

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

N75-29130

Unclas
31379

CSCI 14B G3/09



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 ~ volts
E_{m0}	= Externally applied input voltage ~ volts
$E_{m_{tot}}$	= Total voltage to torque motor ~ volts
G	= Friction in active cable system ~ in. lbs/rps
I_a	= Amperes in motor ~ amps
J_m	= Inertia of active cable system ~ inches ⁴
K_T	= Motor torque constant ~ in. lbs/amp
K_V	= Motor velocity constant ~ volts/rps
K_q	= Model pitch rate feedback gain ~ volts/rps
$K_{\theta_m}^r$	= Pitch motor rate (tachometer) feedback gain ~ volts/rps
K_{θ_m}	= Motor position feedback gain ~ volts/rad
K_r	= Model yaw rate feedback gain ~ volts/rps
$K_{\dot{\psi}_m}$	= Yaw motor rate feedback gain ~ volts/rps
K_{ψ_m}	= Motor position feedback gain ~ volts/rad
L_a	= Motor armature inductance ~ henry
\mathcal{L}	= Rolling moment ~ ft. lb.
M	= Pitching moment ~ ft. lb.
N	= Yawing moment ~ ft. lb.
Q_o	= Output torque from motor ~ in. lb.
Q_L	= Load torque on motor ~ in. lb.
R_G	= Motor armature resistance ~ ohms
R_d	= Torque motor pulley radius ~ in.
s	= Laplace operator
ΔT	= Cable tension change ~ lbs.
ΔT_c	= One half the total cable tension change due to active cable system ($\Delta T_i - \Delta T_{fb}$) ~ lbs.
ΔT_F	= Front cable tension change due to fixed length constraint ~ lbs.

- ΔT_{fb} = one half the cable tension change due to feedback = $\delta T \sim$ lbs.
- ΔT_i = Externally applied tension change \sim lbs.
- δT = 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.
- (θ, ψ, ϕ) = Model angular displacement in pitch, yaw, and roll resp. \sim rad.
- B_g = Lateral wind gust \sim rad.
- α_g = Vertical wind gust \sim rad.
- δ_a = Aileron deflection \sim rad.
- δ_e = Elevator deflection \sim rad.
- δ_r = Rudder deflection \sim rad.
- θ_m = Vertical plane torque motor pulley angular displacement \sim rad.
- ψ_m = Lateral plane torque motor pulley angular displacement \sim rad.

TABLE OF CONTENTS

	<u>Page</u>
LIST OF SYMBOLS	iii
1.0 - INTRODUCTION	1
2.0 - ACTIVE FEEDBACK CONTROL SYSTEM LOGIC	2
3.0 - FLYING CABLE SNUBBER SYSTEM	7
3.1 - LONGITUDINAL AXIS	7
3.2 - LATERAL/DIRECTIONAL AXIS	8
4.0 - ADDITIONAL PROGRAM OPTIONS	9
4.1 - TRANSFER FUNCTIONS	9
4.1.1 - LONGITUDINAL AXIS	9
4.1.2 - LATERAL/DIRECTIONAL AXIS	10
4.2 - FREQUENCY RESPONSE	12
4.3 - WIND OFF CHARACTERISTICS	13
4.4 - CABLELESS MODEL CHARACTERISTICS.	13
5.0 - REVISED DATA INPUT.	14
REFERENCES	31
APPENDIX A - A Discussion of the Differences in Cable Attachment Points Between the Passive and Active Cable Mount System.	32
APPENDIX B - Derivation of Motor and Cable Tension Equations	35
APPENDIX C - Program Listing	37
APPENDIX D - Sample Input	99
APPENDIX E - Sample Output.	101
FIGURES	110

I. INTRODUCTION

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

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

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

2.0 ACTIVE FEEDBACK CONTROL SYSTEM LOGIC

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

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

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

θ_m and ψ_m are torque motor pulley angular displacements in the vertical and horizontal planes respectively. Note that the sense of rotation is unaltered whether the cable is located in the front or rear. Positive motor rotation corresponds to an increase in cable tension on the sides noted in the figure by "+". Positive ΔT is an increase in cable tension and negative ΔT 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, ΔT_c , will result in a model motion defined by variables x , z , and θ .

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_{θ_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_{IO} , 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, Δl , as a function of the model motion. The " Δl " 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, θ_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 ΔT . The magnitude of ΔT is half the total cable tension. The ΔT 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, ΔT_{fb} , is subtracted from the input ΔT_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 , Ψ , ϕ are the perturbation variables. The feedback gains K_r , $K_{\dot{\psi}_m}$ and $K_{\dot{\phi}_m}$ 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 ΔT_c terms in equations 1 through 3. The coefficients of ΔT_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 ΔT_c terms in equations 1 through 3. The remaining equations are similar to those of figure 4. The only difference being that these equations represent the lateral-directional mode..

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

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

Open and closed loop characteristic roots as well as numerator roots can be derived from these matrices. The procedure for obtaining this information from the program is discussed in Section 4.0.

3.0 FLYING CABLE SNUBBER SYSTEM

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

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

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

3.1 LONGITUDINAL AXIS

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

The matrix is reduced to a 3 x 3 in x, z, θ and put in the FXS array. The longitudinal stability is a 3 x 3 matrix in x, z, and θ . The matrix no longer contains ΔT_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, ψ , and ϕ , 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, ΔT_i , a model elevator input, δ_e , or a vertical gust input, α_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, θ/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/δ_e , θ/δ_e and x/δ_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 desired. These are respectively a cable tension change, ΔT_c , a rudder input, δ_r , an aileron input, δ_a , or a side gust, B_g . The dependent variable is determined by KODE (17) which may vary from 1 through 3. The denominator roots are obtained by setting KODE (16) to zero.

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

The closed loop numerator for the active cable system is determined by setting KODE (9) to 10. The forcing function is defined by KODE (16). This code can be 11, 12, 14, 15, or 16. They correspond to a test voltage, E_{mo} , test tension, ΔT_1 , rudder input, δ_r , aileron input, δ_a , or a side wind gust, βg .

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 ΔT_p 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 TABINI (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:

<u>ARRAY</u>	<u>DESCRIPTION</u>
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.

<u>NAME</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
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.

<u>NAME</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
KODE (8)	+3	Longitudinal matrix - Cableless Model (see Section 4.3)
	+4	Longitudinal matrix - no stability augmentation.
	+5	Longitudinal matrix - internal stabil- ity 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	Longitudinal matrix - Close loop res- ponse 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 2.0 , 4.1.2)
	+10	Lateral/directional matrix - Close loop response of Active Cable Mount System (See Section 2.0 , 4.1.2)

<u>NAME</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
KODE (10)	0	No snubbers.
	+1	Analyze conventional snubbers in un-snubbed condition - see Section 8.1, Reference 1.
	+2	Analyze conventional snubbers in snubbed condition - See Section 8.2, Reference 1.
KODE (11)	+3	Analyze flying cable snubber system.
	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, ΔT_c . (See Section 4.1.1)
	+10	Longitudinal System - numerator and/or frequency characteristics of active cable mount system open loop for cable tension input, ΔT_c . (See Section 4.1.1)

<u>NAME</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
	+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, ΔT_1 . (See Section 4.1.1)
	+15	Longitudinal system - numerator and/or frequency characteristics for pitch control response (δe)
	+16	Longitudinal system - numerator and/or frequency characteristics for gust response (α_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

NAMEVALUEDESCRIPTION

+10	Lateral/directional system - numerator and/or frequency characteristics of inactive cable mount system for tension input ΔT_c . (See Section 4.1.2).
+10	Lateral/directional system - numerator and/or frequency characteristics of active cable mount system open loop for tension input, ΔT_c . (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, $(\Delta T)/$ (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).

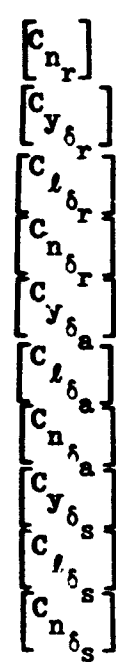
NAMEVALUEDESCRIPTION

	+16	Lateral/directional system - numerator and/or frequency characteristics for gust response (β_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	$\partial C_D / \partial (u/V_0)$
AERO (2)	N.D.	CLU	$\partial C_L / \partial (u/V_0)$
AERO (3)	N.D.	CMU	$\partial C_m / \partial (u/V_0)$
AERO (4)	1/rad	CDA	$\partial C_D / \partial (\alpha)$
AERO (5)	1/rad	CLA	$\partial C_L / \partial (\alpha)$
AERO (6)	1/rad	CMA	$\partial C_m / \partial (\alpha)$
AERO (7)	N.D.	CDQ	$\partial C_D / \partial (q\bar{C}/2V_0)$
AERO (8)	N.D.	CLQ	$\partial C_L / \partial (q\bar{C}/2V_0)$
AERO (9)	N.D.	CMQ	$\partial C_m / \partial (q\bar{C}/2V_0)$
AERO (10)	N.D.	CDO	Drag coefficient at $\alpha = 0$
AERO (11)	N.D.	CLO	Lift coefficient at $\alpha = 0$
AERO (12)	N.D.	CMO	Pitching moment at $\alpha = 0$
AERO (13)	1/rad	CDDE	$\partial C_D / \partial (\delta_e)$
AERO (14)	1/rad	CLDE	$\partial C_L / \partial (\delta_e)$
AERO (15)	1/rad	CMDE	$\partial C_m / \partial (\delta_e)$
AERO (16)	N.D.	CDAD	$\partial C_D / \partial (\dot{\alpha}\bar{C}/2V_0)$
AERO (17)	N.D.	CLAD	$\partial C_L / \partial (\dot{\alpha}\bar{C}/2V_0)$
AERO (18)	N.D.	CMAD	$\partial C_m / \partial (\dot{\alpha}\bar{C}/2V_0)$
AERO (19)	1/rad	CYB	$\partial C_y / \partial (\theta)$
AERO (20)	1/rad	CLB	$\partial C_l / \partial (\theta)$
AERO (21)	1/rad	CNB	$\partial C_n / \partial (\theta)$
AERO (22)	N.D.	CYP	$\partial C_y / \partial (pb/2V_0)$
AERO (23)	N.D.	CLP	$\partial C_l / \partial (pb/2V_0)$
AERO (24)	N.D.	CNP	$\partial C_n / \partial (pb/2V_0)$
AERO (25)	N.D.	CYR	$\partial C_y / \partial (rb/2V_0)$
AERO (26)	N.D.	CLR	$\partial C_l / \partial (rb/2V_0)$

C_{D_u}
 C_{L_u}
 C_{m_u}
 C_{D_α}
 C_{L_α}
 C_{m_α}
 $C_{D_{q\bar{C}}}$
 $C_{L_{q\bar{C}}}$
 $C_{m_{q\bar{C}}}$
 C_{D_0}
 C_{L_0}
 C_{m_0}
 $C_{D_{\delta_e}}$
 $C_{L_{\delta_e}}$
 $C_{m_{\delta_e}}$
 $C_{D_{\dot{\alpha}\bar{C}}}$
 $C_{L_{\dot{\alpha}\bar{C}}}$
 $C_{m_{\dot{\alpha}\bar{C}}}$
 C_{y_θ}
 C_{l_θ}
 C_{n_θ}
 $C_{y_{pb}}$
 $C_{l_{pb}}$
 $C_{n_{pb}}$
 $C_{y_{rb}}$
 $C_{l_{rb}}$

NAME	UNITS	LABEL	DESCRIPTION
AERO (27)	N.D.	CNR	$\partial C_n / \partial (rb/2V_o)$
AERO (28)	1/rad	CYDR	$\partial C_y / \partial (\delta_r)$
AERO (29)	1/rad	CLDR	$\partial C_l / \partial (\delta_r)$
AERO (30)	1/rad	CNDR	$\partial C_n / \partial (\delta_r)$
AERO (31)	1/rad	CYDA	$\partial C_y / \partial (\delta_a)$
AERO (32)	1/rad	CLDA	$\partial C_l / \partial (\delta_a)$
AERO (33)	1/rad	CNDA	$\partial C_n / \partial (\delta_a)$
AERO (34)	1/rad	CYDS	$\partial C_y / \partial (\delta_s)$
AERO (35)	1/rad	CLDS	$\partial C_l / \partial (\delta_s)$
AERO (36)	1/rad	CNDS	$\partial C_n / \partial (\delta_s)$
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-ft ²	XIXZ	Model cross product of inertia (I _{XZ})
AERO (57)	slug-ft ²	XIXX	Model roll inertia (I _{xx}), body axis at C.G.
AERO (58)	slug-ft ²	YIYY	Model pitch inertia (I _{yy}), body axis at C.G.
AERO (59)	slug-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	Radius 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	SNY	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	SNUEL	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_{θ_m})
AERO (138)	volts/rad	AKTH	Pulley rotation displacement feedback (K_{θ_m})
AERO (139)	in-lbs/rad/sec	GMP	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_Y)
AERO (143)	volts/rad	AKY	Pulley rotation displacement feedback (K_{Y_m})
AERO (144)	volts/rad/sec	AKYD	Pulley rotation rate feedback ($K_{\dot{Y}_m}$)
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 AER. (36). The table input format is shown in Appendix A of Reference 1. This data is read in under TABINI.

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.

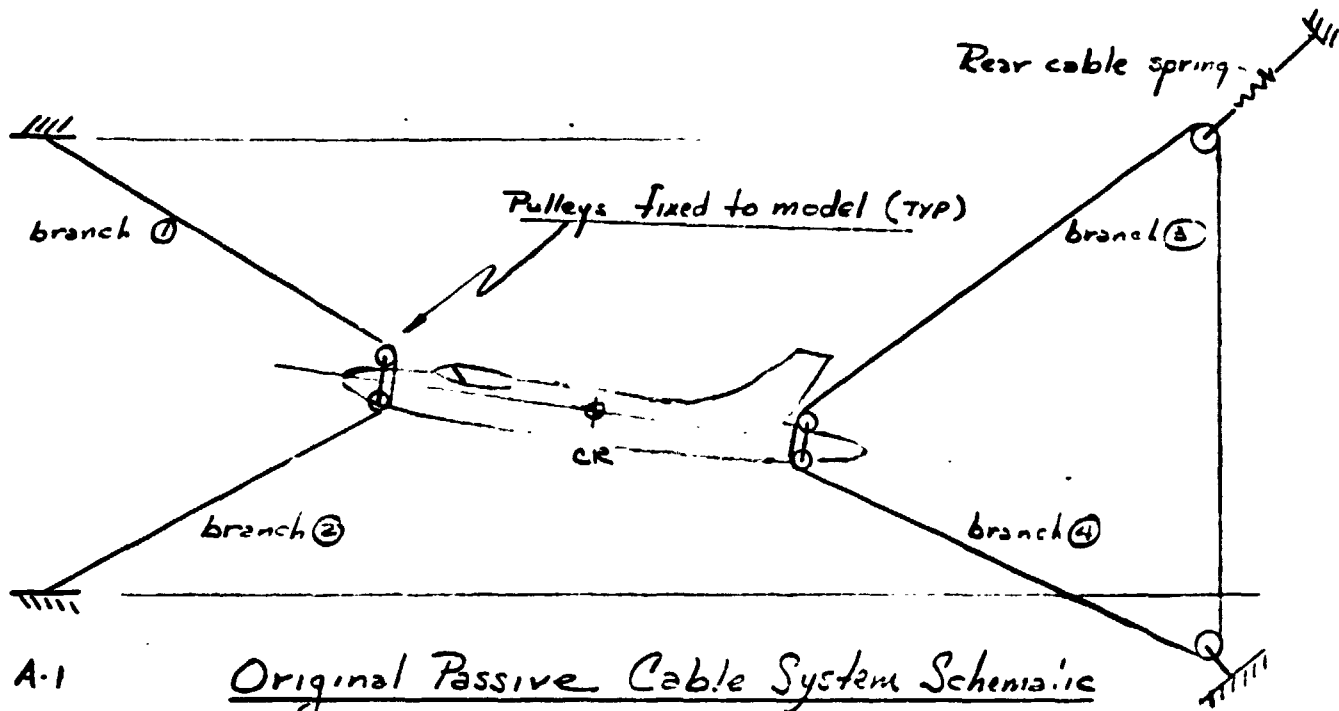


fig A-1 Original Passive Cable System Schematic

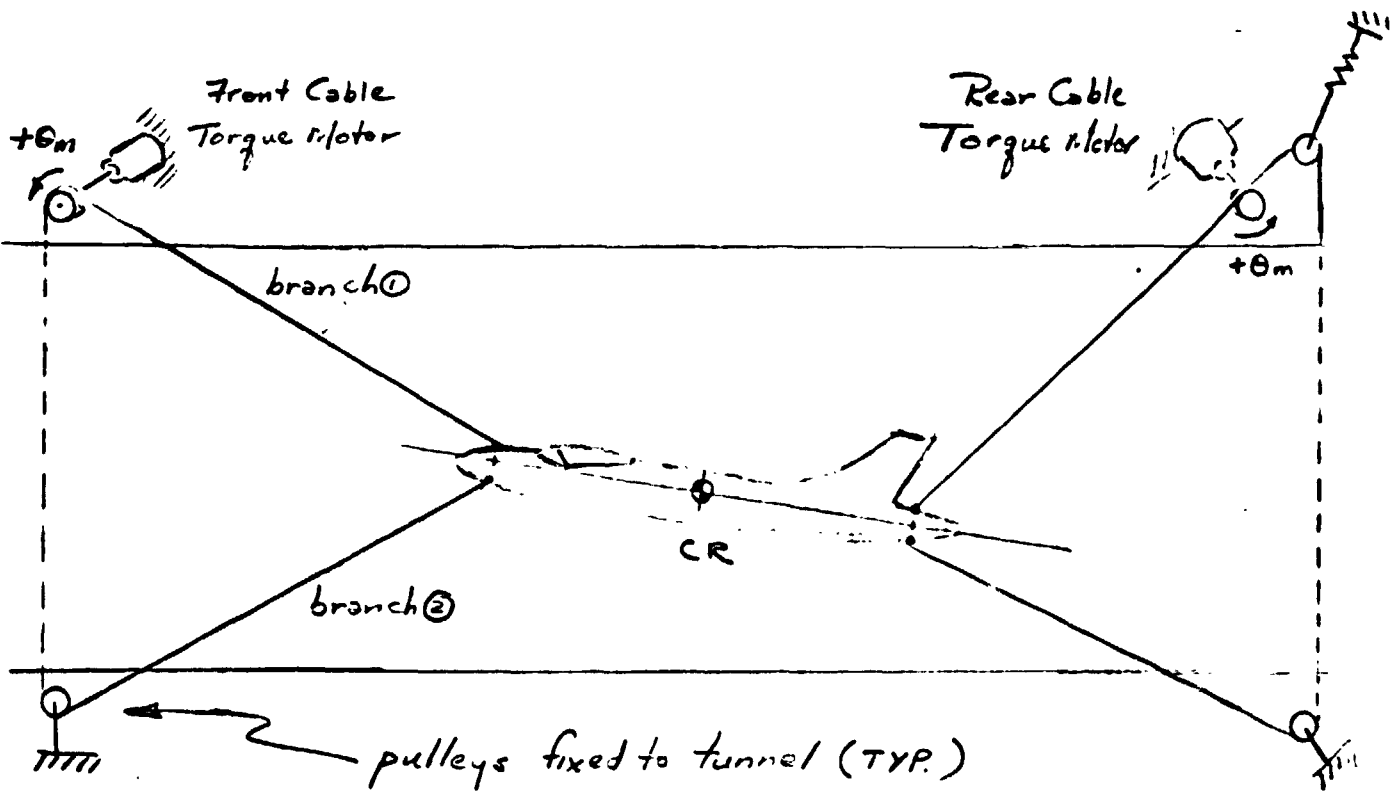


fig A-2 Active 2-Cable Mount System Schematic

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.

$$Q_o = K_T I_a = K_T \left[\frac{E_{mTOT} - K_v s \theta_m}{R_a + sL_a} \right] \quad (1)$$

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

$$Q_L = \Delta T_{TOT} r_d + Gs \theta_m \quad (2)$$

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

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

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

$$\Delta T_{tot} = \frac{1}{r_d} \left\{ \left[J_M s^2 + Gs + \frac{2K_T K_v s}{R_a + sL_a} \right] \theta_m - \frac{2K_T E_{mTOT}}{R_a + sL_a} \right\} \quad (4)$$

ΔT_{tot} is positive when the cable is in tension.

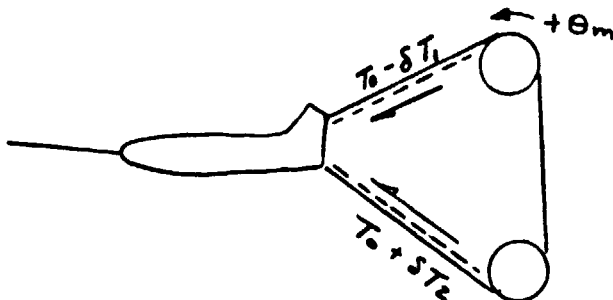


fig B-1

Looking at the larger picture shown in figure B-1, the total change in cable tension can be split into two increments δT_1 and δT_2 . Writing the equation of motion of the cable

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

$$\text{For } \ddot{a} = 0$$

$$-\delta T_1 - \delta T_2 = 0 \quad (6)$$

$$\text{and } \delta T_2 = -\delta T_1 \quad (7)$$

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

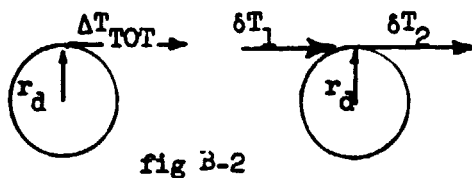


Figure B-2 shows the relation of the change in cable tension on one side of the torque motor, δT , to the total change in cable tension ΔT_{TOT} . Thus

$$\delta T_2 - \delta T_1 = \Delta T_{TOT} \quad (8)$$

Substituting results from equation 7 into equation 8

$$\delta T_2 = \frac{\Delta T_{TOT}}{2} \quad (9)$$

Replacing ΔT_{TOT} in equation (4) with equation (9), δT_2 is determined.

The δT_2 corresponds to ΔT_{fb} in figures 4 and 5

$$\delta T_2 = \frac{1}{2r_d} \left\{ \left[J_m s^2 + Gs + \frac{2K_T K_v s}{R_a + sL_a} \right] \theta_m - \frac{2K_T E_{mTOT}}{R_a + sL_a} \right\} \quad (10)$$

APPENDIX C
PROGRAM LISTINGS

C THIS IS THE ACTIVE TWO CABLE MOUNT SYSTEM ANALYSIS PROGRAM
 C DEVELOPED JULY.74 TO MAY.75
 C

CBL00010
 CBL00020
 CBL00030

COMMON/INPUT/IW,IF

CBL00040

COMMON/DAT/AERD(175),AERDP(50),KCODE(26),LL

CBL00050

COMMON/SNJ33/SNU(3,3),SN(30),THUSN,THLSN,SNUP(3,7)

CBL00060

COMMON ZZZ(200)

CBL00070

COMMON/TAB1/ZZ(300)

CBL00080

COMMON/DU/DUM(10,10)

CBL00090

COMMON/ANAME/NAME(16),NAME1(16)

CBL00100

DIMENSION TITLE(20),SAVE(50),SAVE1(150),IKH(160)

CBL00110

EQUIVALENCE(AERD(1), CDU),(AERD(2), CLU),(AERD(3), CMJ),

CBL00120

1 (AERD(4), CDA),(AERD(5), CLA),(AERD(6), CMA),

CBL00130

2 (AERD(7), CDD),(AERD(8), CLD),(AERD(9), CMQ),

CBL00140

3 (AERD(10), CDD),(AERD(11), CLD),(AERD(12), CMO),

CBL00150

4 (AERD(13), CDDF),(AERD(14), CLDF),(AERD(15), CMDF),

CBL00160

5 (AERD(16), CDDI),(AERD(17), CLDI),(AERD(18), CMAD),

CBL00170

6 (AERD(19), CYB),(AERD(20), CLB),(AERD(21), CNB),

CBL00180

7 (AERD(22), CYP),(AERD(23), CLP),(AERD(24), CNP),

CBL00190

8 (AERD(25), CYP),(AERD(25), CLP),(AERD(27), CNP),

CBL00200

9 (AERD(28), CYDP),(AERD(29), CLDP),(AERD(30), CNDP),

CBL00210

A (AERD(31), CYDA),(AERD(32), CLDA),(AERD(33), CNDA),

CBL00220

B (AERD(34), CYDS),(AERD(35), CLDS),(AERD(36), CNDS),

CBL00230

C (AERD(44), XPEF),(AERD(45), ZPEF),(AERD(46), XCG),

CBL00240

D (AERD(47), ZCG)

CBL00250

EQUIVALENCE(AERD(48),AWACH),(AERD(49),V0),(AERD(50), AM)

CBL00260

EQUIVALENCE(AERD(51),RHO),(AERD(52), WT),(AERD(53),B)

CBL00270

EQUIVALENCE(AERD(54),CBAR),(AERD(55),SW),(AERD(56), XIXZ)

CBL00280

EQUIVALENCE(AERD(57),XIXX),(AERD(58),YIYY),(AERD(59),ZIZZ)

CBL00290

EQUIVALENCE(AERD(60),CLT),(AERD(61),COT),(AERD(62),CMT),

CBL00300

1 (AERD(63),THETA)

CBL00310

EQUIVALENCE(AERD(66),WLUF),(AERD(67),WLLF),(AERD(68),WLUF),

CBL00320

1 (AERD(69),WLLR),(AERD(70),WLHF),(AERD(71),WLHF),

CBL00330

2 (AERD(72),STAR),(AERD(73),STAR),(AERD(74),PLHF),

CBL00340

3 (AERD(75),BLHF),(AERD(76),WLCF),(AERD(77),STACR),

CBL00350

4 (AERD(78),HLCF),(AERD(79), EF),(AERD(80), EF),

CBL00360

5 (AERD(81), AF),(AERD(82), AF),(AERD(83),HUCF),

CBL00370

6 (AERD(84),HLCF),(AERD(85),HUCF),(AERD(85),HLCF),

CBL00380

7 (AERD(87),DCF),(AERD(88),DCR),

CBL00390

8 (AERD(90),RVF),(AERD(91),RHF),(AERD(92),RVF),

CBL00400

9 (AERD(93),RHF),(AERD(94), TCG),(AERD(95),AKR),

CBL00410

A (AERD(96), CDU),(AERD(97),STLTT),(AERD(98),WLLTT),

CBL00420

B (AERD(99),TLFTC),(AERD(100),AKLFT),

CBL00430

C (AERD(102),ALTZ),(AERD(103),ALTZ),(AERD(104), CMP)

CBL00440

EQUIVALENCE(AERD(105),SNUX),(AERD(106),SNUY),(AERD(107),SNUZ),

CBL00450

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

CBL00460

2 (AERD(111),SNUST),(AERD(112),SNUWL),(AERD(113),SNUAL),

CBL00470

3 (AERD(114),SNLST),(AERD(115),SNLWL),(AERD(116),SNLRL),

CBL00480

4 (AERD(117),TUSNO),(AERD(118),TLSNO),(AERD(119),AKSNU),

CBL00490

5 (AERD(120),AKSNL),(AERD(121),ADSNU),(AERD(122),ADSNL),

CBL00500

6 (AERD(123),AKSY),(AERD(124),AKPHI),(AERD(125),AKTHE),

CBL00510

7 (AERD(126),AKAZ),(AERD(127), T1SY),(AERD(128),T2PHI),

CBL00520

8 (AERD(129),T3THE),(AERD(130), TAAZ)

CBL00530

EQUIVALENCE(AERDP(1),CXUP),(AERDP(2),CZUP),(AERDP(3),CMUP),

CBL00540

1 (AERDP(4),CXAP),(AERDP(5),CZAP),(AERDP(6),CMAP),

CBL00550

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), CYFP),(AEROP(26), CLFP),(AEROP(27), CNFP),	CBL00620
9	(AEROP(28),CYDFP),(AEROP(29),CLDFP),(AEROP(30),CNDFP),	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=1	CBL00760
	IP=5	CBL00770
	IN=6	CBL00780
	LLL=1	CBL00790
	DJ 1: J=1.50	CBL00800
11	SAVE(J)=9999.	CBL00810
	DJ 12 I=1.150	CBL00820
12	AEROP(I)=0.	CBL00830
	LL=0	CBL00840
	READ(IP,150)(TITLE(I),I=1,20)	CBL00850
	READ(IP,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 SITE	CBL00900
	WRITE(IW,171)(I,I=1,24),(KODE(I),I=1,24)	CBL00910
	IF(KODE(7).EQ.1) GO TO 10	CBL00920
	READ(IP,100)(AEROP(I),I=1,36)	CBL00930
	GO TO 20	CBL00940
10	CALL TABIN(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(IP,100)(AEROP(I),I=44,59)	CBL01000
	READ(IP,100)(AEROP(I),I=66,130)	CBL01010
	IF(KODE(13).GT.0.)READ(IP,100)(AEROP(I),I=131,160)	CBL01020
100	FORMAT(6E12.5)	CBL01030
	IF(AEROP(48).EQ.0..AND.AEROP(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	CALC1110
1000	DO 28 I=1,150	CALC1120
	28 AERO(I)=SAVE1(I)	CALC1130
	READ(IP,150,END=500)(TITLE(I),I=1,20)	CALC1140
150	FORMAT(20A4)	CALC1150
	KASE=1	CALC1150
	DO 34 J=1,50	CALC1170
34	SAVE(J)=9999.	CALC1180
	READ(IP,200)(KODE(I),I=1,24)	CALC1190
	WRITE(IW,170) KODE(I),(TITLE(I),I=1,20)	CALC1200
	CALL PITE	CALC1210
	IKV=0	CALC1220
	DO 26 I=1,160	CALC1230
	READ(IP,350)K,VALUE	CALC1240
	IKH(I)=K	CALC1250
	IF(K.LT.1)GO TO 22	CALC1260
	IKV=IKV+1	CALC1270
	AERO(K)=VALUE	CALC1290
26	IF(K.LT.37)SAVE(K)=AERO(K)	CALC1290
22	IF(AERO(48).EQ.0..AND.AERO(49).EQ.0.)WRITE(IW,1003)	CALC1300
	WRITE(IW,171)(I,I=1,24),(KODE(I),I=1,24)	CALC1310
171	FORMAT(// ' CODE NOS. FOR THIS CASE.',/,24I5,/,24I5)	CALC1320
	WRITE(IW,352)	CALC1330
352	FORMAT(3X,'DATA CHANGE')	CALC1340
350	FORMAT(I3,F12.5)	CALC1350
	IF(IKV.LE.0)GO TO 24	CALC1360
	DO 24 I=1,7	CALC1370
	K=IKH(I)	CALC1380
	VALUE=AERO(K)	CALC1390
24	WRITE(IW,350)K,VALUE	CALC1400
351	FORMAT(3X,I3,3X,F12.5)	CALC1410
	LL=0	CALC1420
32	IF(KODE(7).EQ.0) GO TO 31	CALC1430
	DO 30 I=1,36	CALC1440
	CALL STINT1(AMACH,0.0,I,I,AERO(I),NG)	CALC1450
	IF(NG.NE.0) GO TO 40	CALC1460
30	CONTINUE	CALC1470
	DO 36 J=1,36	CALC1480
36	IF(SAVE(J).NE.9999.) AERO(J)=SAVE(J)	CALC1490
	GO TO 31	CALC1500
40	WRITE(IW,400) I,NG	CALC1510
400	FORMAT(//,' ERROR IN TABLE NO',I4,'NG=',I3)	CALC1520
	GO TO 500	CALC1530
360	FORMAT(6E10.3)	CALC1540
31	IF(KASE.EQ.1) GO TO 9	CALC1550
	WRITE(IW,801)	CALC1560
801	FORMAT(5X,'INPUT DATA AS SPECIFIED IN AERO ARRAY')	CALC1570
	WRITE(IW,800)(I,AERO(I),I=1,150)	CALC1580
800	FORMAT(5(2X,'AERO(',I3,')='),G10.3))	CALC1590
9	DO 25 I=1,150	CALC1600
25	SAVE1(I)=AERO(I)	CALC1610
	IF(KODE(3).EQ.0) GO TO 48	CALC1620
	IF(KODE(3).EQ.2)WRITE(IW,43)	CALC1630
43	FORMAT(' FREQUENCY RESPONSE COMPUTATION')	CALC1640
	IF(KODE(3).EQ.2)GO TO 48	CALC1650

