NASA CR-132692

# 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

(NASA-CR-132692) USER'S GUIDE FOR A REVISED COMPUTER PROGRAM TO ANALYZE THE LRC 16 FOOT TRANSONIC DYNAMICS TUNNEL ACTIVE CABLE MOUNT SYSTEM (Grumman Aerospace Corp.) 129 p HC $5.75 CSCL 14B G3/09

N75-29130

Unclas
31379



Prepared under Contract No. NAS 1-10635-22 by

GRUMMAN AEROSPACE CORPORATION

BETHPAGE, N. Y.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

July, 1975

# FOREWORD

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

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

## LIST OF SYMBOLS

$E_m$ = Command voltage from feedback loop $\sim$ volts

$E_{mo}$ = Externally applied input voltage $\sim$ volts

$E_{m_{tot}}$ = 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

$K_{\dot{\theta}_m}$ = Pitch motor rate (tachometer) feedback gain $\sim$ volts/rps

$K_{\theta_m}$ = Motor position feedback gain $\sim$ volts/rad

$K_r$ = Model yaw rate feedback gain $\sim$ volts/rps

$K_{\dot{\psi}_m}$ = Yaw motor rate feedback gain $\sim$ volts/rps

$K_{\psi_m}$ = Motor position feedback gain $\sim$ volts/rad

$L_a$ = Motor armature inductance $\sim$ henry

$\mathcal{L}$ = 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

$R_d$ = 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 $(\Delta T_i - \Delta T_{fb})$ $\sim$ lbs.

$\Delta T_F$ = Front cable tension change due to fixed length contraint $\sim$ lbs.

$\Delta T_{fb}$ = one half the cable tension change due to feedback = $\delta T \sim$ lbs.

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

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

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

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

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

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

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

$B_g$ = Laterial wind gust $\sim$ rad.

$\alpha_g$ = Vertical wind gust $\sim$ rad.

$\delta_a$ = Aileron deflection $\sim$ rad.

$\delta_e$ = Elevator deflection $\sim$ rad.

$\delta_r$ = Rudder deflection $\sim$ rad.

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

$\psi_m$ = Lateral plane torque motor pulley angular displacement $\sim$ rad.

TABLE OF CONTENTS

## I. INTRODUCTION

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

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

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

## 2.0 ACTIVE FEEDBACK CONTROL SYSTEM LOGIC

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

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

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

$\theta_m$ and $\Psi_m$ 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 "+ [illegible]". Positive $\Delta T$ is an increase in cable tension and negative $\Delta T$ is a decrease in cable tension. Positive pulley displacements results in a positive rotational moment imparted by the cable onto the model. The letters "M" and "N" show the direction of the moments induced by the positive motor rotation.

Figures 2 and 3 show block diagrams of the cable mount system with feedback loops for the longitudinal and the lateral-directional modes respectively. These two figures are similar and the discussion of figure 2 applies equally to figure 3.

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

The multi-feedback loops shown represent the active feedback logic, motor dynamics and system friction. The feedback loop containing the gains $K_q$, $K_{\dot{\theta}_m}$, and $K_{\theta_m}$ are the active elements controlling the torque motor. They are respectively, the model pitch rate gyro gain, the motor rate or tachometer gain, and the motor pulley position gain. The signals emanating from these elements are summed to give a voltage $E_m$. This voltage is combined with any externally applied test voltage, $E_{mo}$, 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, "$\frac{K_T}{R_a + sL_a}$", contained in various blocks represents the torque motor characteristics. $K_T$ 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, $\Psi$, $\phi$ are the perturbation variables. The feedback gains $K_r$, $K_{\dot{\Psi}_m}$ and $K_{\Psi_m}$ 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 x 4 matrix in the upper left-hand corner of figure 4. The additional cable tension modulation due to the active feedback logic, including motor and pulley dynamics, is represented by the added $\Delta T_c$ terms in equations 1 through 3. The coefficients of $\Delta T_c$ are derived from equations 5.4-3.3 and 5.4-8 of reference 1.

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

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

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

The expanded matrices are activated in the program by KODE (13). When this code is greater than zero, the program will read in additional data to define the active feedback parameters. These parameters are tabulated in Section 5.0.

Open and closed loop characteristic roots as well as numerator roots can be derived from these matrices. The procedure for obtaining this informat'or 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 sn[illegible] 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 calcul[illegible] separately, summed and placed in the characteristic polynomial matrix. [illegible] calculations are made within subroutines LONG and LAT.

### 3.1 LONGITUDINAL AXIS

The general derivation for the longitudinal cable influence coefficients is presented in reference 1 and will not be repeated here. A 7 x 7 matrix with the form shown in Figure 8A is used to model each cable.

The matrix is reduced to a 3 x 3 in x, z, $\theta$ and put in the FXS array. The longitudinal stability is a 3 x 3 matrix in x, z, and $\theta$. The matrix no longer contains $\Delta T_F$ as an independent variable because the front cable constraint equation (no change in total front cable length) is not required in the snubbed condition. Each cable acts as an independent spring restraint.

## 3.2 LATERAL-DIRECTIONAL AXIS

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

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

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

## 4.0 ADDITIONAL PROGRAM OPTIONS

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

### 4.1 TRANSFER FUNCTION OPTIONS

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

#### 4.1.1 LONGITUDINAL AXIS

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

KODE (14) and KODE (15) are the parameters which indicate to the program whether numerator or denominator roots are to be extracted. If KODE (14) is zero, the characteristic or denominator roots are extracted. If KODE (14) is non-zero, the program assumes that numerator roots are to be extracted. The program will then replace the column defined by KODE (15) by the column defined by KODE (14) in the matrix.

The basic no feedback system transfer function can be evaluated by setting KODE (8) to 4 and KODE (14) to 10. Setting KODE (15) from 1 to 4 will determine the numerator roots of the $z/\Delta T_c$, $\theta/\Delta T_c$, $\Delta T_F/\Delta T_c$ and $x/\Delta T_c$ transfer functions. Setting KODE (14) to zero will determine the denominator roots of these transfer functions. Thus the complete transfer function can be determined. Transfer function response to either elevator or gust input is possible by setting KODE (14) to 15 or 16 respectively.

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

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

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

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

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

4.1.2 Lateral Directional Axis

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

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

Specifically, the numerator characteristics of the basic cable system 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 etting 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, $E_{mo}$, test tension, $\Delta T_i$, rudder input, $\delta_r$, aileron input, $\delta_a$, or a side wind gust, $\beta g$.

## 4.2 Frequency Response Option

The frequency response option will compute the complete transfer function according to Section 4.1; and then evaluates for the computed transfer function over a range of frequencies, the amplitude ratio in actual value, db's, and the phase angle. The option will compute up to 60 points over a 3 decade bandwidth with a maximum of 20 points 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 KODE (3) to +2. Since a complete transfer function must be generated prior to developing the frequency response data, KODE (14) and KODE (15) or KODE (16) and KODE (17) must be set to non zero values to define the desired transfer function. Two additional parameters, KODE (18) and KODE (19), must be set to define the frequency range and number of points to be computed. KODE (18) sets the order of the lowest frequency to be computed, e.g., KODE (18) set to -1 corresponds to .1 rps and a "+1" corresponds to 10 rps. KODE (19) set to 60 means sixty points are computed for the three decade bandwidth of the frequency response.

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

The information is then passed to subroutine 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 $\Delta T_F$ column in figure 4. Thus the cableless model option requires a KODE (8) of 3 reflecting a 3 x 3 matrix size.

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

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

## 5.0 INPUT DATA

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

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

```
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.

```
READ (IR,100) (AERO (I), I = 1,36)                           (3a)
100 FORMAT (6E12.5)
CALL TABIN1 (1, 36) (See Appendix A, Ref. 1)                 (3b)
```

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

```
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.

```
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.

```
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) (8)

READ (IR, 200) (KODE (I), I = 1, 24) (9)

READ (IR, 350) K,. VALUE (10)

K = element in "AERO" array to be changed

VALUE = new value of the element

If I = 1 this "READ" statement is repeated

If I = 0 the program begins computation

All succeeding cases follow the same input format starting with statement (8).

A general description of the input arrays follows:

| ARRAY | DESCRIPTION |
|---|---|
| TITLE | Alpha-numeric array containing title for each run. |
| KODE | Array specifying program options to be exercised. |
| AERO | Array containing all the input data pertaining to the model, the mount system, tunnel conditions. etc. |

A description of each element in the "KODE" and "AERO" arrays follows.

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| KODE (1) | - | Run number. |
| KODE (2) | -1 | Calculate longitudinal stability. |
| | 0 | Calculate lateral/directional stability. |
| | +1 | Calculate both longitudinal and lateral/directional stability. |
| KODE (3) | 0 | No root locus or frequency response. |
| | +1 | Do root locus. |
| | +2 | Do frequency response. |
| KODE (4) | | Element in "AERO" array to be varied for root locus. |
| KODE (5) | 0 | Basic printout. |
| | +1 | Basic printout plus various test parameters. |
| KODE (6) | +1 | Front cable vertical-rear cable horizontal. |
| | +2 | Front cable horizontal-rear cable vertical. |
| | +3 | Front and rear cable vertical. |
| | +4 | Front and rear cable horizontal |
| KODE (7) | 0 | Aero data to be input at specific mach number. |
| | +1 | Aero data to be input in the form of tables. |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| KODE (8) | +3 | Longitudinal matrix - Cableless Model (see Section 4.3) |
| | +4 | Longitudinal matrix - no stability augmentation. |
| | +5 | Longitudinal matrix - internal stability augmentation (see Section 9.0, Reference 1.) |
| | +9 | Longitudinal matrix - Open loop response of Active Cable Mount System (see Section 2.0 , 4.1.1) |
| | +10 | Longit. linal matrix - Close loop response of Active Cable Mount System (see Section 2.0, 4.1.1) |
| KODE (9) | +3 | Lateral/directional matrix - no stability augmentation or cableless model. |
| | +4 | Lateral/directional matrix - internal yaw stability augmentation, (see Section 9.0, Reference 1.) |
| | +5 | Lateral/directional matrix - internal roll and yaw stability augmentation, (see Section 9.0, Reference 1. |
| | +9 | Lateral/directional matrix - open loop response of Active Cable Mount System (see Section 2C , 4.1.2) |
| | +10 | Lateral/directional matrix - Close loop response of Active Cable Mount System (See Section 2.0 4.1.2) |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| KODE (10) | 0 | No snubbers. |
| | +1 | Analyze conventional snubbers in unsnubbed condition - see Section 8.1, Reference 1. |
| | +2 | Analyze conventional snubbers in snubbed condition - See Section 8.2, Reference 1. |
| | +3 | Analyze flying cable snubber system. |
| KODE (11) | 0 | No anti-lift cable. |
| | +1 | Anti-lift cable in. |
| KODE (12) | 0 | No unsnubbed snubber data input. |
| | +1 | Unsnubbed snubber data will be read in. |
| KODE (13) | -1 | Cableless Airframe Characteristics. (See Section 4.3) |
| | 0 | No active cable stability augmentation. |
| | +1 | Active cable stability augmentation in. (See Section 2.0) |
| KODE (14) | 0 | Longitudinal system - compute denominator characteristics only. |
| | +10 | Longitudinal system - numerator and/or frequency characteristics of inactive cable mount system for cable tension input, $\Delta Tc$. (See Section 4.1.1) |
| | +10 | Longitudinal System - numerator and/or frequency characteristics of active cable mount system open loop for cable tension input, $\Delta Tc$. (See Section 4.1.1) |

| NAME | VALUE | DESCRIPTION |
| --- | --- | --- |
| | +11 | 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) |
| | +12 | Longitudinal System - numerator and/or frequency characteristics of active cable mount system close loop response for externally applied tension, $\Delta T_1$. (See Section 4.1.1) |
| | +15 | Longitudinal system - numerator and/or frequency characteristics for pitch control response ($\delta e$) |
| | +16 | Longitudinal system - numerator and/or frequency characteristics for gust response ($\alpha_G$). |
| KODE (15) | | 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) |
| KODE (16) | 0 | Lateral/directional system - compute denominator characteristics only |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| | +10 | Lateral/directional system - numerator and/or frequency characteristics of inactive cable mount system for tension input ΔTc. (See Section 4.1.2). |
| | +10 | Lateral/directional system - numerator and/or frequency characteristics of active cable mount system open loop for tension input, ΔTc. (See Section 4.1.2) |
| | +11 | Lateral/directional system - numerator and/or frequency characteristics of active cable mount system close loop for test voltage input $E_{mo}$. (See Section 4.1.2) |
| | +12 | 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) |
| | +14 | Lateral/directional system - numerator and/or frequency characteristics for yaw control response (δr). |
| | +15 | Lateral/directional system - numerator and or frequency characteristics for roll control response (δa). |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| | +16 | Lateral/directional system - numerator and/or frequency characteristics for gust response ($\beta_G$). |
| KODE (17) | | Lateral/directional system - column number of independent variable for which numerator and/or frequency data is desired. |
| KODE (18) | | Order of lowest frequency (RPS) for frequency response data. |
| KODE (19) | | Number of data points in frequency response (Max of 60.) |

| NAME | UNITS | LABEL | DESCRIPTION | |
|---|---|---|---|---|
| AERO (1) | N.D. | CDU | $\partial C_D/\partial(u/V_o)$ | $[C_{D_u}]$ |
| AERO (2) | N.D. | CLU | $\partial C_L/\partial(u/V_o)$ | $[C_{L_u}]$ |
| AERO (3) | N.D. | CMU | $\partial C_m/\partial(u/V_o)$ | $[C_{m_u}]$ |
| AERO (4) | 1/rad | CDA | $\partial C_D/\partial(\alpha)$ | $[C_{D_\alpha}]$ |
| AERO (5) | 1/rad | CLA | $\partial C_L/\partial(\alpha)$ | $[C_{L_\alpha}]$ |
| AERO (6) | 1/rad | CMA | $\partial C_m/\partial(\alpha)$ | $[C_{m_\alpha}]$ |
| AERO (7) | N.D. | CDQ | $\partial C_D/\partial(q\bar{c}/2V_o)$ | $[C_{D_q}]$ |
| AERO (8) | N.D. | CLQ | $\partial C_L/\partial(q\bar{c}/2V_o)$ | $[C_{L_q}]$ |
| AERO (9) | N.D. | CMQ | $\partial C_m/\partial(q\bar{c}/2V_o)$ | $[C_{m_q}]$ |
| AERO (10) | N.D. | CDO | Drag coefficient at $\alpha = 0$ | $[C_{D_o}]$ |
| AERO (11) | N.D. | CLO | Lift coefficient at $\alpha = 0$ | $[C_{L_o}]$ |
| AERO (12) | N.D. | CMO | Pitching moment at $\alpha = 0$ | $[C_{m_o}]$ |
| AERO (13) | 1/rad | CDDE | $\partial C_D/\partial(\delta_e)$ | $[C_{D_{\delta_e}}]$ |
| AERO (14) | 1/rad | CLDE | $\partial C_L/\partial(\delta_e)$ | $[C_{D_{\delta_e}}]$ |
| AERO (15) | 1/rad | CMDE | $\partial C_m/\partial(\delta_e)$ | $[C_{m_{\delta_e}}]$ |
| AERO (16) | N.D. | CDAD | $\partial C_D/\partial(\dot{\alpha}\bar{c}/2V_o)$ | $[C_{D_{\dot{\alpha}}}]$ |
| AERO (17) | N.D. | CLAD | $\partial C_L/\partial(\dot{\alpha}\bar{c}/2V_o)$ | $[C_{L_{\dot{\alpha}}}]$ |
| AERO (18) | N.D. | CMAD | $\partial C_m/\partial(\dot{\alpha}\bar{c}/2V_o)$ | $[C_{m_{\dot{\alpha}}}]$ |
| AERO (19) | 1/rad | CYB | $\partial C_y/\partial(\beta)$ | $[C_{y_\beta}]$ |
| AERO (20) | 1/rad | CLB | $\partial C_\ell/\partial(\beta)$ | $[C_{\ell_\beta}]$ |
| AERO (21) | 1/rad | CNB | $\partial C_n/\partial(\beta)$ | $[C_{n_\beta}]$ |
| AERO (22) | N.D. | CYP | $\partial C_y/\partial(pb/2V_o)$ | $[C_{y_p}]$ |
| AERO (23) | N.D. | CLP | $\partial C_\ell/\partial(pb/2V_o)$ | $[C_{\ell_p}]$ |
| AERO (24) | N.D. | CNP | $\partial C_n/\partial(pb/2V_o)$ | $[C_{n_p}]$ |
| AERO (25) | N.D. | CYR | $\partial C_y/\partial(rb/2V_o)$ | $[C_{y_n}]$ |
| AERO (26) | N.D. | CLR | $\partial C_\ell/\partial(rb/2V_o)$ | $[C_{\ell_r}]$ |

| NAME | UNITS | LABEL | DESCRIPTION | |
|---|---|---|---|---|
| AERO (27) | N.D. | CNR | $\partial C_n/\partial(rb/2V_o)$ | $[C_{n_r}]$ |
| AERO (28) | 1/rad | CYDR | $\partial C_y/\partial(\delta_r)$ | $[C_{y_{\delta_r}}]$ |
| AERO (29) | 1/rad | CLDR | $\partial C_\ell/\partial(\delta_r)$ | $[C_{\ell_{\delta_r}}]$ |
| AERO (30) | 1/rad | CNDR | $\partial C_n/\partial(\delta_r)$ | $[C_{n_{\delta_r}}]$ |
| AERO (31) | 1/rad | CYDA | $\partial C_y/\partial(\delta_a)$ | $[C_{y_{\delta_a}}]$ |
| AERO (32) | 1/rad | CLDA | $\partial C_\ell/\partial(\delta_a)$ | $[C_{\ell_{\delta_a}}]$ |
| AERO (33) | 1/rad | CNDA | $\partial C_n/\partial(\delta_a)$ | $[C_{n_{\delta_a}}]$ |
| AERO (34) | 1/rad | CYDS | $\partial C_y/\partial(\delta_s)$ | $[C_{y_{\delta_s}}]$ |
| AERO (35) | 1/rad | CLDS | $\partial C_\ell/\partial(\delta_s)$ | $[C_{\ell_{\delta_s}}]$ |
| AERO (36) | 1/rad | CNDS | $\partial C_n/\partial(\delta_s)$ | $[C_{n_{\delta_s}}]$ |
| AERO (44) | in | XREF* | Distance from aerodynamic ref. center to the equation ref. center along the X body axis | |
| AERO (45) | in | ZREF | Distance from aerodynamic ref. center to the equation ref. center along the Z body axis | |
| AERO (46) | in | XCG | Distance from model mass & inertia ref. center to the equation ref. center along the X body axis | |
| AERO (47) | in | ZCG | Distance from model mass & inertia ref. center to the equation ref. center along the Z body axis | |
| AERO (48) | | AMACH | Tunnel mach number | |
| AERO (49) | ft/sec | VO | Tunnel velocity | |
| AERO (50) | slugs | AM | Model mass | |
| AERO (51) | slug/ft$^3$ | RHO | Tunnel density | |

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (52) | lbs | WT | Model weight |
| AERO (53) | ft | B | Model reference span |
| AERO (54) | ft | CBAR | Model reference chord |
| AERO (55) | $ft^2$ | SW | Model reference wing area |
| AERO (56) | $slug\text{-}ft^2$ | XIXZ | Model cross product of inertia ($I_{XZ}$) |
| AERO (57) | $slug\text{-}ft^2$ | XIXX | Model roll inertia ($I_{xx}$), body axis at C.G. |
| AERO (58) | $slug\text{-}ft^2$ | YIYY | Model pitch inertia ($I_{yy}$), body axis at C.G. |
| AERO (59) | $slug\text{-}ft^2$ | ZIZZ | Model yaw inertia ($I_{zz}$), body axis at C.G. |
| AERO (66) | in | WLUF | Water line-upper front cable tie-down point (fr. vert.) |
| AERO (67) | in | WLLF | Water line-lower front cable tie-down point (fr. vert.) |
| AERO (68) | in | WLUR | Water line-upper rear cable tie-down point (rr. vert.) |
| AERO (69) | in | WLLR | Water line-lower rear cable tie-down point (rr. vert.) |
| AERO (70) | in | WLHF | Water line-horizontal front cable tie-down point (fr. hor.) |
| AERO (71) | in | WLHR | Water line-horizontal rear cable tie-down point (rr. hor.) |
| AERO (72) | in | STAF | Station-front cable tie-down point (fr. vert. or hor.) |
| AERO (73) | in | STAR | Station-rear cable tie-down point (rr. vert. or hor.) |
| AERO (74) | in | BLHF | Butt line-horizontal front cable tie-down point (fr. hor.) |

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (75) | in | BLHR | Butt line-horizontal rear cable tie-down point (rr. hor.) |
| AERO (76) | in | WLCR | Water line-equation reference point |
| AERO (77) | in | STACR | Station - equation reference point |
| AERO (78) | in | BLCR | Butt line-equation reference point |
| AERO (79) | in | EF** | Distance along X body axis from ref. center to vertical front pulley |
| AERO (80) | in | ER | Distance along X body axis from ref. center to vertical rear pulley |
| AERO (81) | in | AF | Distance along X body axis from ref. center to horizontal front pulley |
| AERO (82) | in | AR | Distance along X body axis from ref. center to horizontal rear pulley |
| AERO (83) | in | HUCF | Distance along Z body axis from ref. center to upper front pulley |
| AERO (84) | in | HLCF | Distance along Z body axis from ref. center to lower front pulley |
| AERO (85) | in | HUCR | Distance along Z body axis from ref. center to upper rear pulley |
| AERO (86) | in | HLCR | Distance along Z body axis from ref. center to lower rear pulley |
| AERO (87) | in | DCF | Distance along Y body axis from ref. center to horizontal front pulley |
| AERO (88) | in | DCR | Distance along Y body axis from ref. center to horizontal rear pulley |

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (89) | | blank | |
| AERO (90) | in | RVF | Radius of vertical front pulley |
| AERO (91) | in | RHF | Radius of horizontal front pulley |
| AERO (92) | in | RVR | Radius of vertical rear pulley |
| AERO (93) | in | RHR | Padius of horizontal rear pulley |
| AERO (94) | lbs | TRO | Rear cable tension |
| AERO (95) | lbs/in | AKR | Rear cable spring constant |
| AERO (96) | ft lbs/rad | COU | Pulley Coulomb friction ($a_c$) |
| AERO (97) | in | STLTT | Station - lift cable tie-down point |
| AERO (98) | in | WLLTT | Water line - lift cable tie-down point |
| AERO (99) | lbs | TLFTO | Lift cable tension |
| AERO (100) | lbs/in | AKLFT | Lift cable spring constant |
| AERO (101) | | blank | |
| AERO (102) | in | ALTX* | Distance along X body axis from lift cable attachment point to the equation reference center |
| AERO (103) | in | ALTZ | Distance along Z body axis from lift cable attachment point to the equation reference center |
| (1) AERO (104) | ft lbs/rad/sec | CMP | Pulley rolling friction coefficient |
| AERO (105) | in | SNUX*** | Distance along X body axis from model upper attachment point to the equation reference center |
| AERO (106) | in | SNUY | Distance along Y body axis from model upper snubber attachment point to the equation reference center |
| AERO (107) | in | SNUZ | Distance along Z body axis from model upper snubber attachment point to the equation reference center |

(1) AERO (104) through AERO (122) refer to conventional snubbers except where noted.

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (108) | in | SNLX | Distance along X body axis from model lower snubber attachment point to the equation reference center |
| AERO (109) | in | SNLY | Distance along Y body axis from model lower snubber attachment point to the equation reference center |
| AERO (110) | in | SNLZ | Distance along Z body axis from model lower snubber attachment point to the equation reference center |
| AERO (111) | in | SNUST | Station - upper snubber tie-down point |
| AERO (112) | in | SNUWL | Water line - upper snubber tie-down point |
| AERO (113) | in | SNUBL | Butt line - upper snubber tie-down point |
| AERO (114) | in | SNLST | Station - lower snubber tie-down point |
| AERO (115) | in | SNLWL | Water line - lower snubber tie-down point |
| AERO (116) | in | SNLBL | Butt line - lower snubber tie-down point |
| AERO (117) | lbs | TUSNO | Upper snubber, snubbed tension or flying cable snubber rear cable tension. |
| AERO (118) | lbs | TLSNO | Lower snubber, snubber tension |
| AERO (119) | lbs/in | AKSNU | Upper snubber, snubbed spring constant |
| AERO (120) | lbs/in | AKSNL | Lower snubber, snubbed spring constant flying cable snubber rear cable spring constant. |
| AERO (121) | lbs/in/sec | ADSNU | Upper snubber, snubbed damping constant or flying cable snubber front cable spring constant. |

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (122) | lbs/in/sec | ADSNL | Lower snubber, snubbed damping constant. |
| AERO (123) | rad/rad/sec | AKSY | Feedback gain- yaw rate to rudder |
| AERO (124) | rad/rad/sec | AKPHI | Feedback gain - roll rate to aileron. |
| AERO (125) | rad/rad/sec | AKTHE | Feedback gain - pitch rate to elevator. |
| AERO (126) | blank | | |
| AERO (127) | sec | T1SY | Time constant for lag on yaw rate feedback. |
| AERO (128) | sec | T2PHI | Time constant for lag on roll rate feedback. |
| AERO (129) | sec | T3THE | Time constant for lag on pitch rate feedback. |
| AERO (130) | blank | | |
| AERO (131) | in-lbs/amp | AKSBT**** | Motor torque constant ($K_t$) |
| AERO (132) | volts/rad/sec | AKSBV | Motor velocity constant ($K_v$) |
| AERO (133) | in-lbs-$sec^2$ | AJASM | Motor inertia ($J_M$) |
| AERO (134) | ohms | RSBA | Motor armature resistance ($R_a$) |
| AERO (135) | henry | ELSBA | Motor armature inductance ($L_a$) |
| AERO (136) | in | RSBD | Radius of motor pulley ($r_d$) |
| AERO (137) | volts/rad/sec | AKTHD | Pulley rotation rate feedback ($K_{\dot{\theta}_m}$) |
| AERO (138) | volts/rad | AKTH | Pulley rotation displacement feedback ($K_{\theta_m}$) |
| AERO (139) | in-lbs/rad/sec | GDMP | Pulley friction (G) |
| AERO (140) | volts/rad/sec | AKQ | Model pitch rate feedback ($K_q$) |

| NAME | UNITS | LABEL | DESCRIPTION |
|---|---|---|---|
| AERO (142) | volts/rad/sec | AKPSD | Model yaw rate feedback ($K_r$) |
| AERO (143) | volts/rad | AKY | Pulley rotation displacement feedback ($K_{\Psi_m}$) |
| AERO (144) | volts/rad/sec | AKYD | Pulley rotation rate feedback ($K_{\dot{\Psi}_m}$) |

AERO (145) to AERO (160) blank

*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 TABIN1.

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

1. Barbero, P. and Chin, J.: User's Guide for a Computer Program to Analyze the LRC 16' Transonic Dynamics Tunnel Cable Mount System. NASA CR 132313, NASA Langley, Hampton, Va., Oct. 1973

2. Mc Ruer, D.T. and Bates, C.L.: Methods of Analysis and Synthesis of Piloted Aircraft Flight Control System. Bu Aer Rept AE-61-41, Bureau of Aeronautics, Navy Dept., Washington, D.C. March 1952

## 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.

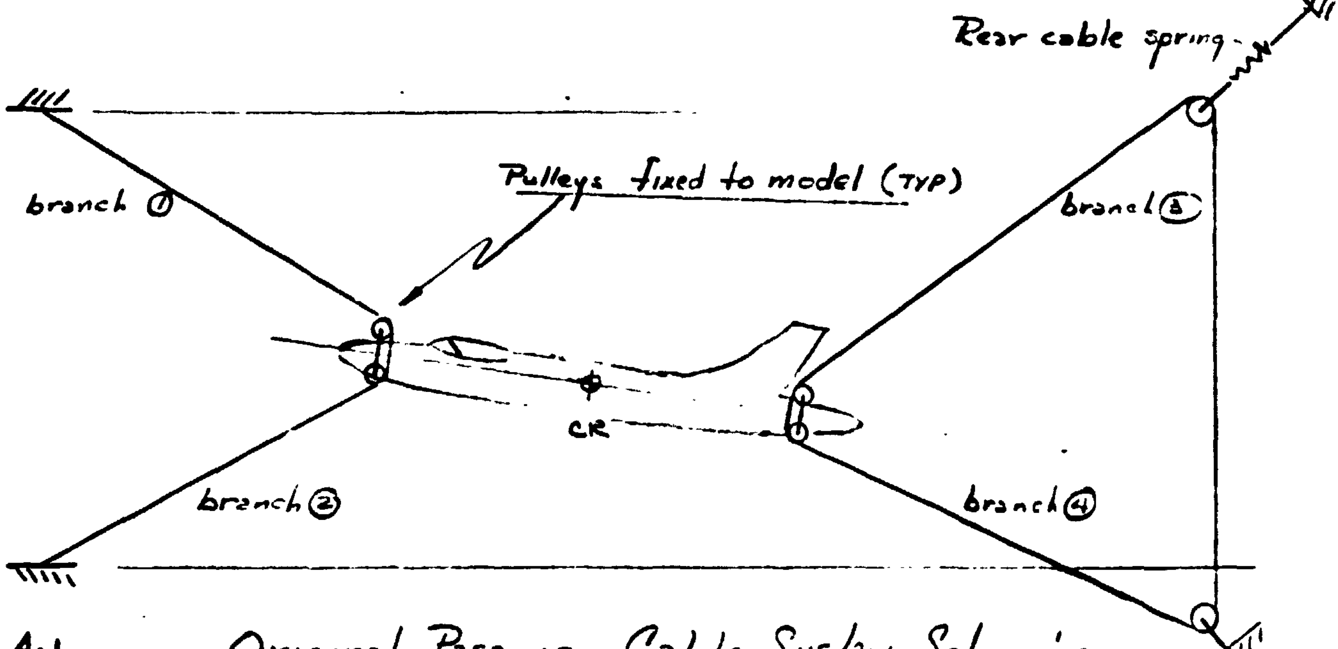



fig A·1 Original Passive Cable System Schematic

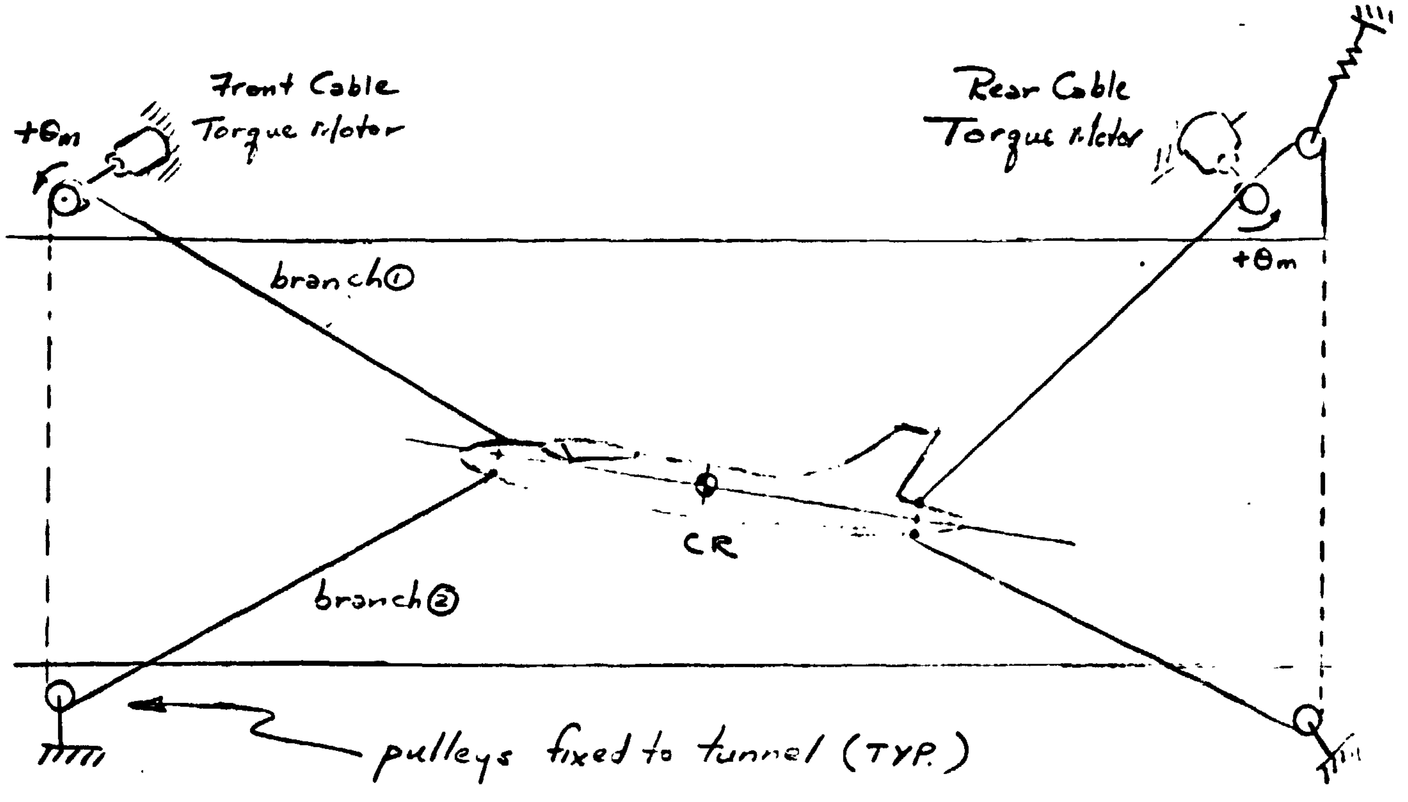



fig A·2 Active 2-Cable Mount System Schematic

## 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.

$$Q_o = K_T I_a = K_T \left[ \frac{E_{m_{TOT}} - K_V s \theta_m}{R_a + sL_a} \right] . \tag{1}$$

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

$$Q_o' = 2Q_o$$

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

$$Q_L = \Delta T_{TOT}\, r_d + Gs\theta_m \tag{2}$$

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

$$Q_o' - Q_L = J_M \ddot{\theta}_m = J_M s^2 \theta_m \tag{3}$$

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

$$\Delta T_{tot} = \frac{1}{r_d} \left\{ \left[ J_M s^2 + Gs + \frac{2K_T K_V{}' s}{R_a + sL_a} \right] \theta_m - \frac{2K_T E_{m_{TOT}}}{R_a + sL_a} \right\} \tag{4}$$

$\Delta T$ tot is positive when the cable is in tension.

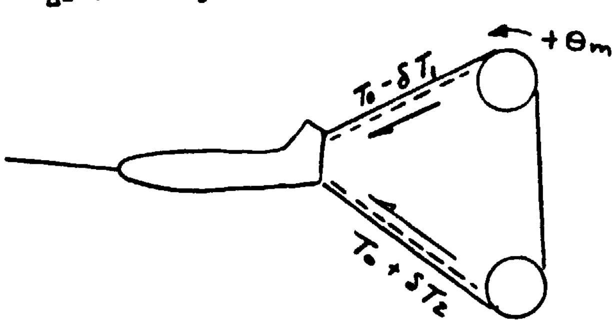



fig B-1

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

$$T_0 - \delta T_1 - (T_0 + \delta T_2) = m\ddot{a} \tag{5}$$

For $m\ddot{a} \doteq 0$

$$-\delta T_1 - \delta T_2 = 0 \tag{6}$$

and

$$\delta T_2 = -\delta T_1 \tag{7}$$

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.

