)-ARM(IC,1)*DUM(2,3))/12.
CALL DGLTH (CX,CZ,CT,1.,0)
CALL DGLTH (CXP,CZP,CTP,2.,0)
CX= CX + CXP
X2Z = -(CZ+CZP)/CX
DUM(4,1)= X2Z
XPT = -(CT+CTP)/CX
DUM(4,2)=XPT
DUM(4,4)= -1
CALL PCS_G (IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(5,5),
1DUM(4,1),DUM(6,3))
DUM(5,5)= -1
DUM(6,5)= -1
CALL MASH(3,6)
DO 4 J=1,3
DO 4 K=1,3
4 FXS(J,K)=FXS(J,K)+DUM(J,K)
GO TO 1
2 IF(TIC.*GT.*4) GO TO 5
CALL DGLTH (CX,CZ,CT,3.,0)
CALL DGLTH (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,1)*DUM(1,4)-ARM(IC,1)*DUM(2,4))/12.
CALL PCS_G (IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(5,5),DUM(6,3),
1(DUM(6,1),DUM(6,2))
DUM(4,4)= -1
DUM(5,5)= -1
DUM(6,6)= -1
DUM(7,7)= -1
CALL MASH(3,7)
DO 6 J=1,3
DO 6 K=1,3
IF(K.NE.3)FXS(J,K)=FXS(J,K)+SUM(J,K)
6 IF(K.EQ.3)FXS(J,4)=FXS(J,4)+SUM(J,4)
GO TO 1
5 IF(KODE(11).EQ.0)GO TO 1
CALL DLGTH(DUM(7,3),DUM(7,1),DUM(7,2),5.0)
DUM(4,7)=AKLFT*12.
GO TO 8
1 CONTINUE
C ADD SNUDDER INCREMENTS
CALL LGON
DO 7 J=1,3
FXS(J,1)=FXS(J,1)+SNU(J,2)
FXS(J,2)=FXS(J,2)+SNU(J,3)
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=1,4
85 FRIC(J,K)=0.
DO 86 J=1,3
DO 86 K=1,4
86 SNU(J,K)=0.
C THE CABLE FORCES/MOMENTS PARTIALS ARE COMPLETED
C AEOG DATA IS NOW COMPUTED
649 Q=SHD*VD*VD/2.
QS=Q*S
IF(VD.NE.0.)QS=QS/VD
IF(VD.EQ.0.)QS=0.
XU=CXUF=QS
ZJ=CZUP=QS
SUM=CMUP=QS*CBAF
XA=CXAP=QS
ZA=CZAP=QS
EMA=CMAP=QS*CBAF
IF(VD.NE.0.)XO=CXOP=QS*CBAF/(VD*2).
IF(VD.EQ.0.)XO=0.
IF(VD.NE.0.)ZD=CZOP=QS*CBAF/(VD*2).
IF(VD.EQ.0.)ZD=0.
EMD=CMOP=QS*CBAF/2.
XDE=CXDEP*25
ZDE=CZDEP*8
EMDE=CMDEP=QS*CBAF
IF(VD.NE.0.)XAD=CXADP=QS*CBAF/(VD*2).
IF(VD.EQ.0.)XAD=0.
IF(VD.NE.0.)ZAD=CZADP=QS*CBAF/(VD*2).
IF(VD.EQ.0.)ZAD=0.
EMAD=CMADP=QS*CBAF/(2.*VD)
IF(VD.EQ.0.)EMAD=0.
IF(D=.14
FILE0 CABLE FORTRAN T1

ICOL=14
IORDER=3

42 DO 20 J=1,ICOL
20 CMAT(J,J,K)=CMAT(J,J,K)+C
IF(KODE(10).EQ.3)GO TO 650

C FX EQUATION
CMAT(1,1,1)=FXS(1,1)
CMAT(1,1,2)=-XA-SNUD(1,2)-FRIC(1,5)-FRIC(1,2)
CMAT(1,2,3)=XAD
CMAT(1,2,4)=-FXS(1,2)+WX*CS(THETA)-XAD*VO
CMAT(1,2,5)=-XO-XAD*VO-SNUD(1,3)-FRIC(1,6)-FRIC(1,3)
CMAT(1,2,6)=ZCG*AX/12.
CMAT(1,3,1)=FXS(1,3)
CMAT(1,3,4)=FXS(1,4)
CMAT(1,4,2)=-XU-SNUD(1,1)-FRIC(1,4)-FRIC(1,1)
CMAT(1,4,3)=AX
CMAT(1,5,1)=-XDE

C FZ 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)=AX-ZAD
CMAT(2,2,1)=FXS(2,2)+WX*SIN(THETA)-ZAX*VO
CMAT(2,2,2)=20-7A*AD*VO-SNUD(2,3)-FRIC(2,6)-FRIC(2,3)
CMAT(2,2,3)=XCG*AX/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)=AY-SNUD(3,2)-FRIC(3,5)-FRIC(3,2)
CMAT(3,1,3)=EV*AD-CR4-XCG*AX/12.
CMAT(3,2,1)=FXS(3,2)+WX*VD+ZCG*WT*COS(THETA)/12.
CMAT(3,2,2)=-EV*AD*VD*CR4=SNUD(3,3)-FRIC(3,6)-FRIC(3,7)
CMAT(3,2,3)=YIYY
CMAT(3,3,1)=FXS(3,3)
CMAT(3,4,1)=FXS(3,4)
CMAT(3,4,2)=EUY-SNUD(3,1)-FRIC(3,4)-FRIC(3,1)
CMAT(3,5,1)=EMDF
CMAT(3,5,2)=EYDF

C ELIMINATION OF DTF COL FOR CABLELESS MODEL CHAR.
IF(KODE(10).NE.1,1)GO TO 61

IF(KODE(8).NE.3,1)IFTE(14,R2)
82 FORMAT(5X,'KODE(9) HAS BEEN SET BY PROG. TO 3, FOR CABLELESS MODE')
11L CHARACTERISTICS*)
KODE(9)=3
DO 93 J=1,3
93 CONTINUE
DO 97 J=1,3
97 CMAT(1,3,4)=CMAT(1,4,4)
GO TO 2

C CONSTRAINT EQUATION
81 CMAT(4,1,1)=XP7
FILE: CABLE  FORTRAN TI

GRUMAN DATA SYSTEMS

CMAT(4,2,1) = X PT
CMAT(4,4,1) = 1

C ACTIVE CABLE CONTROL EQS.

IF (KODE(13) .LE. 4 ) GO TO 30
CMAT(1,5,1) = 0
CMAT(2,5,1) = 0
CMAT(3,5,1) = 0
IF (KODE(6) = 0 .OR. KODE(6) .EQ. 3) GO TO 46
IC2 = 4
IC1 = 3
GO TO 47

46 IC2 = 1
IC1 = 2

47 CMAT(1,10,1) = -(COS(ADC(IC2,1)) - COS(ADC(IC1,1)))
CMAT(2,10,1) = -(COS(ADC(IC2,3)) - COS(ADC(IC1,3)))
CMAT(3,10,1) = -(4*sin(IC2,3)*COS(ADC(IC2,1)) - 4*sin(IC2,1)*COS(ADC(IC2,3)) - 1) / 12 + (4*cos(IC1,3)*COS(ADC(IC1,1)) - 4*cos(IC1,1)*COS(ADC(IC1,3))) / 12

C EQ OF MOTOR DYN.
CMAT(5,5,1) = 2 * DS * FSBA
CMAT(5,5,2) = 2 * DR * FSBA
CMAT(5,7,1) = AK*SAT*2
CMAT(5,6,2) = AK*SAT*2 - AK*ST*2 - COM*FSBA
CMAT(5,6,3) = AJSAF*FSBA = GS * FSBA
CMAT(5,8,3) = AJSAF*ELSCA

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,IC)
CMAT(6,6,1) = COSD / 12

C ACTIVE CABLE FEEDBACK EQS.
CMAT(7,2,2) = AKO
CMAT(7,6,1) = AKTH
CMAT(7,6,2) = AKTHD
CMAT(7,9,1) = 1

C TOTAL VOLTAGE EQ F M + EM
CMAT(3,7,1) = 1
CMAT(3,9,1) = 1
CMAT(9,11,1) = 1

C RELATION OF THM TO THMD
CMAT(8,9,1) = 1
CMAT(8,6,2) = 1

C RELATION OF TENSOR TO TDC AND INPUT DT
CMAT(10,5,1) = 1
CMAT(10,10,1) = 1
CMAT(12,12,1) = 1
GO TO 31

C FEEDBACK LOOP EQUATION
30 CMAT(5,2,2) = AKTHF
CMAT(5,5,2) = A4-HE
CMAT(5,6,1) = 1

31 ITH=0
IF (KODE(14) .EQ. 4) GO TO 32

C SUBST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS
IDK = KODE(14)
IDN = KODE(15)
FILE: CABLE FORTRAN 77

GRUMMAN DATA SYSTEMS

IF(IDN.NE.13) GO TO 52
IDN=2
I*HD=13
52 IF(IDX.GT.14) GO TO 38
D0 34 I=1,14
D0 34 K=1,3
CMAT(I,K)=CMAT(I,IDX,K)
34 CMAT(I,IDX,K)=CMAT(I,IDN,K)
GO TO 32
38 D0 37 I=1,14
D0 37 K=1,3
CMAT(K,I)=CMAT(IDX,I)
37 CMAT(IDX,IDX)=0.0
IF(IDX.EQ.16) GO TO 39
CMAT(IDX,1)=XDE
CMAT(1,IDX)=YDE
GO TO 32
39 CMAT(1,IDX)=X
CMAT(2,IDX)=Y
CMAT(3,IDX)=Z
32 N=KODE(1)
65A CALL MATRIX(CMAT,N,ROOTS,K4A,IER).
IF(KODE(1).NE.0) GO TO 35
D0 16 I=1,14
D0 36 K=1,3
36 CMAT(I,IDX,K)=CMAT(I,K)
C 35 IF(KODE(5).NE.0) WRITE(IW,100) IER
C 10 FORMAT(2X,2X,'IER=',13,3X,'SEE SUPP POEF AND PIFMF FOR ERROR CODE')
C THE ROOTS OF THE CHARAC. CMAT ARE IN THE COMPLEX ARRAY 'ROOTS'
C AND THE NUMBER OF ROOTS IS *K4A*
35 K4A=K4A+1
30 ROOTS(K4A)=(C,C,C,C)
D0 71 I=1,K4A
C4(K4A+2-I)=C4(K4A+1-I)
71 CONTINUE
C4(1)=0.
70 CALL PRINTR(IW,ROOTS,K4A)
GO TO 651
65C CONTINUE
C NEW SUBRE EFFECTS
KODE(14)=0
D0 700 IC=1,4
D0 201 I=1,10
D0 201 J=1,10
201 DUM(I,J)=0.
TC=T+T*T*SIN
IF(IC.GT.2) TC=T/3
DUM(1,3)=-TC*COS
DUM(1,4)=-TC*SIN
DUM(1,6)=COS
DUM(2,3)=TC*COS
DUM(2,5)=TC*SIN
DUM(3,6)=-TC*COS
DUM(3,1)=TC*SIN
DUM(4,2)=TC*COS
DUM(4,5)=TC*SIN
CABO 262C
CABO 270C
CABO 2710
CABO 272C
CABO 274C
CABO 275C
CABO 276C
CABO 277C
CABO 278C
CABO 279C
CABO 280C
CABO 281C
CABO 282C
CABO 283C
CABO 284C
CABO 285C
CABO 286C
CABO 287C
CABO 288C
CABO 289C
CABO 290C
CABO 291C
CABO 292C
CABO 293C
CABO 294C
CABO 295C
CABO 296C
CABO 297C
CABO 298C
CABO 299C
CABO 300C
CABO 301C
CABO 302C
CABO 303C
CABO 304C
CABO 305C
CABO 306C
CABO 307C
CABO 308C
CABO 309C
CABO 310C
CABO 311C
CABO 312C
CABO 313C
CABO 314C
CABO 315C
CABO 316C
FILE: CABLE FORTRAN I1
GRUMMAN DATA SYSTEMS

DUM(7,6)=COS(ARC(TC,3))
DUM(3,3)=ARC(TC,3)*DUM(1,3)-ARC(TC,1)*DUM(2,3))/12.
DUM(3,4)=ARC(TC,4)*DUM(1,4)/12.
DUM(3,5)=ARC(TC,1)*DUM(2,5)/12.
DUM(3,6)=ARC(TC,1)*DUM(1,6)-ARC(TC,1)*DUM(2,6))/12.
CALL DCOSLGC(TC,DUM(4,1),DUM(4,2),DUM(4,3),DUM(5,1),DUM(5,2).
1 DUM(5,3))
DUM(4,4)=1.
DUM(5,5)=1.
DUM(6,6)=1.
DUM(6,7)=AKSNUL12.
IF(TC.GT.2) DUM(6,7)=AKSNUL12.
CALL DLGTH(DUM(7,1),DUM(7,2),DUM(7,3),IC,0)
DUM(7,7)=1.
CALL HASM(3,7)
DJ 20? J=1,3
Dj 20? K=1,3
200 FXS(J,K)=FXS(J,K)+DUM(J,K)
600 CONTINUE
CMAT(1,2,2)=XA
CMAT(1,2,3)=XAD
CMAT(1,3,1)=WT*COS(THETA)-XA*V0
CMAT(1,3,2)=X0-XAD*V0
CMAT(1,3,3)=XCG*AM/12.
CMAT(1,1,2)=XU
CMAT(1,1,3)=AW
CMAT(2,2,2)=ZA
CMAT(2,2,3)=AM-ZAD
CMAT(2,3,1)=WT*SIN(THETA)-ZA*V0
CMAT(2,3,2)=ZO-ZAD*V0
CMAT(2,3,3)=XCG*AM/12.
CMAT(2,1,2)=ZU
CMAT(3,2,2)=EMA.
CMAT(3,2,3)=VWV=CBAR-XCG*AM/12.
CMAT(3,3,1)=EMA*V0+XCG*AT*COS(THETA)/12-XCG*WT*SIN(THETA)/12.
CMAT(3,3,2)=(-V0-EWV)CBAR
CMAT(3,3,3)=V06
CMAT(3,1,2)=FNU
CMAT(3,1,3)=XCG*AM/12.
DJ 70? I=1,3
Dj 70? K=1,3
700 CMAT(I,1,J)=CMAT(I,J,1)-FXS(I,J)
Iw=6
N=3
GO TO 655
651 CONTINUE
IF(KODE(3).NE.2)RETURN
IF(KODE(14).EQ.0)GO TO 41
WRITE(Iw,43)
43 FORMAT(/" COMPUTATION OF THE DENOMINATOR FOTES/"
LKO=KODE(14)
KODE(14)=0
CALL FEN1(FOOTS,K4A,C4(K4A+1))
GO TO 42
41 KODE(14)=LKO

