https://ntrs.nasa.gov/search.jsp?R=19700016277 2020-03-12T03:08:07+00:00Z

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

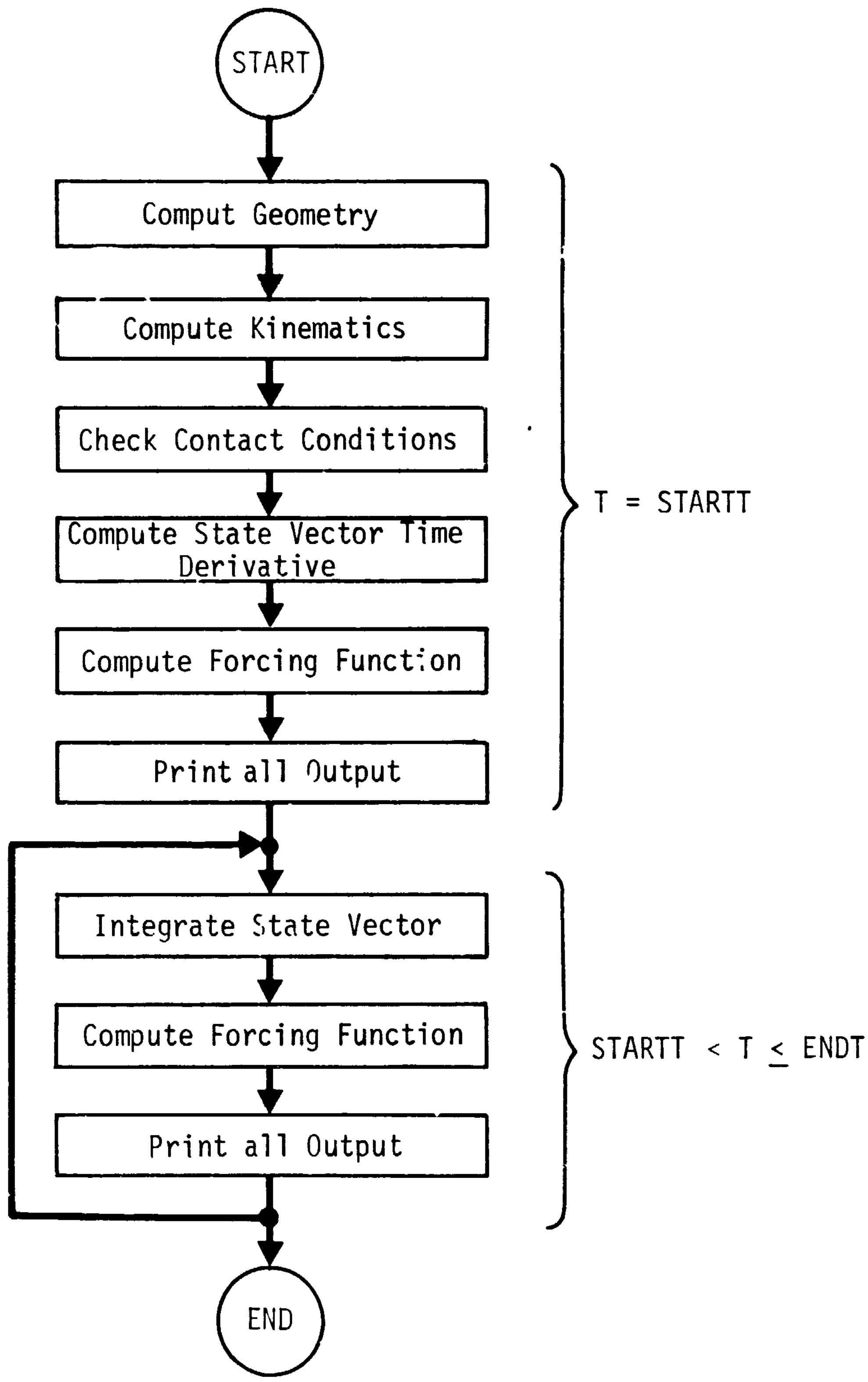


Fig. III-7 PROGRAM TIMHST Logic Flow

4. OVERLAY (3,0) PROGRAM PLOTER

The final overlay (Fig. III-8) is only called if the optional plot indicator so specifies. If it is called it reads plot titles from tape, reads the plot variable time histories from tape, and plots those time histories that have been previously specified. (Fig. III-9).

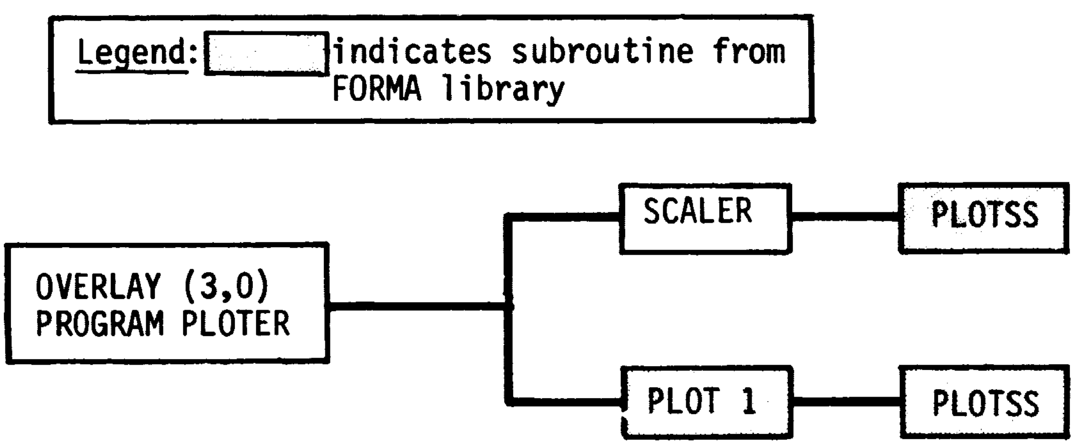


Fig. III-8. PROGRAM PLOTER Calling Structure

C. PROGRAM INPUT DATA

The main program overlay, DOCK00, reads the initial program data. These data consist of three cards that are actually read by Subroutine START. This subroutine is from the FORMA library (Ref III-1). The first card sets a program run number (RUNNO); the second and third cards are used as title cards. The user may here indicate any comments applicable to the particular run.

The next card reads four integer variables with a 16I5 format. These variables are:

MINIC = 0 if intend to read entire state vector (e.g., continuing the simulation from a previous run)
= 1 if desire to read 6x3 array of minimum initial conditions

IFPLOT = 0 if no time history plot output is desired
= 1 if plot output is desired. It will be necessary to insert (in the proper location) as many plot title cards as desired plots.

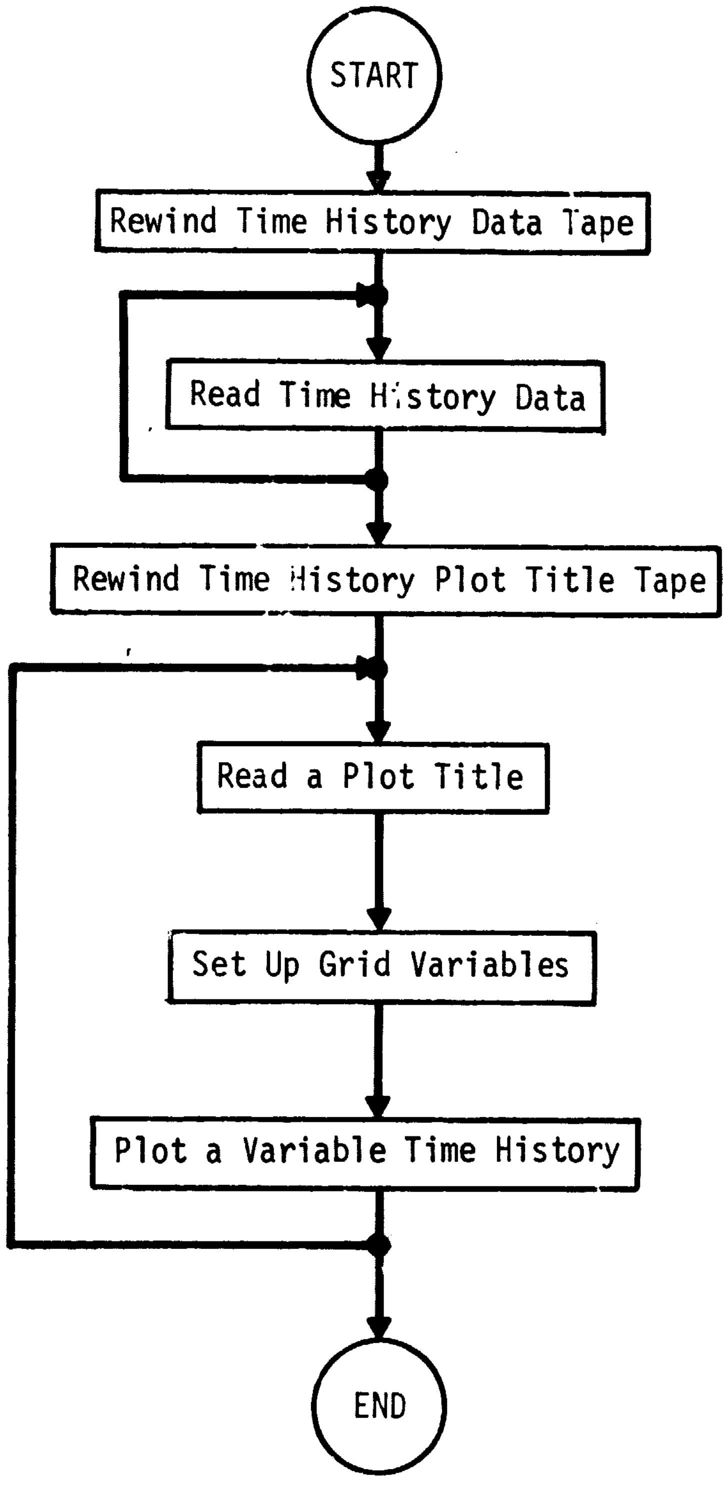


Fig. III-9. PROGRAM PLOTER Logic Flow

IFPER	= 0 if no prespective or stereo pairs plots are desired
	= 1 if perspective plots are desired
	= 2 if stereo pairs plots are desired
IFPNCH	= 0 if do not want to punch state vector at end of simulation
	= 1 if desire to punch state vector at end of simulation (e.g., want to read state vector in as initial conditions for next run)

The next card reads two integer variables with a 16I5 format:

NCNSYS	the number of control system parameters (34 in the present version of the program)
NFORCS	the number of control system forces (10 in the present version)

The next card reads the simulation time variables and certain plot variables with an 8E10.0 format:

STARTT	start time for the simulation
DELTAT	time increment for numerical integration
ENDT	end time for the simulation
XDLTA	indicator for abscissa division on output plots. The total abscissa will be XDLTA x NXPL x 10 wide. The variable NXPL is read individually for each plot on the appropriate title card.
XPRNT	print interval indicator. Program will print every XPRNT interval.

The next four cards read required program input scalars with an 8E10.0 format. Refer to Appendix A for a description of these data.

AA	2.25	probe geometry
AB	0.85	probe geometry
AD	0.708	probe geometry
AE	0.965	probe geometry
AF	3.085	probe geometry
AP	4.25	probe geometry

DLB	14.48	probe geometry
DLD	12.90	probe geometry
DLE	17.59	probe geometry
DLF	4.75	probe geometry
DLI	14.10	probe geometry
DLP	3.30	probe geometry
DLQ	16.53	probe geometry
DLS	2.055	probe geometry
FDO	11.5	probe geometry
PEO	10.5	probe geometry
ALPHAD	42.75	drogue cone half angle
CSPHER	2.58	radius of probe head
RADCON	13.56	radius of drogue cone
AKD1		retract force constant
AKD2		retract force constant
AKD3		retract force constant
RHOSO	2.08	stroke to open secondary orifice
FRETL	1000.0	limiting retract force
EXTL	11.12	probe extended length
TLAG		time lag for initiation of retract mode
TSTAR		capture time-program will compute if capture achieved during the run; if continuing a run where capture occurred during the previous simulation, must read in the correct value of capture time
HDMU		coefficient of friction for probe head on drogue
ARMMU		coefficient of friction for pitch arms on drogue
BINDMU		binding friction coefficient

The next three cards read the initial components of the vector from point T to point D in the T coordinate system (TDTO). These data are read by Subroutine READ (Ref III-1). The initial components of the vector from point C to point P in the C coordinate system (CPCO) are read next. This array is followed by an array that reads the initial angular position (SIGMA) of the three pitch arms relative to the C coordinate system.

Vehicle inertia data are read next. Target vehicle mass and inertia is followed by chase vehicle mass and inertia. Mass is read by a scalar (AMT and AMC); the 3x3 inertia arrays (AIT and AIC) are read by Subroutine READ.

The next data read concerns probe stiffness and damping characteristics. There are eight of these arrays. They are, in order,

FSPROX	probe barrel load stroke curve
FDMROX	probe barrel damping-accounts for barrel friction
FSPALL	moment-rotation curve for probe loading
FDMALY	damping for probe rotation coordinates (y axis)
FSPALU	moment-rotation curve for probe unloading
FDMALZ	damping for probe rotation coordinates (z axis)
FSPATT	attenuator spring force curve-identical for three attenuators
FDMATT	attenuator damping curve-identical for three attentuators

Note that in setting up these arrays, the 1,1 element of each must be set to a floating point value one less than the number of columns in the array. The 2,1 element is a dummy value and is not used in the program.

With the exception of the first three data cards, the information read to this point has been read by PROGRAM SETUP, the first overlay. SETUP now calls Subroutine HSIG in order to read elastic data. HSIG initially reads five variables with a 3I5,2E10.0 format. These variables are:

IFELAS	= 0 no elastic properties to be read = 1 read elastic properties	
NMODET	number of TV modes	not required if IFELAS = 0
NMODEC	number of CV modes	not required if IFELAS = 0
ZETAT	TV modal damping (% critical)	not required if IFELAS = 0
ZETAC	CV modal damping (% critical)	not required if IFELAS = 0

If IFELAS = 1, the subroutine now reads the necessary modal data for the TV and then for the CV. Frequencies are read into a work space as are modal amplitudes. Subroutine REVISE, another FORMA subroutine, operates with three positioning vectors to take only those desired modes and store them in an array. The balance of the modal data is neglected.

Control now returns to SETUP and, depending on the value of MINIC extablished previously, two options are available:

MINIC	= 0	the complete state vector is read by READ. This option provides for a continuous series of runs; the output from one run is used as input to the next.
	1	a 6x3 array of minimum initial conditions are read. These conditions have been previously detailed (Table II-1). These data are actually read by a call to Subroutine MINYS.

If the parameter IFPLOT was previously set to 1 it is now necessary to read a plot title card for each individual plot that is desired as output. This information will be written on a tape and therefore will be available when the plot overlay (PROGRAM PLOTER) is called. The following information is required on each card. The format is (2I5,2X,I1,2X,2(A6,4X),4A10).

NCX	column indicator for abscissa variable. Refer to Subroutine SETPLT for location of individual variable in plot array.
NCY	column indicator for ordinate variable
NXPL	refer to description of XDLTA
XNAME	variable name for abscissa
YNAME	variable name for ordinate
PTITLE	plot title

A more detailed description of these data and the required format will be found under Subroutine PLOT1 (Ref III-1).

This concludes the data read by PROGRAM SETUP. The remainder of the data is read by Subroutines CONSYS and SETPER when they are called for the first time.

Subroutine CONSYS reads all data associated with the several control systems. This was done so that changes in control systems could be made without impacting the entire program. If it is desired to change the control laws it is necessary only to modify CONSYS and read new data. The basic program logic is unaltered although it might be necessary to read a new value for NCNSYS or NFORCS. Three data cards are required. The variables read here, using the 8E10.0 format, are:

FXCSM	CV axial thrust (400 lb in the current version)
AKRX	CMG roll channel gain
AKRY	CMG pitch channel gain
AKRZ	CMG yaw channel gain
AKDX	CMG roll channel gain
AKDY	CMG pitch channel gain
AKDZ	CMG yaw channel gain
OMFX	CMG roll channel filter break frequency
OMFY	CMG pitch channel filter break frequency
OMFZ	CMG yaw channel filter break frequency
ZETFX	CMG roll channel filter damping ratio
ZETFY	CMG pitch channel filter damping ratio
ZETFZ	CMG yaw channel filter damping ratio
AX	CMG roll channel transfer function gain
AY	CMG pitch channel transfer function gain
AZ	CMG yaw channel transfer function gain
BX	CMG roll channel time constant
BY	CMG pitch channel time constant
BZ	CMG yaw channel time constant
TLIM	total CMG torque limiting value

A detailed description of the CMG control system logic and definition of the input variables will be found in Appendix B. If the value of IFELAS is not equal to zero an array of modal slopes at the CMG sensor locations is now required. There must be as many modes here as target vehicle modes, NMODET. The input data for the TACS control system and the RCS control system is read as a 6x12 array CVEC by Subroutine READ. Definition of these data will be found in Appendix B of this volume and in Appendix A of Volume I.

The final data are required only if the variable IFPER is not equal to zero. This information is required for Subroutine SETPER which plots perspective or stereo pairs views of the probe and drogue. A complete description of required data will be found under the description of Subroutine PLOT3 (Ref III-1).

There is no further data required. When the simulation reaches ENDT, control returns to Subroutine START. The run is terminated with a STOP card.

Note that a consistent set of input units is used throughout this program. These units are

time in seconds

length in inches

mass in lb-sec^2/in.

angle in degrees

A comprehensive summary of program input is detailed in Table III-1.

D. PROGRAM OUTPUT

A summary of PROGRAM DOCKEL output is presented in this section. Actual program output generated from the input listed in Appendix D will be found in Appendix E.

The output begins with a print of all data. Data that were read into the program by Subroutine READ is printed (by READ) immediately after they are read. Other data are printed just before the beginning of the numerical integration. The time history and other pertinent program output follows. This information is printed at every XPRNT interval during the course of the simulation. The following information is available at each print time:

SIMULATION TIME	Docking simulation time
Y(I) YDOT(I)	State vector and state vector time derivative
U - TARGET	Six velocity components for TV (See Equation II-143, Vol I)
X - TARGET	Three components of TV position vector in inertial space (see Equation II-135, Vol I)

TABLE III- 1 .   SUMMARY OF INPUT DATA

```
 300 CALL START

1001 FORMAT (16I5)
1002 FORMAT (3I5, 2E10.0)
1003 FORMAT (2I5, 2X, I1, 2X, 2(A6,4X), 4A10)
1004 FORMAT (8E10.0)
2001 FORMAT (8A10)
2002 FORMAT (I5,5X,6F10.0)
2003 FORMAT (8E10.0)

     READ (NIT,1001)  MINIC, IFPLOT, IFPER, IFPNCH

     READ (NIT,1001)  NCNSYS, NFORCS

     READ (NIT,1004)  STARTT, DELTAT, ENDT, XDLTA, XPRNT
     READ (NIT,1004)  AA, AB, AD, AE, AF, AP, DLB, DLD, DLE,
    .                 DLF, DLI, DLP, DLQ, DLS, FDO, PEO,
    .                 ALPHAD, CSPHER, RADCON, AKD1, AKD2, AKD3, RHOSO,
    .                 FRETL, EXTL, TLAG, TSTAR, HDMU, ARMMU, BINOMU

     CALL READ (TDTD   , N1, N2,  1,  3)

     CALL READ (CPCC   , N1, N2,  1,  3)

     CALL READ (SIGMA  , N1, N2,  1,  3)

     READ (NIT,1004) AMT
     CALL READ (AIT    , N1, N2,  3,  3)

     READ (NIT,1004) AMC
     CALL READ (AIC    , N1, N2,  3,  3)

     DO 15 I = 1, 4
     CALL READ (FSPRNG(1,1,I), N1, N2,  2, KTABLE)
     CALL READ (FDAMP (1,1,I), N1, N2,  2, KTABLE)
  15 CONTINUE

     READ (NIT,1002) IFELAS, NMODET, NMODEC, ZETAT, ZETAC
     IF (IFELAS .EQ. 0) GO TO 200
     DO 100 L = 1, 2
     CALL READ (VWORK , N1,NCF,  1,100)
     CALL READ (AWORK , NR, NC,100,100)
     CALL READIM (JVEC  , N1,NC1,  1,100)
     CALL READIM (IVEC  , N1,NC1,  1,100)
     CALL READIM (IVEC  , N1,NC1,  1,100)
 100 CONTINUE
 200 CONTINUE

     IF (MINIC .EQ. 0) CALL READ (Y      , N1, N2,  1,100)
     IF (MINIC .EQ. 1) CALL READ (DATMIN,  6,  3,  6,  3)

     IF (IFPLOT .EQ. 0) GO TO 90
 180 READ (NIT,1003) NCX, NCY, NXPL, XNAME, YNAME, PTITLE
     IF (NCX .NE. 0) GO TO 180
  90 CONTINUE

     READ (NIT,1004)  FXCSM, AKRX, AKRY, AKRZ,
    .                 AKDX, AKDY, AKDZ, OMFX, OMFY, OMFZ,
    .                 ZETFX, ZETFY, ZETFZ, AX, AY, AZ,
    .                 BX, BY, BZ, TLIM
     IF (IFELAS .EQ. 0) GO TO 95
     CALL READ (CMGSLP,6,NMODET,6,26)
  95 CONTINUE
     CALL READ (  CVEC,N1,N2, 6,12)

     IF (IFPER .EQ. 0) GO TO 300
     READ (NIT,2001) (PTITLE(I),I=1,8)
     READ (NIT,2002) IFFIXD,RANGLE,EED,CANGLE
     READ (NIT,2003) (CDELOC(I),I=1,3)
     READ (NIT,2003) (VPLOC (I),I=1,3)

     GO TO 300
```

GAMMA TARGET	Six direction cosines which orient TV body x and y axes with respect to inertial space (see Equation II-135, Vol I)
U - CHASE	Six velocity components for CV
X - CHASE	Three components of CV position vector in inertial space.
GAMMA CHASE	Six direction cosines which orient CV body x and y axes with respect to inertial space.
RHOS	Six parameters that characterize the deformation of the probe assembly (see Equation II-136, Vol I)
MODE VEL - TARGET	TV normal velocity coordinates
MODE DISP - TARGET	TV normal displacement coordinates
MODE VEL - CHASE	CV normal velocity coordinates
MODE DISP - CHASE	CV normal displacement coordinates
CON SYS PARAMETERS	Attitude control system parameters (see Appendix A, Vol I)
DSD VECTOR	x, y, z components of vector from D (apex of cone) to S (probe swivel point) in D coordinate system
PHI	Constraint displacement parameter (instantaneous distance between potential contact points) for each of 10 possible constraints (see Section II, Vol I)
LAMBDA	Constraint force for each of 10 possible constraints (see Section II, Vol I)
MODE	Constraint indicator; if zero constraint is "out", if unity the constraint is "in" (See Section II, Vol I)
RETRACT FORCE	Magnitude of retract force acting on probe barrel piston during the retract mode
BINDING FRICTION FORCE	Binding friction force acting on probe inner barrel.
FORCE AT PT D	Three components of force at apex of drogue cone
MOMENT AT PT D	Three components of moment at apex of drogue cone

FORCE AT PT P	Three components of force at probe hard point
MOMENT AT PT P	Three components of moment at probe hard point
ELAPSED CP TIME	Total elapsed CDC 6400/6500 Central Processor time since beginning of run.