42	DN 27 I=1.150	C9L01660
27	AERD(I)=SAVE1(I)	C9L01670
	CALL OUTLJC	C9L01680
	IF(LL.EQ.0) GO TO 1000	C9L01690
48	CALL TRAN1	C9L01700
	IF(KODE(5).EQ.0) GO TO 49	C9L01710
	WRITE(IW,802)	C9L01720
802	FORMAT(4X,'AERD DATA IN STAB. AXIS AT EQUAT. REF. CENTER')	C9L01730
	WRITE(IW,800)(I,AERD(I),I=1,36)	C9L01740
49	CALL TRIM	C9L01750
	CALL TRANS	C9L01760
	IF(KODE(5).EQ.0) GO TO 50	C9L01770
	WRITE(IW,803)	C9L01780
803	FORMAT(4X,'AERD DATA IN BODY AXIS AT EQUAT. REF. CENTER')	C9L01790
	WRITE(IW,804)(I,AERD(I),I=1,36)	C9L01800
804	FORMAT(5(2X,'AERDP(',I3,')='',G10,3))	C9L01810
50	IF(KODE(2)) 70,80,90	C9L01820
70	WRITE(IW,700)	C9L01830
700	FORMAT(' ++++ LONGITUDINAL STABILITY ++++')	C9L01840
	IF(KODE(14).EQ.0)GO TO 702	C9L01850
	IDX=KODE(14)	C9L01860
	IDN=KODE(15)	C9L01870
	IF(KODE(13).NE.-1.)GO TO 706	C9L01880
	IF(KODE(15).EQ.3.)IDN=4	C9L01890
	IF(KODE(15).LE.3.)GO TO 706	C9L01900
	KODE(15)=3.	C9L01910
	WRITE(IW,707)	C9L01920
707	FORMAT(3X,'KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION,KODE(1	C9L01930
	5) IS SET TO 3.')	C9L01940
706	WRITE(IW,701)NAME(IDN),NAME(IDX)	C9L01950
701	FORMAT(' COMPUTATION OF ',A4,'/',A4,' NUMERATOR ROOTS')	C9L01960
702	CALL LONG	C9L01970
	IF(KODE(3).EQ.1) GO TO 42	C9L01980
	GO TO 1000	C9L01990
80	WRITE(IW,750)	C9L02000
750	FORMAT(' ++++ LATERAL/DIRECTIONAL STABILITY ++++')	C9L02010
	IF(KODE(16).EQ.0)GO TO 703	C9L02020
	IDX=KODE(16)	C9L02030
	IDN=KODE(17)	C9L02040
	WRITE(IW,701)NAME1(IDN),NAME1(IDX)	C9L02050
703	CALL LAT	C9L02060
	IF(KODE(3).EQ.1) GO TO 42	C9L02070
	GO TO 1000	C9L02080
90	WRITE(IW,700)	C9L02090
	IF(KODE(14).EQ.0) GO TO 704	C9L02100
	IDX=KODE(14)	C9L02110
	IDN=KODE(15)	C9L02120
	IF(KODE(13).NE.-1.)GO TO 708	C9L02130
	IF(KODE(15).EQ.3.)IDN=4	C9L02140
	IF(KODE(15).LE.3.)GO TO 708	C9L02150
	KODE(15)=3.	C9L02160
	WRITE(IW,707)	C9L02170
708	WRITE(IW,701)NAME(IDN),NAME(IDX)	C9L02180
704	CALL LONG	C9L02190
	WRITE(IW,750)	C9L02200

```

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 OUTLOC
COMMON/INOUT/IW,IP
COMMON/DAT/AERO(175),AEROP(50),KODE(26),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,)'1
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=1
RETURN
END
BLOCK DATA
COMMON/ANAME/NAME(16),NAME1(16)
DATA NAME/' Z ',' THET',' DTF',' X ',' DTFB',' THTM',' EMT',
:' THMD',' EM ',' DTC',' EMD',' DT ',' THTD',' ',
2'DELE',' ALEF'/,NAME1/' Y ',' PSI',' PHI',' DTFB',' PSIM',
2' EMT',' PSMD',' PSID',' EM ',' DTC',' EMD',' DT ',' ',
3'DELR',' DELA',' BETG'/
END
SUBROUTINE FREQ (ROOTS,K4A,TEG)
COMMON/INOUT/IW,IP
COMMON /DAT/AERO(175),AEROP(50),KODE(26),LL
COMMON/PLOT/DM(61),AMP(61),ANGLE(61),XMP(61),KV
COMMON/ANAME/NAME(16),NAME1(16)
COMPLEX ROOTS(1)
COMPLEX CNU(29)
DIMENSION DM(21)
DATA DM/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.7.)IN2=4
IF(KODE(15).LE.3.)GO TO 32

```

KODE(15)=3.	CBL02750
WRITE(IW,707)	CBL02770
707 FORMAT(3X,'KODE(15) IS INCORRECT FOR CABLELESS MODEL OPTION,KODE(1	CBL02780
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)=OM((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,'/',2X,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 FFORMAT('/',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
1 ANGLE(30+I),XMP(30+I)                                                   CBL03450
8 WRITE(IW,14)                                                            CBL03460
14 FORMAT(1H1)                                                            CBL03470
   RETURN                                                                    CBL03480
   ENTRY FREQ1(ROOTS,KAA,TFG)                                              CBL03490
   KN=KAA                                                                    CBL03500
   TGN=TFG                                                                    CBL03510
   IF(KN.EQ.0)RETURN                                                       CBL03520
   DO 20 I=1,KAA                                                           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
   XPL=-CXU(1,I)                                                            CBL03660
   YIM=OM-CXU(2,I)                                                         CBL03670
   AMP=SQRT(XPL*XPL+YIM*YIM)*AMP                                           CBL03680
   IF(XPL.EQ.0..AND.YIM.EQ.0.)GO TO 2                                       CBL03690
   ANG=ATAN2(YIM,XPL)+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,XPL,YIM)                                       CBL03760
   END                                                                        CBL03770
   SUBROUTINE TRANS                                                         CAB00010
C THIS ROUTINE CALCULATES BODY AXIS AERO DATA AT CR FROM STAR.          CAB00020
C AXIS AERO DATA AT CR                                                  CAB00030
COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL                               CAB00040
EQUIVALENCE(AERO( 1), CDU),(AERO( 2), CLU),(AERO( 3), CMU),              CAB00050
1 (AERO( 4), CDA),(AERO( 5), CLA),(AERO( 6), CMA),                      CAB00060
2 (AERO( 7), CDO),(AERO( 8), CLO),(AERO( 9), CM2),                      CAB00070
3 (AERO(10), CDD),(AERO(11), CLD),(AERO(12), CMD),                      CAB00080

```

4	(AERO(13), CDDF), (AERO(14), CLDF), (AERO(15), CMDF),	CAB00090
5	(AERO(16), CDAD), (AERO(17), CLAD), (AERO(18), CMAD),	CAB00100
6	(AERO(19), CYR), (AERO(20), CLB), (AERO(21), CNB),	CAB00110
7	(AERO(22), CYP), (AERO(23), CLP), (AERO(24), CNP),	CAB00120
8	(AERO(25), CYF), (AERO(26), CLF), (AERO(27), CNF),	CAB00130
9	(AERO(28), CYD), (AERO(29), CLD), (AERO(30), CND),	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), CXDP), (AEROP(11), CZDP), (AEROP(12), CMDP),	CAB00220
4	(AEROP(13), CXDF), (AEROP(14), CZDF), (AEROP(15), CMDF),	CAB00230
5	(AEROP(16), CXAD), (AEROP(17), CZAD), (AEROP(18), CMAD),	CAB00240
6	(AEROP(19), CYBP), (AEROP(20), CLB), (AEROP(21), CNB),	CAB00250
7	(AEROP(22), CYP), (AEROP(23), CLP), (AEROP(24), CNP),	CAB00260
8	(AEROP(25), CYF), (AEROP(26), CLF), (AEROP(27), CNF),	CAB00270
9	(AEROP(28), CYD), (AEROP(29), CLD), (AEROP(30), CND),	CAB00280
A	(AEROP(31), CYDA), (AEROP(32), CLDA), (AEROP(33), CNDA),	CAB00290
B	(AEROP(34), CYDS), (AEROP(35), CLDS), (AEROP(36), CNDS)	CAB00300
	ALPHA=THETA	CAB00310
	SNALF= SIN(ALPHA)	CAB00320
	COALF= COS(ALPHA)	CAB00330
	SNSQ = SNALF**2	CAB00340
	COSQ = COALF**2	CAB00350
	SNCD = SNALF*COALF	CAB00360
	CDU=CDU+2.*(CDD+CDA*THETA)	CAB00370
	CLU=CLU+2.*(CLD+CLA*THETA)	CAB00380
	CDA=CDA-(CLD+CLA*THETA)	CAB00390
	CLA=CLA+CDD+CDA*THETA	CAB00400
	CXUP=-CLA*SNSQ-CDU*COSQ+(CDA+CLU)*SNCD	CAB00410
	CZUP= CDA*SNSQ-CLU*COSQ+(CLA-CDU)*SNCD	CAB00420
	CMUP= -CMA *SNALF+ CMU *COALF	CAB00430
	CXAP= CLU*SNSQ-CDA*COSQ+(CLA-CDU)*SNCD	CAB00440
	CZAP=-CDU*SNSQ-CLA*COSQ-(CDA+CLU)*SNCD	CAB00450
	CMAP= CMU *SNALF+ CMA *COALF	CAB00460
	CXQP= CLQ*SNALF-CDQ*COALF	CAB00470
	CZQP=- (CDQ*SNALF+CLQ*COALF)	CAB00480
	CMQP= CMQ	CAB00490
	CZAD=-CLAD*COALF+CDAD*SNALF	CAB00500
	CXAD=-CDAD*COALF-CLAD*SNALF	CAB00510
	CMAD= CMAD	CAB00520
	CXDF= CLDF*SNALF-CDDF*COALF	CAB00530
	CZDF=-CDF*SNALF-CLDF*COALF	CAB00540
	CMDF= CMDF	CAB00550
	CXAD=-CDQ*COALF-CLQ*SNALF	CAB00560
	CZAD=-CLQ*COALF+CDQ*SNALF	CAB00570
	CMDF= CMQ	CAB00580
	CYBP= CYB	CAB00590
	CNBP= CLB *SNALF+ CNB *COALF	CAB00600
	CLBP= -CNB *SNALF+ CLB *COALF	CAB00610
	CYP= (-CYF*SNALF+ CYP*COALF)	CAB00620
	CNPP= (-CLF*SNALF+ CNP*COSQ+ (CLP- CNF)*SNCD)	CAB00630

```

CLPP=( CNP*SNSQ+ CLP*CO SQ+ (CLP+ CNP)*SNCO) CAB00640
CYRP=( CYP*SNALF+ CYR*COALF) CAB00650
CNRP=( CLP*SNSQ+ CNP*CO SQ+ (CLP+ CNP)*SNCO) CAB00660
CLRP=(-CNP*SNSQ+ CLP*CO SQ+ (CLP- CNP)*SNCO) CAB00670
CYDA= CYDA CAB00680
CNDAP= CLDA*SNALF+ CNDA*COALF CAB00690
CLDAP= -CNDA*SNALF+ CLDA*COALF CAB00700
CYDRP= CYDR CAB00710
CNDRP= CLDR*SNALF+ CNDR*COALF CAB00720
CLDRP=-CNDR*SNALF+ CLDR*COALF CAB00730
CYDSP= CYDS CAB00740
CLDSP=-CNDS*SNALF+ CLDS*COALF CAB00750
CNDSP= CLDS*SNALF+ CNDS*COALF CAB00760
RETURN CAB00770
END CAB00780
SUBROUTINE TRAN1 CAB00790
C THIS ROUTINE TRANSFORMS INERTIA DATA & STABILITY AXIS AERO DATA CAB00800
C TO THE EQUATION REFERENCE CENTER CAB00810
COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL CAB00820
EQUIVALENCE(AERO( 1), CDU),(AERO( 2), CLU),(AERO( 3), CMU), CAB00830
1 (AERO( 4), CDA),(AERO( 5), CLA),(AERO( 6), CMA), CAB00840
2 (AERO( 7), CDQ),(AERO( 8), CLQ),(AERO( 9), CMQ), CAB00850
3 (AERO(10), CDG),(AERO(11), CLG),(AERO(12), CMG), CAB00860
4 (AERO(13), CODE),(AERO(14), CLDE),(AERO(15), CMDE), CAB00870
5 (AERO(16), CDAD),(AERO(17), CLAD),(AERO(18), CMAD), CAB00880
6 (AERO(19), CYB),(AERO(20), CLB),(AERO(21), CNB), CAB00890
7 (AERO(22), CYP),(AERO(23), CLP),(AERO(24), CNP), CAB00900
8 (AERO(25), CYR),(AERO(26), CLR),(AERO(27), CNR), CAB00910
9 (AERO(28), CYDR),(AERO(29), CLDR),(AERO(30), CNDR), CAB00920
A (AERO(31), CYDA),(AERO(32), CLDA),(AERO(33), CNDA), CAB00930
B (AERO(34), CYDS),(AERO(35), CLDS),(AERO(36), CNDS), CAB00940
C (AERO(44), XREF),(AERO(45), ZREF),(AERO(46), XCG), CAB00950
D (AERO(47), ZCG),(AERO(63),THETA) CAB00960
EQUIVALENCE(AERO(42),AMACH),(AERO(49),VO ),(AERO(50), AM) CAB00970
EQUIVALENCE(AERO(51),RHO ),(AERO(52), WT),(AERO(53),B ) CAB00980
EQUIVALENCE(AERO(54),CBAR ),(AERO(55),SW ),(AERO(56), XIXZ) CAB00990
EQUIVALENCE(AERO(57),XIXX ),(AERO(58),YIYY ),(AERO(59),ZIZZ ) CAB01000
EQUIVALENCE(AERO(60),CLT ),(AERO(61),CDT ),(AERO(62),CMT ) CAB01010
C INERTIA TRANSFORMATIONS CAB01020
X=XCG/12. CAB01030
Z=ZCG/12. CAB01040
XIXX=XIXX+AM*(Z**2) CAB01050
YIYY=YIYY+AM*(X**2)+AM*(Z**2) CAB01060
ZIZZ=ZIZZ+AM*(X**2) CAB01070
XIXZ=XIXZ-AM*X*Z CAB01080
C AERO DATA TRANSFORMATIONS CAB01090
X=XREF/(12.*CBAR) CAB01100
Z=ZREF/(12.*CBAR) CAB01110
CMQ=CMQ-Z*CDQ+X*CLQ CAB01120
CMQ=CMQ-X*(-CLQ+2.*CMA)-2.*X*X*CLA-Z*CDQ+2.*X*Z=CDA CAB01130
CLQ=CLQ-2.*X*CLA+4.*Z*CLD CAB01140
CDQ=CDQ-2.*X*CDA+4.*Z*CDG CAB01150
CMA=CMA-Z*CDA+X*CLA CAB01160
CMDE=CMDE-Z*CODE+X*CLDE CAB01170
X=XREF/(12.*B) CAB01180

```

Z=ZREF/(12.*8)	CABC1190
CNR=CNR+X*(2.*CNR+CYR+2.*X*CYB)	CABC1200
CLR=CLR+X*(CLB-Z*CYB)-7*CYR	CABC1210
CNP=CNP-2.*Z*(CNR+X*CYB)+X*CYP	CABC1220
CLP=CLP-7*(CYP-2.*Z*CYB)-2.*Z*CLB	CABC1230
CYR=CYR+2.*X*CYB	CABC1240
CYP=CYP-2.*Z*CYB	CABC1250
CNB=CNR+X*CYB	CABC1260
CNDR=CNDR+X*CYDR	CABC1270
CNDA=CNDA+X*CYDA	CABC1280
CNDS=CNDS+X*CYDS	CABC1290
CLB=CLR-7*CYR	CABC1300
CLDR=CLDR-Z*CYDR	CABC1310
CLDA=CLDA-7*CYDA	CABC1320
CLDS=CLDS-Z*CYDS	CABC1330
RETURN	CABC1340
END	CABC1350
SUBROUTINE LATSIN	CABC0010
COMMON/INDUT/IW,IS	CABC0020
COMMON/DAT/AERD(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(AERD(105), SNUX),(AERD(106), SNUY),(AERD(107), SNUZ),	CABC0080
1(AERD(108), SNLX),(AERD(109), SNLY),(AERD(110), SNLZ),	CABC0090
2(AERD(111), SNUST),(AERD(112), SNUWL),(AERD(113), SNUBL),	CABC0100
3(AERD(114), SNLST),(AERD(115), SNLWL),(AERD(116), SNLSL),	CABC0110
4(AERD(117), TUSND),(AERD(118), TUSND),(AERD(119), AKSNU),	CABC0120
5(AERD(120), AKSNL),(AERD(121), ADSNU),(AERD(122), ADSNL)	CABC0130
6(AERD(123), THETA),(AERD(124), ADSNU),(AERD(125), 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 TOPP(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
GY(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
GZ(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	CAB00390
DO 1005 J=1,3	CAB00400
SNJ(I,J)=0	CAB00410
1005 SNJD(I,J)=0	CAB00420
DO 1006 I=1,10	CAB00430
DO 1006 J=1,10	CAB00440
1006 DUM(I,J)=0	CAB00450
IF(KODE(10).EQ.0) GO TO 1002	CAB00460
C TERMS FOR SNUBBER EFFECTS (LAT)	CAB00470
CALL DRCSN(THETA)	CAB00480
IF(KODE(10).EQ.1) CALL DRCSN(THETA)	CAB00490
DUM(1,2) = -TUSN1*GX1	CAB00500
DUM(1,3) = TUSN1*GZ1	CAB00510
DUM(1,5) = -TUSN1*SIN(THGY1)	CAB00520
DUM(1,7) = GY1	CAB00530
DUM(2,2) = SNUX*TUSN1*GX1/12.+SNUY*TUSN1*GY1/12.	CAB00540
DUM(2,3) = -SNUX*TUSN1*GZ1/12.	CAB00550
DUM(2,4) = -SNUY*TUSN1*SIN(THGX1)/12.	CAB00560
DUM(2,5) = SNUX*TUSN1*SIN(THGY1)/12.	CAB00570
DUM(2,7) = (-SNUX*GY1+SNUY*GX1)/12.	CAB00580
DUM(3,2) = -SNUZ*TUSN1*GX1/12.	CAB00590
DUM(3,3) = SNUZ*TUSN1*GZ1/12.+SNUY*TUSN1*GY1/12.	CAB00600
DUM(3,5) = -SNUZ*TUSN1*SIN(THGY1)/12.	CAB00610
DUM(3,6) = SNUY*TUSN1*SIN(THGZ1)/12.	CAB00620
DUM(3,7) = (-SNUY*GZ1+SNUZ*GY1)/12.	CAB00630
DUM(4,1) = GXY(GY1,THGX1,ALU)	CAB00640
DUM(4,2) = GXSY(-SNUY,THGX1,-SNUX,GY1,THGX1,ALU)	CAB00650
DUM(4,3) = GXPFI(-SNUZ,GY1,THGX1,-SNUY,GZ1,THGX1,ALU)	CAB00660
DUM(4,4) = -1.	CAB00670
DUM(5,1) = GYY(THGY1,ALU)	CAB00680
DUM(5,2) = GYSY(-SNUY,GX1,THGY1,-SNUX,THGY1,ALU)	CAB00690
DUM(5,3) = GYPFI(-SNUZ,THGY1,-SNUY,GZ1,THGY1,ALU)	CAB00700
DUM(5,5) = -1.	CAB00710
DUM(6,1) = GZY(GY1,THGZ1,ALU)	CAB00720
DUM(6,2) = GZSY(-SNUY,GX1,THGZ1,-SNUX,GY1,THGZ1,ALU)	CAB00730
DUM(6,3) = GZPFI(-SNUZ,GY1,THGZ1,-SNUY,THGZ1,ALU)	CAB00740
DUM(6,6) = -1.	CAB00750
IF(KODE(10).EQ.2) GO TO 1010	CAB00760
CALL DRCSN(THETA)	CAB00770
Q=.5*RHO*V0*V0	CAB00780
ALU1=ALU+1.	CAB00790
CALL STINT(Q,ALU1,0,1,1,TUSN1,NG)	CAB00800
IF(NG.NE.0) GO TO 5000	CAB00810
ALU2=ALU-1.	CAB00820
CALL STINT(Q,ALU2,0,1,1,TUSN2,NG)	CAB00830
IF(NG.NE.0) GO TO 5000	CAB00840
GO TO 5001	CAB00850
5000 WRITE(IW,5002) NG,ALL,ALU,G	CAB00860
5002 FORMAT('ERROR IN SNUBBER TABLE 1,NG=',I2,3X'10.3')	CAB00870
RETURN	CAB00880
5001 CONTINUE	CAB00890
AKTU=(TUSN1-TUSN2)/2.	CAB00900
AKSNU=AKTU	CAB00910
1010 CONTINUE	CAB00920
DUM(7,7) = -1.	CAB00930

DUM(7,8) =	AKSNU*12.	CAB00940
DUM(8,1) =	ALY(GY1)	CAB00950
DUM(8,2) =	ALSY(-SNUY,GX1,-SNUX,GY1)	CAB00960
DUM(8,3) =	ALPHI(-SNUZ,GY1,-SNUY,GZ1)	CAB00970
DUM(8,8) =	-1.	CAB00980
IF(KODE(10).EQ.1) GO TO 1015		CAB00990
DO 1016 I=1,3		CAB01000
DO 1016 J=1,3		CAB01010
1016 SNUD(I,J)=DUM(I,7)*ADSNU*DUM(8,J)*12.		CAB01020
1015 CALL MASH(3,8)		CAB01030
DO 1050 I=1,3		CAB01040
DO 1050 J=1,3		CAB01050
1050 TOPR(I,J)= DUM(I,J)		CAB01060
IF(KODE(10).EQ.1) CALL DRCSN(THETA)		CAB01070
DUM(1,2) =	-TUSNO*GX2	CAB01080
DUM(1,3) =	TUSNO*GZ1	CAB01090
DUM(1,5) =	-TUSNO*SIN(THGY2)	CAB01100
DUM(1,7) =	GY2	CAB01110
DUM(2,2) =	SNUX*TUSNO*GX2/12.-SNUY*TUSNO*GY2/12.	CAB01120
DUM(2,3) =	-SNUX*TUSNO*GZ2/12.	CAB01130
DUM(2,4) =	SNUY*TUSNO*SIN(THGX2)/12.	CAB01140
DUM(2,5) =	SNUX*TUSNO*SIN(THGY2)/12.	CAB01150
DUM(2,7) =	(-SNUX*GY2-SNUY*GX2)/12.	CAB01160
DUM(3,2) =	-SNUZ*TUSNO*GX2/12.	CAB01170
DUM(3,3) =	SNUZ*TUSNO*GZ2/12.-SNUY*TUSNO*GY2/12.	CAB01180
DUM(3,5) =	-SNUZ*TUSNO*SIN(THGY2)/12.	CAB01190
DUM(3,6) =	-SNUY*TUSNO*SIN(THGZ2)/12.	CAB01200
DUM(3,7) =	(SNUY*GZ2+SNUZ*GY2)/12.	CAB01210
DUM(4,1) =	GXY(GY2,THGX2,ALU)	CAB01220
DUM(4,2) =	GXSX(SNUY,THGX2,-SNUX,GY2,THGX2,ALU)	CAB01230
DUM(4,3) =	GXPHI(-SNUZ,GY2,THGX2,SNUY,GZ2,THGX2,ALU)	CAB01240
DUM(4,4) =	-1.	CAB01250
DUM(5,1) =	GYI(THGY2,ALU)	CAB01260
DUM(5,2) =	GYSX(SNUY,GX2,THGY2,-SNUX,THGY2,ALU)	CAB01270
DUM(5,3) =	GYPHI(-SNUZ,THGY2,SNUY,GZ2,THGY2,ALU)	CAB01280
DUM(5,5) =	-1.	CAB01290
DUM(6,1) =	GZY(GY2,THGZ2,ALU)	CAB01300
DUM(6,2) =	GZSX(SNUY,GX2,THGZ2,-SNUX,GY2,THGZ2,ALU)	CAB01310
DUM(6,3) =	GZPHI(-SNUZ,GY2,THGZ2,SNUY,THGZ2,ALU)	CAB01320
DUM(6,6) =	-1.	CAB01330
IF(KODE(10).EQ.2) GO TO 1020		CAB01340
CALL DRCSN(THETA)		CAB01350
ALU1=ALU+1.		CAB01360
CALL STINT(0,ALU1,0,1,1,TUSN1,NG)		CAB01370
IF(NG.NE.0) GO TO 5000		CAB01380
ALU2=ALU-1.		CAB01390
CALL STINT(0,ALU2,0,1,1,TUSN2,NG)		CAB01400
IF(NG.NE.0) GO TO 5000		CAB01410
AKTU=(TUSN1-TUSN2)/2.		CAB01420
AKSNU=AKTU		CAB01430
1020 CONTINUE		CAB01440
DUM(7,7) =	-1.	CAB01450
DUM(7,8) =	AKSNU*12.	CAB01460
DUM(8,1) =	ALY(GY2)	CAB01470
DUM(8,2) =	ALSY(SNUY,GX2,-SNUX,GY2)	CAB01480

DUM(8,3) =	ALPHI(-SNUZ,GY2,SNUY,GZ2)	CABC1490
DUM(8,8) =	-1.	CABC1500
IF(KODE(10).EQ.1) GO TO 1025		CABC1510
DO 1026 I=1,3		CABC1520
DO 1026 J=1,3		CABC1530
1026 SNUD(I,J) =	SNUD(I,J)+DUM(I,7)*AGSNU*DUM(8,J)*12.	CABC1540
1025 CALL WASH(3,8)		CABC1550
DO 1060 I=1,3		CABC1560
DO 1060 J=1,3		CABC1570
1060 TOPL(I,J) =	DUM(I,J)	CABC1580
IF(KODE(10).EQ.1) CALL DRCUEN(THETA)		CABC1590
DUM(1,2) =	-TLSND*GX3	CABC1600
DUM(1,3) =	TLSND*GZ3	CABC1610
DUM(1,5) =	-TLSND*SIN(THGY3)	CABC1620
DUM(1,7) =	GY3	CABC1630
DUM(2,2) =	SNLX*TLSND*GX3/12.-SNLY*TLSND*GY3/12.	CABC1640
DUM(2,3) =	-SNLX*TLSND*GZ3/12.	CABC1650
DUM(2,4) =	SNLY*TLSND*SIN(THGX3)/12.	CABC1660
DUM(2,5) =	SNLX*TLSND*SIN(THGY3)/12.	CABC1670
DUM(2,7) =	(-SNLX*GY3-SNLY*GX3)/12.	CABC1680
DUM(3,2) =	SNLZ*TLSND*GX3/12.	CABC1690
DUM(3,3) =	-SNLZ*TLSND*GZ3/12.-SNLY*TLSND*GY3/12.	CABC1700
DUM(3,5) =	SNLZ*TLSND*SIN(THGY3)/12.	CABC1710
DUM(3,6) =	-SNLY*TLSND*SIN(THGZ3)/12.	CABC1720
DUM(3,7) =	(SNLY*GZ3-SNLZ*GY3)/12.	CABC1730
DUM(4,1) =	GXY(GY3,THGX3,ALL)	CABC1740
DUM(4,2) =	GXSX(SNLY,THGX3,-SNLX,GY3,THGX3,ALL)	CABC1750
DUM(4,3) =	GXPXI(SNLZ,GY3,THGX3,SNLY,GZ3,THGX3,ALL)	CABC1760
DUM(4,4) =	-1.	CABC1770
DUM(5,1) =	GYX(THGY3,ALL)	CABC1780
DUM(5,2) =	GYSX(SNLY,GX3,THGY3,-SNLX,THGY3,ALL)	CABC1790
DUM(5,3) =	GYPXI(SNLZ,THGY3,SNLY,GZ3,THGY3,ALL)	CABC1800
DUM(5,5) =	-1.	CABC1810
DUM(6,1) =	GZY(GY3,THGZ3,ALL)	CABC1820
DUM(6,2) =	GZSX(SNLY,GX3,THGZ3,-SNLX,GY3,THGZ3,ALL)	CABC1830
DUM(6,3) =	GZPXI(SNLZ,GY3,THGZ3,SNLY,THGZ3,ALL)	CABC1840
DUM(6,6) =	-1.	CABC1850
IF(KODE(10).EQ.2) GO TO 1030		CABC1860
CALL DRCSN(THETA)		CABC1870
ALL1=ALL+1.		CABC1880
CALL STINT(0,ALL1,C,1,1,TLN1,NG)		CABC1890
IF(NG.NE.0) GO TO 5000		CABC1900
ALL2=ALL-1.		CABC1910
CALL STINT(0,ALL2,C,1,1,TLN2,NG)		CABC1920
IF(NG.NE.0) GO TO 5000		CABC1930
AKTL=(TLN1-TLN2)/2.		CABC1940
AKSNL=AKTL		CABC1950
1030 CONTINUE		CABC1960
DUM(7,7) =	-1.	CABC1970
DUM(7,8) =	AKSNL*12.	CABC1980
DUM(8,1) =	ALY(GY3)	CABC1990
DUM(8,2) =	ALSY(SNLY,GX3,-SNLX,GY3)	CABC2000
DUM(8,3) =	ALPHI(SNL7,GY3,SNLY,GZ3)	CABC2010
DUM(8,8) =	-1.	CABC2020
IF(KODE(10).EQ.1) GO TO 1035		CABC2030