66
CALL FREQ(ROOSS,K4A,C4(K4A+1))
RETURN
END

SUBROUTINE FREQ (LOUT,ST,NROOT)
COMMON/FO/CA(30)
DIMENSION ST(2,1)
K4=4000+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/SEC)

1 *D PERIOD-SEC DNAT-CPS UNDNAT-CPS DAMP *

2 *SATIO DECAY RATIC *

NEXT=1
IF (NROOT,GT,0) GO TO 5
WRITE(LOUT,2)
RETURN

5 DO 30 I=1,NROOT
IF (NEXT.EQ.2) GO TO 777
SIG=SIG(1)
AWD=ABS(SIG(2))
THDI=SIG*1.442595
THD=99999
IF (THDI.GT.1.E-5) THD=1/THDI
IF (AWE.EQ.0.) GO TO 531

NEXT=2
WD=-AWD
DNAT=AWD * .159155
PER=99999
IF (DNAT.GT.1.E-5) PER=1/DNAT
UNDNAT=SIG**2+AWD**2 *.159155
DAMP=0
IF (AWD-1.E15 * SIG) 503,504,504

503 DAMP=SIGN (COS( ATAN ( AWD/SIG ) ) , -SIG )

504 CHDI=THDI*PER
DEC=999999
APG=SIG * PER

IF (APG.LT.174.6) DECR=EXP (APG)
WRITE(LOUT,529) SIG,WD,THD,THDI,PEP,DNAT,UNDNAT,DAMP,DECR

520 FORMAT(E12.4,A2X,1H+,F11.4,8E13.4)
GO TO 530

531 WRITE(LOUT,532) SIG,THD,THDI
532 FORMAT(E12.4,14X,2E3.4)
GO TO 530

777 NFXT=1

530 CONTINUE
RETURN
END

SUBROUTINE MASH (NN,N)
COMMON /DU/DUM(10,10)
C NN = FINAL MATRIX SIZE
C N = ORIGINAL MATRIX SIZE
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

INN=IN
D0 1001 LL=1,IN
L=N+1-LL
II=1
JJ=1
D0 1001 I=1,II
D0 1001 J=1,JJ
1001 DUU(I,J)= DUM(I,J)+DUM(L,II)*DUM(I,I)/(-DUM(L,II))
RETURN
END

SUBROUTINE LAT
COMMON /INOUT/TW,TP
COMMON /DAT/AERO(175),AERO(50),KDE(26),LL
COMMON /PLYCHAR/TO,PLT(5),ARC(5,3),ARM(5,3),TR,TLFT,TF
COMMON MS/NUM(17,17)
COMMON AERO/C4(32)

EQUIVALENCE(AERO(46),XCG), (AERO(47),ZCG)
EQUIVALENCE(AERO (63),THETA), (AERO (49),VD), (AERO (50), AM)
EQUIVALENCE(AERO (51),HO), (AERO (52), WT), (AERO (53), R)
EQUIVALENCE(AERO (54),CBAF), (AERO (55),SW), (AERO (56), XIXZ)
EQUIVALENCE(AERO (57),XIXZ), (AERO (58),YIYY), (AERO (59), ZIZZ)
1
(AERO(95),AKY), (AERO(100),AKF), (AERO(102),AKLT)
EQUIVALENCE(AERO (117),TUSY), (AERO (119),AKSNU), (AERO (122),AKSNL)
EQUIVALENCE(AERO (123),AKSY), (AERO (124),AKPHI), (AERO (125),ATHE)
2
(AERO (126),AKAZ), (AERO (127),TISY), (AERO (128),TIPHI), (AERO (129),TZAZ)
EQUIVALENCE(AERO (131),KSRV), (AERO (132),AKSRV), (AERO (133),AKSRW)
1
(AERO (134),SBA), (AERO (135),ELSB), (AERO (136),TPRD)
EQUIVALENCE(AERO (137),AKTHR), (AERO (138),AKTH), (AERO (139),GJDR)
2
(AERO (140),AKO), (AERO (141),AK7), (AERO (142),APOS)
EQUIVALENCE(AERO (143),AKY), (AERO (144),AKYO)
1
EQUIVALENCE(AERO (1),CUP), (AEROP (2),CZUP), (AEROP (3),CMUP)
1
(AEROP(4),C45), (AEROP(5),CZAP), (AEROP(6),CMAP)
2
(AEROP(7),CXP), (AEROP(8),CZP), (AEROP(9),CMP)
3
(AEROP(10),CXP), (AEROP(11),CZP), (AEROP(12),CMP)
4
(AEROP(13),CZXP), (AEROP(14),CZP), (AEROP(15),CMP)
5
(AEROP(16),CA4), (AEROP(17),CZAP), (AEROP(18),CMAP)
6
(AEROP(19),CYP), (AEROP(20),CLP), (AEROP(21),CNP)
7
(AEROP(22),CYP), (AEROP(23),CLP), (AEROP(24),CNP)
8
(AEROP(25),CYP), (AEROP(26),CLP), (AEROP(27),CNP)
9
(AEROP(28),CYP), (AEROP(29),CLP), (AEROP(30),CNP)
4
(AEROP(31),CYDP), (AEROP(32),CLDP), (AEROP(33),CNDP)
A
(AEROP(34),CYDP), (AEROP(35),CLDP), (AEROP(36),CNDP)
8
DIMENSION CHAT(14,14,3),BHAT(14,3)
COMPLEX SNRT(44)
COMMON /SNRT/SNU(3,3),SNV(3,3),THUSN,THLSN,SNUD(3,3)
COMMON /C4GH,FC1C(3,6)
DIMENSION FXS(3,3)
D0 10 JC=1,3
D0 10 K=1,3
10 FXS(J,K)=".
IF(KODE(1)=E0.3)GO TO 650
D0 111 I=1,5
IF(KODE(1)=E0.3)AND.IC.EQ.5)GO TO 1
D0 1 J=1,4
168
DO 3 K=1,N
       3 DUM(J,K)=0.
       TENSS=0.
       IF(IE*G7.2)TEN=0.
       IF(IE*G7.4)TEN=STL.
       CAI=COS(ADC(IE,1)).
       CA2=COS(ADC(IE,2)).
       CA3=COS(ADC(IE,3)).
       CA1=ABS(CAI).&LT.;&LT.;41) CA1=0.
       IF(ABS(CAI).&LT.;&LT.;21) CA2=0.
       CA3=ABS(CAI).&LT.;&LT.;21) CA3=0.
       DUM(1,2)=TEN*CA1.
       DUM(1,3)*=TEN*CA3.
       DUM(1,4)=CA2.
       DUM(1,5)=TEN*SIN(ADC(IE,2)).
       DUM(2,2)=1. &AM(IE,1)*DUM(1,2)-&AM(IE,2)*TEN*CA2)/12.
       DUM(2,3)=&AM(IE,1)*DUM(1,3)/12.
       DUM(2,4)=&AM(IE,2)*CA2-&AM(IE,2)*CA1)/12.
       DUM(2,5)=&AM(IE,2)*TEN*SIN(ADC(IE,1))/12.
       DUM(2,6)=&AM(IE,1)*DUM(1,6)/12.
       DUM(4,4)=1.
       DUM(4,9)=C.
       IF(IE*G7.2)DUM(4,9)=AK1=12.
       IF(IE*G7.4)DUM(4,9)=AK1=12.
       DUM(3,2)=&AM(IE,3)*DUM(1,2)/12.
       DUM(3,3)=&AM(IE,3)*DUM(1,3)-&AM(IE,2)*TEN*CA3)/12.
       DUM(3,4)=&AM(IE,2)*CA3-&AM(IE,3)*CA2)/12.
       DUM(3,5)=&AM(IE,2)*TEN*SIN(ADC(IE,3))/12.
       DUM(3,6)=&AM(IE,3)*DUM(1,6)/12.
       CALL DCONS(IIE,UM(5,1),DUM(5,2),DUM(5,3),DUM(6,1),DUM(6,2),DUM(1,1),DUM(7,1),DUM(7,2),DUM(7,3))
       DUM(5,5)=1.
       DUM(6,6)=1.
       DUM(7,7)=1.
       IF(IE*G7.2)G0 TO 2
       CALL WASH(3,7)
  6  DD 4 J=1,3
   DD 4 K=1,3
  4  FXS(J,K)=FXS(J,K)+DUM(J,K)
     G0 TO 1
  2  IF(IE*G7.4)G0 TO 5
     CALL DLGTH(CY,CPS,CPH,1)
     CALL DLGTH(CYP,CPS,CPH,4)
     DUM(9,1)=CY+CPD
     DUM(9,2)=CPS+CPD
     DUM(9,3)=CPH+CPH
     DUM(9,9)=1.
     CALL WASH(3,8)
     G0 TO 6
  5  IF(IE*G7(11)=7.0)G0 TO 7
     CALL DLGTH(DUM(9,1),DUM(9,2),DUM(9,3),5,1)
     DUM(9,9)=1.
     CALL WASH(3,9)
     G0 TO 6
  1  CONTINUE
FILE: CALB FILE: FORTRAN 71

GRUMMAN DATA SYSTEMS

DIM(3,7) = AFW(IC,2) + TC*SIN(ADC(IC,3))/12.
DIM(3,6) = AFW(IC,3) + DUM(1,6)/12.
DUM(4,4) = 1.
DUM(4,3) = < SNUM 12.
IF(IC.GT.2) DUM(4,8) = AKSNL = 12.
CALL DC05((IC, 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))
DUM(4,6) = 1.
DUM(6,6) = 1.
DUM(7,7) = 1.
DUM(9,8) = 1.
CALL CLGTH(DUM(8,1), DUM(8,2), DUM(8,3), IC, 1)
640 FXS(J,K) = XS(J,K) + DUM(J,K)
660 CONTINUE
C ADD AERO INCREMENTS
620 Q5 = Q5 + VD*VQ
Q5 = Q5 + W
IF(VQ.NE.0.) Q5V = Q5/VQ
IF(VQ.EQ.0.) Q5V = 0.
IF(VQ.NE.0.) OBV = V/(2.*VQ)
IF(VQ.EQ.0.) OBV = 0.
YV = CYRP*J5
ELV = CLAP*Q5V*B
ENV = CNAP*Q5V*Q5
YD = CYRP*Q5*Q5V
ELP = CLAP*Q5V*B
END = CNAP*Q5V*Q5
YF = CYRP*Q5*Q5V
ELA = CLAP*Q5V*Q5
ENDA = CNAP*Q5V*Q5
YDS = CYRP*Q5
ENDS = CNAP*Q5V*Q5
ELDS = CLAP*Q5V*Q5
42 DO 113 I = 1, 14
DO 113 J = 1, 14
DO 113 K = 1, 3
113 CVAT(I,J,K) = 0.0
C Y FORCING EQUATION
CVAT(1,1,1) = FXS(1,1)
CVAT(1,1,2) = YV - SNUD(1,1) + FRIC(1,4) - FRIC(1,1)
CVAT(1,1,3) = MX
CVAT(1,2,1) = FXS(1,2) + YV*VC - W*W*SIN(THETA)
CVAT(1,2,2) = YF - SNUD(1,2) + FRIC(1,5) - FRIC(1,1)
CVAT(1,2,3) = MXCG/12
CVAT(1,3,1) = FXS(1,3) + W*CGS(THETA)
CVAT(1,3,2) = YP - SNUD(1,3) - FRIC(1,4) - FRIC(1,1)
71
CMAT(1,3,3) = AM = 7CG/12.

C YAW EQUATION
CMAT(2,1,1) = EPS(2,1)
CMAT(2,1,2) = ENV-SNUD(2,1)-FRIC(2,1)-FRIC(2,1)
CMAT(2,1,3) = AM = XCG/12.
CMAT(2,2,1) = EPS(2,2) + ENV+VO+XCG+WT*SN(THEA)/12.
CMAT(2,2,2) = ENV-SNUD(2,2)-FRIC(2,1)-FRIC(2,1)
CMAT(2,2,3) = Z1ZZ
CMAT(2,3,1) = EPS(2,3) -XCG+WT*COS(THEA)/12.
CMAT(2,3,2) = ENV-SNUD(2,3)-FRIC(2,6)-FRIC(2,3)
CMAT(2,3,3) = X1XX

C ROLL EQUATION
CMAT(3,1,1) = EPS(3,1)
CMAT(3,1,2) = ELV-SNUD(3,1)-FRIC(3,4) -FRIC(3,1)
CMAT(3,1,3) = AM = 7CG/12.
CMAT(3,2,1) = EPS(3,2) + ELV+VO+XCG+WT*SN(THEA)/12.
CMAT(3,2,2) = ELV-SNUD(3,2)-FRIC(3,5)-FRIC(3,2)
CMAT(3,2,3) = X1XX
CMAT(3,3,1) = EPS(3,3) - XCG*WT*COS(THEA)/12.
CMAT(3,3,2) = ELV-SNUD(3,3)-FRIC(3,6)-FRIC(3,3)
CMAT(3,3,3) = X1XX

C ACTIVE CABLE CONTROL EQUATIONS
IF(KDGE(13).NE.1) GO TO 30
IF(KDGE(6).EQ.1.OR.KDGE(6).EQ.4) GO TO 46
IC2 = 2
IC1 = 1
GO TO 47

46 IC2 = 4
IC1 = 7

47 CMAT(1,1,1) = (XPRC(1C1,2) + COS(ADC(1C1,2)) - COS(ADC(1C1,2)))
CMAT(3,1,1) = (XPRC(1C1,2) * COS(ADC(1C1,2)) - AM*IC2-3) * COS(ADC(1C1,2) - 2) / 21
CMAT(2,1,1) = (XPRC(1C1,2) * COS(ADC(1C1,2)) - AM*IC2-3) * COS(ADC(1C1,2) - 2) / 21
CMAT(5,1,1) = 2S3D/12.

C EQ. OF MOTION.
CMAT(4,4,1) = 2*EPS+EPS A
CMAT(4,4,2) = 2*EPS+EPS A
CMAT(4,6,1) = 2AKS3T4.
CMAT(4,5,2) = 2AKS4T+2*AKS4V+3MPC*EPS A
CMAT(4,5,3) = 2AKS4T+2*AKS4V+3MPC*EPS A
CMAT(4,7,1) = 2AKS4T+2*AKS4V+3MPC*EPS A
CALL D166H(CMAT(5,1,1),CMAT(5,2,1),CMAT(5,3,1),IC1,1)
CMAT(5,5,1) = 2S3D/12.

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

C FEEDBACK CONTROL EQ.
CMAT(6,2,2) = AKSP
CMAT(6,5,1) = AKY
CMAT(6,7,1) = AKY
CMAT(6,9,1) = 1.