IV. CONCLUSIONS

This volume has presented the graphical output from a series of demonstration runs made with the digital computer program for docking simulation (DOCKEL). The results have a common denominator in that the several runs all considered a fixed orbital configuration. Also, the definition of the probe/drogue docking mechanism was held constant throughout as were the initial conditions. The intent was to demonstrate the overall versatility of the docking program and to present results that might indicate, in a qualitative way, the variations in response due to the inclusion of typical control systems, CV axial thrust, and elastic properties. A total of 12 cases were considered.

As the intent here was to demonstrate capability rather than provide a complete parametric study, conclusions drawn from these results must be limited to the particular configuration that was considered.

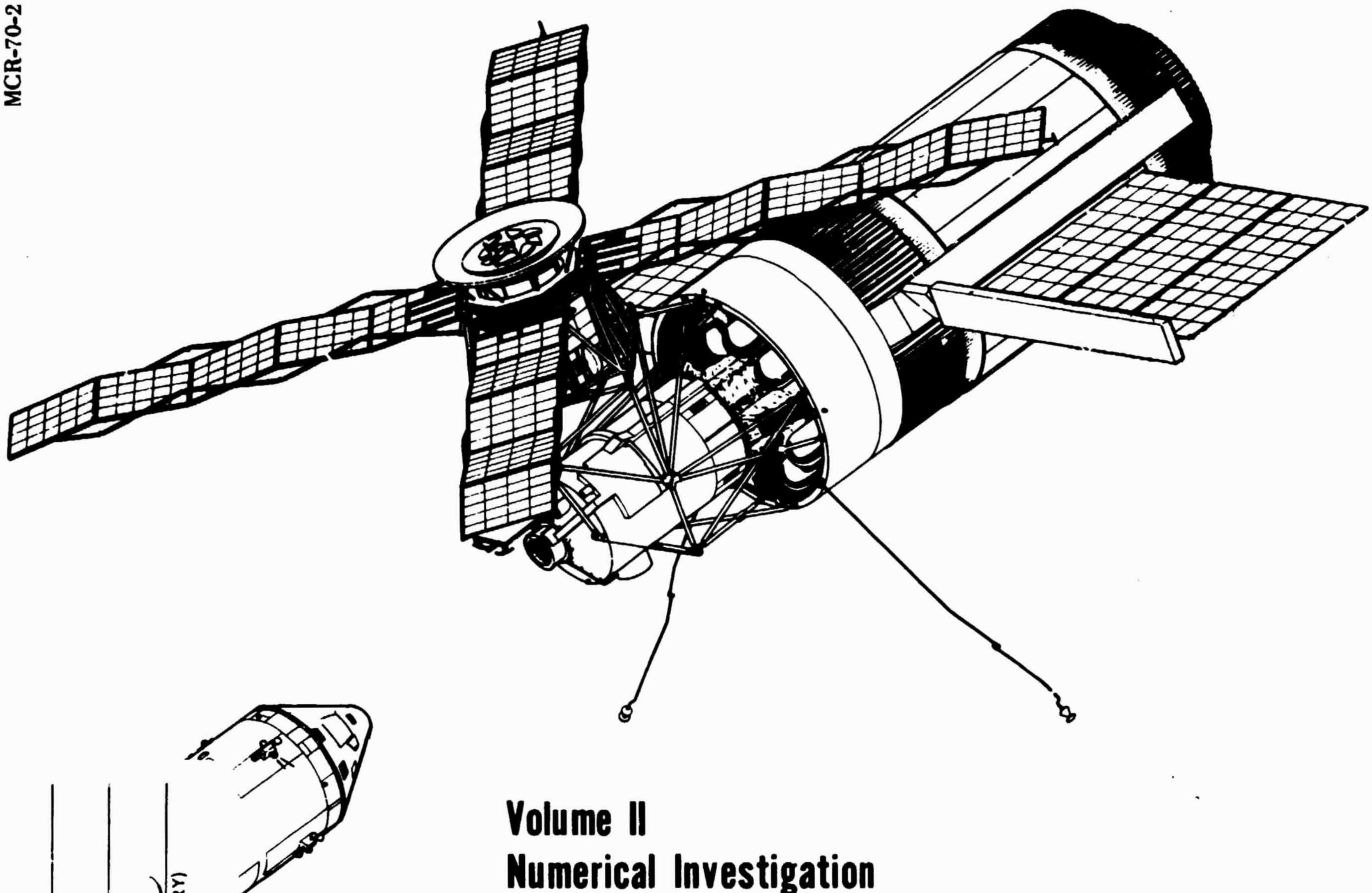
Chase vehicle axial thrust is required for successful probe capture. The effect of increasing thrust is to decrease time to capture while increasing postimpact velocities and loads. There appears to be no significant variation in probe deformation characteristics. The addition of a CMG attitude control system has little effect on probe deformation, capture success, and resulting loads. However, there is a sizable variation observed in postimpact rotational velocities. The CMG system appears to have a stabilizing effect, the degree of stabilization tends to increase with increasing CMG torque capability.

Addition of a TACS has a small effect on capture time, probe deformations, or vehicle rates. A slight increase in resultant forces can be noted as the magnitude of the TACS thruster output is increased. Increasing TACS capability tends to increase the stabilizing effectiveness of the system. The results for the case that considered a RCS show no great difference than the results for the case without this control system except that an RCS does have a tendency to correct the initial CV attitude error.

The most important conclusion to be drawn regarding the exclusion of vehicle elastic effects is that the resultant forces and moments acting on the TV are increased as elastic properties are deleted. Force peaks may be as much as 10% greater for the rigid case than for the case that assumed the TV to have all significant elastic modes with frequency less than 5 Hz. Conversely, the effect of omitting elastic properties is to radically decrease computer running time. The computer time required for the rigid case is only about 1/6 of that required for the case with 14 TV elastic modes.

MCR-70-2 (Vol II) Copy No. 6

Response of Flexible Space Vehicles to Docking Impact



Volume II
Numerical Investigation

MARCH 1970

MARTIN MARIETTA CORPORATION DENVER DIVISION

N70-25587 (ACCESSION NUMBER)
361 (PAGES)
NASA CR-102633 (NASA CR OR TMX OR AD NUMBER)
1 (THRU)
(CODE)
32 (CATEGORY)

As all results were generated with the same vehicle and docking mechanism properties, no conclusions regarding variation in vehicle geometry, inertial properties, or probe mechanism characteristics can be established. A study of the effects on the docking response due to variations in these parameters would require a more extensive investigation than was possible here.

As has been mentioned previously, the intent of the numerical investigations conducted during the course of this study and presented in this volume was not to provide a complete parametric study for a particular configuration or class of configurations. Rather the intent was to provide an illustration of program capability and to illustrate the various available options, particularly those regarding control systems.

The program has, however, been formulated in a manner that lends itself to a parametric study. Care has been exercised in the formulation of the subroutines that read input data in order to make the actual data required a minimum. Likewise, the program output, particularly the time history plots, has been chosen to reflect that information most useful to the analyst.

V. REFERENCES

III-1. FORMA (FORtran Matrix Analysis). Dynamics Memorandum 144. Martin Marietta Corporation, Denver Division, October 1969.

A-1. Internal Letter to R. F. Nicholas from K. A. Bloom. North American Rockwell Space Division, November 15, 1968.

A-2. Dwg DTT100340. North American Aviation, November 5, 1964.

A-3. Verbal Communication, J. Barneburg, Manned Spacecraft Center, Houston, Texas.

A-4. Letter 69MA2739 to H. Harcrow from C. F. DeWitt, North American Rockwell Space Division, March 6, 1969.

APPENDIX A

DOCKING MECHANISM CHARACTERISTICS

This appendix presents a summary of several docking mechanism characteristics that are necessary program input data. A description of probe geometry, friction characteristics, probe preload calculations, attenuator and attachment structure force-deflection data, and retract mechanism characteristics are included.

A. GEOMETRY

Figure A-1 shows a line drawing of the docking probe. The geometry defined on this figure is delineated in Table A-1. The values, except where noted, are from Ref A-1.

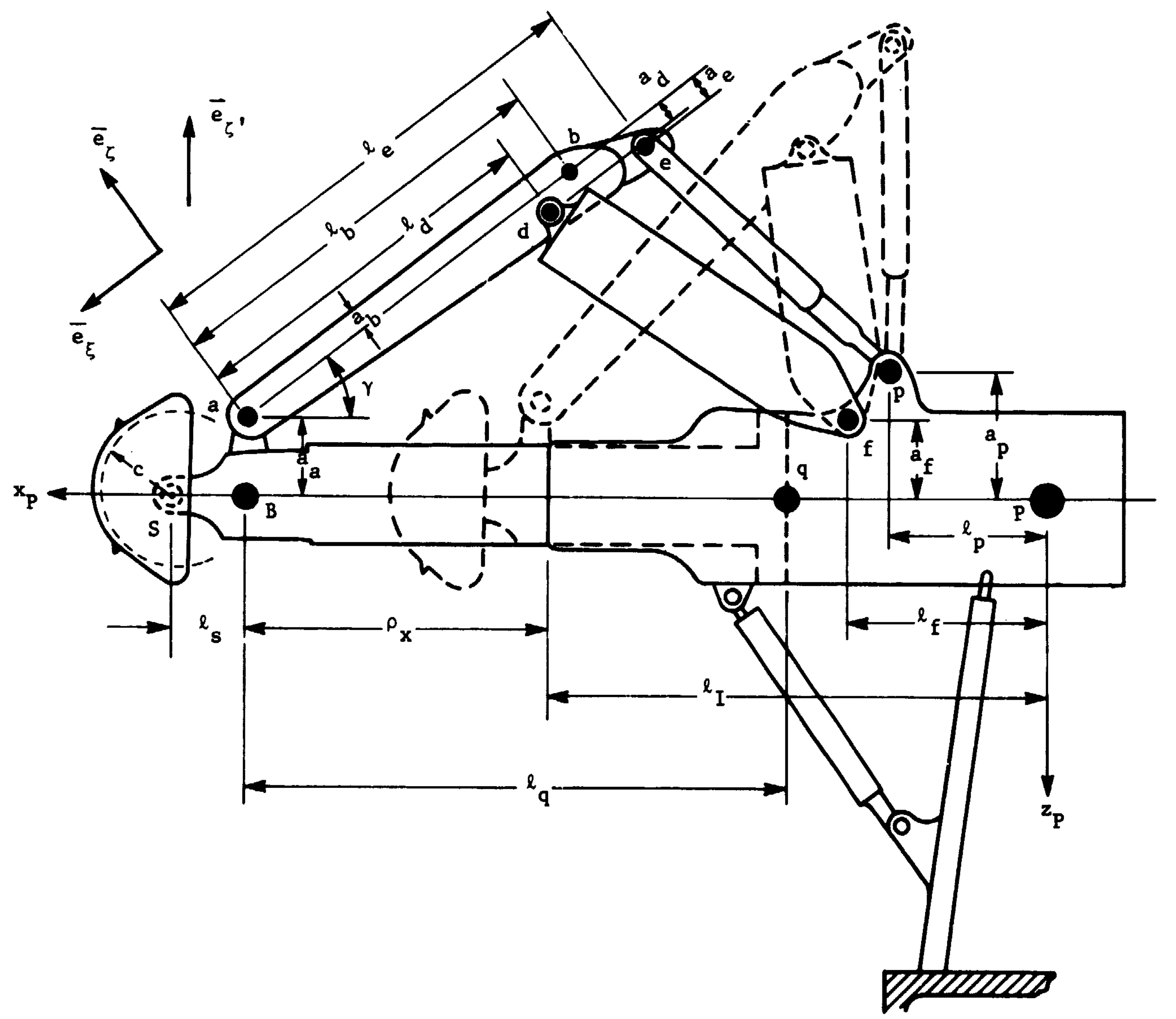


Fig. A-1 Probe Mechanism Geometry

Table A-1 Probe Geometry

ℓ_I = 14.10 in.	a_a = 2.25 in.
ℓ_p = 3.30 in.	a_b^* = 0.85 in.
ℓ_f = 4.75 in.	a_d^* = 0.708 in.
ℓ_s = 2.055 in.	a_e^* = 0.965 in.
a_p = 4.25 in.	ℓ_b^* = 14.48 in.
a_f = 3.085 in.	ℓ_d = 12.90 in.
	ℓ_e = 17.59 in.
	ℓ_q = 16.53 in.
	c = 2.58 in.
*Indicates value scaled from drawing, Ref A-2.	

Note that the analysis considers the probe head to be modeled as a sphere with a radius of 2.58 in. (Ref A-3).

B. BINDING FRICTION

A side load on the probe resulting from probe/drogue contact causes the probe motion to be influenced by a binding friction force. This force can be determined from a static representation of the probe and the pertinent forces, Fig. A-2.

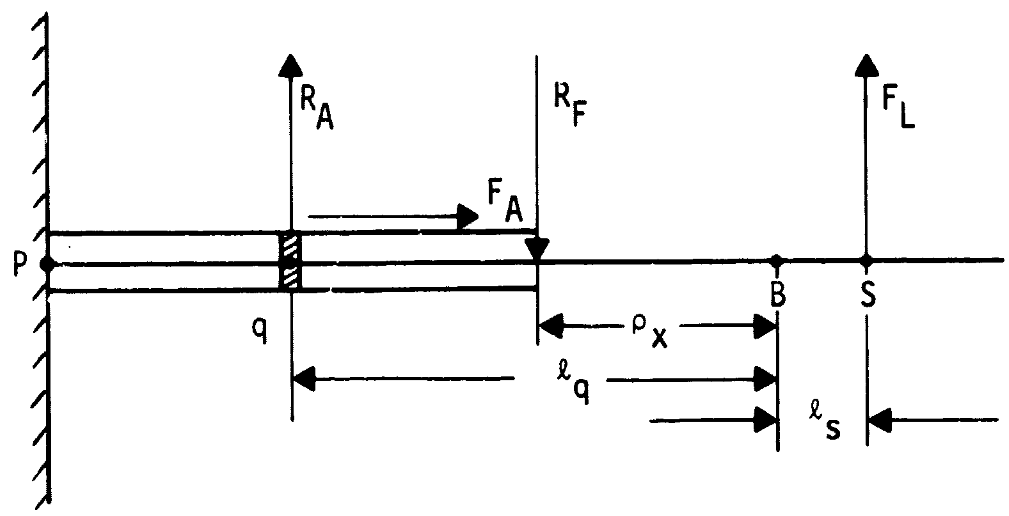


Fig. A-2 Binding Friction Model

From statics it follows that

$$F_L\left(\ell_q + \ell_s\right) = R_F\left(\ell_q - \rho_x\right)$$

$$R_A + F_L = R_F$$

$$R_F = F_L\left[\frac{\ell_q + \ell_s}{\ell_q - \rho_x}\right]; \qquad R_A = F_L\left[\frac{\ell_q + \ell_s}{\ell_q - \rho_x} - 1\right]$$

and the binding friction force is

$$F_A = \mu^*\left(R_F + R_A\right)$$

where μ^* is the binding friction coefficient. The ratio of forces

$$\frac{F_A}{F_L} = \mu^*\left[\frac{\ell_q + 2\ell_s + \rho_x}{\ell_q - \rho_x}\right]$$

can be written

$$\frac{F_A}{F_L} = \mu^*(T)f^*\left(\rho_x\right) = \mu(T)f\left(\rho_x\right)$$

where

$$\mu(T) \equiv f^*\left(\rho_{x_o}\right)\mu^*(T)$$

and is a function of temperature (T),

and $$f\left(\rho_x\right) = f^*\left(\rho_x\right)\Big/f^*\left(\rho_{x_o}\right).$$

Evaluation of $f\left(\rho_x\right)$ is presented in Table A-2 and shown in Fig. A-3. Values of $\mu(T)$ are given in Ref A-1.

Table A-2 Calculation of $f\left(\rho_x\right)$

ρ_x (in.)	$\ell_q + 2\ell_s + \rho_x$ (in.)	$\ell_q - \rho_x$ (in.)	$f^*\left(\rho_x\right)$	$f\left(\rho_x\right)$
11.12	31.77	5.41	5.87	1.0
9.	29.65	7.53	3.94	0.672
7.	27.65	9.53	2.902	0.495
5.	25.65	11.53	2.223	0.379
3.	23.65	13.53	1.747	0.298
1.	21.65	15.53	1.394	0.238
0.54	21.19	15.99	1.322	0.226

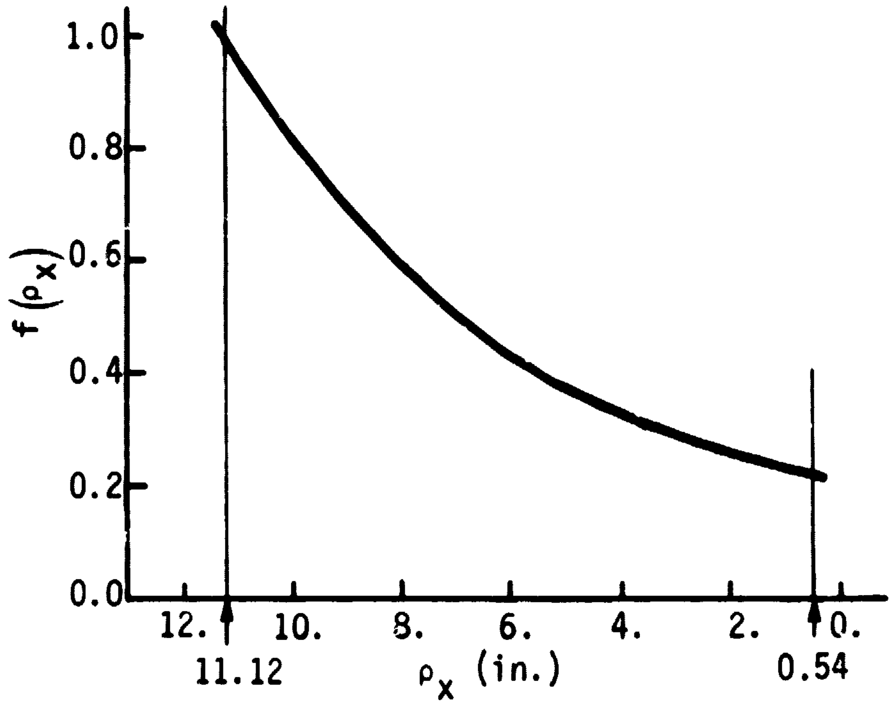


Fig. A-3 Graphical Representation of $f(\rho_x)$

C. ATTACHMENT STRUCTURE FORCE-DEFLECTION DATA

Probe attachment structure force-deflection data (moment versus angular deflection) is presented in Table A-3 and Fig. A-4. The data are from Ref A-4.