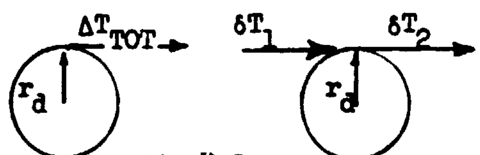



fig B-2

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

$$\delta T_2 - \delta T_1 = \Delta T_{TOT} \tag{8}$$

Substituting results from equation 7 into equation 8

$$\delta T_2 = \frac{\Delta T_{TOT}}{2} \tag{9}$$

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

$$\delta T_2 = \frac{1}{2r_d}\left\{\left[J_M s^2 + Gs + \frac{2K_T K_V s}{R_a + sL_a}\right]\theta_m - \frac{2K_T E_{mTOT}}{R_a + sL_a}\right\} \tag{10}$$

# APPENDIX C

# PROGRAM LISTINGS

```
C  THIS IS THE ACTIVE TWO CABLE MOUNT SYSTEM ANALYSES PROGRAM
C  DEVELOPED JULY,74 TO MAY,75
C
      COMMON/INPUT/IW,IR
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
      COMMON ZZZ(200)
      COMMON/TAB1/ZZ(300)
      COMMON/DU/DUM(10,10)
      COMMON/ANAME/NAME(16),NAME1(16)
      DIMENSION TITLE(20),SAVE(50),SAVE1(150),IKH(160)
      EQUIVALENCE(AERO( 1),  CDU),(AERO( 2),  CLU),(AERO( 3),  CMU),
     1           (AERO( 4),  CDA),(AERO( 5),  CLA),(AERO( 6),  CMA),
     2           (AERO( 7),  CDQ),(AERO( 8),  CLQ),(AERO( 9),  CMQ),
     3           (AERO(10),  CDO),(AERO(11),  CLO),(AERO(12),  CMO),
     4           (AERO(13), CDDE),(AERO(14), CLDE),(AERO(15), CMDE),
     5           (AERO(16), CDAD),(AERO(17),CLAD),(AERO(18),CMAD),
     6           (AERO(19),  CYB),(AERO(20),  CLB),(AERO(21),  CNB),
     7           (AERO(22),  CYP),(AERO(23),  CLP),(AERO(24),  CNP),
     8           (AERO(25),  CYR),(AERO(26),  CLR),(AERO(27),  CNR),
     9           (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR),
     A           (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA),
     B           (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS),
     C           (AERO(44), XREF),(AERO(45), ZREF),(AERO(46),  XCG),
     D           (AERO(47),  ZCG)
      EQUIVALENCE(AERO (48),AMACH),(AERO (49),VO   ),(AERO (50),    AM)
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),    WT),(AERO (53),B    )
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (59),ZIZZ )
      EQUIVALENCE(AERO (60),CLT  ),(AERO (61),CDT  ),(AERO (62),CMT  ),
     1           (AERO (63),THETA)
      EQUIVALENCE(AERO (66), WLUF),(AERO(67), WLLF),(AERO(68), WLUR),
     1           (AERO (69), WLLR),(AERO(70), WLHF),(AERO(71), WLHR),
     2           (AERO (72), STAF),(AERO(73), STAR),(AERO(74), BLHF),
     3           (AERO (75), BLHR),(AERO(76), WLCR),(AERO(77),STACR),
     4           (AERO (78), BLCR),(AERO(79),   EF),(AERO(80),   ER),
     5           (AERO (81),   AF),(AERO(82),   AR),(AERO(83), HUCF),
     6           (AERO (84), HLCF),(AERO(85), HUCR),(AERO(86), HLCR),
     7           (AERO (87),  DCF),(AERO(88),  DCR),
     8           (AERO (90),  RVF),(AERO(91),  RHF),(AERO(92),  RVR),
     9           (AERO(93),   RHR),(AERO(94),  TRC),(AERO(95),  AKR),
     A           (AERO(96),   COU),(AERO(97),STLTT),(AERO(98),WLLTT),
     B           (AERO(99),TLFTC),(AERO(100),AKLFT),
     C           (AERO(102),ALTX),(AERO(103), ALTZ),(AERO(104),  CMP)
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ),
     1           (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),
     2           (AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),
     3           (AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),
     4           (AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),
     5           (AERO(120),AKSNL),(AERO(121),ADSNU),(AERO(122),ADSNL),
     6           (AERO(123), AKSY),(AERO(124),AKPHI),(AERO(125),AKTHE),
     7           (AERO(126), AKAZ),(AERO(127), T1SY),(AERO(128),T2PHI),
     8           (AERO(129),T3THE),(AERO(130), T4AZ)
      EQUIVALENCE(AEROP( 1), CXUP),(AEROP( 2), CZUP),(AEROP( 3), CMUP),
     1           (AEROP( 4), CXAP),(AEROP( 5), CZAP),(AEROP( 6), CMAP),
```

```
     2          (AEROP( 7), CXQP),(AEROP( 8), CZQP),(AEROP( 9), CMQP), CBL00560
     3          (AEROP(10), CXQP),(AEROP(11), CZQP),(AEROP(12), CMQP), CBL00570
     4          (AEROP(13),CXDEP),(AEROP(14),CZDEP),(AEROP(15),CMDEP), CBL00580
     5          (AEROP(16),CXADP),(AEROP(17),CZADP),(AEROP(18),CMADP), CBL00590
     6          (AEROP(19), CYBP),(AEROP(20), CLBP),(AEROP(21), CNBP), CBL00600
     7          (AEROP(22), CYPP),(AEROP(23), CLPP),(AEROP(24), CNPP), CBL00610
     8          (AEROP(25), CYRP),(AEROP(26), CLRP),(AEROP(27), CNRP), CBL00620
     9          (AEROP(28),CYDRP),(AEROP(29),CLDRP),(AEROP(30),CNDRP), CBL00630
     A          (AEROP(31),CYDAP),(AEROP(32),CLDAP),(AEROP(33),CNDAP), CBL00640
     B          (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP)  CBL00650
      EQUIVALENCE (SN( 1),  GX1),(SN( 2),  GY1),(SN( 3),  GZ1),        CBL00660
     1            (SN( 4),  GX2),(SN( 5),  GY2),(SN( 6),  GZ2),        CBL00670
     2            (SN( 7),  GX3),(SN( 8),  GY3),(SN( 9),  GZ3),        CBL00680
     3            (SN(10),  GX4),(SN(11),  GY4),(SN(12),  GZ4),        CBL00690
     4            (SN(13),  THU),(SN(14),  THL),(SN(15),  ALU),        CBL00700
     5            (SN(16),  ALL),                                      CBL00710
     6            (SN(19),THGX1),(SN(20),THGY1),(SN(21),THGZ1),        CBL00720
     7            (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),        CBL00730
     8            (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),        CBL00740
     9            (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)         CBL00750
      KASE=0                                                           CBL00760
      IR=5                                                             CBL00770
      IW=6                                                             CBL00780
      LLL=0                                                            CBL00790
      DO 11 J=1,50                                                     CBL00800
   11 SAVE(J)=9999.                                                    CBL00810
      DO 12 I=1,150                                                    CBL00820
   12 AERO(I)=0.                                                       CBL00830
      LL=0                                                             CBL00840
      READ(IR,150)(TITLE(I),I=1,20)                                    CBL00850
      READ(IR,200)(KODE(I),I=1,24)                                     CBL00860
  200 FORMAT(26I3)                                                     CBL00870
      WRITE(IW,170) KODE(1),(TITLE(I),I=1,20)                          CBL00880
  170 FORMAT(1H1,3X,'CASE NO=',I3,4X,20A4)                             CBL00890
      CALL RITE                                                        CBL00900
      WRITE(IW,171)(I,I=1,24),(KODE(I),I=1,24)                         CBL00910
      IF(KODE(7).EQ.1) GO TO 10                                        CBL00920
      READ(IR,100)(AERO(I),I=1,36)                                     CBL00930
      GO TO 20                                                         CBL00940
   10 CALL TABIN1(1,36,NG)                                             CBL00950
      IF(NG.EQ.0) GO TO 20                                             CBL00960
      WRITE(IW,300) NG                                                 CBL00970
  300 FORMAT(//,' ERROR IN READING TABLES 1-36,NG=',I2)                CBL00980
      GO TO 500                                                        CBL00990
   20 READ(IR,100)(AERO(I),I=44,59)                                    CBL01000
      READ(IR,100)(AERO(I),I=66,130)                                   CBL01010
      IF(KODE(13).GT.0.)READ(IR,100)(AERO(I),I=131,160)                CBL01020
  100 FORMAT(6E12.5)                                                   CBL01030
      IF(AERO(48).EQ.0..AND.AERO(49).EQ.0.)WRITE(IW,1003)              CBL01040
 1003 FORMAT(25X,'WIND OFF CHARACTERISTICS')                           CBL01050
      IF(KODE(12).NE.1) GO TO 32                                       CBL01060
      CALL TABIN(1,2,NG)                                               CBL01070
      IF(NG.EQ.0) GO TO 32                                             CBL01080
      WRITE(IW,420) NG                                                 CBL01090
  420 FORMAT(' ERROR IN READING SNUBBER DATA TABLE,NG=',I3)            CBL01100
```

```
      GO TO 500                                                         CBL01110
 1000 DO 28 I=1,150                                                     CBL01120
   28 AERO(I)=SAVE1(I)                                                  CBL01130
      READ(IR,150,END=500)(TITLE(I),I=1,20)                             CBL01140
  150 FORMAT(20A4)                                                      CBL01150
      KASE=1                                                            CBL01160
      DO 34 J=1,50                                                      CBL01170
   34 SAVE(J)=9999.                                                     CBL01180
      READ(IR,200)(KODE(I),I=1,24)                                      CBL01190
      WRITE(IW,170) KODE(1),(TITLE(I),I=1,20)                           CBL01200
      CALL RITE                                                         CBL01210
      IKM=0                                                             CBL01220
      DO 26 I=1,160                                                     CBL01230
      READ(IR,350)K,VALUE                                               CBL01240
      IKH(I)=K                                                          CBL01250
      IF(K.LT.1)GO TO 22                                                CBL01260
      IKM=IKM+1                                                         CBL01270
      AERO(K)=VALUE                                                     CBL01280
   26 IF(K.LT.37)SAVE(K)=AERO(K)                                        CBL01290
   22 IF(AERO(48).EQ.0..AND.AERO(49).EQ.0.)WRITE(IW,1003)               CBL01300
      WRITE(IW,171)(I,I=1,24),(KODE(I),I=1,24)                          CBL01310
  171 FORMAT(//' CODE NOS. FOR THIS CASE.',/,24I5,/,24I5)               CBL01320
      WRITE(IW,352)                                                     CBL01330
  352 FORMAT(3X,'DATA CHANGE')                                          CBL01340
  350 FORMAT(I3,E12.5)                                                  CBL01350
      IF(IKM.LE.0)GO TO 24                                              CBL01360
      DO 24 I=1,?                                                       CBL01370
      K=IKH(I)                                                          CBL01380
      VALUE=AERO(K)                                                     CBL01390
   24 WRITE(IW,351)K,VALUE                                              CBL01400
  351 FORMAT(3X,I3,3X,E12.5)                                            CBL01410
      LL=0                                                              CBL01420
   32 IF(KODE(7).EQ.0) GO TO 31                                         CBL01430
      DO 30 I=1,36                                                      CBL01440
      CALL STINT1(AMACH,0.0,I,I,AERO(I),NG)                             CBL01450
      IF(NG.NE.0) GO TO 40                                              CBL01460
   30 CONTINUE                                                          CBL01470
      DO 36 J=1,36                                                      CBL01480
   36 IF(SAVE(J).NE.9999.) AERO(J)=SAVE(J)                              CBL01490
      GO TO 31                                                          CBL01500
   40 WRITE(IW,400) I,NG                                                CBL01510
  400 FORMAT(//,' ERROR IN TABLE NO',I4,'NG=',I3)                       CBL01520
      GO TO 500                                                         CBL01530
  360 FORMAT(6E10.3)                                                    CBL01540
   31 IF(KASE.EQ.1) GO TO 9                                             CBL01550
      WRITE(IW,801)                                                     CBL01560
  801 FORMAT(5X,'INPUT DATA AS SPECIFIED IN AERO ARRAY')                CBL01570
      WRITE(IW,800)(I,AERO(I),I=1,150)                                  CBL01580
  800 FORMAT(5(2X,'AERO(',I3,')=',G10.3))                               CBL01590
    9 DO 25 I=1,150                                                     CBL01600
   25 SAVE1(I)=AERO(I)                                                  CBL01610
      IF(KODE(3).EQ.0) GO TO 48                                         CBL01620
      IF(KODE(3).EQ.2)WRITE(IW,43)                                      CBL01630
   43 FORMAT('     FREQUENCY RESPONSE  COMPUTATION')                    CBL01640
      IF(KODE(3).EQ.2)GO TO 48                                          CBL01650
```

```
   42 DO 27 I=1,150                                                        CBLC1660
   27 AERO(I)=SAVE1(I)                                                     CBLC1670
      CALL OUTLOC                                                          CBLC1680
      IF(LL.EQ.0) GO TO 1000                                               CBLC1690
   48 CALL TRAN1                                                           CBLC1700
      IF(KODE(5).EQ.0) GO TO 49                                            CBLC1710
      WRITE(IW,802)                                                        CBLC1720
  802 FORMAT(4X,'AERO DATA IN STAB. AXIS AT EQUAT. REF. CENTER')          CBLC1730
      WRITE(IW,800)(I,AERO(I),I=1,36)                                      CBLC1740
   49 CALL TRIM                                                            CBLC1750
      CALL TRANS                                                           CBLC1760
      IF(KODE(5).EQ.0) GO TO 50                                            CBLC1770
      WRITE(IW,803)                                                        CBLC1780
  803 FORMAT(4X,'AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER')           CBLC1790
      WRITE(IW,804)(I,AEROP(I),I=1,36)                                     CBLC1800
  804 FORMAT(5(2X,'AEROP(',I3,')=',G10.3))                                 CBLC1810
   50 IF(KODE(2)) 70,80,90                                                 CBLC1820
   70 WRITE(IW,700)                                                        CBLC1830
  700 FORMAT('  ++++ LONGITUDINAL STABILITY ++++')                         CBLC1840
      IF(KODE(14).EQ.0)GO TO 702                                           CBLC1850
      IDX=KODE(14)                                                         CBLC1860
      IDN=KODE(15)                                                         CBLC1870
      IF(KODE(13).NE.-1.)GO TO 706                                         CBLC1880
      IF(KODE(15).EQ.3.)IDN=4                                              CBLC1890
      IF(KODE(15).LE.3.)GO TO 706                                          CBLC1900
      KODE(15)=3.                                                          CBLC1910
      WRITE(IW,707)                                                        CBLC1920
  707 FORMAT(3X,'KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION.KODE(1CBLC1930
     1 5) IS SET TO 3.')                                                   CBLC1940
  706 WRITE(IW,701)NAME(IDN),NAME(IDX)                                     CBLC1950
  701 FORMAT('  COMPUTATION OF ',A4,'/',A4,' NUMERATOR ROOTS')             CBLC1960
  702 CALL LONG                                                            CBLC1970
      IF(KODE(3).EQ.1) GO TO 42                                            CBLC1980
      GO TO 1000                                                           CBLC1990
   80 WRITE(IW,750)                                                        CBLC2000
  750 FORMAT('  ++++ LATERAL/DIRECTIONAL STABILITY ++++')                  CBLC2010
      IF(KODE(16).EQ.0)GO TO 703                                           CBLC2020
      IDX=KODE(16)                                                         CBLC2030
      IDN=KODE(17)                                                         CBLC2040
      WRITE(IW,701)NAME1(IDN),NAME1(IDX)                                   CBLC2050
  703 CALL LAT                                                             CBLC2060
      IF(KODE(3).EQ.1) GO TO 42                                            CBLC2070
      GO TO 1000                                                           CBLC2080
   90 WRITE(IW,700)                                                        CBLC2090
      IF(KODE(14).EQ.0) GO TO 704                                          CBLC2100
      IDX=KODE(14)                                                         CBLC2110
      IDN=KODE(15)                                                         CBLC2120
      IF(KODE(13).NE.-1.)GO TO 708                                         CBLC2130
      IF(KODE(15).EQ.3.)IDN=4                                              CBLC2140
      IF(KODE(15).LE.3.)GO TO 708                                          CBLC2150
      KODE(15)=3.                                                          CBLC2160
      WRITE(IW,707)                                                        CBLC2170
  708 WRITE(IW,701)NAME(IDN),NAME(IDX)                                     CBLC2180
  704 CALL LONG                                                            CBLC2190
      WRITE(IW,750)                                                        CBLC2200
```

```
      IF(KODE(16).EQ.0) GO TO 705
      IDX=KODE(16)
      IDN=KODE(17)
      WRITE(IW,701)NAME1(IDN),NAME1(IDX)
  705 CALL LAT
      IF(KODE(3).EQ.1) GO TO 42
      GO TO 1000
  500 STOP
      END
      SUBROUTINE PUTLOC
      COMMON/INPUT/IW,IR
      COMMON/DAT/AERO(175),AEROP(50),KODE(25),LL
      IF(LL.GT.0) GO TO 42
      II=KODE(4)
      VARY= ABS(AERO(II)*.1)
      ANOM= AERO(II)
      L=0
      LL=1
      WRITE(IW ,600) II
  600 FORMAT(      1H1,3X,'  ROOT LOCUS VARYING AERO(',I3,')')
   42 L=L+1
      II=KODE(4)
      AERO(II)=ANOM-5.*VARY+L*VARY
      IF(L.GT.9) GO TO 44
      WRITE(IW,180) KODE(4),AERO(II)
  180 FORMAT(/2X,5HAERO(,I3,2H)=,G12.5)
      RETURN
   44 AERO(II)=ANOM
      LL=0
      RETURN
      END
      BLOCK DATA
      COMMON/ANAME/NAME(16),NAME1(16)
      DATA NAME/'  Z ','THET',' DTF',' X ','DTFB','THTM',' EMT',
     1'THMD',' EM ',' DTC',' EMD',' DT ','THTD','    ',
     2'DELE','ALFG'/,NAME1/'  Y ',' PSI',' PHI','DTFB','PSIM',
     2' EMT','PSMD','PSID',' EM ',' DTC',' EMD',' DT ','    ',
     3'DELR','DELA','BETG'/
      END
      SUBROUTINE FREQ (ROOTS,K4A,TFG)
      COMMON/INPUT/IW,IR
      COMMON /DAT/AERO(175),AEROP(50),KODE(26),LL
      COMMON/PLOT/OM(61),AMP(61),ANGLE(61),XMP(61),KV
      COMMON/ANAME/NAME(16),NAME1(16)
      COMPLEX ROOTS(1)
      COMPLEX CNU(29)
      DIMENSION DOM(21)
      DATA DOM/1.,1.2,1.5,1.7,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,
     16.0,6.5,7.0,7.5,8.0,8.5,9.0,9.5,10./
      IL=0
      IN1=KODE(14)
      IN2=KODE(15)
      IF(KODE(13).NE.-1.)GO TO 32
      IF(KODE(15).EQ.3.)IN2=4
      IF(KODE(15).LE.3.)GO TO 32
```

```
      KODE(15)=3.                                                       CBL02760
      WRITE(IW,707)                                                     CBL02770
  707 FORMAT(3X,'KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION,KODE(1CBL02780
     1 5) IS SET TO 3.')                                                CBL02790
   32 GO TO 31                                                          CBL02800
      ENTRY FREQ2(ROOTS,K4A,TFG)                                        CBL02810
      IL=1                                                              CBL02820
      IN1=KODE(16)                                                      CBL02830
      IN2=KODE(17)                                                      CBL02840
   31 CALL ANP(CNU,0.,KN,AMPNO,PHSNO,ITYPN)                             CBL02850
      CALL ANP(ROOTS,0.,K4A,AMPDO,PHSDO,ITYPD)                          CBL02860
      TGAIN=TGN/TFG                                                     CBL02870
      SGN=ABS(TGAIN)/TGAIN                                              CBL02880
      IF(AMPDO.NE.0.)SSGN=TGAIN*AMPNO/AMPDO                             CBL02890
      ITYPE=ITYPD-ITYPN                                                 CBL02900
      IF(KODE(19).LE.10)GO TO 3                                         CBL02910
      IN=20                                                             CBL02920
      IK=1                                                              CBL02930
      GO TO 4                                                           CBL02940
    3 IF(KODE(19).LE.5)GO TO 5                                          CBL02950
      IN=10                                                             CBL02960
      IK=2                                                              CBL02970
      GO TO 4                                                           CBL02980
    5 IN=5                                                              CBL02990
      IK=4                                                              CBL03000
    4 INIT=KODE(18)                                                     CBL03010
      K=IN*3+1                                                          CBL03020
      KV=K                                                              CBL03030
      IDX=0                                                             CBL03040
      DO 1 I=1,K                                                        CBL03050
      IDX=IDX+1                                                         CBL03060
      IF(IDX.LE.IN)GO TO 2                                              CBL03070
      INIT=INIT+1                                                       CBL03080
      IDX=1                                                             CBL03090
    2 OM(I)=DOM((IDX-1)*IK+1)*(10.)**INIT                               CBL03100
      CALL ANP(CNU,OM(I),KN,AMPN,PHSN,IDUM)                             CBL03110
      CALL ANP(ROOTS,OM(I),K4A,AMPD,PHSD,IDUM)                          CBL03120
      AMP(I)=20.*(ALOG10(AMPN/AMPD)+ALOG10(ABS(TGAIN)))                 CBL03130
      XMP(I)=TGAIN*AMPN/AMPD                                            CBL03140
      ANGLE(I)=(PHSN-PHSD)*57.29578                                     CBL03150
      IF(SGN.LT.0.)ANGLE(I)=ANGLE(I)+180.                               CBL03160
    1 CONTINUE                                                          CBL03170
      IF(IL.EQ.0)WRITE(IW,10)NAME(IN2),NAME(IN1)                        CBL03180
      IF(IL.NE.0)WRITE(IW,10)NAME1(IN2),NAME1(IN1)                      CBL03190
   10 FORMAT(1H1,' FREQUENCY RESPONSE OF THE ',2X,1A4,'/',1A4,2X,       CBL03200
     1'TRANSFER FUNCTION')                                              CBL03210
      IF(AMPDO.NE.0.)WRITE(IW,17)SSGN                                   CBL03220
      IF(AMPDO.EQ.0.)WRITE(IW,18)ITYPE                                  CBL03230
   17 FORMAT('  STEADY STATE GAIN =',2X,E11.4,//)                       CBL03240
   18 FORMAT('  SYSTEM TYPE =',2X,I4)                                   CBL03250
      IF(IN.GE.20)GO TO 6                                               CBL03260
      WRITE(IW,11)                                                      CBL03270
   11 FORMAT(//,2X,' FREQ(RPS) ',2X,'AMP RAT(DB)',2X,' PHASE(DEG) '     CBL03280
     1,2X,'AMP. VALUE ')                                                CBL03290
      DO 7 I=1,K                                                        CBL03300
```

```
    7 WRITE( IW,12)OM(I),AMP(I),ANGLE(I),XMP(I)                            CBL03310
   12 FORMAT(4(2X,E11.4),5X,4(2X,E11.4))                                   CBL03320
      GO TO 8                                                              CBL03330
    6 WRITE( IW,13)                                                        CBL03340
   13 FORMAT(//,2X,' FREQ(RPS) ',2X,'AMP RAT(DB)',2X,' PHASE(DEG) ',2X     CBL03350
     1,'AMP. VALUE ',7X,                                                   CBL03360
     2' FREQ(RPS) ',2X,'AMP RT(DB)',2X,' PHASE(DEG) ',2X,'AMP. VALUE ')    CBL03370
      K=K/2+1                                                              CBL03380
      DO 9 I=1,K                                                           CBL03390
      IF(I.NE.K)GO TO 9                                                    CBL03400
      WRITE( IW,15)OM( I+30),AMP(I+30),ANGLE(I+30),XMP(I+30)               CBL03410
   15 FORMAT(57X,4(2X,E11.4))                                              CBL03420
      GO TO 8                                                              CBL03430
    9 WRITE( IW,12)OM(I),AMP(I),ANGLE(I),XMP(I),OM(30+I),AMP(30+I),        CBL03440
     1ANGLE(30+I),XMP(30+I)                                                CBL03450
    8 WRITE( IW,14)                                                        CBL03460
   14 FORMAT(1H1)                                                          CBL03470
      RETURN                                                               CBL03480
      ENTRY FREQ1(ROOTS,K4A,TFG)                                           CBL03490
      KN=K4A                                                               CBL03500
      TGN=TFG                                                              CBL03510
      IF(KN.EQ.0)RETURN                                                    CBL03520
      DO 20 I=1,K4A                                                        CBL03530
      CNU(I)=ROOTS(I)                                                      CBL03540
   20 CONTINUE                                                             CBL03550
      RETURN                                                               CBL03560
C     DEBUG UNIT(3), INIT                                                  CBL03570
      END                                                                  CBL03580
      SUBROUTINE ANP(CXU,OM,KX,AMP,ANG,ITYPE)                              CBL03590
      DIMENSION CXU(2,1)                                                   CBL03600
      ITYPE=0                                                              CBL03610
      ANG=0.                                                               CBL03620
      AMP=1.0                                                              CBL03630
      IF(KX.EQ.0)RETURN                                                    CBL03640
      DO 1 I=1,KX                                                          CBL03650
      XRL=-CXU(1,I)                                                        CBL03660
      YIM=OM-CXU(2,I)                                                      CBL03670
      AMP=SQRT(XRL*XRL+YIM*YIM)*AMP                                        CBL03680
      IF(XRL.EQ.0..AND.YIM.EQ.0.)GO TO 2                                   CBL03690
      ANG=ATAN2(YIM,XRL)+ANG                                               CBL03700
      GO TO 1                                                              CBL03710
    2 ANG=ANG                                                              CBL03720
      ITYPE=ITYPE+1                                                        CBL03730
    1 CONTINUE                                                             CBL03740
      RETURN                                                               CBL03750
C     DEBUG UNIT(3), INIT(ANG,XRL,YIM)                                     CBL03760
      END                                                                  CBL03770
      SUBROUTINE TRANS                                                     CABC0010
C  THIS ROUTINE CALCULATES BODY AXIS AERO DATA AT CR FROM STAB.            CABC0020
C  AXIS AERO DATA AT CR                                                    CABC0030
      COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL                         CABC0040
      EQUIVALENCE(AERO( 1),  CDU),(AERO( 2),  CLU),(AERO( 3),  CMU),       CABC0050
     1            (AERO( 4),  CDA),(AERO( 5),  CLA),(AERO( 6),  CMA),       CABC0060
     2            (AERO( 7),  CDO),(AERO( 8),  CLQ),(AERO( 9),  CMQ),       CABC0070
     3            (AERO(10),  CDD),(AERO(11),  CLD),(AERO(12),  CMD),       CABC0080
```

```
     4          (AERO(13), CDDE),(AERO(14), CLDE),(AERO(15), CMDE),       CAB00090
     5          (AERO(16), CDAD),(AERO(17),CLAD),(AERO(18),CMAD),         CAB00100
     6          (AERO(19),  CYB),(AERO(20),  CLB),(AERO(21),   CNB),      CAB00110
     7          (AERO(22),  CYP),(AERO(23),  CLP),(AERO(24),   CNP),      CAB00120
     8          (AERO(25),  CYR),(AERO(26),  CLR),(AERO(27),   CNR),      CAB00130
     9          (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR),       CAB00140
     A          (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA),       CAB00150
     B          (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS),       CAB00160
     C          (AERO(44), XREF),(AERO(45), ZREF),(AERO(46),  XCG),       CAB00170
     D          (AERO(47),  ZCG),(AERO(63),THETA)                         CAB00180
      EQUIVALENCE(AEROP( 1), CXUP),(AEROP( 2), CZUP),(AEROP( 3), CMUP),   CAB00190
     1          (AEROP( 4), CXAP),(AEROP( 5), CZAP),(AEROP( 6), CMAP),    CAB00200
     2          (AEROP( 7), CXQP),(AEROP( 8), CZQP),(AEROP( 9), CMQP),    CAB00210
     3          (AEROP(10), CXOP),(AEROP(11), CZOP),(AEROP(12), CMOP),    CAB00220
     4          (AEROP(13),CXDEP),(AEROP(14),CZDEP),(AEROP(15),CMDEP),    CAB00230
     5          (AEROP(16),CXADP),(AEROP(17),CZADP),(AEROP(18),CMADP),    CAB00240
     6          (AEROP(19), CYBP),(AEROP(20), CLBP),(AEROP(21), CNBP),    CAB00250
     7          (AEROP(22), CYPP),(AEROP(23), CLPP),(AEROP(24), CNPP),    CAB00260
     8          (AEROP(25), CYRP),(AEROP(26), CLRP),(AEROP(27), CNRP),    CAB00270
     9          (AEROP(28),CYDRP),(AEROP(29),CLDRP),(AEROP(30),CNDRP),    CAB00280
     A          (AEROP(31),CYDAP),(AEROP(32),CLDAP),(AEROP(33),CNDAP),    CAB00290
     B          (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP)     CAB00300
      ALPHA=THETA                                                         CAB00310
      SNALF= SIN(ALPHA)                                                   CAB00320
      COALF= COS(ALPHA)                                                   CAB00330
      SNSQ = SNALF**2                                                     CAB00340
      COSQ = COALF**2                                                     CAB00350
      SNCO = SNALF*COALF                                                  CAB00360
      CDU=CDU+2.*(CDO+CDA*THETA)                                          CAB00370
      CLU=CLU+2.*(CLO+CLA*THETA)                                          CAB00380
      CDA=CDA-(CLO+CLA*THETA)                                             CAB00390
      CLA=CLA+CDO+CDA*THETA                                               CAB00400
      CXUP=-CLA*SNSQ-CDU*COSQ+(CDA+CLU)*SNCO                              CAB00410
      CZUP= CDA*SNSQ-CLU*COSQ+(CLA-CDU)*SNCO                              CAB00420
      CMUP= -CMA *SNALF+ CMU *COALF                                       CAB00430
      CXAP= CLU*SNSQ-CDA*COSQ+(CLA-CDU)*SNCO                              CAB00440
      CZAP=-CDU*SNSQ-CLA*COSQ-(CDA+CLU)*SNCO                              CAB00450
      CMAP=  CMU *SNALF+ CMA *COALF                                       CAB00460
      CXQP= CLQ*SNALF-CDQ*COALF                                           CAB00470
      CZQP=-(CDQ*SNALF+CLQ*COALF)                                         CAB00480
      CMQP= CMQ                                                           CAB00490
      CZADP=-CLAD*COALF+CDAD*SNALF                                        CAB00500
      CXADP=-CDAD*COALF-CLAD*SNALF                                        CAB00510
      CMADP= CMAD                                                         CAB00520
      CXDEP= CLDE*SNALF-CDDE*COALF                                        CAB00530
      CZDEP=-CDDE*SNALF-CLDE*COALF                                        CAB00540
      CMDEP= CMDE                                                         CAB00550
      CXOP=-CDO*COALF-CLO*SNALF                                           CAB00560
      CZOP=-CLO*COALF+CDO*SNALF                                           CAB00570
      CMOP=CMO                                                            CAB00580
      CYBP= CYB                                                           CAB00590
      CNBP=  CLB *SNALF+ CNB *COALF                                       CAB00600
      CLBP= -CNB *SNALF+ CLB *COALF                                       CAB00610
      CYPP= (-CYR*SNALF+ CYP*COALF)                                       CAB00620
      CNPP=(-CLR*SNSQ+ CNP*COSQ+ (CLP- CNR)*SNCO)                         CAB00630
```

```
      CLPP=( CNR*SNSQ+ CLP*COSQ- (CLR+ CNP)*SNCO)
      CYRP= ( CYP*SNALF+ CYR*COALF)
      CNRP=( CLP*SNSQ+ CNR*COSQ+ (CLR+ CNP)*SNCO)
      CLRP=(-CNP*SNSQ+ CLR*COSQ+ (CLP- CNR)*SNCO)
      CYDAP= CYDA
      CNDAP= CLDA*SNALF+ CNDA*COALF
      CLDAP= -CNDA*SNALF+ CLDA*COALF
      CYDRP= CYDR
      CNDRP= CLDR*SNALF+ CNDR*COALF
      CLDRP=-CNDR*SNALF+ CLDR*COALF
      CYDSP= CYDS
      CLDSP=-CNDS*SNALF+ CLDS*COALF
      CNDSP= CLDS*SNALF+ CNDS*COALF
      RETURN
      END
      SUBROUTINE TRAN1
C  THIS ROUTINE TRANSFORMS INERTIA DATA & STABILITY AXIS AERO DATA
C  TO THE EQUATION REFERENCE CENTER
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
      EQUIVALENCE(AERO( 1),  CDU),(AERO( 2),  CLU),(AERO( 3),  CMU),
     1           (AERO( 4),  CDA),(AERO( 5),  CLA),(AERO( 6),  CMA),
     2           (AERO( 7),  CDQ),(AERO( 8),  CLQ),(AERO( 9),  CMQ),
     3           (AERO(10),  CDD),(AERO(11),  CLD),(AERO(12),  CMD),
     4           (AERO(13), CDDE),(AERO(14), CLDE),(AERO(15), CMDE),
     5           (AERO(16), CDAD),(AERO(17),CLAD),(AERO(18),CMAD),
     6           (AERO(19),  CYB),(AERO(20),  CLB),(AERO(21),  CNB),
     7           (AERO(22),  CYP),(AERO(23),  CLP),(AERO(24),  CNP),
     8           (AERO(25),  CYR),(AERO(26),  CLR),(AERO(27),  CNR),
     9           (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR),
     A           (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA),
     B           (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS),
     C           (AERO(44), XREF),(AERO(45), ZREF),(AERO(46),  XCG),
     D           (AERO(47),  ZCG),(AERO(63),THETA)
      EQUIVALENCE(AERO (48),AMACH),(AERO (49),VO   ),(AERO (50),   AM)
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),   WT),(AERO (53),B    )
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (59),ZIZZ )
      EQUIVALENCE(AERO (60),CLT  ),(AERO (61),CDT  ),(AERO (62),CMT  )
C   INERTIA TRANSFORMATIONS
      X=XCG/12.
      Z=ZCG/12.
      XIXX=XIXX+AM*(Z**2)
      YIYY=YIYY+AM*(X**2)+AM*(Z**2)
      ZIZZ=ZIZZ+AM*(X**2)
      XIXZ=XIXZ-AM*X*Z
C   AERO DATA TRANSFORMATIONS
      X=XREF/(12.*CBAR)
      Z=ZREF/(12.*CBAR)
      CMO=CMO-Z*CDO+X*CLO
      CMQ=CMQ-X*(-CLQ+2.*CMA)-2.*X*X*CLA-Z*CDQ+2.*X*Z*CDA
      CLQ=CLQ-2.*X*CLA+4.*Z*CLD
      CDQ=CDQ-2.*X*CDA+4.*Z*CDD
      CMA=CMA-Z*CDA+X*CLA
      CMDE=CMDE-Z*CDDE+X*CLDE
      X=XREF/(12.*B)
```

```
      Z=ZREF/(12.*8)                                                    CABC1190
      CNR=CNR+X*(2.*CNB+CYR+2.*X*CYB)                                   CABC1200
      CLR=CLR+X*(CLB-Z*CYB)-Z*CYR                                       CABC1210
      CNP=CNP-2.*Z*(CNB+X*CYB)+X*CYP                                    CABC1220
      CLP=CLP-Z*(CYP-2.*Z*CYB)-2.*Z*CLB                                 CABC1230
      CYR=CYR+2.*X*CYB                                                  CABC1240
      CYP=CYP-2.*Z*CYB                                                  CABC1250
      CNB=CNB+X*CYB                                                     CABC1260
      CNDR=CNDR+X*CYDR                                                  CABC1270
      CNDA=CNDA+X*CYDA                                                  CABC1280
      CNDS=CNDS+X*CYDS                                                  CABC1290
      CLB=CLB-Z*CYB                                                     CABC1300
      CLDR=CLDR-Z*CYDR                                                  CABC1310
      CLDA=CLDA-Z*CYDA                                                  CABC1320
      CLDS=CLDS-Z*CYDS                                                  CABC1330
      RETURN                                                            CABC1340
      END                                                               CABC1350
      SUBROUTINE LATSN                                                  CABC0010
      COMMON/INOUT/IW,IR                                                CABC0020
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL                        CABC0030
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)                CABC0040
      COMMON ZZZ(200)                                                   CABC0050
      COMMON/DU/DUM(10,10)                                              CABC0060
      COMMON/TAB1/ZZ(800)                                               CABC0070
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ), CABC0080
     1(AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),            CABC0090
     2(AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),            CABC0100
     3(AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),            CABC0110
     4(AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),            CABC0120
     5(AERO(120),AKSNL),(AERO(49),   VO),(AERO(51),   RHO),             CABC0130
     6(AERO(63),THETA),(AERO(121),ADSNU),(AERO(122),ADSNL)              CABC0140
      EQUIVALENCE (SN( 1),  GX1),(SN( 2),  GY1),(SN( 3),  GZ1),         CABC0150
     1(SN( 4),  GX2),(SN( 5),  GY2),(SN( 6),  GZ2),                     CABC0160
     2(SN( 7),  GX3),(SN( 8),  GY3),(SN( 9),  GZ3),                     CABC0170
     3(SN(10),  GX4),(SN(11),  GY4),(SN(12),  GZ4),                     CABC0180
     4(SN(13),  THU),(SN(14),  THL),(SN(15),  ALU),                     CABC0190
     5(SN(16),  ALL),                                                   CABC0200
     6(SN(19),THGX1),(SN(20),THGY1),(SN(21),THGZ1),                     CABC0210
     7(SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),                     CABC0220
     8(SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),                     CABC0230