```

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 WASH(3,8)
-----
DO 1070 I=1,3
DO 1070 J=1,3
1070 POTL(I,J)= DUM(I,J)
IF(KODE(10).EQ.1) CALL DRCSN(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) = GXSX(-SNLY,THGX4,-SNLX,GY4,THGX4,ALL)
DUM(4,3) = GXPXI(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) = GYPXI(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) = GZPXI(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(0,ALL1,C,1,1,TLN1,NG)
IF(NG.NE.0) GO TO 5000
ALL2=ALL-1.
CALL STINT(0,ALL2,C,1,1,TLN2,NG)
IF(NG.NE.0) GO TO 5000
AKTL=(TLN1-TLN2)/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(8,8) = -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 1090 I=1,3 CAB02600
      DO 1090 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/INJUT/IW,IP CAB00030
COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL CAB00040
COMMON / PLYCHA/RTC,XLGTH(5),ACC(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), CDE),(AERO(14), CLDE),(AERO(15), CMDE), CAB00110
5 (AERO(16), CDDA),(AERO(17), CLDA),(AERO(18), CMDA), 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),PHO ),(AERO(52), WT),(AERO(53),R ) 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), WLUF), CAB00260
1 (AERO(69), WLLR),(AERO(70), WLHF),(AERO(71), WLHF), CAB00270
2 (AERO(72), STAF),(AERO(73), STAR),(AERO(74), RLHF), CAB00280
3 (AERO(75), RLHF),(AERO(76), WLCR),(AERO(77), STACR), CAB00290
4 (AERO(78), BLCF),(AERO(79), EF),(AERO(80), EF), CAB00300
5 (AERO(81), AF),(AERO(82), AR),(AERO(83), HUCF), CAB00310
6 (AERO(84), HLCF),(AERO(85), HUCF),(AERO(86), HLCF), CAB00320
7 (AERO(87), DCF),(AERO(88), DCF),(AERO(89), ALF), CAB00330
8 (AERO(90), RVF),(AERO(91), RHF),(AERO(92), RVR), CAB00340
9 (AERO(93), RHR),(AERO(94), TRQ),(AERO(95), AKQ), CAB00350
A (AERO(96), ALFO),(AERO(97),STLTT),(AERO(98),WLLTT), CAB00360
B (AERO(99),TLFTC),(AERO(100),AKLFT),(AERO(101),ALLTC), CAB00370

```

```

C      (AEROP(102),ALTZ),(AEROP(103),ALTZ)          CAB00380
EQUIVALENCE(AEROP( 1),CXUP),(AEROP( 2),CZUP),(AEROP( 3),C1UP), 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),CXOP),(AEROP(11),CZOP),(AEROP(12),CMOP), CAB00420
4      (AEROP(13),CXDP),(AEROP(14),CZDP),(AEROP(15),CMDP), CAB00430
5      (AEROP(16),CXADP),(AEROP(17),CZADP),(AEROP(18),CMADP), CAB00440
6      (AEROP(19),CYSP),(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),CYDP),(AEROP(29),CLDP),(AEROP(30),CNDP), 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=C.            CAB00520
DELALF=.001        CAB00530
DTF=.1             CAB00540
DALFAW=C.0        CAB00550
DDELTE=C.0        CAB00560
DTHRST=C.0        CAB00570
ICNTR=C           CAB00580
FIRST=C.          CAB00590
THINT=C.          CAB00600
ALFINT=THETA      CAB00610
DELINT=C.         CAB00620
THRST=THINT       CAB00630
1  IF(VJ.EQ.0.)THRST=-TR*(COS(ADC(3,1))+COS(ADC(4,1)))/(COS(ADC(1,1)
1)+COS(ADC(2,1))) CAB00640
VAL5=COS(ADC(3,1)) CAB00650
VAL6=COS(ADC(4,1)) CAB00660
VAL7=COS(ADC(1,1)) CAB00670
VAL8=COS(ADC(2,1)) CAB00680
ALFAW=ALFINT      CAB00690
DELTE=DELINT      CAB00700
QS=RHO*VO*VO*.5*SW CAB00720
209 THRSTI=THRST+DTHRST CAB00730
ALFAWI=ALFAW+DALFAW CAB00740
DELTEI=DELTE+DDELTE CAB00750
ICNTRI=ICNTR+1   CAB00760
IF(ICNTRI.GT.100)GO TO 520 CAB00770
VAL1=ALFAWI*RTD  CAB00780
VAL2=DELTEI*RTD  CAB00790
VAL3=THRSTI      CAB00800
CALL EQU(ALFAWI,DELTEI,THRSTI,F0,GO,H0,FIRST) CAB00810
IF (VD.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+C.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+C.5 CAB00910
ALFWO=(F1-F2)/DELALF CAB00920

```

```

GALFWD=(G1-G2)/DELALF
HALFWD=(H1-H2)/DELALF
FDELED=-QS*(CLDE*COS(ALFAWI)+CDDE*SIN(ALFAWI))
GOELED=QS*(CLDE*SIN(ALFAWI)-CDDE*COS(ALFAWI))
HDELED=QS*CBAR*CMDE
THRSTI=THRSTI+DTF
CALL EQU(ALFAWI,DELTEI,THRSTI,F1,G1,H1,1.)
THRSTI=THRSTI-2.*DTF
CALL EQU(ALFAWI,DELTEI,THRSTI,F2,G2,H2,1.)
THRSTI=THRSTI+DTF
FTHSTD=(F1-F2)/(DTF*2.)
GTHSTD=(G1-G2)/(DTF*2.)
HTHSTD=(H1-H2)/(DTF*2.)
C SET UP ITERATION EQUATIONS
FI=FO+GALFWD*DALFAW+FDELED*DDELTE+FTHSTD*DTHRST
GI=GO+HALFWD*DALFAW+GOELED*DDELTE+GTHSTD*DTHRST
HI=HO+HALFWD*DALFAW+HDELED*DDELTE+HTHSTD*DTHRST
ACCZ=FI/AM
ACCX=GI/AM
THEDDT=HI/YIYV
IF(VD.EQ.0.)GO TO 42
IF(ABS(ACCZ).LT..01)GO TO 1005
GO TO 1100
1005 IF(ABS(ACCX).LT..01)GO TO 1007
GO TO 1100
1007 IF(ABS(THEDDT).LE.0.001)GO TO 42
C NOW COMPUTE PARAMETER INCREMENTS FROM MATRIX EQUATIONS
1100 DETRM=FALEWD*GOELED*HTHSTD+FDELED*GTHSTD*HALFWD+FTHSTD*GALFWD*
:HDELED-FTHSTD*GOELED*HALFWD-FALEWD*GTHSTD*HDELED-FDELED*GALFWD*
2HTHSTD
DALFAW=(-(GOELED*HTHSTD-GTHSTD*HDELED)*FO+(FDELED*HTHSTD-FTHSTD
1*HDELED)*GO-(FDELED*GTHSTD-FTHSTD*GOELED)*HO)/DETRM
DDELTE=(+(GALFWD*HTHSTD-GTHSTD*HALFWD)*FO-(FALEWD*HTHSTD-HALFWD
1*FTHSTD)*GO+(FALEWD*GTHSTD-FTHSTD*GALFWD)*HO)/DETRM
DTHRST=(-(GALFWD*HDELED-GDELED*HALFWD)*FO+(FALEWD*HDELED-FDELED
1*HALFWD)*GO-(FALEWD*GOELED-FDELED*GALFWD)*HO)/DETRM
THRSTD=THRSTI
ALFAWD=ALFAWI
DELTEO=DELTEI
GO TO 209
520 WRITE(IW,521)
521 FORMAT(' TRIM ITERATION EXCEEDS LIMITS')
GO TO 522
42 CALL EQU(ALFAWI,DELTEI,THRSTI,FC,GO,HO,1.)
522 DO 523 IZZ=1,4
DO 523 IZK=1,3
ANG(IZZ,IZK)=ADC(IZZ,IZK)*FTD
523 CONTINUE
THETA=ALFAWI
DE=DELTEI
TE=THRSTI
THETD=THETA*RTD
DED=DE*RTD
DO 524 IZZ=1,4
IF(KODE(5).EQ.0) GO TO 528

```

CABC0930
CABC0940
CABC0950
CABC0960
CABC0970
CABC0980
CABC0990
CABC1000
CABC1010
CABC1020
CABC1030
CABC1040
CABC1050
CABC1060
CABC1070
CABC1080
CABC1090
CABC1100
CABC1110
CABC1120
CABC1130
CABC1140
CABC1150
CABC1160
CABC1170
CABC1180
CABC1190
CABC1200
CABC1210
CABC1220
CABC1230
CABC1240
CABC1250
CABC1260
CABC1270
CABC1280
CABC1290
CABC1300
CABC1310
CABC1320
CABC1330
CABC1340
CABC1350
CABC1360
CABC1370
CABC1380
CABC1390
CABC1400
CABC1410
CABC1420
CABC1430
CABC1440
CABC1450
CABC1460
CABC1470

```

WRITE(IW,525)IZZ,XLGTH(IZZ),(ANG(IZZ,IZK),ARM(IZZ,IZK),IZK=1,3) CAB01480
525 FORMAT(' CABLE GEOMETRY-CABLE NO.',I2,5X,'CABLE LENGTH=',E15.6, CAB01490
1' IN',/,3X,' DIR. COS.=DEG ARM-IN',/,3(3X,2E15.6,/),/) CAB01500
524 CONTINUE CAB01510
IF(VD.EQ.0.)WRITE(IW,529) CAB01520
529 FORMAT(' COMPUTATION OF WIND OFF CONDITION,TRIM ROUTINE NOT USED') CAB01530
WRITE(IW,526)ICNTR,ACCZ,ACCX,THEDOT CAB01540
526 FORMAT(' ITERATION PARAMETER =',I5,/,2X,'ACCZ =',E15.8, CAB01550
1/,2X,'ACCX =',E15.8,/,2X,'THEDOT=',E15.8,' RAD/SEC') CAB01560
528 WRITE(IW,527)THETA,DED,TF,TR CAB01570
527 FORMAT(/,' VEH. ATT.,DEFLTN,& CABLE TENSION',/, CAB01580
12X,' THETA =',F6.2,' DEG',/,2X,' DELTA =',F6.2,' DEG',/,2X
2,' FRT CAB. TENSION=',E15.6,' LBS',/, CAB01590
32X,' RE CAB. TENSION =',E15.6,' LBS') CAB01600
RETURN CAB01620
C DEBUG UNIT(3),INIT(VAL1,VAL2,VAL3,FI,GI,HI, CAB01630
C 1FALEWD,GALEWD,HALFWD,FDELEQ,GDELEQ,HDELEQ, CAB01640
C 2FTSTQ,GTHSTQ,HTHSTQ,DALFAW,DDELTE,DTHRST, CAB01650
C 3ACCZ,ACCX,THEDOT,TF,VAL5,VAL6,VAL7,VAL8) CAB01660
END CAB01670
SUBROUTINE EQU(THETA,DE,TF,FF,GG,HH,FIRST) CAB00010
C CABLE SUSPENSION SYSTEM TRIM EQUATIONS CAB00020
COMMON/INPUT/IW,IF CAB00030
COMMON /DAT/ AERO(175),AEROP(50),KODE(26),LL CAB00040
COMMON /FLYCHA/RTD,XLGTH(5),ACC(5,3),ARM(5,3),TR,TLFT,DUMMY CAB00050
REAL*8 XNM1,XNM2,YNM1,YNM2 CAB00060
EQUIVALENCE(AERO(1), COU),(AERO(2), CLU),(AERO(3), CMJ), CAB00070
1 (AERO(4), CDA),(AERO(5), CLA),(AERO(6), CMA), CAB00080
2 (AERO(7), COQ),(AERO(8), CLO),(AERO(9), CMQ), CAB00090
3 (AERO(10), COO),(AERO(11), CLO),(AERO(12), CMO), CAB00100
4 (AERO(13), COBE),(AERO(14), CLDE),(AERO(15), CMDE), CAB00110
5 (AERO(16), COAD),(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), CYF),(AERO(26), CLF),(AERO(27), CNF), CAB00150
9 (AERO(28), CYDF),(AERO(29), CLDF),(AERO(30), CNDF), 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),VD ),(AERO(50), AM) CAB00200
EQUIVALENCE(AERO(51),PHO ),(AERO(52), WT),(AERO(53),B ) CAB00210
EQUIVALENCE(AERO(54),CBAR ),(AERO(55),SW ),(AERO(56), XIX7) CAB00220
EQUIVALENCE(AERO(57),XIXY ),(AERO(58),YIYY ),(AERO(59),ZIZZ ) CAB00230
EQUIVALENCE(AERO(60),CLT ),(AERO(61),COT ),(AERO(62),CMT ) CAB00240
EQUIVALENCE(AERO(66), WLUF),(AERO(67), WLLF),(AERO(68), WLUF), CAB00250
1 (AERO(69), WLLF),(AERO(70), WLHF),(AERO(71), WLHF), CAB00260
2 (AERO(72), STAF),(AERO(73), STAR),(AERO(74), HLHF), CAB00270
3 (AERO(75), BLHF),(AERO(76), WLCR),(AERO(77), STACR), CAB00280
4 (AERO(78), BLCR),(AERO(79), EF),(AERO(80), ER), CAB00290
5 (AERO(81), AF),(AERO(82), AR),(AERO(83), HUCF), CAB00300
6 (AERO(84), HLCF),(AERO(85), HUCR),(AERO(86), HLCR), CAB00310
7 (AERO(87), DCF),(AERO(88), DCR),(AERO(89), ALF), CAB00320
8 (AERO(90), RVF),(AERO(91), RHF),(AERO(92), RVR), CAB00330
9 (AERO(93), RHR),(AERO(94), TRD),(AERO(95), AKR), CAB00340
A (AERO(96), ALFC),(AERO(97), STLT), (AERO(98), WLLT), CAB00350

```

B	(AERO(99),TLFT0),(AERO(100),AKLFT),(AERO(101),ALLT0),	CAB00360
C	(AERO(102),ALTZ),(AERO(103),ALTZ)	CAB00370
	DATA XNM1,XNM2 /'VERTICAL', 'HORIZONTAL'/	CAB00380
	RTD=57.2958	CAB00390
	VAL1=THETA	CAB00400
	Q = RHO*VD*VD/2.C	CAB00410
64	IND=KODE(5)	CAB00420
	GO TO (501,502,503,504),IND	CAB00430
501	YNM1=XNM1	CAB00440
	YNM2=XNM2	CAB00450
	CALL FPLYV(STAF,WLUF,WLLF,HUCF,HLCF,EF,RVF,THETA,1)	CAB00460
	CALL RPLYH(STAF,RLHF,WLHF,-AF,DCF,C.,RHF,THETA,3)	CAB00470
	GO TO 505	CAB00480
502	YNM1=YNM2	CAB00490
	YNM2=XNM1	CAB00500
	CALL FPLYH(STAF,RLHF,WLHF,AF,DCF,C.,RHF,THETA,1)	CAB00510
	CALL FPLYV(STAF,WLUF,WLLR,HUCF,HLCF,EF,RVF,THETA,3)	CAB00520
	GO TO 505	CAB00530
503	YNM1=XNM1	CAB00540
	YNM2=XNM1	CAB00550
	CALL FPLYV(STAF,WLUF,WLLF,HUCF,HLCF,EF,RVF,THETA,1)	CAB00560
	CALL FPLYV(STAF,WLUR,WLLF,HUCF,HLCF,ER,RVP,THETA,3)	CAB00570
	GO TO 505	CAB00580
504	YNM1=XNM2	CAB00590
	YNM2=XNM2	CAB00600
	CALL RPLYH(STAF,RLHF,WLHF,AF,DCF,C.,RHF,THETA,1)	CAB00610
	CALL RPLYH(STAF,RLHF,WLHF,-AF,DCF,C.,RHF,THETA,3)	CAB00620
505	IF(KODE(11))506,507,506	CAB00630
506	WLLT = WLCR + ALTZ*SIN(THETA) - ALTZ*COS(THETA)	CAB00640
	STALT = STACR - ALTZ*COS(THETA) - ALTZ*SIN(THETA)	CAB00650
	XLGTH(5) = SQRT((WLLT - WLLT)**2 + (STLT - STALT)**2)	CAB00660
	IF(FIRST.NE.C.)GO TO 12	CAB00670
	ELLO=XLGTH(5)	CAB00680
12	ELL=XLGTH(5)	CAB00690
	TLFT = TLFT0+AKLFT*(ELL-ELLO)	CAB00700
	ARM(5,1)=ALTX	CAB00710
	ARM(5,2)=C	CAB00720
	ARM(5,3)=ALTZ	CAB00730
	FXLTT = (TLFT*(STALT - STLT))/XLGTH(5)	CAB00740
	FZLTT = (TLFT*(WLLT - WLLT))/XLGTH(5)	CAB00750
	FXLTB = FXLTT*COS(THETA) - FZLTT*SIN(THETA)	CAB00760
	FZLTB = FZLTT*COS(THETA) + FXLTT*SIN(THETA)	CAB00770
	YMLFT = (FXLTB*ALTZ - FZLTB*ALTX)/12.	CAB00780
	ADC(5,1)=ARCOS(FXLTB/TLFT)	CAB00790
	ADC(5,2)=3.14159/2.	CAB00800
	ADC(5,3)=ARCOS(FZLTB/TLFT)	CAB00810
	GO TO 509	CAB00820
507	FXLTB=C.	CAB00830
	FZLTB=C.	CAB00840
	YMLFT=C.	CAB00850
	XLGTH(5)=C.	CAB00860
	TLFT=C.	CAB00870
	DO 13 IA=1,3	CAB00880
	ARM(5,IA)=C.	CAB00890
	ADC(5,IA)=C.	CAB00900

13 CONTINUE	CAB00910
508 CALL SNTRM(FXSN,FZSN,EMSN,THETA)	CAB00920
IF (FIRST.NF.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 ',A9,' AND REAR CABLE IS ',A9)	CAB00970
512 ELQ=XLGTH(3)+XLGTH(4)	CAB00980
510 EL=XLGTH(3)+XLGTH(4)	CAB00990
TR=TFQ+AKR*(EL-ELQ)	CAB01000
ELIFT=Q*SW*(CLD+CLA*THETA+CLDE*DE)	CAB01010
ADRAG=Q*SW*(CDD+CDA*THETA+CDDE*DE)	CAB01020
FXAIP=-ADRAG*COS(THETA)+ELIFT*SIN(THETA)	CAB01030
FZAIP=-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
EMDC=0.	CAB01120
DO 511 I=1,4	CAB01130
TENS=TF	CAB01140
IF(I.GT.2) TENS=TP	CAB01150
EMDC=EMDC+TENS*(COS(ADC(I,1))*ARM(I,3)-COS(ADC(I,3))*ARM(I,1))	CAB01160
511 CONTINUE	CAB01170
EMDC=EMDC/12.	CAB01180
AERDM=Q*SW*CBAR*(CMD+CM1*THETA+CMDE*DE)	CAB01190
FF=FZCFH+FZCR+FZLTR+FZSN+WGTZ+FZAIP	CAB01200
GG=FXCFH+FXCR+FXLTR+FXSN+WGTZ+FXAIP	CAB01210
HH=EMDC+YMLFT+EMSN+EMWGT+AERDM	CAB01220
RETURN	CAB01230
END	CAB01240
SUBROUTINE FPLYV(STAV,WLU,WLL,HHU,HHL,EP,PAO,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,TLET,TF	CAB00030
EQUIVALENCE (AERO(76),WLCF),(AERO(77),STACF),(AERO(78),BLCF)	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= WLCF +SQRT(T1)*SIN(T2)	CAB00100
T3= WLU -WLUC	CAB00110
T4= ABS(STACF -STAV) -SQRT(T1)*COS(T2)	CAB00120
XLUP= SQRT(T3*T3+T4*T4)	CAB00130
XLU= SQRT(XLUP*XLUP -PAO*PAO)	CAB00140
BUP= ATAN(T3/T4)	CAB00150
DRU= ATAN(PAO/XLU)	CAB00160
RTAU=(RUP -DRU)*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
T9= 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)=-HHL+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)=-HHL-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
YB=XWT*COS(THETA)-ZWT*SIN(THETA)	CAB00690
ZB=XWT*SIN(THETA)+ZWT*COS(THETA)	CAB00700
T9= BLD -YP	CAB00710
T10=X9-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-ZD
XL=SQRT(XLHI*XLHI+T11*T11)
TH1)=T10-FAD*COS(BHI)
TH9=T0-FAD*SIN(BHI)
IF(IF.EQ.3)G7 TO 3
XLGTH(1)=XL
XLGTH(2)=XL
ADC(1,1)=ARCOS(TH1C/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(TH1C/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/RTO,XLGTH(5),ADC(5,3),AFM(5,3),TR,TLFT,TF
IF(,DX,WF,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 EOS FOR X-Z-THETA COEFF.
COMMON/PLYCHA/RTO,XLGTH(5),ADC(5,3),AFM(5,3),TR,TLFT,TF
CX1=SIN(ADC(IC,1))/XLGTH(IC)*12.
IF(ABS(ADC(IC,3)-3.14159).GT..0001) GO TO 2

```

CAB00770
CAB00780
CAB00790
CAB00800
CAB00810
CAB00820
CAB00830
CAB00840
CAB00850
CAB00860
CAB00870
CAB00880
CAB00890
CAB00900
CAB00910
CAB00920
CAB00930
CAB00940
CAB00950
CAB00960
CAB00970
CAB00980
CAB00990
CAB01000
CAB01010
CAB01020
CAB01030
CAB01040
CAB01050
CAB01060
CAB01070
CAB01080
CAB01090
CAB01100
CAB01110
CAB01120
CAB01130
CAB01140
CAB01150
CAB01160
CAB01170
CAB01180
CAB01190
CAB01200
CAB01210
CAB01220
CAB01230
CAB01240
CAB01250
CAB01260
CAB01270
CAB01280
CAB01290
CAB01300
CAB01310


```

XVAL=1000.
GO TO 1
2 XVAL=COTAN(ADC(IC,3))
1 CZ1=-COS(ADC(IC,3))*COTAN(ADC(IC,1))/XLGTH(IC)*12.
XWT=ARM(IC,1)
ZWT=ARM(IC,3)
CT1=(ZWT*SIN(ADC(IC,1))+XWT*COS(ADC(IC,3))*COTAN(ADC(IC,1)))/
1 XLGTH(IC)
CX3=-COS(ADC(IC,1))*XVAL/XLGTH(IC)*12.
CZ3=SIN(ADC(IC,3))/XLGTH(IC)*12.
CT3=-(ZWT*COS(ADC(IC,1))*XVAL+XWT*SIN(ADC(IC,3)))/
1 XLGTH(IC)
RETURN
END
C THIS IS A DOUBLE PRECISION VERSION OF CABLE4 TO BE USED
C WITH THE LFC MATRIX REDUCTION AND IBM FOOT
C FINDING ROUTINE
SUBROUTINE LONG
COMMON/INPUT/IW,IR
COMMON /DAT/ AERD(175),AERDP(50),KODE(25),LL
COMMON / PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
COMMON /DUVDUM(10,10)
COMMON/FRD/C4(30)
EQUIVALENCE(AERD(46),XCG),(AERD(47),ZCG)
EQUIVALENCE(AERD(48),THETA),(AERD(49),VD),(AERD(50),AM)
EQUIVALENCE(AERD(51),RHD),(AERD(52),WT),(AERD(53),B)
EQUIVALENCE(AERD(54),CBAR),(AERD(55),SW),(AERD(56),XIXZ)
EQUIVALENCE(AERD(57),XIXX),(AERD(58),YIYY),(AERD(59),ZIZZ),
1 (AERD(95),AKR),(AERD(100),AKLFT)
EQUIVALENCE(AERD(117),TUSNO),(AERD(119),AKSNU),(AERD(120),AKSNL)
EQUIVALENCE(AERD(123),AKSY),(AERD(124),AKPHI),(AERD(125),AKTHE),
1 (AERD(126),AKAZ),(AERD(127),TISY),(AERD(128),T2PHI),
2 (AERD(129),T3THE),(AERD(130),T4AZ)
EQUIVALENCE(AERD(131),AKSBT),(AERD(132),AKSRV),(AERD(133),AJASM),
1 (AERD(134),RSRA),(AERD(135),ELSBA),(AERD(136),RSRD),
2 (AERD(137),AKTHD),(AERD(138),AKTH),(AERD(139),GDMP),
3 (AERD(140),AKO)
EQUIVALENCE(AERDP(1),CXUP),(AERDP(2),CZUP),(AERDP(3),CMUP),
1 (AERDP(4),CXAP),(AERDP(5),CZAP),(AERDP(6),CMAP),
2 (AERDP(7),CXQP),(AERDP(8),CZQP),(AERDP(9),CMQP),
3 (AERDP(10),CXDP),(AERDP(11),CZDP),(AERDP(12),CMDP),
4 (AERDP(13),CXDP),(AERDP(14),CZDP),(AERDP(15),CMDP),
5 (AERDP(16),CXADP),(AERDP(17),CZADP),(AERDP(18),CMADP),
6 (AERDP(19),CYBP),(AERDP(20),CLBP),(AERDP(21),CNBP),
7 (AERDP(22),CYBP),(AERDP(23),CLBP),(AERDP(24),CNBP),
8 (AERDP(25),CYBP),(AERDP(26),CLBP),(AERDP(27),CNBP),
9 (AERDP(28),CYDP),(AERDP(29),CLDP),(AERDP(30),CNDP),
A (AERDP(31),CYDAP),(AERDP(32),CLDAP),(AERDP(33),CNDAP),
B (AERDP(34),CYDSP),(AERDP(35),CLDSP),(AERDP(36),CNDSP)
DIMENSION CMAT(14,14,3),RWAT(14,3)
COMPLEX PDITS(44)
COMMON/SNURB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
COMMON /RDUIGH/FRIC(3,6)
DIMENSION FXS(3,4)
DO 10 J=1,3

```

CAB01320
CAB01330
CAB01340
CAB01350
CAB01360
CAB01370
CAB01380
CAB01390
CAB01400
CAB01410
CAB01420
CAB01430
CAB01440
CAB01450
CAH00010
CAB00020
CAB00030
CAB00040
CAB00050
CAB00060
CAB00070
CAB00080
CAB00090
CAB00100
CAB00110
CAB00120
CAB00130
CAB00140
CAB00150
CAB00160
CAB00170
CAB00180
CAB00190
CAB00200
CAB00210
CAB00220
CAB00230
CAB00240
CAB00250
CAB00260
CAB00270
CAB00280
CAB00290
CAB00300
CAB00310
CAB00320
CAB00330
CAB00340
CAB00350
CAB00360
CAB00370
CAB00380
CAB00390
CAB00400
CAB00410

DO 10 K=1,4	CAB00420
10 FXS(J,K)=0.	CAB00430
DO 1 IC=1,5	CAB00440
DO 3 J=1,10	CAB00450
DO 3 K=1,10	CAB00460
3 DUM(J,K)=0.	CAB00470
IF(KODE(10).EQ.3)GO TO 649	CAB00480
TENS=TF	CAB00490
IF(IC.GT.2) TENS=TF	CAB00500
IF(IC.GT.4) TENS=TLFT	CAB00510
DUM(1,2)= - TENS * COS(ADC(IC,3))	CAB00520
DUM(1,5)= - TENS * SIN(ADC(IC,1))	CAB00530
DUM(2,2)= TENS * COS(ADC(IC,1))	CAB00540
DUM(2,6)= - TENS * SIN(ADC(IC,3))	CAB00550
DUM(3,2)=(ARM(IC,3)*DUM(1,2)-ARM(IC,1)*DUM(2,2))/12.	CAB00560
DUM(3,5)= ARM(IC,3)*DUM(1,5)/12.	CAB00570
DUM(3,6)=-ARM(IC,1)*DUM(2,6)/12.	CAB00580
IF(IC.GT.2) GO TO 2	CAB00590
DUM(1,3)=COS(ADC(IC,1))	CAB00600
DUM(2,3)=COS(ADC(IC,3))	CAB00610
DUM(3,3)=(ARM(IC,3)*DUM(1,3)-ARM(IC,1)*DUM(2,3))/12.	CAB00620
CALL DLGTH(CX,CZ,CT,1,0)	CAB00630
CALL DLGTH(CXP,CZP,CTP,2,0)	CAB00640
CX= CX + CXP	CAB00650
XPZ =-(CZ+CZP)/CX	CAB00660
DUM(4,1) =XPZ	CAB00670
XPT =-(CT+CTP)/CX	CAB00680
DUM(4,2)=XPT	CAB00690
DUM(4,4)= -!	CAB00700
CALL DCOSLG(IC,DUM(5,4),DUM(5,1),DUM(5,2),DUM(6,4),	CAB00710
1DUM(6,1),DUM(6,2))	CAB00720
DUM(5,5)=-1	CAB00730
DUM(6,6)=-!	CAB00740
CALL MASH(3,6)	CAB00750
DO 4 J=1,3	CAB00760
DO 4 K=1,3	CAB00770
4 FXS(J,K)=FXS(J,K)+DUM(J,K)	CAB00780
GO TO 1	CAB00790
2 IF(IC.GT.4)GO TO 5	CAB00800
CALL DLGTH(CX,CZ,CT,3,0)	CAB00810
CALL DLGTH(CXP,CZP,CTP,4,0)	CAB00820
DUM(7,1)=CZ+CZP	CAB00830
DUM(7,2)=CT+CTP	CAB00840
DUM(7,3)=CX+CXP	CAB00850
DUM(4,7)=AKP*12.	CAB00860
8 DUM(1,4)=COS(ADC(IC,1))	CAB00870
DUM(2,4)=COS(ADC(IC,3))	CAB00880
DUM(3,4)=(ARM(IC,3)*DUM(1,4)-ARM(IC,1)*DUM(2,4))/12.	CAB00890
CALL DCOSLG(IC,DUM(5,3),DUM(5,1),DUM(5,2),DUM(6,3),DUM(6,1),DUM	CAB00900
1(6,2))	CAB00910
DUM(4,4)=-1	CAB00920
DUM(5,5)=-1	CAB00930
DUM(6,6)=-!	CAB00940
DUM(7,7)=-!	CAB00950
CALL MASH(3,7)	CAB00960