Table A-3 Attachment Structure Force-Deflection Data

Tip Deflection (in.)	Load (lb)	Unload (lb)	Angular Deflection (rad)	Load Moment (in.-lb)	Unload Moment (in.-lb)
0	0	0	0	0	0
0.2	230	150	0.00734	6,265	4,085
0.4	475	305	0.01468	12,940	8,310
0.6	715	450	0.02202	19,500	12,280
0.8	960	630	0.02936	26,170	17,180
1.0	1220	815	0.03670	33,220	22,200
1.2	1500	1040	0.04405	40,800	28,350
1.4	1800	1320	0.05135	49,100	36,000
1.6	2170	1725	0.05870	59,150	47,000
1.8	2650	2390	0.06605	72,300	65,150
1.93	3000	3000	0.07080	81,750	81,750

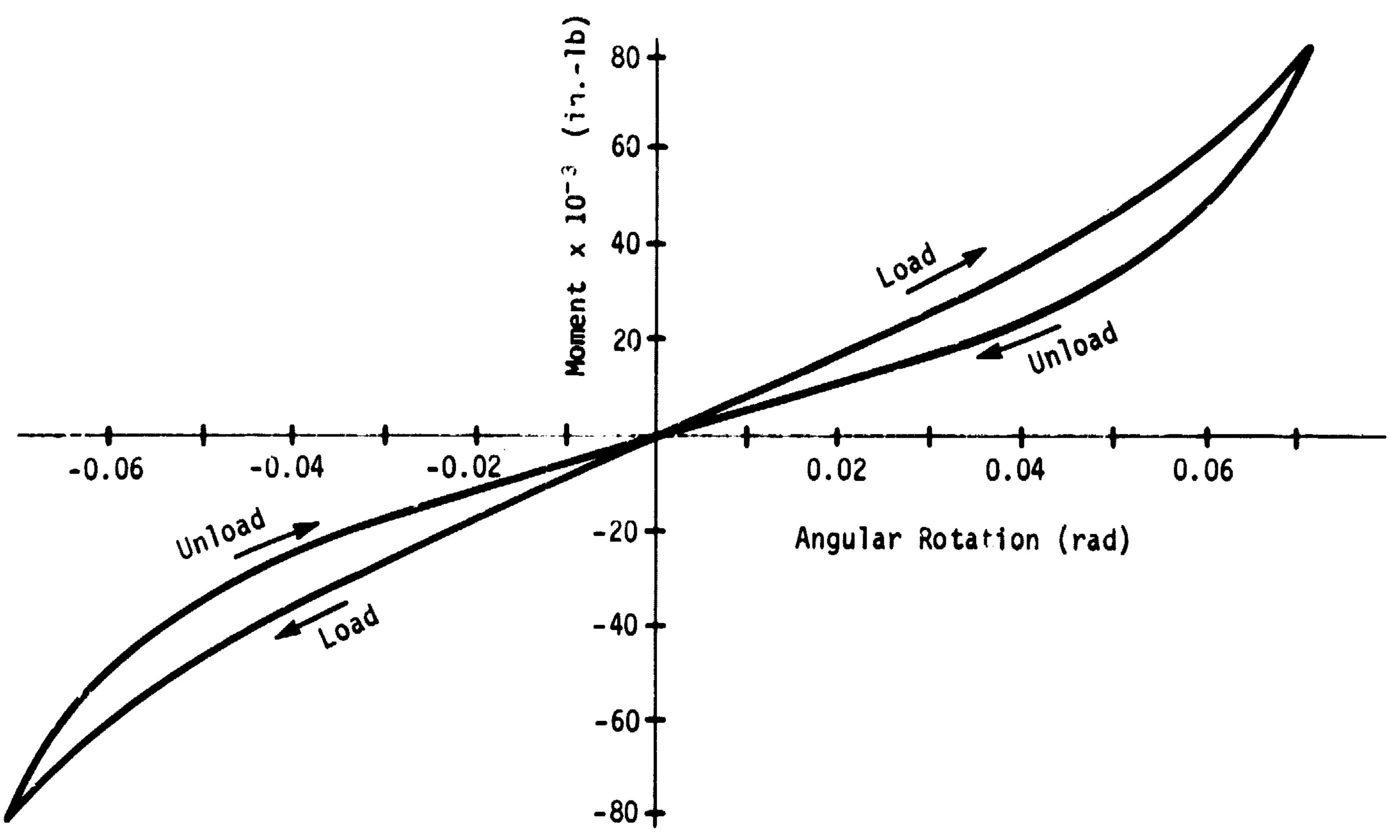


Fig. A-4 Graphical Representation of Attachment Structure Force - Deflection Data

D. PRELOAD BALANCE CALCULATION

When the probe is in the extended position, the three attenuator spring forces, the three tension link forces, and the probe extension spring force must be in static balance. These forces can be determined from the expression for virtual power:

$$\psi_B = F_B v_1^* + \sum_{j=1}^{3}\left(F_{A_j} v_{j+3}^* + F_{T_j} v_{j+6}^*\right).$$

The velocities v_1, v_{j+3}, and v_{j+6}, expressed in terms of $\dot{\rho}_x$, $\dot{\gamma}_1$, $\dot{\gamma}_2$, and $\dot{\gamma}_3$ are (refer to Vol I):

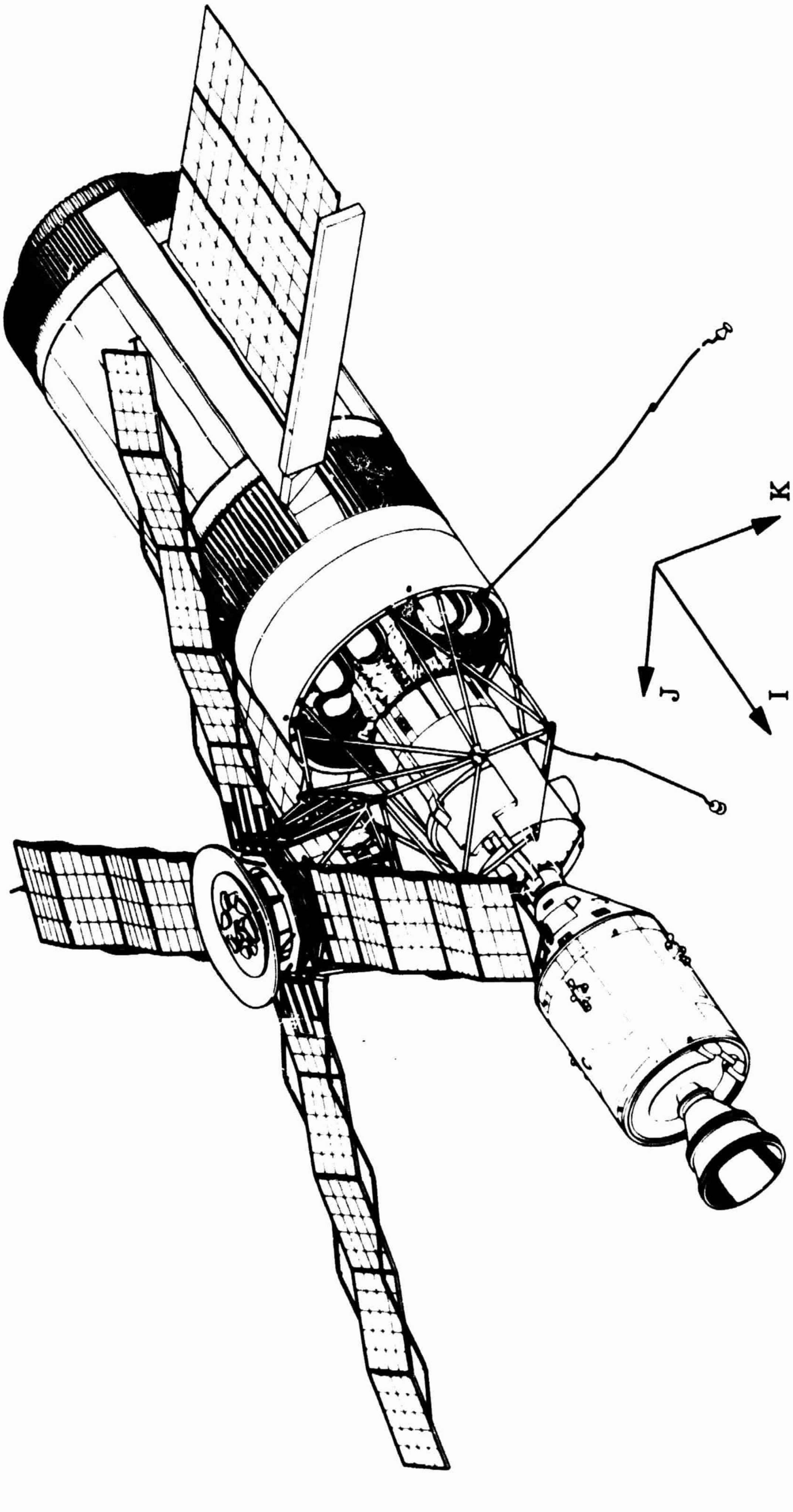


Fig. I-1 Baseline Configuration

$$\begin{Bmatrix} v_1 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{Bmatrix} = \begin{bmatrix} 1 & & & \\ f_1 & f_7 & & \\ f_2 & & f_8 & \\ f_3 & & & f_9 \\ f_4 & f_{10} & & \\ f_5 & & f_{11} & \\ f_6 & & & f_{12} \end{bmatrix} \begin{Bmatrix} \dot{\rho}_x \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ \dot{\gamma}_3 \end{Bmatrix}.$$

It follows that

$$\psi_B = \begin{bmatrix} F_B & F_{A_1} & F_{A_2} & F_{A_3} & F_{T_1} & F_{T_2} & F_{T_3} \end{bmatrix} \begin{bmatrix} 1 & & & \\ f_1 & f_7 & & \\ f_2 & & f_8 & \\ f_3 & & & f_9 \\ f_4 & f_{10} & & \\ f_5 & & f_{11} & \\ f_6 & & & f_{12} \end{bmatrix} \begin{Bmatrix} \dot{\rho}_x^* \\ \dot{\gamma}_1^* \\ \dot{\gamma}_2^* \\ \dot{\gamma}_3^* \end{Bmatrix}.$$

Symmetry of the probe requires that

$$F_{A_1} = F_{A_2} = F_{A_3}, \quad F_{T_1} = F_{T_2} = F_{T_3},$$

$$\gamma_1 = \gamma_2 = \gamma_3, \quad \dot{\gamma}_1 = \dot{\gamma}_2 = \dot{\gamma}_3,$$

and therefore

$$\psi_B = \begin{bmatrix} F_B & F_{A_1} & F_{T_1} \end{bmatrix} \begin{bmatrix} 1 & \\ (f_1 + f_2 + f_3) & (f_7 + f_8 + f_9) \\ (f_4 + f_5 + f_6) & (f_{10} + f_{11} + f_{12}) \end{bmatrix} \begin{Bmatrix} \dot{\rho}_x^* \\ \dot{\gamma}_1^* \end{Bmatrix}.$$

The virtual power, expressed in another way, is

$$\psi_B = F_{\rho_x}\dot{\rho}_x^* + F_{\gamma 1}\dot{\gamma}_1^*$$

and the preload equilibrium equations are

$$\begin{Bmatrix} F_{\rho_x} \\ F_{\gamma 1} \end{Bmatrix} = \begin{bmatrix} 1 & (f_1 + f_2 + f_3) & (f_4 + f_5 + f_6) \\ & (f_7 + f_8 + f_9) & (f_{10} + f_{11} + f_{12}) \end{bmatrix} \begin{Bmatrix} F_B \\ F_{A1} \\ F_{T1} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}.$$

Again, symmetry of the probe implies that

$$f_1 = f_2 = f_3, \quad f_4 = f_5 = f_6$$

$$f_7 = f_8 = f_9, \quad f_{10} = f_{11} = f_{12}$$

and the equilibrium equations reduce to

$$\begin{bmatrix} 1 & 3f_1 & 3f_4 \\ & 3f_7 & 3f_{10} \end{bmatrix} \begin{Bmatrix} F_B \\ F_{A1} \\ F_{T1} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}.$$

The coefficients f_1, f_4, f_7, f_{10} can be calculated using known geometry. The notation used here is that of Volume I.

$$f_1 = \overline{fd} \cdot \overline{i}_P / |\overline{fd}|$$

$$\overline{fd} \cdot \overline{i}_P = \ell_I - \ell_f + \rho_x - \ell_d \cos\gamma - a_d \sin\gamma = 9.665$$

$$\overline{fd} \cdot \overline{j}_P = 0$$

$$\overline{fd} \cdot \overline{k}_P = a_f - a_a - \ell_d \sin\gamma + a_d \cos\gamma = -6.247$$

$$|\overline{fd}| = 11.508$$

$$f_1 = u_{A1x} = 0.84113, \quad u_{A1y} = 0., \quad u_{A1z} = -0.5437$$

$$f_7 = \left(\ell_d \sin\gamma - a_d \cos\gamma\right) u_{A1x} - \left(\ell_d \cos\gamma + a_d \sin\gamma\right) u_{A1z} = 11.8312$$

$$f_4 = -\overline{pe} \cdot \overline{i}_P \Big/ \left|\overline{pe}\right|$$

$$\overline{pe} \cdot \overline{i}_P = \ell_I - \ell_p + \rho_x - \ell_e \cos\gamma - a_e \sin\gamma = 7.187$$

$$\overline{pe} \cdot \overline{j}_P = 0$$

$$\overline{pe} \cdot \overline{k}_P = a_p - a_a - \ell_e \sin\gamma + a_e \cos\gamma = -7.657$$

$$\left|\overline{pe}\right| = 10.5012$$

$$f_4 = -u_{T1x} = -0.6844,\ u_{T1y} = 0.,\ u_{T1z} = -0.7292$$

$$f_{10} = -\left(\ell_e \sin\gamma - a_e \cos\gamma\right) u_{T1x} + \left(\ell_e \cos\gamma + a_e \sin\gamma\right) u_{T1z} = -17.683$$

Assuming now that the attenuator preload (Ref A-1) is

$$F_{A_1} = 65/2 = 32.5 \text{ lb}$$

yields $F_{T_1} = 21.8$ lb and $F_B = -37.2$ lb.

E. EXTEND SPRING LOAD-DEFLECTION DATA

The data presented in Table A-4 and Fig. A-5 for the probe extend spring load-deflection are from Ref A-1.

Table A-4 Extend Spring Load-Deflection Data

Deflection (in.)	Load (lb)	ρ_x (in.)
15.	- 69.	11.12*
13.5	- 74.8	9.62
12.	- 81.	8.12
10.	- 89.3	6.12
9.	- 93.75	5.12
7.5	-101.	3.62
6.5	-106.2	2.62
5.5	-112.5	1.62
4.75	-118.	0.87
4.44	-121.	0.56
4.2	-125.	0.32
4.07	-130.	0.19

<u>Note</u>: 1. * indicates fully extended probe.
2. $\rho_x = \delta - 3.88$ in.
3. Bottoming spring constant ≡ 40×10^3 lb/in.

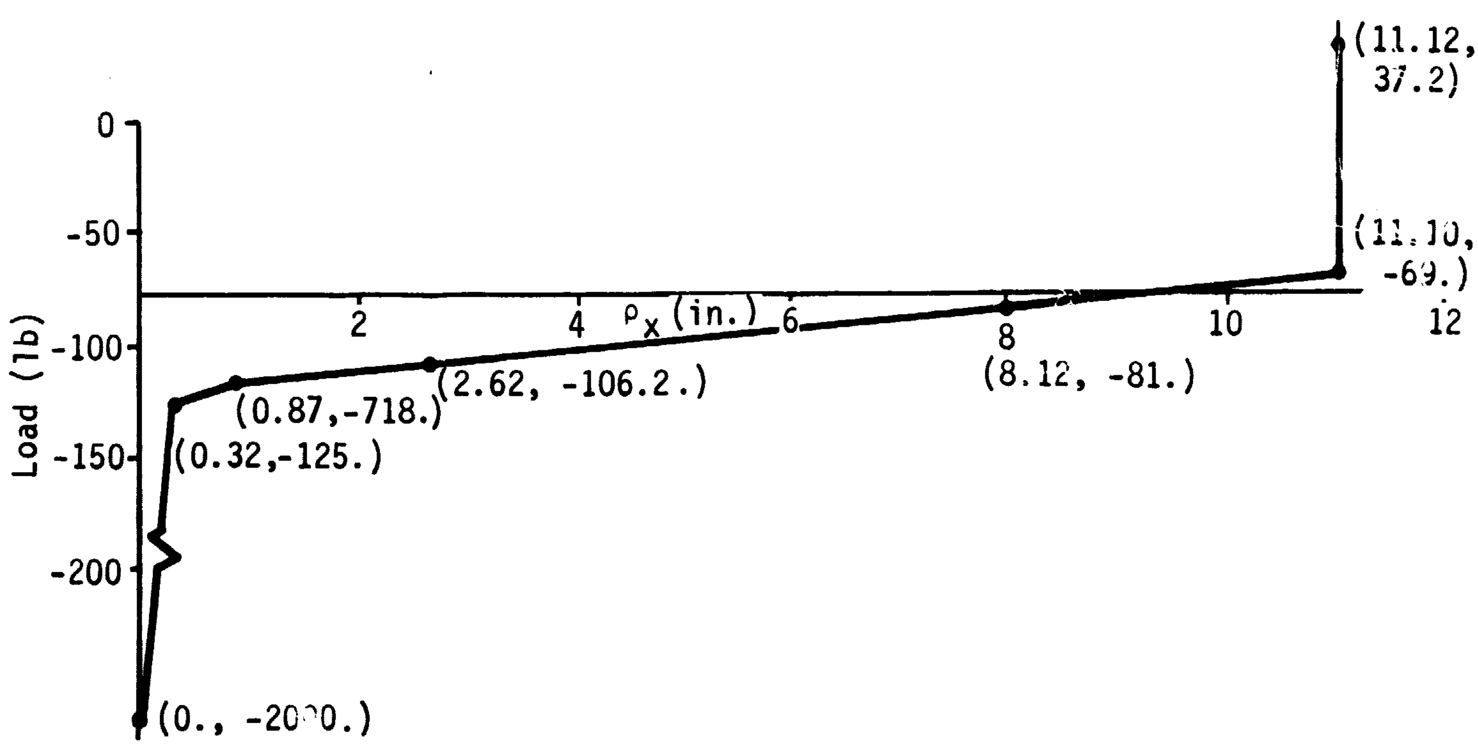


Fig. A-5 Graphical Representation of Extend Spring Load-Deflection Data

2. Target Vehicle

The TV coordinate system is defined in Fig. I-3. Vehicle inertial properties are shown in Table I-3. The elastic effects of this vehicle are considered through the inclusion of significant elastic modes with natural frequencies less than 5 Hz.

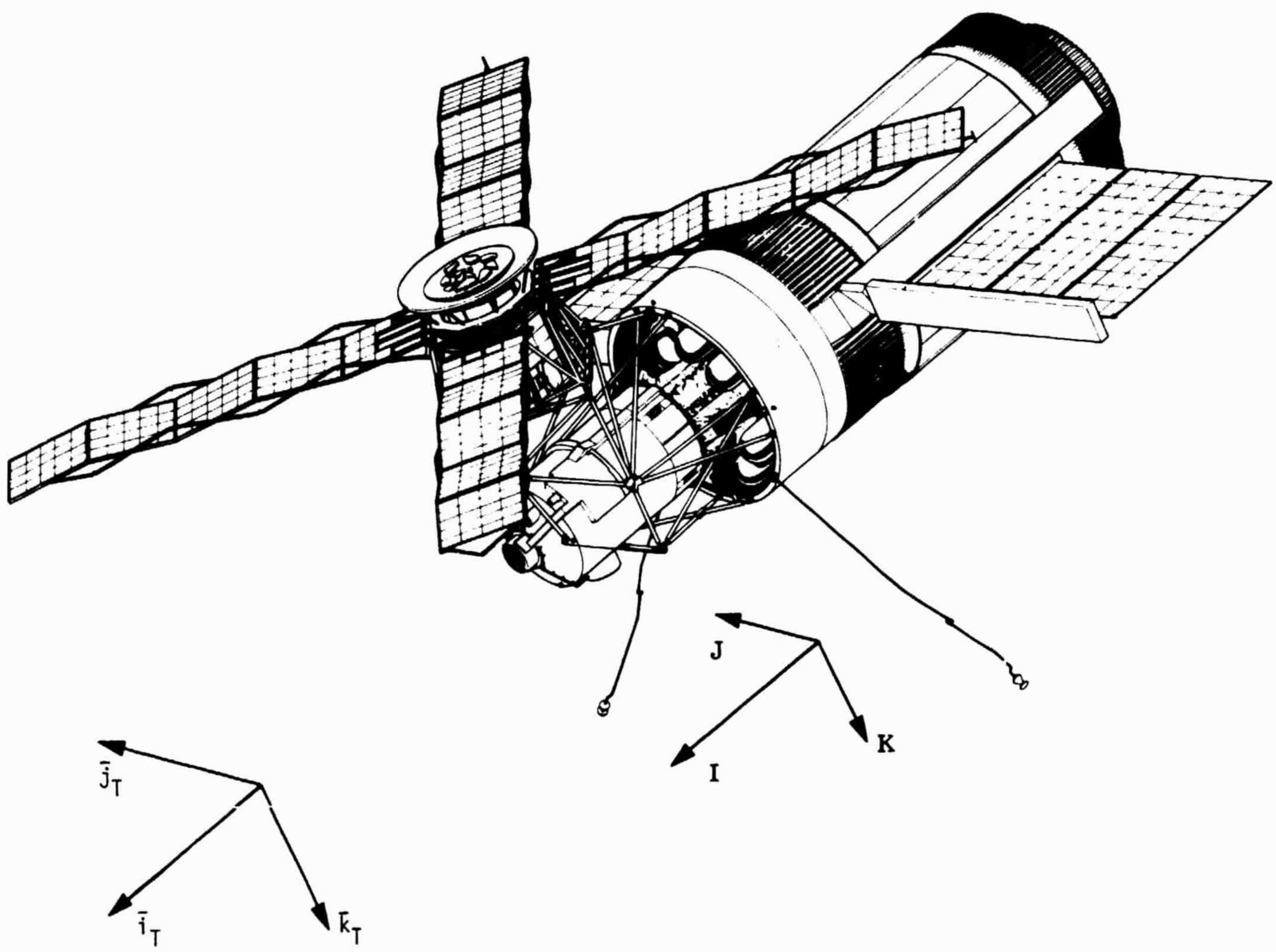


Fig. I-3 Target Vehicle Coordinate System

Table I-3 Target Vehicle Inertial Properties

M = 296.4 lb-sec^2/in.	
I_{xx} = 6.428x10^6 lb-in.-sec^2	I_{xy} = 1.622x10^4 lb-in.-sec^2
I_{yy} = 2.474x10^7 lb-in.-sec^2	I_{xz} = 3.528x10^6 lb-in.-sec^2
I_{zz} = 2.512x10^7 lb-in.-sec^2	I_{yz} = -1.575x10^5 lb-in.-sec^2

F. ATTENUATOR LOAD-STROKE DATA

The attenuator load-stroke data shown in Table A-5 is from Ref A-1. The data are shown graphically in Fig. A-6.

Table A-5 Attenuator Load-Stroke Data

Attenuator Deflection, δ_A (in.)	Load (lb)	Unload (lb)	Δ_A (in.)	Average Load (lb)
0	65	0	11.5	- 32.5
0.25	56	21	11.25	- 38.5
0.5	63	28	11.0	- 45.5
1.0	78	47	10.5	- 62.5
1.5	103	70	10.0	- 86.5
2.0	140	107	9.5	-123.5
2.5	203	167	9.0	-185.0
3.0	330	280	8.5	-305.0
3.07	360	360	8.43	-360.0

Note: $fd_0 = 11.5$, $\Delta_A = fd_0 - \delta_A$.

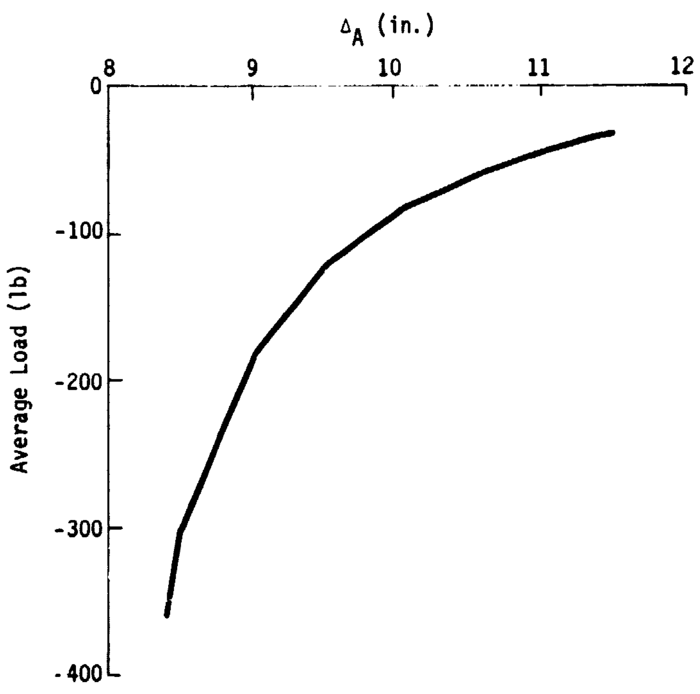


Fig. A-6 Graphical Representation of Attenuator Load-Stroke Data

G. ATTENUATOR DAMPING CHARACTERISTICS

The attenuator damping characteristics shown in Table A-6 are from Ref A-1.