     9(SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)                      CABC0240
      DIMENSION TOPR(3,3),TOPL(3,3),BOTR(3,3),BOTL(3,3)                 CABC0250
      COT(BBB)=1./TAN(BBB)                                              CABC0260
      GXY(A,AA,C)              = (-A*COT(AA)/C)*12.                     CABC0270
      GXSY(A,AA,C,D,E,F)       = -(A*SIN(AA)+C*D*COT(E))/F              CABC0280
      GXPHI(A,AA,C,D,E,F,G)    = (A*AA*COT(C)-D*E*COT(F))/G             CABC0290
      GYY(A,AA)                = (SIN(A)/AA)*12.                        CABC0300
      GYSY(A,AA,C,D,E,F)       = (A*AA*COT(C)+D*SIN(E))/F               CABC0310
      GYPHI(A,AA,C,D,E,F)      = -(A*SIN(AA)+C*D*COT(E))/F              CABC0320
      GZY(A,AA,C)              = (-A*COT(AA)/C)*12.                     CABC0330
      GZSY(A,AA,C,D,E,F,G)     = (A*AA*COT(C)-D*E*COT(F))/G             CABC0340
      GZPHI(A,AA,C,D,E,F)      = (A*AA*COT(C)+D*SIN(E))/F               CABC0350
      ALY(A)                  = -A                                      CABC0360
      ALSY(A,AA,C,D)          =  (A*AA-C*D)/12.                         CABC0370
      ALPHI(A,AA,C,D)         =  (A*AA-C*D)/12.                         CABC0380
```

```
      DO 1005 I=1,3
      DO 1005 J=1,3
      SNU(I,J)=0
 1005 SNUD(I,J)=0
      DO 1006 I=1,10
      DO 1006 J=1,10
 1006 DUM(I,J)=0
      IF(KODE(10).EQ.0) GO TO 1002
C  TERMS FOR SNUBBER EFFECTS (LAT)
      CALL DRCSN(THETA)
      IF(KODE(10).EQ.1) CALL DRCUSN(THETA)
      DUM(1,2) = -TUSNO*GX1
      DUM(1,3) =  TUSNO*GZ1
      DUM(1,5) =  -TUSNO*SIN(THGY1)
      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(THGY1)/12.
      DUM(3,6) =   SNUY*TUSNO*SIN(THGZ1)/12.
      DUM(3,7) =  (-SNUY*GZ1+SNUZ*GY1)/12.
      DUM(4,1) =   GXY(GY1,THGX1,ALU)
      DUM(4,2) =   GXSY(-SNUY,THGX1,-SNUX,GY1,THGX1,ALU)
      DUM(4,3) =   GXPHI(-SNUZ,GY1,THGX1,-SNUY,GZ1,THGX1,ALU)
      DUM(4,4) =  -1.
      DUM(5,1) =   GYY(THGY1,ALU)
      DUM(5,2) =   GYSY(-SNUY,GX1,THGY1,-SNUX,THGY1,ALU)
      DUM(5,3) =   GYPHI(-SNUZ,THGY1,-SNUY,GZ1,THGY1,ALU)
      DUM(5,5) =  -1.
      DUM(6,1) =   GZY(GY1,THGZ1,ALU)
      DUM(6,2) =   GZSY(-SNUY,GX1,THGZ1,-SNUX,GY1,THGZ1,ALU)
      DUM(6,3) =   GZPHI(-SNUZ,GY1,THGZ1,-SNUY,THGZ1,ALU)
      DUM(6,6) =  -1.
      IF(KODE(10).EQ.2) GO TO 1010
      CALL DRCSN(THETA)
      Q=.5*RHO*VO*VO
      ALU1=ALU+1.
       CALL STINT(Q,ALU1,0,1,1,TUSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALU2=ALU-1.
      CALL STINT(Q,ALU2,0,1,1,TUSN2,NG)
      IF(NG.NE.0) GO TO 5000
      GO TO 5001
 5000 WRITE(IW,5002) NG,ALL,ALU,G
 5002 FORMAT('ERROR IN SNUBBER TABLE 1,NG=',I2,3XF10.3)
      RETURN
 5001 CONTINUE
      AKTU=(TUSN1-TUSN2)/2.
      AKSNU=AKTU
 1010 CONTINUE
      DUM(7,7) =  -1.
```

```
      DUM(7,8) =   AKSNU*12.
      DUM(8,1) =   ALY(GY1)
      DUM(8,2) =   ALSY(-SNUY,GX1,-SNUX,GY1)
      DUM(8,3) =   ALPHI(-SNUZ,GY1,-SNUY,GZ1)
      DUM(8,8) =  -1.
      IF(KODE(10).EQ.1) GO TO 1015
      DO 1016 I=1,3
      DO 1016 J=1,3
 1016 SNUD(I,J)=DUM(I,7)*ADSNU*DUM(8,J)*12.
 1015 CALL MASH(3,8)
      DO 1050 I=1,3
      DO 1050 J=1,3
 1050 TOPR(I,J)= DUM(I,J)
      IF(KODE(10).EQ.1) CALL DRCUSN(THETA)
      DUM(1,2) =  -TUSNO*GX2
      DUM(1,3) =   TUSNO*GZ1
      DUM(1,5) =  -TUSNO*SIN(THGY2)
      DUM(1,7) =  GY2
      DUM(2,2) =   SNUX*TUSNO*GX2/12.-SNUY*TUSNO*GY2/12.
      DUM(2,3) =  -SNUX*TUSNO*GZ2/12.
      DUM(2,4) =   SNUY*TUSNO*SIN(THGX2)/12.
      DUM(2,5) =   SNUX*TUSNO*SIN(THGY2)/12.
      DUM(2,7) =  (-SNUX*GY2-SNUY*GX2)/12.
      DUM(3,2) =  -SNUZ*TUSNO*GX2/12.
      DUM(3,3) =   SNUZ*TUSNO*GZ2/12.-SNUY*TUSNO*GY2/12.
      DUM(3,5) =  -SNUZ*TUSNO*SIN(THGY2)/12.
      DUM(3,6) =  -SNUY*TUSNO*SIN(THGZ2)/12.
      DUM(3,7) =   (SNUY*GZ2+SNUZ*GY2)/12.
      DUM(4,1) =   GXY(GY2,THGX2,ALU)
      DUM(4,2) =   GXSY(SNUY,THGX2,-SNUX,GY2,THGX2,ALU)
      DUM(4,3) =   GXPHI(-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) =   GYPHI(-SNUZ,THGY2,SNUY,GZ2,THGY2,ALU)
      DUM(5,5) =  -1.
      DUM(6,1) =   GZY(GY2,THGZ2,ALU)
      DUM(6,2) =   GZSY(SNUY,GX2,THGZ2,-SNUX,GY2,THGZ2,ALU)
      DUM(6,3) =   GZPHI(-SNUZ,GY2,THGZ2,SNUY,THGZ2,ALU)
      DUM(6,6) =  -1.
      IF(KODE(10).EQ.2) GO TO 1020
      CALL DRCSN(THETA)
      ALU1=ALU+1.
       CALL STINT(0,ALU1,0,1,1,TUSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALU2=ALU-1.
      CALL STINT(0,ALU2,0,1,1,TUSN2,NG)
      IF(NG.NE.0) GO TO 5000
      AKTU=(TUSN1-TUSN2)/2.
      AKSNU=AKTU
 1020 CONTINUE
      DUM(7,7) =  -1.
      DUM(7,8) =   AKSNU*12.
      DUM(8,1) =   ALY(GY2)
      DUM(8,2) =   ALSY(SNUY,GX2,-SNUX,GY2)
```

```
      DUM(8,3) =   ALPHI(-SNUZ,GY2,SNUY,GZ2)
      DUM(8,8) =  -1.
      IF(KODE(10).EQ.1) GO TO 1025
      DO 1026 I=1,3
      DO 1026 J=1,3
 1026 SNUD(I,J) =SNUD(I,J)+DUM(I,7)*ABSNU*DUM(8,J)*12.
 1025 CALL MASH(3,8)
      DO 1060 I=1,3
      DO 1060 J=1,3
 1060 TOPL(I,J)= DUM(I,J)
      IF(KODE(10).EQ.1) CALL DRCUSN(THETA)
      DUM(1,2) =  -TLSNO*GX3
      DUM(1,3) =   TLSNO*GZ3
      DUM(1,5) =  -TLSNO*SIN(THGY3)
      DUM(1,7) =  GY3
      DUM(2,2) =   SNLX*TLSNO*GX3/12.-SNLY*TLSNO*GY3/12.
      DUM(2,3) =  -SNLX*TLSNO*GZ3/12.
      DUM(2,4) =   SNLY*TLSNO*SIN(THGX3)/12.
      DUM(2,5) =   SNLX*TLSNO*SIN(THGY3)/12.
      DUM(2,7) =  (-SNLX*GY3-SNLY*GX3)/12.
      DUM(3,2) =   SNLZ*TLSNO*GX3/12.
      DUM(3,3) =  -SNLZ*TLSNO*GZ3/12.-SNLY*TLSNO*GY3/12.
      DUM(3,5) =   SNLZ*TLSNO*SIN(THGY3)/12.
      DUM(3,6) =  -SNLY*TLSNO*SIN(THGZ3)/12.
      DUM(3,7) =   (SNLY*GZ3-SNLZ*GY3)/12.
      DUM(4,1) =   GXY(GY3,THGX3,ALL)
      DUM(4,2) =   GXSY(SNLY,THGX3,-SNLX,GY3,THGX3,ALL)
      DUM(4,3) =   GXPHI(SNLZ,GY3,THGX3,SNLY,GZ3,THGX3,ALL)
      DUM(4,4) =  -1.
      DUM(5,1) =   GYY(THGY3,ALL)
      DUM(5,2) =   GYSY(SNLY,GX3,THGY3,-SNLX,THGY3,ALL)
      DUM(5,3) =   GYPHI(SNLZ,THGY3,SNLY,GZ3,THGY3,ALL)
      DUM(5,5) =  -1.
      DUM(6,1) =   GZY(GY3,THGZ3,ALL)
      DUM(6,2) =   GZSY(SNLY,GX3,THGZ3,-SNLX,GY3,THGZ3,ALL)
      DUM(6,3) =   GZPHI(SNLZ,GY3,THGZ3,SNLY,THGZ3,ALL)
      DUM(6,6) =  -1.
      IF(KODE(10).EQ.2) GO TO 1030
      CALL DRCSN(THETA)
      ALL1=ALL+1.
      CALL STINT(O,ALL1,C,1,1,TLSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALL2=ALL-1.
      CALL STINT(O,ALL2,C,1,1,TLSN2,NG)
      IF(NG.NE.0) GO TO 5000
       AKTL=(TLSN1-TLSN2)/2.
      AKSNL=AKTL
 1030 CONTINUE
      DUM(7,7) =  -1.
      DUM(7,8) =   AKSNL*12.
      DUM(8,1) =   ALY(GY3)
      DUM(8,2) =   ALSY(SNLY,GX3,-SNLX,GY3)
      DUM(8,3) =   ALPHI(SNLZ,GY3,SNLY,GZ3)
      DUM(8,8) =  -1.
      IF(KODE(10).EQ.1) GO TO 1035
```

```
      DO 1036 I=1,3
      DO 1036 J=1,3
 1036 SNUD(I,J)=SNUD(I,J)+DUM(I,7)*ADSNL*DUM(8,J)*12.
 1035 CALL MASH(3,8)
      DO 1070 I=1,3
      DO 1070 J=1,3
 1070 ROTL(I,J)= DUM(I,J)
      IF(KODE(10).EQ.1) CALL DRCUSN(THETA)
      DUM(1,2) =  -TLSNO*GX4
      DUM(1,3) =   TLSNO*GZ4
      DUM(1,5) =  -TLSNO*SIN(THGY4)
      DUM(1,7) =   GY4
      DUM(2,2) =    SNLX*TLSNO*GX4/12.+SNLY*TLSNO*GY4/12.
      DUM(2,3) =  -SNLX*TLSNO*GZ4/12.
      DUM(2,4) =  -SNLY*TLSNO*SIN(THGX4)/12.
      DUM(2,5) =    SNLX*TLSNO*SIN(THGY4)/12.
      DUM(2,7) =  (-SNLX*GY3+SNLY*GX4)/12.
      DUM(3,2) =    SNLZ*TLSNO*GX4/12.
      DUM(3,3) =  -SNLZ*TLSNO*GZ4/12.+SNLY*TLSNO*GY4/12.
      DUM(3,5) =    SNLZ*TLSNO*SIN(THGY4)/12.
      DUM(3,6) =    SNLY*TLSNO*SIN(THGZ4)/12.
      DUM(3,7) =  (-SNLY*GZ4-SNLZ*GY4)/12.
      DUM(4,1) =    GXY(GY4,THGX4,ALL)
      DUM(4,2) =    GXSY(-SNLY,THGX4,-SNLX,GY4,THGX4,ALL)
      DUM(4,3) =    GXPHI(SNLZ,GY4,THGX4,-SNLY,GZ4,THGX4,ALL)
      DUM(4,4) =  -1.
      DUM(5,1) =    GYY(THGY4,ALL)
      DUM(5,2) =    GYSY(-SNLY,GX4,THGY4,-SNLX,THGY4,ALL)
      DUM(5,3) =    GYPHI(SNLZ,THGY4,-SNLY,GZ4,THGY4,ALL)
      DUM(5,5) =  -1.
      DUM(6,1) =    GZY(GY4,THGZ4,ALL)
      DUM(6,2) =    GZSY(-SNLY,GX4,THGZ4,-SNLX,GY4,THGZ4,ALL)
      DUM(6,3) =    GZPHI(SNLZ,GY4,THGZ4,-SNLY,THGZ4,ALL)
      DUM(6,6) =  -1.
      IF(KODE(10).EQ.2) GO TO 1040
      CALL DRCSN(THETA)
      ALL1=ALL+1.
      CALL STINT(Q,ALL1,C,1,1,TLSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALL2=ALL-1.
      CALL STINT(Q,ALL2,C,1,1,TLSN2,NG)
      IF(NG.NE.0) GO TO 5000
       AKTL=(TLSN1-TLSN2)/2.
      AKSNL=AKTL
 1040 CONTINUE
      DUM(7,7) =  -1.
      DUM(7,8) =    AKSNL*12.
      DUM(8,1) =    ALY(GY4)
      DUM(8,2) =    ALSY(-SNLY,GX4,-SNLX,GY4)
      DUM(8,3) =    ALPHI(SNLZ,GY4,-SNLY,GZ4)
      DUM(8,9) =  -1.
      IF(KODE(10).EQ.1) GO TO 1045
      DO 1046 I=1,3
      DO 1046 J=1,3
 1046  SNUD(I,J)=SNUD(I,J)+DUM(I,7)*ADSNL*DUM(8,J)*12.
```

```
 1045 CALL MASH(3,8)                                                    CAB02590
      DO 1080 I=1,3                                                     CAB02600
      DO 1080 J=1,3                                                     CAB02610
 1080 BOTR(I,J)= DUM(I,J)                                               CAB02620
      DO 1090 I=1,3                                                     CAB02630
      DO 1090 J=1,3                                                     CAB02640
 1090 SNU (I,J)= TOPR(I,J)+TOPL(I,J)+BOTL(I,J)+BOTR(I,J)                CAB02650
      IF(KODE(10).EQ.2) RETURN                                          CAB02660
      DO 1095 I=1,3                                                     CAB02670
      DO 1095 J=1,3                                                     CAB02680
 1095 SNUD(I,J)=0                                                       CAB02690
      RETURN                                                            CAB02700
 1002 DO 1004 I=1,3                                                     CAB02710
      DO 1004 J=1,3                                                     CAB02720
      SNUD(I,J)=0                                                       CAB02730
 1004 SNU(I,J)=0                                                        CAB02740
      RETURN                                                            CAB02750
      END                                                               CAB02760
      SUBROUTINE TRIM                                                   CAB00010
C      CABLE  SUSPENSION  SYSTEM  TRIM  ROUTINE                         CAB00020
      COMMON/INPUT/IW,IR                                                CAB00030
      COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL                      CAB00040
      COMMON / PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF         CAB00050
      DIMENSION ANG(5,3)                                                CAB00060
      EQUIVALENCE(AERO( 1),  CDU),(AERO( 2),  CLU),(AERO( 3),  CMU),    CAB00070
     1           (AERO( 4),  CDA),(AERO( 5),  CLA),(AERO( 6),  CMA),    CAB00080
     2           (AERO( 7),  CDQ),(AERO( 8),  CLQ),(AERO( 9),  CMQ),    CAB00090
     3           (AERO(10),  CDD),(AERO(11),  CLD),(AERO(12),  CMD),    CAB00100
     4           (AERO(13), CDDE),(AERO(14), CLDE),(AERO(15), CMDE),    CAB00110
     5           (AERO(16), CDAD),(AERO(17),CLAD),(AERO(18),CMAD),      CAB00120
     6           (AERO(19),  CYB),(AERO(20),  CLB),(AERO(21),  CNB),    CAB00130
     7           (AERO(22),  CYP),(AERO(23),  CLP),(AERO(24),  CNP),    CAB00140
     8           (AERO(25),  CYR),(AERO(26),  CLR),(AERO(27),  CNR),    CAB00150
     9           (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR),    CAB00160
     A           (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA),    CAB00170
     B           (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS)     CAB00180
      EQUIVALENCE(AERO(46),XCG),(AERO(47),ZCG)                          CAB00190
      EQUIVALENCE(AERO (48),AMACH),(AERO (49),VO   ),(AERO (50),   AM)  CAB00200
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),   WT),(AERO (53),B    )  CAB00210
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)  CAB00220
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (59),ZIZZ )  CAB00230
      EQUIVALENCE(AERO (60),CLT  ),(AERO (61),CDT  ),(AERO (62),CMT  ), CAB00240
     1(AERO(63),THETA)                                                  CAB00250
      EQUIVALENCE(AERO (66), WLUF),(AERO(67), WLLF),(AERO(68), WLUR),   CAB00260
     1           (AERO (69), WLLR),(AERO(70), WLHF),(AERO(71), WLHR),   CAB00270
     2           (AERO (72), STAF),(AERO(73), STAR),(AERO(74), BLHF),   CAB00280
     3           (AERO (75), BLHR),(AERO(76), WLCR),(AERO(77),STACR),   CAB00290
     4           (AERO (78), BLCF),(AERO(79),   EF),(AERO(80),   ER),   CAB00300
     5           (AERO (81),   AF),(AERO(82),   AR),(AERO(83), HUCF),   CAB00310
     6           (AERO (84), HLCF),(AERO(85), HUCR),(AERO(86), HLCR),   CAB00320
     7           (AERO (87),  DCF),(AERO(88),  DCR),(AERO(89),  ALF),   CAB00330
     8           (AERO (90),  RVF),(AERO(91),  RHF),(AERO(92),  RVR),   CAB00340
     9           (AERO(93),  RHR),(AERO(94),  TRO),(AERO(95),  AKR),    CAB00350
     A           (AERO(96), ALRO),(AERO(97),STLTT),(AERO(98),WLLTT),    CAB00360
     B           (AERO(99),TLFTO),(AERO(100),AKLFT),(AERO(101),ALLTO),  CAB00370
```

```
C               (AERO(102),ALTX),(AERO(103),ALTZ)                         CAB00380
      EQUIVALENCE(AEROP( 1), CXUP),(AEROP( 2), CZUP),(AEROP( 3), CMUP),  CAB00390
     1          (AEROP( 4), CXAP),(AEROP( 5), CZAP),(AEROP( 6), CMAP),   CAB00400
     2          (AEROP( 7), CXQP),(AEROP( 8), CZQP),(AEROP( 9), CMQP),   CAB00410
     3          (AEROP(10), CXDP),(AEROP(11), CZDP),(AEROP(12), CMDP),   CAB00420
     4          (AEROP(13),CXDEP),(AEROP(14),CZDEP),(AEROP(15),CMDEP),   CAB00430
     5          (AEROP(16),CXADP),(AEROP(17),CZADP),(AEROP(18),CMADP),   CAB00440
     6          (AEROP(19), CYBP),(AEROP(20), CLBP),(AEROP(21), CNBP),   CAB00450
     7          (AEROP(22), CYPP),(AEROP(23), CLPP),(AEROP(24), CNPP),   CAB00460
     8          (AEROP(25), CYRP),(AEROP(26), CLRP),(AEROP(27), CNRP),   CAB00470
     9          (AEROP(28),CYDRP),(AEROP(29),CLDRP),(AEROP(30),CNDRP),   CAB00480
     A          (AEROP(31),CYDAP),(AEROP(32),CLDAP),(AEROP(33),CNDAP),   CAB00490
     B          (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP)    CAB00500
      RTD=57.2958                                                        CAB00510
      THETA=0.                                                           CAB00520
      DELALF=.001                                                        CAB00530
      DTF=.1                                                             CAB00540
      DALFAW=0.0                                                         CAB00550
      DDELTE=0.0                                                         CAB00560
      DTHRST=0.0                                                         CAB00570
      ICNTR=0                                                            CAB00580
      FIRST=0.                                                           CAB00590
      THINT=0.                                                           CAB00600
      ALFINT=THETA                                                       CAB00610
      DELINT=0.                                                          CAB00620
      THRSTO=THINT                                                       CAB00630
    1 IF(V0.EQ.0.)THRSTO=-TR*(COS(ADC(3,1))+COS(ADC(4,1)))/(COS(ADC(1,1)CAB00640
     1)+COS(ADC(2,1)))                                                   CAB00650
      VAL5=COS(ADC(3,1))                                                 CAB00660
      VAL6=COS(ADC(4,1))                                                 CAB00670
      VAL7=COS(ADC(1,1))                                                 CAB00680
      VAL8=COS(ADC(2,1))                                                 CAB00690
      ALFAWO=ALFINT                                                      CAB00700
      DELTEO=DELINT                                                      CAB00710
      QS=RHO*V0*V0*.5*SW                                                 CAB00720
  209 THRSTI=THRSTO+DTHRST                                               CAB00730
      ALFAWI=ALFAWO+DALFAW                                               CAB00740
      DELTEI=DELTEO+DDELTE                                               CAB00750
      ICNTR=ICNTR+1                                                      CAB00760
      IF(ICNTR.GT.100)GO TO 520                                          CAB00770
      VAL1=ALFAWI*RTD                                                    CAB00780
      VAL2=DELTEI*RTD                                                    CAB00790
      VAL3=THRSTI                                                        CAB00800
      CALL EQU(ALFAWI,DELTEI,THRSTI,F0,G0,H0,FIRST)                      CAB00810
      IF (V0.NE.0..OR.FIRST.NE.0.) GO TO 2                               CAB00820
      FIRST=1.                                                           CAB00830
      GO TO 1                                                            CAB00840
    2 IF(FIRST.NE.1.)FIRST=1.                                            CAB00850
C  COMPUTES PARTIALS                                                     CAB00860
      ALFAWI=ALFAWI+DELALF*0.5                                           CAB00870
      CALL EQU(ALFAWI,DELTEI,THRSTI,F1,G1,H1,1.)                         CAB00880
      ALFAWI=ALFAWI-DELALF                                               CAB00890
      CALL EQU(ALFAWI,DELTEI,THRSTI,F2,G2,H2,1.)                         CAB00900
      ALFAWI=ALFAWI+DELALF*0.5                                           CAB00910
      FALFWO=(F1-F2)/DELALF                                              CAB00920
```

```
      GALFWO=(G1-G2)/DELALF                                             CABC0930
      HALFWO=(H1-H2)/DELALF                                             CABC0940
      FDELEO=-QS*(CLDE*COS(ALFAWI)+CDDE*SIN(ALFAWI))                    CABC0950
      GDELEO=QS*(CLDE*SIN(ALFAWI)-CDDE*COS(ALFAWI))                     CABC0960
      HDELEO=QS*CBAR*CMDE                                               CABC0970
      THRSTI=THRSTI+DTF                                                 CABC0980
      CALL EQU(ALFAWI,DELTEI,THRSTI,F1,G1,H1,1.)                        CABC0990
      THRSTI=THRSTI-2.*DTF                                              CABC1000
      CALL EQU(ALFAWI,DELTEI,THRSTI,F2,G2,H2,1.)                        CABC1010
      THRSTI=THRSTI+DTF                                                 CABC1020
      FTHSTO=(F1-F2)/(DTF*2.)                                           CABC1030
      GTHSTO=(G1-G2)/(DTF*2.)                                           CABC1040
      HTHSTO=(H1-H2)/(DTF*2.)                                           CABC1050
C     SET UP ITERATION EQUATIONS                                        CABC1060
      FI=FO+FALFWO*DALFAW+FDELEO*DDELTE+FTHSTO*DTHRST                   CABC1070
      GI=GO+GALFWO*DALFAW+GDELEO*DDELTE+GTHSTO*DTHRST                   CABC1080
      HI=HO+HALFWO*DALFAW+HDELEO*DDELTE+HTHSTO*DTHRST                   CABC1090
      ACCZ=FI/AM                                                        CABC1100
      ACCX=GI/AM                                                        CABC1110
      THEDOT=HI/YIYY                                                    CABC1120
      IF(VO.EQ.0.)GO TO 42                                              CABC1130
      IF(ABS(ACCZ).LT..01)GO TO 1005                                    CABC1140
      GO TO 1100                                                        CABC1150
 1005 IF(ABS(ACCX).LT..01)GO TO 1007                                    CABC1160
      GO TO 1100                                                        CABC1170
 1007 IF(ABS(THEDOT).LE.0.001)GO TO 42                                  CABC1180
C     NOW COMPUTE PARAMETER INCREMENTS FROM MATRIX EQUATIONS            CABC1190
 1100 DETRM=FALFWO*GDELEO*HTHSTO+FDELEO*GTHSTO*HALFWO+FTHSTO*GALFWO*    CABC1200
     1HDELEO-FTHSTO*GDELEO*HALFWO-FALFWO*GTHSTO*HDELEO-FDELEO*GALFWO*   CABC1210
     2HTHSTO                                                            CABC1220
      DALFAW=(-(GDELEO*HTHSTO-GTHSTO*HDELEO)*FO+(FDELEO*HTHSTO-FTHSTO   CABC1230
     1*HDELEO)*GO-(FDELEO*GTHSTO-FTHSTO*GDELEO)*HO)/DETRM               CABC1240
      DDELTE=(+(GALFWO*HTHSTO-GTHSTO*HALFWO)*FO-(FALFWO*HTHSTO-HALFWO   CABC1250
     1*FTHSTO)*GO+(FALFWO*GTHSTO-FTHSTO*GALFWO)*HO)/DETRM               CABC1260
      DTHRST=(-(GALFWO*HDELEO-GDELEO*HALFWO)*FO+(FALFWO*HDELEO-FDELEO   CABC1270
     1*HALFWO)*GO-(FALFWO*GDELEO-FDELEO*GALFWO)*HO)/DETRM               CABC1280
      THRSTO=THRSTI                                                     CABC1290
      ALFAWO=ALFAWI                                                     CABC1300
      DELTEO=DELTEI                                                     CABC1310
      GO TO 200                                                         CABC1320
  520 WRITE(IW,521)                                                     CABC1330
  521 FORMAT('  TRIM ITERATION EXCEEDS LIMITS')                         CABC1340
      GO TO 522                                                         CABC1350
   42 CALL EQU(ALFAWI,DELTEI,THRSTI,FO,GO,HO,1.)                        CABC1360
  522 DO 523 IZZ=1,4                                                    CABC1370
      DO 523 IZK=1,3                                                    CABC1380
      ANG(IZZ,IZK)=ADC(IZZ,IZK)*RTD                                     CABC1390
  523 CONTINUE                                                          CABC1400
      THETA=ALFAWI                                                      CABC1410
      DE=DELTEI                                                         CABC1420
      TF=THRSTI                                                         CABC1430
      THETD=THETA*RTD                                                   CABC1440
      DED=DE*RTD                                                        CABC1450
      DO 524 IZZ=1,4                                                    CABC1460
      IF(KODE(5).EQ.0) GO TO 528                                        CABC1470
```

```
      WRITE(IW,525)IZZ,XLGTH(IZZ),(ANG(IZZ,IZK),ARM(IZZ,IZK),IZK=1,3)
  525 FORMAT(' CABLE GEOMETRY-CABLE NO.',I2,5X,'CABLE LENGTH=',E15.6,
     1' IN',/,3X,'  DIR. COS.=DEG   ARM-IN',/,(3(3X,2E15.6,/)),//)
  524 CONTINUE
      IF(VO.EQ.0.)WRITE(IW,529)
  529 FORMAT(' COMPUTATION OF WIND OFF CONDITION,TRIM ROUTINE NOT USED')
      WRITE(IW,526)ICNTR,ACCZ,ACCX,THEDOT
  526 FORMAT(' ITERATION PARAMETER =',I5,/,2X,'ACCZ  =',E15.8,
     1/,2X,'ACCX =',E15.8,/,2X,'THEDOT=',E15.8,' RAD/SEC')
  528 WRITE(IW,527)THETD,DED,TF,TR
  527 FORMAT(//,'VEH. ATT.,DEFLTN,& CABLE TENSION',/,
     12X,'THETA =',F6.2,' DEG',/,2X,'DELTA =',F6.2,' DEG',/,2X
     2,'FRT CAB. TENSION=',E15.6,' LBS',/,
     32X,'RR CAB. TENSION =',E15.6,' LBS')
      RETURN
C     DEBUG UNIT(3),INIT(VAL1,VAL2,VAL3,FI,GI,HI,
C    1FALFWD,GALFWD,HALFWD,FDELFD,GDELFD,HDELFD,
C    2FTHSTD,GTHSTD,HTHSTD,DALFAW,DDELTE,DTHRST,
C    3ACCZ,ACCX,THEDOT,TF,VAL5,VAL6,VAL7,VAL8)
      END
      SUBROUTINE EQU(THETA,DE,TF,FF,GG,HH,FIRST)
C        CABLE  SUSPENSION  SYSTEM  TRIM  EQUATIONS
      COMMON/INPUT/IW,IR
      COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL
      COMMON / FLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,DUMMY
      REAL*8 XNM1,XNM2,YNM1,YNM2
      EQUIVALENCE(AERO( 1),  CDU),(AERO( 2),  CLU),(AERO( 3),  CMU),
     1           (AERO( 4),  CDA),(AERO( 5),  CLA),(AERO( 6),  CMA),
     2           (AERO( 7),  CDQ),(AERO( 8),  CLQ),(AERO( 9),  CMQ),
     3           (AERO(10),  CDO),(AERO(11),  CLO),(AERO(12),  CMO),
     4           (AERO(13), CDDE),(AERO(14), CLDE),(AERO(15), CMDE),
     5           (AERO(16), CDAD),(AERO(17),CLAD),(AERO(18),CMAD),
     6           (AERO(19),  CYB),(AERO(20),  CLB),(AERO(21),  CNB),
     7           (AERO(22),  CYP),(AERO(23),  CLP),(AERO(24),  CNP),
     8           (AERO(25),  CYR),(AERO(26),  CLR),(AERO(27),  CNR),
     9           (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR),
     A           (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA),
     B           (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS)
      EQUIVALENCE(AERO(46),XCG),(AERO(47),ZCG)
      EQUIVALENCE(AERO (48),AMACH),(AERO (49),VO   ),(AERO (50),   AM)
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),   WT),(AERO (53),B    )
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (59),ZIZZ )
      EQUIVALENCE(AERO (60),CLT  ),(AERO (61),CDT  ),(AERO (62),CMT  )
      EQUIVALENCE(AERO (66), WLUF),(AERO(67), WLLF),(AERO(68), WLUR),
     1           (AERO (69), WLLR),(AERO(70), WLHF),(AERO(71), WLHR),
     2           (AERO (72), STAF),(AERO(73), STAR),(AERO(74), BLHF),
     3           (AERO (75), BLHR),(AERO(76), WLCR),(AERO(77),STACR),
     4           (AERO (78), BLCR),(AERO(79),   EF),(AERO(80),   ER),
     5           (AERO (81),   AF),(AERO(82),   AR),(AERO(83), HUCF),
     6           (AERO (84), HLCF),(AERO(85), HUCR),(AERO(86), HLCR),
     7           (AERO (87),  DCF),(AERO(88),  DCR),(AERO(89),  ALF),
     8           (AERO (90),  RVF),(AERO(91),  RHF),(AERO(92),  RVR),
     9           (AERO(93),   RHR),(AERO(94),  TRO),(AERO(95),  AKR),
     A           (AERO(96), ALRC),(AERO(97),STLTT),(AERO(98),WLLTT),
```

```
     B             (AERO(99),TLFT0),(AERO(100),AKLFT),(AERO(101),ALLT0),
     C             (AERO(102),ALTX),(AERO(103),ALTZ)
      DATA XNM1,XNM2 /'VERTICAL','HORIZNTL'/
      RTD=57.2958
      VAL1=THETA
      Q = RHO*VO*VO/2.0
   64 IND=KODE(6)
      GO TO (501,502,503,504),IND
  501 YNM1=XNM1
      YNM2=XNM2
      CALL FPLYV(STAF,WLUF,WLLF,HUCF,HLCF,EF,RVF,THETA,1)
      CALL RPLYH(STAR,BLHR,WLHR,-AR,DCR,0.,RHR,THETA,3)
      GO TO 505
  502 YNM1=XNM2
      YNM2=XNM1
      CALL FPLYH(STAF,BLHF,WLHF,AF,DCF,0.,RHF,THETA,1)
      CALL FPLYV(STAR,WLUR,WLLR,HUCR,HLCR,ER,RVR,THETA,3)
      GO TO 505
  503 YNM1=XNM1
      YNM2=XNM1
      CALL FPLYV(STAF,WLUF,WLLF,HUCF,HLCF,EF,RVF,THETA,1)
      CALL FPLYV(STAR,WLUR,WLLR,HUCR,HLCR,ER,RVR,THETA,3)
      GO TO 505
  504 YNM1=XNM2
      YNM2=XNM2
      CALL FPLYH(STAF,BLHF,WLHF,AF,DCF,0.,RHF,THETA,1)
      CALL RPLYH(STAR,BLHR,WLHR,-AR,DCR,0.,RHR,THETA,3)
  505 IF(KODE(11))506,507,506
  506 WLLT = WLCR + ALTX*SIN(THETA) - ALTZ*COS(THETA)
      STALT = STACR - ALTX*COS(THETA) - ALTZ*SIN(THETA)
      XLGTH(5) = SQRT((WLLTT - WLLT)**2 + (STLTT - STALT)**2)
      IF(FIRST.NE.0.)GO TO 12
      ELLO=XLGTH(5)
   12 ELL=XLGTH(5)
      TLFT = TLFT0+AKLFT*(ELL-ELLO)
      ARM(5,1)=ALTX
      ARM(5,2)=0
      ARM(5,3)=ALTZ
      FXLTT = (TLFT*(STALT - STLTT))/XLGTH(5)
      FZLTT = (TLFT*(WLLT - WLLTT))/XLGTH(5)
      FXLTB = FXLTT*COS(THETA) - FZLTT*SIN(THETA)
      FZLTB = FZLTT*COS(THETA) + FXLTT*SIN(THETA)
      YMLFT =( FXLTB*ALTZ - FZLTB*ALTX)/12.
      ADC(5,1)=ARCOS(FXLTB/TLFT)
      ADC(5,2)=3.14159/2.
      ADC(5,3)=ARCOS(FZLTB/TLFT)
      GO TO 508
  507 FXLTB=0.
      FZLTB=0.
      YMLFT=0.
      XLGTH(5)=0.
      TLFT=0.
      DO 13 IA=1,3
      ARM(5,IA)=0.
      ADC(5,IA)=0.
```

```
   13 CONTINUE                                                          CAB00910
  508 CALL SNTRM(FXSN,FZSN,EMSN,THETA)                                  CAB00920
      IF (FIRST.NE.0.)GO TO 510                                         CAB00930
      IF(KODE(5).EQ.0) GO TO 512                                        CAB00940
      WRITE(IW,509)YNM1,YNM2                                            CAB00950
  509 FORMAT(' CABLE CONFIGURATION ON MODEL',/,                         CAB00960
     1' FRONT CABLE IS ',A8,' AND REAR CABLE IS ',A8)                   CAB00970
  512 ELO=XLGTH(3)+XLGTH(4)                                             CAB00980
  510 EL=XLGTH(3)+XLGTH(4)                                              CAB00990
      TR=TRO+AKR*(EL-ELO)                                               CAB01000
      ELIFT=Q*SW*(CLO+CLA*THETA+CLDE*DE)                                CAB01010
      ADRAG=Q*SW*(CDO+CDA*THETA+CDDE*DE)                                CAB01020
      FXAIR=-ADRAG*COS(THETA)+ELIFT*SIN(THETA)                          CAB01030
      FZAIR=-ADRAG*SIN(THETA)-ELIFT*COS(THETA)                          CAB01040
      WGTX=-32.2*AM*SIN(THETA)                                          CAB01050
      WGTZ=32.2*AM*COS(THETA)                                           CAB01060
      EMWGT=(ZCG*WGTX-XCG*WGTZ)/12.                                     CAB01070
      FXCR=TR*(COS(ADC(3,1))+COS(ADC(4,1)))                             CAB01080
      FZCR=TR*(COS(ADC(3,3))+COS(ADC(4,3)))                             CAB01090
      FXCFH=TF*(COS(ADC(1,1))+COS(ADC(2,1)))                            CAB01100
      FZCFH=TF*(COS(ADC(1,3))+COS(ADC(2,3)))                            CAB01110
      EMOC=0.                                                           CAB01120
      DO 511 I=1,4                                                      CAB01130
      TENS=TF                                                           CAB01140
      IF(I.GT.2)TENS=TR                                                 CAB01150
      EMOC=EMOC+TENS*(COS(ADC(I,1))*ARM(I,3)-COS(ADC(I,3))*ARM(I,1))   CAB01160
  511 CONTINUE                                                          CAB01170
      EMOC=EMOC/12.                                                     CAB01180
      AEROM=Q*SW*CBAR*(CMO+CMA*THETA+CMDE*DE)                           CAB01190
      FF=FZCFH+FZCR+FZLTR+FZSN+WGTZ+FZAIR                               CAB01200
      GG=FXCFH+FXCR+FXLTR+FXSN+WGTX+FXAIR                               CAB01210
      HH=EMOC+YMLFT+EMSN+EMWGT+AEROM                                    CAB01220
      RETURN                                                            CAB01230
      END                                                               CAB01240
      SUBROUTINE FPLYV(STAV,WLU,WLL,HHU,HHL,EP,RAD,THETA,IF)            CAB00010
      COMMON /DAT/AERO(175),AEROP(50),KODE(26),LL                       CAB00020
      COMMON /PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF          CAB00030
      EQUIVALENCE (AERO(76),WLCR),(AERO(77),STACR),(AERO(78),BLCR)      CAB00040
      PI=3.14159                                                        CAB00050
   33 GAMU= ATAN(HHU/EP)                                                CAB00060
      T1= EP*EP +HHU*HHU                                                CAB00070
      T2= THETA +GAMU                                                   CAB00080
      IF(IF.EQ.3) T2=GAMU-THETA                                         CAB00090
      WLUC= WLCR +SQRT(T1)*SIN(T2)                                      CAB00100
      T3= WLU -WLUC                                                     CAB00110
      T4= ABS(STACR -STAV) -SQRT(T1)*COS(T2)                            CAB00120
      XLUP= SQRT(T3*T3+T4*T4)                                           CAB00130
      XLU= SQRT(XLUP*XLUP -RAD*RAD)                                     CAB00140
      BUP= ATAN(T3/T4)                                                  CAB00150
      DBU= ATAN(RAD/XLU)                                                CAB00160
      BETAU=(BUP -DBU)*RTD                                              CAB00170
      GAML= ATAN(HHL/EP)                                                CAB00180
      T5= EP*EP +HHL*HHL                                                CAB00190
      T6= THETA -GAML                                                   CAB00200
      IF(IF.EQ.3) T6=-(THETA+GAML)                                      CAB00210
```

```
      WLLC= WLCR +SQRT(T5)*SIN(T6)                                      CAB00220
      T7= WLLC -WLL                                                     CAB00230
      T8= ABS(STACR -STAV) -SQRT(T5)*COS(T6)                            CAB00240
      XLLP= SQRT(T7*T7 +T8*T8)                                          CAB00250
      XLL= SQRT(XLLP*XLLP -RAD*RAD)                                     CAB00260
      BLP= ATAN(T7/T8)                                                  CAB00270
      DBL= ATAN(RAD/XLL)                                                CAB00280
      BETAL= (BLP -DBL)*RTD                                             CAB00290
      IF(IF.EQ.1)GO TO 1                                                CAB00300
      XLGTH(3)=XLU                                                      CAB00310
      XLGTH(4)=XLL                                                      CAB00320
      ADC(3,1)=BETAU/RTD-THETA+PI                                       CAB00330
      ADC(3,2)=-PI/2.                                                   CAB00340
      ADC(3,3)=PI/2.-ADC(3,1)                                           CAB00350
      ADC(4,1)=PI-(BETAL/RTD-THETA)                                     CAB00360
      ADC(4,2)=-PI/2                                                    CAB00370
      ADC(4,3)=PI/2-ADC(4,1)                                            CAB00380
      ARM(3,1)=-EP+RAD*SIN(ADC(3,1))                                    CAB00390
      ARM(3,2)=0.                                                       CAB00400
      ARM(3,3)=-HHU+RAD*COS(ADC(3,1))                                   CAB00410
      ARM(4,1)=-EP-RAD*SIN(ADC(4,1))                                    CAB00420
      ARM(4,2)=0.                                                       CAB00430
      ARM(4,3)=HHL-RAD*COS(ADC(4,1))                                    CAB00440
      RETURN                                                            CAB00450
    1 XLGTH(1)=XLU                                                      CAB00460
      XLGTH(2)=XLL                                                      CAB00470
      ADC(1,1)=-BETAU/RTD+THETA                                         CAB00480
      ADC(1,2)=PI/2.                                                    CAB00490
      ADC(1,3)=PI/2.-ADC(1,1)                                           CAB00500
      ADC(2,1)=BETAL/RTD+THETA                                          CAB00510
      ADC(2,2)=PI/2.                                                    CAB00520
      ADC(2,3)=PI/2.-ADC(2,1)                                           CAB00530
      ARM(1,1)=EP+RAD*SIN(ADC(1,1))                                     CAB00540
      ARM(1,2)=0.                                                       CAB00550
      ARM(1,3)=-HHU-RAD*COS(ADC(1,1))                                   CAB00560
      ARM(2,1) EP-RAD*SIN(ADC(2,1))                                     CAB00570
      ARM(2,2)=0.                                                       CAB00580
      ARM(2,3)=HHL+RAD*COS(ADC(2,1))                                    CAB00590
      RETURN                                                            CAB00600
      END                                                               CAB00610
      SUBROUTINE RPLYH(STAD,BLD,WLD,XP,YP,ZP,RAD,THETA,IF)              CAB00620
      COMMON /DAT/AERO(175),AEROP(50),KODE(26),LL                       CAB00630
      COMMON /PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF          CAB00640
      EQUIVALENCE(AERO(76),WLCR),(AERO(77),STACR),(AERO(78),BLCR)       CAB00650
      PI=3.14159                                                        CAB00660
      XWT=STACR-STAD                                                    CAB00670
      ZWT=WLCR-WLD                                                      CAB00680
      XB=XWT*COS(THETA)-ZWT*SIN(THETA)                                  CAB00690
      ZB=XWT*SIN(THETA)+ZWT*COS(THETA)                                  CAB00700
      T9= BLD -YP                                                       CAB00710
      T10=XB-XP                                                         CAB00720
      XLHIP= SQRT(T9*T9 +T10*T10)                                       CAB00730
      BHIP= ATAN2(T9,T10)                                               CAB00740
      XLHI= SQRT(XLHIP*XLHIP -RAD*RAD)                                  CAB00750
      DBHI= ATAN(RAD/XLHI)                                              CAB00760
```

```
      BHI= BHIP -DBHI
      T11=ZB-ZP
      XL=SQRT(XLHI*XLHI+T11*T11)
      TH10=T10-RAD*COS(BHI)
      TH9=T9-RAD*SIN(BHI)
      IF(IF.EQ.3)GO TO 3
      XLGTH(1)=XL
      XLGTH(2)=XL
      ADC(1,1)=ARCOS(TH10/XL)
      ADC(1,2)=ARCOS(TH9/XL)
      ADC(1,3)=ARCOS(T11/XL)
      ADC(2,1)=-ADC(1,1)
      ADC(2,2)=PI-ADC(1,2)
      ADC(2,3)=ADC(1,3)
      ARM(1,1)=XP-RAD*SIN(BHI)
      ARM(1,2)=YP+RAD*COS(BHI)
      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(TH10/XL)
      ADC(3,2)=ARCOS(TH9/XL)
      ADC(3,3)=ARCOS(T11/XL)
      ADC(4,1)=-ADC(3,1)
      ADC(4,2)=PI-ADC(3,2)
      ADC(4,3)=ADC(3,3)
      ARM(3,1)=XP+RAD*SIN(BHI)
      ARM(3,2)=YP-RAD*COS(BHI)
      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   COMPUTES D/L-LGTH EQ FOR X-Z-THETA OR Y-PSI-PHI COEFF
      COMMON/PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
      IF(IDX.NE.0)GO TO 1
      C1=-COS(ADC(IC,1))
      C2=-COS(ADC(IC,3))
      C3=(ARM(IC,1)*COS(ADC(IC,3))-ARM(IC,3)*COS(ADC(IC,1)))/12.
      RETURN
    1 C1=-COS(ADC(IC,2))
      C2=(ARM(IC,2)*COS(ADC(IC,1))-ARM(IC,1)*COS(ADC(IC,2)))/12.
      C3=(ARM(IC,3)*COS(ADC(IC,2))-ARM(IC,2)*COS(ADC(IC,3)))/12.
      RETURN
      END
      SUBROUTINE DCOSLG(IC,CX1,CZ1,CT1,CX3,CZ3,CT3)
C   COMPUTES D-DIR COS EQS FOR X-Z-THETA COEFF.
      COMMON/PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
      CX1=SIN(ADC(IC,1))/XLGTH(IC)*12.
      IF(ABS(ADC(IC,3)-3.14159).GT..00001) GO TO 2
```

```
      XVAL=1000.                                                        CABC1320
      GO TO 1                                                           CABC1330
    2 XVAL=COTAN(ADC(IC,3))                                             CABC1340
    1 CZ1=-COS(ADC(IC,3))*COTAN(ADC(IC,1))/XLGTH(IC)*12.                CABC1350
      XWT=ARM(IC,1)                                                     CABC1360
      ZWT=ARM(IC,3)                                                     CABC1370
      CT1=(ZWT*SIN(ADC(IC,1))+XWT*COS(ADC(IC,3))*COTAN(ADC(IC,1)))/     CABC1380
     1XLGTH(IC)                                                         CABC1390
      CX3=-COS(ADC(IC,1))*XVAL/XLGTH(IC)*12.                            CABC1400
      CZ3=SIN(ADC(IC,3))/XLGTH(IC)*12.                                  CABC1410
      CT3=-(ZWT*COS(ADC(IC,1))*XVAL+XWT*SIN(ADC(IC,3)))                 CABC1420
     1/XLGTH(IC)                                                        CABC1430
      RETURN                                                            CABC1440