C RELATE ANGULAR RATES TO ANGULAR DISPLACEMENTS

FILE: CABLE FORTRAN T1 GRUMMAN DATA SYSTEMS
CARC2730
CARC2740
CARC2750
CARC2760
CARC2770
CARC2780
CARC2790
CARC2800
CARC2810
CARC2820
CARC2830
CARC2840
CARC2850
CARC2860
CARC2870
CARC2880
CARC2890
CARC2900
CARC2910
CARC2920
CARC2930
CARC2940
CARC2950
CARC2960
CARC2970
CARC2980
CARC2990
CARC3000
CARC3010
CARC3020
CARC3030
CARC3040
CARC3050
CARC3060
CARC3070
CARC3080
CARC3090
CARC3100
CARC3110
CARC3120
CARC3130
CARC3140
CARC3150
CARC3160
CARC3170
CARC3180
CARC3190
CARC3200
CARC3210
CARC3220
CARC3230
CARC3240
CARC3250
CARC3260
CARC3270

72
C RELATION OF DTC TO DT AND DTFB
CMAT(10, 4, 1) = 1.
CMAT(11, 10, 1) = 1.
CMAT(10, 12, 1) = 1.
GO TO 31

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

C AILERON FEEDBACK LOOP
CMAT(5, 3, 2) = AKPHI
CMAT(5, 5, 2) = T3PHI
CMAT(5, 5, 1) = 1.
CMAT(1, 4, 1) = OS*CYDEP
CMAT(1, 4, 1) = OS*CYDEP
CMAT(2, 4, 1) = OS*3*CNDEP
CMAT(2, 5, 1) = OS*3*CNDEP
CMAT(3, 4, 1) = OS*3*CNDEP
CMAT(3, 5, 1) = OS*3*CNDEP

31 IF (KODE(16) .EQ. 0) GO TO 32
C SUBST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS
IDX = KODE(16)
IDN = KODE(17)
IF (IDX .LT. 13) GO TO 38
DO 34 I = 1, 14
DO 34 K = 1, 3
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
CMAT(I, K) = CMAT(I, IDN, K)
37 CMAT(I, IDN, K) = 0.0
IF (IDX .EQ. 15) GO TO 39
IF (IDX .EQ. 16) GO TO 41
CMAT(1, IDN, 1) = YDR
CMAT(2, IDN, 1) = ENSF
CMAT(3, IDN, 1) = CLDP
GO TO 32

39 CMAT(1, IDN, 1) = YDA
CMAT(2, IDN, 1) = ENDA
CMAT(3, IDN, 1) = ELDA
GO TO 32

41 CMAT(1, IDN, 1) = YV
CMAT(2, IDN, 1) = ENV
CMAT(3, IDN, 1) = ELV
GO TO 32

37 N = KODE(9)
CALL MATRIX(CMAT, N, PROTS, K44, IFR)
IF (KODE(16) .NE. 0) GO TO 35
DO 36 I = 1, 14
DO 36 K = 1, 3

FILED CABLE FORTRAN T1 GRUMAN DATA SYSTEMS

36 C1AT(1,1ON.K) = 0.1AT(I.K) CABC3374
C 35 IF(KODE(5).NE.0) WRITE(1,0) IEF CABC3440
C 10 FORMAT(2X,15E13.3) STOP. POWER AND PRESS FOR ERROR CODES) CABC3550
C THE Roots OF THE CHARACTERISTIC EQUAT. ARE IN THE COMPLEX ARRAY CABC3580
C *ROTS* AND THE NUMBER OF ROOTS IS *K4A*
35 K4A=K4A+1
CALL PRTRS(IW,SN7S,K4A)
IF(KODE(3).NE.2) RETURN
IF(KODE(16).EQ.1) GO TO 44
WRITE(IW,49)
43 FORMAT(/// COMPUTATION OF THE DENOMINATOR ROOTS///)
L=KODE(KODE(16))
KODE(16)=L*C
CALL FREQ1(ROOTS,K4A,C4(K4A+1))
GO TO 42
44 KODE(16)=L*KODE
CALL FREQ2(ROOTS,K4A,C4(K4A+1))
RETURN
END

SUBROUTINE DCMOD(IC,CY1,CPS11,CMH1,CY2,CPS12,CMHI2,CY3,CPS13)
COMMON /PLSCHAV/ST,XLGT(5),ADC(5.3),ARM(5.3),TP,TLT,TF
I=ABS(ADC(IC,3)-3.14159)*GT.*COOIC)GO TO 2
XVAL=I*IC
GO TO 1
2 XVAL=COTAN(ADC(IC,3))
1 XWT=ARM(IC,1)
YWT=ARM(IC,2)
ZWT=ARM(IC,3)
CY1=-COS(ADC(IC,1))*COTAN(ADCIC,1))/XLGT(1C)*12.
CPS11=--(YWT*SIN(ADC(IC,11)*XWT*COS(ADCIC,2))*COTAN(ADC(IC,1)))
/XLGT(1C)
CMH11=(XWT*COS(ADCIC,2))*COTAN(ADCIC,1)-YWT*COS(ADCIC,3))*
COTAN(ADCIC,1))/XLGT(1C)
CY2=SIN(ADC(IC,2))/XLGT(1C)*12.
CPS12=(YWT*COS(ADCIC,2))*COTAN(ADCIC,2)+XWT*SIN(ADCIC,2))/
XLGT(1C)
CMH12=-(XWT*SIN(ADCIC,2))*YWT*COS(ADCIC,3)*COTAN(ADCIC,2))
/XLGT(1C)
CY3=-COS(ADCIC,2))*XVAL/XLGT(1C)*12.
CPS13=(YWT*COS(ADCIC,1))*XVAL-XWT*COS(ADCIC,2)*)
/XLGT(1C)
CMH13=(XWT*COS(ADCIC,2))*XVAL+YWT*SIN(ADCIC,3))
/XLGT(1C)
RETURN
END

SUBROUTINE SNTEM(FXSN,FZSN,AYSN,THETA)
COMMON/INCUT/1W,iP
COMMON/DATE,AEFO(175),AEFG(50),KODE(261),LL
COMMON 7777(257)
COMMON/TAF1/T27(T47)
COMMON/SN9R/SNUM3,SN3,SN(3C),THUSN,THLSN,SNUD(3,3)
EQUIVALENCE(AEFO(15),SNUM2,AEFOIC6C),SNUMY),(AEFG(107),SNUY,)
1 (AEFG(109),SNUMX),(AEFG(109),SNUM2),(AEFG(110),SNUM3),CABC3770
2 (AEFG(111),SNUMT),(AEFG(112),SNUML),(AEFG(113),SNUML),CABC3770

74
FILE: CABLE FORTRAN TI

GRUMMAN DATA SYSTEMS

3 (AER0(114), SNLST), (AER0(115), SNLST), (AER0(116), SNLST), CABC0100
4 (AER0(117), TUSN), (AER0(118), TUSN), (AER0(119), AKNJ), CABC0110
5 (AER0(120), AKNJ), (AER0(49), VO), (AER0(51), RHO), CABC0120
6 (AER0(76), WLCR), (AER0(77), STACO), CABC0130
7 (AER0(78), SLCF), CABC0140

EQUIVALENCE (SN(1), GXI), (SN(2), GYI), (SN(3), GZI),
1 (SN(4), GX2), (SN(5), GY2), (SN(6), GZ2), CABC0150
2 (SN(7), GX3), (SN(8), GY3), (SN(9), GZ3), CABC0160
3 (SN(10), GX4), (SN(11), GY4), (SN(12), GZ4), CABC0170
4 (SN(13), THU), (SN(14), TLL), (SN(15), ALU), CABC0180
5 (SN(16), ALL), CABC0190
6 (SN(19), HGX1), (SN(20), THG1), (SN(21), THG2), CABC0200
7 (SN(22), HGX2), (SN(23), THG2), (SN(24), THG2), CABC0210
8 (SN(25), HGX3), (SN(26), THG3), (SN(27), THG2), CABC0220
9 (SN(28), HGX4), (SN(29), THG2), (SN(30), THG2), CABC0230

IF(KODE(10).EQ.2) GO TO 5005
IF(KODE(10).EQ.3) GO TO 5005
CALL DECNSN(THETA)
IF(KODE(10).NE.1) GO TO 5003

C TERMS TO MODEL SNUBBER EFFECTS (MODEL UNSNUBBED)

CALL STINTO(AL, 1, 1, TUSN, NG)
CALL STINTO(ALL, 1, 1, TUSN, NG)
CALL STINTO(ALU, 2, 2, THUSN, NG)
CALL STINTO(ALU, 2, 2, THUSN, NG)
CALL STINTO(ALL, 2, 2, THLSN, NG)

IF(KODE.eq.1) GO TO 5001

5001 CONTINUE

C CALCULATING FORCE AND MOMENT EFFECTS

CALL DFCUSN(THETA)
FXUSN = 2*THUSN*GX1
FZUSN = 2*THUSN*GZ1
AMUSN = FXUSN*SNU2 + SNUX*FZUSN
FLSN = 2*THLSN*GX3
FZLSN = 2*THLSN*GZ3
AMLSN = FLSN*SMLZ2 + FZLSN*SNLX
FXSN = FXUSN*FLSL
FZSN = FZUSN*FZLS
AASN = (AMUSN + AMLS7)/12.
RETURN

5003 CONTINUE

C TERMS TO MODEL SNUBBER EFFECTS (MODEL SNUBBED)

FXUSN = 2*THUSN*GX1
FZUSN = 2*THUSN*GZ1
AMUSN = -FXUSN*SNU2 + FZUSN*SNUX
FLSN = 2*THLSN*GX3
FZLSN = 2*THLSN*GZ2
AMLSN = FLSN*SMLZ2 + FZLSN*SNLX
FXSN = FXUSN*FLSN
FZSN = FZUSN*FZLS
AASN = (AMUSN + AMLS7)/12.
RETURN
EXTERNAL LONGLNX
COMMON/INDUT/IN,10
COMMON/DAT/AERO(175),AEROP(50),KODE(24),LL
COMMON/SNU3/SNUL3,3),SN(30),THUSN,THL,SN,SNUD(3,3)
COMMON Z777(200)
COMMON/TAT1/ZZ(400)
COMMON/JOU/DUM(10,10)
EQUIVALENCE(AERO(105), SVUX),(AERO(106), SNUY),(AERO(107), SNJ)
1
2
3
4
5
6
EQUIVALENCE (SN(1), GX1), (SN(2), GY1), (SN(3), GZ1)
1
2
3
4
5
6
7
8
9
DIMENSION FTDP(3,3), FECT(3,3)
COS(1)=1./TAN(1)
DO 1001 J=1,3
DO 1001 J=1,3
1001 SNU(J)=?
DO 5102 J=1,10
SNU(1,J)=?
DO 5102 J=1,10
5102 DOU(J)=?
IF(KODE(1) NE-1) GO TO 1000
C TERMS FOR UNSNURPED SNUDEEP EFFECTS (LONG)
DO 1004 I=1,7
DO 1004 J=1,7
1004 DOU(I,J)=?
CALL DRCSN(THETA)
DOU(1,1)=-2.*TUSNC2Z1
DOU(1,4)=-2.*TUSNC2SN(THGXI)
DOU(1,6)=2.*GX1
DOU(2,3)=2.*TUSNC2GX1
DOU(2,5)=-2.*TUSNC2SN(THGZ1)
DOU(2,6)=2.*GZ1
DOU(3,1)=-(SNUX*DUM(1,3)+SVUX*DUM(2,3))/12.
DOU(3,4)=-(SNUX*DUM(1,4)+SVUX*DUM(2,4))/12.
DOU(3,6)=SVUX*DUM(2,5)/12.
COMPUTER NAME: GRUMMAN DATA SYSTEMS
FILE CABLE FORTRAN T1

GRUMMAN DATA SYSTEMS

DUM(3,6) = (-SNUX*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 - SNUX*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(5,4) = -1.
CALL DEFCSV(THETA)
G = 5*HID*Y*A**V
ALU = ALU+1.
CALL STINT(Q,ALU1,C,1,1,TUSN1,NG)
IF(NG.5) GO TO 5000
ALU2 = ALU-1.
CALL STINT(Q,ALU2,C,1,1,TUSN2,NG)
IF(NG.5) GO TO 5001
5000 WRITE(IW,522) NG,ALL,ALU,Q
5U02 F3VAM(TABLE 1-2,NG=*123X=1C3)
RETURN
5001 CONTINUE
AKT1 = (TUSN1-TUSN2)/2.
DUM(6,6) = -1.
DUM(6,7) = AKT1*12.
DUM(7,1) = -GZ1
DUM(7,2) = -GZ1
DUM(7,7) = ((-SNUX*ALU*GZ1)*G71 - (-SNUX*ALU*GZ1)*G1)/12.
DUM(7,7) = -1.
CALL WSH(7,7)
DO 1005 J=1,3
DO 1005 J=1,3
1005 FSH(1,1) = DUM(1,1)
CALL DEFCSV(THETA)
DUM(1,3) = -2*TLSNC*GZ3
DUM(1,4) = -2*TLSNC*SIN(THGX3)
DUM(1,6) = 2*GZ3
DUM(2,3) = 2*TLSNC*GZ3
DUM(2,4) = 2*TLN*GZ3
DUM(3,3) = (SNLZ*DUM(1,3)+SNLX*DUM(2,3))/12.
DUM(3,4) = SNLZ*DUM(1,4)/12.
DUM(3,5) = SNLX*DUM(2,5)/12.
DUM(7,6) = (SNLZ*DUM(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(5,4) = -1.
CALL DEFCSV(THETA)
ALL = ALL+1.
CALL STINT(Q,ALL1,C,1,1,TUSN1,NG)
IF(NG.5) GO TO 5003

77
ALL? = ALL = 1.
CALL STINT(2,ALL,3,1,1,1,1,8NSN2,NG)
IF (NG.EQ.0) GO TO 5004

! 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) = ((SMLX*ALL*GX3)*GZ3 - (SML7*ALL*GZ3)*GX3)/12.
DUM(7,4) = -1.
CALL 'ASHF(3,7)
DO 1006 I = 1,3
DT 1008 J = 1,3
1008 FSNB(I,J) = DUM(I,J)
DT 1009 I = 1,3
DT 1007 J = 1,3
SNBD(I,J) = 0
1000 SNU1(I,J) = FTPD(I,J) + FNU1(I,J)
RETURN

! 1002 IF (KODC(1) .EQ. 0) GO TO 1002
C TERMS FOR SNUDDRED SNUDDRED EFFECTS(LONG)
CALL OFCSN(THETA)
DT 1006 I = 1,7
DT 1007 J = 1,7
1006 DUM(I,J) =
DUM(1,1) = -2.*TUSNC*GZ1
DUM(1,2) = -2.*TUSNC*SIN(THG1)
DUM(1,3) = 2.*GX1
DUM(2,4) = -2.*TUSNC*GXL1
DUM(2,5) = -2.*TUSNC*SIN(THG1)
DUM(2,6) = 2.*GZ1
DUM(3,7) = -SNUZ*SUM(1,3) + SNUX*SUM(2,3))/12.
DUM(3,4) = -SNUZ*SUM(2,4)/12.
DUM(3,5) = SNUX*SUM(2,5)/12.
DUM(3,6) = -SNUZ*SUM(1,6) + SNUX*SUM(2,6))/12.
DUM(4,1) = (SIN(THG1)/ALU) * 12.
DUM(4,2) = (-GZ1*COT(THG1)/ALU) * 12.
DUM(4,3) = -SNUZ*SIN(THG1)/ALU - SNUX*GZ1*COT(THG1)/ALU
DUM(4,4) = -1.
DUM(5,1) = -3*X1*COT(THG1)/ALU) * 12.
DUM(5,2) = (SIN(-THG1)/ALU) * 12.
DUM(5,3) = SNUZ*GX1*COT(THG1)/ALU + SNUX*SIN(THG1)/ALU
DUM(5,4) = -1.
DUM(5,5) = -1.
DUM(6,6) = -1.
DUM(6,7) = AKSNU = 12.
DUM(7,1) = -DX1
DUM(7,2) = -GZ1
DUM(7,3) = ((SUMX+ALU*GX1)*GZ1 - (-SNUZ+ALU*GZ1)*GX1)/12.
DUM(7,4) = -1.
DO 10 I = 1,3
DO 10 K = 1,7
10 SNU1(I,J) = DUM(I,J) + AKSNU + DUM(7,J)*12.