Table A-6 Attenuator Damping Characteristics

Attenuator Velocity (in./sec)	Load (lb)		
	T = 70°F - 250°F	T = -65°F	T = -80°F
1.32	78	78	78
0.12	18	18	18
- 0.12	- 18	- 18	- 18
- 0.24	- 210	- 500	- 690
- 0.60	- 405	- 680	- 915
- 1.20	- 630	- 890	-1205
- 2.76	- 910	-1240	-1710
- 4.8	-1175	-1610	-2210
- 7.2	-1435	-2005	-2770
- 9.6	-1680	-2370	-3335
-12.0	-1920	-2730	
-14.4	-2170	-3115	
-16.8	-2450	-3550	
-18.25	-2685	-3850	
-19.2	-2980		

H. RUNNING FRICTION

The probe running friction data shown in Fig. A-7 are from Ref A-1.

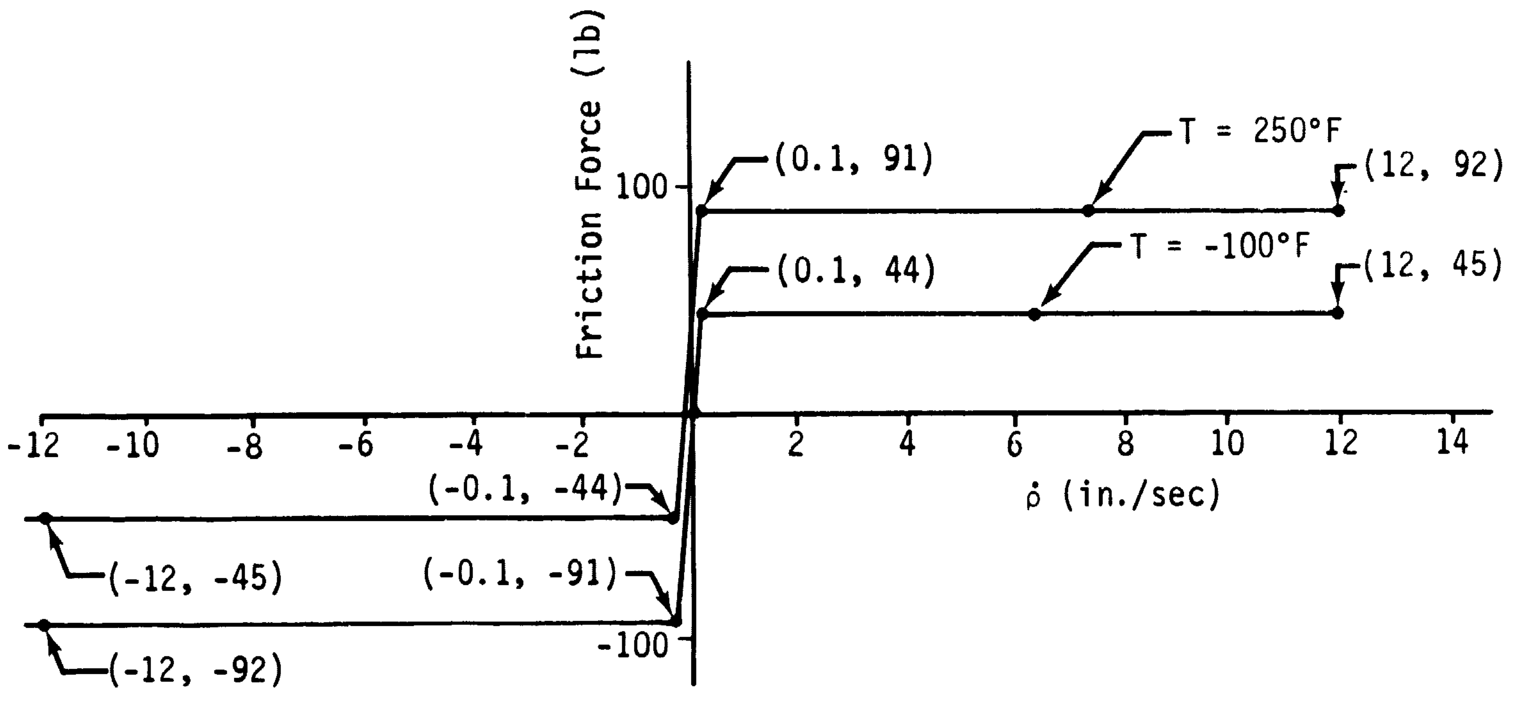


Fig. A-7 Graphical Representation of Running Friction

I. RETRACT FORCE

The flow rate of a compressible perfect gas through an orifice can be expressed by the equation,

$$\dot{W} = C_o \sqrt{\frac{g\gamma}{R}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{\gamma-1}}} \frac{A_o p_B}{\sqrt{T_B}}$$

Data from Ref A-1 indicate the following values for the docking probe retract system:

p_B = Bottle pressure = 7.2×10^5 lb/ft^2;

Bottle volume = 3.55×10^{-3} ft^3;

T_B = Bottle temperature = 530°R;

Bottle gas mass = 0.0827 lb-sec^2/ft;

A_o = Bottle orifice area = 1.35×10^{-7} ft^2;

Bottle secondary orifice area = 9.43×10^{-7} ft^2;

Stroke to open secondary orifice = 0.7547 ft;

γ = Specific heat ratio = 1.05;

Retract piston area = 0.0328 ft^2;

R = Gas constant = 55.16 ft/°R;

C_o = Orifice coefficient = 0.62 (assumed).

The flow rates for the primary and secondary orifices are then:

$\dot{W}_{primary} = 1.22 \times 10^{-3}$ lb/sec;

$\dot{W}_{secondary} = 8.5 \times 10^{-3}$ lb/sec;

An expression for the retract piston force time history can be constructed from the flow rate and probe geometry, Fig. A-8.

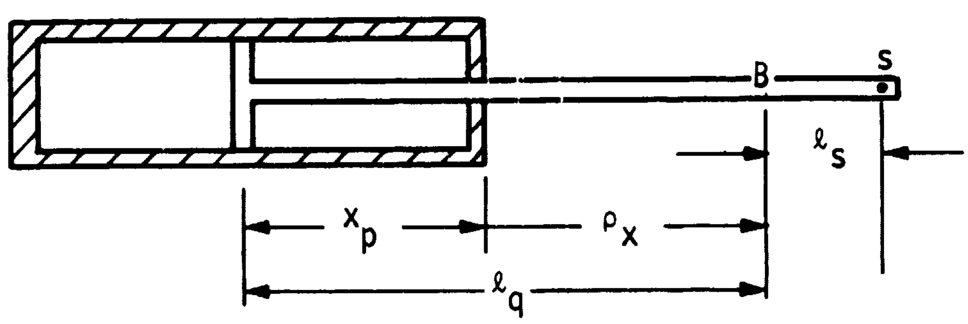


Fig. 8 Schematic for Retract Piston Force Time History

The force acting on the piston is

$$F_p = \frac{12\dot{W}RT\ \Delta t}{x_p} = \frac{12\dot{W}RT\ \Delta t}{\ell_q - \rho_x}$$

Evaluating the above yields $F_{p1} = \frac{428.22\ \Delta t}{16.53 - \rho_x}$ for $2.08 \le \rho_x \le 11.12$ (primary orifice open) and $F_{p2} = F^*_{p1} + \frac{2983.5(\Delta t - \Delta t^*)}{16.53 - \rho_x}$ for $\rho_x <$ 2.08 (primary and secondary orifices open). The value F^*_{p1} is the value of F_{p1} at the time (Δt^*) that the secondary orifice opens.

The output chosen for presentation represents only a fraction of that output available from the program and was selected as being most representative for the individual cases. A summary of the output for each case is shown in Table II-3.

Table II-3 Output Table

Case	Configuration	Output Set No.												Figure
		1	2	3	4	5	6	7	8	9	10A	10B	10C	
SC-1	Baseline Configuration (BLC)	■	■	■	■	■		■	■	■				II-1
SC-2	BLC + Nominal CMGs								■	■	■			II-2
SC-3	BLC + "High" CMGs	■							■	■	■			II-3
SC-4	BLC + "Low" TACS	■							■	■		■		II-4
SC-5	BLC + "Moderate" TACS	■							■	■		■		II-5
SC-6	BLC + Nominal TACS	■							■	■		■		II-6
SC-7	BLC + Nominal RCS	■						■	■	■			■	II-7
SC-8	BLC - Elastic Modes	■		■				■	■	■				II-8
SC-9	BLC - Elastic Modes with f > hz	■						■	■	■				II-9
SC-10	BLC - Axial Thrust	■	■		■				■	■				II-10
SC-11	BLC + 200-lb Axial Thrust	■	■		■				■	■				II-11
SC-12	BLC + Nominal CMGs, Nominal RCS, Nominal TACS, + Probe Retract (Elastic Modes f ≤ 2 Hz)	■	■						■	■				II-12

APPENDIX B

CONTROL SYSTEMS CHARACTERISTICS

This appendix summarizes the attitude control systems considered in the docking simulation. Three control systems are described; each system may be included or omitted at the user's option.

A. CONTROL MOMENT GYRO ATTITUDE CONTROL SYSTEM

A block diagram for the CMG system acting on the target vehicle is shown in Fig. B-1. The transfer function (1/F) for each channel is of the form.

$$\frac{1}{F_{x,y,z}} = \frac{1}{\left[\frac{s^2}{\omega_f^2} + \frac{2\zeta s}{\omega_f} + 1\right]^2}$$

with the parameters for the three channels shown in Table B-1.

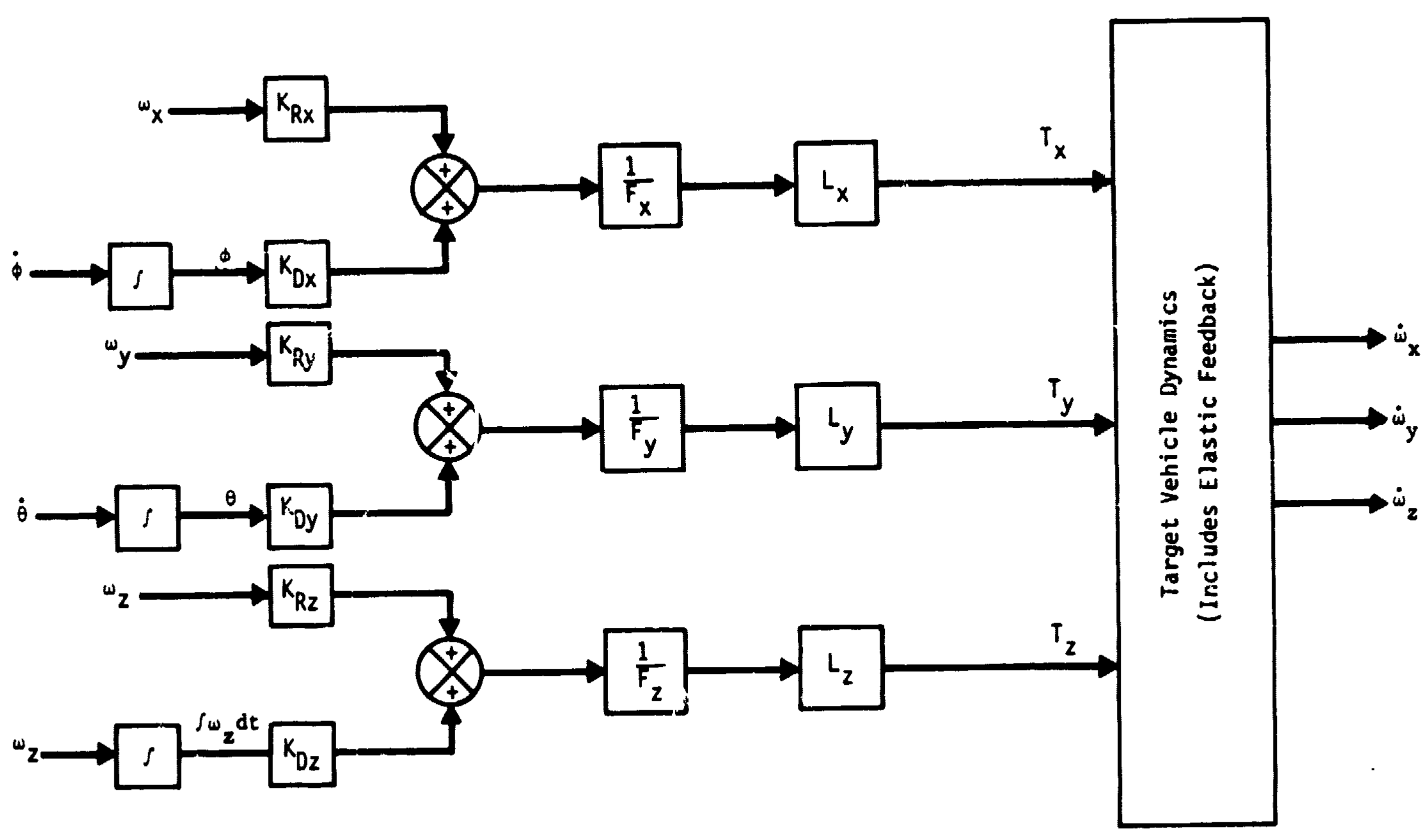


Fig. B-1 Control Moment Gyro Attitude Control System Block Diagram

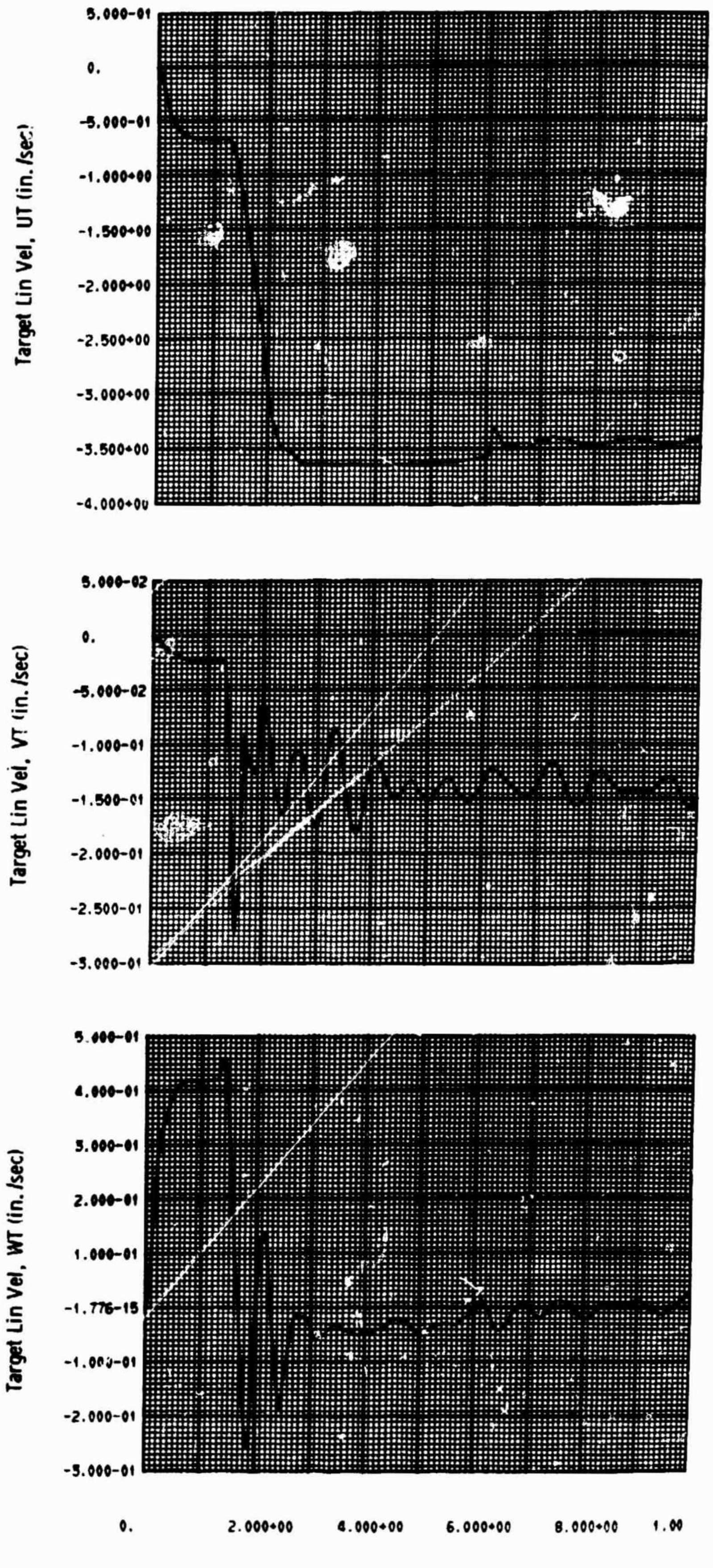


Fig. II-1 Output for Case SC-1 (Sheet 1 of 15)

Table B-1 CMG Attitude Control System Parameters

Parameter	Control Channel		
	x	y	z
K_D (in.-lb/rad)	5.57×10^5	27.18×10^5	53.8×10^5
K_R (in.-lb sec/rad)	25×10^5	176×10^5	242×10^5
ζ, Filter Damping Ratio	0.5	0.5	0.5
ω_f, Filter Break Frequency (rad/sec)	2.2	1.25	1.9

The transfer function $L_{x,y,z}$ is

$$L_x = L_y = L_z = \frac{a}{\tau s + 1}$$

with $a = 0.61$, $\tau = 0.24$. The total CMG output torque is limited to 1440 in.-lb.

B. THRUSTER ATTITUDE CONTROL SYSTEM/REACTION CONTROL SYSTEM

A block diagram for the TACS/RCS is shown in Fig. B-2. The TACS is a control system acting on the target vehicle. The RCS controls the chase vehicle. Significant control system parameters for the two systems are presented in Tables B-2 and B-3.

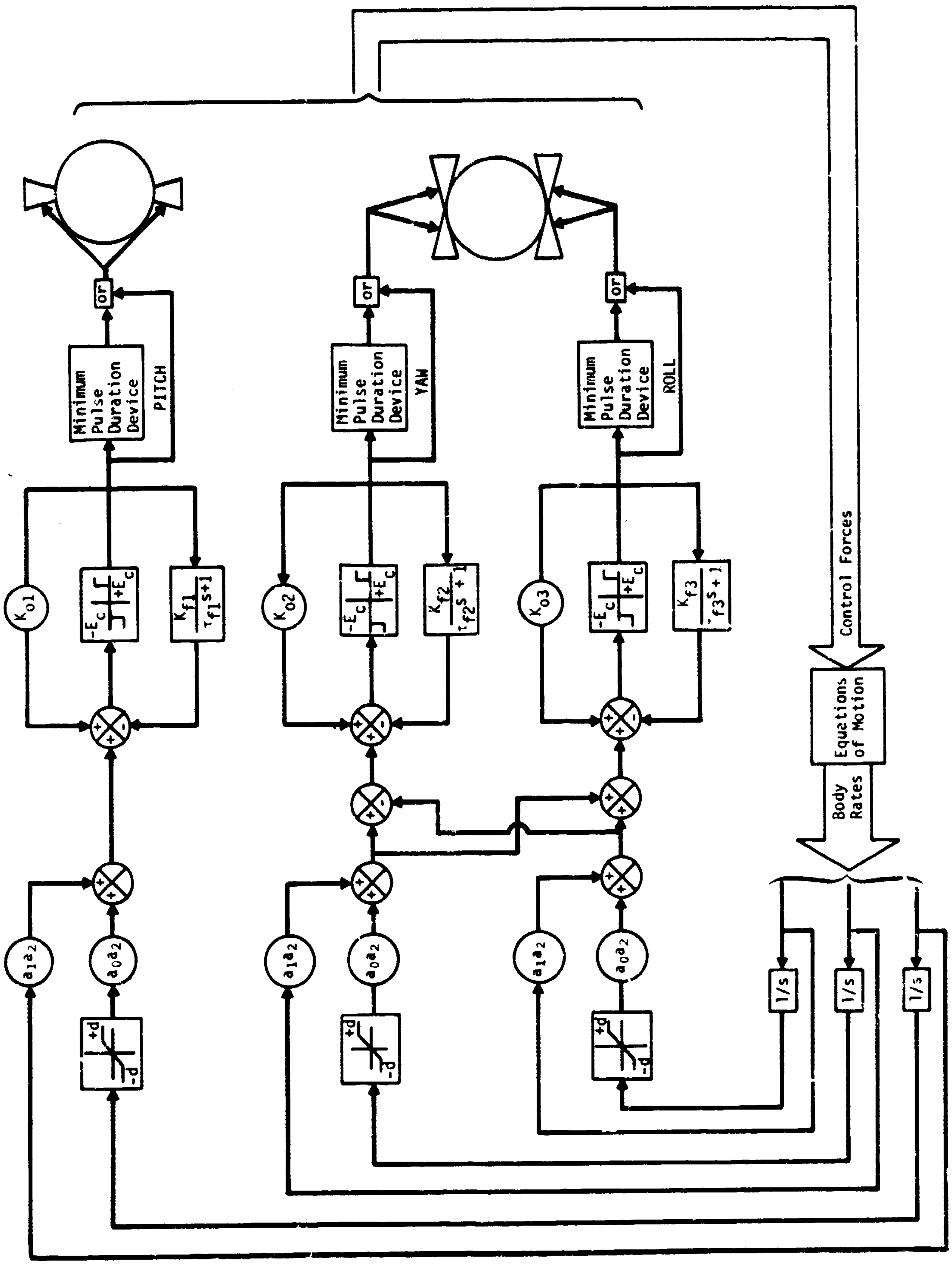


Fig. B-2 Thruster Attitude Control System/Reaction Control System Block Diagram

Table B-2 Thruster Atti[illegible] Control System Parameters

Parameter	Control Channel		
	Pitch	Yaw	Roll
d, Displacement limit (deg)	±2	±2	±2
a_1, Rate gain (deg/deg/sec)	20	20	20
a_0, Displacement gain (deg/deg)	2	2	2
a_2, Mixer gain (volt/deg)	20	20	20
K_o, Feedback gain (volt/volt)	0	0	0
K_f, Feedback gain (volt/volt)	0	0	0
τ_f, Time constant (sec)	0.5	0.5	0.5
Switch dead space (volts)	±20	±20	±20
E_c, Switch output (volts)	±100	±100	±100
Minimum pulse width (sec)	0	0	0

Table B-3 Reaction Control System Parameters

Parameter	Control Channel		
	Pitch	Yaw	Roll
d, Displacement limit (deg)	±4	±4	±4
a_1, Rate gain (deg/deg/sec)	1	1	1
a_0, Displacement gain (deg/deg)	0.5	0.5	0.5
a_2, Mixer gain (volt/deg)	10	10	10
K_o, Feedback gain, (volt/volt)	0.007	0.007	0.007
K_f, Feedback gain (volt/volt)	1.0	1.0	1.0
τ_f, Time constant (sec)	1.0	1.0	1.0
Switch dead space (volts)	±20	±20	±20
E_c, Switch output (volts)	±100	±100	±100
Minimum pulse width (sec)	0	0	0

Each thruster is assumed to produce 100 lb of thrust. Using known geometry for the TV and the CV yields the control forces (torques) shown in Table B-4.