DO 6 J=1,3	CABC0970
DO 6 K=1,3	CABC0980
IF(K.NE.3)FXS(J,K)=FXS(J,K)+DUM(J,K)	CABC0990
6 IF(K.EQ.3)FXS(J,4)=FXS(J,4)+DUM(J,K)	CABC1000
GO TO 1	CABC1010
5 IF(KODE(11).EQ.0)GO TO 1	CABC1020
CALL DLGTH(DUM(7,3),DUM(7,1),DUM(7,2),5,0)	CABC1030
DUM(4,7)=AKLFT*12.	CABC1040
GO TO 8	CABC1050
1 CONTINUE	CABC1060
C ADD SNUBBER INCREMENTS	CABC1070
CALL LONGSN	CABC1080
DO 7 J=1,3	CABC1090
FXS(J,1)=FXS(J,1)+SNU(J,2)	CABC1100
FXS(J,2)=FXS(J,2)+SNU(J,3)	CABC1110
7 FXS(J,4)=FXS(J,4)+SNU(J,1)	CABC1120
CALL FRICT(0)	CABC1130
C ZERO CABLE EFFECTS FOR CABLELESS MODEL CHAR.	CABC1140
IF(KODE(13).NE.-1.)GO TO 649	CABC1150
DO 84 J=1,3	CABC1160
DO 84 K=1,4	CABC1170
84 FXS(J,K)=0.	CABC1180
DO 85 J=1,3	CABC1190
DO 85 K=1,6	CABC1200
85 FRIC(J,K)=0.	CABC1210
DO 86 J=1,3	CABC1220
DO 86 K=1,3	CABC1230
86 SNUD(J,K)=0.	CABC1240
C THE CABLE FORCES/MOMENTS PARTIALS ARE COMPLETED	CABC1250
C AERO. DATA IS NOW COMPUTED	CABC1260
649 Q=RH0*V0*V0/2.	CABC1270
QS=Q*SW	CABC1280
IF(V0.NE.0.)QSV=QS/V0	CABC1290
IF(V0.EQ.0.)QSV=0.	CABC1300
XU=CXUP*QSV	CABC1310
ZU=CZUP*QSV	CABC1320
EMU=CMUP*QSV*CBAR	CABC1330
XA=CXAP*QSV	CABC1340
ZA=CZAP*QSV	CABC1350
EMA=CMAP*QSV*CBAR	CABC1360
IF(V0.NE.0.)XQ=CXQP*QSV*CBAR/(V0*2.)	CABC1370
IF(V0.EQ.0.)XQ=0.	CABC1380
IF(V0.NE.0.)ZQ=CZQP*QSV*CBAR/(V0*2.)	CABC1390
IF(V0.EQ.0.)ZQ=0.	CABC1400
EMQ=CMQP*QSV*CBAR/2.	CABC1410
XQE=CXQEP*QS	CABC1420
ZQE=CZQEP*QS	CABC1430
EMQE=CMQEP*QS*CBAR	CABC1440
IF(V0.NE.0.)XAD=CXADP*QSV*CBAR/(V0*2.)	CABC1450
IF(V0.EQ.0.)XAD=0.	CABC1460
IF(V0.NE.0.)ZAD=CZADP*QSV*CBAR/(V0*2.)	CABC1470
IF(V0.EQ.0.)ZAD=0.	CABC1480
IF(V0.NE.0.)EMAD=CMADP*QSV*CBAR/(2.*V0)	CABC1490
IF(V0.EQ.0.)EMAD=0.	CABC1500
IFDW=14	CABC1510

```

      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*CDOS(THETA)-XA*VD
      CMAT(1,2,2)=-XQ-XAD*VD-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)=-7A-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*VD
      CMAT(2,2,2)=-ZQ-7AD*VD-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*VD+ZCG*WT*CDOS(THETA)/12.
      CMAT(3,2,2)=(-EMQ-EMAD*VD)*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
      IL 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

```

CAB01520
 CAB01530
 CAB01540
 CAB01550
 CAB01560
 CAB01570
 CAB01580
 CAB01590
 CAB01600
 CAB01610
 CAB01620
 CAB01630
 CAB01640
 CAB01650
 CAB01660
 CAB01670
 CAB01680
 CAB01690
 CAB01700
 CAB01710
 CAB01720
 CAB01730
 CAB01740
 CAB01750
 CAB01760
 CAB01770
 CAB01780
 CAB01790
 CAB01800
 CAB01810
 CAB01820
 CAB01830
 CAB01840
 CAB01850
 CAB01860
 CAB01870
 CAB01880
 CAB01890
 CAB01900
 CAB01910
 CAB01920
 CAB01930
 CAB01940
 CAB01950
 CAB01960
 CAB01970
 CAB01980
 CAB01990
 CAB02000
 CAB02010
 CAB02020
 CAB02030
 CAB02040
 CAB02050
 CAB02060

CMAT(4,2,1)=-XPT	CABC2070
CMAT(4,4,1)=2	CABC2080
C ACTIVE CABLE CONTROL EQS.	CABC2090
IF (KODE(13).LE.0)GO TO 30	CABC2100
CMAT(1,5,1)=0.0	CABC2110
CMAT(2,5,1)=0.0	CABC2120
CMAT(3,5,1)=0.0	CABC2130
IF(KODE(6).EQ.1.OR.KODE(6).EQ.3)GO TO 46	CABC2140
IC2=4	CABC2150
IC1=3	CABC2160
GO TO 47	CABC2170
46 IC2=1	CABC2180
IC1=2	CABC2190
47 CMAT(1,10,1)=- (COS(ADC(IC2,1))-COS(ADC(IC1,1)))	CABC2200
CMAT(2,10,1)=- (COS(ADC(IC2,3))-COS(ADC(IC1,3)))	CABC2210
CMAT(3,10,1)=- (ARM(IC2,3)*COS(ADC(IC2,1))-ARM(IC2,1)*COS(ADC(IC2,3	CABC2220
1)))/12.+(ARM(IC1,3)*COS(ADC(IC1,1))-ARM(IC1,1)*COS(ADC(IC1,3)))/12	CABC2230
2.	CABC2240
C EQ OF MOTOR DYN.	CABC2250
CMAT(5,5,1)=+2.*R.D*PSBA	CABC2260
CMAT(5,5,2)=+2.*RSBD*FLSBA	CABC2270
CMAT(5,7,1)=+AKSRT*2.	CABC2280
CMAT(5,6,2)=-AKSBT*2.*AKSRV-GDMP*FSBA	CABC2290
CMAT(5,6,3)=-AJASM*FSBA-GEM*FLSBA	CABC2300
CMAT(5,8,3)=-AJASM*ELSEA	CABC2310
C EQ RELATING PULLEY ROTATION TO SYS. GEOM., MOTOR ON TOP	CABC2320
CALL DLGTH(CMAT(6,4,1),CMAT(6,1,1),CMAT(6,2,1),IC1,0)	CABC2330
CMAT(6,6,1)=-RSBD/12.	CABC2340
C ACTIVE CABLE FEEDBACK EQ.	CABC2350
CMAT(7,2,2)=AKQ	CABC2360
CMAT(7,6,1)=AKTH	CABC2370
CMAT(7,6,2)=AKTHD	CABC2380
CMAT(7,9,1)=-1.	CABC2390
C TOTAL VOLTAGE EQ FM + EMC	CABC2400
CMAT(9,7,1)=-1.	CABC2410
CMAT(9,9,1)=1.	CABC2420
CMAT(9,11,1)=1.	CABC2430
C RELATION OF THM TO THMD	CABC2440
CMAT(8,8,1)=-1.	CABC2450
CMAT(8,6,2)=1.	CABC2460
C RELATION OF TDRK TO DTC AND INPUT DT	CABC2470
CMAT(10,5,1)=1.	CABC2480
CMAT(10,10,1)=1.	CABC2490
CMAT(10,12,1)=-1.	CABC2500
GO TO 31	CABC2510
C FEEDBACK LOOP EQUATION	CABC2520
30 CMAT(5,2,2)=AKTHE	CABC2530
CMAT(5,5,2)=-T4THE	CABC2540
CMAT(5,5,1)=-1.	CABC2550
31 ITHD=0	CABC2560
IF(KODE(14).EQ.0)GO TO 32	CABC2570
C SURST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS	CABC2580
IDX=KODE(14)	CABC2590
IDN=KODE(15)	CABC2600

IF(IDN.NE.13)GO TO 52	CABC 2620
IDN=2	CABC 2630
ITHD=13	CABC 2640
52 IF(IDX.GT.14)GO TO 38	CABC 2650
DO 34 I=1,14	CABC 2660
DO 34 K=1,3	CABC 2670
BMAT(I,K)=CMAT(I,IDN,K)	CABC 2680
34 CMAT(I,IDN,K)=-CMAT(I,IDX,K)	CABC 2690
GO TO 32	CABC 2700
38 DO 37 I=1,14	CABC 2710
DO 37 K=1,3	CABC 2720
BMAT(I,K)=CMAT(I,IDN,K)	CABC 2730
37 CMAT(I,IDN,K)=0.0	CABC 2740
IF(IDX.EQ.16)GO TO 39	CABC 2750
CMAT(1,IDN,1)=XDE	CABC 2760
CMAT(2,IDN,1)=7DE	CABC 2770
CMAT(3,IDN,1)=EMDE	CABC 2780
GO TO 32	CABC 2790
39 CMAT(1,IDN,1)=XA	CABC 2800
CMAT(2,IDN,1)=ZA	CABC 2810
CMAT(3,IDN,1)=EMA	CABC 2820
32 N=KODE(8)	CABC 2830
655 CALL MATRIX(CMAT,N,ROOTS,K4A,IER)	CABC 2840
IF(KODE(14).EQ.0)GO TO 35	CABC 2850
DO 36 I=1,14	CABC 2860
DO 36 K=1,3	CABC 2870
36 CMAT(I,IDN,K)=BMAT(I,K)	CABC 2880
C 35 IF(KODE(5).NE.0) WRITE(IW,100) IER	CABC 2890
C 100 FORMAT(2X,'IER=',I3.3X,'SEE SUPP PQFB AND PRFM FOR ERROR CODE')	CABC 2900
C THE ROOTS OF THE CHARAC. EQNAT. ARE IN THE COMPLEX ARRAY 'ROOTS'	CABC 2910
C AND THE NUMBER OF ROOTS IS 'K4A'	CABC 2920
35 K4A=K4A-	CABC 2930
IF(ITHD.NE.13)GO TO 70	CABC 2940
K4A=K4A+1	CABC 2950
ROOTS(K4A)=(0.0,0.0)	CABC 2960
DO 71 I=1,K4A	CABC 2970
C4(K4A+2-I)=C4(K4A+1-I)	CABC 2980
71 CONTINUE	CABC 2990
C4(1)=0.	CABC 3000
70 CALL PRINTR(IW,ROOTS,K4A)	CABC 3010
GO TO 651	CABC 3020
650 CONTINUE	CABC 3030
C NEW SNUBBER EFFECTS	CABC 3040
KODE(14)=0	CABC 3050
DO 600 IC=1,4	CABC 3060
DO 201 I=1,10	CABC 3070
DO 201 J=1,10	CABC 3080
201 DUM(I,J)=0.	CABC 3090
TC=TF-TR+TUSND	CABC 3100
IF(IC.GT.2) TC=TUSND	CABC 3110
DUM(1,3)=-TC*COS(ADC(1,3))	CABC 3120
DUM(1,4)=-TC*SIN(ADC(1,1))	CABC 3130
DUM(1,6)=COS(ADC(1,1))	CABC 3140
DUM(2,3)=TC*COS(ADC(1,1))	CABC 3150
DUM(2,5)=-TC*SIN(ADC(1,3))	CABC 3160

```

DUM(2,6)=C7S(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)=-FMU
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

```

CABC 3170
CABC 3180
CABC 3190
CABC 3200
CABC 3210
CABC 3220
CABC 3230
CABC 3240
CABC 3250
CABC 3260
CABC 3270
CABC 3280
CABC 3290
CABC 3300
CABC 3310
CABC 3320
CABC 3330
CABC 3340
CABC 3350
CABC 3360
CABC 3370
CABC 3380
CABC 3390
CABC 3400
CABC 3410
CABC 3420
CABC 3430
CABC 3440
CABC 3450
CABC 3460
CABC 3470
CABC 3480
CABC 3490
CABC 3500
CABC 3510
CABC 3520
CABC 3530
CABC 3540
CABC 3550
CABC 3560
CABC 3570
CABC 3580
CABC 3590
CABC 3600
CABC 3610
CABC 3620
CABC 3630
CABC 3640
CABC 3650
CABC 3660
CABC 3670
CABC 3680
CABC 3690
CABC 3700
CABC 3710

```

CALL FREQ(ROOTS,K4A,C4(K4A+1))
RETURN
END
SUBROUTINE PRINTR (LOUT,RT,NROOT)
COMMON/FRO/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 570 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
DAMP= 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+,F11.4,8E13.4)
GO TO 530
531 WRITE(LOUT,532) SIG,THD,THDI
532 FORMAT(E12.4,14X,2E'3.4)
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

```

```

CABC3720
CABC3730
CABC3740
CABC0010
CABC0020
CABC0030
CABC0040
CABC0050
CABC0060
CABC0070
CABC0080
CABC0090
CABC0100
CABC0110
CABC0120
CABC0130
CABC0140
CABC0150
CABC0160
CABC0170
CABC0180
CABC0190
CABC0200
CABC0210
CABC0220
CABC0230
CABC0240
CABC0250
CABC0260
CABC0270
CABC0280
CABC0290
CABC0300
CABC0310
CABC0320
CABC0330
CABC0340
CABC0350
CABC0360
CABC0370
CABC0380
CABC0390
CABC0400
CABC0410
CABC0420
CABC0430
CABC0440
CABC0450
CABC0460
CABC0470
CABC0480
CABC0490
CABC0500
CABC0510
CABC0520

```



```

INN=N-NN
DO 1001 LL=1,INN
L=N+1-LL
II=L-1
JJ=L-1
DO 1001 I=1,II
DO 1001 J=1,JJ
1001 DUM(I,J)=DUM(I,J)+DUM(L,J)*DUM(I,L)/(-DUM(L,L))
RETURN
END
SUBROUTINE LAT
COMMON/INOUT/IW,IP
COMMON /DAT/ AERD(175),AEROP(50),KODE(26),LL
COMMON / PLYCHA/RTD,XLGT(5),ADC(5,3),ARM(5,3),TR,TLFT,TF
COMMON /DU/DUM(10,10)
COMMON/ERD/C4(30)
EQUIVALENCE(AERD(46),XCG),(AERD(47),ZCG)
EQUIVALENCE(AERD(63),THETA),(AERD(49),VO ),(AERD(50), AM)
EQUIVALENCE(AERD(51),PHD ),(AERD(52), WT),(AERD(53),B )
EQUIVALENCE(AERD(54),CBAP ),(AERD(55),SW ),(AERD(56), XIXZ)
EQUIVALENCE(AERD(57),XIXX ),(AERD(58),YIYY ),(AERD(59),ZIZZ ),
1 (AERD(95),AKF ),(AERD(100),AKLFT)
EQUIVALENCE(AERD(117),TUSND),(AERD(119),AKSNU),(AERD(120),AKSNL)
EQUIVALENCE(AERD(123), AKSY),(AERD(124),AKPHI),(AERD(125),AKTHE),
1 (AERD(126), AKAZ),(AERD(127), T1SY),(AERD(129),T2PHI),
2 (AERD(129),T3THE),(AERD(130), T4AZ)
EQUIVALENCE(AERD(131),AKSRT),(AERD(132),AKSRV),(AERD(133),AJASM),
1 (AERD(134),RSBA ),(AERD(135),EL5BA),(AERD(136), PSBD),
2 (AERD(137),AKTHD),(AERD(138), AKTH),(AERD(139), GMP),
3 (AERD(140), AKO),(AERD(141), AKZ),(AERD(142),AKPSD),
4 (AERD(143), AKY),(AERD(144), AKYD)
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),
3 (AEROP(10),CXDP),(AEROP(11),CZDP),(AEROP(12),CMDP),
4 (AEROP(13),CXDP),(AEROP(14),CZDP),(AEROP(15),CMDP),
5 (AEROP(16),CXADP),(AEROP(17),CZADP),AEROP(18),CADP),
6 (AEROP(19),CYRP),(AEROP(20),CLBP),AEROP(21),CNRP),
7 (AEROP(22),CYRP),(AEROP(23),CLRP),AEROP(24),CNDP),
8 (AEROP(25),CYRP),(AEROP(26),CLRP),(AEROP(27),CNDP),
9 (AEROP(28),CYDP),(AEROP(29),CLDP),(AEROP(30),CNDP),
A (AEROP(31),CYDP),(AEROP(32),CLDP),(AEROP(33),CNDP),
B (AEROP(34),CYDSP),(AEROP(35),CLDSP),(AEROP(36),CNDSP)
DIMENSION CMAT(14,14,3),BMAT(14,3)
COMPLEX ROOTS(44)
COMMON/SNURR/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
COMMON /FOUGH/ERIC(3,6)
DIMENSION FXS(3,3)
DO 10 J=1,3
DO 10 K=1,3
10 FXS(J,K)=0.
IF(KODE(10).EQ.3)GO TO 650
DO 111 IC=1,5
IF(KODE(11).EQ.7.AND.IC.EQ.5)GO TO 1
DO 3 J=1,8

```

DO 3 K=1,8	CABC 1080
3 DUM(J,K)=0.	CABC 1090
TENS=TF	CABC 1100
IF(IC.GT.2)TENS=TR	CABC 1110
IF(IC.GT.4)TENS=TLFT	CABC 1120
CA1=COS(ADC(IC,1))	CABC 1130
CA2=COS(ADC(IC,2))	CABC 1140
CA3=COS(ADC(IC,3))	CABC 1150
IF(ABS(CA1).LT..0001) CA1=0.	CABC 1160
IF(ABS(CA2).LT..0001) CA2=0.	CABC 1170
IF(ABS(CA3).LT..0001) CA3=0.	CABC 1180
DUM(1,2)=-TENS*CA1	CABC 1190
DUM(1,3)=TENS*CA3	CABC 1200
DUM(1,4)=CA2	CABC 1210
DUM(1,5)=-TENS*SIN(ADC(IC,2))	CABC 1220
DUM(2,2)=(AFM(IC,1)*DUM(1,2)-AFM(IC,2)*TENS*CA2)/12.	CABC 1230
DUM(2,3)= AFM(IC,1)*DUM(1,3)/12.	CABC 1240
DUM(2,4)=(AFM(IC,1)*CA2-AFM(IC,2)*CA1)/12.	CABC 1250
DUM(2,5)= AFM(IC,2)*TENS*SIN(ADC(IC,1))/12.	CABC 1260
DUM(2,6)= AFM(IC,1)*DUM(1,6)/12.	CABC 1270
DUM(4,4)=-1.	CABC 1280
DUM(4,8)=0.	CABC 1290
IF(IC.GT.2)DUM(4,8)=AKR*12.	CABC 1300
IF(IC.GT.4)DUM(4,8)=AKLFT*12.	CABC 1310
DUM(3,2)=-AFM(IC,3)*DUM(1,2)/12.	CABC 1320
DUM(3,3)=(-AFM(IC,3)*DUM(1,3)-AFM(IC,2)*TENS*CA2)/12.	CABC 1330
DUM(3,4)=(AFM(IC,2)*CA3-AFM(IC,3)*CA2)/12.	CABC 1340
DUM(3,7)=-AFM(IC,2)*TENS*SIN(ADC(IC,3))/12.	CABC 1350
DUM(3,6)=-AFM(IC,3)*DUM(1,6)/12.	CABC 1360
CALL DCOSD(IC,DUM(5,1),DUM(5,2),DUM(5,3),DUM(5,1),DUM(6,2),DUM(CABC 1370
16,3),DUM(7,1),DUM(7,2),DUM(7,3))	CABC 1380
DUM(5,5)=-1.	CABC 1390
DUM(6,6)=-1.	CABC 1400
DUM(7,7)=-1.	CABC 1410
IF(IC.GT.2)GO TO 2	CABC 1420
CALL MASH(3,7)	CABC 1430
6 DO 4 J=1,3	CABC 1440
DO 4 K=1,3	CABC 1450
4 FXS(J,K)=FXS(J,K)+DUM(J,K)	CABC 1460
GO TO 1	CABC 1470
2 IF(IC.GT.4)GO TO 5	CABC 1480
CALL DLGTH(CY,CPS,CPH,3,1)	CABC 1490
CALL DLGTH(CYP,CPSD,CPHD,4,1)	CABC 1500
DUM(8,1)=CY+CYP	CABC 1510
DUM(8,2)=CPS+CPSP	CABC 1520
DUM(8,3)=CPH+CPHD	CABC 1530
DUM(8,8)=-1.	CABC 1540
CALL MASH(3,8)	CABC 1550
GO TO 6	CABC 1560
5 IF(KOZF(11).EQ.0)GO TO 1	CABC 1570
CALL DLGTH(DUM(8,1),DUM(8,2),DUM(8,3),5,1)	CABC 1580
DUM(8,8)=-1.	CABC 1590
CALL MASH(3,8)	CABC 1600
GO TO 6	CABC 1610
1 CONTINUE	CABC 1620

111 CONTINUE	CABC1630
C COMPLETE SUMMATION OF CABLE FORCES & MOMENTS	CABC1640
C ADD SNURBER INCREMENTS	CABC1650
112 CALL LATS4	CABC1660
DO 3 J=1,3	CABC1670
DO 4 K=1,3	CABC1680
8 FXS(J,K)=FXS(J,K)+SNU(J,K)	CABC1690
CALL FRICT(1)	CABC1700
C ZERO CABLE EFFECTS FOR CABLELESS MODEL OPTION	CABC1710
IF(KODE(13).NE.-1)GO TO 620	CABC1720
IF(KODE(9).NE.3)WRITE(IW,22)	CABC1730
22 FORMAT(5X,'KODE(9) HAS BEEN SET BY PROG TO 3 FOR CABLELESS MODEL C	CABC1740
CHARACTERISTICS')	CABC1750
KODE(9)=3	CABC1760
DO 20 J=1,3	CABC1770
DO 20 K=1,3	CABC1780
SNUD(J,K)=0.	CABC1790
20 FXS(J,K)=0.	CABC1800
DO 21 J=1,3	CABC1810
DO 21 K=1,6	CABC1820
21 FRICT(J,K)=1.	CABC1830
GO TO 620	CABC1840
650 CONTINUE	CABC1850
KODE(16)=0	CABC1860
DO 610 I=1,3	CABC1870
DO 610 J=1,6	CABC1880
610 FRICT(I,J)=1.	CABC1890
DO 611 I=1,3	CABC1900
DO 611 J=1,3	CABC1910
SNU(I,J)=0.	CABC1920
611 SNUD(I,J)=0.	CABC1930
DO 600 IC=1,4	CABC1940
DO 605 I=1,10	CABC1950
DO 605 J=1,10	CABC1960
605 DUM(I,J)=0.	CABC1970
TC=TF-TF+TUSND	CABC1980
IF(IC.GT.2) TC=TUSND	CABC1990
CA1=CDS(ADC(IC,1))	CABC2000
CA2=CDS(ADC(IC,2))	CABC2010
CA3=CDS(ADC(IC,3))	CABC2020
IF(ABS(CA1).LT..0001) CA1=0.	CABC2030
IF(ABS(CA2).LT..0001) CA2=0.	CABC2040
IF(ABS(CA3).LT..0001) CA3=0.	CABC2050
DUM(1,2)=-TC*CA1	CABC2060
DUM(1,3)=TC*CA3	CABC2070
DUM(1,4)=CA2	CABC2080
DUM(1,6)=-TC*SIN(ADC(IC,2))	CABC2090
DUM(2,2)=(ARM(IC,1)*DUM(1,2)-ARM(IC,2)*TC*CA2)/12.	CABC2100
DUM(2,3)=ARM(IC,1)*DUM(1,3)/12.	CABC2110
DUM(2,4)=(ARM(IC,1)*CA2-ARM(IC,2)*CA1)/12.	CABC2120
DUM(2,5)=ARM(IC,2)*TC*SIN(ADC(IC,1))/12.	CABC2130
DUM(2,6)=ARM(IC,1)*DUM(1,6)/12.	CABC2140
DUM(3,2)=-ARM(IC,3)*DUM(1,2)/12.	CABC2150
DUM(3,3)=(-ARM(IC,3)*DUM(1,3)-ARM(IC,2)*TC*CA2)/12.	CABC2160
DUM(3,4)=(ARM(IC,2)*CA3-ARM(IC,3)*CA2)/12.	CABC2170

```

DUM(3,7)=-ARM(IC,2)*TC*SIN(ADC(IC,3))/12.          CABO 2180
DUM(3,6)=-ARM(IC,3)*DUM(1,6)/12.                  CABO 2190
DUM(4,4)=-1.                                        CABO 2200
DUM(4,8)=AK*SNV*12.                                CABO 2210
IF(IC.GT.2) DUM(4,8)=AKSNL*12.                     CABO 2220
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))           CABO 2230
DUM(5,5)=-1.                                        CABO 2240
DUM(6,6)=-1.                                        CABO 2250
DUM(7,7)=-1.                                        CABO 2260
DUM(8,8)=-1.                                        CABO 2270
CALL DLGTH(DUM(8,1),DUM(8,2),DUM(8,3),IC,1)       CABO 2280
CALL MASH(3,8)                                       CABO 2290
D0 649 J=1,3                                         CABO 2300
D0 649 K=1,3                                         CABO 2310
649 FXS(J,K)=FXS(J,K)+DUM(J,K)                     CABO 2320
600 CONTINUE                                         CABO 2330
C ADD AERO INCREMENTS                                CABO 2340
620 Q=.5*RHQ*VJ*VJ*VJ                               CABO 2350
QS=Q*SW                                              CABO 2360
IF(VJ.NE.0.)QSV=QS/VJ                               CABO 2370
IF(VJ.EQ.0.)QSV=0.                                  CABO 2380
IF(VJ.NE.0.)BOV=B/(2.*VJ)                           CABO 2390
IF(VJ.EQ.0.)BOV=0.                                  CABO 2400
YV=CYRP*QSV                                          CABO 2410
ELV=CLRP*QSV*B                                       CABO 2420
ENV=CNRP*QSV*B                                       CABO 2430
YD=CYPD*QS*BOV                                       CABO 2440
ELP=CLRP*BQV*QS*B                                    CABO 2450
END=CNDP*BQV*QS*B                                    CABO 2460
YD=CYPD*QS*BOV                                       CABO 2470
ELR=CLRP*BQV*QS*B                                    CABO 2480
END=CNDP*BQV*QS*B                                    CABO 2490
YDR=CYPD*QS                                           CABO 2500
ENDR=CNDP*QS*B                                       CABO 2510
ELDR=CLDP*QS*B                                       CABO 2520
YDA=CYDAP*QS                                          CABO 2530
ENDA=CNDAP*QS*B                                       CABO 2540
ELDA=CLDAP*QS*B                                       CABO 2550
YDS=CYDSP*QS                                          CABO 2560
ENDS=CNDSP*QS*B                                       CABO 2570
ELDS=CLDSP*QS*B                                       CABO 2580
42 D0 113 I=1,14                                     CABO 2590
D0 113 J=1,14                                       CABO 2600
D0 113 K=1,3                                         CABO 2610
113 CMAT(I,J,K)=0.0                                   CABO 2620
C Y FORCE EQUATION                                    CABO 2630
CMAT(1,1,1)=-FXS(1,1)                                 CABO 2640
CMAT(1,1,2)=-YV-SNUD(1,1)-FRIC(1,4)-FRIC(1,1)      CABO 2650
CMAT(1,1,3)=YV                                       CABO 2660
CMAT(1,2,1)=-FXS(1,2)+YV*VJ-WT*SIN(THETA)          CABO 2670
CMAT(1,2,2)=-YR-SNUD(1,2)-FRIC(1,5)-FRIC(1,2)      CABO 2680
CMAT(1,2,3)=AM*XCX/12.                               CABO 2690
CMAT(1,3,1)=-FXS(1,3)-WT*COS(THETA)                 CABO 2700
CMAT(1,3,2)=-YR-SNUD(1,3)-FRIC(1,6)-FRIC(1,3)      CABO 2710

```

CMAT(1,3,3)=-AM*7CG/12.	CABC2730
C YAW EQUATION	CABC2740
CMAT(2,1,1)=-FXS(2,1)	CABC2750
CMAT(2,1,2)=-ENV-SNUD(2,1)-FRIC(2,4)-FRIC(2,1)	CABC2760
CMAT(2,1,3)=AM*XCG/12.	CABC2770
CMAT(2,2,1)=-FXS(2,2)+ENV*VO-XCG*WT*SIN(THETA)/12.	CABC2780
CMAT(2,2,2)=-ENV-SNUD(2,2)-FRIC(2,5)-FRIC(2,2)	CABC2790
CMAT(2,2,3)=ZIZZ	CABC2800
CMAT(2,3,1)=-FXS(2,3)-XCG*WT*COS(THETA)/12.	CABC2810
CMAT(2,3,2)=-ENV-SNUD(2,3)-FRIC(2,6)-FRIC(2,3)	CABC2820
CMAT(2,3,3)=-XIXZ	CABC2830
C ROLL EQUATION	CABC2840
CMAT(3,1,1)=-FXS(3,1)	CABC2850
CMAT(3,1,2)=-ELV-SNUD(3,1)-FRIC(3,4)-FRIC(3,1)	CABC2860
CMAT(3,1,3)=-AM*7CG/12.	CABC2870
CMAT(3,2,1)=-FXS(3,2)+ELV*VO+7CG*WT*SIN(THETA)/12.	CABC2880
CMAT(3,2,2)=-ELV-SNUD(3,2)-FRIC(3,5)-FRIC(3,2)	CABC2890
CMAT(3,2,3)=-XIXZ	CABC2900
CMAT(3,3,1)=-FXS(3,3)+7CG*WT*COS(THETA)/12.	CABC2910
CMAT(3,3,2)=-ELV-SNUD(3,3)-FRIC(3,6)-FRIC(3,3)	CABC2920
CMAT(3,3,3)=XIXX	CABC2930
C ACTIVE CABLE CONTROL EQUATIONS	CABC2940
IF(KODE(13).NE.1)GO TO 30	CABC2950
IF(KODE(6).EQ.1.OR.KODE(6).EQ.4)GO TO 46	CABC2960
IC2=2	CABC2970
IC1=1	CABC2980
GO TO 47	CABC2990
46 IC2=4	CABC3000
IC1=3	CABC3010
47 CMAT(1,10,1)=+(COS(ADC(IC2,2))-COS(ADC(IC1,2)))	CABC3020
CMAT(3,10,1)=+(AFM(IC2,2)*COS(ADC(IC2,3))-AFM(IC2,3)*COS(ADC(IC2,2	CABC3030
1)))/12.-((AFM(IC1,2)*COS(ADC(IC1,3))-AFM(IC1,3)*COS(ADC(IC1,2)))/12	CABC3040
2.	CABC3050
CMAT(2,10,1)=+(AFM(IC2,1)*COS(ADC(IC2,2))-AFM(IC2,2)*COS(ADC(IC2,	CABC3060
1)))/12.-((AFM(IC1,1)*COS(ADC(IC1,2))-AFM(IC1,2)*COS(ADC(IC1,1	CABC3070
2)))/12.	CABC3080
C EQ. OF MOTION DYN.	CABC3090
CMAT(4,4,1)=+2.*RSPD*RSRA	CABC3100
CMAT(4,4,2)=+2.*RSPD*ELSRA	CABC3110
CMAT(4,6,1)=+AKSRT*2.	CABC3120
CMAT(4,5,2)=-AKSRT*2.*AKSBV-GDMP*RSRA	CABC3130
CMAT(4,5,3)=-AJASM*RSRA-GDMP*ELSRA	CABC3140
CMAT(4,7,3)=-AJASM*ELSRA	CABC3150
CALL DLGTH(CMAT(5,1,1),CMAT(5,2,1),CMAT(5,3,1),IC1,1)	CABC3160
CMAT(5,5,1)=+RSPD/12.	CABC3170
C EQ FOR TOTAL VOLTAGE=ACTIVE SYSTEM+INPUT VOLTAGE,EMO	CABC3180
CMAT(9,6,1)=-1.	CABC3190
CMAT(9,9,1)=1.	CABC3200
CMAT(9,11,1)=1.	CABC3210
C FEEDBACK CONTROL EQ.	CABC3220
CMAT(6,2,2)=AKPSD	CABC3230
CMAT(6,5,1)=AKY	CABC3240
CMAT(6,7,1)=AKYD	CABC3250
CMAT(6,9,1)=-1.	CABC3260
C RELATE ANGULAR RATES TO ANGULAR DISPLACEMENTS	CABC3270