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

1007 DUV(1,1)=SUM(I,J)

DUM(1,1)= -2.*TSLN*GZ3
DUM(1,4)= -2.*TSLN*SIN(THGX3)
DUM(1,6)= 2.*GK3

DUM(1,2)= 2.*TSLN*GZ3
DUM(1,5)= -2.*TSLN*SIN(THGZ3)
DUM(2,6)= 2.*GK3

DUM(3,3)= (SNLZ*SUM(1,3)) / SNLX*SUM(2,3)/12.
DUM(3,4)= SNLZ*SUM(1,4)/12.
DUM(3,5)= SNLX*SUM(2,5)/12.
DUM(3,6)= (SNLZ*SUM(1,6)+SNLX*SUM(2,6))/12.
DUM(4,1)= (SIN(THGX3)/ALL)*12.
DUM(4,2)= (-GZ3*COT(THGX3)/ALL)*12.
DUM(4,3)= SNLX*SUM(THGX3)/ALL - SNLX*GZ3*COT(THGX3)/ALL.

DUM(4,5)= -1.
DUM(5,1)= (-GZ3*COT(THGZ3)/ALL)*12.
DUM(5,2)= (SIN(THGZ3)/ALL)*12.
DUM(5,3)= -SNLX*GZ3*COT(THGZ3)/ALL + SNLX*SUM(THGZ3)/ALL.
DUM(5,4)= -1.

DUM(6,6)= AK3+12.
DUM(7,1)= -GK
DUM(7,2)= -GZ:
DUM(7,3)= ((-SNLX+ALL*GX3)*GZ3 -(SNLZ+ALL*GZ3)*GZ3)/12.
DUM(7,4)= -1.

DO 20 I=1,3
DO 20 J=1,3

20 SUM(I,J)= .JO(I,J)+SUM(I,6)*30.*SUM(3,7)*12.

CALL WASH(3,7)
DO 1008 I=1,3
DO 1008 J=1,3

1008 FOT(I,J)=DUM(I,J)

DO 1009 I=1,3
DO 1009 J=1,3

1009 SUM(I,J)= FOT(I,J)+FOT(I,J)
RETURN

1002 DO 1002 I=1,3
1002 SUM(I,J)=

1003 SUM(I,J)=
RETURN
END

SUBROUTINE DCSSN(THETA)
COMM/DAT/AERO(175),AER0P(55),KODE(25),LL
COMM/SNUH3/SNU(3,3),SNU(3,3),THSN,THLSN,SNUH(3,3)
EQUIVALENCE(AERO(15),SUM),AERO(16),SUM),AERO(17),SUM),
1 (AERO(18),SNUX),(AERO(19),SNUY),(AERO(20),SNUZ),AERO(21),SNU)
2 (AERO(22),SNUST),(AERO(23),SNUWL),(AERO(20),SNUYL),AERO(24),SNU)
3 (AERO(25),SNUST),(AERO(26),SNUWL),(AERO(20),SNUYL),AERO(27),SNU)
4 (AERO(28),SNUX),(AERO(29),SNUY),(AERO(30),SNUZ),AERO(31),SNU)
5 (AERO(32),SNUST),(AERO(33),SNUWL),AERO(34),SNU)
6 (AERO(35),SNUX),(AERO(36),SNUY),(AERO(37),SNUZ),AERO(38),SNU)
FILE: CABLE FORTAN 11
GRUMMAN DATA SYSTEMS

EQUIVALENCE (SN(1), GX1), (SN(2), GY1), (SN(3), GZ1),
(SN(4), GX2), (SN(5), GY2), (SN(6), GZ2),
(SN(7), GX3), (SN(8), GY3), (SN(9), GZ3),
(SN(10), GX4), (SN(11), GY4), (SN(12), GZ4),
(SN(13), THU), (SN(14), THL), (SN(15), ALU),
(SN(16), ALL),
(SN(19), THGX1), (SN(20), THGY1), (SN(21), THGZ1),
(SN(22), THGX2), (SN(23), THGY2), (SN(24), THGZ2),
(SN(25), THGX7), (SN(26), THGY3), (SN(27), THGZ7),
(SN(28), THGX4), (SN(29), THGY4), (SN(30), THGZ4),

C CALCULATION OF CABLE DIRECTION COSINES

X31 = (STAC-SNUST)*COS(THETA) - (WLCE-SNUWL)*SIN(THETA)
X32 = X31
X33 = (STAC-SNLST)*COS(THETA) - (WLCE-SNLWL)*SIN(THETA)

X41 = X31
X42 = X32
X43 = X33

ALUSQ = DX1**2 + DY1**2 + DZ1**2
ALU = SQRT(ALUSQ)
ALLSQ = DX3**2 + DY3**2 + DZ3**2
ALL = SQRT(ALLSQ)

GX1 = DX1/ALU
GY1 = DY1/ALU
GZ1 = DZ1/ALU
GX2 = DX2/ALU
GY2 = DY2/ALU
GZ2 = DZ2/ALU
GX3 = DX3/ALL
GY3 = DY3/ALL
GZ3 = DZ3/ALL
GX4 = DX4/ALL
GY4 = DY4/ALL
GZ4 = DZ4/ALL

DO 1 I=19,3
J=I-19
1 SN(J)=ACOS(SN(J))
RETURN
END

SUBROUTINE DECUSN(THETA)
COMMON/DATE/AE=0.175, AE=0.55, KOPC=26, LL
COMMON/SN(3), VN(3), THUSN, THLSN, SNUD(3,3)

CALL CABLE

80
EQUIVALENCE ((AF58(15)), (SNXU), (AF58(16)), (SNYU), (AF58(17)), (SNZU)), CABCC6460
1 (AF58(109)), (SNLX), (AF58(109)), (SNLY), (AF58(11)), (SNLZ)), CABCC6700
2 (AF58(111)), (SNUST), (AF58(112)), (SNUUL), (AF58(113)), (SNUUL)), CABCC6400
3 (AF58(114)), (SNUST), (AF58(115)), (SNUUL), (AF58(115)), (SNUUL)), CABCC6490
4 (AF58(117)), (TUSC), (AF58(118)), (TUSLO), (AF58(119)), (AKSNU)), CABCC6700
5 (AF58(120)), (AKSNL)), CABCC6710
6 (AF58(76)), (MLC5), (AF58(77)), (STACF), (AF58(78)), (SLCP)), CABCC6720

EQUIVALENCE (SN(1), GX1), (SN(2), GY1), (SN(3), GZ1), CABCC6730
1 (SN(4)), GX2), (SN(5), GY2), (SN(6), GZ2), CABCC6740
2 (SN(7)), GX3), (SN(8), GY3), (SN(9), GZ3), CABCC6750
3 (SN(10)), GX4), (SN(11), GY4), (SN(12), GZ4), CABCC6760
4 (SN(13)), THU1), (SN(14), THL), (SN(15), ALU), CABCC6770
5 (SN(16)), ALL), CABCC6780
6 (SN(19)), THG1), (SN(20), THG1), (SN(21), THG1), CABCC6790
7 (SN(22)), THG2), (SN(23), THG2), (SN(24), THG2), CABCC6800
8 (SN(25)), THG3), (SN(26), THG3), (SN(27), THG3), CABCC6810
9 (SN(28)), THG4), (SN(29), THG4), (SN(30), THG4), CABCC6820

C CALCULATION FOR EFFECTIVE DIRECTION COSINWS FOR UNSNUGGED CASE
AYL = SNLNL- (MLC5*SNLY)
AZL = -SNLNL-(MLC5*SNLY)
AYU = SNLNL-(MLC5*SNLY)
AZU = SNLNL-(MLC5*SNLY)
THU = ATAN(AYU/AYU)
THL = ATAN(AZL/AZU)
ALU = ATAN(SYN(THUSN)*COS(THU))

GX1S = -COS(THUSN)
GY1S = -AYU/ALU
GZ1S = -AZU/ALU
GX1 = GX1S+COS(THETA)-GZ1S*SIN(THETA)
GY1 = GY1S
GZ1 = GZ1S*COS(THETA)+GZ1S*SIN(THETA)
GX2 = GX1
GY2 = GY1
GZ2 = GZ1
ALL = AYU/SIN(THUSN)*COS(THUL)
GX3S = -COS(THUSN)

GY3S = AYU/ALL
GZ3S = AZL/ALL
GX3 = GX3S+COS(THETA)-GZ3S*SIN(THETA)
GY3 = GY3S
GZ3 = GZ3S*COS(THETA)+GZ3S*SIN(THETA)
GX4 = GX3
GY4 = GY3
GZ4 = GZ3
DQ 1 = 19, 3C
J = I-18
1: SN1(I) = ACOS(SN(J))
RETURN
END
SUBROUTINE F C T E
COMMON / IN F / I F
COMMON / DAT / A, (176), (AF58(50), KODE(26)), ALL
IF (KODE(4), GT, 1) GO TO 1
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')
4 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')
7 CONTINUE
IF(KODE(11).EQ.0) GO TO 9
WRITE(IW,800)
800 FORMAT (25X,'LIFT/ANTI-LIFT CABLE IN')
GO TO 9
8 WRITE(IW,900)
900 FORMAT (25X,'NO LIFT/ANTI-LIFT CABLE')
9 CONTINUE
IF(KODE(13).LE.0) WRITE(IW,1000)
IF(KODE(13).LE.1) WRITE(IW,1001)
IF(KODE(13).EQ.2) 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,MINTL,MAXTL,FCT,NG)
EQUIVALENCE (X(1),NUMPTS(I))
COMMON NUMPTS(I)
DIMENSION X(1)
I=NUMPTS(I)/3
70 IF(MINTL-MAXTL)/3. I=1
71 GO TO 73
72 IF(A3-X(NJ))72,74,73
74 NI=NUMPTS(I)+1
73 CONTINUE
GO TO 112
75 IK = NI
IL =2
NJ=NI
101 CT 97 IF=IK,IL
NJ =NUMPTS(I)+1
FILE: CABLE FORTRAN T1

GRUMMAN DATA SYSTEM

11 = 1 + II
10 = NUMPTS(11)
1P = 10 + NJ
D0 47 I0 = I1, 17
NN = NJ + 19
IF (AI = X(NV)) 76, 79, 77
76 IF (I0-1) 112, 112, 79
77 CONTINUE
GO TO 112
78 IG = -1
GO TO 96
79 IG = +1
80 N1 = N1 + 12
81 NS = NUMPTS(N1)
DO 92 IA = 1, 19
92 CONTINUE
GO TO 112
83 IH = -1
GO TO 85
84 IH = +1
85 NE = IP + IP + IO + IO - IA - IO
N5 = NE - IO
IF (IG + IH) 96, 98, 91
96 IF (X(N5) = 99298.5E9) 87, 113, 113
97 FCT = X(N5)
GO TO 95
98 IF (IG) 89, 119, 93
99 IF (AMAX1(X(NF), X(NV)) = 99298.5E9) 90, 113, 113
100 FCT = X(NF) - (X(NS) - A2) * (X(NF) - X(NF)) / (X(NS) - X(NS-1))
GO TO 95
101 IF (AMAX1(X(NF), X(NV)) = X(NF-1), X(NV-1)) 99298.5E9) 92, 113, 113
102 FCT = (1 - (X(NS) - A2) * (X(NS) - A1) * (X(NF-1) - X(NF))
1 - (X(NS-1) - A2) * (X(NS-1) - A1) * (X(NS-1) - (X(NV-1) - A1) * (X(NF-1)))
2/(X(NS) - X(NS-1)) = (X(NS) - X(NS-1))
GO TO 95
103 IF (AMAX1(X(NF), X(NV)) = X(NF-1), X(NV-1)) 99298.5E9) 94, 113, 113
104 FCT = (X(NF) - (X(NF-1) - X(NF)) / (X(NV-1)) = (X(NV-1))
GO TO 95
105 IF (94, 95, 99, 99, 99)
106 DUMSTG = FCT
107 II = II - 1
108 FCT = DUMSTG - (X(NM) - A3) * (DUMSTG - FCT) / (X(NM) - X(NM))
RETURN
110 IC = 3
111 IC = 3
GO TO 101
112 NG = 2
GO TO 99
113 NG = 4
GO TO 99
END
FILE: CABLE FORTRAN T1