Table B-4 TACS/RCS Control Forces and Moments

TACS		
	T_x = 28,320 in.-lb	(2 nozzles, 11.8 ft center line to nozzle)
	F_y = 200 lb	(2 nozzles)
	F_z = 100 lb	(1 nozzle)
RCS		
	T_x = 15,360 in.-lb	(2 nozzles, 6.4 ft center-line to nozzle)
	T_y = 15,360 in.-lb	
	T_z = 15,360 in.-lb	

APPENDIX C

PROGRAM DOCKEL LISTING

Appendix C presents a complete listing of DOCKEL. Sample input data appear in Appendix D. Sample output, using the input of Appendix D, appears in Appendix E. Subroutines required by the program that are included in the FORMA library are not included in this appendix. A listing and a complete description of these subroutines will be found in Ref III-1.

```
      SUBROUTINE ATXBV (A,B,C)
      DIMENSION  A(3,3), B(3), C(3)
C
C   CODED BY CARL BODLEY   MARCH 1968
C
      DO 10 I=1,3
      C(I) = 0.0
      DO 10 K=1,3
   10 C(I) = C(I) + A(K,I)*B(K)
      RETURN
      END


      SUBROUTINE AVPBV (ALP,A,BET,B,C)
      DIMENSION A(3), B(3), C(3)
C
C   CODED BY CARL BODLEY   MARCH 1968
C
      DO 10 I=1,3
   10 C(I) = ALP*A(I) + BET*B(I)
      RETURN
      END


      SUBROUTINE  AXB3  (A,B,C)
      DIMENSION   A(3,3), B(3,3), C(3,3)
C
C   CODED BY CARL BODLEY   JUNE  1968
C
      DO 10 I=1,3
      DO 10 J=1,3
      C(I,J) = 0.0
      DO 10 K=1,3
   10 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RETURN
      END


      SUBROUTINE  AXBT3 (A,B,C)
      DIMENSION   A(3,3), B(3,3), C(3,3)
C
C   CODED BY CARL BODLEY   MARCH 1968
C
      DO 10 I=1,3
      DO 10 J=1,3
      C(I,J) = 0.0
      DO 10 K=1,3
   10 C(I,J) = C(I,J) + A(I,K)*B(J,K)
      RETURN
      END


      SUBROUTINE AXBV  (A,B,C)
      DIMENSION  A(3,3), B(3), C(3)
C
C   CODED BY CARL BODLEY   MARCH 1968
C
      DO 10 I=1,3
      C(I) = 0.0
      DO 10 K=1,3
   10 C(I) = C(I) + A(I,K)*B(K)
      RETURN
      END
```

```
      SUBROUTINE BAKOFF
C
C     SUBROUTINE TO INCREMENT THE RHO COORDINATES IN SUCH A WAY THAT NO
C     PHICON .LT. 0.
C
C     SUBROUTINES THAT ARE CALLED ARE --- INV2NP AND MULT
C
C     CODED BY CARL BODLEY, JULY 1969
C
      COMMON /BTDATA/ TRHO(9,6), BLS(10,6), BLV(7,39)
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION DELPHI(10),DELRHO(6),BC(6,6),BCI(6,6),WKV(6)
      DIMENSION                    IVEC(10), JVEC(6)
C
      DATA  EPSN/0.005  /
      DATA JVEC/1,2,3,4,5,6/
C
      DO 10 I=1,10
      AMODE = MODE(I)
   10 DELPHI(I) = AMODE*(EPSN - PHICON(I))
C
      IVEC(4) = 0
      DO 15 I=1,3
      IP4 = I + 4
      IP7 = I + 7
      IVEC(I   ) = MODE(I   )
      IVEC(IP4) = MODE(IP4)
      IF (MODE(I) .EQ. 0 .AND. MODE(IP4) .EQ. 0) IVEC(IP4) = 1
   15 IVEC(IP7) = 1
C
      NVAL = 0
      DO 20 I=1,10
      IF (IVEC(I) .EQ. 0) GO TO 20
      NVAL = NVAL + 1
      IVEC(I) = NVAL
   20 CONTINUE
C
      CALL REVISE (BLS,IVEC,JVEC,BC,10,6,6,6,10,6)
      CALL REVISE (DELPHI,IVEC,JVEC(1),WKV,10,1,6,1,10,6)
C
      IF (MODE(4) .EQ. 0) GO TO 50
C
      DO 25 I=1,3
   25 WKV(I+3) = -DELPHI(4)*TRHT(2,I)
C
   50 CALL INV2NP (BC,BCI,6,6)
      CALL MULT   (BCI,WKV,DELRHO,6,6,1,6,6)
C
      DO 60 I=1,6
      L = I + NRHO - 1
   60 Y(L) = Y(L) + DELRHO(I)
C
      RETURN
      END
```

```
      SUBROUTINE CONSYS
C
      COMMON /BC1234/ BC1(10,6), BC2(10,6), BC3(10,26), BC4(10,26)
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /CSYSTM/ CVEC(6,15), EULR(6), EDOT(6)
      COMMON /CMGDTA/ FXCSM, AKRX,AKRY,AKRZ,AKDX,AKDY,AKDZ,OMFX,OMFY,
     *                      OMFZ,ZETFX,ZETFY,ZETFZ,AX,AY,AZ,BX,BY,BZ,
     *                      TLIM
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /JBATCH/ JBA
      COMMON /MISCNO/ ANUM
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /PULSES/ QPA(6)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FRET
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /VECTOR/ Y(150), YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION OMGG(3)
C
      DATA NIT, NOT  / 5, 6 /
      DATA I1ST, TEST / 0,  0.0 /
C
C    SET FORCE TRANSFORMATIONS AND READ CONTROL AND ELASTIC INPUT.
C
      IF (JBA .EQ. 1 .AND. I1ST .EQ. 0) GO TO 3
      GO TO 5
    3 CALL ZERO (BC1, NFORCS, 6, 10)
      CALL ZERO (BC2, NFORCS, 6, 10)
      CALL ZERO (BC3, NFORCS, NMODET, 10)
      CALL ZERO (BC4, NFORCS, NMODEC, 10)
      READ (NIT,2001) FXCSM, AKRX,AKRY,AKRZ,AKDX,AKDY,AKDZ,OMFX,OMFY,
     *                      OMFZ,ZETFX,ZETFY,ZETFZ,AX,AY,AZ,BX,BY,BZ,
     *                      TLIM
 2001 FORMAT (8E10.0)
      IF (IFELAS .EQ. 0) GO TO 15
      CALL READ   (BC3(2,1),N1,NMDT,10,26)
      IF (NMDT .EQ. NMODET)  GO TO 15
      WRITE (NOT,1001)
 1001 FORMAT (25H1GYRO SIGS NOT COMPATIBLE)
      STOP
   15 BC2(1,1) = 1.
      BC1(5,4) = 1.
      BC1(6,2) = 1.
      BC1(7,3) = 1.
      DO 4 I = 1,3
      BC1(I+1,I+3) = 1.
    4 BC2(I+7,I+3) = 1.0
C
    5 CONTINUE
      IF (I1ST .EQ. 1) GO TO 10
C
      CALL READ (CVEC, NRC, NCC, 6, 15)
      BC1(6,4) = -CVEC(3,12)
      BC1(6,6) =  CVEC(1,12)
      BC1(7,4) =  CVEC(2,12)
      BC1(7,5) = -CVEC(1,12)
      IF (MINIC .EQ. 0) GO TO 8
      DO 7 I=1,3
      Y(NDELCS   +I) = CVEC(I   ,1)*ANUM
```

```
    7 Y(NDELCS+18+I) = CVEC(I+3,1)*ANUM
C
C   COMPUTE CONSTANTS FOR THE CMG SYSTEM.
C
    8 OMF4X = OMFX**4
      OMF4Y = OMFY**4
      OMF4Z = OMFZ**4
      OMFZ1X = 4.*ZETFX*OMFX
      OMFZ1Y = 4.*ZETFY*OMFY
      OMFZ1Z = 4.*ZETFZ*OMFZ
      OMFZ2X = (4.*ZETFX**2 + 2.)*OMFX**2
      OMFZ2Y = (4.*ZETFY**2 + 2.)*OMFY**2
      OMFZ2Z = (4.*ZETFZ**2 + 2.)*OMFZ**2
      OMFZ3X = OMFZ1X*OMFX**2
      OMFZ3Y = OMFZ1Y*OMFY**2
      OMFZ3Z = OMFZ1Z*OMFZ**2
   10 IIST = 1
C
C
C   COMPUTE CONTROL SIGNALS AND FORCES FOR THE CMG SYSTEM.
C
      DLT = T-(TSTAR-TLAG)
      IF (MODE(4) .EQ. 1) TEST = 1.
      IF (MODE(8) .EQ. 1  .AND.  DLT .GE. 0.4) TEST = 0.
      FORCS(1) = FXCSM*TEST
      YDT(NDELCS) = FORCS(1)
C
      OMGG(1) = Y(NUT + 3)
      OMGG(2) = Y(NUT + 4)
      OMGG(3) = Y(NUT + 5)
      CALL MULTAD (BC3(2,1),Y(NXITDT),OMGG,3,NMODET,1,10,150,3)
      SNP = SIN(Y(NDELCS + 1))
      CSP = COS(Y(NDELCS + 1))
      SNT = SIN(Y(NDELCS + 2))
      CST = COS(Y(NDELCS + 2))
      YDT(NDELCS + 1) = OMGG(1)+OMGG(2)*SNP*SNT/CST
     *                          +OMGG(3)*CSP*SNT/CST
      YDT(NDELCS + 2) = OMGG(2)*CSP - OMGG(3)*SNP
      YDT(NDELCS + 3) = OMGG(3)
      YDT(NDELCS + 4) = OMF4X*(AKRX*OMGG(1)+AKDX*Y(NDELCS+1)
     *    -Y(NDELCS+ 7))- OMFZ1X*Y(NDELCS+ 4)
     *    -OMFZ2X*Y(NDELCS+ 5)-OMFZ3X*Y(NDELCS+ 6)
      YDT(NDELCS + 5) = Y(NDELCS + 4)
      YDT(NDELCS + 6) = Y(NDELCS + 5)
      YDT(NDELCS + 7) = Y(NDELCS + 6)
      YDT(NDELCS + 8) = (AX*Y(NDELCS + 7) - Y(NDELCS + 8))/BX
      YDT(NDELCS + 9) = OMF4Y*(AKRY*OMGG(2)+AKDY*Y(NDELCS+2)
     *    -Y(NDELCS+12))- OMFZ1Y*Y(NDELCS+ 9)
     *    -OMFZ2Y*Y(NDELCS+10)-OMFZ3Y*Y(NDELCS+11)
      YDT(NDELCS +10) = Y(NDELCS + 9)
      YDT(NDELCS +11) = Y(NDELCS +10)
      YDT(NDELCS +12) = Y(NDELCS +11)
      YDT(NDELCS +13) = (AY*Y(NDELCS +12) - Y(NDELCS +13))/BY
      YDT(NDELCS +14) = OMF4Z*(AKRZ*OMGG(3)+AKDZ*Y(NDELCS+3)
     *    -Y(NDELCS+17))- OMFZ1Z*Y(NDELCS+14)
     *    -OMFZ2Z*Y(NDELCS+15)-OMFZ3Z*Y(NDELCS+16)
      YDT(NDELCS +15) = Y(NDELCS +14)
      YDT(NDELCS +16) = Y(NDELCS +15)
      YDT(NDELCS +17) = Y(NDELCS +16)
      YDT(NDELCS +18) = (AZ*Y(NDELCS +17) - Y(NDELCS +18))/BZ
C
      FORCS(2) = -Y(NDELCS +  8)
```

```
      FORCS(3) = -Y(NDELCS + 13)
      FORCS(4) = -Y(NDELCS + 18)
      IF (FORCS(2) .LT. -TLIM) FORCS(2) =  -TLIM
      IF (FORCS(2) .GT.  TLIM) FORCS(2) =   TLIM
      IF (FORCS(3) .LT. -TLIM) FORCS(3) =  -TLIM
      IF (FORCS(3) .GT.  TLIM) FORCS(3) =   TLIM
      IF (FORCS(4) .LT. -TLIM) FORCS(4) =  -TLIM
      IF (FORCS(4) .GT.  TLIM) FORCS(4) =   TLIM
C
C
C   CALCULATE CONTROL SIGNALS (INCLUDING SIGNALS TO BE INTEGRATED) AND
C   FORCES FOR TACS AND RCS.
C
      DO 17 I=1,3
      EULR(I   ) = Y(NDELCS+I    )
      EULR(I+3) = Y(NDELCS+I+18)
      EDOT(I) = OMGG(I)
   17 EDOT(I+3) = Y(NUC+I+2)
      SNP = SIN(Y(NDELCS + 19))
      CSP = COS(Y(NDELCS + 19))
      SNT = SIN(Y(NDELCS + 20))
      CST = COS(Y(NDELCS + 20))
      YDT(NDELCS + 19) = EDOT(4)+EDOT(5)*SNP*SNT/CST
     *                          +EDOT(6)*CSP*SNT/CST
      YDT(NDELCS + 20) = EDOT(5)*CSP - EDOT(6)*SNP
      YDT(NDELCS + 21) = EDOT(5)*SNP/CST + EDOT(6)*CSP/CST
C
      DO 50 I = 1,6
   50 CALL CSLOOP (I)
C
      STOR = QPA(2)
      QPA(2) = QPA(3)
      QPA(3) = STOR
      DO 65 I=1,6
      FORCS(I + 4) = -QPA(I)
   65 YDT(NDELCS + 27 + I) = ABS(QPA(I))
C     .
      RETURN
      END
```

```
      SUBROUTINE CONTAK
C
C   SUBROUTINE TO SET CONSTRAINT MODE BASED ON INPUT GEOMETRY ONLY.
C   THE SUBROUTINE IS CALLED INITIALLY AND ONLY ON THE 4TH TIME OF THE
C   RUNGE-KUTTA-GILL LOOP.
C
C   SUBROUTINES THAT ARE CALLED ARE --- BAKOFF,GEOM AND KINMAT.
C
C   CODED BY CARL BODLEY, JULY 1969
C
      COMMON /LIPARM/ FLAMC(3),QRMAG(3)
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
C
      DIMENSION IFTEST(8)
C
      DATA NIT,NOT/5,6/
      DATA   EPSU,  IFCAP,  MAXITR,  I1ST /
     *      0.01,       0,       15,   0/
C
      IF (I1ST .GT. 0) GO TO 5
      DO 8 I=1,8
    8 MODE(I) = 0
      I1ST = 1
    5 ICOUNT = 0
  100 DO 10 I=1,8
      IFTEST(I) = 0
      IF (PHICON(I) .LE. EPSU) MODE(I) = 1
      IF (PHICON(I) .LT. 0.) IFTEST(I) = 1
   10 CONTINUE
C
      IF (IFCAP .EQ. 1) MODE(8) = 1
      IF (MODE(8) .EQ. 1) MODE(4) = 0
      IF (MODE(8) .EQ. 1) IFCAP = 1
      MODE( 9) = MODE(8)
      MODE(10) = MODE(8)
C
      DO 15 I=1,3
      IP4 = I + 4
      IF (FLAMC(I) .LE. 0.0) MODE(IP4) = 0
      IF (MODE(I) .EQ. 1 .AND. MODE(IP4) .EQ. 1) GO TO 20
      GO TO 15
   20 IF (PHICON(I) .LE. PHICON(IP4)) MODE(IP4) = 0
      IF (PHICON(I) .GT. PHICON(IP4)) MODE(I   ) = 0
   15 CONTINUE
C
      IF (MODE(8) .EQ. 1 .AND. PHICON(8) .GE. EPSU) IFTEST(8) = 1
      PHCN9  = ABS(PHICON(9 ))
      PHCN10 = ABS(PHICON(10))
      IF (MODE(8) .EQ. 1 .AND. PHCN9  .GE. EPSU) IFTEST(8) = 1
      IF (MODE(8) .EQ. 1 .AND. PHCN10 .GE. EPSU) IFTEST(8) = 1
C
      IFOUT = 0
      DO 25 I=1,8
   25 IFOUT = IFOUT + IFTEST(I)*MODE(I)
      IF (IFOUT .EQ. 0 .AND. ICOUNT .GT. 1) RETURN
C
      ICOUNT = ICOUNT + 1
      IF (ICOUNT .EQ. MAXITR) GO TO 999
C
      CALL BAKOFF
      CALL GEOM
      CALL KINMAT
      GO TO 100
C
  999 WRITE (NOT,1001)
 1001 FORMAT (31H1SUBR. CONTAK DOES NOT CONVERGE/
     *         17H0PROGRAM STOPPED.)
      STOP
      END
```

```
      SUBROUTINE CSLOOP (MCHNL)
C
C   SUBROUTINE CSLOOP COMPUTES THE SIGNALS AT NECESSARY POINTS IN THE
C   CONTROL LOOPS FOR ALL CHANNELS.
C
C  VARIABLE DEFINITION (PER CHANNEL)
C   D1 THRU D5 - SIGNALS WITHIN THE CONTROL SYSTEM.
C   ANGLE - ANGULAR ORIENTATION (RADIANS)
C   TFL - CONTROL SYSTEM FIRST ORDER TIME CONSTANT.
C   A0, A1, A2, C01, CFL - CONTROL SYSTEM GAIN CONSTANTS.
C   ANGVEL - ANGULAR RATE OF CHANGE OF BODY (RAD/SEC).
C   ANGLMT - ANGULAR ORIENTATION ERROR LIMIT OF BODY (INPUT IN DEGREES).
C   QP - OUTPUT FORCE REQUEST SIGNAL TO MINIMUM PULSING DEVICE.
C   D4D - DERIVATIVE OF SIGNAL D4.
C
C
      COMMON /CSYSTM/ CVEC(6,15), EULR(6), EDOT(6)
      COMMON /MISCNO/ ANUM
      COMMON /PULSES/ QPA(6)
      COMMON /VECTOR/ Y(150), YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION D5(6)
      DATA D5 /0.,0.,0.,0.,0.,0. /
C
      D4 = Y(NDELCS + MCHNL + 21)
C
C   INITIALIZE CONTROL SYSTEM VARIABLES.
C
    5 ANGLMT  = CVEC(MCHNL,2) * ANUM
      ECLMI   = CVEC(MCHNL,3)
      ECLMO   = CVEC(MCHNL,4)
      FTRSTR  = CVEC(MCHNL,5)
      A0      = CVEC(MCHNL,6)
      A1      = CVEC(MCHNL,7)
      A2      = CVEC(MCHNL,8)/ANUM
      C01     = CVEC(MCHNL,9)
      CFL     = CVEC(MCHNL,10)
      TFL     = CVEC(MCHNL,11)
C
      ANGLE = EULR(MCHNL)
      ANGVEL = EDOT(MCHNL)
C
      IF (ANGLE .GT.  ANGLMT) ANGLE =  ANGLMT
      IF (ANGLE .LT. -ANGLMT) ANGLE = -ANGLMT
      D1 = (A1 * ANGVEL + A0 * ANGLE) * A2
C
C   COMPUTE THE COUPLING TERMS IN CASE OF THE YAW AND ROLL CHANNELS.
C
      IF (MCHNL .EQ. 2 .OR. MCHNL .EQ. 5) GO TO 20
      IF (MCHNL .EQ. 1 .OR. MCHNL .EQ. 4) MALT = MCHNL + 2
      IF (MCHNL .EQ. 3 .OR. MCHNL .EQ. 6) MALT = MCHNL - 2
      ANGLMT  = CVEC(MALT,2) * ANUM
      A0      = CVEC(MALT,6)
      A1      = CVEC(MALT,7)
      A2      = CVEC(MALT,8) /ANUM
      ANGLE = EULR(MALT)
      ANGVEL = EDOT(MALT)
      IF (ANGLE .GT.  ANGLMT) ANGLE =  ANGLMT
      IF (ANGLE .LT. -ANGLMT) ANGLE = -ANGLMT
      DALT = (A1 * ANGVEL + A0 * ANGLE) * A2
```

```
      IF (MCHNL .EQ. 1 .OR. MCHNL .EQ. 4) D1 = D1 + DALT
      IF (MCHNL .EQ. 3 .OR. MCHNL .EQ. 6) D1 = D1 - DALT
C
C   COMPUTE REMAINING SIGNALS IN THE CONTROL LOOP.
C
   20 D2 = D1 - D4 + D5(MCHNL)
      D3 = 0.0
      IF (D2 .GE.  ECLMI) D3 =  ECLMO
      IF (D2 .LE. -ECLMI) D3 = -ECLMO
      D5(MCHNL) = C01 * D3
      D4D = (CFL * D3 - D4) / TFL
      QP = FTRSTR*D3 / ECLMO
      QPA(MCHNL) = QP
      YDT(NDELCS + MCHNL + 21) = D4D
C
      RETURN
      END
```

```
      PROGRAM DOCKOO (INPUT,OUTPUT,FILMPL,TAPE5=INPUT,TAPE6=OUTPUT,
     *           PUNCH,TAPE1,TAPE2,TAPE3)
C
C                MAIN OVERLAY OF PROGRAM DOCKEL
C
C          CODED BY BODLEY/NERZ/PARK    1969
C
      COMMON /ANGLES/ ALPHAD,SALPD,CALPD,TALPD, SSIGMA(3),CSIGMA(3)
      COMMON /BC1234/ BC1(10,6), BC2(10,6), BC3(10,26), BC4(10,26)
      COMMON /BINDFR/ FI,BINDMU
      COMMON /BTDATA/ TRHO(9,6), BLS(10,6), BLV(7,39)
      COMMON /CHANGE/ Z(150),ZDT(150)
      COMMON /CMGDTA/ FXCSM, AKRX,AKRY,AKRZ,AKDX,AKDY,AKDZ,OMFX,OMFY,
     *                     OMFZ,ZETFX,ZETFY,ZETFZ,AX,AY,AZ,BX,BY,BZ,
     *                     TLIM
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /CSYSTM/ CVEC(6,15), EULR(6), EDOT(6)
      COMMON /FORMOM/ FANDM(12)
      COMMON /FUNGAM/ SGAMMA(3),CGAMMA(3),SGSS(3),SGCS(3),
     *                                   CGSS(3),CGCS(3)
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTO(3),CPCD(3),RADCON,CSPHER,ROTALP,PEO,FDO
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /JBATCH/ JBA
      COMMON /JCOUNT/ JIL
      COMMON /LATCHL/ VLAM(6)
      COMMON /LIPARM/ FLAMC(3), QRMAG(3)
      COMMON /LOCLEN/ LOC(12), LEN(12)
      COMMON /LOCPTC/ CSC(3), CAC(3,3),CQC(3,3),
     *                CPC(3),PSC(3),PAC(3,3),PQC(3,3)
      COMMON /LOCPTD/ DSD(3),DHD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /LOCPTH/  DSH(3)