CMAT(9,2,2)=1.	CABC 3280
CMAT(8,8,1)=-1.	CABC 3290
CMAT(7,5,2)=1.	CABC 3300
CMAT(7,7,1)=-1.	CABC 3310
C RELATION OF DTC TO DT AND DTFB	CABC 3320
CMAT(10,4,1)=1.	CABC 3330
CMAT(10,10,1)=1.	CABC 3340
CMAT(10,12,1)=-1.	CABC 3350
GO TO 31	CABC 3360
C RUDDER FEEDBACK LOOP	CABC 3370
30 CMAT(4,2,2)=AKSY	CABC 3380
CMAT(4,4,2)=-T3SY	CABC 3390
CMAT(4,4,1)=-1.	CABC 3400
C AILERON FEEDBACK LOOP	CABC 3410
CMAT(5,3,2)=AKPHI	CABC 3420
CMAT(5,5,2)=-T2PHI	CABC 3430
CMAT(5,5,1)=-1.	CABC 3440
CMAT(1,4,1)=-OS*CYDRP	CABC 3450
CMAT(1,5,1)=-OS*CYDAP	CABC 3460
CMAT(2,4,1)=-OS*B*CNDRP	CABC 3470
CMAT(2,5,1)=-OS*B*CNDAF	CABC 3480
CMAT(3,4,1)=-OS*B*CLDRP	CABC 3490
CMAT(3,5,1)=-OS*B*CLDAF	CABC 3500
31 IF(KODE(16).EQ.0)GO TO 32	CABC 3510
C SURST. COL IDX INTO COL IDN TO GET NUMERATOR ROOTS	CABC 3520
IDX=KODE(16)	CABC 3530
IDN=KODE(17)	CABC 3540
IF(IDX.GT.13)GO TO 38	CABC 3550
DO 34 I=1,14	CABC 3560
DO 34 K=1,3	CABC 3570
BMAT(I,K)=CMAT(I,IDN,K)	CABC 3580
34 CMAT(I,IDN,K)=-CMAT(I,IDX,K)	CABC 3590
GO TO 32	CABC 3600
35 DO 37 I=1,14	CABC 3610
DO 37 K=1,3	CABC 3620
BMAT(I,K)=CMAT(I,IDN,K)	CABC 3630
37 CMAT(I,IDN,K)=0.0	CABC 3640
IF(IDX.EQ.15)GO TO 39	CABC 3650
IF(IDX.EQ.16)GO TO 41	CABC 3660
CMAT(1,IDN,1)=YDR	CABC 3670
CMAT(2,IDN,1)=ENDF	CABC 3680
CMAT(3,IDN,1)=ELDF	CABC 3690
GO TO 32	CABC 3700
39 CMAT(1,IDN,1)=YDA	CABC 3710
CMAT(2,IDN,1)=ENDA	CABC 3720
CMAT(3,IDN,1)=ELDA	CABC 3730
GO TO 32	CABC 3740
41 CMAT(1,IDN,1)=YV	CABC 3750
CMAT(2,IDN,2)=ENV	CABC 3760
CMAT(3,IDN,3)=FLV	CABC 3770
32 N=KODE(9)	CABC 3780
CALL MATRIX(CMAT,N,ROOTS,KAA,IF9)	CABC 3790
IF(KODE(16).EQ.0)GO TO 35	CABC 3800
DO 36 I=1,14	CABC 3810
DO 36 K=1,3	CABC 3820

```

36 CMAT(I, IDN, K) = RCMAT(I, K) CAB03830
C 35 IF(KODE(5).NE.0) WRITE(IW,100) IEF CAB03840
C 100 FORMAT(2X, 'IEF=', I3, 3X, 'SEE SUBR. PQFR 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, CPS11, CPH11, CY2, CPS12, CPH12, CY3, CPS13, CAB04020
1CPH13) CAB04030
COMMON /PLYCHA/RTD, XLGTH(5), ADC(5,3), ARM(5,3), TP, TLFT, TR 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
CPS11=-(YWT*SIN(ADC(IC,1))+XWT*COS(ADC(IC,2))*COTAN(ADC(IC,1))) CAB04130
1/XLGTH(IC) CAB04140
CPH11=(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
CPS12=(YWT*COS(ADC(IC,1))*COTAN(ADC(IC,2))+XWT*SIN(ADC(IC,2)))/ CAB04180
1XLGTH(IC) CAB04190
CPH12=-(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
CPS13=(YWT*COS(ADC(IC,1))*XVAL-XWT*CCS(ADC(IC,2))* CAB04230
1XVAL)/XLGTH(IC) CAB04240
CPH13=(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/INOUT/IW, IR CAB00020
COMMON/DAT/AERO(175), AEROP(50), KODE(26), LL CAB00030
COMMON ZZZ(200) CAB00040
COMMON/TAB1/ZZ(300) CAB00050
COMMON/SNUBR/SNU(3,3), SN(30), THUSN, THLSN, SNUD(3,3) CAB00060
EQUIVALENC((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      (AERD(114),SNLST),(AERD(115),SNLWL),(AERD(116),SNLBL),CAB00100
4      (AERD(117),TUSNO),(AERD(118),TLSNO),(AERD(119),AKSNU),CAB00110
5      (AERD(120),AKSNL),(AERD(49), VO),(AERD(51), RHO), CAB00120
6      (AERD(76),WLCR),(AERD(77),STACR), CAB00130
7      (AERD(78),BLCF) CAB00140
EQUIVALENC (SN( 1), GX1),(SN( 2), GY1),(SN( 3), GZ1), CAB00150
1      (SN( 4), GX2),(SN( 5), GY2),(SN( 6), GZ2), CAB00160
2      (SN( 7), GX3),(SN( 8), GY3),(SN( 9), GZ3), CAB00170
3      (SN(10), GX4),(SN(11), GY4),(SN(12), GZ4), CAB00180
4      (SN(13), THU),(SN(14), THL),(SN(15), ALU), CAB00190
5      (SN(16), ALL), CAB00200
6      (SN(19),THGX1),(SN(20),THGY1),(SN(21),THGZ1), CAB00210
7      (SN(22),THGX2),(SN(23),THGY2),(SN(24),THGZ2), CAB00220
8      (SN(25),THGX3),(SN(26),THGY3),(SN(27),THGZ3), CAB00230
9      (SN(28),THGX4),(SN(29),THGY4),(SN(30),THGZ4) CAB00240
IF(KODE(10).EQ.0) GO TO 5005 CAB00250
IF(KODE(10).EQ.3) GO TO 5005 CAB00260
CALL DRCNSN(THETA) CAB00270
IF(KODE(10).NE.1) GO TO 5003 CAB00280
C TERMS TO MODEL SNUBBER EFFECTS (MODEL UNSNUBBED) CAB00290
Q=.5*RHO*V7*VO CAB00300
CALL STINT(Q,ALU,0,1,1,TUSN,NG) CAB00310
IF(NG.NE.0) GO TO 5000 CAB00320
CALL STINT(Q,ALL,0,1,1,TLN,NG) CAB00330
IF(NG.NE.0) GO TO 5000 CAB00340
CALL STINT(Q,ALU,0,2,2,THUSN,NG) CAB00350
IF(NG.NE.0) GO TO 5000 CAB00360
CALL STINT(Q,ALL,0,2,2,THLSN,NG) CAB00370
IF(NG.EQ.0) GO TO 5001 CAB00380
5000 WRITE(IW,5002) NG,ALL,ALU,0 CAB00390
5002 FORMAT(2X,'ERROR IN SNUBBER TABLE 1-2 , NG=',I3,3E10.3) CAB00400
RETURN CAB00410
5001 CONTINUE CAB00420
C CALCULATING FORCE AND MOMENT EFFECTS CAB00430
CALL DRCNSN(THETA) CAB00440
FXUSN= 2.*TUSN*GX1 CAB00450
FZUSN= 2.*TUSN*GZ1 CAB00460
AMUSN= -FXUSN*SNUZ+SNUX*FZUSN CAB00470
FXLSN= 2.*TLN*GX3 CAB00480
FZLSN= 2.*TLN*GZ3 CAB00490
AMLSN= FXLSN*SNLZ+FZLSN*SNLX CAB00500
FXSN = FXUSN+FXLSN CAB00510
FZSN = FZUSN+FZLSN CAB00520
AMSN = (AMUSN+AMLSN)/12. CAB00530
RETURN CAB00540
5003 CONTINUE CAB00550
C TERMS TO MODEL SNUBBER EFFECTS (MODEL SNUBBED) CAB00560
FXUSN= 2.*TUSNO*GX1 CAB00570
FZUSN= 2.*TUSNO*GZ1 CAB00580
AMUSN =-FXUSN*SNUZ+FZUSN*SNUX CAB00590
FXLSN= 2.*TLNO*GX3 CAB00600
FZLSN= 2.*TLNO*GZ3 CAB00610
AMLSN = FXLSN*SNLZ+FZLSN*SNLX CAB00620
FXSN = FXUSN+FXLSN CAB00630
FZSN = FZUSN+FZLSN CAB00640

```



```

      AMSN=(AMUSN+AMLSN)/12.
      RETURN
5005 FYSN=0
      FZSN=0
      AMSN=0
      RETURN
      END
      SUBROUTINE LONGSN
      COMMON/INDUT/IW,IP
      COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL
      COMMON/SNUB/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)
      COMMON ZZ7(200)
      COMMON/TAH1/ZZ(300)
      COMMON/DU/DUM(10,10)
      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), SNUJL),
3      (AERO(114), SNLST),(AERO(115), SNLWL),(AERO(116), SNLBL),
4      (AERO(117), TUSNO),(AERO(118), TUSNO),(AERO(119), AKSNU),
5      (AERO(120), AKSNL),(AERO(121), ADSNU),(AERO(122), ADSNL),
6      (AERO(123), THETA),(AERO(124), THETA),(AERO(125), THETA)
      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(28), THGY4),(SN(30), THGZ4)
      DIMENSION FTOP(3,3),FBOT(3,3)
      COT(A)=1./TAN(A)
      DO 1001 I=1,3
      DO 1001 J=1,3
      SNU(I,J)=0
1001 SNUD(I,J)=0
      DO 5102 I=1,10
      DO 5102 J=1,10
5102 DUM(I,J)=0
      IF(KODE(10).NE.1) GO TO 1000
C TERMS FOR UNSNUBBED SNUBBER EFFECTS (LONG)
      DO 1004 I=1,7
      DO 1004 J=1,7
1004 DUM(I,J)=0
      CALL DRCSN(THETA)
      DUM(1,3)= -2.*TUSNO*CZI
      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.	CABC1200
DUM(4,1) = (SIN(THGX1)/ALU)*12.	CABC1210
DUM(4,2) = (-GZ1*COT(THGX1)/ALU)*12.	CABC1220
DUM(4,3) = -SNUZ*SIN(THGX1)/ALU-SNUX*GZ1*COT(THGX1)/ALU	CABC1230
DUM(4,4) = -1.	CABC1240
DUM(5,1) = (-GX1*COT(THGZ1)/ALU)*12.	CABC1250
DUM(5,2) = (SIN(THGZ1)/ALU)*12.	CABC1260
DUM(5,3) = SNUZ*GX1*COT(THGZ1)/ALU + SNUX*SIN(THGZ1)/ALU	CABC1270
DUM(5,5) = -1.	CABC1280
CALL DPCSN(THETA)	CABC1290
Q = .5*PHO*V0*V0	CABC1300
ALU1 = ALU+1.	CABC1310
CALL STINT(Q,ALU1,0,1,1,TUSN1,NG)	CABC1320
IF(NG.NE.0) GO TO 5000	CABC1330
ALU2 = ALU-1.	CABC1340
CALL STINT(Q,ALU2,0,1,1,TUSN2,NG)	CABC1350
IF(NG.EQ.0) GO TO 5001	CABC1360
5000 WRITE(IW,5002) NG,ALL,ALU,0	CABC1370
5002 FORMAT('ERROR IN TABLE 1-2,NG=',I2,3X'E10.3')	CABC1380
RETURN	CABC1390
5001 CONTINUE	CABC1400
AKTU = (TUSN1-TUSN2)/2.	CABC1410
DUM(6,6) = -1.	CABC1420
DUM(6,7) = AKTU*12.	CABC1430
DUM(7,1) = -GX1	CABC1440
DUM(7,2) = -GZ1	CABC1450
DUM(7,3) = ((-SNUX+ALU*GX1)*GZ1-(-SNUZ+ALU*GZ1)*GX1)/12.	CABC1460
DUM(7,7) = -1.	CABC1470
CALL MASH(7,7)	CABC1480
DO 1005 I=1,3	CABC1490
DO 1005 J=1,3	CABC1500
1005 FTOP(I,J) = DUM(I,J)	CABC1510
CALL DFCUSN(THETA)	CABC1520
DUM(1,3) = -2.*TLSNO*GZ3	CABC1530
DUM(1,4) = -2.*TLSNO*SIN(THGX3)	CABC1540
DUM(1,6) = 2.*GX3	CABC1550
DUM(2,3) = 2.*TLSNO*GX3	CABC1560
DUM(2,5) = -2.*TLSNO*SIN(THGZ3)	CABC1570
DUM(2,6) = 2.*GZ3	CABC1580
DUM(3,3) = (SNLZ*DUM(1,3)+SNLX*DUM(2,3))/12.	CABC1590
DUM(3,4) = SNLZ*DUM(1,4)/12.	CABC1600
DUM(3,5) = SNLX*DUM(2,5)/12.	CABC1610
DUM(3,6) = (SNLZ*DUM(1,6)+SNLX*DUM(2,6))/12.	CABC1620
DUM(4,1) = (SIN(THGX3)/ALL)*12.	CABC1630
DUM(4,2) = (-GZ3*COT(THGX3)/ALL)*12.	CABC1640
DUM(4,3) = SNLZ*SIN(THGX3)/ALL - SNLX*GZ3*COT(THGX3)/ALL	CABC1650
DUM(4,4) = -1.	CABC1660
DUM(5,1) = (-GX3*COT(THGZ3)/ALL)*12.	CABC1670
DUM(5,2) = (SIN(THGZ3)/ALL)*12.	CABC1680
DUM(5,3) = -SNLZ*GX3*COT(THGZ3)/ALL + SNLX*SIN(THGZ3)/ALL	CABC1690
DUM(5,5) = -1.	CABC1700
CALL DPCSN(THETA)	CABC1710
ALL1 = ALL+1.	CABC1720
CALL STINT(Q,ALL1,0,1,1,TLN1,NG)	CABC1730
IF(NG.NE.0) GO TO 5003	CABC1740

ALL2=ALL-1.	CAB01750
CALL STINT(0,ALL2,0,1,1,TLN2,NG)	CAB01750
IF(NG.EQ.0) GO TO 5004	CAB01770
5003 WRITE(IW,5002) NG,ALL,ALU,0	CAB01780
RETURN	CAB01700
5004 CONTINUE	CAB01810
AKTL=(TLN1-TLN2)/2.	CAB01810
DUM(6,6)= -1.	CAB01320
DUM(6,7)= AKTL*12.	CAB01830
DUM(7,1)= -GX3	CAB01840
DUM(7,2)= -GZ3	CAB01850
DUM(7,3)=((-SNLX+ALL*GX3)*GZ3-(SNL7+ALL*GZ3)*GX3)/12.	CAB01860
DUM(7,7)= -1.	CAB01870
CALL WASH(3,7)	CAB01880
DO 1008 I=1,3	CAB01890
DO 1008 J=1,3	CAB01900
1008 FBOT(I,J)=DUM(I,J)	CAB01910
DO 1009 I=1,3	CAB01920
DO 1009 J=1,3	CAB01930
SNUD(I,J)=0	CAB01940
1009 SNU(I,J)= FBTB(I,J)+FBOT(I,J)	CAB01950
RETURN	CAB01960
1000 IF(KODE(10).EQ.0) GO TO 1002	CAB01970
C TERMS FOR SNUBBER EFFECTS(LONG)	CAB01980
CALL DRCSN(THETA)	CAB01990
DO 1006 I=1,7	CAB02000
DO 1006 J=1,7	CAB02010
1006 DUM(I,J)=0	CAB02020
DUM(1,3)= -2.*TUSND*GZ1	CAB02030
DUM(1,4)= -2.*TUSND*SIN(THGX1)	CAB02040
DUM(1,6)= 2.*GX1	CAB02050
DUM(2,7)= 2.*TUSND*GX1	CAB02060
DUM(2,5)= -2.*TUSND*SIN(THGZ1)	CAB02070
DUM(2,6)= 2.*GZ1	CAB02080
DUM(3,3)= (-SNUZ*DUM(1,3)+SNUX*DUM(2,3))/12.	CAB02090
DUM(3,4)= -SNUZ*DUM(1,4)/12.	CAB02100
DUM(3,5)= SNUX*DUM(2,5)/12.	CAB02110
DUM(3,6)= (-SNUZ*DUM(1,6)+SNUX*DUM(2,6))/12.	CAB02120
DUM(4,1)= (SIN(THGX1)/ALU)*12.	CAB02130
DUM(4,2)= (-GZ1*COT(THGX1)/ALU)*12.	CAB02140
DUM(4,3)= -SNUZ*SIN(THGX1)/ALU-SNUX*GZ1*COT(THGX1)/ALU	CAB02150
DUM(4,4)= -1.	CAB02160
DUM(5,1)= (-GX1*COT(THGZ1)/ALU)*12.	CAB02170
DUM(5,2)= (SIN(THGZ1)/ALU)*12.	CAB02180
DUM(5,3)= SNUZ*GX1*COT(THGZ1)/ALU + SNUX*SIN(THGZ1)/ALU	CAB02190
DUM(5,5)= -1.	CAB02200
DUM(6,6)= -1.	CAB02210
DUM(6,7)= AKSNU*12.	CAB02220
DUM(7,1)= -GX1	CAB02230
DUM(7,2)= -GZ1	CAB02240
DUM(7,7)= ((-SNUX+ALU*GX1)*GZ1-(-SNUZ+ALU*GZ1)*GX1)/12.	CAB02250
DUM(7,7)= -1.	CAB02260
DO 10 I=1,3	CAB02270
DO 10 J=1,3	CAB02280
10 SNUD(I,J)=DUM(I,6)*ADSNU*DUM(7,J)*12.	CAB02290

CALL MASH(3,7)	CAB02300
DO 1007 I=1,3	CAB02310
DO 1007 J=1,3	CAB02320
1007 FTOP(I,J)=DUM(I,J)	CAB02330
DUM(1,3)= -2.*TLSND*GZ3	CAB02340
DUM(1,4)= -2.*TLSND*SIN(THGX3)	CAB02350
DUM(1,6)= 2.*GX3	CAB02360
DUM(2,3)= 2.*TLSND*GX3	CAB02370
DUM(2,5)= -2.*TLSND*SIN(THGZ3)	CAB02380
DUM(2,6)= 2.*GZ3	CAB02390
DUM(3,3)= (SNL7*DUM(1,3)+SNLX*DUM(2,3))/12.	CAB02400
DUM(3,4)= SNL7*DUM(1,4)/12.	CAB02410
DUM(3,5)= SNLX*DUM(2,5)/12.	CAB02420
DUM(3,6)= (SNLZ*DUM(1,6)+SNLX*DUM(2,6))/12.	CAB02430
DUM(4,1)= (SIN(THGX3)/ALL)*12.	CAB02440
DUM(4,2)= (-GZ3*COT(THGX3)/ALL)*12.	CAB02450
DUM(4,3)= SNLZ*SIN(THGX3)/ALL - SNLX*GZ3*COT(THGX3)/ALL	CAB02460
DUM(4,4)= -1.	CAB02470
DUM(5,1)= (-GX3*COT(THGZ3)/ALL)*12.	CAB02480
DUM(5,2)= (SIN(THGZ3)/ALL)*12.	CAB02490
DUM(5,3)= -SNLZ*GX3*COT(THGZ3)/ALL + SNLX*SIN(THGZ3)/ALL	CAB02500
DUM(5,5)= -1.	CAB02510
DUM(6,6)= -1.	CAB02520
DUM(6,7)= AKSN *12.	CAB02530
DUM(7,1)= -GX	CAB02540
DUM(7,2)= -GZ3	CAB02550
DUM(7,3)= ((-SNLX+ALL*GX3)*GZ3 - (SNLZ+ALL*GZ3)*GX3)/12.	CAB02560
DUM(7,7)= -1.	CAB02570
DO 20 I=1,3	CAB02580
DO 20 J=1,3	CAB02590
20 SNUD(I,J)= SNU(I,J)+DUM(I,6)*30*SNL*DUM(7,J)*12.	CAB02600
CALL MASH(3,7)	CAB02610
DO 1010 I=1,3	CAB02620
DO 1010 J=1,3	CAB02630
1010 FBOT(I,J)=DUM(I,J)	CAB02640
DO 1011 I=1,3	CAB02650
DO 1011 J=1,3	CAB02660
1011 SNU(I,J)= FTOP(I,J)+FBOT(I,J)	CAB02670
RETURN	CAB02680
1002 DO 1003 I=1,3	CAB02690
DO 1003 J=1,3	CAB02700
SNUD(I,J)=0	CAB02710
1003 SNU(I,J)=0	CAB02720
RETURN	CAB02730
END	CAB02740
SUBROUTINE DFCSN(THETA)	CAB00010
COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL	CAB00020
COMMON/SNUH3/SNU(3,3),SN(30),THUSN,THLSN,SNUD(3,3)	CAB00030
EQUIVALENCE(AERO(105), SNUX),(AERO(106), SNUY),(AERO(107), SNUZ),	CAB00040
1 (AERO(108), SNLX),(AERO(109), SNLY),(AERO(110), SNLZ),	CAB00050
2 (AERO(111), SNUST),(AERO(112), SNUWL),(AERO(113), SNUBL),	CAB00060
3 (AERO(114), SNLST),(AERO(115), SNLWL),(AERO(116), SNLBL),	CAB00070
4 (AERO(117), TUSND),(AERO(118), TLSND),,AERO(119), AKSNU),	CAB00080
5 (AERO(120), AKSNL),	CAB00090
6 (AERO(76), WLCH),(AERO(77), STACF),(AERO(78), BLCF)	CAB00100

	EQUIVALENCE (SN(1), GX1),(SN(2), GY1),(SN(3), GZ1),	CAB00110
1	(SN(4), GX2),(SN(5), GY2),(SN(6), GZ2),	CAB00120
2	(SN(7), GX3),(SN(8), GY3),(SN(9), GZ3),	CAB00130
3	(SN(10), GX4),(SN(11), GY4),(SN(12), GZ4),	CAB00140
4	(SN(13), THU),(SN(14), THL),(SN(15), ALU),	CAB00150
5	(SN(16), ALL),	CAB00160
6	(SN(19), THGX1),(SN(20), THGY1),(SN(21), THGZ1),	CAB00170
7	(SN(22), THGX2),(SN(23), THGY2),(SN(24), THGZ2),	CAB00180
8	(SN(25), THGX3),(SN(26), THGY3),(SN(27), THGZ3),	CAB00190
9	(SN(28), THGX4),(SN(29), THGY4),(SN(30), THGZ4)	CAB00200
C	CALCULATION OF SNUBBED CABLE DIRECTION COSINES	CAB00210
	X31= (STACR-SNUST)*COS(THETA)-(WLCR-SNUWL)*SIN(THETA)	CAB00220
	Z31= (WLCR-SNUWL)*COS(THETA)+(STACR-SNUST)*SIN(THETA)	CAB00230
	X32= X31	CAB00240
	Z32= Z31	CAB00250
	X33= (STACR-SNLST)*COS(THETA)-(WLCR-SNLWL)*SIN(THETA)	CAB00260
	Z33= (WLCR-SNLWL)*COS(THETA)+(STACR-SNLST)*SIN(THETA)	CAB00270
	X34= X33	CAB00280
	Z34= Z33	CAB00290
	DX1= X31+SNUX	CAB00300
	DY1= -SNUBL+SNUY	CAB00310
	DZ1= Z31+SNUZ	CAB00320
	DX2= DX1	CAB00330
	DY2= SNUBL-SNUY	CAB00340
	DZ2= DZ1	CAB00350
	DX3= X33+SNLX	CAB00360
	DY3= SNLRL-SNLY	CAB00370
	DZ3= Z33-SNLZ	CAB00380
	DX4= DX3	CAB00390
	DY4= -SNLBL+SNLY	CAB00400
	DZ4= DZ3	CAB00410
	ALUSQ= DX1**2 + DY1**2 + DZ1**2	CAB00420
	ALU = SQRT(ALUSQ)	CAB00430
	ALLSQ = DX3**2 + DY3**2 + DZ3**2	CAB00440
	ALL = SQRT(ALLSQ)	CAB00450
	GX1 = DX1/ALU	CAB00460
	GY1 = DY1/ALU	CAB00470
	GZ1 = DZ1/ALU	CAB00480
	GX2 = DX2/ALU	CAB00490
	GY2 = DY2/ALU	CAB00500
	GZ2 = DZ2/ALU	CAB00510
	GX3 = DX3/ALL	CAB00520
	GY3 = DY3/ALL	CAB00530
	GZ3 = DZ3/ALL	CAB00540
	GX4 = DX4/ALL	CAB00550
	GY4 = DY4/ALL	CAB00560
	GZ4 = DZ4/ALL	CAB00570
	DO I I=19,30	CAB00580
	J=I-18	CAB00590
	I SN(I)=ARCOS(SN(J))	CAB00600
	RETURN	CAB00610
	END	CAB00620
	SUBROUTINE DFCUSN(THETA)	CAB00630
	COMMON/DAT/ASFO(175),ASFOR(50),KODE(26),LL	CAB00640
	COMMON/SNUBB/SNU(3,3),SN(20),THUSN,THLSN,SNUD(3,3)	CAB00650

```

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), TUSNC),(AERO(118), TLSNO),(AERO(119), AKSNU), CAB00700
5      (AERO(120), AKSNL), CAB00710
6      (AERO(76), WLCR),(AERO(77), STACP),(AERO(78), BLCR) 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-(BLCR+SNLY) CAB00840
AZL =-SNLW- (WLCR+SNLZ+SNLX*SIN(THETA)) CAB00850
AYU = SNUBL-(BLCR+SNUY) CAB00860
AZU = SNUWL-(WLCR+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
: SN(I)=ARCOS(SN(J)) CAB01120
RETURN CAB01130
END CAB01140
SUBROUTINE PITE CAB01150
COMMON/INDL I,J,IF CAB01160
COMMON/DAT/A,AL(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,MINTRL,MAXTRL,FCT,NG)	CAB00010
	EQUIVALENCE (X(1),NUMPTS(1))	CAB00020
	COMMON NUMPTS(1)	CAB00030
	DIMENSION X(1)	CAB00040
	I7=NUMPTS(1)/3	CAB00050
70	IF(MINTRL-MAXTRL)71,71,110	CAB00060
71	DO 73 II=MINTRL,MAXTRL	CAB00070
	NJ=NUMPTS(II)+1	CAB00080
	IF(A3-X(NJ))72,74,73	CAB00090
72	IF(II-MINTRL) 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	CAR00190
ID = NUMPTS(NI)	CAR00190
IP = ID+NJ	CAR00200
DD 77 IO=1, ID	CAR00210
NN= NJ+ID	CAR00220
IF (A1-X(NN))76,79,77	CAR00230
76 IF(IO-1) 110,112,79	CAR00240
77 CONTINUE	CAR00250
GO TO 112	CAR00260
78 IG =-1	CAR00270
GO TO 90	CAR00280
79 IG =+1	CAR00290
80 NI=NI+IZ	CAR00300
IS = NUMPTS(NI)	CAR00310
DD 92 IA=1, IS	CAR00320
NS=IP+IA	CAR00330
IF (A2-X(NS))81,83,82	CAR00340
81 IF(IA-1) 110,112,84	CAR00350
82 CONTINUE	CAR00360
GO TO 112	CAR00370
83 IH =-1	CAR00380
GO TO 85	CAR00390
84 IH =+1	CAR00400
85 NE=IP+IP+IO+IO*IA-ID	CAR00410
NR=NE-ID	CAR00420
IF(IG+IH) 86,88,91	CAR00430
86 IF (X(NE)-99998.5E9)87,113,113	CAR00440
87 FCT = X(NE)	CAR00450
GO TO 95	CAR00460
88 IF(IG) 89,110,93	CAR00470
89 IF(AMAX1(X(NF),X(NR))-99998.5E9)90,113,113	CAR00480
90 FCT = X(NE)-(X(NS)-A2)*(X(NE)-X(NF))/(X(NS)-X(NS-1))	CAR00490
GO TO 95	CAR00500
91 IF(AMAX1(X(NE),X(NF),X(NE-1),X(NR-1))-99998.5E9)92,113,113	CAR00510
92 FCT = ((X(NS)-A2)*((X(NN)-A1)*X(NE-1)-(X(NN-1)-A1)*X(NF)	CAR00520
1)-(X(NS-1)-A2)*((X(NN)-A1)*X(NE-1)-(X(NN-1)-A1)*X(NE)))	CAR00530
2/((X(NS)-X(NS-1))*(X(NN)-X(NN-1)))	CAR00540
GO TO 95	CAR00550
93 IF(AMAX1(X(NE), X(NE-1))-99998.5E9) 94,113,113	CAR00560
94 FCT = X(NE)-(X(NN)-A1)*(X(NE)-X(NE-1))/(X(NN)-X(NN-1))	CAR00570
95 GO TO (96,98,99),IF	CAR00580
96 DUMSTG =FCT	CAR00590
97 II =II-1	CAR00600
98 FCT =DUMSTG-(X(NM)-A3)*(DUMSTG-FCT)/(X(NM)-X(NJ))	CAR00610
99 RETURN	CAR00620
74 IK =3	CAR00630
IL =3	CAR00640
GO TO 101	CAR00650
110 NG =2	CAR00660
GO TO 99	CAR00670
112 NG =3	CAR00680
GO TO 99	CAR00690
113 NG =4	CAR00700
GO TO 99	CAR00710
END	CAR00720