SUBROUTINE TAHN(NUMTBL, NZ, NG)
 COMMON NUMPTS(1)
 COMMON /INQ/ I1, L2N
 COMMON /INTABOUT/ NINTBL, ISOQ
 DIMENSION NUMPTS(1)
 INTEGER I, LABEL(27)
 EQUIVALENCE (NUMPTS(1), NUMPTS(1), (NUMPTS(1), NUMPTS(1))
 DIMENSION DUMMY(10)
 MCP=0
 10 IF(Z+I43(N2))
 NUNIT=5
 IF(NZ.LT.8) NUNIT=8
 NINTBL = NUMTBL
 PNG=0
 NUMPTS(1) = Z+I2+Z
 102 READ(NUNIT, 57) K, L1N, L2N, LABEL, ISOQ
 IF(MCP.EQ.1) GO TO 3
 4 WRITE(IW+1) K, L1N, L2N, LABEL, ISOQ
 1 FORMAT(115, 10X, 27A2, 146)
 57 FORMAT(AK14, 212, 27A2, 12)
 3 IF(I$1$EQ) 49, 58, 69
 59 IF(K) 99, 99, 59
 59 IF = Z + NINTBL
 NUMPTS(1) = L1N
 M = V + I2
 NUMPTS(1) = L2N
 IF(NUMTBL-NINTBL) 17, 70, 77
 17 NUMPTS(NINTBL) = NUMMY
 70 X = (L1N+1) / 8 + 1
 DO 68 IS = 1, X
 L3 = (IS-1) * 9 + 1
 IF(IS-N2) 60, 61, 60
 60 L4 = L3 * 8
 GO TO 62
 61 L4 = L1N
 62 L5 = NUMPTS(NINTBL) + 1
 L6 = L5 + L7
 L7 = L5 + L4
 JJ = 0
 L4 = L5 + LIN
 LIN = LM + L2N
 63 READ(NUNIT, 64) (DUMMY(K), K=1, 10), ISOQ
 64 FORMAT (11FE7.5, 12)
 IF(CD.5Q.0) GO TO 5
 6 WRITE(IW+2) DUMMY, ISOQ
 2 FORMAT(10Cl2.4, 15)
 5 NUMPTS(L5) = DUMMY(1)
 K = 2
 DO 65 J = L6, L7
 NUMPTS(J) = DUMMY(K)
 65 K = K+1
 ISOQ = (IS-1) * (L2N+1) + JJ+1
 IF(I$1$EQ-ISOQ) 69, 66, 69
 66 L6 = LN + L3
 L7 = LN + L4
FILE CABLE Fortran T1

GRUMMAN DATA SYSTEMS

LS = LM + JJ
IF (JJ-LN) 67, 69
67 JJ = JJ + 1
LN = LN + LIN
GO TO 63
68 CONTINUE

109 NUMY = 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,ECT,NG)

EXTERNAL X

EQUIVALENCE (X(1),NUMPTS(1))

DIMENSION X(1)

IZ=NUMPTS(1)/3

70 IF(NIMTL-MAXTBL)71,71,11C
71 DO 73 I=1,NIMTL,MAXTBL
NJ=NUMPTS(I)+1
72 IF(A1-X(NJ))72,74,73
73 CONTINUE
GO TO 112
75 I=1
IL =2
NJ=NL
101 DO 97 IF=IK,IL
NJ =NUMPTS(I)+1
74 I = IZ+II
75 DO 92 I=1,1C
76 I(I-I)+11C,112,79
77 CONTINUE
GO TO 112
78 IG =1
GO TO 77 8C
79 IG =-1
90 NI=NI+12
91 DO 92 IA=1,1C
93 NS=1P+1A
94 IF(A2-X(NS))91,G3,92
95 IF(I(A-1))11C,112,94
92 CONTINUE
GO TO 112
93 IH =1
GO J 95
96 I =1
97 =-I9+10+I9*IA+1C
GO J 39,88,91
FILE: CABLE  FORTRAN T1  . GRUMMAN DATA SYSTEMS

86 IF (X(NF)-9999.5E9) 187, 113, 113
87 FCT = X(NF)
GO TO 95
88 IF (IG) 89, 110, 93
89 IF (MAX1(X(NF), X(NR)) - 9999.5E9) 92, 113, 113
90 FCT = X(NF) - (X(NS) - A2) * (X(NF) - X(NS)) / (X(NS) - X(NS-1))
GO TO 95
91 IF (MAX1(X(NF), X(NF), X(NF-1), X(NR-1)) - 9999.5E9) 92, 113, 113
92 FCT = ( (X(NS) - A2) * (X(NN) - A1) * X(NF) - (X(NN-1) - A1) * X(NF)
11 - (X(NS-1) - A2) * (X(NN) - A1) * X(NF-1) - (X(NN-1) - A1) * X(NF))
2 * ( (X(NS) - X(NS-1)) * (X(NN) - X(NN-1))
GO TO 95
93 IF (MAX1(X(NF), X(NF-1)) - 9999.5E9) 94, 113, 113
94 FCT = X(NF-1) - X(NF) * X(NF-1) / (X(NN) - X(NN-1))
GO TO 96, 99, 99, IF
95 DUMSTG = FCT
96 II = II - 1
97 IF (DUMSTG = FCT)
GO TO 99
98 RETURN
99 IK = 3
IL = 3
GO TO 101
101 YG = 2
GO TO 99
112 NG = 4
GO TO 99
END

SUBROUTINE TABIN(NUMTBL, NZ, NG)
COMMON/INDUT/1W, IF

COMMON/TABIN, NUMPTS(1)
COMMON/TABIN, NUMTBL, ISOQ
DIMENSION NUMPTS(1)
INTEGER*2 LABEL(27)
EQUIVALENCE (NUMPTS(1), NUMPTS(1)), (DUMMY(1), NUMPTS(1))
DIMENSION DUMMY(1)

D = 1
IZ = 1
NUT = E

IF (NZ < LTN, E) NUNIT = 8
NUMTBL = NUMTBL
NG = 3
NUMPTS(1) = IZ + IZ + 17
102 READ(NUNIT, 57) K, N, L2N, LABEL, ISF0
IF (K = EQ, 0) GO TO 3
WRITE(1W, 1) K, LIN, L2N, LABEL, ISF0
1 F30.4(3,5, 10, 27A2, 146)
57 F30.4(9X14, 21, 27A2, 12)
3 IF (ISF0 = 99, 89, 90
58 IF (K = 99, 99, 99
59 M = 17 + NUMTOL
NUMPTS(N) = LIN
W = W + 17
NUMPTS(M) = L2N

86
FILE CAMELE FORTRAN T1

GRUMMAN DATA SYSTEMS

IF (NUMTBL-NIMTBL) 17, 70, 17
17 NUMPTS(NIMTBL) = NUMXY

70 NL = (LIN-1) / 9 + 1
D0 68 IS = 1, NL

L3 = (IS-1) * 9 + 1
IF (IS-N1) 60, 61, 60

60 L4 = L3 + 8
GO TO 52

61 L4 = LIN

62 LS = NUMPTS(NIMTBL) + 1
L6 = LS + L3
L7 = LS + L4
JJ = 0

L4 = LS + LIN
L5 = LM + L2N

63 READ(NUNIT,64) (DUMMY(K),K=1,1C), ISEO

64 FORMAT (1CE7.2,12)

1 IF (NCF.EQ.,) GO TO 5
6 WRITE (11,2) DUMMY, ISEO

2 FORMAT (1CE12.4,15)

5 XUPTS(L5) = dummy(1)

K = 2
D0 62 J = L6, L7

XUPTS(J) = dummy(K)

65 K = K+1

ISOO = (IS-1) * (L2N+1) + JJ + 1
IF (ISEQ-ISOO) 69, 66, 69

66 L6 = LN + L3
L7 = LN + L4
L5 = LM + 1 + JJ
IF (JJ-L2N) 67, 68, 69

67 JJ = JJ + 1
LN = LN + LIN
GO TO 63

68 CONTINUE

109 NUMXY = NUMPTS(NIMTBL) + (LIN+1) * (L2N+1)

108 NIMTBL = NIMTBL + 1
GO TO 102

69 NG = 1

99 RETURN

END

SUBROUTINE FRIC(TDX)

C000000 C000000 C000000 (3, 6)
C000000 C000000 C000000 (AERO(96), COU), (AERO(104), CMP)

DO 1 I=1, 3

DO 1 J=1, 6

1 FRIC(I,J)=0
IF (CMP. EQ., AND COU. EQ.,) RETURN
IND=KODE(6)
IF (IND. NE.,) GO TO 2

C LONGITUDINAL PULLEY FRICTION COMPUTATION
GO TO (1, 11, 12, 13), IND

10 CALL FRVT(1)
CHANGE
11 CALL FEVT(3)  
RETURN  
12 CALL FEVT(1)  
CALL FEVT(3)  
13 RETURN  

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

SUBROUTINE FEVT(IC)  
C COMPUTES THE FRICTION EFFECT OF THE VERT PULLEYS ON THE LONG.  
COMMON/DATA/AE(0,175),AE(0,50),KOE(24)  
COMMON/PLCH/AE(0,10),XLTH(5),AD(5,3),AA(5,3),TR,TLET,TF  
COMMON/ROUGH/FRIC(3,6)  
EQUIVALENCE(AE(90,1),RFV),(AE(92,1),RVP),(AE(95,1),CQ).  
LENGTH(AE(1C4),CF)  
DIMENSION DT(3),CT(3)  
IF(IC.EQ.3) GO TO 1  
TENS=TF  
RAD=VF/12.  
AVX=(AD(2,1)-AD(1,1))/2.  
CA=COV(AVX)  
CAZ=SVN(AVX)  
GO TO 2  
1 TENS=TF  
RAD=VF/12.  
AVX=7.14157+(AD(4,1)-AD(3,1))/2.  
CA=COV(AVX)  
CAZ=SVN(AVX)  
2 ARMX=(ARMX+IC,1)+ARMX(1C+1,1))/2.  
ARMZ=(ARMX+IC,1)+ARMX(1C+1,1))/2.  
ENDX=TNX*COV(AD(1C,1))  
ENDZ=TNX*(1+COV(AD(1C,1)))  
ENDM=SORT(ENCX**2+ENCZ**2)  
CMPP=CMP/ENDM  
FACT=CMPP*ENDM/RAD**2  
ENDX=TNX*CMPP(AD(1C+1,1))  
ENDZ=TNX*(1*CMPP(AD(1C+1,3,1)))  
ENDM=SORT(ENCX**2+ENDZ**2)  
CMPP=CMPP/ENDM  
FACT=CMPP*ENDM/RAD**2  
FACTA=4.14157+FACT**2  
CALL DLTH(CX+CZ,CT,1C,C)  
CALL DLTH(CZ,CZ,CT,1C,C)  
DT(1)=FACT *(CX**2-CX)  
DT(2)=FACT *(CZ**2-CZ)  
DT(3)=FACT *(CT**2-CT)  
DT(4)=FACTA*(CZ**2-CX)  
DT(5)=FACTA*(CT**2-CT)
CONTINUE
RETURN
END

SUBROUTINE FCHZ(IC)
C COMPUTES THE FRIC. EFFECT OF THE HDFZ PULLEYS ON THE LAT. DIR. DYN.
COMMON/DAT/AERO(175),AESP(50),KODE(26)
COMMON/PLYCHA/RD,XLGH(5),ADUC(5,3),AEXP(5,3),TR,TLR,TF
COMMON/ROUGH/FRIC(2,6)
EQUIVALENCE (AESP(91),FHP),(AESP(93),PHP),(AESP(96),COU),
1(AESP(104),CMP)
DIMENSION DT1(3),DT2(3)
IF(IC.EQ.3)GO TO 1
TENS=TF
RAD=9RF/12.
GO TO 2
1 TENS=TF
RAD=9RF/12.
RAD=9RF/12.
2 ENORY=TENS*COS(ADC(IC,1))
ENORY=TENS*(1+COS(ADC(IC,2)))
ENORM=SQR(ENORY*ENORY+ENFX*ENFX)
CMP=CMP/ENORM
FACL=CMP*ENFX/RAD**2
FACT=4.*COS(7.14159*RAD**2)
CALL DLGTH(CY,CPHI,IC,1)
CALL DLGTH(CYP,CPHISP,IC,1)
DT1(1)=FACT*(CY-CYP)
DT1(2)=FACT*(CPSI-CPSI)
DT1(3)=FACT*(CPHI-CPHISP)
DT2(1)=FACL*(CY-CYP)
DT2(2)=FACL*(CPSI-CPSI)
DT2(3)=FACL*(CPHI-CPHISP)
DT 3 I=1,3
FRIC(1,1)=FRIC(1,1)+DT1(1)*COS(ADC(IC,1))
FRIC(1,1+3)=FRIC(1,1+3)+DT2(1)*COS(ADC(IC,1))
FRIC(2,1)=FRIC(2,1)+DT1(1)*COS(ADC(IC,1))
1/12.*DT1(1)*COS(ADC(IC,1))
FRIC(2,1+3)=FRIC(2,1+3)+DT2(1)*COS(ADC(IC,1))
1/12.*DT2(1)*COS(ADC(IC,1))
FRIC(3,1)=FRIC(3,1)+DT1(1)*COS(ADC(IC,1))
1/12.*DT1(1)*COS(ADC(IC,1))
FRIC(3,1+3)=FRIC(3,1+3)+DT2(1)*COS(ADC(IC,1))
1/12.*DT2(1)*COS(ADC(IC,1))
CONTINUE
RETURN
END

SUBROUTINE MATRIX(CMAT,N,FOOT3,K4A,IEP)
COMMON/DAT/AERO(175),AESP(50),KODE(26),LL
COMMON/F07/C4(3) 
DIMENSION C MAT(14,14,3), MAT(14,14), KOUNT(30), CS(30) 
COMPLEX AMAT(14,14), ROOTS(29) 
DOUBLE PRECISION BMAT(14,14,3), D(30,4) 
NP=2 
IF(KD0=5,C0,1) NP=1 
CALL MAPY(CMAT,C4,ROOTS,K4A,14,NP,3,30,KOUNT) 
1 AMAT,BMAT,MAT,C5,D,N) 
RETURN 
SUBROUTINE MAPY(CMAT,C4,ROOTS,K4A,MCOL,NP) 
1 IN, N, KOUNT, AMAT, BMAT, MAT, C5, D) 
COMMON/INOUT/IN, NP 
DIMENSION AMAT(MCOL,1), MAT(MCOL,1), BMAT(MCOL,MCOL,1) 
1 C4(1), ROOTS(1), KOUNT(1), CMAT(MCOL,MCOL,1) 
2 C5(1) 
DOUBLE PRECISION AMAT, SA, E, D(N+1,0) 
* FORMAT 
COMPLEX DET=FCMPLX, AMAT, G, YAI, YA, SCMPLX 
COMPLEX G3, ROOTS, C 
12 FORMAT (213, 1PE3.6/1D22.6, 4D16.6) 
14 FORMAT (1HC213. 1PE24.6, 4D16.6) 
15 FORMAT (1HC213. 1PE24.6, 4D16.6) 
22 FORMAT (1HC213. 1PE24.6, 4D16.6) 
2614 FORMAT (/1PE24.6, 4D16.6) 
DATA CF/Z7FPEFEEEEEEE/
NCOL=MCOL 
10 NR=MNCOL 
END=1+IP(NP) 
INN=IN+1 
DO 107 I=1,NP) 
DO 107 J=1,MCOL 
MAT(I,J) = C 
DJ 112 K=INN+1 
112 BMAT(I,J,K)=0,0 
DO 107 K=1,IP 
BMAT(I,J,K)=CMAT(I,J,K) 
IF(CMAT(I,J,K))108,107,108 
108 BMAT(I,J,K)=K 
C THE NUMBER IN MAT IS ONE GREATER THAN THE DEGREE OF THE POLYNOMIAL 
107 CONTINUE 
JS=1 
IF(NP,LT,C)GO TO 128 
ASSIGN 128 TO MZ 
GO TO 920 
99 ASSIGN 257 TO MZ 
920 "FITE(IW,23) 
23 FORMAT(5SHLINPOSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX) 
DO 951 J=1,MCOL 
DO 951 I=1,MCOL 
K1 = MAT(I,J) 
IF(K1) 951,951,952 
952 K=IF(IW,191,J9) 
(AMAT(I,J,K), K=1,K1) 
951 CONTINUE 
GO TO 97, (130, 257, 128, 1106)
C DET CONTAINS VALUE OF DETERMINANT OF SMAT WITH G=1