      COMMON /LOCPTI/ OTI(3), OCI(3), TCI(3)
      COMMON /LOCPTP/ PSP(3),PAP(3,3),PQP(3,3)
      COMMON /LOCPTT/ THT(3),TCT(3),TST(3),TAT(3,3),TRT(3,3),
     *                CST(3),CAT(3,3),   QRT(3,3), TDT(3),
     *                DHT(3),DGT(3),DST(3), DAT(3,3),DQT(3,3),DRT(3,3)
      COMMON /LSTART/ RUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
      COMMON /MASS/  AMT, AMC, AIT(3,3), AIC(3,3)
      COMMON /MFREQD/ FREQT(26), FREQC(26), ZETAT, ZETAC
      COMMON /MISCNO/ ANUM
      COMMON /MODATA/ HATD(3,26),SIGATD(3,26), HATP(3,26),SIGATP(3,26)
      COMMON /NCNTRL/ NCNSYS
      COMMON /NEQTNS/ NEQTN
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PERTIT/ PTITL(8),IFFIXD,RANGLE,EED,CANGLE,
     *                COELOC(3),VPLOC(3)
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /PLTR  / K1,NCOLS,NTIMES,XDLTA,NPLTS
      COMMON /PULSES/ QPA(6)
      COMMON /PUNCHY/ IFPNCH
      COMMON /QPRKTA/ QRK(150), PRK(4)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FREY
      COMMON /SKEWMS/ TDTSQ(3,3),THTSQ(3,3),TRTSQ(3,3,3), DRTSQ(3,3,3),
     *                PSCSQ(3,3),PSPPSQ(3,3),PQCSQ(3,3,3),PQPPSQ(3,3,3),
     *                CSCSQ(3,3),CQCSQ(3,3,3),  DHTSQ(3,3)
      COMMON /SLIDFR/ HDMU, ARMMU, BFV(8,39)
      COMMON /SPRDMP/ FSPRNG(2,20,4),FDAMP(2,20,4)
       COMMON /TAPED/ NTAPE1, NTAPE2, NTAPE3
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
```

```
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECMAG/ FDMAG(3), PEMAG(3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DATA NTAPE1, NTAPE2, NTAPE3, JBA,  DOCKEL/
     *          1,      2,      3,   0,6LDOCKEL/
      DATA  NUT, NXT, NGAMT, NUC, NXC, NGAMC, NRHO, NXITDT
     1    /   1,   7,    10,  16,  22,    25,   31,     37     /
      DATA   K1 /30000/
C
C ----------------------------------------------------------------------------
C
C                      PROGRAM DOCKEL
C                         COMMON MAP
C
C
C
C                                          SUBROUTINE
C
C                       B C C C D F F G H H K L L M P P P R R R S S S S T V
C                       A O O S O O T E L S I A O I I L R E O U F E E E I E
C                       K N N L C R A O A I N M C N E O E T O N T T T T M C
C                       O S T O K F B M T G M R A Y R T E R T K P P T U H T
C                       F Y A O D U L   C   A H T S C E N A S T E L P P S R
C                       F S K P O N E   H   T O E   E R T K   A R T E   T
C
C   COMMON
C-------------------------------------------------------------------------
C
C   ANGLES                      X     X     X                   X     X
C   BC1234                X     X                                         X
C   BINDFR                      X             X         X         X   X
C   BTDATA              X       X X         X X                           X
C   CHANGE                      X                       X         X
C   CMGDTA                X     X
C   CSFORC                X     X                                 X X X   X
C   CSYSTM                X   X X
C   FORMOM                      X X                     X         X X
C   FUNGAM                      X     X     X                   X
C   GEOMTY                      X     X     X X   X             X     X
C   IFOUTS                X     X             X         X       X     X
C   JBATCH                X     X                               X     X
C   JCOUNT                      X     X X     X               X       X   X
C   LATCHL                      X       X                         X X
C   LIPARM                  X   X     X     X
C   LOCLEN                      X               X       X
C   LOCPTC                      X     X X   X
C   LOCPTD                      X     X     X           X       X X
C   LOCPTH                      X     X     X
C   LOCPTI                      X     X                         X
C   LOCPTP                      X     X     X     X             X
C   LOCPTT                      X     X X   X                   X
C   LSTART                      X
C   MASS                        X       X                             X   X
C   MFREQD                      X         X                           X   X
C   MISCNO                X   X X     X       X   X     X       X     X   X
C   MODATA                      X     X X X X
```

```
C   NCNTRL               X               X                     X
C   NEQTNS         X     X               X X     X     X       X   X
C   NMODES         X     X     X X X X X X                 X   X   X
C   PERTIT               X                               X
C   PHILAM       X X X   X X         X X         X X       X       X
C   PLTR                 X                     X               X
C   PULSES         X   X X
C   PUNCHY               X                       X             X
C   QPRKTA               X                             X       X
C   RETRCT         X     X             X         X X       X   X
C   SKEWMS               X X         X
C   SLIDFR               X           X                         X   X
C   SPRDMP               X             X                       X
C   TAPED                X                     X           X X X
C   TIMESS         X     X X   X     X X       X X X   X   X X X X X
C   TRNFMS       X       X X   X X   X X                 X         X
C   VECMAG               X           X X
C   VECTOR       X X   X X     X X   X X   X     X X   X X     X   X
C   YPSTNS       X X   X X     X X   X X X X     X X     X X       X
C
C
C ----------------------------------------------------------------------
C
C   THE FOLLOWING REQUIRED SUBROUTINES HAVE NO COMMON BLOCKS
C
C         ATXBV
C         AVPBV
C         AXB3
C         AXBT3
C         AXBV
C         INV2NP
C         MULT3
C         MULTAD
C         SKEWV
C
C   THE FOLLOWING REQUIRED SUBROUTINES ARE FROM THE FORMA LIBRARY
C
C         AABB
C         MULT
C         PAGEHD
C         PLOT1
C         PLOT3
C         PLOTSS
C         PUNCH
C         READ
C         READIM
C         REVISE
C         START
C         VCROSS
C         VDOT
C         WRITE
C         WRITIM
C         ZERO
C
C -----------------------------------------------------------------------------
C
C
      CALL BPLT(0,2HLC)
      CALL INIT280
C
    1 CALL START
C
      JBA = JBA + 1
C
      CALL OVERLAY (DOCKEL,1,0)
      CALL OVERLAY (DOCKEL,2,0)
      IF (IFPLOT .NE. 0) CALL OVERLAY (DOCKEL,3,0)
C
      GO TO 1
C
      END
```

```
      SUBROUTINE FTABLE(X,Y,SLOPE,BINTR,TABLE,POINT,NDIR,MM)
C
      DIMENSION TABLE(2,1)
C
      II = 1
      JJ = 2
      NPOINT = POINT + 0.1
      NTABLE = NPOINT + 1
      IF (NDIR .EQ. 1) GO TO 1
      II = 2
      JJ = 1
    1 IF (X .GE. TABLE(II,2)) GO TO 5
      MM = 2
      GO TO 25
    5 IF (X .LE. TABLE(II,NTABLE)) GO TO 10
      MM = NTABLE - 1
      GO TO 25
   10 DO 15 KK = 3,NTABLE
      IN = KK
      IF (X .LE. TABLE(II,KK)) GO TO 20
   15 CONTINUE
   20 MM = IN - 1
   25 LL = MM + 1
      SLOPE = (TABLE(JJ,LL) - TABLE(JJ,MM)) / (TABLE(II,LL) - TABLE
     * (II,MM))
      Y = TABLE(JJ,MM) + SLOPE * (X - TABLE(II,MM))
      BINTR = Y - X*SLOPE
C
      RETURN
      END
```

```
      SUBROUTINE FORFUN
C
C           SUBROUTINE TO COMPUTE FORCING FUNCTION
C
C              CODED BY A C PARK    SEPT 1969
C
      COMMON /BTDATA/ TRHO(9,6), BLS(10,6), BLV(7,39)
      COMMON /FORMOM/ FANDM(12)
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /SKEWMS/ TDTSQ(3,3),THTSQ(3,3),TRTSQ(3,3,3), DRTSQ(3,3,3),
     *                PSCSQ(3,3),PSPPSQ(3,3),PQCSQ(3,3,3),PQPPSQ(3,3,3),
     *                CSCSQ(3,3),CQCSQ(3,3,3),  DHTSQ(3,3)
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
C
      DIMENSION TLDS(7,12)
C
      IF (TMST .EQ. 0.) CALL ZERO (TLDS(5,4),3,3,7)
C
      DO 12 I=1,7
      DO 12 J=1,3
      J6 = J+6
      TLDS(I,J ) = BLV(I,J )
   12 TLDS(I,J6) = BLV(I,J6)
C
      CALL MULT3 (TRHT(2,1),DHTSQ,TLDS(1, 4),1,3,3,3,3,7)
      CALL MULT3 (TRHC(2,1),PSCSQ,TLDS(1,10),1,3,3,3,3,7)
C
      DO 18 K=1,3
      K1 = K+1
      CALL MULT3 (TRRT(2,1,K),DRTSQ(1,1,K),TLDS(K1, 4),1,3,3,3,3,7)
      CALL MULT3 (TRRC(2,1,K),PQCSQ(1,1,K),TLDS(K1,10),1,3,3,3,3,7)
   18 CONTINUE
C
      DO 20 I= 1, 4
      DO 20 J=10,12
   20 TLDS(I,J) = -TLDS(I,J)
C
      CALL MULT3 (TRTC,PSCSQ,TLDS(5,10),3,3,3,3,3,7)
C
      CALL MULT (ALAM(4),TLDS,FANDM,1,7,12,1,7)
C
      RETURN
      END
```

```
      SUBROUTINE GEOM
C
C                COMPUTES GEOMETRY AND ROTATION TRANSFORMATIONS
C
      COMMON /ANGLES/ ALPHAD,SALPD,CALPD,TALPD, SSIGMA(3),CSIGMA(3)
      COMMON /FUNGAM/ SGAMMA(3),CGAMMA(3),SGSS(3),SGCS(3),
     *                                   CGSS(3),CGCS(3)
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTD(3),CPCD(3),RADCON,CSPHER,ROTALP,PED,FDD
      COMMON /JCOUNT/ JIL
      COMMON /LIPARM/ FLAMC(3), QRMAG(3)
      COMMON /LOCPTC/ CSC(3), CAC(3,3),CQC(3,3),
     *                CPC(3),PSC(3),PAC(3,3),PQC(3,3)
      COMMON /LOCPTD/ DSD(3),DHD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /LOCPTH/  DSH(3)
      COMMON /LOCPTI/ OTI(3), OCI(3), TCI(3)
      COMMON /LOCPTP/ PSP(3),PAP(3,3),PQP(3,3)
      COMMON /LOCPTT/ THT(3),TCT(3),TST(3),TAT(3,3),TRT(3,3),
     *                CST(3),CAT(3,3),   QRT(3,3), TDT(3),
     *                DHT(3),DGT(3),DST(3), DAT(3,3),DQT(3,3),DRT(3,3)
      COMMON /MISCNO/ ANUM
      COMMON /MODATA/ HATD(3,26),SIGATD(3,26), HATP(3,26),SIGATP(3,26)
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION UCLP(3),UCLD(3),UCMD(3),UCND(3),SKEWLD(3,3)
      DIMENSION EAP(3),EAD(3)
      DIMENSION WV(3),VW(3)
C
      IF (TMST .GT. 0.0) GO TO 100
C
      PSP(2) = 0.0
      PSP(3) = 0.0
C
      TRHD(1,1) = CALPD
      TRHD(2,1) =-SALPD
      TRHD(3,1) = 0.0
C
      DO 12 J=1,3
      PAP(2,J) = -AA*SSIGMA(J)
   12 PAP(3,J) =  AA*CSIGMA(J)
C
  100 CONTINUE
C
      CALL MULT (SIGATP,Y(NXIC),EAP,3,NMODEC,1,3,150)
      EAP(2) = EAP(2)+Y(NRHO+1)
      EAP(3) = EAP(3)+Y(NRHO+2)
      CALL SKEWV (EAP,TRPC)
      CALL MULT (SIGATD,Y(NXIT),EAD,3,NMODET,1,3,150)
      CALL SKEWV (EAD,TRDT)
      DO 11 I=1,3
      TRPC(I,I) = 1.0
   11 TRDT(I,I) = 1.0
```

```
C
      DO 6 I=1,2
      IT = 3*I+NGAMT-4
      IC = 3*I+NGAMC-4
      DO 6 J=1,3
      TRTI(I,J) = Y(IT+J)
    6 TRCI(I,J) = Y(IC+J)
      TRTI(3,1) = TRTI(1,2)*TRTI(2,3)-TRTI(2,2)*TRTI(1,3)
      TRTI(3,2) = TRTI(1,3)*TRTI(2,1)-TRTI(1,1)*TRTI(2,3)
      TRTI(3,3) = TRTI(1,1)*TRTI(2,2)-TRTI(2,1)*TRTI(1,2)
      TRCI(3,1) = TRCI(1,2)*TRCI(2,3)-TRCI(2,2)*TRCI(1,3)
      TRCI(3,2) = TRCI(1,3)*TRCI(2,1)-TRCI(1,1)*TRCI(2,3)
      TRCI(3,3) = TRCI(1,1)*TRCI(2,2)-TRCI(2,1)*TRCI(1,2)
      FNRM1T  = SQRT(TRTI(1,1)*TRTI(1,1)+TRTI(1,2)*TRTI(1,2)
     *             + TRTI(1,3)*TRTI(1,3))
      FNRM3T  = SQRT(TRTI(3,1)*TRTI(3,1)+TRTI(3,2)*TRTI(3,2)
     *             + TRTI(3,3)*TRTI(3,3))
      FNRM1C  = SQRT(TRCI(1,1)*TRCI(1,1)+TRCI(1,2)*TRCI(1,2)
     *             + TRCI(1,3)*TRCI(1,3))
      FNRM3C  = SQRT(TRCI(3,1)*TRCI(3,1)+TRCI(3,2)*TRCI(3,2)
     *             + TRCI(3,3)*TRCI(3,3))
      DO 40 J=1,3
      TRTI(1,J) = TRTI(1,J) / FNRM1T
      TRTI(3,J) = TRTI(3,J) / FNRM3T
      TRCI(1,J) = TRCI(1,J) / FNRM1C
   40 TRCI(3,J) = TRCI(3,J) / FNRM3C
      TRTI(2,1) = TRTI(3,2)*TRTI(1,3) - TRTI(3,3)*TRTI(1,2)
      TRTI(2,2) = TRTI(3,3)*TRTI(1,1) - TRTI(3,1)*TRTI(1,3)
      TRTI(2,3) = TRTI(3,1)*TRTI(1,2) - TRTI(3,2)*TRTI(1,1)
      TRCI(2,1) = TRCI(3,2)*TRCI(1,3) - TRCI(3,3)*TRCI(1,2)
      TRCI(2,2) = TRCI(3,3)*TRCI(1,1) - TRCI(3,1)*TRCI(1,3)
      TRCI(2,3) = TRCI(3,1)*TRCI(1,2) - TRCI(3,2)*TRCI(1,1)
      DO 45 I=1,2
      IT = 3*I + NGAMT - 4
      IC = 3*I + NGAMC - 4
      DO 45 J=1,3
      Y(IT+J) = TRTI(I,J)
   45 Y(IC+J) = TRCI(I,J)
C
      DO 14 I=1,3
      INXT = I + NXT -1
      INXC = I + NXC -1
      OTI(I) =  Y(INXT)
   14 OCI(I) =  Y(INXC)
C
      DO 15 I=1,3
      II = I+NRHO+2
      SGAMMA(I) = SIN( Y(II))
      CGAMMA(I) = COS( Y(II))
      SGSS(I) = SGAMMA(I)*SSIGMA(I)
      SGCS(I) = SGAMMA(I)*CSIGMA(I)
      CGSS(I) = CGAMMA(I)*SSIGMA(I)
   15 CGCS(I) = CGAMMA(I)*CSIGMA(I)
C
      EXTN = DLI +  Y(NRHO)
      PSP(1) = EXTN + DLS
C
      CALL MULT (HATP,Y(NXIC),WV,3,NMODEC,1,3,150)
      CALL AVPBV (1.0,CPCO,1.0,WV,CPC)
      CALL MULT (HATD,Y(NXIT),WV,3,NMODET,1,3,150)
      CALL AVPBV (1.0,TDTO,1.0,WV,TDT)
      CALL ATXBV (TRPC,PSP,PSC)
```

```
C
      CALL AVPBV (1.0,CPC,1.0,PSC,CSC)
C
      DO 18 J=1,3
      PAP(1,J) = EXTN
      CALL ATXBV (TRPC,PAP(1,J),PAC(1,J))
   18 CONTINUE
C
      CALL AXBT3 (TRTI,TRCI,TRTC)
      CALL AXBT3 (TRPC,TRTC,TRPT)
      CALL AXBT3 (TRPT,TRDT,TRPD)
      CALL AVPBV (1.0,OCI,-1.0,OTI,TCI)
      CALL AXBV  (TRTI,TCI,TCT)
      CALL AXBV  (TRTC,CSC,CST)
      CALL AVPBV (1.0,TCT, 1.C,CST,TST)
      CALL AVPBV (1.0,TST,-1.0,TDT,DST)
      CALL AXBV (TRDT,DST,DSD)
C
C
      PHIH = 0.0
      IF (DSD(2).EQ.0.0  .AND.  DSD(3).EQ.0.0)  GO TO 21
      PHIH = ATAN2(DSD(3), DSD(2))
   21 CONTINUE
      SPHIH = SIN(PHIH)
      CPHIH = COS(PHIH)
C
      TRHD(1,2) = CPHIH*SALPD
      TRHD(1,3) = SPHIH*SALPD
      TRHD(2,2) = CPHIH*CALPD
      TRHD(2,3) = SPHIH*CALPD
      TRHD(3,2) =-SPHIH
      TRHD(3,3) = CPHIH
      CALL AXBV  (TRHD,DSD,DSH)
      BIGR = DSH(1)
C
      CALL AXB3 (TRHD,TRDT,TRHT)
      CALL AXB3  (TRHT,TRTC,TRHC)
C
      DO 22 I=1,3
      DHD(I) =     BIGR*TRHD(1,I)
   22 DGD(I) = DSD(I) + CSPHER*TRHD(2,I)
C
      CALL ATXBV (TRDT,DHD,DHT)
      CALL ATXBV (TRDT,DGD,DGT)
      CALL AVPBV (1.0,TDT,1.0,DHT,THT)
      CALL AXBT3 (TRHC,TRPC,TRHP)
      CALL AXBT3 (TRTC,TRPC,TRTP)
C
      DO 24 K=1,3
C
      UCLP(1) = -CGAMMA(K)
      UCLP(2) = -SGSS(K)
      UCLP(3) =  SGCS(K)
      CALL ATXBV (TRPD,UCLP,UCLD)
      CALL SKEWV (UCLD,SKEWLD)
      CALL AVPBV (1.0,CPC, 1.0,PAC(1,K),CAC(1,K))
      CALL AXBV  (TRTC,CAC(1,K),CAT(1,K))
      CALL AVPBV (1.0,TCT, 1.0,CAT(1,K),TAT(1,K))
      CALL AVPBV (1.0,TAT(1,K),-1.0,TDT,DAT(1,K))
      CALL AXBV  (TRDT,DAT(1,K),DAD(1,K))
C
C     CHECK FOR PIERCE POINT
```

```
      CALL PIERCE (DAD(1,K),UCLD,DSD,RADCON,ALPHAD,SALPD,CALPD,TALPD,
     1               ROTALP,DLB)
C     IF NO PIERCE POINT, CONTINUE
      CALL ROOTS (DAD(1,K),UCLD,RADCON,ROTALP,PHIC,FLAMC(K),QRMAG(K),
     *              JIL,K)