      END                                                               CABC1450
C  THIS IS A DOUBLE PRECISION VERSION OF CABLE4 TO BE USED              CABC0010
C  WITH THE LRC MATRIX REDUCTION AND IBM ROOT                           CABC0020
C  FINDING ROUTINE                                                      CABC0030
      SUBROUTINE LONG                                                   CABC0040
      COMMON/INPUT/IW,IR                                                CABC0050
      COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL                      CABC0060
      COMMON / PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF         CABC0070
      COMMON /DUM/DUM(10,10)                                            CABC0080
      COMMON/FRQ/C4(30)                                                 CABC0090
      EQUIVALENCE(AERO(46),XCG),(AERO(47),ZCG)                          CABC0100
      EQUIVALENCE(AERO (63),THETA),(AERO (49),VO   ),(AERO (50),    AM) CABC0110
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),     WT),(AERO (53),B    )CABC0120
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)  CABC0130
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (56),ZIZZ ), CABC0140
     1          (AERO(95),AKR),(AERO(100),AKLFT)                        CABC0150
      EQUIVALENCE(AERO(117),TUSNO),(AERO(119),AKSNU),(AERO(120),AKSNL)  CABC0160
      EQUIVALENCE(AERO(123), AKSY),(AERO(124),AKPHI),(AERO(125),AKTHE), CABC0170
     1          (AERO(126), AKAZ),(AERO(127), T1SY),(AERO(128),T2PHI),  CABC0180
     2          (AERO(129),T3THE),(AERO(130), T4AZ)                     CABC0190
      EQUIVALENCE(AERO(131),AKSBT),(AERO(132),AKSBV),(AERO(133),AJASM), CABC0200
     1          (AERO(134),RSBA) ,(AERO(135),ELSBA),(AERO(136), RSBD),  CABC0210
     2          (AERO(137),AKTHD),(AERO(138), AKTH),(AERO(139), GDMP),  CABC0220
     3          (AERO(140), AKO )                                       CABC0230
      EQUIVALENCE(AEROP( 1), CXUP),(AEROP( 2), CZUP),(AEROP( 3), CMUP), CABC0240
     1          (AEROP( 4), CXAP),(AEROP( 5), CZAP),(AEROP( 6), CMAP),  CABC0250
     2          (AEROP( 7), CXQP),(AEROP( 8), CZQP),(AEROP( 9), CMQP),  CABC0260
     3          (AEROP(10), CXOP),(AEROP(11), CZOP),(AEROP(12), CMOP),  CABC0270
     4          (AEROP(13),CXDEP),(AEROP(14),CZDEP),(AEROP(15),CMDEP),  CABC0280
     5          (AEROP(16),CXADP),(AEROP(17),CZADP),(AEROP(18),CMADP),  CABC0290
     6          (AEROP(19), CYBP),(AEROP(20), CLBP),(AEROP(21), CNBP),  CABC0300
     7          (AEROP(22), CYPP),(AEROP(23), CLPP),(AEROP(24), CNPP),  CABC0310
     8          (AEROP(25), CYRP),(AEROP(26), CLRP),(AEROP(27), CNRP),  CABC0320
     9          (AEROP(28),CYDRP),(AEROP(29),CLDRP),(AEROP(30),CNDRP),  CABC0330
     A          (AEROP(31),CYDAP),(AEROP(32),CLDAP),(AEROP(33),CNDAP),  CABC0340
     B          (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP)   CABC0350
      DIMENSION CMAT(14,14,3),RMAT(14,3)                                CABC0360
      COMPLEX ROOTS(44)                                                 CABC0370
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)                CABC0380
      COMMON /ROUGH/FRIC(3,6)                                           CABC0390
      DIMENSION FXS(3,4)                                                CABC0400
      DO 10 J=1,3                                                       CABC0410
```

```
      DO 10 K=1,4
   10 FXS(J,K)=0.
      DO 1 IC=1,5
      DO 3 J=1,10
      DO 3 K=1,10
    3 DUM(J,K)=0.
      IF(KODE(10).EQ.3)GO TO 649
      TENS=TF
      IF(IC.GT.2) TENS=TR
      IF(IC.GT.4) TENS=TLFT
      DUM(1,2)= - TENS * COS(ADC(IC,3))
      DUM(1,5)= - TENS * SIN (ADC(IC,1))
      DUM(2,2)= TENS * COS(ADC(IC,1))
      DUM(2,6)= - TENS * SIN(ADC(IC,3))
      DUM(3,2)=( ARM(IC,3)*DUM(1,2)-ARM(IC,1)*DUM(2,2))/12.
      DUM(3,5)= ARM(IC,3)*DUM(1,5)/12.
      DUM(3,6)=-ARM(IC,1)*DUM(2,6)/12.
      IF(IC.GT.2) GO TO 2
      DUM(1,3)=COS(ADC(IC,1))
      DUM(2,3)=COS(ADC(IC,3))
      DUM(3,3)=(ARM(IC,3)*DUM(1,3)-ARM(IC,1)*DUM(2,3))/12.
      CALL DLGTH (CX,CZ,CT,1,0)
      CALL DLGTH (CXP,CZP,CTP,2,0)
      CX= CX + CXP
      XPZ =-(CZ+CZP)/CX
      DUM(4,1) =XPZ
      XPT =-(CT+CTP)/CX
      DUM(4,2)=XPT
      DUM(4,4)= -1
      CALL DCOS_G (IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(6,4),
     1DUM(6,1),DUM(6,2))
      DUM(5,5)=-1
      DUM(6,6)=-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(IC.GT.4)GO TO 5
      CALL DLGTH(CX,CZ,CT,3,0)
      CALL DLGTH(CXP,CZP,CTP,4,0)
      DUM(7,1)=CZ+CZP
      DUM(7,2)=CT+CTP
      DUM(7,3)=CX+CXP
      DUM(4,7)=AKR*12.
    8 DUM(1,4)=COS(ADC(IC,1))
      DUM(2,4)=COS(ADC(IC,3))
      DUM(3,4)=(ARM(IC,3)*DUM(1,4)-ARM(IC,1)*DUM(2,4))/12.
      CALL DCOS_G(IC,DUM(5,3),DUM(5,1),DUM(5,2),DUM(6,3),DUM(6,1),DUM
     1(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)+DUM(J,K)
    6 IF(K.EQ.3)FXS(J,4)=FXS(J,4)+DUM(J,K)
      GO TO 1
    5 IF(KODE(11).EQ.0)GO TO 1
      CALL DLGTH(DUM(7,3),DUM(7,1),DUM(7,2),5,0)
      DUM(4,7)=AKLFT*12.
      GO TO 8
    1 CONTINUE
C  ADD SNUBBER INCREMENTS
      CALL LONGSN
      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 FRICT(0)
C  ZERO  CABLE EFFECTS FOR CABLELESS MODEL CHAR.
      IF(KODE(13).NE.-1.)GO TO 649
      DO 84 J=1,3
      DO 84 K=1,4
   84 FXS(J,K)=0.
      DO 85 J=1,3
      DO 85 K=1,6
   85 FRIC(J,K)=0.
      DO 86 J=1,3
      DO 86 K=1,3
   86 SNUD(J,K)=0.
C  THE CABLE FORCES/MOMENTS PARTIALS ARE COMPLETED
C  AERO. DATA IS NOW COMPUTED
  649 Q=RHO*VO*VO/2.
      QS=Q*SW
      IF(VO.NE.0.)QSV=QS/VO
      IF(VO.EQ.0.)QSV=0.
      XU=CXUP*QSV
      ZU=CZUP*QSV
      EMU=CMUP*QSV*CBAR
      XA=CXAP*QSV
      ZA=CZAP*QSV
      EMA=CMAP*QSV*CBAR
      IF(VO.NE.0.)XQ=CXQP*QSV*CBAR/(VO*2.)
      IF(VO.EQ.0.)XQ=0.
      IF(VO.NE.0.)ZQ=CZQP*QSV*CBAR/(VO*2.)
      IF(VO.EQ.0.)ZQ=0.
      EMQ=CMQP*QSV*CBAR/2.
      XDE=CXDEP*QS
      ZDE=CZDEP*QS
      EMDE=CMDEP*QS*CBAR
      IF(VO.NE.0.)XAD=CXADP*QSV*CBAR/(VO*2.)
      IF(VO.EQ.0.)XAD=0.
      IF(VO.NE.0.)ZAD=CZADP*QSV*CBAR/(VO*2.)
      IF(VO.EQ.0.)ZAD=0.
      IF(VO.NE.0.)EMAD=CMADP*QSV*CBAR/(2.*VO)
      IF(VO.EQ.0.)EMAD=0.
      IROW=14
```

```
      ICOL=14
      IORDER=3
   42 DO 20 I=1,IROW
      DO 20 J=1,ICOL
      DO 20 K=1,IORDER
   20 CMAT(I,J,K)=0.0
      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(1,1,3)=-XAD
      CMAT(1,2,1)=-FXS(1,2)+WT*COS(THETA)-XA*VO
      CMAT(1,2,2)=-XO-XAD*VO-SNUD(1,3)-FRIC(1,6)-FRIC(1,3)
      CMAT(1,2,3)=ZCG*AM/12.
      CMAT(1,3,1)=-FXS(1,3)
      CMAT(1,4,1)=-FXS(1,4)
      CMAT(1,4,2)=-XU-SNUD(1,1)-FRIC(1,4)-FRIC(1,1)
      CMAT(1,4,3)=AM
      CMAT(1,5,1)=-XDE
C  FZ EQUATION
      CMAT(2,1,1)=-FXS(2,1)
      CMAT(2,1,2)=-ZA-SNUD(2,2)-FRIC(2,5)-FRIC(2,2)
      CMAT(2,1,3)=AM-ZAD
      CMAT(2,2,1)=-FXS(2,2)+WT*SIN(THETA)-ZA*VO
      CMAT(2,2,2)=-ZO-ZAD*VO-SNUD(2,3)-FRIC(2,6)-FRIC(2,3)
      CMAT(2,2,3)=-XCG*AM/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)=-ZDE
C  MOMENT EQUATION
      CMAT(3,1,1)=-FXS(3,1)
      CMAT(3,1,2)=-EMA-SNUD(3,2)-FRIC(3,5)-FRIC(3,2)
      CMAT(3,1,3)=-EMAD*CBAR-XCG*AM/12.
      CMAT(3,2,1)=-FXS(3,2)-EMA*VO+ZCG*WT*COS(THETA)/12.
     1-XCG*WT*SIN(THETA)/12.
      CMAT(3,2,2)=(-EMO-EMAD*VO)*CBAR-SNUD(3,3)-FRIC(3,6)-FRIC(3,3)
      CMAT(3,2,3)=YIYY
      CMAT(3,3,1)=-FXS(3,3)
      CMAT(3,4,1)=-FXS(3,4)
      CMAT(3,4,2)=-EMU-SNUD(3,1)-FRIC(3,4)-FRIC(3,1)
      CMAT(3,4,3)=ZCG*AM/12.
      CMAT(3,5,1)=-EMDE
C  ELIMINATION OF DTF COL FOR CABLELESS MODEL CHAR.
      IF(KODE(13).NE.-1.)GO TO 81
      IF(KODE(8).NE.3.)WRITE(IW,82)
   82 FORMAT( 5X,'KODE(8).HAS BEEN SET BY PROG. TO 3. FOR CABLELESS MODE
     1L CHARACTERISTICS')
      KODE(8)=3.
      DO 83 I=1,3
      DO 83 J=1,3
   83 CMAT(I,3,K)=CMAT(I,4,K)
      GO TO 31
C  CONSTRAINT EQUATION
   81 CMAT(4,1,1)=-XPZ
```

```
      CMAT(4,2,1)=-XPT
      CMAT(4,4,1)=1
C   ACTIVE CABLE CONTROL EQS.

      IF (KODE(13).LE.0)GO TO 30
      CMAT(1,5,1)=0.0
      CMAT(2,5,1)=0.0
      CMAT(3,5,1)=0.0
      IF(KODE(6).EQ.1.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)=-(ARM(IC2,3)*COS(ADC(IC2,1))-ARM(IC2,1)*COS(ADC(IC2,3
     1)))/12.+(ARM(IC1,3)*COS(ADC(IC1,1))-ARM(IC1,1)*COS(ADC(IC1,3)))/12
     2.
C   EQ OF MOTOR DYN.
      CMAT(5,5,1)=+2.*RSBD*RSBA
      CMAT(5,5,2)=+2.*RSBD*FLSBA
      CMAT(5,7,1)=+AKSBT*2.
      CMAT(5,6,2)=-AKSBT*2.*AKSBV-GDMP*RSBA
      CMAT(5,6,3)=-AJASM*RSBA-GDMP*FLSBA
      CMAT(5,8,3)=-AJASM*FLSBA
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),IC1,0)
      CMAT(6,6,1)=-RSBD/12.
C   ACTIVE CABLE FEEDBACK EQ.
      CMAT(7,2,2)=AKO
      CMAT(7,6,1)=AKTH
      CMAT(7,6,2)=AKTHD
      CMAT(7,9,1)=-1.
C   TOTAL VOLTAGE EQ EM + EMC
      CMAT(9,7,1)=-1.
      CMAT(9,9,1)=1.
      CMAT(9,11,1)=1.
C   RELATION OF THM TO THMD
      CMAT(8,8,1)=-1.
      CMAT(8,6,2)=1.
C   RELATION OF TFDBK TO DTC AND INPUT DT
      CMAT(10,5,1)=1.
      CMAT(10,10,1)=1.
      CMAT(10,12,1)=-1.
      GO TO 31
C   FEEDBACK LOOP EQUATION
   30 CMAT(5,2,2)=AKTHE
      CMAT(5,5,2)=-TATHE
      CMAT(5,5,1)=-1.
   31 ITHD=0
      IF(KODE(14).EQ.0)GO TO 32
C   SUBST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS
      IDX=KODE(14)
      IDN=KODE(15)
```

```
      IF(IDN.NE.13)GO TO 52
      IDN=2
      ITHD=13
   52 IF(IDX.GT.14)GO TO 38
      DO 34 I=1,14
      DO 34 K=1,3
      BMAT(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
      BMAT(I,K)=CMAT(I,IDN,K)
   37 CMAT(I,IDN,K)=0.0
      IF(IDX.EQ.16)GO TO 39
      CMAT(1,IDN,1)=XDE
      CMAT(2,IDN,1)=ZDE
      CMAT(3,IDN,1)=EMDE
      GO TO 32
   39 CMAT(1,IDN,1)=XA
      CMAT(2,IDN,1)=ZA
      CMAT(3,IDN,1)=EMA
   32 N=KODE(8)
  655 CALL MATRIX(CMAT,N,ROOTS,K4A,IER)
      IF(KODE(14).EQ.0)GO TO 35
      DO 36 I=1,14
      DO 36 K=1,3
   36 CMAT(I,IDN,K)=BMAT(I,K)
C  35 IF(KODE(5).NE.0) WRITE(IW,100) IER
C 100 FORMAT(2X,'IER=',I3,3X,'SEE SUBR  PQEB AND PRBM FOR ERROR CODE')
C  THE ROOTS OF THE CHARAC. EQUAT. ARE IN THE COMPLEX ARRAY 'ROOTS'
C  AND THE NUMBER OF ROOTS IS 'K4A'
   35 K4A=K4A-1
      IF(ITHD.NE.13)GO TO 70
      K4A=K4A+1
      ROOTS(K4A)=(0.0,0.0)
      DO 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
  650 CONTINUE
C     NEW SNUBBER EFFECTS
      KODE(14)=0
      DO 600 IC=1,4
      DO 201 I=1,10
      DO 201 J=1,10
  201 DUM(I,J)=0.
      TC=TF-TR+TUSNO
      IF(IC.GT.2) TC=TUSNO
      DUM(1,3)=-TC*COS(ADC(IC,3))
      DUM(1,4)=-TC*SIN(ADC(IC,1))
      DUM(1,6)=COS(ADC(IC,1))
      DUM(2,3)=TC*COS(ADC(IC,1))
      DUM(2,5)=-TC*SIN(ADC(IC,3))
```

```
      DUM(2,6)=COS(ADC(IC,3))
      DUM(3,3)=(ARM(IC,3)*DUM(1,3)-ARM(IC,1)*DUM(2,3))/12.
      DUM(3,4)=ARM(IC,3)*DUM(1,4)/12.
      DUM(3,5)=-ARM(IC,1)*DUM(2,5)/12.
      DUM(3,6)=(ARM(IC,3)*DUM(1,6)-ARM(IC,1)*DUM(2,6))/12.
      CALL DCOSLG(IC,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)=AKSNU*12.
      IF(IC.GT.2) DUM(6,7)=AKSNL*12.
      CALL DLGTH(DUM(7,1),DUM(7,2),DUM(7,3),IC,0)
      DUM(7,7)=-1.
      CALL MASH(3,7)
      DO 200 J=1,3
      DO 200 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*VO
      CMAT(1,3,2)=-XQ-XAD*VO
      CMAT(1,3,3)=ZCG*AM/12.
      CMAT(1,1,2)=-XU
      CMAT(1,1,3)=AM
      CMAT(2,2,2)=-ZA
      CMAT(2,2,3)=AM-ZAD
      CMAT(2,3,1)=WT*SIN(THETA)-ZA*VO
      CMAT(2,3,2)=-ZQ-ZAD*VO
      CMAT(2,3,3)=-XCG*AM/12.
      CMAT(2,1,2)=-ZU
      CMAT(3,2,2)=-EMA
      CMAT(3,2,3)=-EMAD*CBAR-XCG*AM/12.
      CMAT(3,3,1)=-EMA*VO+ZCG*WT*COS(THETA)/12.-XCG*WT*SIN(THETA)/12.
      CMAT(3,3,2)=(-EMQ-EMAD*VO)*CBAR
      CMAT(3,3,3)=YIYY
      CMAT(3,1,2)=-EMU
      CMAT(3,1,3)=ZCG*AM/12.
      DO 700 I=1,3
      DO 700 J=1,3
  700 CMAT(I,J,1)=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 ROOTS'//)
      LKODE=KODE(14)
      KODE(14)=0
      CALL FREQ1(ROOTS,K4A,C4(K4A+1))
      GO TO 42
   41 KODE(14)=LKODE
```

```
      CALL FREQ(ROOTS,K4A,C4(K4A+1))
      RETURN
      END
      SUBROUTINE PRINTR (LOUT,RT,NROOT)
      COMMON/FRQ/C4(30)
      DIMENSION RT(2,1)
      K4=NROOT+1
      WRITE(LOUT,1)(C4(I),I=1,K4)
    1 FORMAT('  POLYNOMIAL W CONST TERM FIRST',/,(E27.6,4E16.6))
COMMENT   PRINTS PERTINENT INFORMATION ABOUT CHARACTERISTIC ROOTS
      WRITE(LOUT,507)
  507 FORMAT(        '       REAL          IMAGINARY      T H/D-SEC      1/T H/',
     1               'D      PERIOD-SEC     DNATF-CPS    UNDNAT-CPS     DAMP ',
     2               'RATIO      DECAY RATIO '                                  )
      NEXT=1
      IF( NROOT.GT.0 ) GO TO 5
      WRITE(LOUT,2)
    2 FORMAT(5X,'NO ROOTS')
      RETURN
    5 DO 530 I=1,NROOT
      IF(NEXT.EQ.2) GO TO 777
      SIG=RT(1,I)
      ASIG=ABS(SIG)
      AWD=ABS(RT(2,I))
      THDI= ASIG*1.442695
      THD= 99999.
      IF(THDI.GT.1.E-5) THD= 1./THDI
      IF(AWD.EQ.0.) GO TO 531
      NEXT=2
      WD=-AWD
      DNAT= AWD * .159155
      PER= 99999.
      IF(DNAT.GT.1.E-5) PER= 1./DNAT
      UNDNAT= SQRT(ASIG**2+AWD**2) *.1591550
      DAMPR= 0.
      IF( AWD - 1.E15 * ASIG ) 503,504,504
  503 DAMPR= SIGN ( COS( ATAN ( AWD/ASIG ) ), -SIG )
  504 CHDI= THDI*PER
      DECR= 99999.
      ARG= SIG * PER
      IF(ARG.LT.174.6) DECR= EXP (ARG)
      WRITE(LOUT,529) SIG,WD,THD,THDI,PER,DNAT,UNDNAT,DAMPR,DECR
  529 FORMAT(E12.4,2X,1H+,E11.4,8E13.4)
      GO TO 530
  531 WRITE(LOUT,532) SIG,THD,THDI
  532 FORMAT(E12.4,14X,2E13.4)
      GO TO 530
  777 NEXT=1
  530 CONTINUE
      RETURN
      END
      SUBROUTINE MASH (NN,N)
      COMMON /DU/DUM(10,10)
C   NN = FINAL MATRIX SIZE
C   N =  ORIGINAL MATRIX SIZE
```

```
      INN=N-NN                                                          CAB00530
      DO 1001 LL=1,INN                                                  CAB00540
      L=N+1-LL                                                          CAB00550
      II=L-1                                                            CAB00560
      JJ=L-1                                                            CAB00570
      DO 1001 I=1,II                                                    CAB00580
      DO 1001 J=1,JJ                                                    CAB00590
 1001 DUM(I,J)= DUM(I,J)+DUM(L,J)*DUM(I,L)/(-DUM(L,L))                  CAB00600
      RETURN                                                            CAB00610
      END                                                               CAB00620
      SUBROUTINE LAT                                                    CAB00630
      COMMON/INPUT/IW,IR                                                CAB00640
      COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL                      CAB00650
      COMMON / PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF         CAB00660
      COMMON /DU/DUM(10,10)                                             CAB00670
      COMMON/FRQ/C4(30)                                                 CAB00680
      EQUIVALENCE(AERO(46),XCG),(AERO(47),ZCG)                          CAB00690
      EQUIVALENCE(AERO (63),THETA),(AERO (49),VO   ),(AERO (50),    AM) CAB00700
      EQUIVALENCE(AERO (51),RHO  ),(AERO (52),    WT),(AERO (53),B    ) CAB00710
      EQUIVALENCE(AERO (54),CBAR ),(AERO (55),SW   ),(AERO (56), XIXZ)  CAB00720
      EQUIVALENCE(AERO (57),XIXX ),(AERO (58),YIYY ),(AERO (59),ZIZZ ), CAB00730
     1            (AERO(95),AKF),(AERO(100),AKLFT)                      CAB00740
      EQUIVALENCE(AERO(117),THUSNO),(AERO(119),AKSNU),(AERO(122),AKSNL) CAB00750
      EQUIVALENCE(AERO(123), AKSY),(AERO(124),AKPHI),(AERO(125),AKTHE), CAB00760
     1            (AERO(126), AKAZ),(AERO(127), T1SY),(AERO(128),T2PHI), CAB00770
     2            (AERO(129),T3THE),(AERO(130), T4AZ)                   CAB00780
      EQUIVALENCE(AERO(131),AKSBT),(AERO(132),AKSBV),(AERO(133),AJASM), CAB00790
     1            (AERO(134),FSBA) ,(AERO(135),ELSBA),(AERO(136), PSBD), CAB00800
     2            (AERO(137),AKTHO),(AERO(138), AKTH),(AERO(139), DOMP), CAB00810
     3            (AERO(140),  AKO),(AERO(141),  AKZ),(AERO(142),AKPSO), CAB00820
     4            (AERO(143),  AKY),(AERO(144), AKYO)                   CAB00830
      EQUIVALENCE(AEROP( 1), CXUP),(AEROP( 2), CZUP),(AEROP( 3), CMUP), CAB00840
     1            (AEROP( 4), CXAP),(AEROP( 5), CZAP),(AEROP( 6), CMAP), CAB00850
     2            (AEROP( 7), CXQP),(AEROP( 8), CZQP),(AEROP( 9), CMQP), CAB00860
     3            (AEROP(10), CXPP),(AEROP(11), CZPP),(AEROP(12), CMPP), CAB00870
     4            (AEROP(13),CXDEP),(AEROP(14),CZDEP),(AEROP(15),CMDEP), CAB00880
     5            (AEROP(16),CXADP),(AEROP(17),CZADP)  AEROP(18),CMADP), CAB00890
     6            (AEROP(19), CYBP),(AEROP(20), CLBP)  EROP(21), CNBP), CAB00900
     7            (AEROP(22), CYPP),(AEROP(23), CLPP),.AEROP(24), CNPP), CAB00910
     8            (AEROP(25), CYRP),(AEROP(26), CLRP),(AEROP(27), CNRP), CAB00920
     9            (AEROP(28),CYDRP),(AEROP(29),CLDRP),(AEROP(30),CNDRP), CAB00930
     A            (AEROP(31),CYDAP),(AEROP(32),CLDAP),(AEROP(33),CNDAP), CAB00940
     B            (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP) CAB00950
      DIMENSION CMAT(14,14,3),BMAT(14,3)                                CAB00960
      COMPLEX ROOTS(44)                                                 CAB00970
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)                CAB00980
      COMMON /ROUGH/FRIC(3,6)                                           CAB00990
      DIMENSION FXS(3,3)                                                CAB01000
      DO 10 J=1,3                                                       CAB01010
      DO 10 K=1,3                                                       CAB01020
   10 FXS(J,K)=0.                                                       CAB01030
      IF(KODE(10).EQ.3)GO TO 650                                        CAB01040
      DO 111 IC=1,5                                                     CAB01050
      IF(KODE(11).EQ.0.AND.IC.EQ.5)GO TO 1                              CAB01060
      DO 3 J=1,8                                                        CAB01070
```

```
      DO 3 K=1,8
    3 DUM(J,K)=0.
      TENS=TF
      IF(IC.GT.2)TENS=TR
      IF(IC.GT.4)TENS=TLFT
      CA1=COS(ADC(IC,1))
      CA2=COS(ADC(IC,2))
      CA3=COS(ADC(IC,3))
      IF(ABS(CA1).LT..0001) CA1=0.
      IF(ABS(CA2).LT..0001) CA2=0.
      IF(ABS(CA3).LT..0001) CA3=0.
      DUM(1,2)=-TENS*CA1
      DUM(1,3)=TENS*CA3
      DUM(1,4)=CA2
      DUM(1,6)=-TENS*SIN(ADC(IC,2))
      DUM(2,2)=( ARM(IC,1)*DUM(1,2)-ARM(IC,2)*TENS*CA2)/12.
      DUM(2,3)= ARM(IC,1)*DUM(1,3)/12.
      DUM(2,4)=(ARM(IC,1)*CA2-ARM(IC,2)*CA1)/12.
      DUM(2,5)= ARM(IC,2)*TENS*SIN(ADC(IC,1))/12.
      DUM(2,6)= ARM(IC,1)*DUM(1,6)/12.
      DUM(4,4)=-1.
      DUM(4,8)=0.
      IF(IC.GT.2)DUM(4,8)=AKR*12.
      IF(IC.GT.4)DUM(4,8)=AKLFT*12.
      DUM(3,2)=-ARM(IC,3)*DUM(1,2)/12.
      DUM(3,3)=(-ARM(IC,3)*DUM(1,3)-ARM(IC,2)*TENS*CA2)/12.
      DUM(3,4)=(ARM(IC,2)*CA3-ARM(IC,3)*CA2)/12.
      DUM(3,7)=-ARM(IC,2)*TENS*SIN(ADC(IC,3))/12.
      DUM(3,6)=-ARM(IC,3)*DUM(1,6)/12.
      CALL DCOSD(IC,DUM(5,1),DUM(5,2),DUM(5,3),DUM(6,1),DUM(6,2),DUM(
     16,3),DUM(7,1),DUM(7,2),DUM(7,3))
      DUM(5,5)=-1.
      DUM(6,6)=-1.
      DUM(7,7)=-1.
      IF(IC.GT.2)GO TO 2
      CALL MASH(3,7)
    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(IC.GT.4)GO TO 5
      CALL DLGTH(CY,CPS,CPH,3,1)
      CALL DLGTH(CYP,CPSP,CPHP,4,1)
      DUM(8,1)=CY+CYP
      DUM(8,2)=CPS+CPSP
      DUM(8,3)=CPH+CPHP
      DUM(8,8)=-1.
      CALL MASH(3,8)
      GO TO 6
    5 IF(KODE(11).EQ.0)GO TO 1
      CALL DLGTH(DUM(8,1),DUM(8,2),DUM(8,3),5,1)
      DUM(8,8)=-1.
      CALL MASH(3,8)
      GO TO 6
    1 CONTINUE
```

```
  111 CONTINUE
C     COMPLETE SUMMATION OF CABLE FORCES & MOMENTS
C     ADD SNUBBER INCREMENTS
  112 CALL LATSN
      DO 8 J=1,3
      DO 8 K=1,3
    8 FXS(J,K)=FXS(J,K)+SNU(J,K)
      CALL FRICT(1)
C     ZERO CABLE EFFECTS FOR CABLELESS MODEL OPTION
      IF(KODE(13).NE.-1)GO TO 620
      IF(KODE(9).NE.3)WRITE(IW,22)
   22 FORMAT(5X,'KODE(9) HAS BEEN SET BY PROG TO 3 FOR CABLELESS MODEL C
     1HARACTERISTICS')
      KODE(9)=3
      DO 20 J=1,3
      DO 20 K=1,3
      SNUD(J,K)=0.
   20 FXS(J,K)=0.
      DO 21 J=1,3
      DO 21 K=1,6
   21 FRIC(J,K)=0.
      GO TO 620
  650 CONTINUE
      KODE(16)=0
      DO 610 I=1,3
      DO 610 J=1,6
  610 FRIC(I,J)=0.
      DO 611 I=1,3
      DO 611 J=1,3
      SNU(I,J)=0.
  611 SNUD(I,J)=0.
      DO 600 IC=1,4
      DO 605 I=1,10
      DO 605 J=1,10
  605 DUM(I,J)=0.
      TC=TF-TR+TUSNO
      IF(IC.GT.2) TC=TUSNO
      CA1=COS(ADC(IC,1))
      CA2=COS(ADC(IC,2))
      CA3=COS(ADC(IC,3))
      IF(ABS(CA1).LT..0001) CA1=0.
      IF(ABS(CA2).LT..0001) CA2=0.
      IF(ABS(CA3).LT..0001) CA3=0.
      DUM(1,2)=-TC*CA1
      DUM(1,3)=TC*CA3
      DUM(1,4)=CA2
      DUM(1,6)=-TC*SIN(ADC(IC,2))
      DUM(2,2)=(ARM(IC,1)*DUM(1,2)-ARM(IC,2)*TC*CA2)/12.
      DUM(2,3)=ARM(IC,1)*DUM(1,3)/12.
      DUM(2,4)=(ARM(IC,1)*CA2-ARM(IC,2)*CA1)/12.
      DUM(2,5)=ARM(IC,2)*TC*SIN(ADC(IC,1))/12.
      DUM(2,6)=ARM(IC,1)*DUM(1,6)/12.
      DUM(3,2)=-ARM(IC,3)*DUM(1,2)/12.
      DUM(3,3)=(-ARM(IC,3)*DUM(1,3)-ARM(IC,2)*TC*CA2)/12.
      DUM(3,4)=(ARM(IC,2)*CA3-ARM(IC,3)*CA2)/12.
```

```
      DUM(3,7)=-ARM(IC,2)*TC*SIN(ADC(IC,3))/12.
      DUM(3,6)=-ARM(IC,3)*DUM(1,6)/12.
      DUM(4,4)=-1.
      DUM(4,8)=AKSNU*12.
      IF(IC.GT.2) DUM(4,8)=AKSNL*12.
      CALL DCOSD(IC,DUM(5,1),DUM(5,2),DUM(5,3),DUM(6,1),DUM(6,2),
     1 DUM(6,3),DUM(7,1),DUM(7,2),DUM(7,3))
      DUM(5,5)=-1.
      DUM(6,6)=-1.
      DUM(7,7)=-1.
      DUM(8,8)=-1.
      CALL DLGTH(DUM(8,1),DUM(8,2),DUM(8,3),IC,1)
      CALL MASH(3,8)
      DO 649 J=1,3
      DO 649 K=1,3
  649 FXS(J,K)=FXS(J,K)+DUM(J,K)
  600 CONTINUE
C  ADD AERO INCREMENTS
  620 Q=.5*RHO*VO*VO
      QS=Q*SW
      IF(VO.NE.0.)QSV=QS/VO
      IF(VO.EQ.0.)QSV=0.
      IF(VO.NE.0.)BOV=B/(2.*VO)
      IF(VO.EQ.0.)BOV=0.
      YV=CYBP*QSV
      ELV=CLBP*QSV*B
      ENV=CNBP*QSV*B
      YP=CYPP*QS*BOV
      ELP=CLPP*BOV*QS*B
      ENP=CNPP*BOV*QS*B
      YR=CYRP*QS*BOV
      ELR=CLRP*BOV*QS*B
      ENR=CNRP*BOV*QS*B
      YDR=CYDRP*QS
      ENDR=CNDRP*QS*B
      ELDR=CLDRP*QS*B
      YDA=CYDAP*QS
      ENDA=CNDAP*QS*B
      ELDA=CLDAP*QS*B
      YDS=CYDSP*QS
      ENDS=CNDSP*QS*B
      ELDS=CLDSP*QS*B
   42 DO 113 I=1,14
      DO 113 J=1,14
      DO 113 K=1,3
  113 CMAT(I,J,K)=0.0
C  Y FORCE EQUATION
      CMAT(1,1,1)=-FXS(1,1)
      CMAT(1,1,2)=-YV-SNUD(1,1)-FRIC(1,4)-FRIC(1,1)
      CMAT(1,1,3)=AM
      CMAT(1,2,1)=-FXS(1,2)+YV*VO-WT*SIN(THETA)
      CMAT(1,2,2)=-YP-SNUD(1,2)-FRIC(1,5)-FRIC(1,2)
      CMAT(1,2,3)=AM*XCG/12.
      CMAT(1,3,1)=-FXS(1,3)-WT*COS(THETA)
      CMAT(1,3,2)=-YR-SNUD(1,3)-FRIC(1,6)-FRIC(1,3)
```

```
      CMAT(1,3,3)=-AM*ZCG/12.
C  YAW EQUATION
      CMAT(2,1,1)=-FXS(2,1)
      CMAT(2,1,2)=-ENV-SNUD(2,1)-FRIC(2,4)-FRIC(2,1)
      CMAT(2,1,3)=AM*XCG/12.
      CMAT(2,2,1)=-FXS(2,2)+ENV*VO-XCG*WT*SIN(THETA)/12.
      CMAT(2,2,2)=-ENP-SNUD(2,2)-FRIC(2,5)-FRIC(2,2)
      CMAT(2,2,3)=ZIZZ
      CMAT(2,3,1)=-FXS(2,3)-XCG*WT*COS(THETA)/12.
      CMAT(2,3,2)=-ENP-SNUD(2,3)-FRIC(2,6)-FRIC(2,3)
      CMAT(2,3,3)=-XIXZ
C  ROLL EQUATION
      CMAT(3,1,1)=-FXS(3,1)
      CMAT(3,1,2)=-ELV-SNUD(3,1)-FRIC(3,4)-FRIC(3,1)
      CMAT(3,1,3)=-AM*ZCG/12.
      CMAT(3,2,1)=-FXS(3,2)+ELV*VO+ZCG*WT*SIN(THETA)/12.
      CMAT(3,2,2)=-ELP-SNUD(3,2)-FRIC(3,5)-FRIC(3,2)
      CMAT(3,2,3)=-XIXZ
      CMAT(3,3,1)=-FXS(3,3)+ZCG*WT*COS(THETA)/12.
      CMAT(3,3,2)=-ELP-SNUD(3,3)-FRIC(3,6)-FRIC(3,3)
      CMAT(3,3,3)=XIXX
C  ACTIVE CABLE CONTROL EQUATIONS
      IF(KODE(13).NE.1)GO TO 30
      IF(KODE(6).EQ.1.OR.KODE(6).EQ.4)GO TO 46
      IC2=2
      IC1=1
      GO TO 47
   46 IC2=4
      IC1=3
   47 CMAT(1,10,1)=+(COS(ADC(IC2,2))-COS(ADC(IC1,2)))
      CMAT(3,10,1)=+(ARM(IC2,2)*COS(ADC(IC2,3))-ARM(IC2,3)*COS(ADC(IC2,2
     1)))/12.-(ARM(IC1,2)*COS(ADC(IC1,3))-ARM(IC1,3)*COS(ADC(IC1,2)))/12
     2.
      CMAT(2,10,1)=+(ARM(IC2,1)*COS(ADC(IC2,2))-AR    .2,2)*COS(ADC(IC2,
     11)))/12.-(ARM(IC1,1)*COS(ADC(IC1,2))-ARM(IC1,2)*COS(ADC(IC1,1)
     2))/12.
C EQ. OF MOTOR DYN.
      CMAT(4,4,1)=+2.*RSBD*RSBA
      CMAT(4,4,2)=+2.*RSPD*ELSBA
      CMAT(4,6,1)=+AKSBT*2.
      CMAT(4,5,2)=-AKSBT*2.*AKSBV-GDMP*RSBA
      CMAT(4,5,3)=-AJASM*RSBA-GDMP*ELSBA
      CMAT(4,7,3)=-AJASM*ELSBA
      CALL DLGTH(CMAT(5,1,1),CMAT(5,2,1),CMAT(5,3,1),IC1,1)
      CMAT(5,5,1)=+RSBD/12.
C  EQ FOR TOTAL VOLTAGE=ACTIVE SYSTEM+INPUT VOLTAGE,EMO
      CMAT(9,6,1)=-1.
      CMAT(9,9,1)=1.
      CMAT(9,11,1)=1.
C  FEEDBACK CONTROL EQ.
      CMAT(6,2,2)=AKPSD
      CMAT(6,5,1)=AKY
      CMAT(6,7,1)=AKYD
      CMAT(6,9,1)=-1.
C  RELATE ANGULAR RATES TO ANGULAR DISPLACEMENTS
```

```
      CMAT(8,2,2)=1.
      CMAT(8,8,1)=-1.
      CMAT(7,5,2)=1.
      CMAT(7,7,1)=-1.
C  RELATION OF DTC TO DT AND DTFB
      CMAT(10,4,1)=1.
      CMAT(10,10,1)=1.
      CMAT(10,12,1)=-1.
      GO TO 31
C   RUDDER FEEDBACK LOOP
   30 CMAT(4,2,2)=AKSY
      CMAT(4,4,2)=-T3SY
      CMAT(4,4,1)=-1.
C   AILERON FEEDBACK LOOP
      CMAT(5,3,2)=AKPHI
      CMAT(5,5,2)=-T2PHI
      CMAT(5,5,1)=-1.
      CMAT(1,4,1)=-QS*CYDRP
      CMAT(1,5,1)=-QS*CYDAP
      CMAT(2,4,1)=-QS*B*CNDRP
      CMAT(2,5,1)=-QS*B*CNDAP
      CMAT(3,4,1)=-QS*B*CLDRP
      CMAT(3,5,1)=-QS*B*CLDAP
   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.GT.13)GO TO 38
      DO 34 I=1,14
      DO 34 K=1,3
      BMAT(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
      BMAT(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)=ENDR
      CMAT(3,IDN,1)=ELDR
      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,IND,1)=YV
      CMAT(2,IDN,2)=ENV
      CMAT(3,IDN,3)=ELV
   32 N=KODE(9)
      CALL MATRIX(CMAT,N,ROOTS,K4A,IER)
      IF(KODE(16).EQ.0)GO TO 35
      DO 36 I=1,14
      DO 36 K=1,3
```

```
   36 CMAT(I,IDN,K)=RMAT(I,K)                                              CAB03830
C  35 IF(KODE(5).NE.0) WRITE(IW,100) IER                                   CAB03840
C 100 FORMAT(2X,'IER=',I3,3X,'SEE SUBR. POFB AND PRBM FOR ERROR CODE')      CAB03850
C  THE ROOTS OF THE CHARACTERISTIC EQUAT. ARE IN THE COMPLEX ARRAY         CAB03860
C  'ROOTS' AND THE NUMBER OF ROOTS IS 'K4A'                                CAB03870
   35 K4A=K4A-1                                                            CAB03880
      CALL PRINTR(IW,ROOTS,K4A)                                            CAB03890
      IF(KODE(3).NE.2)RETURN                                               CAB03900
      IF(KODE(16).EQ.0)GO TO 44                                            CAB03910
      WRITE(IW,43)                                                         CAB03920
   43 FORMAT(//'  COMPUTATION OF THE DENOMINATOR ROOTS'//)                 CAB03930
      LKODE=KODE(16)                                                       CAB03940
      KODE(16)=0                                                           CAB03950
      CALL FREQ1(ROOTS,K4A,C4(K4A+1))                                      CAB03960
      GO TO 42                                                             CAB03970
   44 KODE(16)=LKODE                                                       CAB03980
      CALL FREQ2(ROOTS,K4A,C4(K4A+1))                                      CAB03990
      RETURN                                                               CAB04000
      END                                                                  CAB04010
      SUBROUTINE DCOSD(IC,CY1,CPSI1,CPHI1,CY2,CPSI2,CPHI2,CY3,CPSI3,       CAB04020
     1CPHI3)                                                               CAB04030
      COMMON /PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF             CAB04040
      IF(ABS(ADC(IC,3)-3.14159).GT..00001)GO TO 2                          CAB04050
      XVAL=1000.                                                           CAB04060
      GO TO 1                                                              CAB04070
    2 XVAL=COTAN(ADC(IC,3))                                                CAB04080
    1 XWT=ARM(IC,1)                                                        CAB04090
      YWT=ARM(IC,2)                                                        CAB04100
      ZWT=ARM(IC,3)                                                        CAB04110
      CY1=-COS(ADC(IC,2))*COTAN(ADC(IC,1))/XLGTH(IC)*12.                   CAB04120
      CPSI1=-(YWT*SIN(ADC(IC,1))+XWT*COS(ADC(IC,2))*COTAN(ADC(IC,1)))      CAB04130
     1/XLGTH(IC)                                                           CAB04140
      CPHI1=(ZWT*COS(ADC(IC,2))*COTAN(ADC(IC,1))-YWT*COS(ADC(IC,3))*       CAB04150
     1COTAN(ADC(IC,1)))/XLGTH(IC)                                          CAB04160
      CY2=SIN(ADC(IC,2))/XLGTH(IC)*12.                                     CAB04170
      CPSI2=(YWT*COS(ADC(IC,1))*COTAN(ADC(IC,2))+XWT*SIN(ADC(IC,2)))/      CAB04180
     1XLGTH(IC)                                                            CAB04190
      CPHI2=-(ZWT*SIN(ADC(IC,2))+YWT*COS(ADC(IC,3))*COTAN(ADC(IC,2)))      CAB04200
     1/XLGTH(IC)                                                           CAB04210
      CY3=-COS(ADC(IC,2))*XVAL/XLGTH(IC)*12.                               CAB04220
      CPSI3=(YWT*COS(ADC(IC,1))*XVAL-XWT*COS(ADC(IC,2))*                   CAB04230
     1XVAL)/XLGTH(IC)                                                      CAB04240
      CPHI3=(ZWT*COS(ADC(IC,2))*XVAL+YWT*SIN(ADC(IC,3)))                   CAB04250
     1/XLGTH(IC)                                                           CAB04260
      RETURN                                                               CAB04270
      END                                                                  CAB04280
      SUBROUTINE SNTRM (FXSN,FZSN,AMSN,THETA)                              CAB00010
      COMMON/INPUT/IW,IR                                                   CAB00020
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL                           CAB00030
      COMMON ZZZ(200)                                                      CAB00040
      COMMON/TAB1/ZZ(300)                                                  CAB00050
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)                   CAB00060
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ),    CAB00070
     1           (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),    CAB00080
     2           (AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),    CAB00090
```

```
     3           (AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),
     4           (AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),
     5           (AERO(120),AKSNL),(AERO(49),   VO),(AERO(51),   RHO),
     6           (AERO(76),WLCR),(AERO(77),STACR),
     7           (AERO(78),BLCR)
      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),THGX1),(SN(20),THGY1),(SN(21),THGZ1),
     7           (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),
     8           (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),
     9           (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)
      IF(KODE(10).EQ.0) GO TO 5005
      IF(KODE(10).EQ.3) GO TO 5005
      CALL DFCSN(THETA)
      IF(KODE(10).NE.1) GO TO 5003
C  TERMS TO MODEL SNUBBER EFFECTS (MODEL UNSNUBBED)
      Q=.5*RHO*VO*VO
      CALL STINT(Q,ALU,0,1,1,TUSN,NG)
      IF(NG.NE.0) GO TO 5000
      CALL STINT(Q,ALL,0,1,1,TLSN,NG)
      IF(NG.NE.0) GO TO 5000
      CALL STINT(Q,ALU,0,2,2,THUSN,NG)
      IF(NG.NE.0) GO TO 5000
      CALL STINT(Q,ALL,0,2,2,THLSN,NG)
      IF(NG.EQ.0) GO TO 5001
 5000 WRITE(IW,5002) NG,ALL,ALU,Q
 5002 FORMAT(2X,'ERROR IN SNUBBER TABLE 1-2   , NG=',I3,3E10.3)
      RETURN
 5001 CONTINUE
C       CALCULATING FORCE AND MOMENT EFFECTS
      CALL DFCUSN(THETA)
      FXUSN= 2.*TUSN*GX1
      FZUSN= 2.*TUSN*GZ1
      AMUSN= -FXUSN*SNUZ+SNUX*FZUSN
      FXLSN= 2.*TLSN*GX3
      FZLSN= 2.*TLSN*GZ3
      AMLSN= FXLSN*SNLZ+FZLSN*SNLX
      FXSN = FXUSN+FXLSN
      FZSN = FZUSN+FZLSN
      AMSN =(AMUSN+AMLSN)/12.
      RETURN
 5003 CONTINUE
C  TERMS TO MODEL SNUBBER EFFECTS (MODEL SNUBBED)
      FXUSN= 2.*TUSNO*GX1
      FZUSN= 2.*TUSNO*GZ1
      AMUSN =-FXUSN*SNUZ+FZUSN*SNUX
      FXLSN= 2.*TLSNO*GX3
      FZLSN= 2.*TLSNO*GZ3
      AMLSN = FXLSN*SNLZ+FZLSN*SNLX
      FXSN = FXUSN+FXLSN
      FZSN = FZUSN+FZLSN
```

```
      AMSN =(AMUSN+AMLSN)/12.                                           CAB00650
      RETURN                                                            CAB00660
 5005 FXSN=0                                                            CAB00670
      FZSN=0                                                            CAB00680
      AMSN=0                                                            CAB00690
      RETURN                                                            CAB00700
      END                                                               CAB00710
      SUBROUTINE LONGSN                                                 CAB00720
      COMMON/INPUT/IW,IP                                                CAB00730
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL                        CAB00740
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)                CAB00750
      COMMON ZZZ(200)                                                   CAB00760
      COMMON/TAB1/ZZ(800)                                               CAB00770
      COMMON/DU/DUM(10,10)                                              CAB00780
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ), CAB00790
     1            (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),CAB00800
     2            (AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),CAB00810
     3            (AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),CAB00820
     4            (AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),CAB00830
     5            (AERO(120),AKSNL),(AERO(49),    VO),(AERO(51),   RHO),CAB00840
     6            (AERO(63),THETA),(AERO(121),ADSNU),(AERO(122),ADSNL)  CAB00850
      EQUIVALENCE (SN( 1),  GX1),(SN( 2),  GY1),(SN( 3),  GZ1),         CAB00860
     1            (SN( 4),  GX2),(SN( 5),  GY2),(SN( 6),  GZ2),         CAB00870
     2            (SN( 7),  GX3),(SN( 8),  GY3),(SN( 9),  GZ3),         CAB00880
     3            (SN(10),  GX4),(SN(11),  GY4),(SN(12),  GZ4),         CAB00890
     4            (SN(13),  THU),(SN(14),  THL),(SN(15),  ALU),         CAB00900
     5            (SN(16),  ALL),                                       CAB00910
     6            (SN(19),THGX1),(SN(20),THGY1),(SN(21),THGZ1),         CAB00920
     7            (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),         CAB00930
     8            (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),         CAB00940
     9            (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)          CAB00950
      DIMENSION FTOP(3,3),FBOT(3,3)                                     CAB00960
      COT(A)=1./TAN(A)                                                  CAB00970
      DO 1001 I=1,3                                                     CAB00980
      DO 1001 J=1,3                                                     CAB00990
      SNU(I,J)=0                                                        CAB01000
 1001 SNUD(I,J)=0                                                       CAB01010
      DO 5102 I=1,10                                                    CAB01020
      DO 5102 J=1,10                                                    CAB01030
 5102 DUM(I,J)=0                                                        CAB01040
      IF(KODE(10).NE.1) GO TO 1000                                      CAB01050
C  TERMS FOR UNSNUBBED SNUBBER EFFECTS (LONG)                           CAB01060
      DO 1004 I=1,7                                                     CAB01070
      DO 1004 J=1,7                                                     CAB01080
 1004 DUM(I,J)=0                                                        CAB01090
      CALL DRCUSN(THETA)                                                CAB01100
      DUM(1,3)= -2.*TUSNO*GZ1                                           CAB01110
      DUM(1,4)= -2.*TUSNO*SIN(THGX1)                                    CAB01120
      DUM(1,6)=  2.*GX1                                                 CAB01130
      DUM(2,3)=  2.*TUSNO*GX1                                           CAB01140
      DUM(2,5)= -2.*TUSNO*SIN(THGZ1)                                    CAB01150
      DUM(2,6)=  2.*GZ1                                                 CAB01160
      DUM(3,3)= (-SNUZ*DUM(1,3)+SNUX*DUM(2,3))/12.                      CAB01170
      DUM(3,4)= -SNUZ*DUM(1,4)/12.                                      CAB01180
      DUM(3,5)=  SNUX*DUM(2,5)/12.                                      CAB01190
```

```
      DUM(3,6)= (-SNUZ*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)= -SNUZ*SIN(THGX1)/ALU-SNUX*GZ1*COT(THGX1)/ALU
      DUM(4,4)=  -1.
      DUM(5,1)= (-GX1*COT(THGZ1)/ALU)*12.
      DUM(5,2)=  (SIN(THGZ1)/ALU)*12.
      DUM(5,3)=  SNUZ*GX1*COT(THGZ1)/ALU + SNUX*SIN(THGZ1)/ALU
      DUM(5,5)= -1.
      CALL DFCSN(THETA)
      Q=.5*RHO*V0*V0
      ALU1=ALU+1.
      CALL STINT(Q,ALU1,C,1,1,TUSN1,NG)
      IF(NG.NE.0) GO TO 5000
      ALU2=ALU-1.
      CALL STINT(Q,ALU2,C,1,1,TUSN2,NG)
      IF(NG.EQ.0) GO TO 5001
 5000 WRITE(IW,5002) NG,ALL,ALU,Q
 5002 FORMAT('ERROR IN TABLE 1-2,NG=',I2,3XE10.3)
      RETURN
 5001 CONTINUE
      AKTU=(TUSN1-TUSN2)/2.
      DUM(6,6)= -1.
      DUM(6,7)= AKTU*12.
      DUM(7,1)= -GX1
      DUM(7,2)= -GZ1
      DUM(7,3)=((-SNUX+ALU*GX1)*GZ1-(-SNUZ+ALU*GZ1)*GX1)/12.
      DUM(7,7)= -1.
      CALL MASH(3,7)
      DO 1005 I=1,3
      DO 1005 J=1,3
 1005 FTOP(I,J)=DUM(I,J)
      CALL DFCUSN(THETA)
      DUM(1,3)= -2.*TLSNO*GZ3
      DUM(1,4)= -2. TLSNO*SIN(THGX3)
      DUM(1,6)=  2.*GX3
      DUM(2,3)=  2.*TLSNO*GX3
      DUM(2,5)= -2.*TLSNO*SIN(THGZ3)
      DUM(2,6)=  2.*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(3,6)=  (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)=  (-GX3*COT(THGZ3)/ALL)*12.
      DUM(5,2)=  (SIN(THGZ3)/ALL)*12.
      DUM(5,3)=  -SNLZ*GX3*COT(THGZ3)/ALL + SNLX*SIN(THGZ3)/ALL
      DUM(5,5)=  -1.
      CALL DFCSN(THETA)
      ALL1=ALL+1.
      CALL STINT(Q,ALL1,C,1,1,TLSN1,NG)
      IF(NG.NE.0) GO TO 5003
```

```
      ALL2=ALL-1.
      CALL STINT(Q,ALL2,0,1,1,TLSN2,NG)
      IF(NG.EQ.0) GO TO 5004
 5003 WRITE(IW,5002) NG,ALL,ALU,Q
      RETURN
 5004 CONTINUE
      AKTL=(TLSN1-TLSN2)/2.
      DUM(6,6)=  -1.
      DUM(6,7)=  AKTL*12.