1291 WRITE(1W,1292)DET

1292 FORMAT(12H DETERMINANT1P2F15.7)

128 NC = 0

C COUNT NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN COLUMN JS

DO 126 JS = 1, NC

IF (MAT(I,JS)) 129, 120, 121

121 NC = NC + 1

JS = I

120 CONTINUE

IF (NC-1) 17, 125, 130

17 WRITE(1W,16)

16 FORMAT(*) MATRIX IS SINGULAR*)

GO TO 257

125 IF (IS-JS) 99, 140, 123

C ONE INTERCHANGE TRIANGULARIZES THE COLUMN

123 DO 126 JS = JS, NC

KI = MAXC(MAT(IS,JS), MAT(JS,JS))

MA = MAT(IS,JS)

MAT(IS,JS) = MAT(JS,JS)

MAT(JS,JS) = MA

DO 126 K = 1, KI

SA = SMAT(IS,JS,K)

SMAT(IS,JS,K) = SMAT(JS,JS,K)

126 SMAT(JS,JS,K) = SA

GO TO 142

130 JS = JS+1

C LOOP 137 REDUCES ALL ELEMENTS BELOW DIAGONAL IN COLUMN JS BY

C AT LEAST ONE DEGREE

1 = IS

137 IF (MAT(I,JS)) 99, 137, 129

129 IF (MAT(JS,JS)) 99, 133, 132

132 IF (MAT(I,JS) = MAT(JS,JS)) 133, 134, 134

133 DO 131 J = JS, NC

KI = MAXC(MAT(JS,JS), MAT(I,J))

MA = MAT(JS,JS)

MAT(JS,JS) = MAT(I,J)

MAT(I,J) = MA

DO 131 K = 1, KI

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,KI)/SMAT(JS,JS,KJS)

IF (DABS(F) = 4.0) 1052, 1051, 1051

1051 IF (KD) 99, 133, 1052

1052 DO 235 J = JS, NC

KJS = MAT(JS,JS)

IF (KJS.EQ.0) GO TO 235

123 K JS = 1, KJS

KI = K + KD

SMAT = F * SMAT(JS,JS,K)

GO TO 126
FILED CABLE FORTRAN II

GRUMMAN DATA SYSTEMS

IF(K1-N) 141,141,2
2 WRITE(IW,31) C3L01130
3 FORMAT(79H0DEGREE OF POLYNOMIAL FORMED WHILE TRIANGULARIZING ORIGINAL C3L01150
1-NAL MATRIX IS TOO HIGH )
GO TO 257 C3L01160

141 IF (.DABS ( FBMAT - BDMAT(I,J,KI)) *LE. 2D-6 * DABS ( FBMAT )) C3L01180
1 GO TO 134 C3L01200
3BDMAT(I,J,KI)= DBMAT(I,J,KI)- FBMAT...
GO TO 135 C3L01210
136 BDMAT(I,J,KI) = 0.D0 C3L01220
135 CONTINUE C3L01230
235 CONTINUE C3L01240
C
DFONTS LLC 01250

J=JS
142 CONTINUE
K=I=I=KD
KJ=J=J=KJ
IF(KI.LT.KJ) KI=KJ
MAT(I,J) = 0
DO 140 K=1,KI
IF(BDMAT(I,J,KI))139,140,138
138 MAT(I,J) = K
142 CONTINUE
J=J+1
137 I=I:
IF(. . . . *NCOL) GO TO 142
139 IF(I.LE.NROW) GO TO 139
IS(NP) = 128,128,1105
140 JS = JS +1
IF(JS=NCOL)129,150,151
141 IF(NP.LT.C) GO TO 153
WRITE(IW,13)
13 FORMAT ( 1H ,2D13H ,13H FINAL MATRIX )
DO 151 J=1,NCOL
DO 151 I=1,NROW
K1 = MAT(I,J)
IF(K1)99,151,152
152 WRITE(IW,12), I, (BDMAT(I,J,K), K=1,K1)
3 CONTINUE
153 LK=1
C LOOP 150-ROOTS OF POLYNOMIALS ON DIAGONAL OF TRIANGULARIZED MATRICES ARE C3L01430
C FOUND AND STORED IN ARRAY ROOTS.
C SLOPE COEFFICIENTS OF THE POLYNOMIAL EQUIVALENT OF THE DETERMINANT C3L01550
C OF THE MATRIX ARE COMPUTED AND STORED IN ARRAY C4 WITH C3L01560
C C4(1) THE CONSTANT TERM.

C LOOP 150-ROOTS OF POLYNOMIALS ON DIAGONAL OF TRIANGULARIZED MATRICES ARE C3L01530
C FOUND AND STORED IN ARRAY ROOTS.
C SLOPE COEFFICIENTS OF THE POLYNOMIAL EQUIVALENT OF THE DETERMINANT C3L01550
C OF THE MATRIX ARE COMPUTED AND STORED IN ARRAY C4 WITH C3L01560
C C4(1) THE CONSTANT TERM.

DO 160 J=1,NCOL
K1 = MAT(J,J)
MK=M=K1+1
K2=K1-1
DO 163 K=1,K1
MK=M=K
163 D(MK,4)= BDMAT(J,J,K)
16 IF(K1.EQ.1) GO TO 1620
161 CALL POLYRT(0(1,4),ROOTS(2*LK-1),KOUNT(LK),K2,J(1,1),D(1,2),D(1,3)) C3L01650
!1
```
CLEC1680
CLEC1690
CLEC1700
CLEC1710
CLEC1720
CLEC1730
CLEC1740
CLEC1750
CLEC1760
CLEC1770
CLEC1780
CLEC1790
CLEC1800
CLEC1810
CLEC1820
CLEC1830
CLEC1840
CLEC1850
CLEC1860
CLEC1870
CLEC1880
CLEC1890
CLEC1900
CLEC1910
CLEC1920
CLEC1930
CLEC1940
CLEC1950
CLEC1960
CLEC1970
CLEC1980
CLEC1990
CLEC2000
CLEC2010
CLEC2020
CLEC2030
CLEC2040
CLEC2050
CLEC2060
CLEC2070
CLEC2080
CLEC2090
CLEC2100
CLEC2110
CLEC2120
CLEC2130
CLEC2140
CLEC2150
CLEC2160
CLEC2170
CLEC2180
CLEC2190
CLEC2200
CLEC2210
CLEC2220

FILE: CABLE FORTRAN II
GRUMMAN DATA SYSTEMS

K1 = K1/2
L1 = L1 + K1
IF (MOD(K1, 2) .NE. 0) GO TO 1020
C DUMMY ELEMENT STORED IN ARRAY ROOTS IF POLYNOMIAL IS OF ODD DEGREE
54 ROOTS(4+L-2) = -
1020 IF (J.EQ.1) GO TO 1004
1001 DO 1002 K = 1, K4A
1002 C4(K) = C4(K1)
DO 1006 K = 1, 4
1006 C4(K) = 0.0
IF (K1) 99, 160, 100C
100D DO 1003 K = 1, K1
1003 MN = MN - 1
  DO 1003 K = 1, K4A
64. CONTINUE
160 CONTINUE
   CALL JUGGLE(ROOTS, ROOTS, KOUNT, K4A)
   DO 306 J = 1, NCOL
   .& MAT(I,J) = IV
   .& DO 306 K = 1, IN
306 SWAT(I,J,K) = CMAT(I,J,K)
   IF (NP .LT. 1) GO TO 202
   WRITE(IW, 15)
   201 WRITE(IW, 15)
   202 IF (LK .EQ. 1) GO TO 111C
1111 L = 1
62 G = ROOTS(L)
64. ASSIGN 244 TO 40T
   GO TO 2511
244 GI = ABS(C4(I1))
C LOOP 251C - PLACE LARGEST PRODUCT, C4(I)*G***(1-1), IN GI
C G3 = ERROR ESTIMATE
   G = ROOT
   DO 2610 L9 = 2, K4A
G2 = ABS(G)
2610 G2 = ABS(G4(L9) .AND G2**=(L9-1))
   IF (G1 - G2) 2611, 2610
2611 G1 = G2
2410 CONTINUE
C DET CONTAINS VALUE OF POLYNOMIAL EQUIVALENT OF DETERMINANT OF
C MATRICES AT ROOT
   IF (G1 .EQ. 0.) GO TO 25
G3 = DET/G1
   GO TO 26
25 G3 = (O.E. +)
26 IF (ABS(G3) .LE. ENP .AND. NP .LT. -1) GO TO 255
   WRITE(IW, 27)
27 FORMAT(5X, 'THE FOLLOWING EXTRACTED ROOT HAVE POOR ACCURACY')
   WRITE(IW, 15)
```
FILE.CABLE  FORTRAN  TI  GRUMAN  DATA  SYSTEMS

WRITE(IN,14) G,G3
256 L=L+1
IF(K4A-1,GE,L167) TO 62
IF(NP+LT=-1) GO TO 257
1110 WRITE(IN,1010)((C4(K),K=1,K4A)
1010 FORMAT (11H0POLYNOMIALIP5E16.6/(E27.6,4E16.6))
257 RETURN
1105 ASSIGN 1291 TO QDT
C  LOOP  210
C EVALUATE EACH POLYNOMIAL OF THE ORIGINAL MATRIX FOR ROOT G
C AND STORE IN AMAT ARRAY
2511 DO 210 J=1,NCP
     K = MAT(I,J)
     YA(1...0)
     IF(K=-1) 210,255,227
227 YAI=CNP(LX(SNLG(MAT(I,J,K)))+0)
     K = K-1
205 YA1=CNPLX(SNLG(MAT(I,J,K)))+0)
     YA=YA1+YA*G
     K = K-1
     IF(K) 99,210,205
210 AMAT(I,J)=YA
     JJ=1
225 DO 213 J=JJ,NCP
     IF(CABS(MAT(J,J))) 220,213,220
213 CONTINUE
     DET = (C,..,)
     GO TO 229
227 IF(J-JJ)99,230,221
221 DO 222 I=1,NCPW
     SCMPLEX = AMAT(I,J)
     AMAT(I,J)=AMAT(I,J)
222 AMAT(I,JK)=SCMPLEX
230 JSJ = JJ + 1
     DO 224 I=JSJ,NCP
     FCMPLEX=AMAT(I.JJ)/AMAT(JJ,JJ)
     IF(CABS(FCMPLEX)) 226,224,226
226 DO 223 J=JJ,NCP
223 AMAT(I,J)=AMAT(I,JJ)*AMAT(JJ,JJ)*FCMPLEX
224 CONTINUE
     JJ=JJ+1
     IF(JJ+LT,NCP) G7 TO 225
     DET=(1...)
     DO 242 J=1,NCP
242 DET=DET*AMAT(J,J)
229 GO TO 4DT, (1281,244,256)
ENTRY MAPNY (CMAT,C4,ROOTS,K4A,NCP,NP,
    1 IN, V, KOUNT, AMAT, BMAT, MAT, C5, D,NCP)
NCP=NCP+1
GO TO 10
END
SUBROUTINE JUGGLE(ROOTS,RT,KOUNT,K4A)
DOUBLE PRECISION ROOTS(I)
COMPLEX RT(1)
REAL*8 CR
DATA CF//7FFFEFEEFF//
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.0) GO TO 5
RT(K) = CMPLX(SNGL(ROOTS(2*I)),0.)
K=K+1
5 I=I+1
IF(K.GE.KA4) RETURN
GO TO 1
END
SUBROUTINE POLYRT(ACT,RO0T,KOUNT,NN,O,A,T)
DIMENSION KOUNT(3)
DOUBLE PRECISION AC(5),ROOT(5),Q(5),T(3),A(5)
DOUBLE PRECISION D2, DABS, TOL, S1, S2
COMMON /BARK/ D1,D2,X,NIX
V = V
90 IF (V(N+1)) 100,95,100
95 ROOT(N) = 0.DC
KOUNT((N+1)/2) = 0
M = N - 1
GO TO 90
100 TOL = 1.D5
IF (M - 1) 90,103,106
103 KOUNT(1) = -AC(2)/AC(1)
KOUNT(1) = 0
GO TO 460
106 KODE = -1
N = M
N1 = N + 1
K = C
DO 110 I = 1,N1
110 A(I) = AC(I)
IF(A(N-I)) 115,112,115
112 B1(1) = 1.D-5
B2(1) = 1.D-8
GO TO 120
115 B2(1) = -A(N+1)/A(N-1)
B1(1) = -B2(1)* (A(N-2)/A(N-1)) - A(N)/A(N-1)
120 IF (N - 2) 121,122,130
121 KOUNT(K+1) = C
A(2) = -A(2)/A(1)
GO TO 317
122 KOUNT(K+1) = 0
A(2) = -A(2)/A(1)
ILEO CABLE  FORTRAN T1  GRUMMAN DATA SYSTEMS

A(3) = -A(3) / A(1)
GO TO 310
130 CALL GeForce(T(N-2),Q(N))
ITER8 = 0
KEY = 30
INK = 15
MURDER=20
LOVE = 4
220 ITER8 = ITER8 + 1
230 Q(1) = A(1)
230 Q(2) = A(2) + B1(1)*Q(1)
240 Q(J) = A(J) + B1(1)*Q(J-1) + R2(1)*Q(J-2)
250 T(1) = Q(1)
260 T(2) = Q(2) + B1(1)*T(1)
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
R2(1) = R2(1) + D2
IF (KODE) 260,265,260
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) 265,265,460
265 LOVE = LOVE - 1
IF (LOVE) 220,297,220
270 IF (ITER8 - KEY) 220,271,271
271 MURDER = MURDER - 1
IF (MURDER) 479,285,272
272 KEY = KEY + INK
B2(1) = R2(1) + .5D0*(B1(1)**2)
GO TO 220
280 IF (4.D0* DABS(D1) - S1) 281,410,410
281 IF (4.D0* DABS(D2) - S2) 264,410,310
285 ITER8 = 999
290 K = K + 1
KOUNT(K) = ITER8 + 10
A(NI) = B1(1)
A(N1) = B2(1)
N = N - 2
N1 = N - 2
DO 300 I = 1,N1
300 A(I) = 0(I)
IF(DABS(B1(1)) > LT,.1D0*DSQRT1DABS(B2(1)))
B1(1) = .1D0*DSQRT1DABS(B2(1))
GO TO 120
310 DO 320 I = 1,N
X = A0(I+1)
A0(I+1) = X
320 A(I+1) = X