SUBROUTINE TABIN(NUMTBL,NZ,NG)	CAB00730
COMMON NUMPTS(1)	CAB00740
COMMON/INOUT/IW,IR	CAB00750
COMMON /TABOUT/ NIMTBL,ISEQ	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+I7+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,27A2,1A6)	CAB00910
57 FORMAT(8X14,2I2,27A2,12)	CAB00920
3 IF(ISEQ) 60,58,69	CAB00930
58 IF(K) 90, 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	CAB00990
17 NUMPTS(NIMTBL) = MUMMY	CAB01000
70 N1 = (LIN-1) / 9 + 1	CAB01010
DO 68 IS = 1,N1	CAB01020
L3 = (IS-1) * 9 + 1	CAB01030
IF (IS-N1) 60, 61, 60	CAB01040
60 L4 = L3 + 8	CAB01050
GO TO 62	CAB01060
61 L4 = LIN	CAB01070
62 L5 = NUMPTS(NIMTBL) + 1	CAB01080
L6 = L5 + L3	CAB01090
L7 = L5 + L4	CAB01100
JJ = 0	CAB01110
LM = L5 + LIN	CAB01120
LN = LM + L2N	CAB01130
63 READ(NUNIT,64) (DUMMY(K),K=1,10), ISEQ	CAB01140
64 FORMAT (10E7,0,12)	CAB01150
IF(MCR.EQ.0) GO TO 5	CAB01160
6 WRITE(IW,2)DUMMY,ISEQ	CAB01170
2 FORMAT(10E12,4,15)	CAB01180
5 XUMPTS(L5)= DUMMY(1)	CAB01190
K = 2	CAB01200
DO 65 J = L6,L7	CAB01210
XUMPTS(J) = DUMMY(K)	CAB01220
65 K = K+1	CAB01230
ISEQ=(IS-1)*(L2N+1)+JJ+1	CAB01240
IF(ISEQ-ISEQ) 60,56,69	CAB01250
66 L6 = LN + L3	CAB01260
L7 = LN + L4	CAB01270

	L5 = LM + 1 + JJ	CAB01280
	IF (JJ-L2N) 67, 63, 69	CAB01290
67	JJ = JJ + 1	CAB01300
	LN = LN + LIN	CAB01310
	GO TO 63	CAB01320
68	CONTINUE	CAB01330
100	MUMMY = NUMPTS(NIMTBL) + (LIN+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,FACT,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	IL = 1	CAB00130
	IL = 2	CAB00140
	NM=NJ	CAB00150
101	DO 97 IF=IK,IL	CAB00160
	NJ =NUMPTS(II)+1	CAB00170
	NI = IZ+II	CAB00180
	ID =NUMPTS(NI)	CAB00190
	IP =ID+NJ	CAB00200
	DO 77 IQ=1,ID	CAB00210
	NV= NJ+IQ	CAB00220
	IF (A1-X(NV))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
	IR = NUMPTS(NI)	CAB00310
	DO 82 IA=1,IR	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
	IH =+1	CAB00400
	IJ = IJ+IH+IQ+IQ*IA-ID	CAB00410
	IJ = IJ - IJ	CAB00420
	IF 86,88,91	CAB00430

86	IF (X(NF)-99998.5E9)87,113,113	CAB00440
87	FCT = X(NE) GO TO 95	CAB00450 CAB00460
88	IF(IG) 89,110,93	CAB00470
89	IF(AMAX1(X(NE),X(NR))-99998.5E9)90,113,113	CAB00480
90	FCT = X(NE)-(X(NS)-A2)*(X(NE)-X(NF))/(X(NS)-X(NS-1)) GO TO 95	CAB00490 CAB00500
91	IF(AMAX1(X(NE),X(NF),X(NE-1),X(NR-1))-99998.5E9)92,113,113	CAB00510
92	FCT = ((X(NS)-A2)*((X(NN)-A1)*X(NF-1)-(X(NN-1)-A1)*X(NF) 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	CAB00520 CAB00530 CAB00540
93	IF(AMAX1(X(NE), X(NE-1))-99998.5E9) 94,113,113	CAB00550
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 TABIN(NUMTBL,NZ,NG)	CAB00730
	COMMON/INPUT/IW,IF	CAB00740
	COMMON/TAB1/NUMPTS(1)	CAB00750
	COMMON /TABQUI/ NIMTBL,ISFO	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+17	CAB00870
102	READ(NUNIT,57) K, N, L2N, LABEL, ISFO	CAB00880
	IF(MCR.EQ.0) GO TO 3	CAB00890
4	WRITE(IW,1) K,LIN,L2N,LABEL,ISFO	CAB00900
1	FORMAT(3I5, 10X,27A2,I46)	CAB00910
57	FORMAT(9X14,2I2,27A2,I2)	CAB00920
3	IF(ISFO) 69,58,69	CAB00930
58	IF(K) 99, 99, 59	CAB00940
59	M = IZ + NIMTBL	CAB00950
	NUMPTS(M) = LIN	CAB00960
	M = M + 17	CAB00970
	NUMPTS(M) = L2N	CAB00980

IF(NUMTBL-NIMTBL)17,70,17	CABC0090
17 NUMPTS(NIMTBL) = MUMMY	CABC1000
70 N1 = (LIN-1) / 9 + 1	CABC1010
DO 68 IS = 1,N1	CABC1020
L3 = (IS-1) * 9 + 1	CABC1030
IF (IS-N1) 60, 61, 60	CABC1040
60 L4 = L3 + 8	CABC1050
GO TO 62	CABC1060
61 L4 = LIN	CABC1070
62 L5 = NUMPTS(NIMTBL) + 1	CABC1080
L6 = L5 + L3	CABC1090
L7 = L5 + L4	CABC1100
JJ = 0	CABC1110
LM = L5 + LIN	CABC1120
LN = LM + L2N	CABC1130
63 READ(NUNIT,64) (DUMMY(K),K=1,10), ISEQ	CABC1140
64 FORMAT (10E7,0,I2)	CABC1150
IF(MCF.EQ.0) GO TO 5	CABC1160
6 WRITE(IW,2)DUMMY,ISEQ	CABC1170
2 FORMAT(10E12,4,I5)	CABC1180
5 XUMPTS(L5)= DUMMY(1)	CABC1190
K = 2	CABC1200
DO 65 J = L6,L7	CABC1210
XUMPTS(J) = DUMMY(K)	CABC1220
65 K = K+1	CABC1230
ISDQ=(IS-1)*(L2N+1)+JJ+1	CABC1240
IF (ISEQ-ISDQ) 69,66,69	CABC1250
65 L6 = LN + L3	CABC1260
L7 = LN + L4	CABC1270
L5 = LM + 1 + JJ	CABC1280
IF (JJ-L2N) 67, 68, 69	CABC1290
67 JJ = JJ + 1	CABC1300
LN = LN + LIN	CABC1310
GO TO 63	CABC1320
68 CONTINUE	CABC1330
109 MUMMY = NUMPTS(NIMTBL) + (LIN+1) * (L2N+1)	CABC1340
108 NIMTBL = NIMTBL + 1	CABC1350
GO TO 102	CABC1360
69 NG = 1	CABC1370
99 RETURN	CABC1380
END	CABC1390
SUBROUTINE FRIC(IX)	CABC0010
COMMON/LAT/AERO(175),AEROP(50),KODE(26)	CABC0020
COMMON/ROUGH/FRIC(3,6)	CABC0030
EQUIVALENCE (AERO(96),COU),(AERO(104),CMP)	CABC0040
DO 1 I=1,3	CABC0050
DO 1 J=1,6	CABC0060
1 FRIC(I,J)=0.	CABC0070
IF(CMP.EQ.0..AND.COUEQ.0.)RETURN	CABC0080
IND=KODE(6)	CABC0090
IF(IX.NE.0)GO TO 2	CABC0100
C LONGITUDINAL PULLEY FRICTION COMPUTATION	CABC0110
GO TO(10,11,12,13),IND	CABC0120
10 CALL FRVT(1)	CABC0130
RETURN	CABC0140

11 CALL FRVT(3)	CABC0150
RETURN	CABC0160
12 CALL FRVT(1)	CABC0170
CALL FRVT(3)	CABC0180
13 RETURN	CABC0190
C LATERAL DIRECTIONAL FRICTION COMPUTATION	CABC0200
2 GO TO(20,21,22,23),IND	CABC0210
20 CALL FRHZ(3)	CABC0220
RETURN	CABC0230
21 CALL FRHZ(1)	CABC0240
22 RETURN	CABC0250
23 CALL FRHZ(1)	CABC0260
CALL FRHZ(3)	CABC0270
RETURN	CABC0280
END	CABC0290
SUBROUTINE FRVT(IC)	CABC0300
C COMPUTES THE FRICT. EFFECT OF THE VERT PULLEYS ON THE LONG. DYN.	CABC0310
COMMON/DAT/AERO(175),AERDP(50),KODE(26)	CABC0320
COMMON/PLYCHA/RTD,XLGTH(5),ADC(5,3),ARM(5,3),TR,TLFT,TF	CABC0330
COMMON/ROUGH/FRIC(3,6)	CABC0340
EQUIVALENCE (AERO(90),RVF),(AERO(92),RVR),(AERO(96),COU).	CABC0350
1(AERO(IC4),CMP)	CABC0360
DIMENSION DT1(3),DT2(3)	CABC0370
IF(IC.EQ.3)GO TO 1	CABC0380
TENS=TF	CABC0390
RAD=RVF/12.	CABC0400
AVX=(ADC(2,1)-ADC(1,1))/2.	CABC0410
CAX=COS(AVX)	CABC0420
CAZ=SIN(AVX)	CABC0430
GO TO 2	CABC0440
1 TENS=TR	CABC0450
RAD=RVR/12.	CABC0460
AVX=3.14159+(ADC(4,1)-ADC(3,1))/2.	CABC0470
CAX=COS(AVX)	CABC0480
CAZ=SIN(AVX)	CABC0490
2 ARMX=(ARM(IC,1)+ARM(IC+1,1))/24.	CABC0500
ARMZ=(ARM(IC,3)-ARM(IC,3))/24.	CABC0510
ENORX=TENS*COS(ADC(IC,1))	CABC0520
ENORZ=TENS*(1.+COS(ADC(IC,3)))	CABC0530
ENORM=SQRT(ENORX**2+ENORZ**2)	CABC0540
CMPP=CMP/ENORM	CABC0550
FACU=CMPP*ENORM/RAD**2	CABC0560
ENORX=TENS*COS(ADC(IC+1,1))	CABC0570
ENORZ=TENS*(1.+COS(ADC(IC+1,3)))	CABC0580
ENORM=SQRT(ENORX**2+ENORZ**2)	CABC0590
CMPP=CMP/ENORM	CABC0600
FACL=CMPP*ENORM/RAD**2	CABC0610
FACT=4.*COU/(3.14159*RAD**2)	CABC0620
CALL DLGTH(CX,CZ,CT,IC,C)	CABC0630
CALL DLGTH(CXP,CZP,CTP,IC+1,C)	CABC0640
DT1(1)=FACT*(CXP-CX)	CABC0650
DT1(2)=FACT*(CZP-CZ)	CABC0660
DT1(3)=FACT*(CTP-CT)	CABC0670
DT2(1)=FACL*CXP-FACU*CX	CABC0680
DT2(2)=FACL*CZP-FACU*CZ	CABC0690

```

DT2(3)=FACL*CTP-FACU*CT
DO 3 I=1,3
FRIC(1,1)=FRIC(1,1)+DT1(I)*CAX
FRIC(1,I+3)=FRIC(1,I+3)+DT2(I)*CAX
FRIC(2,1)=FRIC(2,1)+DT1(I)*CAZ
FRIC(2,I+3)=FRIC(2,I+3)+DT2(I)*CAZ
FRIC(3,1)=FRIC(3,1)+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 FRIC. EFFECT OF THE HORIZ PULLEYS ON THE LAT. DIR. DYN.
COMMON/DAT/AERO(175),AEROP(50),KODE(26)
COMMON/PLYCHA/RTD,XLGTH(5),ADC(5,3),AFM(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(ENORX*ENORX+ENORY*ENORY)
CMPD=CMPP/ENORM
FACL=CMPP*ENORM/RAD**2
FACT=4.*COU/(7.14159*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(1,1)=FRIC(1,1)+DT1(I)*COS(ADC(IC,2))
FRIC(1,I+3)=FRIC(1,I+3)+DT2(I)*COS(ADC(IC,2))
FRIC(2,1)=FRIC(2,1)+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(3,1)=FRIC(3,1)+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,FOOTS,KAA,IEF)
COMMON/DAT/AERO(175),AEROP(50),KODE(26),LL

```

COMMON/FRQ/C4(30)	CBL00030
DIMENSION CMAT(14,14,3),MAT(14,14),KOUNT(30),C5(30)	CBL00040
COMPLEX AMAT(14,14),ROOTS(29)	CBL00050
DOUBLE PRECISION BMAT(14,14,30),D(30,4)	CBL00060
NP=-2	CBL00070
IF(KODE(5),EQ,1) NP=1	CBL00080
CALL MAPDY(CMAT,C4,ROOTS,K4A,14,NP,3,30,KOUNT,	CBL00090
1 AMAT,BMAT,MAT,C5,D,N)	CBL00100
RETURN	CBL00110
END	CBL00120
SUBROUTINE MAPDL(CMAT,C4,ROOTS,K4A,MCOL,NP,	CBL00130
1 IN, N, KOUNT, AMAT, BMAT, MAT, C5, D)	CBL00140
COMMON/INDUT/IW,IP	CBL00150
DIMENSION AMAT(MCOL,1), MAT(MCOL,1), BMAT(MCOL,MCOL,1)	CBL00160
1 ,C4(1), ROOTS(1), KOUNT(1),CMAT(MCOL,MCOL,1)	CBL00170
2 ,C5(1)	CBL00180
DOUBLE PRECISION BMAT, SA, F, D(N,1),DR	CBL00190
* , FBMAT	CBL00200
COMPLEX DET,FCOMPLX, AMAT, G, YA1, YA, SCOMPLX	CBL00210
COMPLEX G3, ROOTS,C5	CBL00220
12 FORMAT (2I3, 1P5D16.6/(D22.6,4D16.6))	CBL00230
14 FORMAT(1H0,2(1PE24.6,E16.6))	CBL00240
15 FORMAT(1H-,14X,4HPREAL,11X,9HIMAGINARY,19X,5HEFFOR)	CBL00250
22 FORMAT (3I3, 1P5E16.6/(E25.6,4E16.6))	CBL00260
2614 FORMAT (/1PE24.6,F:16.6,E30.6)	CBL00270
DATA CR/Z7FFFFFFFFFFFFFFFF/	CBL00280
NCOL=MCOL	CBL00290
10 NROW=NCOL	CBL00300
ENP=10.**NP	CBL00310
INN=IN+1	CBL00320
DO 107 I=1,NROW	CBL00330
DO 107 J=1,NCOL	CBL00340
MAT(I,J) = 0	CBL00350
DO 112 K=INN,N	CBL00360
112 BMAT(I,J,K)=0.00	CBL00370
DO 107 K=1,IN	CBL00380
BMAT(I,J,K)=CMAT(I,J,K)	CBL00390
IF(CMAT(I,J,K))108,107,108	CBL00400
108 MAT(I,J) = K	CBL00410
C THE NUMBER IN MAT IS ONE GREATER THAN THE DEGREE OF THE POLYNOMIAL	CBL00420
107 CONTINUE	CBL00430
JS= 1	CBL00440
IF(NP.LT.0)GO TO 128	CBL00450
ASSIGN 128 TO MZ	CBL00460
GO TO 920	CBL00470
99 ASSIGN 257 TO MZ	CBL00480
920 WRITE(IW,23)	CBL00490
23 FORMAT(55HPOSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX)	CBL00500
DO 951 J9= 1,NCOL	CBL00510
DO 951 I9= 1,NCOL	CBL00520
K1 = MAT(I9,J9)	CBL00530
IF(K1) 951,951,952	CBL00540
952 WRITE(IW,12)I9,J9, (BMAT(I9,J9,K), K=1,K1)	CBL00550
951 CONTINUE	CBL00560
GO TO MZ, (130,257,128,1105)	CBL00570

C DET CONTAINS VALUE OF DETERMINANT OF BMAT WITH G=1	CBL00580
1281 WRITE(IW,1282)DET	CBL00590
1282 FORMAT (' 12H DETERMINANT1P2F16.7)	CBL00600
128 NC = 0	CBL00610
C COUNT NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN COLUMN JS	CBL00620
DO 120 I=JS,NDJW	CBL00630
IF (MAT(I,JS))99,120,121	CBL00640
121 NC = NC +1	CBL00650
IS = I	CBL00660
120 CONTINUE	CBL00670
IF(NC-1) 17,125,130	CBL00680
17 WRITE(IW,16)	CBL00690
16 FORMAT (' MATRIX IS SINGULAR')	CBL00700
GO TO 257	CBL00710
125 IF (IS-JS)99,1401,123	CBL00720
C ONE INTER CHANGE TRIANGULARIZES THE COLUMN.	CBL00730
123 DO 126 J=JS,NCOL	CBL00740
K1 = MAX0(MAT(IS,J), MAT(JS,J))	CBL00750
MA = MAT(IS,J)	CBL00760
MAT(IS,J) = MAT(JS,J)	CBL00770
MAT(JS,J) = MA	CBL00780
DO 126 K= 1,K1	CBL00790
SA = BMAT(IS,J,K)	CBL00800
BMAT(IS,J,K) = BMAT(JS,J,K)	CBL00810
126 BMAT(JS,J,K) =-SA	CBL00820
GO TO 1401	CBL00830
130 IS = JS+1	CBL00840
C LOOP 137 REDUCES ALL ELEMENTS BELOW DIAGONAL IN COLUMN JS BY	CBL00850
C AT LEAST ONE DEGREE	CBL00860
I=IS	CBL00870
130 IF(MAT(I,JS))99,137,129	CBL00880
129 IF (MAT(JS,JS)) 99,133,132	CBL00890
132 IF (MAT(I,JS) - MAT(JS,JS)) 133,134,134	CBL00900
133 DO 131 J= JS,NCOL	CBL00910
K1= MAX0(MAT(JS,J), MAT(I,J))	CBL00920
MA = MAT(JS,J)	CBL00930
MAT(JS,J) = MAT(I,J)	CBL00940
MAT(I,J) = MA	CBL00950
DO 131 K= 1,K1	CBL00960
SA = BMAT(I,J,K)	CBL00970
BMAT(I,J,K) = BMAT(JS,J,K)	CBL00980
131 BMAT(JS,J,K) =-SA	CBL00990
GO TO 139	CBL01000
134 KI = MAT(I,JS)	CBL01010
KJS = MAT(JS,JS)	CBL01020
KD = KI - KJS	CBL01030
F = BMAT(I,JS,KI)/ BMAT(JS,JS,KJS)	CBL01040
IF(DABS(F)- 4.0) 1052,1051,1051	CBL01050
1051 IF(KD) 99,133,1052	CBL01060
1052 DO 235 J=JS,NCOL	CBL01070
KJS = MAT(JS,J)	CBL01080
IF(KJS.EQ.0) GO TO 235	CBL01090
DO 135 K= 1,KJS	CBL01100
KI = K + KD	CBL01110
BMAT=F*BMAT(JS,J,K)	CBL01120


```

IF(KI-N) 141,141,2                                CBL01130
2 WRITE(IW,3)                                       CBL01140
3 FORMAT(79H0DEGREE OF POLYNOMIAL FORMED WHILE TRIANGULARIZING ORIGINAL MATRIX IS TOO HIGH ) CBL01150
GO TO 257                                           CBL01160
141 IF ( DABS ( FBMAT - BMAT(I,J,KI) ) .LE. 2.D-6 * DABS ( FBMAT ) ) CBL01170
1 GO TO 136                                         CBL01180
BMAT(I,J,KI)= BMAT(I,J,KI)- EBMAT                 CBL01190
GO TO 135                                           CBL01200
136 BMAT(I,J,KI) = 0.D0                            CBL01210
135 CONTINUE                                        CBL01220
235 CONTINUE                                        CBL01230
C                                                    CBL01240
C                                                    DFONTS LLC CBL01250
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 ( . . .E.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
KI = MAT(I,J)                                       CBL01480
IF(KI) 99,151,152                                   CBL01490
152 WRITE(IW,12)I, J, (BMAT(I,J,K), K=1,KI)        CBL01500
151 CONTINUE                                        CBL01510
153 LK=1                                             CBL01520
C_LOOP 160-ROOTS OF POLYNOMIALS ON DIAGONAL OF TRIANGULARIZED MATRIX ARE CBL01530
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
KI = MAT(J,J)                                       CBL01590
MMM=KI+1                                           CBL01600
K2=KI-1                                             CBL01610
DO 163 K=1,KI                                       CBL01620
MM=MMM-K                                           CBL01630
163 D(MM,4)= BMAT(J,J,K)                            CBL01640
IF(KI.EQ.1) GO TO 162C                              CBL01650
161 CALL POLYRT(D(1,4),ROOTS(2*LK-1),KOUNT(LK),K2,2(1,1),D(1,2),D(1,3) CBL01660
1)                                                  CBL01670

```

KM1=K1/2	CALC1680
LK=LK+KM1	CALC1690
IF(MOD(K1,2).NE.0)GO TO 1020	CALC1700
C DUMMY ELEMENT STORED IN ARRAY ROOTS IF POLYNOMIAL IS OF ODD DEGREE	CALC1710
54 ROOTS(2*LK-2)=0	CALC1720
1020 IF(J.EQ.1)GO TO 1004	CALC1730
1001 DO 1002 K=1,K4A	CALC1740
1002 C5(K) = C4(K)	CALC1750
DO 1006 K=1,N	CALC1760
1006 C4(K) = 0.0	CALC1770
IF(K1) 99,160,1000	CALC1780
1000 DO 1003 K=1,K1	CALC1790
MM=MM:	CALC1800
DO 1003 K3=1,K4A	CALC1810
K4 = K+K3-1	CALC1820
1003 C4(K4) = C4(K4) + D(MM,4)*C5(K3)	CALC1830
K4A = K4	CALC1840
GO TO 160	CALC1850
1004 DO 1005 K=1,K1	CALC1860
MM=MM+K	CALC1870
1005 C4(K) = D(MM,4)	CALC1880
K4A = K1	CALC1890
160 CONTINUE	CALC1900
CALL JUGGLE(ROOTS,ROOTS,KOUNT,K4A)	CALC1910
DO 306 I=1,NROW	CALC1920
DO 306 J=1,NCOL	CALC1930
MAT(I,J)=IN	CALC1940
DO 306 K=1,IN	CALC1950
306 BMAT(I,J,K)=CMAT(I,J,K)	CALC1960
IF(NP.LT.-1)GO TO 202	CALC1970
201 WRITE(IW,15)	CALC1980
202 IF(LK.EQ.1) GO TO 1110	CALC1990
1111 L=1	CALC2000
62 G=ROOTS(L)	CALC2010
64 ASSIGN 244 TO MDT	CALC2020
GO TO 2511	CALC2030
244 G1=ABS(C4(1))	CALC2040
C LOOP 2610 - PLACE LARGEST PRODUCT,C4(I)*G**(I-1), IN G1	CALC2050
C G3= ERROR ESTIMATE G=ROOT	CALC2060
DO 2610 L9=2,K4A	CALC2070
G2=CABS(G)	CALC2080
G2=ABS(C4(L9)*G2**(L9-1))	CALC2090
IF(G1-G2)2611,2610,2610	CALC2100
2611 G1=G2	CALC2110
2610 CONTINUE	CALC2120
C DET CONTAINS VALUE OF POLYNOMIAL EQUIVALENT OF DETERMINANT OF	CALC2130
C MATRIX AT ROOT	CALC2140
IF(G1.EQ.0.)GO TO 25	CALC2150
G3=DET/G1	CALC2160
GO TO 26	CALC2170
25 G3=(0.,0.)	CALC2180
26 IF(CABS(G3).LE.END.AND.NP.LT.-1) GO TO 255	CALC2190
WRITE(IW,27)	CALC2200
27 FORMAT(5X,'THE FOLLOWING EXTRACTED ROOT HAVE POOR ACCURACY')	CALC2210
WRITE(IW,15)	CALC2220

```

WRITE(IV,14) G,G3                                CBL02230
256 L=L+1                                          CBL02240
IF(K4A-1.GE.L)G7 TO 62                            CBL02250
IF(NP.LT.-1) GO TO 257                            CBL02260
1110 WRITE(IV,1010)(C4(K),K=1,K4A)                CBL02270
1010 FORMAT (11HOPOLYNOMIAL1P5E16.6/(E27.6,4E16.6)) CBL02280
257 RETURN                                          CBL02290
1105 ASSIGN 1291 TO MDT                            CBL02300
G=(1.,0.)                                          CBL02310
C LOOP 210                                         CBL02320
C EVALUATE EACH POLYNOMIAL OF THE ORIGINAL MATRIX FOR ROOT G CBL02330
C AND STORE IN AMAT ARRAY                          CBL02340
2511 DO 210 I = 1,NROW                              CBL02350
DO 210 J=1,NCOL                                    CBL02350
K = MAT(I,J)                                       CBL02370
YA=(0.,0.)                                         CBL02390
IF(K-1) 210,205,227                                CBL02390
227 YA=CMPLX(SNGL(BMAT(I,J,K)),0.)                 CBL02400
K = K-1                                             CBL02410
205 YA1=CMPLX(SNGL(BMAT(I,J,K)),C.)                CBL02420
YA=YA1+YA*G                                         CBL02430
K = K-1                                             CBL02440
IF(K) 99,210,205                                    CBL02450
210 AMAT(I,J)=YA                                    CBL02460
JJ=1                                                CBL02470
225 DO 213J=JJ,NCOL                                CBL02480
IF(CABS(AMAT(JJ,J))) 220,213,220                  CBL02490
213 CONTINUE                                        CBL02500
DET = (0.,0.)                                       CBL02510
GO TO 229                                           CBL02520
227 IF(J-JJ)99,230,221                              CBL02530
221 DO 222 I= 1,NROW                                CBL02540
SCMPLX = AMAT(I,J)                                  CBL02550
AMAT(I,J)=AMAT(I,JJ)                               CBL02560
222 AMAT(I,JJ)=-SCMPLX                              CBL02570
230 JSI = JJ + 1                                    CBL02590
DO 224 I=JSI,NROW                                  CBL02590
FCMPLX=AMAT(I,JJ)/AMAT(JJ,JJ)                     CBL02600
IF(CABS(FCMPLX)) 226,224,226                       CBL02610
226 DO 223 J=JJ,NCOL                                CBL02620
223 AMAT(I,J)=AMAT(I,J)-AMAT(JJ,J)*FCMPLX         CBL02630
224 CONTINUE                                        CBL02640
JJ=JJ+1                                             CBL02650
IF(JJ.LT.NCOL) G7 TO 225                            CBL02660
DET=(1.,0.)                                         CBL02670
DO 242 J=1,NCOL                                    CBL02680
242 DET=DET*AMAT(J,J)                               CBL02690
229 GO TO MDT, (1281,244,256)                       CBL02700
ENTRY MAPDY (CMAT ,C4 ,ROOTS ,K4A ,MCOL ,NP,       CBL02710
1, IN, N, KOUNT, AMAT, BMAT, MAT, C5, D ,MCO)     CBL02720
NCOL=MCO                                           CBL02730
GO TO 10                                             CBL02740
END                                                  CBL02750
SUBROUTINE JUGGLE(ROOTS,RT,KOUNT,K4A)              CBL02760
DOUBLE PRECISION ROOTS(1)                          CBL02770