      DUM(7,1)=  -GX3
      DUM(7,2)=  -GZ3
      DUM(7,3)=((-SNLX+ALL*GX3)*GZ3-(SNLZ+ALL*GZ3)*GX3)/12.
      DUM(7,7)=  -1.
      CALL MASH(3,7)
      DO 1008 I=1,3
      DO 1008 J=1,3
 1008 FBOT(I,J)=DUM(I,J)
      DO 1009 I=1,3
      DO 1009 J=1,3
      SNUD(I,J)=0
 1009 SNU(I,J)= FTOP(I,J)+FBOT(I,J)
      RETURN
 1000 IF(KODE(10).EQ.0) GO TO 1002
C   TERMS FOR SNUBBED SNUBBER EFFECTS(LONG)
      CALL DRCSN(THETA)
      DO 1006 I=1,7
      DO 1006 J=1,7
 1006 DUM(I,J)=0
      DUM(1,3)= -2.*TUSNO*GZ1
      DUM(1,4)= -2.*TUSNO*SIN(THGX1)
      DUM(1,6)=  2.*GX1
      DUM(2,3)=  2.*TUSNO*GX1
      DUM(2,5)= -2.*TUSNO*SIN(THGZ1)
      DUM(2,6)=  2.*GZ1
      DUM(3,3)= (-SNUZ*DUM(1,3)+SNUX*DUM(2,3))/12.
      DUM(3,4)= -SNUZ*DUM(1,4)/12.
      DUM(3,5)=  SNUX*DUM(2,5)/12.
      DUM(3,6)= (-SNUZ*DUM(1,6)+SNUX*DUM(2,6))/12.
      DUM(4,1)=  (SIN(THGX1)/ALU)*12.
      DUM(4,2)= (-GZ1*COT(THGX1)/ALU)*12.
      DUM(4,3)= -SNUZ*SIN(THGX1)/ALU-SNUX*GZ1*COT(THGX1)/ALU
      DUM(4,4)=  -1.
      DUM(5,1)= (-GX1*COT(THGZ1)/ALU)*12.
      DUM(5,2)=  (SIN(THGZ1)/ALU)*12.
      DUM(5,3)=  SNUZ*GX1*COT(THGZ1)/ALU + SNUX*SIN(THGZ1)/ALU
      DUM(5,5)= -1.
      DUM(6,6)= -1.
      DUM(6,7)= AKSNU*12.
      DUM(7,1)= -GX1
      DUM(7,2)= -GZ1
      DUM(7,3)= ((-SNUX+ALU*GX1)*GZ1-(-SNUZ+ALU*GZ1)*GX1)/12.
      DUM(7,7)= -1.
      DO 10 I=1,3
      DO 10 J=1,3
   10 SNUD(I,J)=DUM(I,6)*ADSNU*DUM(7,J)*12.
```

```
      CALL MASH(3,7)
      DO 1007 I=1,3
      DO 1007 J=1,3
 1007 FTOP(I,J)=DUM(I,J)
      DUM(1,3)= -2.*TLSNO*GZ3
      DUM(1,4)= -2.*TLSNO*SIN(THGX3)
      DUM(1,6)=  2.*GX3
      DUM(2,3)=  2.*TLSNO*GX3
      DUM(2,5)= -2.*TLSNO*SIN(THGZ3)
      DUM(2,6)=  2.*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(3,6)=  (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)=  (-GX3*COT(THGZ3)/ALL)*12.
      DUM(5,2)=  (SIN(THGZ3)/ALL)*12.
      DUM(5,3)=  -SNLZ*GX3*COT(THGZ3)/ALL + SNLX*SIN(THGZ3)/ALL
      DUM(5,5)=  -1.
      DUM(6,6)=  -1.
      DUM(6,7)=  AKSN *12.
      DUM(7,1)=  -GX
      DUM(7,2)=  -GZ
      DUM(7,3)= ((-SNLX+ALL*GX3)*GZ3 -(SNLZ+ALL*GZ3)*GX3)/12.
      DUM(7,7)=  -1.
      DO 20 I=1,3
      DO 20 J=1,3
   20 SNUD(I,J)=SNUD(I,J)+DUM(I,6)*AKSNL*DUM(7,J)*12.
      CALL MASH(3,7)
      DO 1010 I=1,3
      DO 1010 J=1,3
 1010 FBOT(I,J)=DUM(I,J)
      DO 1011 I=1,3
      DO 1011 J=1,3
 1011 SNU(I,J)= FTOP(I,J)+FBOT(I,J)
      RETURN
 1002 DO 1003 I=1,3
      DO 1003 J=1,3
      SNUD(I,J)=0
 1003 SNU(I,J)=0
      RETURN
      END
      SUBROUTINE DECSN(THETA)
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ),
     1            (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),
     2            (AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),
     3            (AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),
     4            (AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),
     5            (AERO(120),AKSNL),
     6            (AERO(76),WLCF),(AERO(77),STACF),(AERO(78),BLCF)
```

```
      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),THGX1),(SN(20),THGY1),(SN(21),THGZ1),
     7            (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),
     8            (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),
     9            (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)
C  CALCULATION OF SNUBBER CABLE DIRECTION COSINES
      XB1= (STACR-SNUST)*COS(THETA)-(WLCR-SNUWL)*SIN(THETA)
      ZB1= (WLCR-SNUWL)*COS(THETA)+(STACR-SNUST)*SIN(THETA)
      XB2= XB1
      ZB2= ZB1
      XB3= (STACR-SNLST)*COS(THETA)-(WLCR-SNLWL)*SIN(THETA)
      ZB3= (WLCR-SNLWL)*COS(THETA)+(STACR-SNLST)*SIN(THETA)
      XB4= XB3
      ZB4=ZB3
      DX1= XB1+SNUX
      DY1= -SNUBL+SNUY
      DZ1= ZB1+SNUZ
      DX2= DX1
      DY2= SNUBL-SNUY
      DZ2= DZ1
      DX3= XB3+SNLX
      DY3= SNLBL-SNLY
      DZ3= ZB3-SNLZ
      DX4= DX3
      DY4= -SNLBL+SNLY
      DZ4= DZ3
      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=19,30
      J=I-18
    1 SN(I)=ARCOS(SN(J))
      RETURN
      END
      SUBROUTINE DECUSN(THETA)
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
      COMMON/SNUBB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
```

```
      EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ), CAB00660
     1           (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),CAB00670
     2           (AERO(111),SNUST),(AERO(112),SNUWL),(AERO(113),SNUBL),CAB00680
     3           (AERO(114),SNLST),(AERO(115),SNLWL),(AERO(116),SNLBL),CAB00690
     4           (AERO(117),TUSNO),(AERO(118),TLSNO),(AERO(119),AKSNU),CAB00700
     5           (AERO(120),AKSNL),                                    CAB00710
     6           (AERO(76),WLCP),(AERO(77),STACP),(AERO(78),BLCP)      CAB00720
      EQUIVALENCE (SN( 1),  GX1),(SN( 2),  GY1),(SN( 3),  GZ1),        CAB00730
     1           (SN( 4),  GX2),(SN( 5),  GY2),(SN( 6),  GZ2),         CAB00740
     2           (SN( 7),  GX3),(SN( 8),  GY3),(SN( 9),  GZ3),         CAB00750
     3           (SN(10),  GX4),(SN(11),  GY4),(SN(12),  GZ4),         CAB00760
     4           (SN(13),  THU),(SN(14),  THL),(SN(15),  ALU),         CAB00770
     5           (SN(16),  ALL),                                       CAB00780
     6           (SN(19),THGX1),(SN(20),THGY1),(SN(21),THGZ1),         CAB00790
     7           (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2),         CAB00800
     8           (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3),         CAB00810
     9           (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4)          CAB00820
C   CALCULATION FOR EFFECTIVE DIRECTION COSINES FOR UNSNUBBED CASE     CAB00830
      AYL = SNLBL-(BLCP+SNLY)                                          CAB00840
      AZL =-SNLWL-(WLCP+SNLZ+SNLX*SIN(THETA))                          CAB00850
      AYU = SNUBL-(BLCP+SNUY)                                          CAB00860
      AZU = SNUWL-(WLCP+SNUZ-SNUX*SIN(THETA))                          CAB00870
      THU= ATAN(AZU/AYU)                                               CAB00880
      THL= ATAN(AZL/AYL)                                               CAB00890
      ALU=AYU/(SIN(THUSN)*COS(THU))                                    CAB00900
      GX1S= -COS(THUSN)                                                CAB00910
      GY1S= -AYU/ALU                                                   CAB00920
      GZ1S= -AZU/ALU                                                   CAB00930
      GX1 = GX1S*COS(THETA)-GZ1S*SIN(THETA)                            CAB00940
      GY1 = GY1S                                                       CAB00950
      GZ1 = GZ1S*COS(THETA)+GX1S*SIN(THETA)                            CAB00960
      GX2 = GX1                                                        CAB00970
      GY2 =-GY1                                                        CAB00980
      GZ2 = GZ1                                                        CAB00990
      ALL=AYL/(SIN(THLSN)*COS(THL))                                    CAB01000
      GX3S= -COS(THLSN)                                                CAB01010
      GY3S= AYL/ALL                                                    CAB01020
      GZ3S= AZL/ALL                                                    CAB01030
      GX3 = GX3S*COS(THETA)-GZ3S*SIN(THETA)                            CAB01040
      GY3 = GY3S                                                       CAB01050
      GZ3 = GZ3S*COS(THETA)+GX3S*SIN(THETA)                            CAB01060
      GX4 = GX3                                                        CAB01070
      GY4 =-GY3                                                        CAB01080
      GZ4 = GZ3                                                        CAB01090
      DO 1 I=19,30                                                     CAB01100
      J=I-18                                                           CAB01110
    1 SN(I)=ARCOS(SN(J))                                               CAB01120
      RETURN                                                           CAB01130
      END                                                              CAB01140
      SUBROUTINE RITE                                                  CAB01150
      COMMON/INOU[illegible],IF                                        CAB01160
      COMMON/DAT/A[illegible](175),AEROP(50),KODE(26),LL               CAB01170
      IF(KODE(6).GT.1) GO TO 1                                         CAB01180
      WRITE(IW,100)                                                    CAB01190
  100 FORMAT(25X,'FRONT CABLE VERTICAL,REAR CABLE HORIZONTAL')         CAB01200
```

```
      GO TO 4                                                           CAB01210
    1 IF(KODE(6).GT.2) GO TO 2                                          CAB01220
      WRITE(IW,200)                                                     CAB01230
  200 FORMAT(25X,'FRONT CABLE HORIZONTAL,REAR CABLE VERTICAL')          CAB01240
      GO TO 4                                                           CAB01250
    2 IF(KODE(6).GT.3) GO TO 3                                          CAB01260
      WRITE(IW,300)                                                     CAB01270
  300 FORMAT(25X,'BOTH CABLES VERTICAL')                                CAB01280
      GO TO 4                                                           CAB01290
    3 WRITE(IW,400)                                                     CAB01300
  400 FORMAT(25X,'BOTH CABLES HORIZONTAL')                              CAB01310
    4 CONTINUE                                                          CAB01320
      IF(KODE(10).EQ.0) GO TO 5                                         CAB01330
      IF(KODE(10).EQ.1) GO TO 6                                         CAB01340
      WRITE(IW,500)                                                     CAB01350
  500 FORMAT(25X,'SNUBBERS SNUBBED')                                    CAB01360
      GO TO 7                                                           CAB01370
    5 WRITE(IW,600)                                                     CAB01380
  600 FORMAT(25X,'NO SNUBBERS')                                         CAB01390
      GO TO 7                                                           CAB01400
    6 WRITE(IW,700)                                                     CAB01410
  700 FORMAT(25X,'SNUBBERS UNSNUBBED')                                  CAB01420
    7 CONTINUE                                                          CAB01430
      IF(KODE(11).EQ.0) GO TO 8                                         CAB01440
      WRITE(IW,800)                                                     CAB01450
  800 FORMAT(25X,'LIFT/ANTI-LIFT CABLE IN')                             CAB01460
      GO TO 9                                                           CAB01470
    8 WRITE(IW,900)                                                     CAB01480
  900 FORMAT(25X,'NO LIFT/ANTI-LIFT CABLE')                             CAB01490
    9 CONTINUE                                                          CAB01500
      IF(KODE(13).LE.0.)WRITE(IW,1000)                                  CAB01510
      IF(KODE(13).GT.0.)WRITE(IW,1001)                                  CAB01520
      IF(KODE(13).EQ.-1.)WRITE(IW,1002)                                 CAB01530
 1000 FORMAT(25X,'FEEDBACK LOGIC NOT IN')                               CAB01540
 1001 FORMAT(25X,'FEEDBACK LOGIC IN')                                   CAB01550
 1002 FORMAT(25X,'CABLELESS MODEL CHARACTERISTICS')                     CAB01560
      RETURN                                                            CAB01570
      END                                                               CAB01580
      SUBROUTINE STINT(A1,A2,A3,MINTBL,MAXTBL,FCT,NG)                   CAB00010
      EQUIVALENCE (X(1),NUMPTS(1))                                      CAB00020
      COMMON NUMPTS(1)                                                  CAB00030
      DIMENSION X(1)                                                    CAB00040
      I7=NUMPTS(1)/3                                                    CAB00050
70    IF(MINTBL-MAXTBL)71,71,110                                        CAB00060
   71 DO 73 II=MINTBL,MAXTBL                                            CAB00070
      NJ=NUMPTS(II)+1                                                   CAB00080
      IF(A3-X(NJ))72,74,73                                              CAB00090
   72 IF(II-MINTBL) 110,112,75                                          CAB00100
   73 CONTINUE                                                          CAB00110
      GO TO 112                                                         CAB00120
   75 IK = 1                                                            CAB00130
      IL =2                                                             CAB00140
      NM=NJ                                                             CAB00150
  101 DO 97 IF=IK,IL                                                    CAB00160
      NJ =NUMPTS(II)+1                                                  CAB00170
```

```
      NI = IZ+II
      IO =NUMPTS(NI)
      IP =IO+NJ
      DO 77 IQ=1,IO
      NN= NJ+IQ
      IF (A1-X(NN))76,78,77
   76 IF(IQ-1) 110,112,79
   77 CONTINUE
      GO TO 112
   78 IG =-1
      GO TO 80
   79 IG =+1
   80 NI=NI+IZ
      IB = NUMPTS(NI)
      DO 82 IA=1,IB
      NS=IP+IA
      IF (A2-X(NS))81,83,82
   81 IF(IA-1) 110,112,84
   82 CONTINUE
      GO TO 112
   83 IH =-1
      GO TO 85
   84 IH =+1
   85 NE=IP+IB+IQ+IO*IA-IO
      NR=NE-IO
      IF(IG+IH) 86,88,91
   86 IF (X(NE)-99998.5E9)87,113,113
   87 FCT = X(NE)
      GO TO 95
   88 IF(IG) 89,110,93
   89 IF(AMAX1(X(NE),X(NR))-99998.5E9)90,113,113
   90 FCT =X(NE)-(X(NS)-A2)*(X(NE)-X(NR))/(X(NS)-X(NS-1))
      GO TO 95
   91 IF(AMAX1(X(NE),X(NR),X(NE-1),X(NR-1))-99998.5E9)92,113,113
   92 FCT = ((X(NS)-A2)*((X(NN)-A1)*X(NR-1)-(X(NN-1)-A1)*X(NR)
     1)-(X(NS-1)-A2)*((X(NN)-A1)*X(NE-1)-(X(NN-1)-A1)*X(NE)))
     2/((X(NS)-X(NS-1))*(X(NN)-X(NN-1)))
      GO TO 95
   93 IF(AMAX1(  X(NE),  X(NE-1))-99998.5E9) 94,113,113
   94 FCT =  X(NE)-(  X(NN)-A1)*(  X(NE)-  X(NE-1))/(  X(NN)-  X(NN-1))
   95 GO TO (96,98,99),IF
   96 DUMSTG =FCT
   97 II =II-1
   98 FCT =DUMSTG-(  X(NM)-A3)*(DUMSTG-FCT)/(  X(NM)-  X(NJ))
   99 RETURN
   74 IK =3
      IL =3
      GO TO 101
  110 NG =2
      GO TO 99
  112 NG =3
      GO TO 99
  113 NG =4
      GO TO 99
      END
```

```
      SUBROUTINE TABIN(NUMTBL,NZ,NG)
      COMMON NUMPTS(1)
      COMMON/INOUT/IW,IR
      COMMON /TABOUT/ NIMTBL,ISEQ
      DIMENSION XUMPTS(1)
      INTEGER*2 LABEL(27)
      EQUIVALENCE (XUMPTS(1),NUMPTS(1)),(DUMMY(1),MUMMY)
      DIMENSION DUMMY(10)
      MCR=0
   10 IZ=IABS(NZ)
      NUNIT=5
      IF(NZ.LT.0) NUNIT=8
      NIMTBL = NUMTBL
      NG=0
      NUMPTS(1)=IZ+IZ+IZ
  102 READ(NUNIT,57) K, L1N, L2N, LABEL, ISEQ
      IF(MCR.EQ.0) GO TO 3
    4 WRITE(IW,1) K,L1N,L2N,LABEL,ISEQ
    1 FORMAT(3I5, 10X,27A2,I46)
   57 FORMAT(8XI4,2I2,27A2,I2)
    3 IF(ISEQ) 69,58,69
   58 IF(K) 99, 99, 59
   59 M = IZ + NIMTBL
      NUMPTS(M) = L1N
      M = M + IZ
      NUMPTS(M) = L2N
      IF(NUMTBL-NIMTBL)17,70,17
   17 NUMPTS(NIMTBL) = MUMMY
   70 N1 = (L1N-1) / 9 + 1
      DO 68 IS = 1,N1
      L3 = (IS-1) * 9 + 1
      IF (IS-N1) 60, 61, 60
   60 L4 = L3 + 8
      GO TO 62
   61 L4 = L1N
   62 L5 = NUMPTS(NIMTBL) + 1
      L6 = L5 + L3
      L7 = L5 + L4
      JJ = 0
      LM = L5 + L1N
      LN = LM + L2N
   63 READ(NUNIT,64) (DUMMY(K),K=1,10), ISEQ
   64 FORMAT (10E7.0,I2)
      IF(MCR.EQ.0) GO TO 5
    6 WRITE(IW,2)DUMMY,ISEQ
    2 FORMAT(10E12.4,I5)
    5 XUMPTS(L5)= DUMMY(1)
      K = 2
      DO 65 J = L6,L7
      XUMPTS(J) = DUMMY(K)
   65 K = K+1
      ISQQ=(IS-1)*(L2N+1)+JJ+1
      IF(ISEQ-ISQQ) 69,66,69
   66 L6 = LN + L3
      L7 = LN + L4
```

```
      L5 = LM + 1 + JJ                                          CAB01280
      IF (JJ-L2N) 67, 63, 69                                    CAB01290
   67 JJ = JJ + 1                                               CAB01300
      LN = LN + L1N                                             CAB01310
      GO TO 63                                                  CAB01320
   68 CONTINUE                                                  CAB01330
  109 MUMMY = NUMPTS(NIMTBL) + (L1N+1) * (L2N+1)                CAB01340
  108 NIMTBL = NIMTBL + 1                                       CAB01350
      GO TO 102                                                 CAB01360
   69 NG = 1                                                    CAB01370
   99 RETURN                                                    CAB01380
      END                                                       CAB01390
      SUBROUTINE STINT1(A1,A2,A3,MINTBL,MAXTBL,FCT,NG)          CAB00010
      EQUIVALENCE (X(1),NUMPTS(1))                              CAB00020
      COMMON/TAB1/NUMPTS(1)                                     CAB00030
      DIMENSION X(1)                                            CAB00040
      IZ=NUMPTS(1)/3                                            CAB00050
   70 IF(MINTBL-MAXTBL)71,71,110                                CAB00060
   71 DO 73 II=MINTBL,MAXTBL                                    CAB00070
      NJ=NUMPTS(II)+1                                           CAB00080
      IF(A3-X(NJ))72,74,73                                      CAB00090
   72 IF(II-MINTBL) 110,112,75                                  CAB00100
   73 CONTINUE                                                  CAB00110
      GO TO 112                                                 CAB00120
   75 IK = 1                                                    CAB00130
      IL =2                                                     CAB00140
      NM=NJ                                                     CAB00150
  101 DO 97 IF=IK,IL                                            CAB00160
      NJ =NUMPTS(II)+1                                          CAB00170
      NI = IZ+II                                                CAB00180
      IO =NUMPTS(NI)                                            CAB00190
      IP =IO+NJ                                                 CAB00200
      DO 77 IQ=1,IO                                             CAB00210
      NN= NJ+IQ                                                 CAB00220
      IF (A1-X(NN))76,78,77                                     CAB00230
   76 IF(IQ-1) 110,112,79                                       CAB00240
   77 CONTINUE                                                  CAB00250
      GO TO 112                                                 CAB00260
   78 IG =-1                                                    CAB00270
      GO TO 80                                                  CAB00280
   79 IG =+1                                                    CAB00290
   80 NI=NI+IZ                                                  CAB00300
      IB = NUMPTS(NI)                                           CAB00310
      DO 82 IA=1,IB                                             CAB00320
      NS=IP+IA                                                  CAB00330
      IF (A2-X(NS))81,83,82                                     CAB00340
   81 IF(IA-1) 110,112,84                                       CAB00350
   82 CONTINUE                                                  CAB00360
      GO TO 112                                                 CAB00370
   83 IH =-1                                                    CAB00380
      GO TO 85                                                  CAB00390
   84 IH =+1                                                    CAB00400
      ... +IB+IQ+IO*IA-IO                                       CAB00410
          -IO                                                   CAB00420
           IH) 86,88,91                                         CAB00430
```

```
   86 IF (X(NF)-99998.5E9)87,113,113                                      CAB00440
   87 FCT = X(NE)                                                         CAB00450
      GO TO 95                                                            CAB00460
   88 IF(IG) 89,110,93                                                    CAB00470
   89 IF(AMAX1(X(NE),X(NF))-99998.5E9)90,113,113                          CAB00480
   90 FCT =X(NE)-(X(NS)-A2)*(X(NE)-X(NF))/(X(NS)-X(NS-1))                 CAB00490
      GO TO 95                                                            CAB00500
   91 IF(AMAX1(X(NE),X(NF),X(NE-1),X(NF-1))-99998.5E9)92,113,113          CAB00510
   92 FCT = ((X(NS)-A2)*((X(NN)-A1)*X(NF-1)-(X(NN-1)-A1)*X(NF)            CAB00520
     1)-(X(NS-1)-A2)*((X(NN)-A1)*X(NE-1)-(X(NN-1)-A1)*X(NE)))             CAB00530
     2/((X(NS)-X(NS-1))*(X(NN)-X(NN-1)))                                  CAB00540
      GO TO 95                                                            CAB00550
   93 IF(AMAX1(  X(NE),  X(NE-1))-99998.5E9) 94,113,113                   CAB00560
   94 FCT =   X(NE)-(   X(NN)-A1)*(   X(NE)-   X(NE-1))/(   X(NN)-   X(NN-1))  CAB00570
   95 GO TO (96,98,99),IF                                                 CAB00580
   96 DUMSTG =FCT                                                         CAB00590
   97 II =II-1                                                            CAB00600
   98 FCT =DUMSTG-(   X(NM)-A3)*(DUMSTG-FCT)/(   X(NM)-   X(NJ))          CAB00610
   99 RETURN                                                              CAB00620
   74 IK =3                                                               CAB00630
      IL =3                                                               CAB00640
      GO TO 101                                                           CAB00650
  110 NG =2                                                               CAB00660
      GO TO 99                                                            CAB00670
  112 NG =3                                                               CAB00680
      GO TO 99                                                            CAB00690
  113 NG =4                                                               CAB00700
      GO TO 99                                                            CAB00710
      END                                                                 CAB00720
      SUBROUTINE TABIN1(NUMTBL,NZ,NG)                                     CAB00730
      COMMON/INPUT/IW,IF                                                  CAB00740
      COMMON/TABI/NUMPTS(1)                                               CAB00750
      COMMON /TABOU1/ NIMTBL,ISOO                                         CAB00760
      DIMENSION XUMPTS(1)                                                 CAB00770
      INTEGER*2 LABEL(27)                                                 CAB00780
      EQUIVALENCE (XUMPTS(1),NUMPTS(1)),(DUMMY(1),MUMMY)                  CAB00790
      DIMENSION DUMMY(10)                                                 CAB00800
      MCR=0                                                               CAB00810
   10 IZ=IABS(NZ)                                                         CAB00820
      NUNIT=5                                                             CAB00830
      IF(NZ.LT.0) NUNIT=8                                                 CAB00840
      NIMTBL = NUMTBL                                                     CAB00850
      NG=0                                                                CAB00860
      NUMPTS(1)=IZ+IZ+IZ                                                  CAB00870
  102 READ(NUNIT,57) K,    LIN, L2N, LABEL, ISEQ                          CAB00880
      IF(MCR.EQ.0) GO TO 3                                                CAB00890
    4 WRITE(IW,1) K,LIN,L2N,LABEL,ISEQ                                    CAB00900
    1 FORMAT(3I5, 10X,27A2,I46)                                           CAB00910
   57 FORMAT(9XI4,2I2,27A2,I2)                                            CAB00920
    3 IF(ISEQ) 59,58,59                                                   CAB00930
   58 IF(K) 99, 99, 59                                                    CAB00940
   59 M = IZ + NIMTBL                                                     CAB00950
      NUMPTS(M) = LIN                                                     CAB00960
      M = M + IZ                                                          CAB00970
      NUMPTS(M) = L2N                                                     CAB00980
```

```
      IF(NUMTBL-NIMTBL)17,70,17
   17 NUMPTS(NIMTBL) = MUMMY
   70 N1 = (L1N-1) / 9 + 1
      DO 68 IS = 1,N1
      L3 = (IS-1) * 9 + 1
      IF (IS-N1) 60, 61, 60
   60 L4 = L3 + 8
      GO TO 62
   61 L4 = L1N
   62 L5 = NUMPTS(NIMTBL) + 1
      L6 = L5 + L3
      L7 = L5 + L4
      JJ = 0
      LM = L5 + L1N
      LN = LM + L2N
   63 READ(NUNIT,64) (DUMMY(K),K=1,10), ISEQ
   64 FORMAT (10E7.0,I2)
      IF(MCP.EQ.0) GO TO 5
    6 WRITE(IW,2)DUMMY,ISEQ
    2 FORMAT(10E12.4,I5)
    5 XUMPTS(L5)= DUMMY(1)
      K = 2
      DO 65 J = L6,L7
      XUMPTS(J) = DUMMY(K)
   65 K = K+1
      ISQQ=(IS-1)*(L2N+1)+JJ+1
      IF(ISEQ-ISQQ) 69,66,69
   65 L6 = LN + L3
      L7 = LN + L4
      L5 = LM + 1 + JJ
      IF (JJ-L2N) 67, 68, 69
   67 JJ = JJ + 1
      LN = LN + L1N
      GO TO 63
   68 CONTINUE
  109 MUMMY = NUMPTS(NIMTBL) + (L1N+1) * (L2N+1)
  108 NIMTBL = NIMTBL + 1
      GO TO 102
   69 NG = 1
   99 RETURN
      END
      SUBROUTINE FRICT(IDX)
      COMMON/LAT/AERO(175),AERDP(50),KODE(26)
      COMMON/ROUGH/FRIC(3,6)
      EQUIVALENCE (AERO(96),COU),(AERO(104),CMP)
      DO 1 I=1,3
      DO 1 J=1,6
    1 FRIC(I,J)=0.
      IF(CMP.EQ.0..AND.COU.EQ.0.)RETURN
      IND=KODE(6)
      IF(IDX.NE.0)GO TO 2
C  LONGITUDINAL PULLEY FRICTION COMPUTATION
      GO TO(10,11,12,13),IND
   10 CALL FRVT(1)
      RETURN
```

```
   11 CALL FRVT(3)
      RETURN
   12 CALL FRVT(1)
      CALL FRVT(3)
   13 RETURN
C  LATERAL DIRECTIONAL FRICTION COMPUTATION
    2 GO TO(20,21,22,23),IND
   20 CALL FRHZ(3)
      RETURN
   21 CALL FRHZ(1)
   22 RETURN
   23 CALL FRHZ(1)
      CALL FRHZ(3)
      RETURN
      END
      SUBROUTINE FRVT(IC)
C  COMPUTES THE FRICT. EFFECT OF THE VERT PULLEYS ON THE LONG. DYN.
      COMMON/DAT/AERO(175),AEROP(50),KODE(26)
      COMMON/PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
      COMMON/ROUGH/FRIC(3,6)
      EQUIVALENCE (AERO(90),RVF),(AERO(92),RVR),(AERO(96),COU),
     1(AERO(104),CMP)
      DIMENSION DT1(3),DT2(3)
      IF(IC.EQ.3)GO TO 1
      TENS=TF
      RAD=RVF/12.
      AVX=(ADC(2,1)-ADC(1,1))/2.
      CAX=COS(AVX)
      CAZ=SIN(AVX)
      GO TO 2
    1 TENS=TF
      RAD=RVF/12.
      AVX=3.14159+(ADC(4,1)-ADC(3,1))/2.
      CAX=COS(AVX)
      CAZ=SIN(AVX)
    2 ARMX=(ARM(IC,1)+ARM(IC+1,1))/24.
      ARMZ=(ARM(IC+1,3)-ARM(IC,3))/24.
      ENORX=TENS*COS(ADC(IC,1))
      ENORZ=TENS*(1.+COS(ADC(IC,3)))
      ENORM=SQRT(ENORX**2+ENORZ**2)
      CMPP=CMP/ENORM
      FACU=CMPP*ENORM/RAD**2
      ENORX=TENS*COS(ADC(IC+1,1))
      ENORZ=TENS*(1.+COS(ADC(IC+1,3)))
      ENORM=SQRT(ENORX**2+ENORZ**2)
      CMPP=CMP/ENORM
      FACL=CMPP*ENORM/RAD**2
      FACT=4.*COU/(3.14159*RAD**2)
      CALL DLGTH(CX,CZ,CT,IC,0)
      CALL DLGTH(CXP,CZP,CTP,IC+1,0)
      DT1(1)=FACT *(CXP-CX)
      DT1(2)=FACT *(CZP-CZ)
      DT1(3)=FACT *(CTP-CT)
      DT2(1)=FACL*CXP-FACU*CX
      DT2(2)=FACL*CZP-FACU*CZ
```

```
      DT2(3)=FACL*CTP-FACU*CT
      DO 3 I=1,3
      FRIC(1,I)=FRIC(1,I)+DT1(I)*CAX
      FRIC(1,I+3)=FRIC(1,I+3)+DT2(I)*CAX
      FRIC(2,I)=FRIC(2,I)+DT1(I)*CAZ
      FRIC(2,I+3)=FRIC(2,I+3)+DT2(I)*CAZ
      FRIC(3,I)=FRIC(3,I)+DT1(I)*RAD+DT1(I)*CAX*ARMZ-DT1(I)*CAZ*ARMX
      FRIC(3,I+3)=FRIC(3,I+3)+DT2(I)*RAD+DT2(I)*CAX*ARMZ-DT2(I)*CAZ*ARMX
    3 CONTINUE
      RETURN
      END
      SUBROUTINE FRHZ(IC)
C  COMPUTES THE FRICT. EFFECT OF THE HORZ PULLEYS ON THE LAT. DIR. DYN.
      COMMON/DAT/AERO(175),AEROP(50),KODE(26)
      COMMON/PLYCHA/STD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
      COMMON/ROUGH/FRIC(3,6)
      EQUIVALENCE (AERO(91),RHF),(AERO(93),RHP),(AERO(96),COU),
     1(AERO(104),CMP)
      DIMENSION DT1(3),DT2(3)
      IF(IC.EQ.3)GO TO 1
      TENS=TF
      RAD=RHF/12.
      GO TO 2
    1 TENS=TF
      RAD=RHF/12.
    2 ENORX=TENS*COS(ADC(IC,1))
      ENORY=TENS*(1.+COS(ADC(IC,2)))
      ENORM=SQRT(ENORY*ENORY+ENORX*ENORX)
      CMPP=CMP/ENORM
      FACL=CMPP*ENORM/RAD**2
      FACT=4.*COU/(3.14159*RAD**2)
      CALL DLGTH(CY,CPSI,CPHI,IC,1)
      CALL DLGTH(CYP,CPSIP,CPHIP,IC+1,1)
      DT1(1)=FACT*(CY-CYP)
      DT1(2)=FACT*(CPSI-CPSIP)
      DT1(3)=FACT*(CPHI-CPHIP)
      DT2(1)=FACL*(CY-CYP)
      DT2(2)=FACL*(CPSI-CPSIP)
      DT2(3)=FACL*(CPHI-CPHIP)
      DO 3 I=1,3
      FRIC(1,I)=FRIC(1,I)+DT1(I)*COS(ADC(IC,2))
      FRIC(1,I+3)=FRIC(1,I+3)+DT2(I)*COS(ADC(IC,2))
      FRIC(2,I)=FRIC(2,I)+DT1(I)*RAD-DT1(I)*COS(ADC(IC,1))*ARM(IC,2)
     1/12.+DT1(I)*COS(ADC(IC,2))*ARM(IC,1)/12.
      FRIC(2,I+3)=FRIC(2,I+3)+DT2(I)*RAD-DT2(I)*COS(ADC(IC,1))*ARM(IC,2)
     1/12.+DT2(I)*COS(ADC(IC,2))*ARM(IC,1)/12.
      FRIC(3,I)=FRIC(3,I)+DT1(I)*RAD+DT1(I)*COS(ADC(IC,3))*ARM(IC,2)
     1/12.-ARM(IC,3)/12.*DT1(I)*COS(ADC(IC,2))
      FRIC(3,I+3)=FRIC(3,I+3)+DT2(I)*RAD+DT2(I)*COS(ADC(IC,3))*ARM(IC,2)
     1/12.-ARM(IC,3)/12.*DT2(I)*COS(ADC(IC,2))
    3 CONTINUE
      RETURN
      END
      SUBROUTINE MATRIX(CMAT,N,ROOTS,KAA,IER)
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
```

```
      COMMON/FRQ/C4(30)
      DIMENSION CMAT(14,14,3),MAT(14,14),KOUNT(30),C5(30)
      COMPLEX AMAT(14,14),ROOTS(29)
      DOUBLE PRECISION BMAT(14,14,30),D(30,4)
      NP=-2
      IF(KODE(5).EQ.1) NP=1
      CALL MAPOY(CMAT,C4,ROOTS,K4A,14,NP,3,30,KOUNT,
     1 AMAT,BMAT,MAT,C5,D,N)
      RETURN
      END
      SUBROUTINE MAPOL(CMAT,C4,ROOTS,K4A,MCOL,NP,
     1    IN,  N,  KOUNT,  AMAT,  BMAT,  MAT,  C5,  D)
      COMMON/INOUT/IW,IR
      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  BMAT, SA, F, D(N,1),DR
     *  ,  FBMAT
      COMPLEX  DET,FCMPLX, AMAT,   G,   YA1,   YA,   SCMPLX
      COMPLEX  G3, ROOTS,CF
   12  FORMAT (   2I3, 1P5D16.6/(D22.6,4D16.6))
   14 FORMAT(1H0,2(1PE24.6,E16.6))
   15 FORMAT(1H-,14X,4HREAL,11X,9HIMAGINARY,19X,5HERROR)
   22  FORMAT (   3I3, 1P5E16.6/(E25.6,4E16.6))
 2614  FORMAT (/1PE24.6,E16.6,E30.6)
      DATA CF/Z7FFFFFFFFFFFFFFF/
      NCOL=MCOL
   10 NROW=NCOL
      ENP=10.**NP
      INN=IN+1
       DO 107 I=1,NROW
       DO 107 J=1,NCOL
       MAT(I,J) = 0
      DO 112 K=INN,N
  112 BMAT(I,J,K)=0.D0
       DO 107  K=1,IN
      BMAT(I,J,K)=CMAT(I,J,K)
      IF(CMAT(I,J,K))108,107,108
  108   MAT(I,J) = K
C     THE NUMBER IN MAT IS ONE GREATER THAN THE DEGREE OF THE POLYNOMIAL
  107 CONTINUE
       JS= 1
      IF(NP.LT.0)GO TO 128
         ASSIGN 128 TO MZ
        GO TO 920
   99  ASSIGN 257 TO MZ
  920  WRITE(IW,23)
   23 FORMAT(55H0POSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX)
        DO 951 J9= 1,NCOL
        DO 951 I9= 1,NCOL
        K1 = MAT(I9,J9)
       IF(K1) 951,951,952
  952   WRITE(IW,12)I9,J9, (BMAT(I9,J9,K), K=1,K1)
  951   CONTINUE
      GO TO MZ, (130,257,128,1105)
```

```
C DET CONTAINS VALUE OF DETERMINANT OF BMAT WITH G=1
 1281   WRITE(IW,1282)DET
 1282   FORMAT ( 12H DETERMINANT1P2F16.7)
  128  NC = 0
C  COUNT NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN COLUMN JS
       DO 120 I=JS,NROW
       IF (MAT(I,JS))99,120,121
  121  NC = NC +1
       IS = I
  120  CONTINUE
      IF(NC-1) 17,125,130
   17 WRITE(IW,16)
   16 FORMAT(' MATRIX IS SINGULAR')
      GO TO 257
  125  IF(IS-JS)99,1401,123
C      ONE INTER CHANGE TRIANGULARIZES THE COLUMN.
  123  DO 126 J=JS,NCOL
       KI = MAX0(MAT(IS,J), MAT(JS,J))
       MA = MAT(IS,J)
       MAT(IS,J) = MAT(JS,J)
       MAT(JS,J) = MA
       DO 126 K= 1,KI
          SA   = BMAT(IS,J,K)
       BMAT(IS,J,K) = BMAT(JS,J,K)
  126  BMAT(JS,J,K) =-SA
       GO TO 1401
  130  IS = JS+1
C LOOP 137 REDUCES ALL ELEMENTS BELOW DIAGONAL IN COLUMN JS BY
C AT LEAST ONE DEGREE
      I=IS
  139 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,NCOL
       KI= MAX0(MAT(JS,J), MAT(I,J))
       MA = MAT(JS,J)
       MAT(JS,J) = MAT(I,J)
       MAT(I,J) =  MA
       DO 131 K= 1,KI
       SA = BMAT(I,J,K)
       BMAT(I,J,K) = BMAT(JS,J,K)
  131  BMAT(JS,J,K) =-SA
       GO TO 139
  134  KI = MAT(I,JS)
       KJS = MAT(JS,JS)
       KD  = KI - KJS
       F = BMAT(I,JS,KI)/ BMAT(JS,JS,KJS)
       IF(DABS(F)- 4.0) 1052,1051,1051
 1051  IF(KD) 99,133,1052
 1052 DO 235 J=JS,NCOL
       KJS = MAT(JS,J)
      IF(KJS.EQ.0) GO TO 235
       DO 135 K=  1 ,KJS
       KI = K + KD
      FBMAT=F*BMAT(JS,J,K)
```

```
      IF(KI-N) 141,141,2                                                CBL01130
    2 WRITE(IW,3)                                                       CBL01140
    3 FORMAT(79H0DEGREE OF POLYNOMIAL FORMED WHILE TRIANGULARIZING ORIGICBL01150
     1NAL MATRIX IS TOO HIGH  )                                         CBL01160
      GO TO 257                                                         CBL01170
  141 IF ( DABS ( FBMAT - BMAT(I,J,KI) ) .LE. 2.D-6 * DABS ( FBMAT ) )  CBL01180
     1   GO TO 136                                                      CBL01190
      BMAT(I,J,KI)= BMAT(I,J,KI)- FBMAT                                 CBL01200
      GO TO 135                                                         CBL01210
  136  BMAT(I,J,KI) = 0.D0                                              CBL01220
  135  CONTINUE                                                         CBL01230
  235 CONTINUE                                                          CBL01240
C                                                              DRONTS LLCBL01250
      J=JS                                                              CBL01260
  142 CONTINUE                                                          CBL01270
      KI=MAT(JS,J)+KD                                                   CBL01280
      KJ=MAT(I,J)                                                       CBL01290
      IF(KI.LT.KJ) KI=KJ                                                CBL01300
       MAT(I,J) = 0                                                     CBL01310
       DO 140 K=1,KI                                                    CBL01320
      IF(BMAT(I,J,K))138,140,138                                        CBL01330
  138  MAT(I,J) = K                                                     CBL01340
  140  CONTINUE                                                         CBL01350
      J=J+1                                                             CBL01360
      IF (J.LE.NCOL) GO TO 142                                          CBL01370
  137 I=I+1                                                             CBL01380
      IF(I.LE.NROW) GO TO 139                                           CBL01390
      IF(NP) 128,128,1105                                               CBL01400
 1401  JS = JS +1                                                       CBL01410
       IF( JS-NCOL)128,150,150                                          CBL01420
  150 IF(NP.LT.0)GO TO 153                                              CBL01430
      WRITE(IW,13)                                                      CBL01440
   13  FORMAT ( 1H ,20(1H ),13H FINAL MATRIX)                           CBL01450
       DO 151 J=1,NCOL                                                  CBL01460
       DO 151 I=1,NROW                                                  CBL01470
       K1 = MAT(I,J)                                                    CBL01480
       IF(K1) 99,151,152                                                CBL01490
  152  WRITE(IW,12)I, J, (BMAT(I,J,K), K=1,K1)                          CBL01500
  151  CONTINUE                                                         CBL01510
  153 LK=1                                                              CBL01520
C LOOP 160-ROOTS OF POLYNOMIALS ON DIAGONAL OF TRIANGULARIZED MATRIX ARECBL01530
C           FOUND AND STORED IN ARRAY ROOTS.                            CBL01540
C           COEFFICIENTS OF THE POLYNOMIAL EQUIVALENT OF THE DETERMINANT CBL01550
C           OF THE MATRIX ARE COMPUTED AND STORED IN ARRAY C4 WITH      CBL01560
C           C4(1) THE CONSTANT TERM.                                    CBL01570
       DO 160 J=1,NCOL                                                  CBL01580
       K1 = MAT(J,J)                                                    CBL01590
      MMM=K1+1                                                          CBL01600
      K2=K1-1                                                           CBL01610
         DO 163 K=1,K1                                                  CBL01620
      MM=MMM-K                                                          CBL01630
  163 D(MM,4)= BMAT(J,J,K)                                              CBL01640
      IF(K1.EQ.1) GO TO 1020                                            CBL01650
  161 CALL POLYRT(D(1,4),ROOTS(2*LK-1),KOUNT(LK),K2,D(1,1),D(1,2),D(1,3)CBL01660
     1)                                                                 CBL01670
```

```
      KM1=K1/2
      LK=LK+KM1
      IF(MOD(K1,2).NE.0)GO TO 1020
C DUMMY ELEMENT STORED IN ARRAY ROOTS IF POLYNOMIAL IS OF ODD DEGREE
   54 ROOTS(2*LK-2)=0.
 1020 IF(J.EQ.1)GO TO 1004
 1001   DO 1002 K= 1,K4A
 1002   C5(K) = C4(K)
        DO 1006 K= 1,N
 1006     C4(K) = 0.0
          IF(K1) 99,160,1000
 1000    DO 1003  K=1,K1
      MM=MMM
         DO 1003  K3=1,K4A
        K4 = K+K3-1
 1003 C4(K4) = C4(K4) + D(MM,4)*C5(K3)
         K4A = K4
        GO TO 160
 1004    DO 1005  K= 1,K1
      MM=MMM-K
 1005 C4(K) = D(MM,4)
        K4A = K1
  160  CONTINUE
      CALL JUGGLE(ROOTS,ROOTS,KOUNT,K4A)
        DO 306 I=1,NROW
        DO 306 J=1,NCOL
      MAT(I,J)=IN
      DO 306 K=1,IN
  306 BMAT(I,J,K)=CMAT(I,J,K)
      IF(NP.LT.-1)GO TO 202
  201  WRITE(IW,15)
  202 IF(LK.EQ.1) GO TO 1110
 1111 L=1
   62 G=ROOTS(L)
   64  ASSIGN 244 TO MDT
         GO TO 2511
  244 G1=ABS(C4(1))
C LOOP 2510 - PLACE LARGEST PRODUCT,C4(I)*G**(I-1), IN G1
C G3= ERROR ESTIMATE       G=ROOT
        DO 2610 L9=2,K4A
      G2=CABS(G)
        G2=ABS(C4(L9)*G2**(L9-1))
      IF(G1-G2)2611,2610,2610
 2611 G1=G2
 2610  CONTINUE
C DET CONTAINS VALUE OF POLYNOMIAL EQUIVALENT OF DETERMINANT OF
C MATRIX AT ROOT
      IF(G1.EQ.0.)GO TO 25
      G3=DET/G1
      GO TO 26
   25 G3=(0.,0.)
   26 IF(CABS(G3).LE.END.AND.NP.LT.-1) GO TO 255
      WRITE(IW,27)
   27 FORMAT(5X,'THE FOLLOWING EXTRACTED ROOT HAVE POOR ACCURACY')
      WRITE(IW,15)
```