96
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
340 ITER8 = 0
Q(1) = A(1)
B1(1) = A(2)
350 ITER8 = ITER8 + 1
DO 340 J = 2, N1
360 Q(J) = A(J) + B1(1)* Q(J-1)
T(1) = Q(1)
DO 370 J = 1, N1
370 T(J-1) = Q(J-1) + B1(1)* T(J-2)
D1 = Q(N1) / T(N)
B1(1) = B1(1) + D1
IF (DABS(F1(1)) - TOL* DABS(D1)) 380, 390, 390
380 IF (ITER8 = 8) 350, 385, 350
385 ITER8 = 9
390 KOUNT(K+1) = ITER8
A0(2) = B1(1)
GO TO 440
400 K = K + 1
KODE = 6
B1(1) = A0(L)
B2(1) = A0(L+1)
ITER8 = KOUNT(K)
KEY = ITER8 + 8
IF (N = 2) 220, 409, 220
409 ITER8 = ITER8 + 1
410 X = B1(1)*2 + 4.00* B2(1)
IF (X) 420, 439, 400
420 A0(L) = .500* DSGRT(-X)
A0(L+1) = .500* 91(1)
KOUNT(K) = -ITER8
L = L - 2
GO TO 330
430 X = DSGRT(X)
IF (B1(1)) 432, 431, 431
432 X = -X
433 A0(L) = .500* (91(1)+ X)
A0(L+1) = -B2(1)/ A0(L)
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)
DO 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
### SAMPLE INPUT FOR ACTIVE CABLE PPOG-BASIC LONG CHAP. W COMP PRT OUT

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>a</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>.0500</td>
<td>.0824</td>
<td>.033</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>.935</td>
<td>-14.69</td>
<td>.0341</td>
<td>-1.630</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.062</td>
<td>-1109</td>
<td>-1227</td>
<td>-1.1923</td>
<td>0.0691</td>
<td>-0.6681</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.3780</td>
<td>.0824</td>
<td>.1332</td>
<td>.0005</td>
<td>-0.0005</td>
<td>-0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.2380</td>
<td>-1162</td>
<td>-0718</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.0005</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.065</td>
<td>446.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.062</td>
<td>-1109</td>
<td>-1227</td>
<td>-1.1923</td>
<td>0.0691</td>
<td>-0.6681</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.3780</td>
<td>.0824</td>
<td>.1332</td>
<td>.0005</td>
<td>-0.0005</td>
<td>-0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.2380</td>
<td>-1162</td>
<td>-0718</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.0005</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.065</td>
<td>446.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SAMPLE DATA-LONG CHAP OF THEETA/EMO TRANS FUNC W FEEDBACK & FREQ RESP.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

### SAMPLE INPUT OF VEL=C, W LIFT CABLE-CHAP. ROOTS OPTION

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>9</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SAMPLE INPUT FOR CABLELESS MODEL W TRANSFER FUNCTION OPTION

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>15</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SAMPLE OF ACTIVE CABLE SYSTEM-LAT DIP MODE W TRANS. FUNC. OP.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

100
Appendix E

Sample Output
<table>
<thead>
<tr>
<th>CODE NO. FOR THIS CASE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT DATA AS SPECIFIED IN AERO ARRAY</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>AERO( 1) ( )</td>
<td>1.0</td>
<td>AERO( 2) ( )</td>
<td>0.0</td>
<td>AERO( 3) ( )</td>
<td>0.0</td>
<td>AERO( 4) ( )</td>
<td>0.0</td>
<td>AERO( 5) ( )</td>
<td>5.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO( 6) ( )</td>
<td>1.0</td>
<td>AERO( 7) ( )</td>
<td>0.0</td>
<td>AERO( 8) ( )</td>
<td>10.9</td>
<td>AERO( 9) ( )</td>
<td>-14.7</td>
<td>AERO(10) ( )</td>
<td>0.50E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO(11) ( )</td>
<td>0.0</td>
<td>AERO(12) ( )</td>
<td>0.0</td>
<td>AERO(13) ( )</td>
<td>0.0</td>
<td>AERO(14) ( )</td>
<td>0.0</td>
<td>AERO(15) ( )</td>
<td>-1.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO(16) ( )</td>
<td>0.0</td>
<td>AERO(17) ( )</td>
<td>0.0</td>
<td>AERO(18) ( )</td>
<td>0.0</td>
<td>AERO(19) ( )</td>
<td>0.0</td>
<td>AERO(20) ( )</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO(21) ( )</td>
<td>0.0</td>
<td>AERO(22) ( )</td>
<td>0.0</td>
<td>AERO(23) ( )</td>
<td>0.0</td>
<td>AERO(24) ( )</td>
<td>0.0</td>
<td>AERO(25) ( )</td>
<td>0.378</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO(26) ( )</td>
<td>0.0</td>
<td>AERO(27) ( )</td>
<td>0.0</td>
<td>AERO(28) ( )</td>
<td>0.0</td>
<td>AERO(29) ( )</td>
<td>0.0</td>
<td>AERO(30) ( )</td>
<td>0.66E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERO(31) ( )</td>
<td>0.0</td>
<td>AERO(32) ( )</td>
<td>0.0</td>
<td>AERO(33) ( )</td>
<td>0.0</td>
<td>AERO(34) ( )</td>
<td>0.0</td>
<td>AERO(35) ( )</td>
<td>0.50E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AERO DATA IN STAB. AXIS AT EQUAT. REF. CENTER**

| AERO( 1) ( ) | 1.0 | AERO( 2) ( ) | 0.0 | AERO( 3) ( ) | 0.0 | AERO( 4) ( ) | 0.0 | AERO( 5) ( ) | 5.80 |
| AERO( 6) ( ) | 1.0 | AERO( 7) ( ) | 0.0 | AERO( 8) ( ) | 10.9 | AERO( 9) ( ) | -14.7 | AERO(10) ( ) | 0.50E-01 |
| AERO(11) ( ) | 0.0 | AERO(12) ( ) | 0.0 | AERO(13) ( ) | 0.0 | AERO(14) ( ) | 0.0 | AERO(15) ( ) | -1.63 |
| AERO(16) ( ) | 0.0 | AERO(17) ( ) | 0.0 | AERO(18) ( ) | 0.0 | AERO(19) ( ) | 0.0 | AERO(20) ( ) | 0.0 |
| AERO(21) ( ) | 0.0 | AERO(22) ( ) | 0.0 | AERO(23) ( ) | 0.0 | AERO(24) ( ) | 0.0 | AERO(25) ( ) | 0.378 |
| AERO(26) ( ) | 0.0 | AERO(27) ( ) | 0.0 | AERO(28) ( ) | 0.0 | AERO(29) ( ) | 0.0 | AERO(30) ( ) | 0.66E-01 |
| AERO(31) ( ) | 0.0 | AERO(32) ( ) | 0.0 | AERO(33) ( ) | 0.0 | AERO(34) ( ) | 0.0 | AERO(35) ( ) | 0.50E-03 |

**CABLE CONFIGURATION ON MODEL**

- **FRONT CABLE IS HORIZONTAL AND REAR CABLE IS VERTICAL**

**CABLE GEOMETRY-CABLE NO. 1**

- **CABLE LENGTH= 0.123377E-03 IN**

**CABLE GEOMETRY-CABLE NO. 2**

- **CABLE LENGTH= 0.123377E-03 IN**
CABLE GEOMETRY-CABLE NO. 3  CABLE LENGTH = 0.123383E 03 IN

\[
\begin{align*}
\cos \theta &= 0.233272E 03 \\
\cos \phi &= 0.499999E 02 \\
\sin \theta &= 0.499999E 02 \\
\sin \phi &= 0.143272E 03
\end{align*}
\]

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

\[
\begin{align*}
\cos \theta &= 0.129705E 03 \\
\cos \phi &= 0.499999E 02 \\
\sin \theta &= 0.499999E 02 \\
\sin \phi &= 0.307033E 02
\end{align*}
\]

Iteration Parameter = 4
ACCZ = -3.95420E-03
ACCK = 0.976482E-03
THEODT = 0.2491590E-03 FAD/SEC

Em. Att. Defl. C. Cable Tension
THETA = 1.02 DEG
DELTA = -1.37 DEG
FFT CAL. TENSION = 0.127501E 09 LBS
BR CAL. TENSION = 0.100214E 03 LBS

AEROP DATA IN BODY AXIS AT EQUAT. REF. CENTER
AEROP (1) = -0.085E-01  AEROP (2) = -0.770  AEROP (3) = 0.264E-01  AEROP (4) = 0.290  AEROP (5) = -5.83E-01
AEROP (6) = -1.449  AEROP (7) = 0.163  AEROP (8) = -10.2  AEROP (9) = -13.8  AEROP (10) = -0.515E-01
AEROP (11) = -0.015E-01  AEROP (12) = 0.394E-01  AEROP (13) = 0.168E-01  AEROP (14) = 0.935E-01
AEROP (15) = 0.056E-01  AEROP (16) = -7.77E-01  AEROP (17) = 0.060E-01  AEROP (18) = -1.06E-01
AEROP (19) = -0.011E-01  AEROP (20) = -0.456E-01  AEROP (21) = 0.186E-01  AEROP (22) = -0.213E-01
AEROP (23) = 0.556E-01  AEROP (24) = 0.103E-01  AEROP (25) = 0.660E-01
AEROP (26) = 0.010E-01  AEROP (27) = 0.711E-01  AEROP (28) = 0.500E-03  AEROP (29) = 0.648E-03
AEROP (30) = 0.104E-03

**** Longitudinal Stability ****

Position and Coefficients of Each Polynomial of Matrix

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>K</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Determinant AEROP = 4.52237E13F 05 0.0
Determinate AEROP = 4.52237E13F 05 0.0
Determinate AEROP = 4.52237E13F 05 0.0
Determinate AEROP = 4.52237E13F 05 0.0
Determinate AEROP = 4.52237E13F 05 0.0
### Determinant

<table>
<thead>
<tr>
<th>Determinant</th>
<th>Value</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.223611E+05</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>-5.223656E+05</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>-5.723756E+05</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>-5.227373E+05</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

### Final Matrix

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>-1.094176E+05</td>
<td>0.0</td>
</tr>
<tr>
<td>1 2</td>
<td>1.014216E+02</td>
<td>1.369000E+00</td>
</tr>
<tr>
<td>1 3</td>
<td>1.362214E+00</td>
<td>0.0</td>
</tr>
<tr>
<td>2 1</td>
<td>-7.160786E+00</td>
<td>2.080203E+01</td>
</tr>
<tr>
<td>2 2</td>
<td>3.290325E+00</td>
<td>0.0</td>
</tr>
<tr>
<td>2 3</td>
<td>-7.457466E+02</td>
<td>-9.719999E+00</td>
</tr>
<tr>
<td>3 1</td>
<td>4.057771E+03</td>
<td>8.592576E+02</td>
</tr>
<tr>
<td>3 2</td>
<td>6.879156E+01</td>
<td>-1.247069E+02</td>
</tr>
<tr>
<td>3 3</td>
<td>-1.885203E+03</td>
<td>-9.792189E+00</td>
</tr>
</tbody>
</table>

### Real Imaginary Error

<table>
<thead>
<tr>
<th>Real</th>
<th>Imaginary</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.39477E-03</td>
<td>4.329021E+00</td>
<td>-2.922648E-07</td>
</tr>
<tr>
<td>-2.905477E-03</td>
<td>-4.329021E+00</td>
<td>2.386554E+07</td>
</tr>
<tr>
<td>-2.351342E+00</td>
<td>1.359896E+01</td>
<td>-9.904530E+00</td>
</tr>
<tr>
<td>-2.351342E+00</td>
<td>-1.359896E+01</td>
<td>4.166396E+07</td>
</tr>
</tbody>
</table>

### Polynomial

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.410945E+05</td>
<td>-1.230982E+04</td>
</tr>
</tbody>
</table>

### Polynomial Constants First

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.42150E+06</td>
<td>0.123189E+05</td>
</tr>
<tr>
<td>0.282076E+05</td>
<td>0.636112E+03</td>
</tr>
<tr>
<td>0.134784E+03</td>
<td>0.6890E+00</td>
</tr>
</tbody>
</table>

### Real Imaginary T H/D Sec T H/D Period Sec DNAT CPS UNDNAT CPS DAMP RATIO DECAY RATIO

<table>
<thead>
<tr>
<th>Real</th>
<th>Imaginary</th>
<th>T H/D</th>
<th>Sec</th>
<th>T H/D</th>
<th>Period</th>
<th>Sec</th>
<th>DNAT CPS</th>
<th>UNDNAT CPS</th>
<th>DAMP RATIO</th>
<th>DECAY RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.4999E+12</td>
<td>-0.4329E+01</td>
<td>0.9253E+02</td>
<td>0.1212E+01</td>
<td>0.1451E+01</td>
<td>0.6990E+00</td>
<td>0.2938E-02</td>
<td>-0.9879E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.3510E+01</td>
<td>0.1360E+02</td>
<td>0.2948E+00</td>
<td>0.3832E+01</td>
<td>0.4820E+00</td>
<td>0.2166E+01</td>
<td>0.2166E+01</td>
<td>0.1794E+00</td>
<td>0.3374E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CASE NO. 2

SAMPLE DATA-LONG CHAR OF THETA/ENO TRANS FUNC W FEEDBACK & FREQ RESP.
FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL
NO SHUBBERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

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 2 0 0 2 0 10 3 0 0 0 1 7 1 2 0 0 -1 60 0 0 0 0 0

DATA CHANGE
137 7.5000
138 1.0000
140 0.00

FREQUENCY RESPONSE COMPUTATION

EM. ATT., DEFLEX & CABLE TENSION

THETA = 1.03 DEG
DELTA = -1.33 DEG
FRT CABLE TENSION = 0.127591E 03 LBS
RRT CABLE TENSION = 0.160216E 03 LBS
+++ LONGITUDINAL STABILITY +++
COMPUTATION OF THE END NUMERATOR ROOTS
POLYNOMIAL W. CONST TERM FIRST
0.693987E 03 0.153241E 02 0.523025E 32

REAL IMAGINARY T M/D-SEC 1/T M/D PERIOD-SEC DNATP-CPS UNONAT-CPS DAMP RATIO DELAY RATIO
-0.1465E 00 +1.4303E 01 1.2173E 01 0.2113E 00 0.1465E 00 0.6853E 00 0.3402E-01 0.8074E 00