```

COMPLEX RT(1)	CBL02780
REAL*8 CR	CBL02790
DATA C0/27FFFFFFFFFFFFFFFF/	CBL02800
DIMENSION KOUNT(1)	CBL02810
K=1	CBL02820
I=1	CBL02830
1 IF(KOUNT(1).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.C0)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.K+4)RETURN	CBL02950
GO TO 1	CBL02960
END	CBL02970
SUBROUTINE POLYRT(AC,ROOT,KOUNT,MM,0,A,T)	CBL02980
DIMENSION KOUNT(3)	CBL02990
DOUBLE PRECISION AC(5),ROOT(5),Q(5),T(3),A(5)	CBL03000
DOUBLE PRECISION Q1(1),Q2(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 (A0(M+1)) 100,95,100	CBL03050
95 ROOT(M) = 0.00	CBL03060
KOUNT((M+1)/2) = 0	CBL03070
M = M - 1	CBL03080
GO TO 90	CBL03090
100 TOL = 1.05	CBL03100
IF (M - 1) 450,103,106	CBL03110
103 ROOT(1) = -A0(2) /A0(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) = A0(I)	CBL03200
IF(A(N-1))115,112,115	CBL03210
112 B1(1)= 1.0-5	CBL03220
B2(1)= 1.0-8	CBL03230
GO TO 120	CBL03240
115 B2(1)=-A(N+1)/A(N-1)	CBL03250
B1(1)= -Q2(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)	C9L03330
GO TO 310	C9L03340
130 CALL GFOWL(T(N-2),Q(N))	C9L03350
ITERB = 0	C9L03360
KEY = 30	C9L03370
INK = 15	C9L03380
MURDER=20	C9L03390
LOVE = 4	C9L03400
220 ITERB = ITERB + 1	C9L03410
230 Q(1) = A(1)	C9L03420
Q(2) = A(2) + B1(1)* Q(1)	C9L03430
DO 240 J = 3,N1	C9L03440
240 Q(J) = A(J) + B1(1)* Q(J-1) + B2(1)* Q(J-2)	C9L03450
T(1) = Q(1)	C9L03460
T(2) = Q(2) + B1(1)* T(1)	C9L03470
DO 250 J = 5,N1	C9L03480
250 T(J-2) = Q(J-2) + B1(1)* T(J-3) + B2(1)* T(J-4)	C9L03490
X = B1(1)* T(N-1) + B2(1)* T(N-2)	C9L03500
CALL RUFF(T,Q)	C9L03510
B1(1) = B1(1) + D1	C9L03520
B2(1) = B2(1) + D2	C9L03530
IF (KODE) 260,260,280	C9L03540
260 IF (TOL* DABS(D1) - DABS(B1(1))) 261,261,270	C9L03550
261 IF (TOL* DABS(D2) - DABS(B2(1))) 262,262,270	C9L03560
262 IF (KODE) 265,265,460	C9L03570
263 KODE = 1	C9L03580
264 S1 = DABS(D1)	C9L03590
S2 = DABS(D2)	C9L03600
GO TO 220	C9L03610
265 LOVE = LOVE - 1	C9L03620
IF (LOVE) 220,290,220	C9L03630
270 IF (ITERB - KEY) 220,271,271	C9L03640
271 MURDER = MURDER - 1	C9L03650
IF (MURDER) 409,285,272	C9L03660
272 KEY = KEY + INK	C9L03670
B2(1) = -B2(1) - .500*(B1(1)**2)	C9L03680
GO TO 220	C9L03690
280 IF (.400* DABS(D1) - S1) 281,410,410	C9L03700
281 IF (.400* DABS(D2) - S2) 264,410,410	C9L03710
285 ITERB = 999	C9L03720
290 K = K + 1	C9L03730
KOUNT(K) = ITERB * 10	C9L03740
A(N) = B1(1)	C9L03750
A(N1) = B2(1)	C9L03760
N = N - 2	C9L03770
N1 = N1 - 2	C9L03780
DO 300 I = 1,N1	C9L03790
300 A(I) = Q(I)	C9L03800
IF(DABS(B1(1)).LT..100*DSOFT(DABS(B2(1))))	C9L03810
B1(1) = .100*DSOFT(DABS(B2(1)))	C9L03820
GO TO 120	C9L03830
310 DO 320 I = 1,M	C9L03840
X = A(I+1)	C9L03850
A(I+1) = A(I+1)	C9L03860
320 A(I+1) = X	C9L03870

MURDER = -1	CBL03880
N = M	CBL03890
N1 = N + 1	CBL03900
L = N	CBL03910
K = 0	CBL03920
CALL GROWL(T(N-2),O(N))	CBL03930
330 IF (L - 1) 440,340,400	CBL03940
340 ITER8 = 0	CBL03950
Q(1) = A(1)	CBL03960
B1(1) = A(2)	CBL03970
350 ITER8 = ITER8 + 1	CBL03980
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)	CBL04030
D1 = Q(N1) / T(N)	CBL04040
B1(1) = B1(1) + D1	CBL04050
IF (DABS(B1(1)) - TOL* DABS(D1)) 380,390,390	CBL04060
380 IF (ITER8 - 8) 350,385,350	CBL04070
385 ITER8 = 9	CBL04080
390 KOUNT(K+1) = ITER8	CBL04090
A2(2) = B1(1)	CBL04100
GO TO 440	CBL04110
400 K = K + 1	CBL04120
KODE = 0	CBL04130
B1(1) = A0(L)	CBL04140
B2(1) = A0(L+1)	CBL04150
ITER8 = KOUNT(K)	CBL04160
KEY = ITER8 + 8	CBL04170
IF (M - 2) 220,409,220	CBL04180
409 ITER8 = ITER8 + 1	CBL04190
410 X = B1(1)**2 + 4.00* B2(1)	CBL04200
IF (X) 420,430,430	CBL04210
420 A2(L) = .500* DSQRT(-X)	CBL04220
A2(L+1) = .500* B1(1)	CBL04230
KOUNT(K) = -ITER8	CBL04240
L = L - 2	CBL04250
GO TO 330	CBL04260
430 X = DSQRT(X)	CBL04270
IF (B1(1)) 432,431,431	CBL04280
432 X = -X	CBL04290
431 A2(L) = .500* (B1(1) + X)	CBL04300
A2(L+1) = -B2(1) / A2(L)	CBL04310
433 KOUNT(K) = ITER8	CBL04320
L = L - 2	CBL04330
GO TO 330	CBL04340
440 J = N1	CBL04350
DO 450 I = 1,N	CBL04360
ROOT(I) = A0(J)	CBL04370
A0(J) = A(J)	CBL04380
450 J = J - 1	CBL04390
460 RETURN	CBL04400
END	CBL04410
SUBROUTINE GROWL(A,Y)	CBL04420

DOUBLE PRECISION A(2),B,X(2),Y(2),T	CBL04430
COMMON /BARK/ X,B,NIX	CBL04440
RETURN	CBL04450
ENTRY RUFF	CBL04460
NIX = 0	CBL04470
IF (ABS(SNGL(B)) - ABS(SNGL(A(2)))) 100,120,110	CBL04490
100 T = B / A(2)	CBL04490
X(2) = (T*Y(1) - Y(2)) / (A(2) - T*A(1))	CBL04500
X(1) = -(A(1)*X(2) + Y(1)) / A(2)	CBL04510
RETURN	CBL04520
110 T = A(2) / B	CBL04530
X(2) = (T*Y(2) - Y(1)) / (A(1) - T*A(2))	CBL04540
X(1) = -(A(2)*X(2) + Y(2)) / B	CBL04550
RETURN	CBL04560
120 IF (SNGL(B)) 110,130,110	CBL04570
130 NIX = 1	CBL04580
RETURN	CBL04590
END	CBL04600
	CBL04610
	CBL04620

Appendix D

Sample Input

SAMPLE INPUT FOR ACTIVE CABLE PPOG-BASIC LONG CHAP. W COMP PET OUT

1	-1	0	0	1	2	0	3	0	0	0	1						
J.		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.							
8.72		.000805		152.0		9.16		1.4		11.5							
-.8		3.60		21.4		22.95											
J.		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									
8.		-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 CHAP OF THETA/ENO 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=C. W LIFT CABLE -CHAP. ROOTS OPTION

1	-1	0	0	0	2	0	9	3	0	1	0	1	0	8	0	0
48	C.															
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	445.													

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

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)= 0.917E-01	AERO(27)= -0.107	AERO(28)= 0.192	AERO(29)= 0.910E-02	AERO(30)= -0.682E-01
AERO(31)= 0.238	AERO(32)= -0.116	AERO(33)= -0.690E-01	AERO(34)= 0.500E-03	AERO(35)= -0.500E-03

CABLE CONFIGURATION ON MODEL

FRONT CABLE IS HORIZNTL AND REAR CABLE IS VERTICAL

CABLE GEOMETRY-CABLE NO. 1 CABLE LENGTH= 0.123377E 03 IN

DIP. COS.=DEG ARM-IN
 0.492127E 02 0.278000E 02
 0.417822E 02 0.400000E 01
 0.800904E 02 0.0

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

-0.482127E 02 0.27000E 02
0.179218E 03 -0.40000E 01
0.89304E 02 0.0

CABLE GEOMETRY-CABLE NO. 3 CABLE LENGTH= 0.123383E 03 IN
DIP. 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
DIP. 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 = -0.3954200E-03
ACCX = -0.9796825E-03
THEDOT=-0.24915906E-03 RAD/SEC

EM. ATT., DELTA, & CABLE TENSION

THETA = 1.03 DEG
DELTA = -1.37 DEG
FRY CAR. TENSION= 0.127591E 03 LBS
FR CAR. TENSION = 0.100214E 03 LBS

AERO DATA IN BODY AXIS AT EQUAT. REF. CENTER

AEROP(1)=-0.095E-01	AEROP(2)=-0.270	AEROP(3)= 0.264E-01	AEROP(4)= 0.290	AEROP(5)= -5.85
AEROP(6)=-1.49	AEROP(7)= 0.183	AEROP(8)= -10.2	AEROP(9)= -13.8	AEROP(10)=-0.515E-01
AEROP(11)=-0.815E-01	AEROP(12)= 0.304E-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)= 0.778E-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(37)=			

++++ LONGITUDINAL STABILITY ++++

POSITION AND COEFFICIENTS OF EACH POLYNOMIAL OF MATRIX

1 1	1.894176D 01	-5.993159D-01		
2 1	3.705937D 01	1.207301D 01	4.719999D 00	
3 1	-5.258563D 01	4.277408D 00	-1.538085D 00	
4 1	2.409516D-02			
1 2	-1.014281D 02	-5.94383D-04		
2 2	5.288530D 03	3.301300D-02	-1.573332D 00	
3 2	2.510702D 03	4.357657D 01	2.192442D 01	
4 2	-5.582039D-02			
1 3	-1.332214D 00			
2 3	-3.209992D-02			
3 3	7.436472D-02			
1 4	7.467446D 02	2.033448D-01	4.719999D 00	
2 4	1.953979D 01	5.576770D-01		
3 4	1.163130D 01	-7.701370D-02		
4 4	1.000000D 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 -5.2237675E 05 0.0
 DETERMINANT -5.2237638E 05 0.0

FINAL MATRIX

1	1	-1.894176D 01							
1	2	1.014281D 02	1.389008D 00						
2	2	-7.562133D 03							
1	3	1.332214D 00							
2	3	-7.160784D 00	2.208023D -01						
3	3	3.295325D 00							
1	4	-7.467446D 02	-2.507622D 01	-4.719999D 00					
2	4	4.057771D 03	8.592576D 02	4.183917D 02	-7.822966D -01				
3	4	-1.885203D 03	-9.879158D 01	-1.247060D 02	-2.545717D 00	-5.792186D -01			
4	4	-1.019207D 00	-2.607884D -02	-5.975908D -02	-1.347630D -03	-2.855462D -04			

REAL	IMAGINARY	ERROR		
-8.394477E-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 -4.910985E 05 -1.230982E 04 -2.820763E 04 -6.361116E 02 -1.347842E 02

POLYNOMIAL W CONST TERM FIRST

REAL	IMAGINARY	T H/D-SEC	1/T H/D	PERIOD-SEC	ONATF-CPS	UNDNAT-CPS	DAMP RATIO	DECAY RATIO
-0.9199E-02	+0.4329E 01	0.8253E 02	0.1212E-01	0.1451E 01	0.6890E 01	0.6890E 00	0.1941E-02	0.9879E 00
-0.1351E 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

138 100.00

140 100.00

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 THE NUMERATOR ROOTS

POLYNOMIAL W CONST TERM FIRST

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

REAL	IMAGINARY	T H/D-SEC	1/T H/D	PERIOD-SEC	DNATF-CPS	UNDNAT-CPS	DAMP RATIO	DECAY RATIO
-0.3984E 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
-0.1506E 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 THET/ CMD TRANSFER FUNCTION
 STEADY STATE GAIN = 0.1607E-03

FREQ(RPS)	AMP RAT(DB)	PHASE(DEG)	AMP. VALUE	FREQ(RPS)	AMP RT(DB)	PHASE(DEG)	AMP. VALUE
0.1000E 00	-0.7538E 02	-0.2813E 00	0.1608E-03	0.5000E 01	-0.7930E 02	0.6857E 01	0.1084E-03
0.1200E 00	-0.7538E 02	-0.3376E 00	0.1608E-03	0.5500E 01	-0.7775E 02	0.4034E 01	0.1295E-03
0.1500E 00	-0.7538E 02	-0.4222E 00	0.1608E-03	0.6000E 01	-0.7693E 02	0.1681E 01	0.1423E-03
0.1700E 00	-0.7538E 02	-0.4786E 00	0.1608E-03	0.6500E 01	-0.7636E 02	-0.1988E 00	0.1521E-03
0.2000E 00	-0.7537E 02	-0.5633E 00	0.1608E-03	0.7000E 01	-0.7587E 02	-0.1800E 01	0.1608E-03
0.2500E 00	-0.7537E 02	-0.7049E 00	0.1608E-03	0.7500E 01	-0.7543E 02	-0.3251E 01	0.1693E-03
0.3000E 00	-0.7537E 02	-0.8468E 00	0.1609E-03	0.8000E 01	-0.7498E 02	-0.4634E 01	0.1781E-03
0.3500E 00	-0.7537E 02	-0.9895E 00	0.1609E-03	0.8500E 01	-0.7453E 02	-0.6005E 01	0.1876E-03
0.4000E 00	-0.7536E 02	-0.1133E 01	0.1610E-03	0.9000E 01	-0.7406E 02	-0.7413E 01	0.1982E-03
0.4500E 00	-0.7536E 02	-0.1277E 01	0.1611E-03	0.9500E 01	-0.7355E 02	-0.8900E 01	0.2101E-03
0.5000E 00	-0.7535E 02	-0.1421E 01	0.1612E-03	0.1000E 02	-0.7300E 02	-0.1051E 02	0.2238E-03
0.5500E 00	-0.7535E 02	-0.1567E 01	0.1612E-03	0.1200E 02	-0.7022E 02	-0.1953E 02	0.3085E-03
0.6000E 00	-0.7534E 02	-0.1714E 01	0.1613E-03	0.1500E 02	-0.6376E 02	-0.6413E 02	0.6487E-03
0.6500E 00	-0.7534E 02	-0.1862E 01	0.1615E-03	0.1700E 02	-0.6376E 02	-0.1278E 03	0.5154E-03
0.7000E 00	-0.7533E 02	-0.2012E 01	0.1616E-03	0.2000E 02	-0.7345E 02	-0.1592E 03	0.2125E-03
0.7500E 00	-0.7533E 02	-0.2167E 01	0.1617E-03	0.2500E 02	-0.8084E 02	-0.1713E 03	0.9078E-04
0.8000E 00	-0.7532E 02	-0.2315E 01	0.1618E-03	0.3000E 02	-0.8551E 02	-0.1760E 03	0.5304E-04
0.8500E 00	-0.7531E 02	-0.2470E 01	0.1620E-03	0.3500E 02	-0.8900E 02	-0.1789E 03	0.3550E-04
0.9000E 00	-0.7530E 02	-0.2625E 01	0.1621E-03	0.4000E 02	-0.9181E 02	-0.1810E 03	0.2566E-04
0.9500E 00	-0.7529E 02	-0.2783E 01	0.1623E-03	0.4500E 02	-0.9419E 02	-0.1829E 03	0.1951E-04
0.1000E 01	-0.7529E 02	-0.2944E 01	0.1625E-03	0.5000E 02	-0.9626E 02	-0.1843E 03	0.1538E-04
0.1200E 01	-0.7528E 02	-0.3109E 01	0.1632E-03	0.5500E 02	-0.9810E 02	-0.1856E 03	0.1245E-04
0.1500E 01	-0.7528E 02	-0.4702E 01	0.1648E-03	0.6000E 02	-0.9975E 02	-0.1869E 03	0.1029E-04
0.1700E 01	-0.7529E 02	-0.5518E 01	0.1661E-03	0.6500E 02	-0.1013E 03	-0.1881E 03	0.8648E-05
0.2000E 01	-0.7527E 02	-0.6932E 01	0.1685E-03	0.7000E 02	-0.1026E 03	-0.1892E 03	0.7372E-05
0.2500E 01	-0.7518E 02	-0.1014E 02	0.1741E-03	0.7500E 02	-0.1039E 03	-0.1903E 03	0.6358E-05
0.3000E 01	-0.7476E 02	-0.1571E 02	0.1828E-03	0.8000E 02	-0.1051E 03	-0.1911E 03	0.5538E-05
0.3500E 01	-0.7431E 02	-0.2865E 02	0.1925E-03	0.8500E 02	-0.1063E 03	-0.1923E 03	0.4865E-05
0.4000E 01	-0.7760E 02	-0.6377E 02	0.1318E-03	0.9000E 02	-0.1073E 03	-0.1933E 03	0.4306E-05
0.4500E 01	-0.8415E 02	0.1521E 00	0.6202E-04	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 -CHAP. ROOTS OPTION

FRONT CABLE HORIZONTAL, REAR CABLE VERTICAL

NO SNURRER3

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 2.0

49 1.0

EM. ATT. DEFLTN. & CABLE TENSION

THETA = -0.00 DEG

DELTA = 0.0 DEG

FR. CAR. TENSION = 0.906780E 02 LBS

RR CAR. 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
-0.3192E 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 3.06500
446.00

EM. ATT., DEFLTN, & CABLE TENSION

THETA = 1.03 DEG

DELTA = -1.33 DEG

FRONT 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

REAL	IMAGINARY	T H/D-SEC	1/T H/D	PERIOD-SEC	DNATF-CPS	UNDNAT-CPS	DAMP RATIO	DECAY RATIO
0.0	0.0	0.315421E 07	-0.999774E 06	-0.582850E 04	0.156021E 04			
0.0	0.0	0.1000E 06	0.0					
0.3146E 01	0.0	0.2203E 00	0.4539E 01					
-0.2402E 02	0.0	0.2771E-01	0.3609E 02					
0.2559E 02	0.0	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-01	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				
REAL	IMAGINARY	T H/D-SEC	1/T H/D	PERIOD-SEC	DNATF-CPS	UNDNAT-CPS	DAMP RATIO	DECAY RATIO
0.0	0.0	0.1000E 06	0.0					
-0.2473E-02	0.0	0.2593E 03	0.3857E-02					
-0.3373E 01	0.0	0.1745E 00	0.5732E 01					
-0.4173E 00	-0.7813E 01	0.8477E 00	0.1180E 01	0.8016E 00	0.1247E 01	0.1254E 01	0.1038E 00	0.6102E 00
0.0	0.0	0.1000E 06	0.0					

CASE NO= 5 SAMPLE OF ACTIVE CABLE SYSTEM-LAT DIP 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

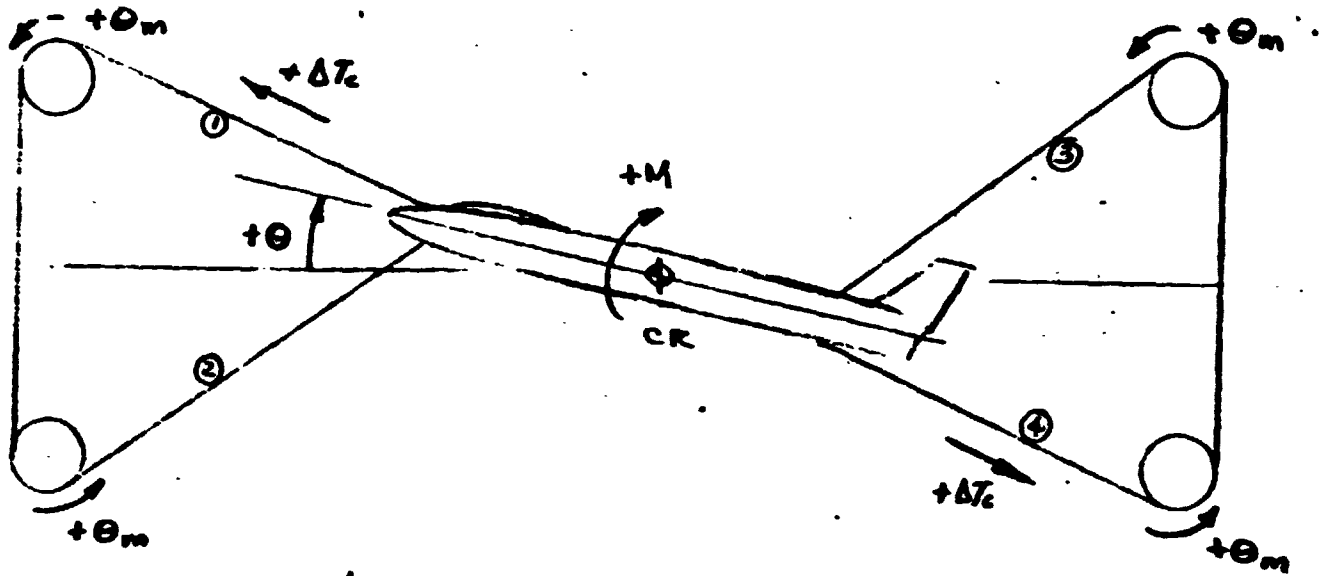
POLYNOMIAL W CONST TERM FIRST

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	0.4847E 00	0.4048E-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

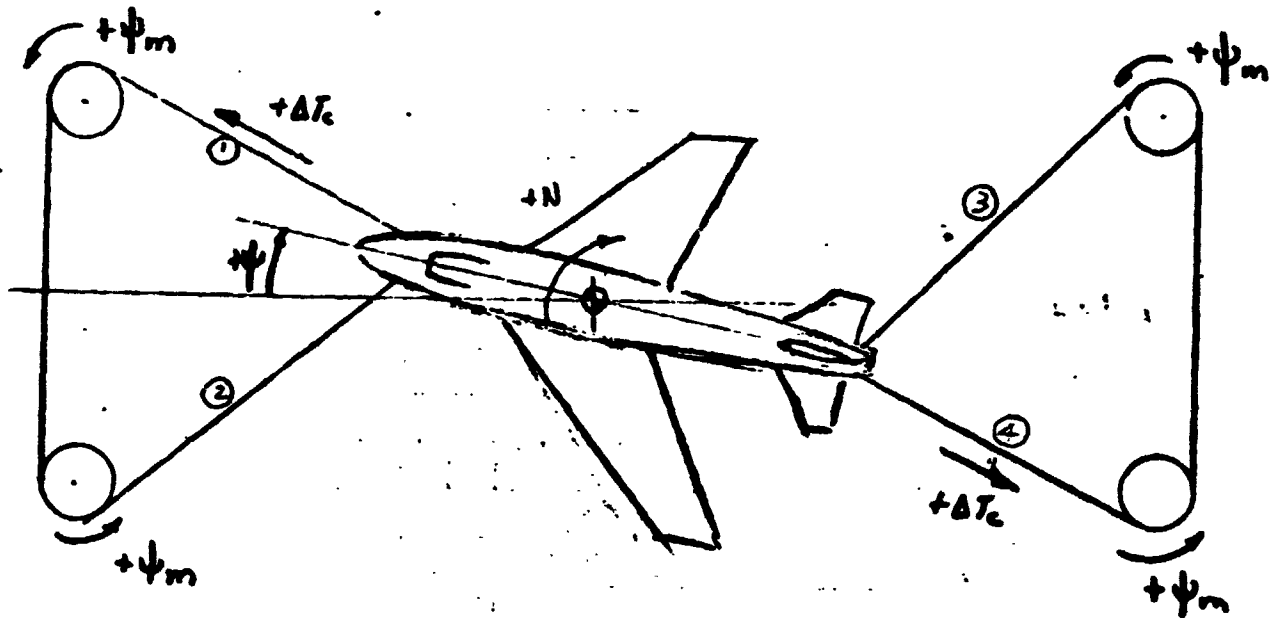
FIGURES

ACTIVE CABLE MOUNT SYSTEM

DEFINITION OF PULLEY MOTION, θ_m, ψ_m



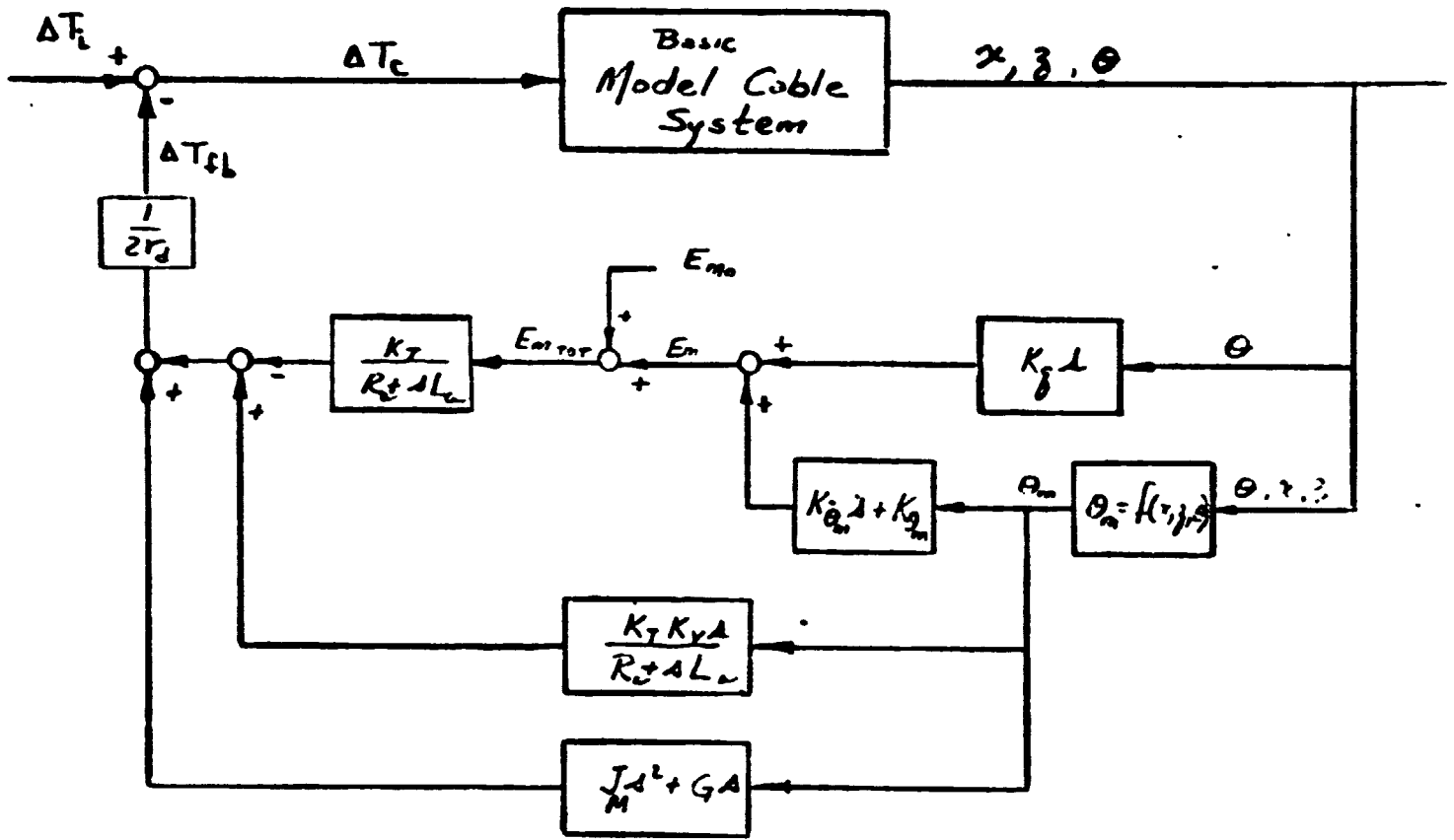
a) LONGITUDINAL CABLE CONTROL



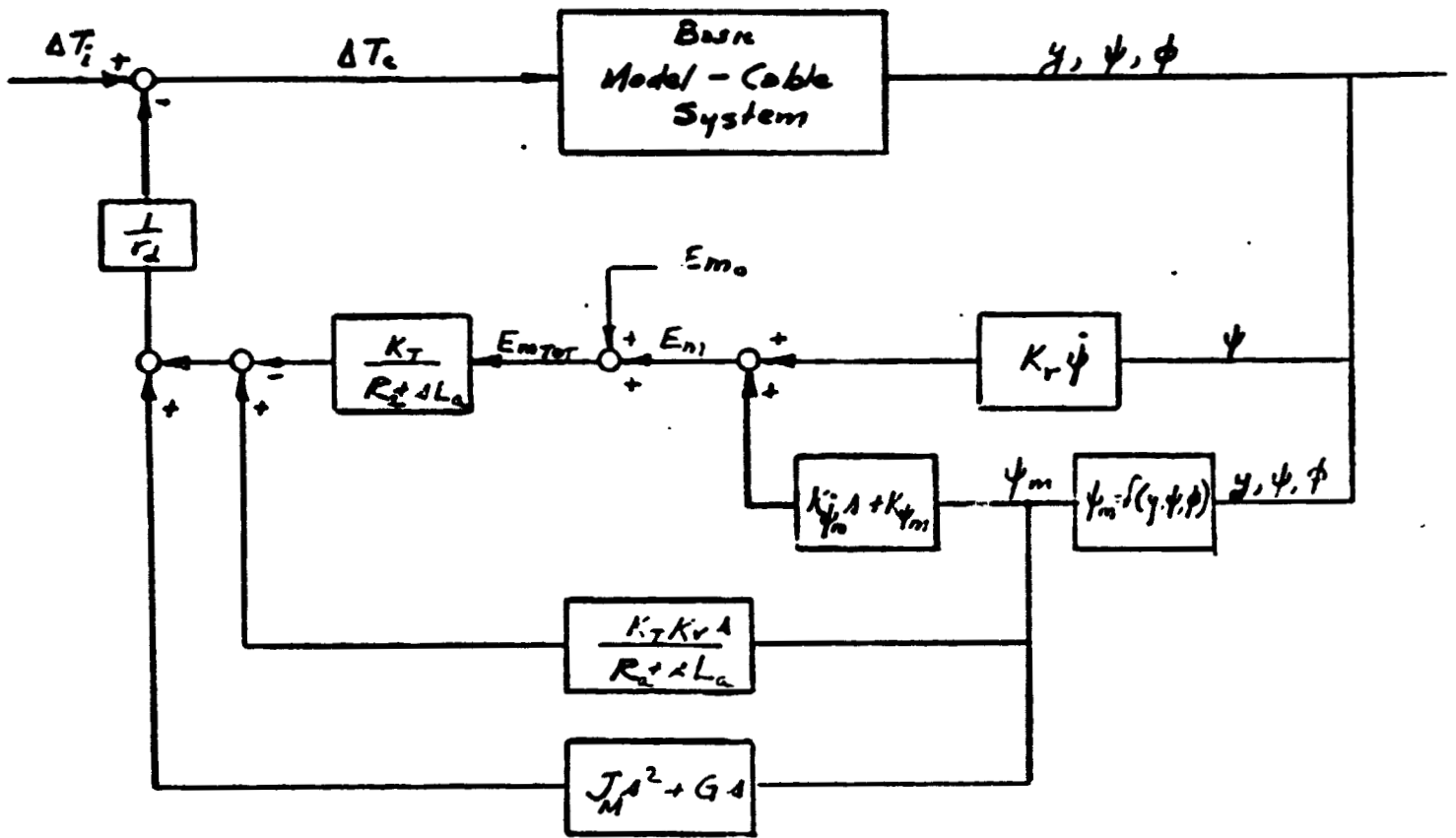
b) DIRECTIONAL CABLE CONTROL

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
\ddot{x}	\ddot{z}	ΔT_c	ν	ΔT_{fb}	θ_m	E_{mTOT}	$\dot{\theta}_m$	E_m	ΔT_c	E_{m0}	ΔT_c
Basic Matrix of Inactive Cable-Model Sys (see ref. 1)					E_1	①					
						②					
						③					
						④					
						⑤					
						⑥					
						⑦					
						⑧					
						⑨					
						⑩					
						⑪					
						⑫					