C-2

```
      WRITE(IW,14) G,G3                                                CBLC2230
  256 L=L+1                                                            CBLC2240
      IF(K4A-1.GE.L)GO TO 62                                           CBLC2250
      IF(NP.LT.-1) GO TO 257                                           CBLC2260
 1110 WRITE(IW,1010)(C4(K),K=1,K4A)                                    CBLC2270
 1010    FORMAT (11HOPOLYNOMIAL1P5E16.6/(E27.6,4E16.6))                CBLC2280
  257 RETURN                                                           CBLC2290
 1105 ASSIGN 1281 TO MDT                                               CBLC2300
      G=(1.,0.)                                                        CBLC2310
C     LOOP 210                                                         CBLC2320
C EVALUATE EACH POLYNOMIAL OF THE ORIGINAL MATRIX FOR ROOT G           CBLC2330
C AND STORE IN AMAT ARRAY                                              CBLC2340
 2511 DO 210 I = 1,NROW                                                CBLC2350
      DO 210 J=1,NCOL                                                  CBLC2360
       K = MAT(I,J)                                                    CBLC2370
      YA=(0.,0.)                                                       CBLC2380
       IF(K-1)  210,205,227                                            CBLC2390
  227 YA=CMPLX(SNGL(BMAT(I,J,K)),0.)                                   CBLC2400
        K = K-1                                                        CBLC2410
  205 YA1=CMPLX(SNGL(BMAT(I,J,K)),0.)                                  CBLC2420
      YA=YA1+YA*G                                                      CBLC2430
       K = K-1                                                         CBLC2440
       IF(K) 99,210,205                                                CBLC2450
  210 AMAT(I,J)=YA                                                     CBLC2460
      JJ=1                                                             CBLC2470
  225  DO 213J=JJ,NCOL                                                 CBLC2480
      IF(CABS(AMAT(JJ,J))) 220,213,220                                 CBLC2490
  213  CONTINUE                                                        CBLC2500
      DET = (0.,0.)                                                    CBLC2510
      GO TO 229                                                        CBLC2520
  220  IF(J-JJ)99,230,221                                              CBLC2530
  221  DO 222 I= 1,NROW                                                CBLC2540
      SCMPLX = AMAT(I,J)                                               CBLC2550
      AMAT(I,J)=AMAT(I,JJ)                                             CBLC2560
  222 AMAT(I,JJ)=-SCMPLX                                               CBLC2570
  230  JS1 = JJ + 1                                                    CBLC2580
       DO 224 I=JS1,NROW                                               CBLC2590
      FCMPLX=AMAT(I,JJ)/AMAT(JJ,JJ)                                    CBLC2600
      IF(CABS(FCMPLX)) 226,224,226                                     CBLC2610
  226  DO 223 J=JJ,NCOL                                                CBLC2620
  223 AMAT(I,J)=AMAT(I,J)-AMAT(JJ,J)*FCMPLX                            CBLC2630
  224     CONTINUE                                                     CBLC2640
      JJ=JJ+1                                                          CBLC2650
      IF(JJ.LT.NCOL) GO TO 225                                         CBLC2660
      DET=(1.,0.)                                                      CBLC2670
       DO 242 J=1,NCOL                                                 CBLC2680
  242 DET=DET*AMAT(J,J)                                                CBLC2690
  229 GO TO MDT, (1281,244,256)                                        CBLC2700
      ENTRY  MAPOY    (CMAT  ,C4    ,ROOTS ,K4A   ,MCOL  ,NP,          CBLC2710
     1,  IN,  N,  KOUNT,  AMAT,  BMAT,  MAT,  C5,  D ,MCO)             CBLC2720
      NCOL=MCO                                                         CBLC2730
      GO TO 10                                                         CBLC2740
      END                                                              CBLC2750
      SUBROUTINE JUGGLE(ROOTS,RT,KOUNT,K4A)                            CBLC2760
      DOUBLE PRECISION ROOTS(1)                                        CBLC2770
```

```
      COMPLEX RT(1)                                                     CBL02780
      REAL*8 CR                                                         CBL02790
      DATA CR/Z7FFFFFFFFFFFFFFF/                                        CBL02800
      DIMENSION KOUNT(1)                                                CBL02810
      K=1                                                               CBL02820
      I=1                                                               CBL02830
    1 IF(KOUNT(I).GE.0)GO TO 3                                          CBL02840
      RT(K) = CMPLX(SNGL(ROOTS(2*I-1)),SNGL(ROOTS(2*I)))                CBL02850
      RT(K+1)=CONJG(RT(K))                                              CBL02860
      K=K+2                                                             CBL02870
      GO TO 5                                                           CBL02880
    3 RT(K)=CMPLX(SNGL(ROOTS(2*I-1)),0.)                                CBL02890
      K=K+1                                                             CBL02900
      IF(ROOTS(2*I).EQ.CR )GO TO 5                                      CBL02910
      RT(K)=CMPLX(SNGL(ROOTS(2*I)),0.)                                  CBL02920
      K=K+1                                                             CBL02930
    5 I=I+1                                                             CBL02940
      IF(K.GE.K4A )RETURN                                               CBL02950
      GO TO 1                                                           CBL02960
      END                                                               CBL02970
      SUBROUTINE POLYRT(AO,ROOT,KOUNT,MM,Q,A,T)                         CBL02980
      DIMENSION KOUNT( 3)                                               CBL02990
      DOUBLE PRECISION AO(5),ROOT(5),Q(5),T(3),A(5)                     CBL03000
      DOUBLE PRECISION B1(1),B2(1),X,Y,D1                               CBL03010
      DOUBLE PRECISION  D2      , DABS    , TOL     , S1      , S2      CBL03020
      COMMON /BARK/ D1,D2,X,NIX                                         CBL03030
      M = MM                                                            CBL03040
   90 IF (AO(M+1)) 100,95,100                                           CBL03050
   95 ROOT(M) = 0.D0                                                    CBL03060
      KOUNT((M+1)/2) = 0                                                CBL03070
      M = M - 1                                                         CBL03080
      GO TO 90                                                          CBL03090
  100 TOL = 1.D5                                                        CBL03100
      IF (M - 1) 460,103,106                                            CBL03110
  103 ROOT(1) = -AO(2) /AO(1)                                           CBL03120
      KOUNT(1) = 0                                                      CBL03130
      GO TO 460                                                         CBL03140
  106 KODE = -1                                                         CBL03150
      N = M                                                             CBL03160
      N1 = N + 1                                                        CBL03170
      K = 0                                                             CBL03180
      DO 110  I = 1,N1                                                  CBL03190
  110 A(I) = AO(I)                                                      CBL03200
      IF(A(N-1))115,112,115                                             CBL03210
  112 B1(1)= 1.D-5                                                      CBL03220
      B2(1)= 1.D-8                                                      CBL03230
      GO TO 120                                                         CBL03240
  115 B2(1)=-A(N+1)/A(N-1)                                              CBL03250
      B1(1)= -B2(1)* (A(N-2)/A(N-1))  -  A(N)/A(N-1)                    CBL03260
  120 IF (N - 2) 121,122,130                                            CBL03270
  121 KOUNT(K+1) = 0                                                    CBL03280
      A(2) =-A(2) / A(1)                                                CBL03290
      GO TO 310                                                         CBL03300
  122 KOUNT(K+1) = 0                                                    CBL03310
      A(2) =-A(2) / A(1)                                                CBL03320
```

```
      A(3) =-A(3) / A(1)
      GO TO 310
  130 CALL GFOWL(T(N-2),Q(N   ))
      ITER8 = 0
      KEY = 30
      INK = 15
      MURDER=20
      LOVE = 4
  220 ITER8 = ITER8 + 1
  230 Q(1) = A(1)
      Q(2) = A(2)   +  B1(1)* Q(1)
      DO 240  J = 3,N1
  240 Q(J) = A(J)   +  B1(1)* Q(J-1)   +  B2(1)* Q(J-2)
      T(1) = Q(1)
      T(2) = Q(2)   +  B1(1)* T(1)
      DO 250  J = 5,N1
  250 T(J-2) = Q(J-2)   +  B1(1)* T(J-3)   +  B2(1)* T(J-4)
      X= B1(1)* T(N-1)   +  B2(1)* T(N-2)
      CALL RUFF (T,Q)
      B1(1)= B1(1)+ D1
      B2(1)= B2(1)+ D2
      IF (KODE) 260,260,280
  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) 265,263,460
  263 KODE = 1
  264 S1 = DABS(D1)
      S2 = DABS(D2)
      GO TO 220
  265 LOVE = LOVE - 1
      IF (LOVE) 220,290,220
  270 IF (ITER8 - KEY) 220,271,271
  271 MURDER = MURDER - 1
      IF (MURDER) 409,285,272
  272 KEY = KEY + INK
      B2(1)=-B2(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) 264,410,410
  285 ITER8 = 999
  290 K = K + 1
      KOUNT(K) = ITER8   +   10
      A(N) = B1(1)
      A(N1) = B2(1)
      N = N - 2
      N1 = N1 - 2
      DO 300   I = 1,N1
  300 A(I) = Q(I)
      IF(DABS(B1(1)).LT..1D0*DSQRT(DABS(B2(1))))
     1B1(1)=.1D0*DSQRT(DABS(B2(1)))
      GO TO 120
  310 DO 320   I = 1,M
      X = A0(I+1)
      A0(I+1) = A(I+1)
  320 A(I+1) = X
```

```
      MURDER = -1                                                       CBLC3880
      N = M                                                             CBL03890
      N1 = N + 1                                                        CBL03900
      L = N                                                             CBL03910
      K = 0                                                             CBL03920
      CALL GROWL(T(N-2),Q(N))                                           CBL03930
  330 IF (L - 1) 440,340,400                                            CBL03940
  340 ITER8 = 0                                                         CBL03950
      Q(1) = A(1)                                                       CBLC3960
      B1(1)= A0(2)                                                      CBL03970
  350 ITER8 = ITER8 + 1                                                 CBLC3980
      DO 360  J = 2,N1                                                  CBL03990
  360 Q(J) = A(J)   +  B1(1)* Q(J-1)                                    CBL04000
      T(1) = Q(1)                                                       CBL04010
      DO 370  J = 3,N1                                                  CBL04020
  370 T(J-1) = Q(J-1)   +  B1(1)* T(J-2)                                CBLC4030
      D1 = Q(N1) / T(N)                                                 CBLC4C40
      B1(1)= B1(1)+ D1                                                  CBL04050
      IF (DABS(B1(1))  -  TOL* DABS(D1)) 380,390,390                    CBLC4060
  380 IF (ITER8 - 8) 350,385,350                                        CBLC4070
  385 ITER8 = 9                                                         CBL04080
  390 KOUNT(K+1) = ITER8                                                CBLC4090
      A0(2) = B1(1)                                                     CBLC4100
      GO TO 440                                                         CBL04110
  400 K = K + 1                                                         CBL04120
      KODE = 0                                                          CBL04130
      B1(1)= A0(L)                                                      CBLC4140
      B2(1)= A0(L+1)                                                    CBLC4150
      ITER8 = KOUNT(K)                                                  CBL04160
      KEY = ITER8 + 8                                                   CBL04170
      IF (M - 2) 220,409,220                                            CBLC4180
  409 ITER8 = ITER8 + 1                                                 CBLC4190
  410 X= B1(1)**2  +  4.D0* B2(1)                                       CBL04200
      IF (X) 420,430,430                                                CBL04210
  420 A0(L) = .5D0* DSQRT(-X)                                           CBLC4220
      A0(L+1) = .5D0* B1(1)                                             CBLC4230
      KOUNT(K) = -ITER8                                                 CBL04240
      L = L - 2                                                         CBLC4250
      GO TO 330                                                         CBLC4260
  430 X = DSQRT(X)                                                      CBL04270
      IF (B1(1)) 432,431,431                                            CBLC4280
  432 X = -X                                                            CBL04290
  431 A0(L) = .5D0* (B1(1)+ X)                                          CBLC4300
      A0(L+1) = -B2(1)/ A0(L)                                           CBL04310
  433 KOUNT(K) = ITER8                                                  CBLC4320
      L = L - 2                                                         CBLC4330
      GO TO 330                                                         CBLC4340
  440 J = N1                                                            CBLC4350
      DO 450  I = 1,N                                                   CBLC4360
      ROOT(I) = A0(J)                                                   CBLC4370
      A0(J) = A(J)                                                      CBL04380
  450 J = J - 1                                                         CBL04390
  460 RETURN                                                            CBLC4400
      END                                                               CBLC4410
      SUBROUTINE GROWL(A,Y)                                             CBLC4420
```

```
      DOUBLE PRECISION A(2),B,X(2),Y(2),T
      COMMON /BARK/ X,B,NIX
      RETURN
      ENTRY RUFF
      NIX = 0
      IF (ABS(SNGL(B)) - 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 PROG-BASIC LONG CHAR. W COMP PRT OUT
  1 -1  0  0  1  2  0  4  3  0  0  0  1
0.          0.          0.          0.          5.80        -1.86
0.          10.95       -14.68      .0500       .0824       .033
0.          .935        -1.630      0.          0.          -7.77
-1.062      -.1109      .1227       -.0341      -.1863      .0231
.3780       .0824       -.1132      .1923       .0091       -.0681
.2380       -.1162      -.0714      .0005       -.0005      -.0001
1.1         0.          4.          0.          .865        446.
4.72        .000805     152.0       9.16        1.4         11.5
-.8         3.60        21.4        22.95
0.          0.          96.         -96.        -5.         -5.
75.0        263.        96.         0.          -5.         185.
0.          0.          6.          27.8        0.          0.
0.          .9          .9          4.0         0.
.01         .00         .01         .00         100.0       40.
0.          181.        96.         152.        6.667
4.          -5.8        0.          2.          3.          2.
2.          3.          2.          180.        96.         72.
180.        -96.        72.         80.         80.         50.
50.         5.          50.         0.          0.          0.
0.          0.          0.          0.
13.8        1.53        .2374       7.0         .022        3.0
.00         00.         3.          0.00        0.          0.
0.          0.          .0          0.0         0.000       0.
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
  SAMPLE DATA-LONG CHAR OF THETA/EMO TRANS FUNC W FEEDBACK & FREQ RESP.
  1 -1  2  0  0  2  0 10  3  0  0  0  1 11  2  0  0 -1 60
137 7.5
138 -100.
140 -100.

  SAMPLE INPUT OF VEL=0. W LIFT CABLE -CHAR. ROOTS OPTION
  1 -1  0  0  0  2  0  9  3  0  1  0  1  0  8  0  0
 48 0.
 49 0.

  SAMPLE INPUT FOR CABLELESS MODEL W TRANSFER FUNCTION OPTION
  1  1  0  0  0  2  0  3 10  0  0  0 -1 15  3
 48 .865
 49 446.

 SAMPLE OF ACTIVE CABLE SYSTEM-LAT DIR MODE W TRANS. FUNC. OP.
  1  0  0  0  0  2  0 10 10  0  0  0  1        11  2
```