COMPUTATION OF THE DENOMINATOR ROOTS

POLYNOMIAL W. CONST TERM FIRST
0.693987E 03 0.391416E 06 0.309395E 06 0.718007E 04 0.150616E 04

REAL IMAGINARY T M/D-SEC 1/T M/D PERIOD-SEC DNATP-CPS UNONAT-CPS DAMP RATIO DECAY RATIO
-0.3080E 00 +0.1464E 01 1.2173E 01 0.5751E 00 0.1566E 01 0.3394E 00 0.6425E 00 0.9875E-01 0.5361E 00
-0.1506E 01 +0.1560E 02 0.4346E 00 0.2302E 01 0.4003E 00 0.9964E 01 0.2491E 01 0.2511E 01 0.1011E 00 0.5279E 00
-0.3194E 03 +0.2170E-02 0.4607E 03
<table>
<thead>
<tr>
<th>GAIN</th>
<th>AMP니다</th>
<th>PHASE</th>
<th>AMP VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100E-03</td>
<td>0.7071E-02</td>
<td>C.3174E-06</td>
<td>0.1667E-03</td>
</tr>
<tr>
<td>0.125E-03</td>
<td>0.6774E-02</td>
<td>C.3174E-06</td>
<td>0.142E-03</td>
</tr>
<tr>
<td>0.150E-03</td>
<td>0.6554E-02</td>
<td>C.3174E-06</td>
<td>0.122E-03</td>
</tr>
<tr>
<td>0.175E-03</td>
<td>0.6324E-02</td>
<td>C.3174E-06</td>
<td>0.100E-03</td>
</tr>
<tr>
<td>0.200E-03</td>
<td>0.6084E-02</td>
<td>C.3174E-06</td>
<td>0.079E-03</td>
</tr>
<tr>
<td>0.225E-03</td>
<td>0.5844E-02</td>
<td>C.3174E-06</td>
<td>0.059E-03</td>
</tr>
<tr>
<td>0.250E-03</td>
<td>0.5604E-02</td>
<td>C.3174E-06</td>
<td>0.040E-03</td>
</tr>
<tr>
<td>0.275E-03</td>
<td>0.5364E-02</td>
<td>C.3174E-06</td>
<td>0.020E-03</td>
</tr>
<tr>
<td>0.300E-03</td>
<td>0.5124E-02</td>
<td>C.3174E-06</td>
<td>0.000E-03</td>
</tr>
</tbody>
</table>

FREQUENCY RESPONSE OF THE TRANSFER FUNCTION

STEADY STATE GAIN = 0.1667E-03
CODE NO'S. FOR THIS CASE.

1  -1 0
2 0 0 0 2 0 9 3 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0

DATA CHANGE

49 10.

EIH ATT. DEFLT. CABLE TENSION

THETA = +0.00 DEG
DELTA = 0.0 DEG

FR. CABLE TENSION = 0.596780E 02 LBS
FR. CABLE TENSION = 0.100000E 03 LBS

POLYNOMIAL w CONST TERM FIRST

0.736854E 06 0.231582E 04 0.659035E 02 0.204398E 03 0.141327E 04

 Imaginary 1/M-SEC 1/7 M/SEC PERIOD-SEC ONATF-CPS UNONAT-CPS DAMP RATIO DECAY RATIO

-0.3192E 03 0.2176E-02 0.4890E 03 0.1399E 01 0.7149E 00 0.7149E 00 0.6 0.1000E 01

0.0 0.5949E 01 0.1000E 00 0.0 0.1236E 01 0.4091E 00 0.0 0.1000E 01
CASE NO. 4

SAMPLE INPUT FOR CABLELESS MODEL W. TRANSFER FUNCTION OPTION
FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL

NO SNUBBERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC NOT IN CABLELESS MODEL CHARACTERISTICS

CODE NO. 74 this case:
1 2 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 3 10 0 0 0 -1 18 3 0 0 0 0 0 0 0 0

DATA CHANGE:
4 246500
44F 00

EN. ATT. (DEFLTN.) CABLE TENSION
THETA = 1.03 DEG
DELTA = -1.33 DEG

FRONT CABLE TENSION = 0.127591E 03 LBS
REAR CABLE TENSION = 0.160214E 03 LBS

LONGITUDINAL STABILITY ****

COMPUTATION OF X VEDE NUMERATOR ROOTS
POLYNOMIAL w CONSTR TERM FIRST

REAL IMAGINARY

PERIOD-SEC DNATF-CPS UNDANAT-CPS DAMP RATIO DECAY RATIO

-0.319421E 07 -0.999777E 06 -0.582650E 04 -0.156621E 04

POLYNOMIAL w CONSTR TERM FIRST

REAL IMAGINARY

PERIOD-SEC DNATF-CPS UNDANAT-CPS DAMP RATIO DECAY RATIO

-2.673390E-01 0.0 7.894731E-02 0.0

ERROR

-0.265945E 03 0.0 0.265945E 03 0.0

LATERAL/DIRECTIONAL STABILITY ****

POLYNOMIAL HAS BEEN SET BY PROG TO 3 FOR CABLELESS MODEL CHARACTERISTICS
THE FOLLOWING EXTRACTED ROOT HAVE POOR ACCURACY

REAL IMAGINARY

PERIOD-SEC DNATF-CPS UNDANAT-CPS DAMP RATIO DECAY RATIO

-0.500000E 00 0.0 0.0 0.0 0.0

UNDERSTANDING THE PROLEM OF
CASE NO. S  SAMPLE OF ACTIVE CABLE SYSTEM - LAT DIP MODE W. TRANS. FUNC. OP.
FRONT CABLE HORIZONTAL. REAR CABLE VERTICAL
NO SHAINERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

<table>
<thead>
<tr>
<th>CODE NO.</th>
<th>FOR THIS CASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

DATA CHANGE
0 0 0 0

EH. ATT. DEFLECTION 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

POLYNOMIAL W/ CONJ. TERM FIRST
-0.345004E 05 -0.170033E 05 -0.904558E 04 -0.159327E 04 -0.294450E 03

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>T H/SEC</td>
<td>1/T H/SEC</td>
</tr>
<tr>
<td>-0.12205E 01</td>
<td>-0.2390E 01</td>
</tr>
<tr>
<td>-0.1645E 01</td>
<td>-0.3755E 01</td>
</tr>
</tbody>
</table>
FIGURES
ACTIVE CABLE MOUNT SYSTEM

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

1) **LONGITUDINAL CABLE CONTROL**

2) **DIRECTIONAL CABLE CONTROL**
ACTIVE CABLE MOUNT SYSTEM

LONGITUDINAL BLOCK DIAGRAM

\[ \Delta T_i \rightarrow \Delta T_c \rightarrow \text{Basic Model Cable System} \rightarrow \theta, \beta, \phi \]

\[ \frac{1}{2\gamma_2} \]

\[ \frac{K_T}{K_e + A L_c} \]

\[ E_{m_0} \rightarrow E_{m_{tot}} \rightarrow E_m \]

\[ K_3 L \]

\[ \theta = K_4 \phi \theta, \beta, \phi \]

\[ \frac{K_7 K_6 A}{R_c + A L_c} \]

\[ J_{4L^2} + GA \]
ACTIVE CABLE MOUNT SYSTEM
LATERAL DIRECTIONAL BLOCK DIAGRAM

Figure 3
## 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>j</td>
<td>0</td>
<td>ΔT</td>
<td>x</td>
<td>ΔT</td>
<td>θ</td>
<td>E_m</td>
<td>ϕ</td>
<td>E_m</td>
<td>ΔT</td>
<td>E_m</td>
<td>ΔT</td>
</tr>
</tbody>
</table>

### Basic Matrix of Inactive Cable-Mount/Sys (see ref. 1)

<table>
<thead>
<tr>
<th>Eq.</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>0</td>
<td></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>
<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>
<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>
<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>
<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>
<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>
<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>
<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>
<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>
<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>
<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>
<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>
<td></td>
</tr>
</tbody>
</table>

### Equations

1. \( \ddot{x} - \gamma F_x - \frac{\partial F_x}{\partial T_c} \Delta T_c = 0 \)
2. \( \ddot{x} - \gamma F_z - \frac{\partial F_z}{\partial T_c} \Delta T_c = 0 \)
3. \( I_{yy} \ddot{\theta} - \gamma M_y - \frac{\partial M_y}{\partial T_c} \Delta T_c = 0 \)
4. \( x - \frac{\partial x}{\partial z} z - \frac{\partial x}{\partial \theta} \theta = 0 \)
5. \( \Delta T_{fb} (2r_d) (R_a + aL_a) - (J_a s^2 + G_s) (R_a + L_a b_m + 2 k_s \theta_m + q T E_m) \Delta T_{TOT} = 0 \)
6. \( \delta_\theta = \left[ \frac{\partial \Delta T}{\partial x} x + \frac{\partial \Delta T}{\partial z} z + \frac{\partial \Delta T}{\partial \theta} \theta \right] = 0 \)
7. \( E_m = K_{\theta_m} \theta_m + K_\theta \dot{\theta}_m + K_q q \) where \( q = \dot{\theta} \)
8. \( \ddot{\theta}_m = \theta_{ms} \)
9. \( E_{m,TOT} = E_m + W_m \)
10. \( \Delta T_c = \Delta T_i - \Delta T_{fb} \)
# Active Cable Mount System

Extended Lateral-Directional 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>y</td>
<td>ψ</td>
<td>φ</td>
<td>ΔT&lt;sub&gt;fb&lt;/sub&gt;</td>
<td>y&lt;sub&gt;m&lt;/sub&gt;</td>
<td>E&lt;sub&gt;nr&lt;/sub&gt;</td>
<td>y&lt;sub&gt;m&lt;/sub&gt;</td>
<td>ψ</td>
<td>E&lt;sub&gt;m&lt;/sub&gt;</td>
<td>ΔT&lt;sub&gt;c&lt;/sub&gt;</td>
<td>E&lt;sub&gt;m&lt;/sub&gt;</td>
<td>ΔT&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

**Basic Matrix of Inactive Cable Model System (see ref. 1):**

<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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Equations:**

1. \[ \ddot{y} + \gamma \dot{y} \Omega = \frac{\partial F}{\partial \Delta T_c} \Delta T_c = 0 \]
2. \[ I_{zz} \ddot{\psi} - I_{xz} \ddot{\phi} - \Sigma N_0 \Delta T_c = 0 \]
3. \[ I_{xx} \ddot{\phi} - I_{xz} \ddot{\psi} - \Sigma T_0 \Delta T_c = 0 \]
4. \[ \Delta T_{fb} (2r_d) (R_a + \omega L_a) - (J_{Ma} + G_s) \Delta T_{fb} + 2K_T \Omega \dot{y}_m + 2K_T E_{m} = 0 \]
5. \[ \ddot{y}_m \dot{r}_d + \left[ \frac{\partial \Delta T_c}{\partial y} \dot{y} + \frac{\partial \Delta T_c}{\partial \psi} \dot{\psi} + \frac{\partial \Delta T_c}{\partial \phi} \dot{\phi} \right] = 0 \]
6. \[ E_m = K_\psi \dot{y}_m + K_\phi \ddot{y}_m + K_\Omega \ddot{\psi} \]
7. \[ \dot{y}_m = K_\Omega y_m = 0 \]
8. \[ \dot{y} - s\dot{y} = 0 \]
9. \[ E_{m_{TOT}} = E_m + E_{m_0} \]
10. \[ \Delta T_c = \Delta T_i \Delta T_{fb} \]
Figure 8. Schematic for Snubbed Model
\[
\begin{array}{cccccc}
\text{} & \mathbf{X} & \mathbf{Z} & \Theta & \Delta \mathbf{x} & \Delta \mathbf{z} & \Delta \mathbf{T} & \Delta \mathbf{L} \\
\mathbf{X} & -2T \cos \alpha_x & -2T \sin \alpha_x & Z \cos \alpha_x & & & & \\
\mathbf{Z} & 2T \cos \alpha_x & -2T \sin \alpha_x & Z \cos \alpha_z & & & & \\
\Theta & -2T \ell_x \cos \alpha_x & -2T \ell_x \sin \alpha_x & 2T \ell_x \sin \alpha_x & 2 \ell_x \cos \alpha_x & -2 \ell_x \cos \alpha_x & & \\
\Delta \mathbf{x} & \ell_x / \ell \sin \alpha_x & -1 & & & & & \\
\Delta \mathbf{z} & \ell_z / \ell \sin \alpha_z & -1 & & & & & \\
\Delta \mathbf{T} & -\cos \alpha_x & -\cos \alpha_z & -\ell_z \cos \alpha_x & -\ell_x \cos \alpha_z & & & \\
\Delta \mathbf{L} & & & & & & -1 & \\
\end{array}
\]

**Figure 84:** Longitudinal Cable Influence Matrix
\[
\begin{array}{cccccccc}
Y & \psi & \phi & T & \Delta\alpha_x & \Delta\alpha_y & \Delta\alpha_z & \Delta R \\
-TC\cos\alpha_x & T\cos\alpha_x & \cos\alpha_y & -T\sin\alpha_x & -x_{TS}\cos\alpha_x & -y_{TS}\cos\alpha_y & -z_{TS}\cos\alpha_z & -R_{TS} \\
-x_{TC}\cos\alpha_x & x_{TC}\cos\alpha_x & x_{TC}\cos\alpha_y & -y_{TS}\sin\alpha_x & -x_{TS}\sin\alpha_y & -y_{TS}\sin\alpha_z & -z_{TS}\sin\alpha_z & -R_{TS} \\
-x_{TC}\cos\alpha_x & -x_{TC}\cos\alpha_x & -x_{TC}\cos\alpha_y & -y_{TS}\sin\alpha_x & -x_{TS}\sin\alpha_y & -y_{TS}\sin\alpha_z & -z_{TS}\sin\alpha_z & -R_{TS} \\
-\cos\psi\cot\theta & \frac{\gamma y_{TS}\sin\alpha_x}{\ell} + \frac{\gamma y_{TS}\cos\alpha_x}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} = \frac{\gamma y_{TS}\cos\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} \\
-\cos\psi\cot\theta & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} & \frac{\gamma y_{TS}\sin\alpha_y}{\ell} \\
-\cos\psi\cot\theta & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} & \frac{\gamma y_{TS}\sin\alpha_z}{\ell} \\
\end{array}
\]

**Figure 8B:** Lateral/Directional Cable Influence Matrix
Note: All variables are positive as shown.

Side view of model.

Fig. 9 - Reference center and lift cable input data.
FRONT VERTICAL - REAR VERTICAL

FRONT HORIZONTAL - REAR HORIZONTAL

FRONT VERTICAL - REAR HORIZONTAL

FRONT HORIZONTAL - REAR VERTICAL

FIG. 10 - CuLLiV GeOGrAPHy
FIG. 11 - SNUBBER CABLE ARRANGEMENT

NOTE: ALL DISTANCES ARE POSITIVE AS SHOWN.