C
      DRD(1,K) = ROTALP
      DRD(2,K) = RADCON*COS(PHIC)
      DRD(3,K) = RADCON*SIN(PHIC)
      CALL ATXBV (TRDT,DRD(1,K),DRT(1,K))
C
      CALL AVPBV (1.0,TDT,1.0,DRT(1,K),TRT(1,K))
      CALL AVPBV (1.0,PAP(1,K), FLAMC(K),UCLP,PQP(1,K))
      CALL ATXBV (TRPC,PQP(1,K),PQC(1,K))
      CALL AVPBV (1.0,CPC,1.0,PQC(1,K),CQC(1,K))
      CALL AVPBV (1.0,DAD(1,K),FLAMC(K),UCLD,DQD(1,K))
      CALL ATXBV (TRDT,DQD(1,K),DQT(1,K))
      CALL AVPBV (1.0,DRT(1,K),-1.0,DQT(1,K),QRT(1,K))
      CALL AVPBV (1.0,DRD(1,K),-1.0,DQD(1,K),QRD(1,K))
C
      DO 26 I=1,3
   26 UCMD(I) = QRD(I,K)/QRMAG(K)
C
      CALL AXBV  (SKEWLD,UCMD,UCND)
C
      DO 28 J=1,3
      TRRD(1,J,K) = UCLD(J)
      TRRD(2,J,K) = UCMD(J)
   28 TRRD(3,J,K) =-UCND(J)
      CALL AXB3  (TRRD(1,1,K),TRDT,TRRT(1,1,K))
      CALL AXB3  (TRRT(1,1,K),TRTC,TRRC(1,1,K))
      CALL AXBT3 (TRRC(1,1,K),TRPC,TRRP(1,1,K))
C
   24 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE HLATCH(L)
C
C     SUBROUTINE TO APPLY HARD-LATCH CONSTRAINT FORCES AT DOCKING
C     COLLAR, MODIFIES STATE VECTOR(S), Y, YDT.
C     CODED BY CARL BODLEY  JAN. 2, 1970
C
      COMMON /JCOUNT/ JIL
      COMMON /LATCHL/ VLAM(6)
      COMMON /LOCPTC/ CSC(3), CAC(3,3),CQC(3,3),
     *                CPC(3),PSC(3),PAC(3,3),PQC(3,3)
      COMMON /LOCPTT/ THT(3),TCT(3),TST(3),TAT(3,3),TRT(3,3),
     *                CST(3),CAT(3,3),   QRT(3,3), TDT(3),
     *                DHT(3),DGT(3),DST(3), DAT(3,3),DQT(3,3),DRT(3,3)
      COMMON /MASS/  AMT, AMC, AIT(3,3), AIC(3,3)
      COMMON /MODATA/ HATD(3,26),SIGATD(3,26), HATP(3,26),SIGATP(3,26)
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION BA1(6,6), BA2(6,6), BA3(6,26), BA4(6,26),
     *   DFT(3),PFC(3),TFT(3),CFC(3),  CFT(3),
     *   DFTSQ(3,3),PFCSQ(3,3),TFTSQ(3,3),CFCSQ(3,3), W63(6,3),
     *   AITI(3,3),AICI(3,3), VFLAM(6),
     *   CUT(6,6),CUC(6,6),CXT(26,6),CXC(26,6),CLAM(6,6),CLAMI(6,6),
     *   V1(6),V2(6),V3(6),V4(6),RV(6),VEC1(3),VEC2(3),VEC3(3),VEC4(3),
     *   TDDTR(3),CPDTR(3),OMDT(3),OMPC(3),OMGD(3),OMGP(3)
C
      DATA I1ST,DFD1 / 0, 18.1 /
C
      IF (I1ST .GT. 0) GO TO 5
      CALL ZERO    (BA1,6,6,6)
      CALL ZERO    (BA2,6,6,6)
      DO 7 I=1,6
    7 BA1(I,I) = 1.
      CALL INV2NP (AIT,AITI,3,3)
      CALL INV2NP (AIC,AICI,3,3)
      I1ST = 1
    5 CONTINUE
      GO TO (100,200), L
  100 DO 10 I=1,3
      DFT(I) = TRDT(1,I)*DFD1
      TFT(I) = TDT(I) + DFT(I)
   10 CFT(I) = TFT(I) - TCT(I)
      CALL MULT3  (CFT,TRTC,CFC,1,3,3,1,3,1)
      CALL AVPBV  (1.,CFC,-1.,CPC,PFC)
      CALL SKEWV  (DFT,DFTSQ)
      CALL SKEWV  (PFC,PFCSQ)
      CALL SKEWV  (TFT,TFTSQ)
      CALL SKEWV  (CFC,CFCSQ)
C
      DO 20 I=1,3
      I3 = I + 3
      DO 20 J=1,3
      J3 = J + 3
      BA1( I,J3) = TFTSQ(I,J)
      BA2( I, J) =  TRTC(I,J)
   20 BA2(I3,J3) =  TRTC(I,J)
```

```
      CALL MULT3  (TRTC,CFCSQ,BA2(1,4),3,3,3,3,3,6)
      CALL MULT3  (TRTC,PFCSQ,W63,3,3,3,3,3,6)
      CALL MULT3  (TRTC,HATP   ,BA4      ,3,3,NMODEC,3,3,6)
      CALL MULT3  (TRTC,SIGATP,BA4(4,1),3,3,NMODEC,3,3,6)
      CALL MULTAD (W63 ,SIGATP,BA4      ,3,3,NMODEC,6,3,6)
C
      DO 30 I=1,3
      I3 = I + 3
      DO 30 J=1,NMODET
      BA3(I,J) = HATD   (I,J)
   30 BA3(I3,J) = SIGATD(I,J)
      CALL MULTAD (DFTSQ,SIGATD,BA3,3,3,NMODET,3,3,6)
C
      CALL MULT3  (BA1(1,4),AITI,W63,6,3,3,6,3,6)
      DO 40 I=1,6
      DO 40 J=1,3
      JP3 = J + 3
      CUT(J   ,I) = BA1(I,J)/AMT
   40 CUT(JP3,I) = W63(I,J)
      CALL MULT3  (BA2(1,4),AICI,W63,6,3,3,6,3,6)
      DO 50 I=1,6
      DO 50 J=1,3
      JP3 = J + 3
      CUC(J   ,I) = BA2(I,J)/AMC
   50 CUC(JP3,I) = W63(I,J)
      DO 45 I=1,6
      DO 45 J=1,NMODET
   45 CXT(J,I) = BA3(I,J)
      DO 55 I=1,6
      DO 55 J=1,NMODEC
   55 CXC(J,I) = BA4(I,J)
      CALL MULT3  (BA1,CUT,CLAM,6,      6,6,6, 6,6)
      CALL MULTAD (BA2,CUC,CLAM,6,      6,6,6, 6,6)
      CALL MULTAD (BA3,CXT,CLAM,6,NMODET,6,6,26,6)
      CALL MULTAD (BA4,CXC,CLAM,6,NMODEC,6,6,26,6)
      CALL INV2NP (CLAM,CLAMI,6,6)
C
      IF (JIL .NE. 4) RETURN
C
      CALL MULT3  (BA1,Y(NUT    ),V1,6,      6,1,6,150,6)
      CALL MULT3  (BA2,Y(NUC    ),V2,6,      6,1,6,150,6)
      CALL MULT3  (BA3,Y(NXITDT),V3,6,NMODET,1,6,150,6)
      CALL MULT3  (BA4,Y(NXICDT),V4,6,NMODEC,1,6,150,6)
      DO 60 I=1,6
   60 RV(I) = -V1(I) + V2(I) - V3(I) + V4(I)
      CALL MULT3  (CLAMI,RV,VFLAM,6,6,1,6,6,6)
      DO 65 I=1,6
   65 V1(I) = -VFLAM(I)
      CALL MULTAD (CUT,VFLAM,Y(NUT    ),      6,6,1, 6,6,150)
      CALL MULTAD (CUC,V1    ,Y(NUC    ),      6,6,1, 6,6,150)
      CALL MULTAD (CXT,VFLAM,Y(NXITDT),NMODET,6,1,26,6,150)
      CALL MULTAD (CXC,V1    ,Y(NXICDT),NMODEC,6,1,26,6,150)
      RETURN
C
C
  200 CALL MULT3  (HATD   ,Y(NXITDT),TDDTR,3,NMODET,1,3,150,3)
      CALL MULT3  (HATP   ,Y(NXICDT),CPDTR,3,NMODEC,1,3,150,3)
      CALL MULT3  (SIGATD,Y(NXITDT),OMDT  ,3,NMODET,1,3,150,3)
      CALL MULT3  (SIGATP,Y(NXICDT),OMPC  ,3,NMODEC,1,3,150,3)
      DO 70 I=1,3
      I1 = NUT + I + 2
      I2 = NUC + I + 2
```

```
      OMGD(I) = Y(I1) + OMDT(I)
   70 OMGP(I) = Y(I2) + OMPC(I)
C
      VEC1(1)=Y(NUT   )+Y(NUT+4)*TDT(3)-Y(NUT+5)*TDT(2)+2.*TDDTR(1)
      VEC1(2)=Y(NUT+1)+Y(NUT+5)*TDT(1)-Y(NUT+3)*TDT(3)+2.*TDDTR(2)
      VEC1(3)=Y(NUT+2)+Y(NUT+3)*TDT(2)-Y(NUT+4)*TDT(1)+2.*TDDTR(3)
      VEC2(1) =  OMGD(2)*DFT(3) - OMGD(3)*DFT(2)
      VEC2(2) =  OMGD(3)*DFT(1) - OMGD(1)*DFT(3)
      VEC2(3) =  OMGD(1)*DFT(2) - OMGD(2)*DFT(1)
      VEC3(1)=Y(NUC   )+Y(NUC+4)*CPC(3)-Y(NUC+5)*CPC(2)+2.*CPDTR(1)
      VEC3(2)=Y(NUC+1)+Y(NUC+5)*CPC(1)-Y(NUC+3)*CPC(3)+2.*CPDTR(2)
      VEC3(3)=Y(NUC+2)+Y(NUC+3)*CPC(2)-Y(NUC+4)*CPC(1)+2.*CPDTR(3)
      VEC4(1) =  OMGP(2)*PFC(3) - OMGP(3)*PFC(2)
      VEC4(2) =  OMGP(3)*PFC(1) - OMGP(1)*PFC(3)
      VEC4(3) =  OMGP(1)*PFC(2) - OMGP(2)*PFC(1)
C
      I = NUT + 3
      J = NUT + 4
      K = NUT + 5
      RV(1) = -Y(J)*VEC1(3)+Y(K)*VEC1(2)-OMGD(2)*VEC2(3)+OMGD(3)*VEC2(2)
      RV(2) = -Y(K)*VEC1(1)+Y(I)*VEC1(3)-OMGD(3)*VEC2(1)+OMGD(1)*VEC2(3)
      RV(3) = -Y(I)*VEC1(2)+Y(J)*VEC1(1)-OMGD(1)*VEC2(2)+OMGD(2)*VEC2(1)
      I = NUC + 3
      J = NUC + 4
      K = NUC + 5
      VEC1(1)=Y(J)*VEC3(3)-Y(K)*VEC3(2)+OMGP(2)*VEC4(3)-OMGP(3)*VEC4(2)
      VEC1(2)=Y(K)*VEC3(1)-Y(I)*VEC3(3)+OMGP(3)*VEC4(1)-OMGP(1)*VEC4(3)
      VEC1(3)=Y(I)*VEC3(2)-Y(J)*VEC3(1)+OMGP(1)*VEC4(2)-OMGP(2)*VEC4(1)
      CALL MULTAD (TRTC,VEC1,RV,3,3,1,3,3,6)
C
      RV(4) = 0.0
      RV(5) = 0.0
      RV(6) = 0.0
C
      CALL MULT3  (BA1,YDT(NUT    ),V1,6,      6,1,6,150,6)
      CALL MULT3  (BA2,YDT(NUC    ),V2,6,      6,1,6,150,6)
      CALL MULT3  (BA3,YDT(NXITDT),V3,6,NMODET,1,6,150,6)
      CALL MULT3  (BA4,YDT(NXICDT),V4,6,NMODEC,1,6,150,6)
      DO 80 I=1,6
   80 RV(I) = RV(I) - V1(I) + V2(I) - V3(I) + V4(I)
      CALL MULT3  (CLAMI,RV,VLAM,6,6,1,6,6,6)
      DO 85 I=1,6
   85 V1(I) = -VLAM(I)
      CALL MULTAD (CUT,VLAM,YDT(NUT    ),6      ,6,1, 6,6,150)
      CALL MULTAD (CUC,V1   ,YDT(NUC    ),6      ,6,1, 6,6,150)
      CALL MULTAD (CXT,VLAM,YDT(NXITDT),NMODET,6,1,26,6,150)
      CALL MULTAD (CXC,V1   ,YDT(NXICDT),NMODEC,6,1,26,6,150)
      RETURN
      END
```

```
      SUBROUTINE HSIG
C
C   SUBROUTINE TO SELECT CERTAIN AMPLITUDES (ROWS) AND MODES (COLUMNS)
C   OF THE CHASE AND TARGET VEHICLE VIBRATION MODAL MATRICES.
C
C   SUBROUTINES CALLED ARE -- READ, READIM AND REVISE.
C
C   CODED BY CARL BODLEY, JULY 1969
C
      DIMENSION AWORK(100,100), VWORK(100), JVEC(100), IVEC(100)
C
      COMMON /MFREQD/ FREQT(26), FREQC(26), ZETAT, ZETAC
      COMMON /MODATA/ HATD(3,26),SIGATD(3,26), HATP(3,26),SIGATP(3,26)
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
C
      DATA NIT,NOT/5,6/
C
      READ (NIT,1002) IFELAS, NMODET, NMODEC, ZETAT, ZETAC
 1002 FORMAT (3I5, 2E10.0)
C
      IF (IFELAS .EQ. 0) GO TO 200
C
      DO 100 L=1,2
      CALL READ   (VWORK,N1,NCF,1,100)
      CALL READ   (AWORK,NR,NC,100,100)
      CALL READIM (JVEC,N1,NC1,1,100)
      IF (NC1 .NE. NC .OR. NCF .NE. NC) GO TO 999
      NM = 0
      DO 10 I=1,NC
      IF (JVEC(I) .NE. 0) NM = NM + 1
   10 CONTINUE
      GO TO (51,52), L
   51 IF (NM .NE. NMODET) GO TO 999
      GO TO 53
   52 IF (NM .NE. NMODEC) GO TO 999
   53 CALL READIM (IVEC,N1,NC1,1,100)
      IF (NC1 .NE. NR) GO TO 999
      N3 = 0
      DO 20 I=1,NR
      IF (IVEC(I) .NE. 0) N3 = N3 + 1
   20 CONTINUE
      IF (N3 .NE. 3) GO TO 999
C  ------REVISE TO FORM H   (DEFL. AMPLITUDE) --------------
      GO TO (54,55), L
   54 CALL REVISE (AWORK,IVEC,JVEC,HATD   ,NR,NC,3,NMODET,100,3)
      GO TO 56
   55 CALL REVISE (AWORK,IVEC,JVEC,HATP   ,NR,NC,3,NMODEC,100,3)
   56 CALL READIM (IVEC,N1,NC1,1,100)
      IF (NC1 .NE. NR) GO TO 999
      N3 = 0
      DO 30 I=1,NR
      IF (IVEC(I) .NE. 0) N3 = N3 + 1
   30 CONTINUE
      IF (N3 .NE. 3) GO TO 999
C  ------REVISE TO FORM SIG  SLOPE AMPLITUDE) --------------
      GO TO (57,58), L
   57 CALL REVISE (AWORK,IVEC,JVEC,SIGATD,NR,NC,3,NMODET,100,3)
      CALL REVISE (VWORK,1,JVEC,FREQT,1,NC,1,NMODET,1,1)
      GO TO 100
   58 CALL REVISE (AWORK,IVEC,JVEC,SIGATP,NR,NC,3,NMODEC,100,3)
      CALL REVISE (VWORK,1,JVEC,FREQC,1,NC,1,NMODEC,1,1)
  100 CONTINUE
```

```
C
      RETURN
C
  200 NMODET = 1
      NMODEC = 1
      ZETAT = 0.0
      ZETAC = 0.0
      FREQT(1) = 0.0
      FREQC(1) = 0.0
      DO 210 I=1,3
      HATD(I,1) = 0.0
      HATP(I,1) = 0.0
      SIGATD(I,1) = 0.0
  210 SIGATP(I,1) = 0.0
C
      RETURN
C
  999 WRITE (NOT,1001)
 1001 FORMAT (33H1DATA INTO HSIG IS NOT COMPATIBLE/
     *          17H0PROGRAM STOPPED.)
C
      STOP
C
      END
```

```
      SUBROUTINE INV2NP  (A,B,N,KR)
      DIMENSION A(KR,1), B(KR,1), W(150), U(150), IV(150),
     *          IRE(150), BIN(150)
      DATA NIT,NOT/5,6/
C
C  MATRIX INVERSION (A**-1 = B) -- RANK ANNIHILATION METHOD.
C  ALGORITHM FORMULATED BY CARL BODLEY.
C  THE INVERSION CHECK B*A IS CALCULATED AND PRINTED.
C  CALLS SUBROUTINE PAGEHD.
C  THE MAXIMUM SIZE IS
C     N = 150
C  CODED BY   CARL BODLEY   JANUARY, 1967.
C
C      SUBROUTINE ARGUMENTS
C  A  = INPUT  MATRIX TO BE INVERTED. SIZE(N,N).
C  B  = OUTPUT MATRIX INVERSE OF A. SIZE(N,N).
C  N  = INPUT  SIZE OF MATRICES A AND B. (MAX=150).
C  KR = INPUT  ROW DIMENSION OF MATRICES A AND B IN CALLING PROGRAM.
C
 1001 FORMAT (26H1N EXCEEDS INV2 ALLOWABLE./ 17H0PROGRAM STOPPED.)
 1003 FORMAT (//10X,45HSUBROUTINE INV2 HAS CALCULATED THE DATA BELOW
     *        ///10X,40HTHE DIAGONALS OF INVERSION CHECK B*A ARE
     *        // (8X,7E16.8))
 1004 FORMAT (///10X,42HTHE MAXIMUM OFF-DIAGONAL ELEMENT OF B*A IS
     *        E17.8, 2X, 4HAT ( I3, 1H, I3, 1H) )
 1006 FORMAT (19H1MATRIX IS SINGULAR/ 17H0PROGRAM STOPPED.)
C
      IF (N .GT. 150) GO TO 97
      IF (N .EQ. 1 .AND. A(1,1) .EQ. 0.0) GO TO 99
      IF (N .EQ. 1 .AND. A(1,1) .NE. 0.0) GO TO 98
C
C     GENERATE INITIAL ROW INDICES.
      IT = 1
      GO TO 90
   91 IT = 2
   90 DO 5 I=1,N
      IRE(I) = I
    5 IV(I) = I
C
C  CONDITION A FOR MAXIMUM DIAGONAL ELEMENTS
      NM1 = N - 1
      DO 6 L=1,NM1
      SMAX = 0.0
      DO 8 J=L,N
      LA = IRE(J)
      I = L
      K = LA
      IF (IT .EQ. 2) I = LA
      IF (IT .EQ. 2) K = L
      IF (ABS(A(K,I)) .LE. SMAX) GO TO 8
      JMAX = J
      SMAX = ABS(A(K,I))
    8 CONTINUE
      LS = IRE(L)
      IRE(L) = IRE(JMAX)
    6 IRE(JMAX) = LS
      DO 7 L=1,N
      LA = IRE(L)
      BIN(L) = A(LA,L)
      IF (IT .EQ. 2) BIN(L) = A(L,LA)
    7 IF (BIN(L) .EQ. 0.0) BIN(L) = 1.0
C
```

```
C  GENERATE INITIAL B AND ABAR
      DO 10 L=1,N
      LA = IRE(L)
      I = L
      K = LA
      IF (IT .EQ. 2) I = LA
      IF (IT .EQ. 2) K = L
      DO 15 J=1,N
      M = J
      M1 = LA
      IF (IT .EQ. 2) M = LA
      IF (IT .EQ. 2) M1 = J
   15 B(M,M1) = 0.0
      B(I,K) = 1.0/BIN(L)
   10 A(K,I) = A(K,I) - BIN(L)
C
C  INVERSION LOOP, USES ROW OF ABAR WITH MAXIMUM S.
      DO 35 L=1,N
      SMAX = 0.0
      DO 23 J =L,N
      LA = IV(J)
      S = 1.0
      DO 26 K=1,N
   26 S = S + A(LA,K)*B(K,LA)
      IF (ABS(S) .LE. SMAX) GO TO 23
      LMAX = J
      SMAX = ABS(S)
   23 CONTINUE
      IF (SMAX .GT. 1.0E-99) GO TO 60
      IF (IT .EQ. 2) GO TO 99
      GO TO 65
   60 LS = IV(L)
      IV(L) = IV(LMAX)
      IV(LMAX) = LS
      LA = IV(L)
      DO 25 I=1,N
      W(I) = 0.0
      DO 25 J=1,N
   25 W(I) = W(I) + A(LA,J)*B(J,I)
      S = 1.0 + W(LA)
      DO 30 I=1,N
   30 U(I) = B(I,LA)
      DO 35 I=1,N
      DO 35 J=1,N
   35 B(I,J) = B(I,J) - U(I)*W(J)/S
C
C  RESTORE A
   65 DO 40 L=1,N
      LA = IRE(L)
      I = L
      K = LA
      IF (IT .EQ. 2) I = LA
      IF (IT .EQ. 2) K = L
   40 A(K,I) = A(K,I) + BIN(L)
      IF (SMAX .LE. 1.0E-99) GO TO 91
C
      RETURN
C
   98 B(1,1) = 1.0/A(1,1)
      RETURN
   99 WRITE (NOT,1006)
      STOP

   97 WRITE (NOT,1001)
      STOP
      END
```

```
      SUBROUTINE KINMAT
C
C           COMPUTES KINEMATICS FROM ROTATION TRANSFORMATIONS
C                        AND GEOMETRY
C
C
      COMMON /ANGLES/ ALPHAD,SALPD,CALPD,TALPD, SSIGMA(3),CSIGMA(3)
      COMMON /BTDATA/ TRHO(9,6), BL5(10,6), BLV(7,39)
      COMMON /FUNGAM/ SGAMMA(3),CGAMMA(3),SGSS(3),SGCS(3),
     *                                  CGSS(3),CGCS(3)
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTO(3),CPCO(3),RADCON,CSPHER,ROTALP,PEO,FDO
      COMMON /LIPARM/ FLAMC(3), QRMAG(3)
      COMMON /LOCPTC/ CSC(3), CAC(3,3),CQC(3,3),
     *                CPC(3),PSC(3),PAC(3,3),PQC(3,3)
      COMMON /LOCPTD/ DSD(3),DHD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /LOCPTH/  DSH(3)
      COMMON /LOCPTP/ PSP(3),PAP(3,3),PQP(3,3)
      COMMON /LOCPTT/ THT(3),TCT(3),TST(3),TAT(3,3),TRT(3,3),
     *                CST(3),CAT(3,3),   QRT(3,3), TDT(3),
     *                DHT(3),DGT(3),DST(3), DAT(3,3),DQT(3,3),DRT(3,3)
      COMMON /MODATA/ HATD(3,26),SIGATD(3,26), HATP(3,26),SIGATP(3,26)
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /SKEWMS/ TDTSQ(3,3),THTSQ(3,3),TRTSQ(3,3,3), DRTSQ(3,3,3),
     *                PSCSQ(3,3),PSPPSQ(3,3),PQCSQ(3,3,3),PQPPSQ(3,3,3),
     *                CSCSQ(3,3),CQCSQ(3,3,3),  DHTSQ(3,3)
      COMMON /SLIDFR/ HDMU, ARMMU, BFV(8,39)
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECMAG/ FDMAG(3), PEMAG(3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION TEMP(3,26), TEMQ(3,26), CGAM(3,3)
C
      DATA IFSF /1/
C
      IF (TMST. GT. 0.0) GO TO 2