## Appendix E

## Sample Output

```
CASE NO= 1      SAMPLE INPUT FOR ACTIVE CABLE PROG-BASIC LONG CHAR. W COMP PRT OUT
                  FRONT CABLE HORIZONTAL,REAR CABLE VERTICAL
                  NO SNUBBERS
                  NO LIFT/ANTI-LIFT CABLE
                  FEEDBACK LOGIC IN

CODE NOS. FOR THIS CASE.
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
   1  -1   0   0   1   2   0   4   3   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
  INPUT DATA AS SPECIFIED IN AERO ARRAY
 AERO(  1)= 0.0          AERO(  2)= 0.0          AERO(  3)= 0.0          AERO(  4)= 0.0          AERO(  5)=  5.80
 AERO(  6)= -1.86        AERO(  7)= 0.0          AERO(  8)=  10.9        AERO(  9)= -14.7        AERO( 10)= 0.500E-01
 AERO( 11)= 0.924E-01    AERO( 12)= 0.330E-01    AERO( 13)= 0.0          AERO( 14)= 0.935        AERO( 15)= -1.63
 AERO( 16)= 0.0          AERO( 17)= 0.0          AERO( 18)= -7.77        AERO( 19)= -1.06        AERO( 20)=-0.111
 AERO( 21)= 0.123        AERO( 22)=-0.341E-01    AERO( 23)=-0.186        AERO( 24)= 0.231E-01    AERO( 25)= 0.378
 AERO( 26)= 0.924E-01    AERO( 27)=-0.113        AERO( 28)= 0.192        AERO( 29)= 0.910E-02    AERO( 30)=-0.681E-01
 AERO( 31)= 0.238        AERO( 32)=-0.116        AERO( 33)=-0.714E-01    AERO( 34)= 0.500E-03    AERO( 35)=-0.500E-03
 AERO( 36)=-0.100E-03    AERO( 37)= 0.0          AERO( 38)= 0.0          AERO( 39)= 0.0          AERO( 40)= 0.0
 AERO( 41)= 0.0          AERO( 42)= 0.0          AERO( 43)= 0.0          AERO( 44)=  1.10        AERO( 45)= 0.0
 AERO( 46)=  4.00        AERO( 47)= 0.0          AERO( 48)= 0.865        AERO( 49)=  446.        AERO( 50)=  4.72
 AERO( 51)= 0.905E-03    AERO( 52)=  152.        AERO( 53)=  9.16        AERO( 54)=  1.40        AERO( 55)=  11.5
 AERO( 56)=-0.900        AERO( 57)=  3.60        AERO( 58)=  21.4        AERO( 59)=  22.9        AERO( 60)= 0.0
 AERO( 61)= 0.0          AERO( 62)= 0.0          AERO( 63)= 0.0          AERO( 64)= 0.0          AERO( 65)= 0.0
 AERO( 66)= 0.0          AERO( 67)= 0.0          AERO( 68)=  96.0        AERO( 69)= -96.0        AERO( 70)= -5.00
 AERO( 71)= -5.00        AERO( 72)=  75.0        AERO( 73)=  263.        AERO( 74)=  96.0        AERO( 75)= 0.0
 AERO( 76)= -5.00        AERO( 77)=  185.        AERO( 78)= 0.0          AERO( 79)= 0.0          AERO( 80)=  6.00
 AERO( 81)=  27.8        AERO( 82)= 0.0          AERO( 83)= 0.0          AERO( 84)= 0.0          AERO( 85)= 0.900
 AERO( 86)= 0.900        AERO( 87)=  4.00        AERO( 88)= 0.0          AERO( 89)= 0.0          AERO( 90)= 0.100E-01
 AERO( 91)= 0.0          AERO( 92)= 0.100E-01    AERO( 93)= 0.0          AERO( 94)=  100.        AERO( 95)=  40.0
 AERO( 96)= 0.0          AERO( 97)=  181.        AERO( 98)=  96.0        AERO( 99)=  152.        AERO(100)=  6.67
 AERO(101)= 0.0          AERO(102)=  4.00        AERO(103)= -5.80        AERO(104)= 0.0          AERO(105)=  2.00
 AERO(106)=  3.00        AERO(107)=  2.00        AERO(108)=  2.00        AERO(109)=  3.00        AERO(110)=  2.00
 AERO(111)=  180.        AERO(112)=  96.0        AERO(113)=  72.0        AERO(114)=  180.        AERO(115)= -96.0
 AERO(116)=  72.0        AERO(117)=  80.0        AERO(118)=  80.0        AERO(119)=  50.0        AERO(120)=  50.0
 AERO(121)=  5.00        AERO(122)=  50.0        AERO(123)= 0.0          AERO(124)= 0.0          AERO(125)= 0.0
 AERO(126)= 0.0          AERO(127)= 0.0          AERO(128)= 0.0          AERO(129)= 0.0          AERO(130)= 0.0
 AERO(131)=  13.8        AERO(132)=  1.53        AERO(133)= 0.237        AERO(134)=  7.00        AERO(135)= 0.220E-01
 AERO(136)=  3.00        AERO(137)= 0.0          AERO(138)= 0.0          AERO(139)=  3.00        AERO(140)= 0.0
 AERO(141)= 0.0          AERO(142)= 0.0          AERO(143)= 0.0          AERO(144)= 0.0          AERO(145)= 0.0
 AERO(146)= 0.0          AERO(147)= 0.0          AERO(148)= 0.0          AERO(149)= 0.0          AERO(150)= 0.0
 AERO(151)= 0.0          AERO(152)= 0.0          AERO(153)= 0.0          AERO(154)= 0.0          AERO(155)= 0.0
 AERO(156)= 0.0          AERO(157)= 0.0          AERO(158)= 0.0          AERO(159)= 0.0          AERO(160)= 0.0
  AERO DATA IN STAB. AXIS AT EQUAT. REF. CENTER
 AERO(  1)= 0.0          AERO(  2)= 0.0          AERO(  3)= 0.0          AERO(  4)= 0.0          AERO(  5)=  5.80
 AERO(  6)= -1.43        AERO(  7)= 0.0          AERO(  8)=  10.2        AERO(  9)= -13.8        AERO( 10)= 0.500E-01
 AERO( 11)= 0.924E-01    AERO( 12)= 0.394E-01    AERO( 13)= 0.0          AERO( 14)= 0.935        AERO( 15)= -1.57
 AERO( 16)= 0.0          AERO( 17)= 0.0          AERO( 18)= -7.77        AERO( 19)= -1.06        AERO( 20)=-0.111
 AERO( 21)= 0.112        AERO( 22)=-0.341E-01    AERO( 23)=-0.186        AERO( 24)= 0.228E-01    AERO( 25)= 0.357
 AERO( 26)= 0.813E-01    AERO( 27)=-0.107        AERO( 28)= 0.192        AERO( 29)= 0.910E-02    AERO( 30)=-0.682E-01
 AERO( 31)= 0.239        AERO( 32)=-0.116        AERO( 33)=-0.690E-01    AERO( 34)= 0.500E-03    AERO( 35)=-0.500E-03
 AERO( 36)=-0.950E-04    AERO(
CABLE CONFIGURATION ON MODEL
FRONT CABLE IS HORIZNTL AND REAR CABLE IS VERTICAL
CABLE GEOMETRY-CABLE NO. 1     CABLE LENGTH=   0.123377E 03 IN
    DIR. COS.=DEG    ARM-IN
     0.492327E 02    0.278000E 02
     0.417822E 02    0.400000E 01
     0.890904E 02    0.0

CABLE GEOMETRY-CABLE NO. 2     CABLE LENGTH=   0.123377E 03 IN
    DIR. COS.=DEG    ARM-IN
```

```
      -0.482327E 02   0.278000E 02
       0.139218E 03  -0.400000E 01
       0.893804E 02   0.0


  CABLE GEOMETRY-CABLE NO. 3      CABLE LENGTH=   0.123383E 03 IN
       DIR. COS.=DEG    ARM-IN
        0.233272E 03  -0.600801E 01
       -0.899999E 02   0.0
       -0.143272E 03  -0.905980E 00


  CABLE GEOMETRY-CABLE NO. 4      CABLE LENGTH=   0.115261E 03 IN
       DIR. COS.=DEG    ARM-IN
        0.129705E 03  -0.600769E 01
       -0.899999E 02   0.0
       -0.397053E 02   0.906388E 00


  ITERATION PARAMETER =      4
   ACCZ   =   -.3954200E-03
   ACCX   =   .9796825E-03
   THEDOT=-0.24915906E-03 RAD/SEC

  EM. ATT..DEFLTN.& CABLE TENSION
   THETA =  1.03 DEG
   DELTA = -1.37 DEG
   FRT CAB. TENSION=   0.127591E 03 LBS
   RR CAB. TENSION =   0.100214E 03 LBS
     AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER
   AEROP(  1)=-0.985E-01  AEROP(  2)=-0.270     AEROP(  3)= 0.266E-01  AEROP(  4)= 0.290      AEROP(  5)= -5.85
   AEROP(  6)= -1.48      AEROP(  7)= 0.183     AEROP(  8)= -10.2      AEROP(  9)= -13.8      AEROP( 10)=-0.515E-01
   AEROP( 11)=-0.815E-01  AEROP( 12)= 0.384E-01 AEROP( 13)= 0.168E-01  AEROP( 14)=-0.935     AEROP( 15)= -1.57
   AEROP( 16)= 0.0        AEROP( 17)= 0.0       AEROP( 18)= -7.77      AEROP( 19)= -1.06      AEROP( 20)=-0.113
   AEROP( 21)= 0.110      AEROP( 22)=-0.405E-01 AEROP( 23)=-0.188      AEROP( 24)= 0.213E-01  AEROP( 25)= 0.356
   AEROP( 26)= 0.738E-01  AEROP( 27)=-0.105     AEROP( 28)= 0.192      AEROP( 29)= 0.103E-01  AEROP( 30)=-0.660E-01
   AEROP( 31)= 0.238      AEROP( 32)=-0.115     AEROP( 33)=-0.711E-01  AEROP( 34)= 0.500E-03  AEROP( 35)=-0.498E-03
   AEROP( 36)=-0.104E-03  AEROP(
   **** LONGITUDINAL STABILITY ****

  POSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX
   1  1      1.8941760 01   -5.9931590-01
   2  1      3.3059370 01    1.2073010 01    4.7199990 00
   3  1     -5.2585630 01    4.2774080 00   -1.5380850 00
   4  1      2.4095160-02
   1  2     -1.0142810 02   -5.9438310-04
   2  2      5.2885830 03    3.3013000-02   -1.5733320 00
   3  2      2.5107020 03    4.3576570 01    2.1924420 01
   4  2     -5.5822390-02
   1  3     -1.3322140 00
   2  3     -3.2099920-02
   3  3      7.4264720-02
   1  4      7.4674460 02    2.0334480-01    4.7199990 00
   2  4      1.9539790 01    5.5767700-01
   3  4      1.1631300 01   -7.7013730-02
   4  4      1.0000000 00
  DETERMINANT  -5.2237613E 05    0.0
  DETERMINANT  -5.2237513E 05    0.0
  DETERMINANT  -5.2237606E 05    0.0
  DETERMINANT  -5.2237594E 05    0.0
  DETERMINANT  -5.2237591E 05    0.0
  DETERMINANT  -5.2237600E 05    0.0
```

```
DETERMINANT  -5.2237631E 05   0.0
DETERMINANT  -5.2237656E 05   0.0
DETERMINANT  -5.2237656E 05   0.0
DETERMINANT  -5.2237673E 05   0.0
DETERMINANT  -5.2237638E 05   0.0
                         FINAL MATRIX
 1  1   -1.9941760 01
 1  2    1.0142810 02    1.3890080 00
 2  2   -7.5621330 03
 1  3    1.3322140 00
 2  3   -7.1607840 00    2.2080230-01
 3  3    3.2953250 00
 1  4   -7.4874460 02   -2.5076220 01   -4.7199990 00
 2  4    4.0577710 03    8.5925760 02    4.1839170 02   -7.8229660-01
 3  4   -1.8852030 03   -9.8791580 01   -1.2470600 02   -2.5457170 00   -5.7921880-01

 4  4   -1.0192070 00   -2.6078880-02   -5.9759080-02   -1.3476300-03   -2.8554620-04
```

```
                REAL              IMAGINARY                  ERROR

           -8.398477E-03     4.329021E 00          -2.922648E-07   -2.386554E-07

           -8.398477E-03    -4.329021E 00          -2.922648E-07    2.386554E-07

           -2.351342E 00     1.359896E 01          -9.908530E-08    4.166396E-07

           -2.351342E 00    -1.359896E 01          -9.908530E-08   -4.166396E-07

POLYNOMIAL   -4.910885E 05   -1.230982E 04   -2.820763E 04   -6.361116E 02   -1.347842E 02

 POLYNOMIAL W CONST TERM FIRST
             -0.481087E 06   -0.123098E 05   -0.282076E 05   -0.636112E 03   -0.134784E 03
   REAL        IMAGINARY      T H/D-SEC     1/T H/D     PERIOD-SEC    DNATF-CPS    UNDNAT-CPS    DAMP RATIO     DECAY RATIO
-0.8398E-02  +-0.4329E 01   0.8253E 02   0.1212E-01   0.1451E 01   0.6890E 00   0.6890E 00   0.1941E-02   0.9879E 00
-0.2351E 01  +-0.1360E 02   0.2948E 00   0.3392E 01   0.4620E 00   0.2164E 01   0.2196E 01   0.1704E 00   0.3374E 00
```

CASE NO= 2 SAMPLE DATA-LONG CHAR OF THETA/EMO TRANS FUNC W FEEDBACK & FREQ RESP.
FRONT CABLE HORIZONTAL,REAR CABLE VERTICAL
NO SNUBBERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

CODE NOS. 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 | 11 | 2 | 0 | 0 | -1 | 60 | 0 | 0 | 0 | 0 | 0 |

DATA CHANGE
137 7.5000
138 [illegible]00.00
140 [illegible].00
FREQUENCY RESPONSE COMPUTATION

EM. ATT.,DEFLTN.& CABLE TENSION
THETA = 1.03 DEG
DELTA = -1.33 DEG
FRT CAB. TENSION= 0.127591E 03 LBS
RR CAB. TENSION = 0.100214E 03 LBS
++++ LONGITUDINAL STABILITY ++++
COMPUTATION OF THETA/ EMO NUMERATOR ROOTS
POLYNOMIAL W CONST TERM FIRST
0.969847E 03 0.153241E 02 0.523025E 02

| REAL | IMAGINARY | T H/D-SEC | 1/T H/D | PERIOD-SEC | DNATF-CPS | UNDNAT-CPS | DAMP RATIO | DECAY RATIO |
|---|---|---|---|---|---|---|---|---|
| -0.1465E 00 | +-0.4304E 01 | 0.4732E 01 | 0.2113E 00 | 0.1460E 01 | 0.6850E 00 | 0.6853E 00 | 0.3402E-01 | 0.8074E 00 |

COMPUTATION OF THE DENOMINATOR ROOTS

POLYNOMIAL W CONST TERM FIRST
0.603159E 07 0.391416E 06 0.399395E 06 0.718007E 04 0.150616E 04
0.465812E 01

| REAL | IMAGINARY | T H/D-SEC | 1/T H/D | PERIOD-SEC | DNATF-CPS | UNDNAT-CPS | DAMP RATIO | DECAY RATIO |
|---|---|---|---|---|---|---|---|---|
| -0.3986E 00 | +-0.4017E 01 | 0.1739E 01 | 0.5751E 00 | 0.1564E 01 | 0.6394E 00 | 0.6425E 00 | 0.9875E-01 | 0.5361E 00 |
| -0.1596E 01 | +-0.1569E 02 | 0.4344E 00 | 0.2302E 01 | 0.4003E 00 | 0.2498E 01 | 0.2511E 01 | 0.1011E 00 | 0.5279E 00 |
| -0.3194E 03 | | 0.2170E-02 | 0.4607E 03 | | | | | |

FREQUENCY RESPONSE OF THE THETA EMO TRANSFER FUNCTION
STEADY STATE GAIN = 0.1607E-03

| FREQ(RPS) | AMP RAT(DB) | PHASE(DEG) | AMP. VALUE |
|---|---|---|---|
| 0.1000E 00 | -0.7588E 02 | -0.2813E 00 | 0.1608E-03 |
| 0.1200E 00 | -0.7588E 02 | -0.3376E 00 | 0.1608E-03 |
| 0.1500E 00 | -0.7588E 02 | -0.4222E 00 | 0.1608E-03 |
| 0.1700E 00 | -0.7587E 02 | -0.4786E 00 | 0.1608E-03 |
| 0.2000E 00 | -0.7587E 02 | -0.5633E 00 | 0.1608E-03 |
| 0.2500E 00 | -0.7587E 02 | -0.7049E 00 | 0.1608E-03 |
| 0.3000E 00 | -0.7587E 02 | -0.8468E 00 | 0.1609E-03 |
| 0.3500E 00 | -0.7587E 02 | -0.9895E 00 | 0.1609E-03 |
| 0.4000E 00 | -0.7586E 02 | -0.1133E 01 | 0.1610E-03 |
| 0.4500E 00 | -0.7586E 02 | -0.1277E 01 | 0.1611E-03 |
| 0.5000E 00 | -0.7585E 02 | -0.1421E 01 | 0.1612E-03 |
| 0.5500E 00 | -0.7585E 02 | -0.1567E 01 | 0.1612E-03 |
| 0.6000E 00 | -0.7584E 02 | -0.1714E 01 | 0.1613E-03 |
| 0.6500E 00 | -0.7584E 02 | -0.1862E 01 | 0.1615E-03 |
| 0.7000E 00 | -0.7583E 02 | -0.2012E 01 | 0.1616E-03 |
| 0.7500E 00 | -0.7583E 02 | -0.2163E 01 | 0.1617E-03 |
| 0.8000E 00 | -0.7582E 02 | -0.2315E 01 | 0.1618E-03 |
| 0.8500E 00 | -0.7581E 02 | -0.2470E 01 | 0.1620E-03 |
| 0.9000E 00 | -0.7580E 02 | -0.2625E 01 | 0.1621E-03 |
| 0.9500E 00 | -0.7579E 02 | -0.2783E 01 | 0.1623E-03 |
| 0.1000E 01 | -0.7579E 02 | -0.2944E 01 | 0.1625E-03 |
| 0.1200E 01 | -0.7574E 02 | -0.3609E 01 | 0.1632E-03 |
| 0.1500E 01 | -0.7566E 02 | -0.4702E 01 | 0.1648E-03 |
| 0.1700E 01 | -0.7559E 02 | -0.5518E 01 | 0.1661E-03 |
| 0.2000E 01 | -0.7547E 02 | -0.6932E 01 | 0.1685E-03 |
| 0.2500E 01 | -0.7518E 02 | -0.1014E 02 | 0.1741E-03 |
| 0.3000E 01 | -0.7476E 02 | -0.1571E 02 | 0.1828E-03 |
| 0.3500E 01 | -0.7431E 02 | -0.2865E 02 | 0.1925E-03 |
| 0.4000E 01 | -0.7760E 02 | -0.6377E 02 | 0.1318E-03 |
| 0.4500E 01 | -0.8415E 02 | 0.1521E 00 | 0.6202E-04 |

| FREQ(RPS) | AMP RT(DB) | PHASE(DEG) | AMP. VALUE |
|---|---|---|---|
| 0.5000E 01 | -0.7930E 02 | 0.6857E 01 | 0.1084E-03 |
| 0.5500E 01 | -0.7775E 02 | 0.4034E 01 | 0.1295E-03 |
| 0.6000E 01 | -0.7693E 02 | 0.1681E 01 | 0.1423E-03 |
| 0.6500E 01 | -0.7636E 02 | -0.1988E 00 | 0.1521E-03 |
| 0.7000E 01 | -0.7587E 02 | -0.1800E 01 | 0.1608E-03 |
| 0.7500E 01 | -0.7543E 02 | -0.3251E 01 | 0.1693E-03 |
| 0.8000E 01 | -0.7498E 02 | -0.4634E 01 | 0.1781E-03 |
| 0.8500E 01 | -0.7453E 02 | -0.6005E 01 | 0.1876E-03 |
| 0.9000E 01 | -0.7406E 02 | -0.7413E 01 | 0.1982E-03 |
| 0.9500E 01 | -0.7355E 02 | -0.8900E 01 | 0.2101E-03 |
| 0.1000E 02 | -0.7300E 02 | -0.1051E 02 | 0.2238E-03 |
| 0.1200E 02 | -0.7022E 02 | -0.1953E 02 | 0.3085E-03 |
| 0.1500E 02 | -0.6376E 02 | -0.6413E 02 | 0.6497E-03 |
| 0.1700E 02 | -0.6576E 02 | -0.1278E 03 | 0.5154E-03 |
| 0.2000E 02 | -0.7345E 02 | -0.1592E 03 | 0.2125E-03 |
| 0.2500E 02 | -0.8084E 02 | -0.1713E 03 | 0.9078E-04 |
| 0.3000E 02 | -0.8551E 02 | -0.1760E 03 | 0.5304E-04 |
| 0.3500E 02 | -0.8900E 02 | -0.1789E 03 | 0.3550E-04 |
| 0.4000E 02 | -0.9181E 02 | -0.1810E 03 | 0.2566E-04 |
| 0.4500E 02 | -0.9419E 02 | -0.1829E 03 | 0.1951E-04 |
| 0.5000E 02 | -0.9626E 02 | -0.1843E 03 | 0.1538E-04 |
| 0.5500E 02 | -0.9810E 02 | -0.1856E 03 | 0.1245E-04 |
| 0.6000E 02 | -0.9975E 02 | -0.1869E 03 | 0.1029E-04 |
| 0.6500E 02 | -0.1013E 03 | -0.1881E 03 | 0.8648E-05 |
| 0.7000E 02 | -0.1026E 03 | -0.1892E 03 | 0.7372E-05 |
| 0.7500E 02 | -0.1039E 03 | -0.1903E 03 | 0.6358E-05 |
| 0.8000E 02 | -0.1051E 03 | -0.1913E 03 | 0.5538E-05 |
| 0.8500E 02 | -0.1063E 03 | -0.1923E 03 | 0.4865E-05 |
| 0.9000E 02 | -0.1073E 03 | -0.1933E 03 | 0.4306E-05 |
| 0.9500E 02 | -0.1083E 03 | -0.1943E 03 | 0.3837E-05 |
| 0.1000E 03 | -0.1093E 03 | -0.1952E 03 | 0.3438E-05 |

```
CASE NO= 3      SAMPLE INPUT OF VEL=0. W LIFT CABLE -CHAR. ROOTS OPTION
                FRONT CABLE HORIZONTAL,REAR CABLE VERTICAL
                NO SNUBBERS
                LIFT/ANTI-LIFT CABLE IN
                FEEDBACK LOGIC IN
                WIND OFF CHARACTERISTICS

CODE NOS. FOR THIS CASE.
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
  1  -1   0   0   0   2   0   9   3   0   1   0   1   0   8   0   0   0   0   0   0   0   0   0
 DATA CHANGE
  48     0.0
  49     0.0

EH. ATT..DEFLTN.& CABLE TENSION
 THETA = -0.00 DEG
 DELTA =  0.0  DEG
 FRT CAB. TENSION=   0.906780E 02 LBS
 RR CAB. TENSION =   0.100000E 03 LBS
 **** LONGITUDINAL STABILITY ****
 POLYNOMIAL W CONST TERM FIRST
            0.736854E 06   0.231583E 04   0.650357E 05   0.204398E 03   0.141327E 04
            0.444172E 01
   REAL        IMAGINARY      T H/D-SEC    1/T H/D     PERIOD-SEC   DNATF-CPS   UNDNAT-CPS  DAMP RATIO   DECAY RATIO
-0.3182E 03                   0.2178E-02   0.4590E 03
 0.0          +-0.4492E 01    0.1000E 06   0.0         0.1399E 01   0.7149E 00  0.7149E 00  0.0          0.1000E 01
 0.0          +-0.5084E 01    0.1000E 06   0.0         0.1236E 01   0.8091E 00  0.8091E 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 NOS. FOR THIS CASE.
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
  1   1   0   0   0   2   0   3  10   0   0   0  -1  15   3   0   0   0   0   0   0   0   0   0
 DATA CHANGE
  44     0.86500
          446.00

EM. ATT..DEFLTN.& CABLE TENSION
 THETA =  1.03 DEG
 DELTA = -1.33 DEG
 FRT CAB. TENSION=   0.127591E 03 LBS
 RR CAB. TENSION =   0.100214E 03 LBS
 **** LONGITUDINAL STABILITY ****
 COMPUTATION OF    X /DELE NUMERATOR ROOTS
 POLYNOMIAL W CONST TERM FIRST
          0.0              0.315421E 07  -0.999774E 06  -0.582850E 04   0.156621E 04
   REAL        IMAGINARY      T H/D-SEC     1/T H/D     PERIOD-SEC    DNATF-CPS   UNDNAT-CPS   DAMP RATIO    DECAY RATIO
 0.0                          0.1000E 06   0.0
 0.3146E 01                   0.2203E 00   0.4539E 01
-0.2502E 02                   0.2771E-01   0.3609E 02
 0.2559E 02                   0.2709E-01   0.3692E 02
 **** LATERAL/DIRECTIONAL STABILITY ****
    KODE(3) HAS BEEN SET BY PROG TO 3 FOR CABLELESS MODEL CHARACTERISTICS
    THE FOLLOWING EXTRACTED ROOT HAVE POOR ACCURACY


          REAL              IMAGINARY                          ERROR

       -2.673369E-03      0.0                       7.894731E-02     0.0
 POLYNOMIAL W CONST TERM FIRST
          0.0              0.0             0.255240E 03    0.955461E 05    0.265519E 05
          0.217112E 04     0.386945E 03
   REAL        IMAGINARY      T H/D-SEC     1/T H/D     PERIOD-SEC    DNATF-CPS   UNDNAT-CPS   DAMP RATIO    DECAY RATIO
 0.0                          0.1000E 06   0.0
-0.2673E-02                   0.2593E 03   0.3857E-02
-0.3377E 01                   0.1745E 00   0.5732E 01
-0.8177E 00   +-0.7834E 01    0.8477E 00   0.1180E 01   0.8016E 00   0.1247E 01   0.1264E 01   0.1038E 00   0.5192E 00
 0.0                          0.1000E 06   0.0
```

CASE NO= 5 SAMPLE OF ACTIVE CABLE SYSTEM-LAT DIR MODE W TRANS. FUNC. OP.
FRONT CABLE HORIZONTAL,REAR CABLE VERTICAL
NO SNUBBERS
NO LIFT/ANTI-LIFT CABLE
FEEDBACK LOGIC IN

CODE NOS. 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 | 0 | 0 | 0 | 0 | 2 | 0 | 10 | 10 | 0 | 0 | 0 | 1 | 0 | 0 | 11 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DATA CHANGE
0 0.0

EH. ATT.,DEFLTN,& CABLE TENSION
THETA = 1.03 DEG
DELTA = -1.33 DEG
FRT CAB. TENSION= 0.127591E 03 LBS
RR CAB. TENSION = 0.100214E 03 LBS
++++ LATERAL/DIRECTIONAL STABILITY ++++
COMPUTATION OF PSI/ EMO NUMERATOR ROOTS
POLYNOMIAL W CONST TERM FIRST
-0.345004E 05 -0.179903E 05 -0.904558E 04 -0.159327E 04 -0.294450E 03

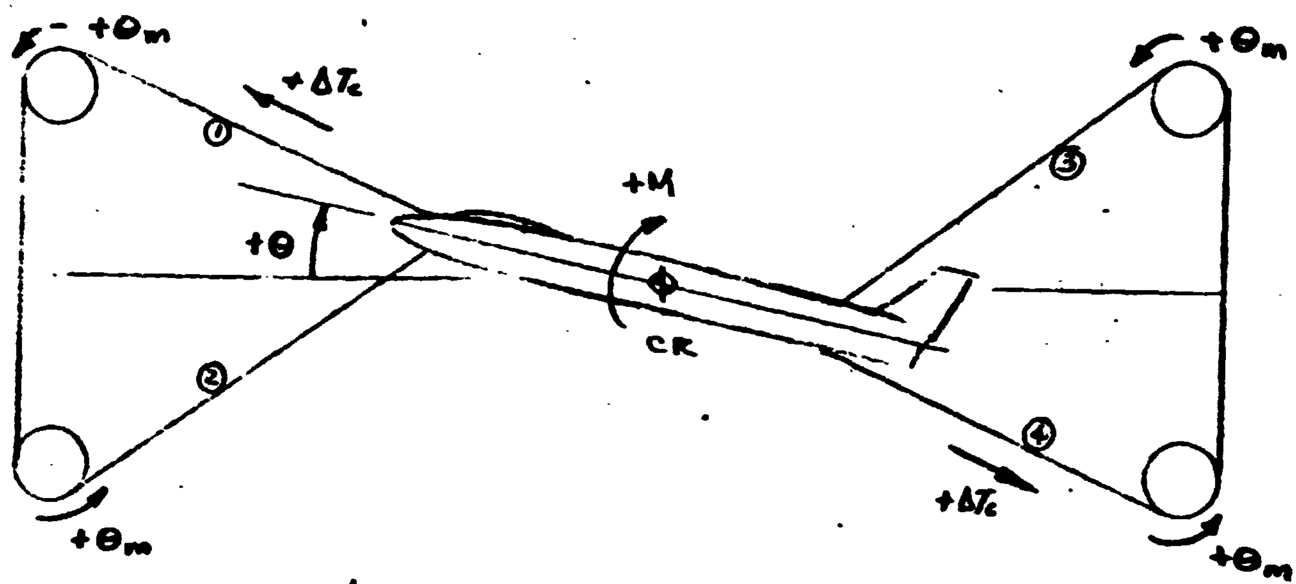
| REAL | IMAGINARY | T H/D-SEC | 1/T H/D | PERIOD-SEC | DNATF-CPS | UNDNAT-CPS | DAMP RATIO | DECAY RATIO |
|---|---|---|---|---|---|---|---|---|
| -0.1220E 01 | +-0.2390E 01 | 0.5680E 00 | 0.1760E 01 | 0.2629E 01 | 0.3804E 00 | 0.4271E 00 | 0.4547E 00 | 0.4045E-01 |
| -0.1485E 01 | +-0.3750E 01 | 0.4667E 00 | 0.2143E 01 | 0.1676E 01 | 0.5968E 00 | 0.6419E 00 | 0.3682E 00 | 0.8303E-01 |