Eq. 1. $\ddot{m}\ddot{x} - \Sigma F_{x_0} - \frac{\partial F_x}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 2. $\ddot{m}\ddot{z} - \Sigma F_{z_0} - \frac{\partial F_z}{\partial \Delta T_c} \Delta T_c = 0$

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

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

Eq. 5. $\Delta T_{fb} (2r_d) (R_a + sL_a) - (J_M s^2 + G_s) (R_a + L_a s) \theta_m + 2K_T K_V s \theta_m + K_T E_{mTOT} = 0$

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

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

Eq. 8. $\dot{\theta}_m = \theta_{ms}$

Eq. 9. $E_{mTOT} = E_m + W_{m0}$

Eq. 10. $\Delta T_c = \Delta T_1 - \Delta T_{fb}$

ACTIVE CABLE MOUNT SYSTEM

EXTENDED LATERAL-DIRECTIONAL MATRIX

①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫		
y	ψ	φ	ΔT _{fb}	ψ _m	E _{mTOT}	ψ _m	ψ̇	E _m	ΔT _c	E _{m0}	ΔT _i		
Basic Matrix of Inactive Cable-Mod.1 Sys (see ref 1.)					E _g	①							
						②							
									③				
									④				
									⑤				
									⑥				
									⑦				
									⑧				
									⑨				
									⑩				
									⑪				
									⑫				

Eq. 1. $m\ddot{y} - \Sigma Fy_0 - \frac{\partial F}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 2. $I_{zz}\ddot{\psi} - I_{xz}\ddot{\phi} - \Sigma N_0 - \frac{\partial N}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 3. $I_{xx}\ddot{\phi} - I_{xz}\ddot{\psi} - \Sigma T_0 - \frac{\partial T}{\partial \Delta T_c} \Delta T_c = 0$

Eq. 4. $\Delta T_{fb}(2r_d)(R_a + sL_a) - (J_M s^2 + Gs)(R_a + sL_a)\psi_m + 2K_T K_V s \psi_m + 2K_T E_{mTOT} = 0$

Eq. 5. $\psi_m r_d + \left[\frac{\partial \Delta l}{\partial y} y + \frac{\partial \Delta l}{\partial \psi} \psi + \frac{\partial \Delta l}{\partial \phi} \phi \right] = 0$

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. $E_{mTOT} = E_m + E_{m0}$

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

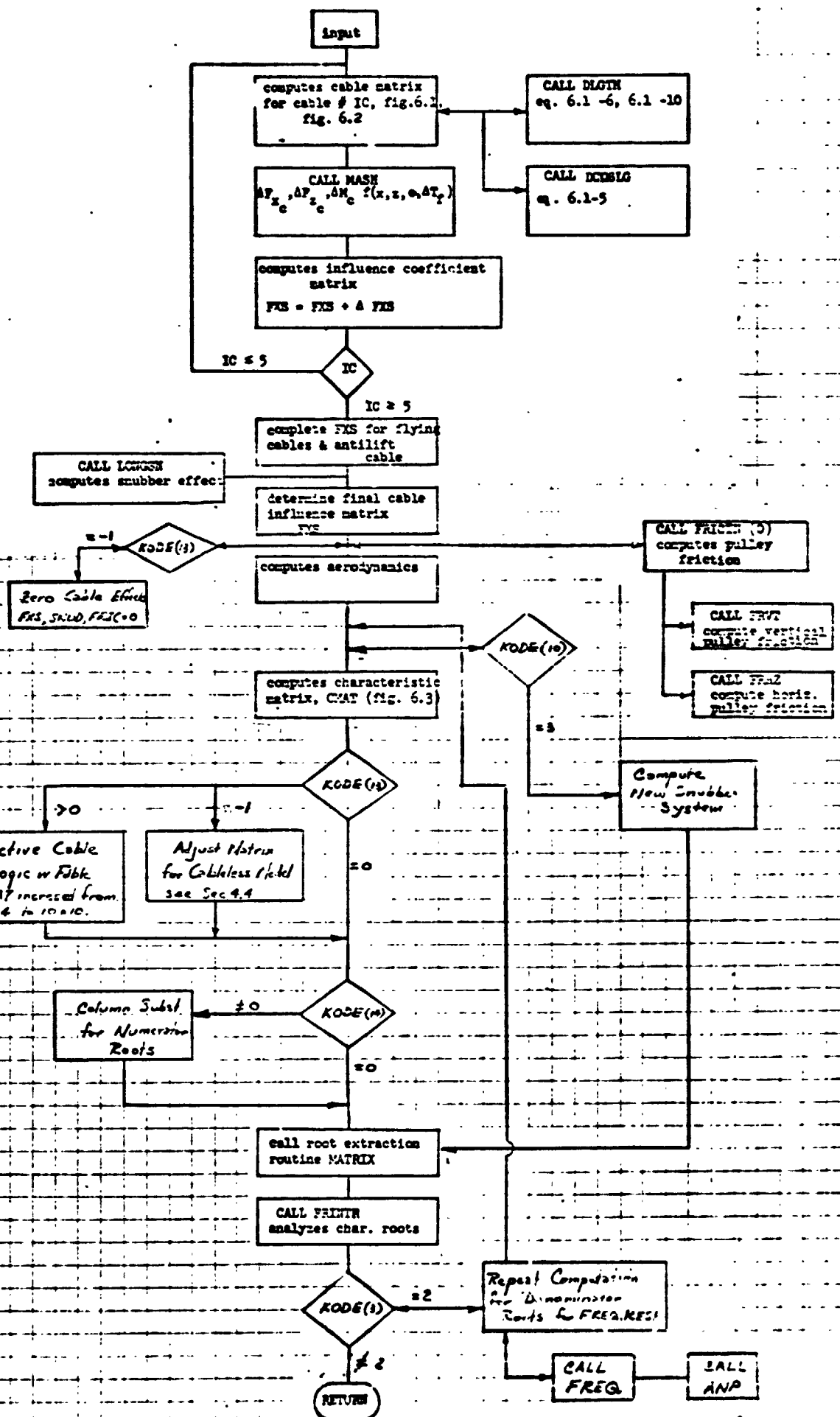


FIGURE 6 SUBROUTINE LONG FLOW CHART

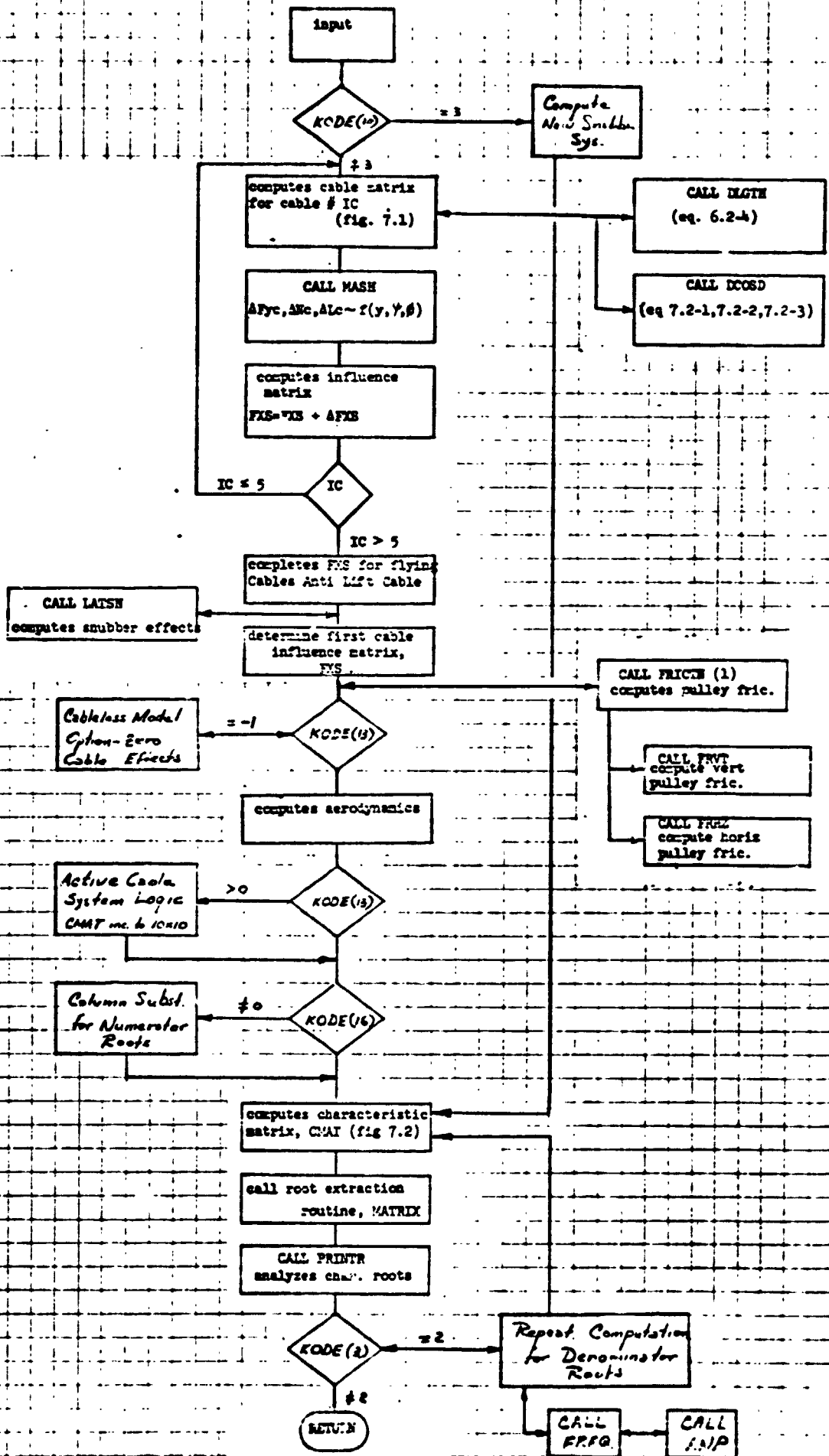


FIGURE 7 SUBROUTINE LAT FLOW CHART

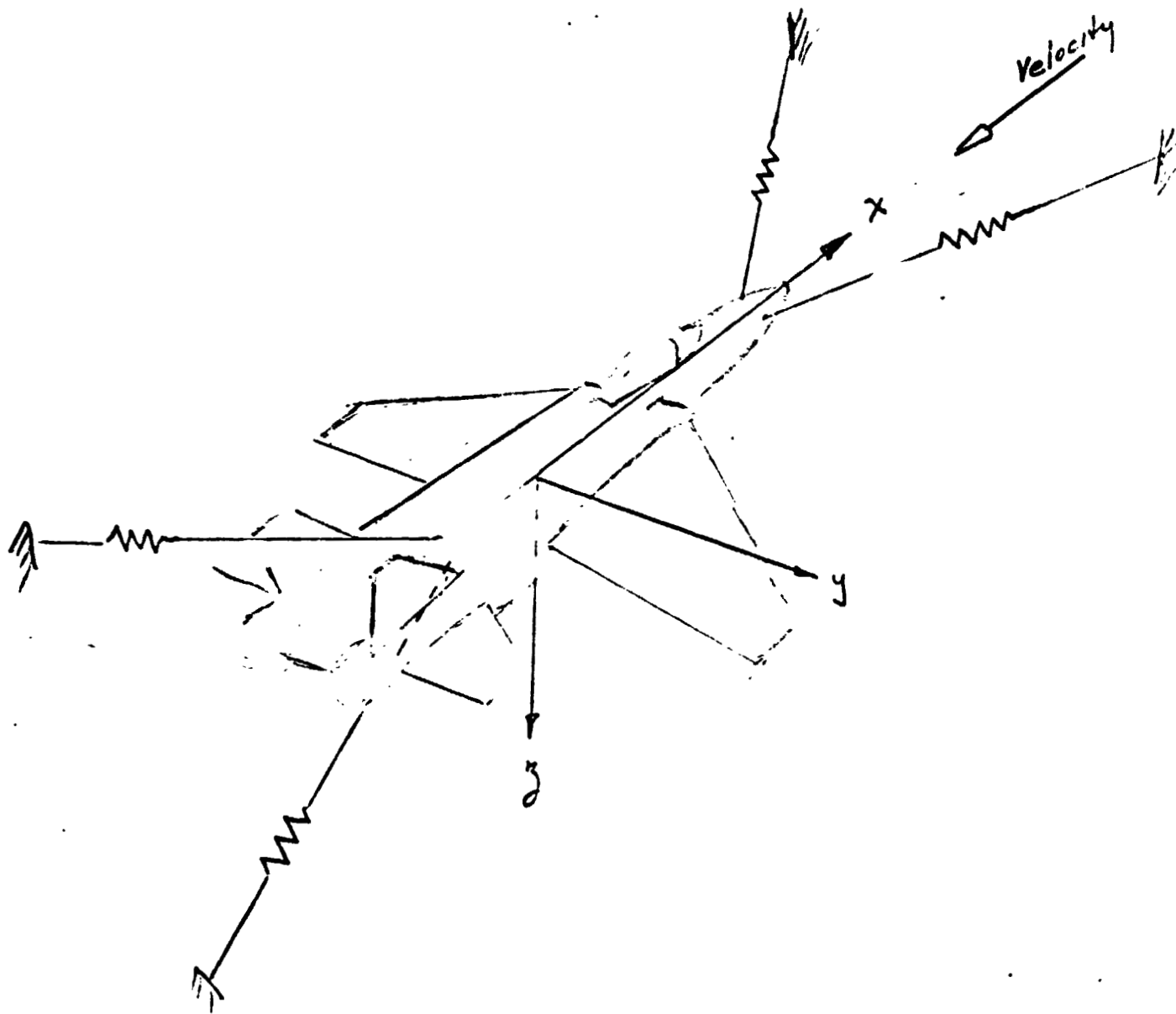


Figure 8. Schematic for Snubbed Model

X	Z	Θ	$\Delta\alpha_x$	$\Delta\alpha_z$	ΔT	Δl
		$-Z T \cos \alpha_z$	$-Z T \sin \alpha_x$		$Z \cos \alpha_x$	
		$Z T \cos \alpha_x$		$-Z T \sin \alpha_z$	$Z \cos \alpha_z$	
		$-Z T l_z \cos \alpha_z$ $-Z T l_x \cos \alpha_x$	$-Z T l_z \sin \alpha_x$	$Z T l_x \sin \alpha_z$	$Z l_z \cos \alpha_x$ $-Z l_x \cos \alpha_z$	
$1/l \sin \alpha_x$		$l_z/l \sin \alpha_x$	-1			
$1/l \sin \alpha_z$		$l_x/l \sin \alpha_z$		-1		$\cot \alpha_z / l$
					-1	K
$-\cos \alpha_x$	$-\cos \alpha_z$	$-l_z \cos \alpha_x$ $-l_x \cos \alpha_z$				-1

FIGURE BA : LONGITUDINAL CABLE INFLUENCE MATRIX

Y	ψ	ϕ	T	$\Delta\alpha_x$	$\Delta\alpha_y$	$\Delta\alpha_z$	Δl
	$-T \cos \alpha_x$	$T \cos \alpha_z$	$\cos \alpha_y$		$-T \sin \alpha_y$		
	$-l_x T \cos \alpha_x$ $-l_y T \cos \alpha_y$	$l_x T \cos \alpha_z$	$l_x \cos \alpha_y$ $-l_y \cos \alpha_x$	$l_y T \sin \alpha_x$	$-l_x T \sin \alpha_y$		
	$+l_x T \cos \alpha_x$	$-l_z T \cos \alpha_z$ $-l_y T \cos \alpha_y$	$l_y \cos \alpha_z$ $-l_z \cos \alpha_y$		$+l_z T \sin \alpha_y$ $-l_y T \sin \alpha_z$		
			-1				K
$\frac{-\cos \alpha_y \cot \alpha_y}{l}$	$l_y \sin \alpha_x +$ $l_x \cos \alpha_y \cot \alpha_x$ $1/l$	$l_z \cos \alpha_y \cot \alpha_z$ $-l_y \cos \alpha_y \cot \alpha_x$ $1/l$		-1			
$\sin \alpha_y / l$	$l_y \cos \alpha_x \cot \alpha_y$ $+l_x \sin \alpha_x / l$	$-l_z \sin \alpha_y +$ $l_y \cos \alpha_z \cot \alpha_y$ $1/l$			-1		
$\frac{-\cos \alpha_y \cot \alpha_z}{l}$	$l_y \cos \alpha_y \cot \alpha_z$ $-l_z \cos \alpha_y \cot \alpha_x$ $1/l$	$l_z \cos \alpha_y \cot \alpha_z$ $+l_y \sin \alpha_z / l$				-1	

FIGURE 8B : LATERAL / DIRECTIONAL CABLE INFLUENCE MATRIX

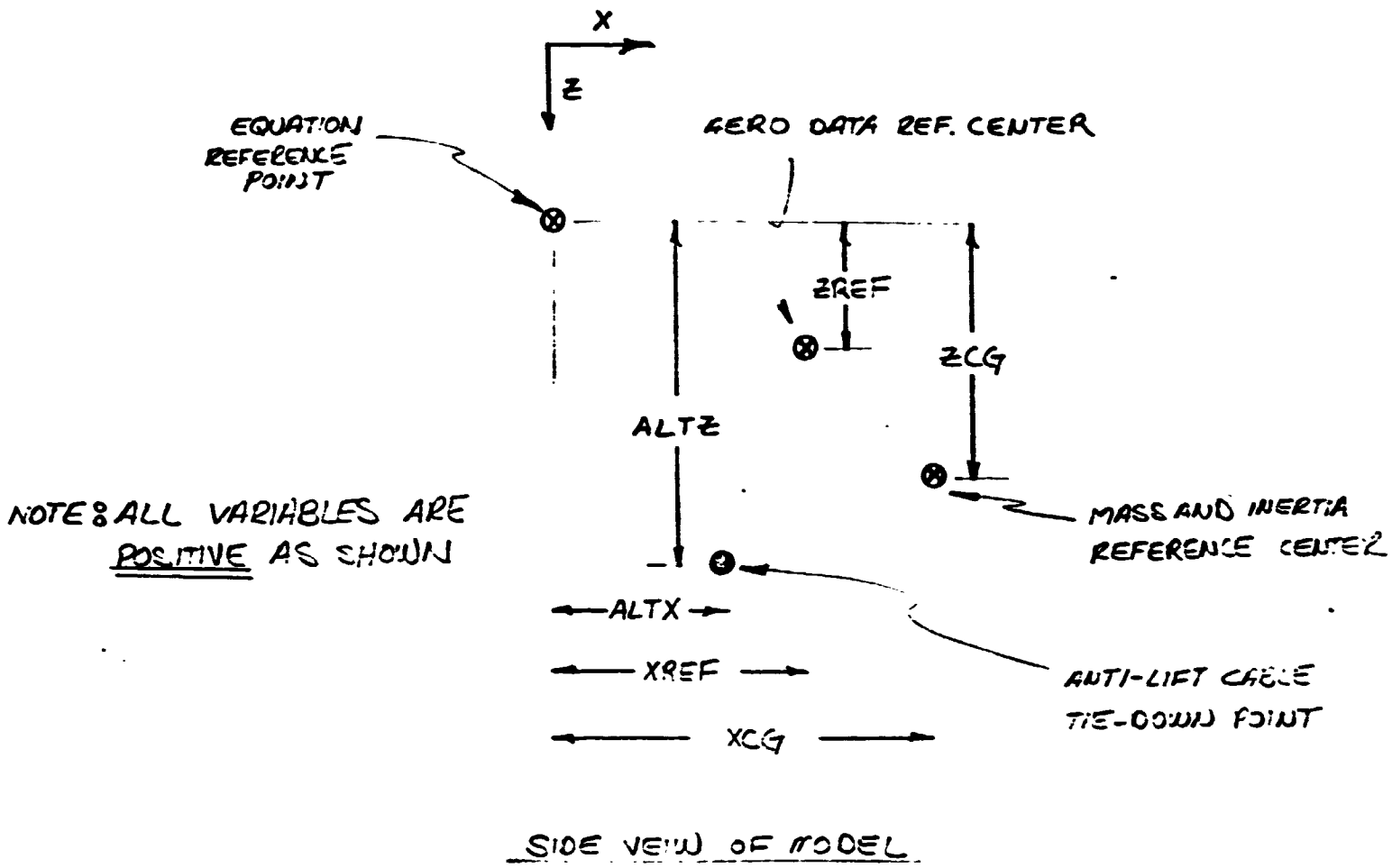
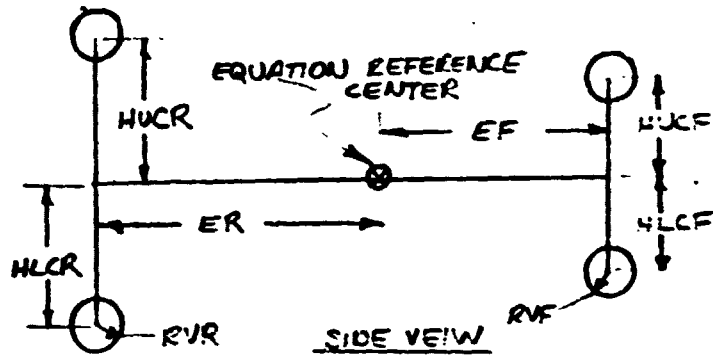
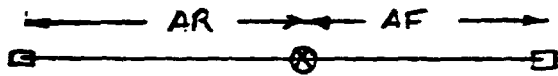


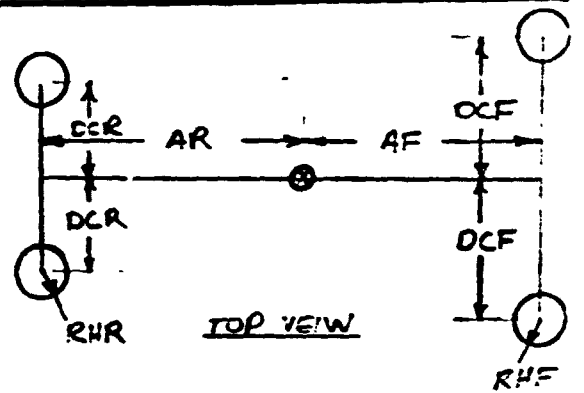
FIG. 9 - REFERENCE CENTER AND LIFT CABLE INPUT DATA



FRONT VERTICAL - REAR VERTICAL

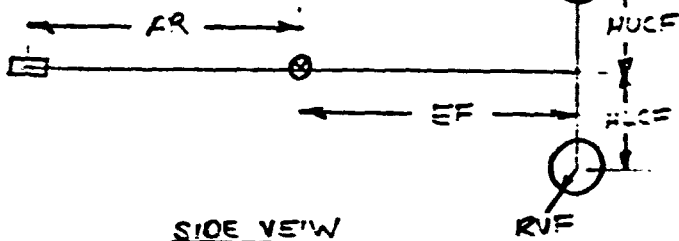


SIDE VIEW

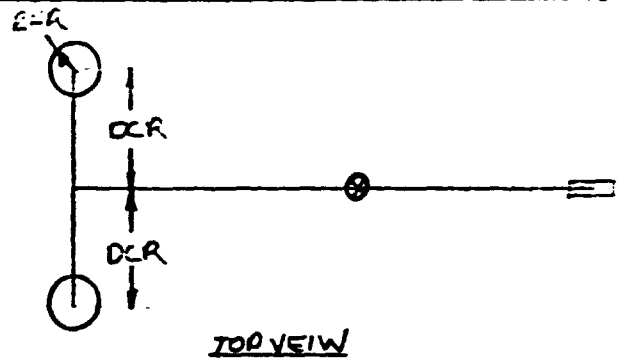


TOP VIEW

FRONT HORIZONTAL - REAR HORIZONTAL

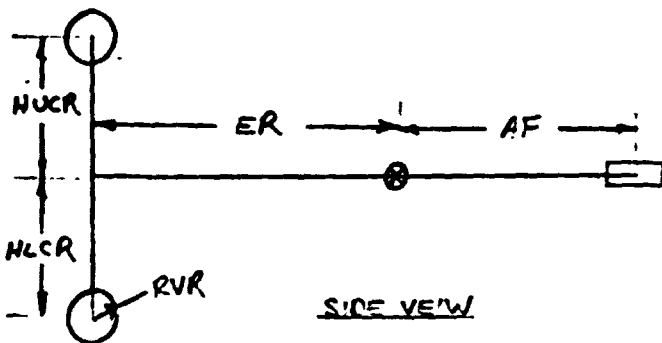


SIDE VIEW

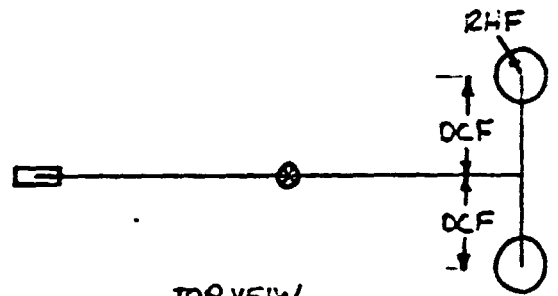


TOP VIEW

FRONT VERTICAL - REAR HORIZONTAL



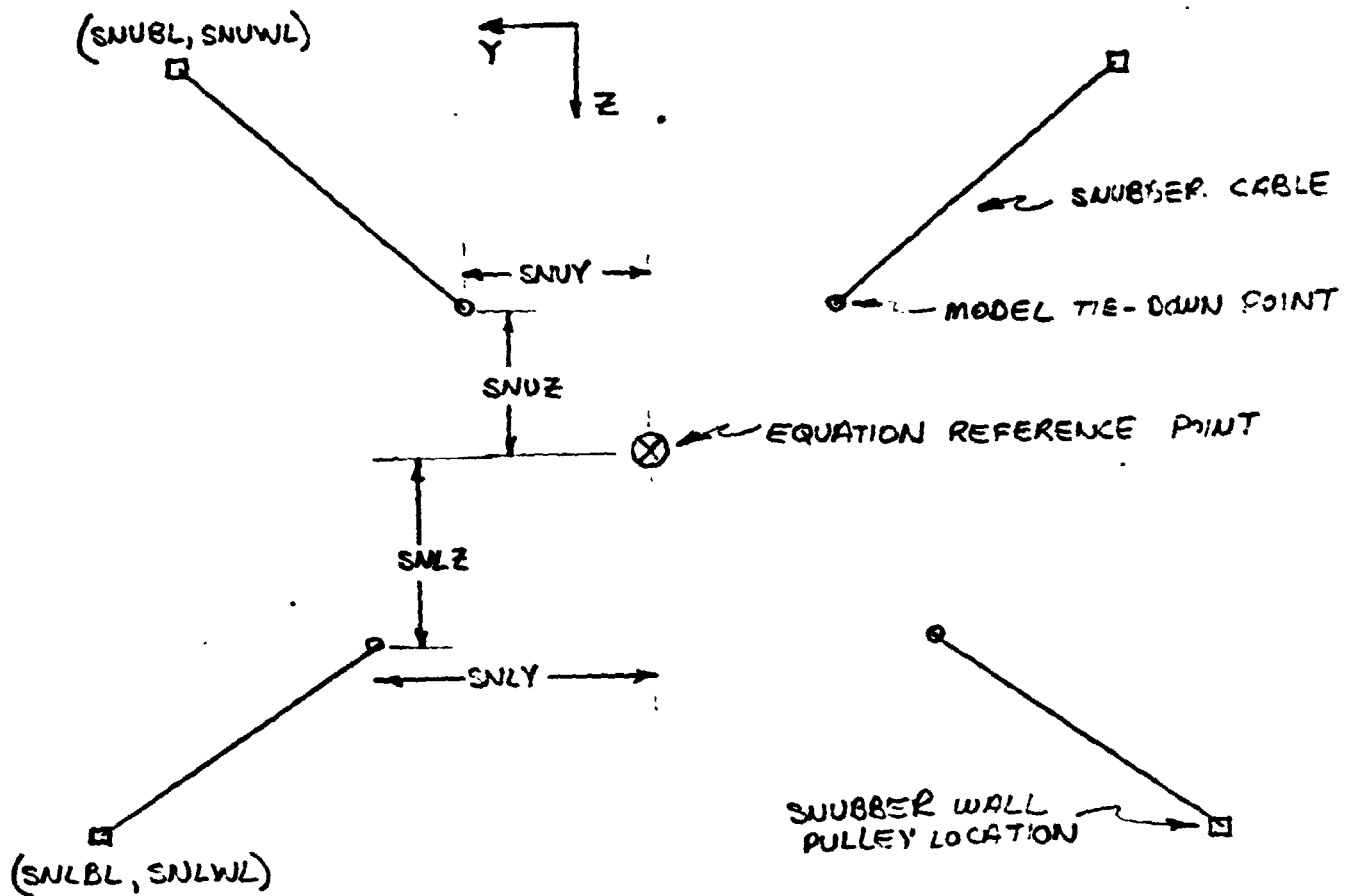
SIDE VIEW



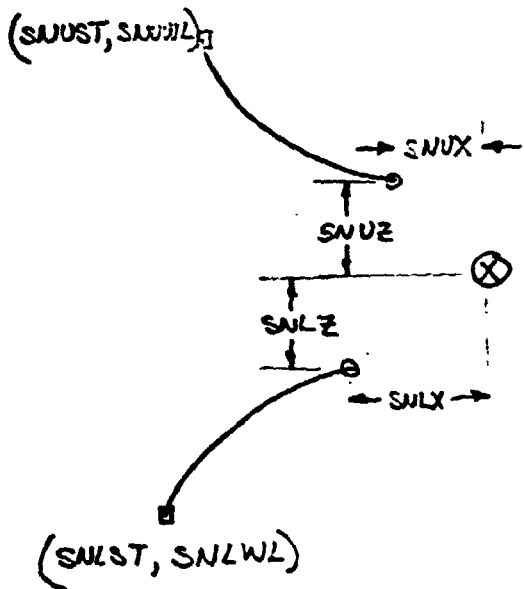
TOP VIEW

FRONT HORIZONTAL - REAR VERTICAL

FIG. 10 - GILLEY GEOMETRY



MODEL - FRONT VIEW



MODEL - SIDE VIEW

NOTE: ALL DISTANCES ARE POSITIVE AS SHOWN

FIG. 11 - SNUBBER CABLE ARRANGEMENT