FIGURES

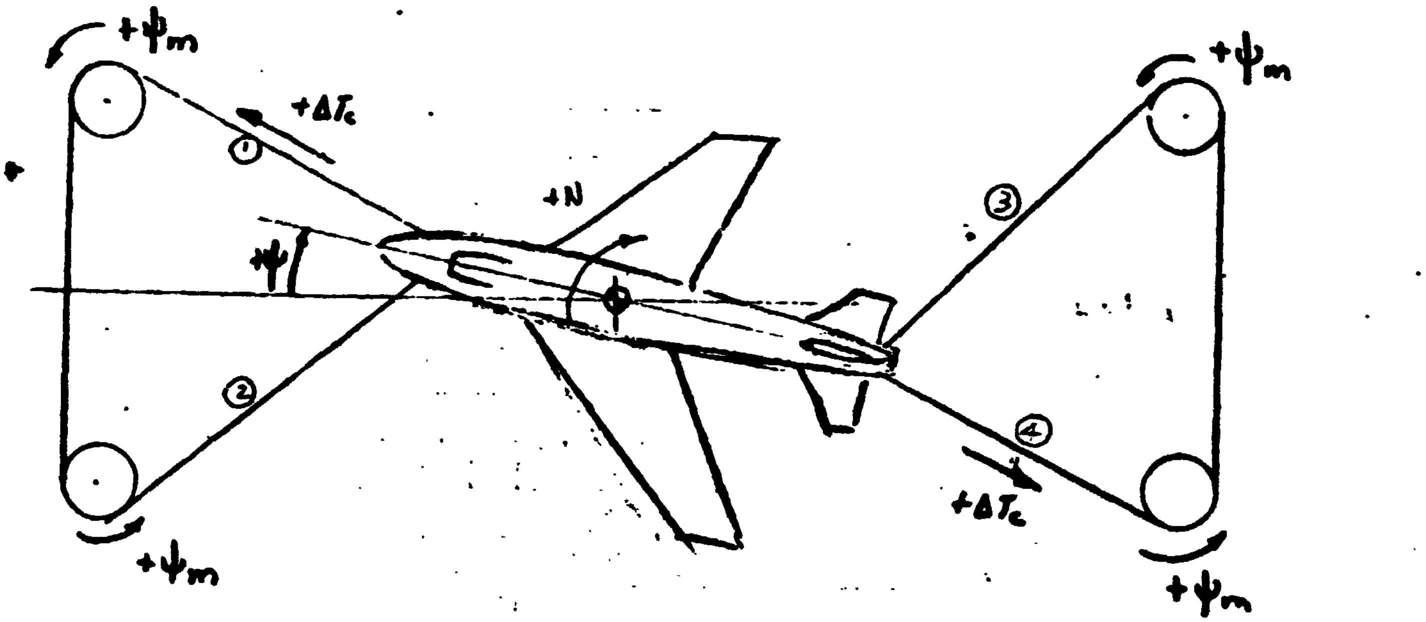
FIGURE 1

# ACTIVE CABLE MOUNT SYSTEM

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




a) LONGITUDINAL CABLE CONTROL

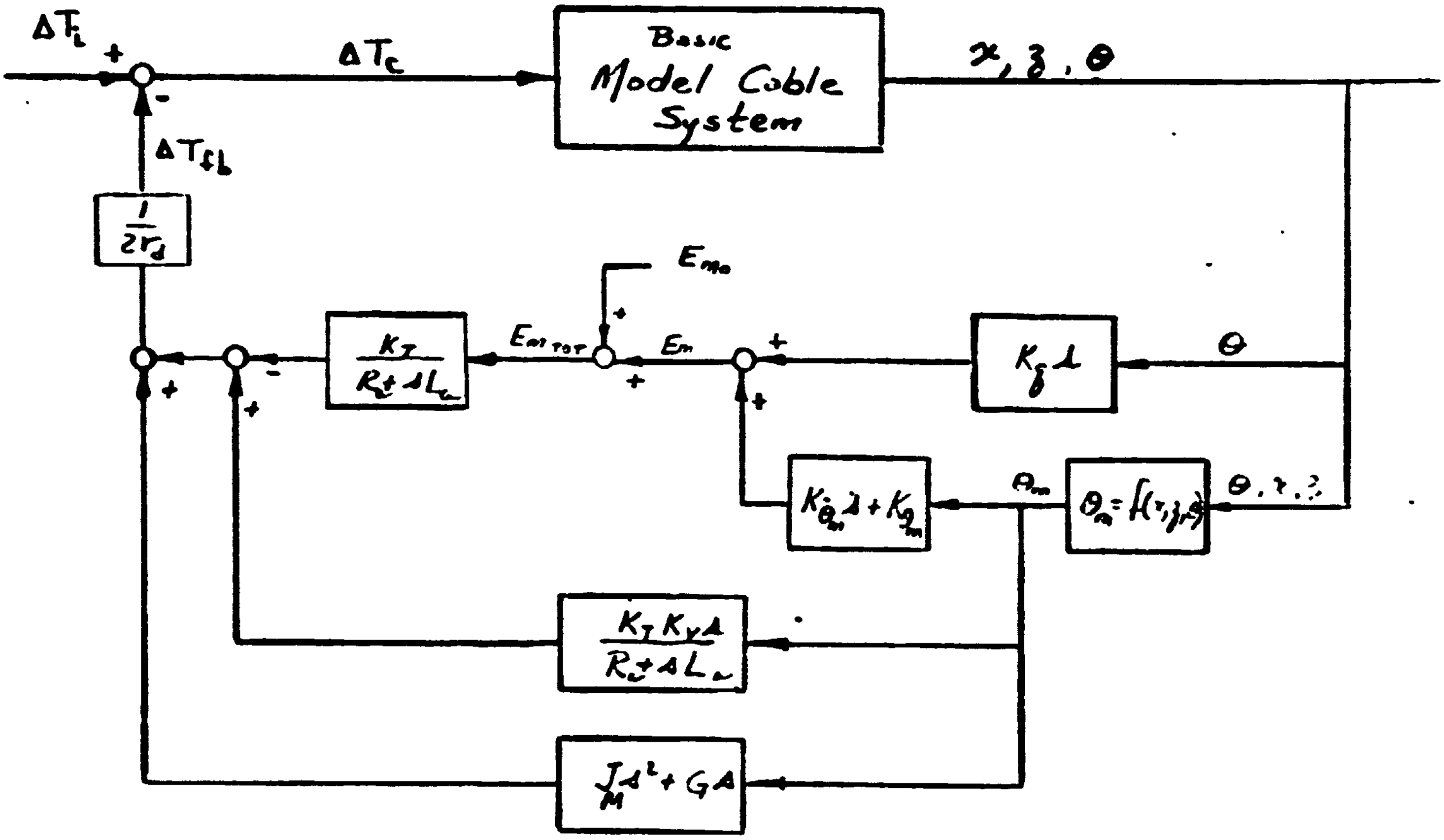



b) DIRECTIONAL CABLE CONTROL

Figure 2

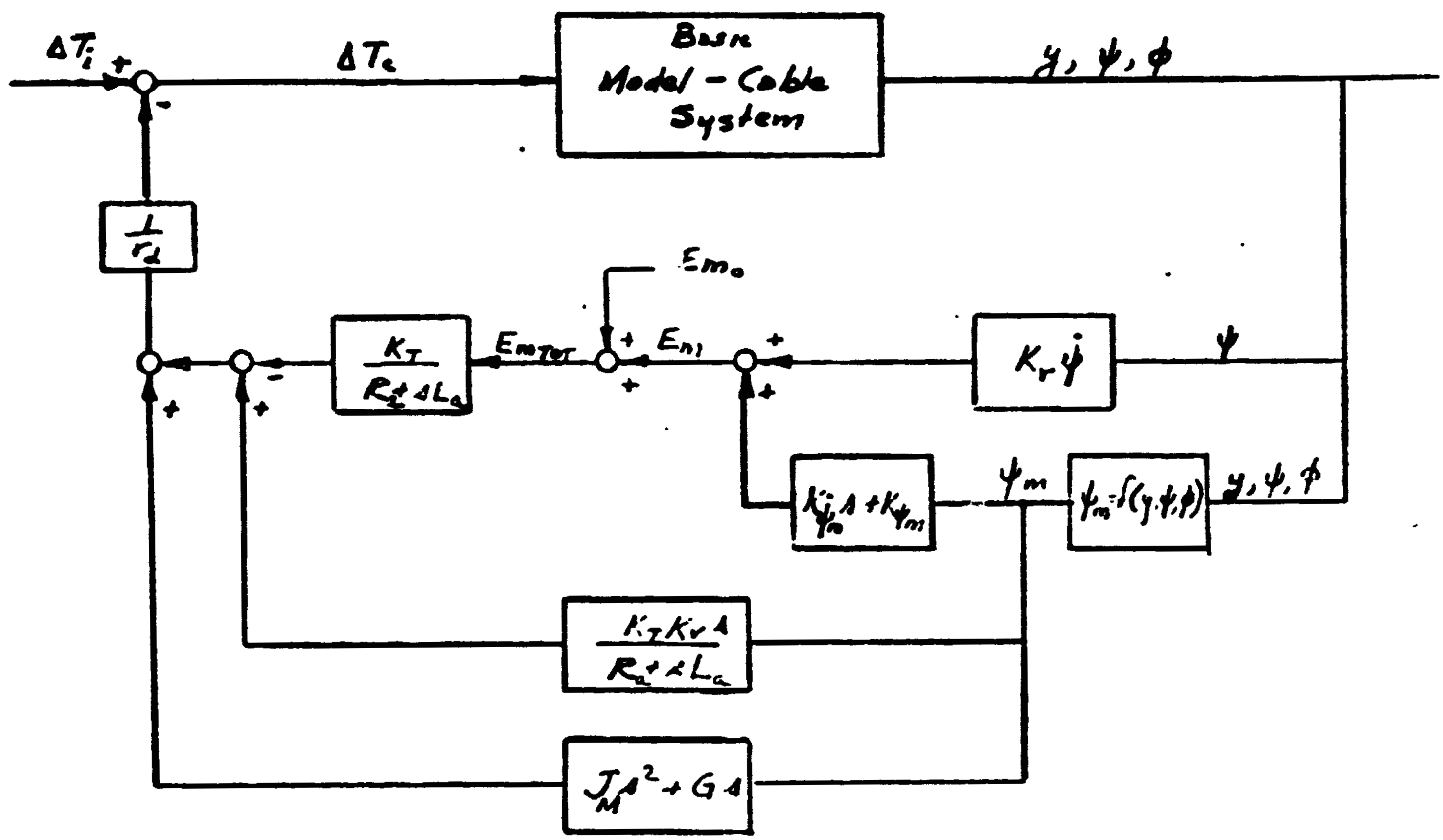
ACTIVE CABLE MOUNT SYSTEM

LONGITUDINAL BLOCK DIAGRAM



## ACTIVE CABLE MOUNT SYSTEM

## LATERAL DIRECTIONAL BLOCK DIAGRAM



## ACTIVE CABLE MOUNT SYSTEM

## EXTENDED LONGITUDINAL MATRIX

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $z$ | $\theta$ | $\Delta T_i$ | $x$ | $\Delta T_{fb}$ | $\theta_m$ | $E_{m_{TOT}}$ | $\dot{\theta}_m$ | $E_m$ | $\Delta T_c$ | $E_{m_o}$ | $\Delta T_i$ |
| Basic Matrix of Inactive Cable-Model Sys (see ref. 1) | | | | | Eq. | 1 | | | | | |
| | | | | | | 2 | | | | | |
| | | | | | | 3 | | | | | |
| | | | | | | 4 | | | | | |
| | | | | | | 5 | | | | | |
| | | | | | | 6 | | | | | |
| | | | | | | 7 | | | | | |
| | | | | | | 8 | | | | | |
| | | | | | | 9 | | | | | |
| | | | | | | 10 | | | | | |
| | | | | | | 11 | | | | | |
| | | | | | | 12 | | | | | |

Eq. 1. $m\ddot{x} - \Sigma F_{x_o} - \dfrac{\partial F_x}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 2. $m\ddot{z} - \Sigma F_{z_o} - \dfrac{\partial F_z}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 3. $I_{yy}\ddot{\theta} - \Sigma M_y - \dfrac{\partial M_y}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 4. $x - \dfrac{\partial x}{\partial z} z - \dfrac{\partial x}{\partial \theta} \theta = 0$

Eq. 5. $\Delta T_{fb} (2r_d)(R_a + sL_a) - (J_M s^2 + Gs)(R_a + L_a s)\theta_m + 2K_T K_v s\, \theta_m + 2K_T E_{m_{TOT}} = 0$

Eq. 6. $\theta_m r_d - \left[ \dfrac{\partial \Delta \ell}{\partial x} x + \dfrac{\partial \Delta \ell}{\partial z} z + \dfrac{\partial \Delta \ell}{\partial \theta} \theta \right] = 0$

Eq. 7. $E_m = K_{\theta_m} \theta_m + K_{\dot{\theta}_m} \dot{\theta}_m + K_q q$ where $q = \dot{\theta}$

Eq. 8. $\dot{\theta}_m = \theta_m s$

Eq. 9. $E_{m_{TOT}} = E_m + W_{m_o}$

Eq. 10. $\Delta T_c = \Delta T_i - \Delta T_{fb}$

## ACTIVE CABLE MOUNT SYSTEM

## EXTENDED LATERAL-DIRECTIONAL MATRIX

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $y$ | $\psi$ | $\phi$ | $\Delta T_{fb}$ | $\psi_m$ | $E_{m_{TOT}}$ | $\dot{\psi}_m$ | $\dot{\psi}$ | $E_m$ | $\Delta T_c$ | $E_{m_o}$ | $\Delta T_i$ |
| Basic Matrix of Inactive Cable-Model Sys (see ref 1.) | | | | | Eq | 1 | | | | | |
| | | | | | | 2 | | | | | |
| | | | | | | 3 | | | | | |
| | | | | | | 4 | | | | | |
| | | | | | | 5 | | | | | |
| | | | | | | 6 | | | | | |
| | | | | | | 7 | | | | | |
| | | | | | | 8 | | | | | |
| | | | | | | 9 | | | | | |
| | | | | | | 10 | | | | | |
| | | | | | | 11 | | | | | |
| | | | | | | 12 | | | | | |

**Eq. 1.** $m\ddot{y} - \Sigma Fy_o - \frac{\partial F_y}{\partial \Delta T_c} \Delta T_c = 0$

**Eq. 2** $I_{zz}\Psi - I_{xz}\dot{\phi} - \Sigma N_o - \frac{\partial N}{\partial \Delta T_c} \Delta T_c = 0$

**Eq. 3.** $I_{xx}\dot{\phi} - I_{xz}\Psi - \Sigma \mathcal{L}_o - \frac{\partial \mathcal{L}}{\partial \Delta T_c} \Delta T_c = 0$

**Eq. 4.** $\Delta T_{fb}(2r_d)(R_a + sL_a) - (J_M s^2 + Gs)(R_a + sL_a)\Psi_m + 2K_T K_v s \Psi_m + 2K_T E_{m_{TOT}} = 0$

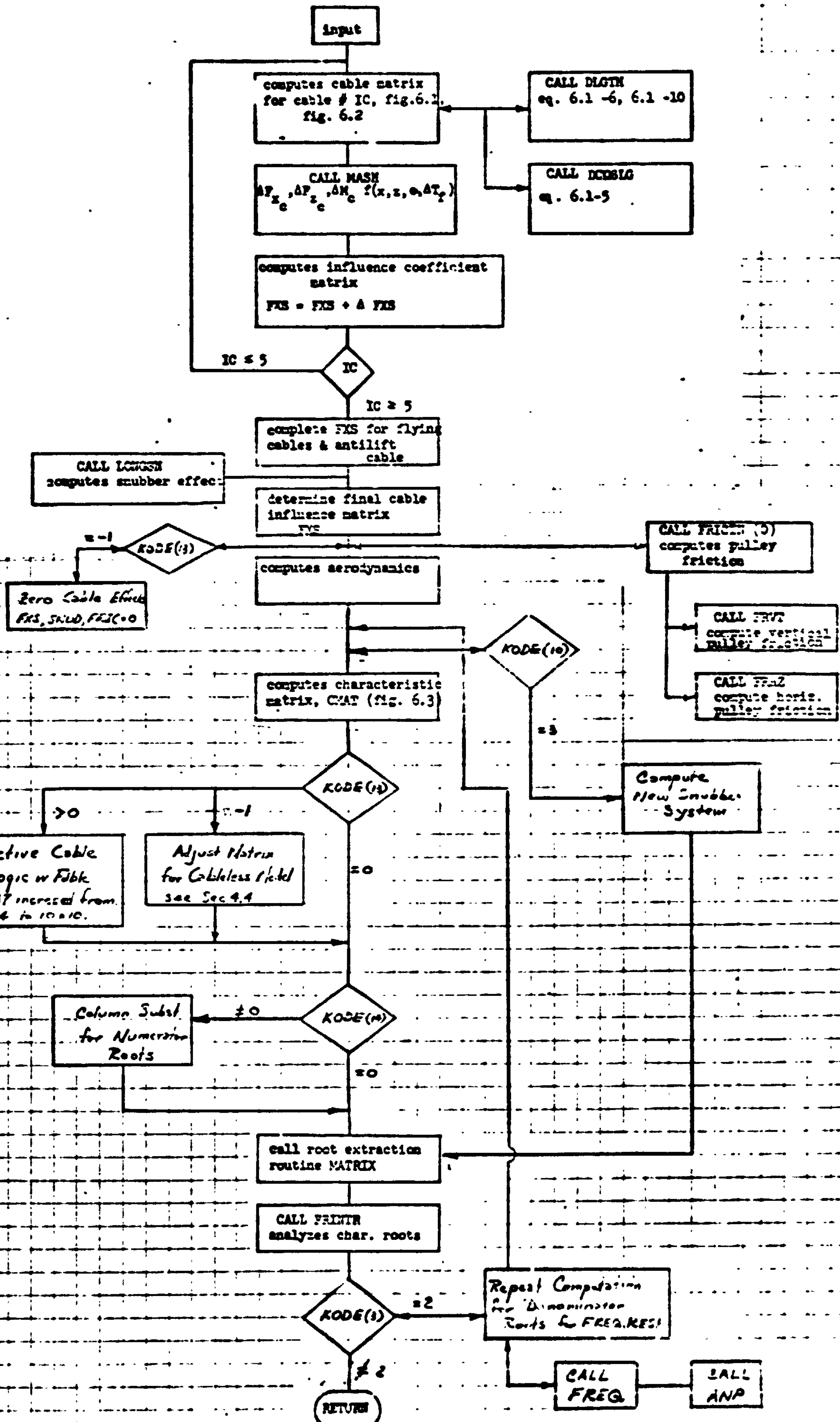
**Eq. 5.** $\Psi_m r_d + \left[ \frac{\partial \Delta \ell}{\partial y} y + \frac{\partial \Delta \ell}{\partial \Psi} \Psi + \frac{\partial \Delta \ell}{\partial \phi} \phi \right] = 0$

**Eq. 6.** $E_m = K_{\Psi_m} \Psi_m + K_{\dot{\Psi}_m} \dot{\Psi}_m + K_r \dot{\Psi}$

**Eq. 7.** $\dot{\Psi}_m - s\Psi_m = 0$ **Eq. 8.** $\dot{\Psi} - s\Psi = 0$

**Eq. 9.** $E_{m_{TOT}} = E_m + E_{m_o}$

**Eq. 10.** $\Delta T_c = \Delta T_i \ \Delta T_{fb}$




116

FIGURE 6 SUBROUTINE LONG FLOW CHART

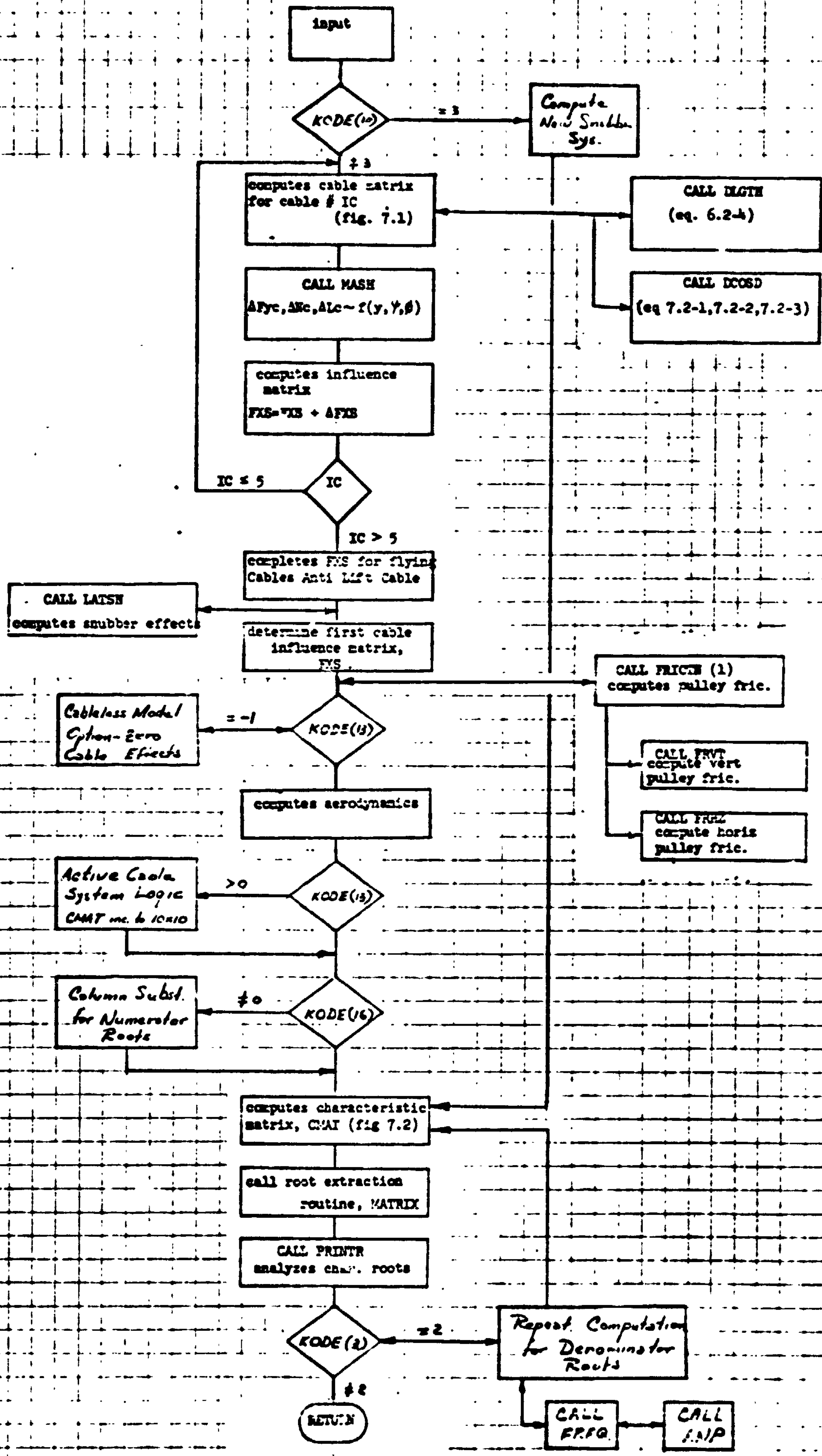





FIGURE 7 SUBROUTINE LAT FLOW CHART

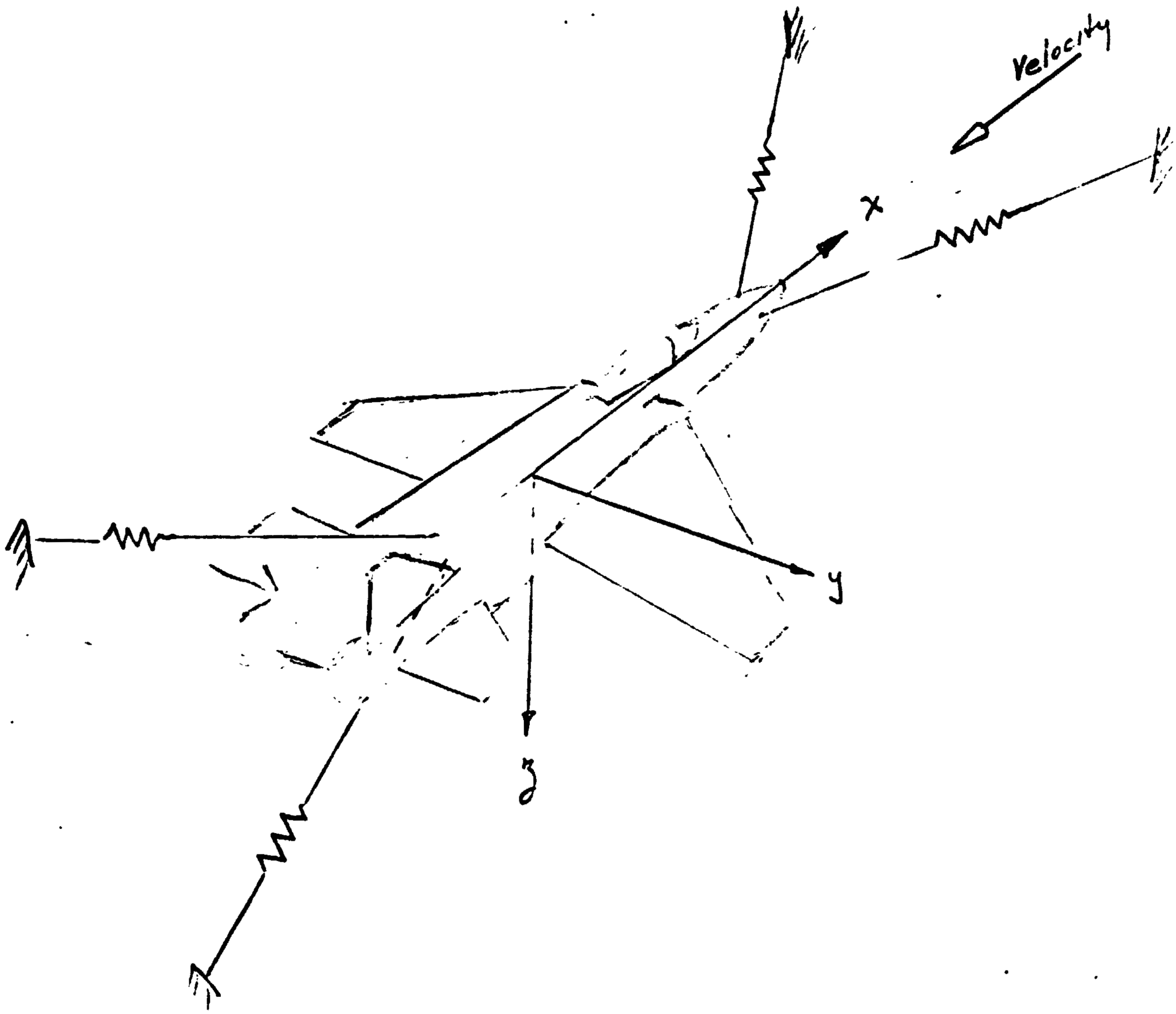





Figure 8. Schematic for Snubbed Model

$$
\begin{array}{ccccccc}
X & Z & \Theta & \Delta\alpha_x & \Delta\alpha_z & \Delta T & \Delta l \\
\end{array}
$$

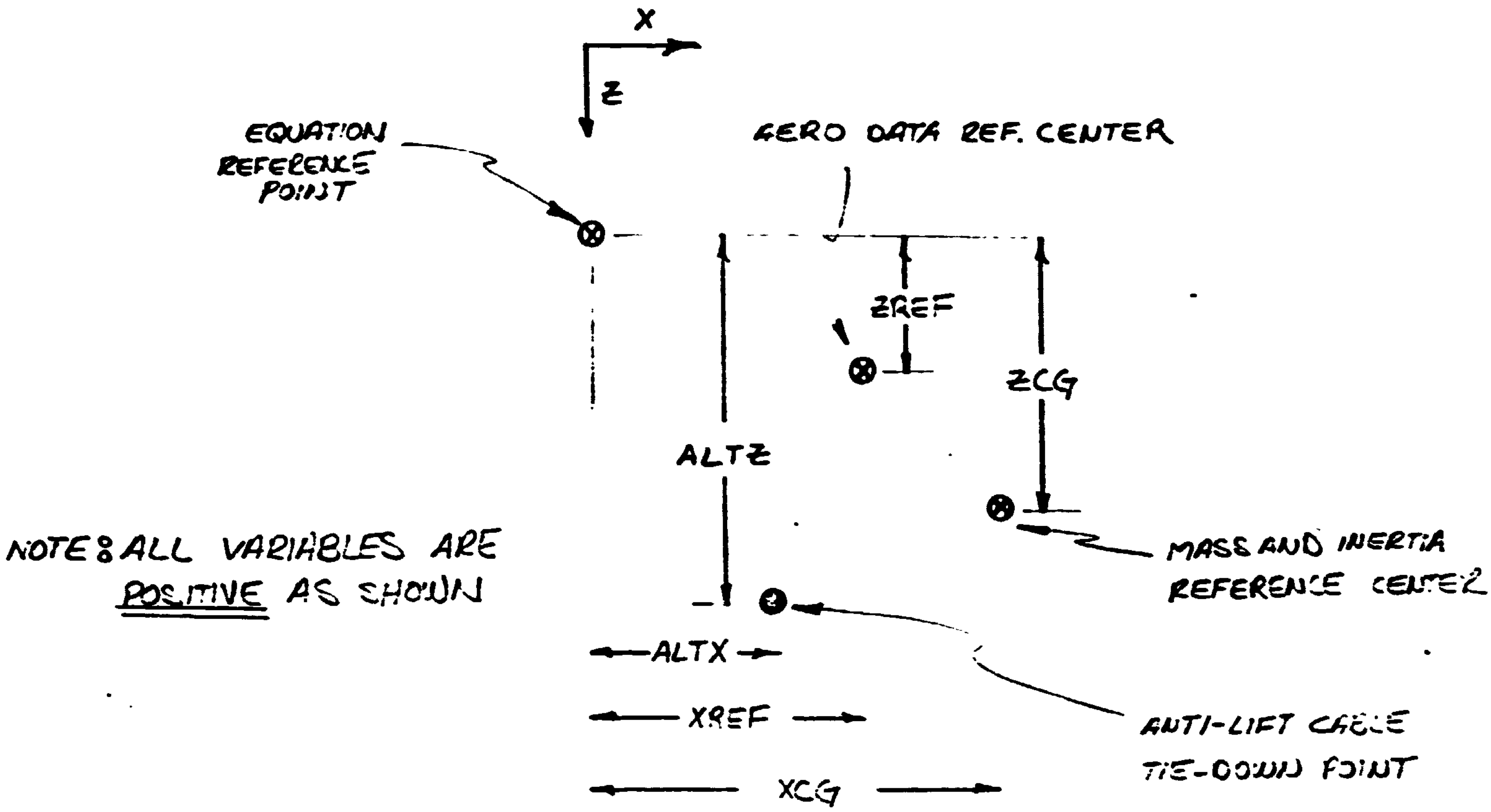
$$
\left[\begin{array}{c|c|c|c|c|c|c}
 & & -2T\cos\alpha_z & -2T\sin\alpha_x & & 2\cos\alpha_x & \\
 & & 2T\cos\alpha_x & & -2T\sin\alpha_z & 2\cos\alpha_z & \\
 & & \begin{array}{c}-2T l_z\cos\alpha_z \\ -2T l_x\cos\alpha_x\end{array} & -2T l_z\sin\alpha_x & 2T l_x\sin\alpha_z & \begin{array}{c}2 l_z\cos\alpha_x \\ -2 l_x\cos\alpha_z\end{array} & \\
1/l\sin\alpha_x & & l_z/l\sin\alpha_x & -1 & & & \\
1/l\sin\alpha_z & & l_x/l\sin\alpha_z & & -1 & & \cot\alpha_z/l \\
 & & & & & -1 & K \\
-\cos\alpha_x & -\cos\alpha_z & \begin{array}{c}-l_z\cos\alpha_x \\ -l_y\cos\alpha_z\end{array} & & & & -1
\end{array}\right]
$$

FIGURE 84: LONGITUDINAL CABLE INFLUENCE MATRIX

$$
\begin{array}{cccccccc}
Y & \psi & \phi & T & \Delta\alpha_x & \Delta\alpha_y & \Delta\alpha_z & \Delta\ell \\
\end{array}
$$

$$
\left[
\begin{array}{cccccccc}
 & -T\cos\alpha_x & T\cos\alpha_z & \cos\alpha_y & & -T\sin\alpha_i & & \\
 & -\ell_x T\cos\alpha_x - \ell_y T\cos\alpha_y & \ell_x T\cos\alpha_z & \ell_x\cos\alpha_y - \ell_y\cos\alpha_x & \ell_y T\sin\alpha_x & -\ell_x T\sin\alpha_y & & \\
 & +\ell\, T\cos\alpha_x & -\ell_z T\cos\alpha_z - \ell_y T\cos\alpha_y & \ell_y\cos\alpha_z - \ell_z\cos\alpha_y & & +\ell_z T\sin\alpha_y & -\ell_y T\sin\alpha_z & \\
 & & & -1 & & & & K \\
\frac{-\cos\alpha_y\cot\alpha_y}{-\ell} & (\ell_y\sin\alpha_x + \ell_x\cos\alpha_y\cot\alpha_x)/\ell & (\ell_z\cos\alpha_y\cot\alpha_x - \ell_y\cos\alpha_y\cot\alpha_x)/\ell & & -1 & & & \\
\sin\alpha_y/\ell & (\ell_y\cos\alpha_x\cot\alpha_y + \ell_x\sin\alpha_y)/\ell & (-\ell_z\sin\alpha_y + \ell_y\cos\alpha_z\cot\alpha_y)/\ell & & & -1 & & \\
\frac{-\cos\alpha_y\cot\alpha_z}{\ell} & (\ell_y\cos\alpha_y\cot\alpha_z - \ell_x\cos\alpha_y\cot\alpha_z)/\ell & \ell_z\cos\alpha_y\cot\alpha_z + \ell_y\sin\alpha_z/\ell & & & & -1 & \\
\end{array}
\right]
$$

FIGURE 8B : LATERAL/DIRECTIONAL CABLE INFLUENCE MATRIX




SIDE VIEW OF MODEL

FIG. 9 - REFERENCE CENTER AND LIFT CABLE INPUT DATA

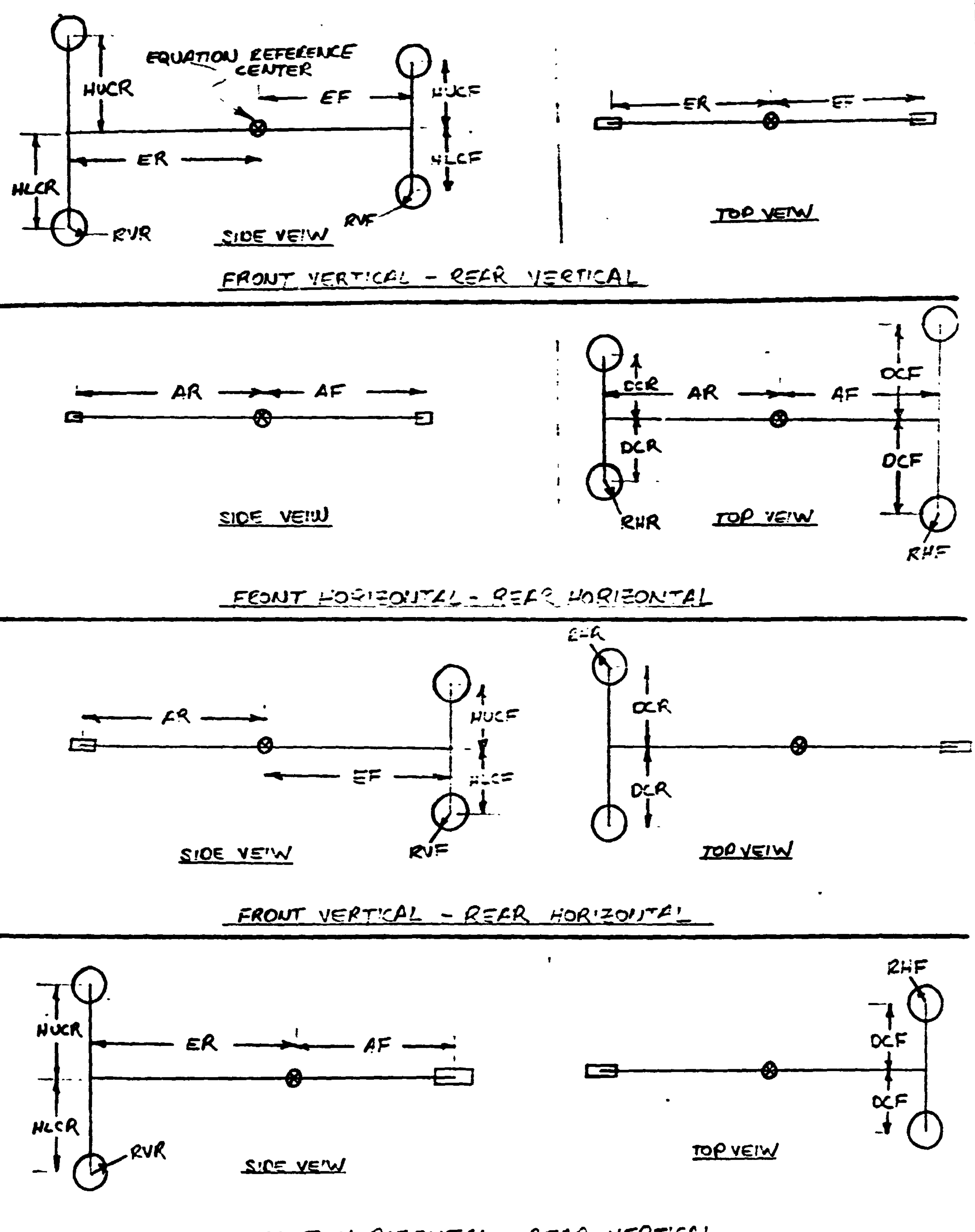



FIG. 10 - PULLEY GEOMETRY

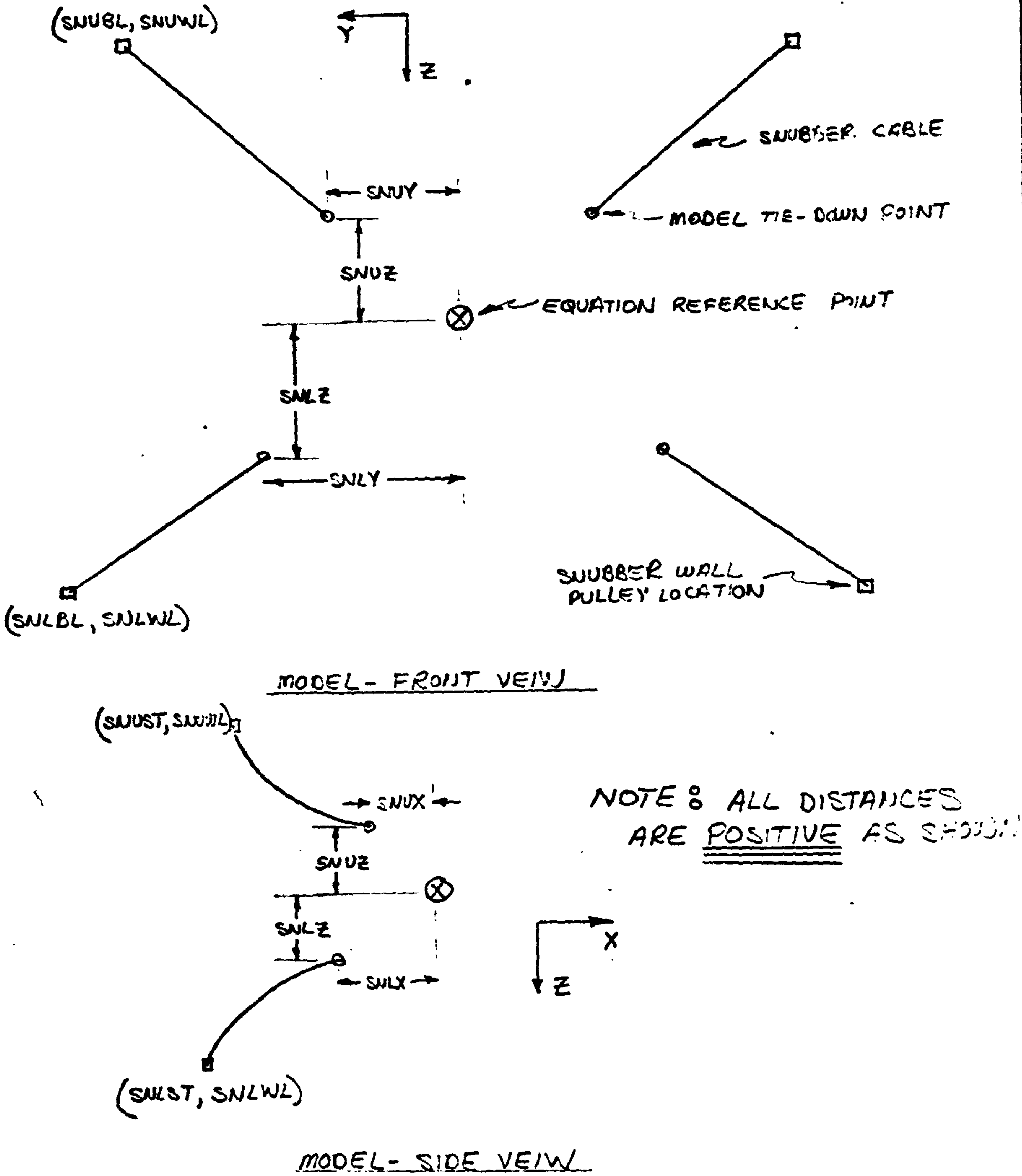



FIG. 11 - SNUBBER CABLE ARRANGEMENT