C
      COSALP = CSPHER/SALPD
      IF (HDMU.EQ.0.0 .AND. ARMMU.EQ.0.0) IFSF = 0
C
      CALL ZERO (TRHO,6,6,9)
      CALL ZERO (BL5,10,6,10)
      CALL ZERO (BLV(5,1),3,3,7)
C
      DO 1 I=1,3
      BLV (I+4,I) =-1.0
    1 TRHO(  I,I) = 1.0
C
    2 CONTINUE
C
      CALL SKEWV (TDT, TDTSQ)
C
      DO 4 I=1,3
```

```
      I3 = I+3
C
      TRM1 = DLD*SGAMMA(I) - AD*CGAMMA(I)
      TRM2 = DLD*CGAMMA(I) + AD*SGAMMA(I)
      TRM3 = AF-AA-TRM1
C
      FDI  = DLI - DLF + Y(NRHO) - TRM2
      FDJ  = TRM3*SSIGMA(I)
      FDK  =-TRM3*CSIGMA(I)
      FDMAG(I) = SQRT(FDI*FDI+FDJ*FDJ+FDK*FDK)
C
      UAJX = FDI/FDMAG(I)
      UAJY = FDJ/FDMAG(I)
      UAJZ = FDK/FDMAG(I)
C
      TRHO(I3, 1)    = UAJX
      TRHO(I3,I3)    = TRM1*UAJX + TRM2*(UAJZ*CSIGMA(I) - UAJY*SSIGMA(I))
C
      TRM1 = DLE*SGAMMA(I) - AE*CGAMMA(I)
      TRM2 = DLE*CGAMMA(I) + AE*SGAMMA(I)
      TRM3 = AP-AA-TRM1
C
      PEI = DLI - DLP + Y(NRHO) - TRM2
      PEJ  = TRM3*SSIGMA(I)
      PEK  =-TRM3*CSIGMA(I)
      PEMAG(I) = SQRT(PEI*PEI+PEJ*PEJ+PEK*PEK)
C
      UTJX = PEI/PEMAG(I)
      UTJY = PEJ/PEMAG(I)
      UTJZ = PEK/PEMAG(I)
C
      BL5(I, 1) =-UTJX
      BL5(I,I3) =-TRM1*UTJX - TRM2*(UTJZ*CSIGMA(I) - UTJY*SSIGMA(I))
C
    4 CONTINUE
C
      CALL SKEWV (THT, THTSQ)
      CALL SKEWV (CSC, CSCSQ)
      CALL SKEWV (DHT, DHTSQ)
      CALL SKEWV (PSC, PSCSQ)
      CALL SKEWV (PSP, PSPPSQ)
C
      PSPPSQ(1,1) = 1.0
      PSPPSQ(2,1) = 0.0
      PSPPSQ(3,1) = 0.0
C
      DO 10 J=1,3
      IF (IFSF .EQ. 0) GO TO 10
      BFV(1,J) = TRHT(1,J)
      BFV(2,J) = TRHT(3,J)
   10 BLV(1,J) = TRHT(2,J)
C
      CALL MULT3(TRHT(2,1),THTSQ,BLV(1,4),1,3,3,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHT(1,1),THTSQ,BFV(1,4),1,3,3,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHT(3,1),THTSQ,BFV(2,4),1,3,3,3,3,8)
C
      DO 11 J=1,3
      IF (IFSF .EQ. 0) GO TO 11
      BFV(1,J+6) = -TRHC(1,J)
      BFV(2,J+6) = -TRHC(3,J)
```

```
   11 BLV(1,J+6) = -TRHC(2,J)
C
      CALL MULT3(TRHC(2,1),CSCSQ,BLV(1,10),1,3,3,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHC(1,1),CSCSQ,BFV(1,10),1,3,3,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHC(3,1),CSCSQ,BFV(2,10),1,3,3,3,3,8)
C
      CALL MULT (DHTSQ,SIGATD,TEMP,3,3,NMODET,3,3)
      CALL AABB (1.0,HATD,1.0,TEMP,TEMQ,3,NMODET,3)
      CALL MULT3(TRHT(2,1),TEMQ,BLV(1,13),1,3,NMODET,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHT(1,1),TEMQ,BFV(1,13),1,3,NMODET,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHT(3,1),TEMQ,BFV(2,13),1,3,NMODET,3,3,8)
C
      CALL MULT (PSCSQ,SIGATP,TEMP,3,3,NMODEC,3,3)
      CALL AABB (1.0,HATP,1.0,TEMP,TEMQ,3,NMODEC,3)
      CALL MULT3(TRHC(2,1),TEMQ,BLV(1,13+NMODET),1,3,NMODEC,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHC(1,1),TEMQ,BFV(1,13+NMODET),1,3,NMODEC,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRHC(3,1),TEMQ,BFV(2,13+NMODET),1,3,NMODEC,3,3,8)
C
      CALL MULT3(TRTC,TEMQ,BLV(5,13+NMODET),3,3,NMODEC,3,3,7)
C
      CALL MULT3(TRHP(2,1),PSPPSQ,BL5(4,1),1,3,3,3,3,10)
C
      CALL MULT3(TRTP,PSPPSQ,BL5(8,1),3,3,3,3,3,10)
C
      DO 16 I=1,3
      I4 = I+4
C
      DO 14 J=1,3
      J3 = J+3
      J6 = J+6
      BLV(I4,J3) =-TDTSQ(I,J)
   14 BLV(I4,J6) = TRTC(I,J)
C
      DO 15 J=1,NMODET
   15 BLV(I4,J+12) =-HATD(I,J)
C
   16 CONTINUE
C
      CALL MULT3(TRTC,CSCSQ,BLV(5,10),3,3,3,3,3,7)
C
      DO 50 K=1,3
      K1 = K+1
      K2 = 2*K+1
      K3 = K2+1
C
      CALL SKEWV (TRT(1,K),  TRTSQ(1,1,K))
      CALL SKEWV (CQC(1,K),  CQCSQ(1,1,K))
      CALL SKEWV (DRT(1,K),  DRTSQ(1,1,K))
      CALL SKEWV (PQC(1,K),  PQCSQ(1,1,K))
      CALL SKEWV (PQP(1,K),  PQPPSQ(1,1,K))
C
      PQPPSQ(1,1,K) = 1.0
      PQPPSQ(2,1,K) = 0.0
      PQPPSQ(3,1,K) = 0.0
C
      DO 20 J=1,3
```

```
      IF (IFSF .EQ. 0) GO TO 20
      BFV(K2,J) = TRRT(1,J,K)
      BFV(K3,J) = TRRT(3,J,K)
   20 BLV(K1,J) = TRRT(2,J,K)
C
      CALL MULT3(TRRT(2,1,K),TRTSQ(1,1,K),BLV(K1,4),1,3,3,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRT(1,1,K),TRTSQ(1,1,K),BFV(K2,4),1,3,3,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRT(3,1,K),TRTSQ(1,1,K),BFV(K3,4),1,3,3,3,3,8)
C
      DO 22 J=1,3
      IF (IFSF .EQ. 0) GO TO 22
      BFV(K2,J+6) = -TRRC(1,J,K)
      BFV(K3,J+6) = -TRRC(3,J,K)
   22 BLV(K1,J+6) = -TRRC(2,J,K)
C
      CALL MULT3(TRRC(2,1,K),CQCSQ(1,1,K),BLV(K1,10),1,3,3,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRC(1,1,K),CQCSQ(1,1,K),BFV(K2,10),1,3,3,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRC(3,1,K),CQCSQ(1,1,K),BFV(K3,10),1,3,3,3,3,8)
C
      CALL MULT (DRTSQ(1,1,K),SIGATD,TEMP,3,3,NMODET,3,3)
      CALL AABB (1.0,HATD,1.0,TEMP,TEMQ,3,NMODET,3)
      CALL MULT3(TRRT(2,1,K),TEMQ,BLV(K1,13),1,3,NMODET,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRT(1,1,K),TEMQ,BFV(K2,13),1,3,NMODET,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRT(3,1,K),TEMQ,BFV(K3,13),1,3,NMODET,3,3,8)
C
      CALL MULT (PQCSQ(1,1,K),SIGATP,TEMP,3,3,NMODEC,3,3)
      CALL AABB (1.0,HATP,1.0,TEMP,TEMQ,3,NMODEC,3)
      CALL MULT3(TRRC(2,1,K),TEMQ,BLV(K1,13+NMODET),1,3,NMODEC,3,3,7)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRC(1,1,K),TEMQ,BFV(K2,13+NMODET),1,3,NMODEC,3,3,8)
      IF (IFSF .NE. 0)
     *CALL MULT3(TRRC(3,1,K),TEMQ,BFV(K3,13+NMODET),1,3,NMODEC,3,3,8)
C
      CALL MULT3(TRRP(2,1,K),PQPPSQ(1,1,K),BL5(K+4,1),1,3,3,3,3,10)
C
      CGAM(1,K) =  FLAMC(K)*SGAMMA(K)
      CGAM(2,K) = -FLAMC(K)*CGSS(K)
      CGAM(3,K) =  FLAMC(K)*CGCS(K)
C
      CALL MULT3(TRRP(2,1,K),CGAM(1,K),BL5(K+4,K+3),1,3,1,3,3,10)
C
   50 CONTINUE
C
      JT = 12+NMODET
C
      DO 69 I=1,4
      I3 = I+3
      I4 = I+4
C
      DO 60 J=10,12
      IF (IFSF .EQ. 0) GO TO 60
      BFV( I, J) =  BFV( I, J)
      BFV(I4, J) = -BFV(I4, J)
   60 BLV( I, J) = -BLV( I, J)
C
      DO 62 J=1,NMODEC
```

```
      JJ = J+JT
      IF (IFSF .EQ. 0) GO TO 62
      BFV( I,JJ) = -BFV( I,JJ)
      BFV(I4,JJ) = -BFV(I4,JJ)
   62 BLV( I,JJ) = -BLV( I,JJ)
C
      DO 64 J=1,3
   64 BL5(I3, J) = -BL5(I3, J)
C
   69 CONTINUE
C
      DO 70 I=4,6
      I1 = I+1
   70 BL5(I1, I) = -BL5(I1,I)
C
      DO 72 I=1,3
      I4 = I+4
      PHICON( I) = PED - PEMAG(I)
   72 PHICON(I4) = QRMAG(I) - AB
C
      PHICON( 4) =-DSH(2) - CSPHER
      PHICON( 8) = DSD(1) - COSALP
      PHICON( 9) = DSD(2)
      PHICON(10) = DSD(3)
C
      RETURN
      END
```

```
      SUBROUTINE LAMRHO
C
      COMMON /BINOFF/ FI,BINOMU
      COMMON /BTDATA/ TRHO(9,6), BLS(10,6), BLV(7,39)
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTO(3),CPCO(3),RADCON,CSPHER,ROTALP,PEO,FDO
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /JCOUNT/ JIL
      COMMON /MISCNO/ ANUM
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FRET
      COMMON /SPRDMP/ FSPRNG(2,20,4),FDAMP(2,20,4)
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECMAG/ FDMAG(3), PEMAG(3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION BC(6,6),BCI(6,6),WKV(6),VW(10),FD(6),DDOT(6),
     *          IVEC(10),JVEC(6),VCON(10),FT(6),SIDF(2),WV3(3),ANVEC(3)
C
      DATA JVEC/1,2,3,4,5,6/
      DATA I1ST,CKV / 0, 0.9975 /
      DATA BTOL,MAXITR/ 5., 15 /
      DATA NIT,NOT /5, 6 /
C
      IF (I1ST .GT. 0) GO TO 7
      FROXO = (DLQ+2.*DLS+EXTL)/(DLQ-EXTL)
      NALY = NRHO + 1
      NALZ = NRHO + 2
C
      ISCD = 13 + NMODET
      DO 15 I=1,3
   15 VW (I) = 0.
C
      I1ST = 1
    7 CALL MULT3  (BLV        ,Y(    NUT),VW ( 4),7,      6,1,7,150,10)
      CALL MULTAD (BLV(1,    7),Y(    NUC),VW ( 4),7,      6,1,7,150,10)
      CALL MULTAD (BLV(1,   13),Y(NXITDT),VW ( 4),7,NMODET,1,7,150,10)
      CALL MULTAD (BLV(1,ISCD),Y(NXICDT),VW ( 4),7,NMODEC,1,7,150,10)
C
  100 IVEC(4) = 0
      DO 17 I=1,3
      IP4 = I + 4
      IP7 = I + 7
      IVEC(I  ) = MODE(I  )
      IVEC(IP4) = MODE(IP4)
      IF (MODE(I) .EQ. 0 .AND. MODE(IP4) .EQ. 0) IVEC(IP4) = 1
   17 IVEC(IP7) = 1
C
      NVAL = 0
      DO 20 I=1,10
      VCON(I) = -VW(I)
      IF (IVEC(I) .EQ. 0) GO TO 20
      NVAL = NVAL + 1
      IVEC(I) = NVAL
   20 CONTINUE
```

```
C
      CALL REVISE (BL5,IVEC,JVEC,BC,10,6,6,6,10,6)
      CALL REVISE (VCON  ,IVEC,JVEC(1),WKV,10,1,6,1,10,6)
C
      IF (MODE(4) .EQ. 1 .OR. MODE(8) .EQ. 1) GO TO 105
      WV3(1) = (1.0-CKV)*(EXTL-Y(NRHO))/DELTAT
      WV3(2) = (CKV-1.0)*Y(NALY)/DELTAT
      WV3(3) = (CKV-1.0)*Y(NALZ)/DELTAT
      CALL MULT3  (BC(4,1),WV3,WKV(4),3,3,1,6,1,6)
      GO TO 50
  105 IF (MODE(4) .EQ. 0) GO TO 50
C
      DO 25 I=1,3
   25 WKV(I+3) = VW(4)*TRHT(2,I)
C
   50 CALL INV2NP (BC,BCI,6,6)
      CALL MULT3  (BCI,WKV,YDT(NRHO),6,6,1,6,6,150)
      DO 65 I=1,3
      NGM = NRHO + 2 + I
      IP4 = I + 4
      IF (MODE(I) .EQ. 1 .OR. MODE(IP4) .EQ. 1) GO TO 65
      YDT(NGM) = (1.0-CKV)*(49.*ANUM-Y(NGM))/DELTAT
   65 CONTINUE
C
      CALL MULT3  (TRHO,YDT(NRHO),DDOT,6,6,1,9,150,6)
C
C     *********************************
C
      CALL FTABLE   (Y(NRHO),FD(1),SLOPE,BINTR,FSPRNG(1,1,1),
     *               FSPRNG(1,1,1),1,MM)
      STEST = Y(NALY)*YDT(NALY)
      IF (STEST .GE. 0.) CALL FTABLE (Y(NALY),FD(2),SLOPE,BINTR,
     *     FSPRNG(1,1,2),FSPRNG(1,1,2),1,MM)
      IF (STEST .LT. 0.) CALL FTABLE (Y(NALY),FD(2),SLOPE,BINTR,
     *     FSPRNG(1,1,3),FSPRNG(1,1,3),1,MM)
      STEST = Y(NALZ)*YDT(NALZ)
      IF (STEST .GE. 0.) CALL FTABLE (Y(NALZ),FD(3),SLOPE,BINTR,
     *     FSPRNG(1,1,2),FSPRNG(1,1,2),1,MM)
      IF (STEST .LT. 0.) CALL FTABLE (Y(NALZ),FD(3),SLOPE,BINTR,
     *     FSPRNG(1,1,3),FSPRNG(1,1,3),1,MM)
      DO 40 I=1,3
      IP3 = I + 3
   40 CALL FTABLE(FDMAG(I),FD(IP3),SLOPE,BINTR,FSPRNG(1,1,4),
     *             FSPRNG(1,1,4),1,MM)
C
      CALL RETRAK
      FD (1) = FD (1) + FRET
C
      DO 27 I=1,6
      IL = 4
      IF (I .LE. 3) IL = I
      CALL FTABLE (DDOT(I),FDD,SLOPE,BINTR,FDAMP(1,1,IL),
     *             FDAMP(1,1,IL),1,MM)
   27 FT(I) = FD(I) + FDD
C
      DO 600 I=1,7
      IF (IVEC(I) .EQ. 0) GO TO 600
      NCN = IVEC(I)
      ANVEC(NCN) = MODE(I)
  600 CONTINUE
C
      CALL MULT3  (FT,TRHO,FD,1,6,6,1,9,1)
```

```
      CALL MULT3  (FD(4),BCI(4,1),WV3,1,3,3,1,6,1)
      DO 605 I=1,3
  605 WKV(I) = WV3(I)*ANVEC(I)
      DO 610 I=1,3
      WV3(I) = FD(I)
      DO 610 J=1,3
  610 WV3(I) = WV3(I) - WKV(J)*BC(J,I)
      CALL MULT3  (WV3,BCI(1,4),WKV(4),1,3,3,1,6,1)
      FI = 0.
      IF (MODE(4) .EQ. 1 .OR. MODE(8) .EQ. 1) GO TO 652
      DO 615 I=4,6
  615 WKV(I) = 0.
      GO TO 151
  652 RODMAG = ABS(DDOT(1))
      IF (RODMAG .LT. 1.E-9) GO TO 151
      VALUE = (DLQ+2.*DLS+Y(NRHO))/(DLQ-Y(NRHO))
      VALUE = BINDMU*VALUE/FROXD
      ICNT = 0
  153 CALL MULT3  (TRPT(2,1),WKV(4),SIDF,2,3,1,3,6,2)
      FSID = SQRT(SIDF(1)*SIDF(1) + SIDF(2)*SIDF(2))
      FIP1 = (DDOT(1)/RODMAG)*VALUE*FSID
      FBDLT = FIP1 - FI
      IF (ABS(FBDLT) .LE. BTOL) GO TO 151
      ICNT = ICNT + 1
      IF (ICNT .EQ. MAXITR) GO TO 999
      FI = FIP1
      DO 152 I=4,6
  152 WKV(I) = WKV(I) + FBDLT*BCI(1,I)
      GO TO 153
C
  151 DO 45 I=1,10
      IV = IVEC(I)
      ALAM(I) = 0.
      IF (IV .EQ. 0) GO TO 45
      IF (MODE(I) .EQ. 1) ALAM(I) = WKV(IV)
   45 CONTINUE
      IF (MODE(4) .EQ. 1) ALAM(4) = -TRHT(2,1)*WKV(4)
     *           - TRHT(2,2)*WKV(5) - TRHT(2,3)*WKV(6)
C
      IF (JIL .NE. 4) RETURN
C
      IF (ALAM(4) .GE. 0.) GO TO 57
      MODE(4) = 0
      GO TO 100
   57 IFT = 0
      DO 55 I=1,7
      IF (ALAM(I) .GE. 0.) GO TO 55
      IFT = 1
      MODE(I) = 0.
   55 CONTINUE
C
      IF (IFT .EQ. 0) RETURN
      GO TO 100
C
  999 WRITE (NOT,1001)
 1001 FORMAT (36H1SUBROUTINE LAMRHO DOES NOT CONVERGE/
     *       17H0PROGRAM STOPPED.)
      STOP
C
      END
```

```
      SUBROUTINE LOCATE
C
C          SUBROUTINE TO ESTABLISH LOCATION OF KEY LEADING ELEMENTS
C          IN STATE VECTOR AND TO ESTABLISH LENGTH OF SUBVECTORS
C                          OF STATE VECTOR
C
      COMMON /LOCLEN/ LOC(12), LEN(12)
      COMMON /NCNTRL/ NCNSYS
      COMMON /NEQTNS/ NEQTN
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DATA   NIT, NOT   /  5,  6  /
C
      NXIT   = NXITDT + NMODET
      NXICDT = NXIT + NMODET
      NXIC   = NXICDT + NMODEC
      NDELCS = NXIC + NMODEC
      NEQTN  = NDELCS + NCNSYS - 1
C
      IF (NEQTN .GT. 150)  GO TO 20
C
      LOC( 1) = NUT
      LOC( 2) = NXT
      LOC( 3) = NGAMT
      LOC( 4) = NUC
      LOC( 5) = NXC
      LOC( 6) = NGAMC
      LOC( 7) = NRHO
      LOC( 8) = NXITDT
      LOC( 9) = NXIT
      LOC(10) = NXICDT
      LOC(11) = NXIC
      LOC(12) = NDELCS
C
      DO 10  I=1,11
   10 LEN( I) = LOC(I+1) - LOC(I)
      LEN(12) = NCNSYS
C
                        CALL WRITIM(LOC,1,12,6H   LOC,1)
                        CALL WRITIM(LEN,1,12,6H   LEN,1)
C
      GO TO 30
C
   20 WRITE(NOT,21)
   21 FORMAT(/,10X, 38HNO. OF EQUATIONS EXCEEDS 150, STOP RUN)
      STOP
C
   30 RETURN
      END
```

```
      SUBROUTINE MINYS
C
C
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTO(3),CPCO(3),RADCON,CSPHER,ROTALP,PEO,FOO
      COMMON /LOCPTP/ PSP(3),PAP(3,3),PQP(3,3)
      COMMON /MISCNO/ ANUM
      COMMON /NEQTNS/ NEQTN
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
      DIMENSION TRIC(3,3),TRIT(3,3),TRTC(3,3)
      DIMENSION DATMIN(6,3)
      DATA NIT,NOT/ 5, 6/
C
 1001 FORMAT (/// 10X,33HTHE DATMIN ARRAY IS AS FOLLOWS -- //
     *10X,50H1 TARGET EULER ANGLES - PSI, THETA, PHI  (DEGREES)/
     *10X,58H2 CHASE / TARGET EULER ANGLES - PSI, THETA, PHI  (DEGREES)/
     *10X,56H3 CHASE ANGULAR VELOCITY - OMEGA X, OMEGA Y, OMEGA Z - (,
     *                  12HDEGREES/SEC)/
     *10X,55H4 CHASE C.G. VELOCITY IN X,Y,Z DIRECTIONS REFERENCED TO,
     *                  19H CHASE (LENGTH/SEC)/
     *10X,58H5 D TO S DISTANCE IN X,Y,Z DIRECTIONS REFERENCED TO TARGET,
     *                   9H (LENGTH)/
     *10X,55H6 RHO(X), (LENGTH),  GAMMA(1,2,3), (DEGREES),  --------)
C
C
C   ROUTINE READS MINIMUM INITIAL CONDITIONS AND INITIALIZES THE Y(I) VECTOR.
C
C   READ MINIMUM INITIAL CONDITIONS AS FOLLOWS.
C     1 TARGET EULER ANGLES - PSI, THETA, PHI - (DEGREES)
C     2 CHASE / TARGET EULER ANGLES - PSI, THETA, PHI  (DEGREES)
C     3 CHASE ANGULAR VELOCITY - OMEGA X, OMEGA Y, OMEGA Z - (DEGREES/SEC)
C     4 CHASE C.G. VELOCITY IN X,Y,Z DIRECTIONS REFERENCED TO CHASE (LENGTH/SEC)
C     5 D TO S DISTANCE IN X,Y,Z DIRECTIONS REFERENCED TO TARGET (LENGTH)
C     6 RHO - X, (LENGTH), GAMMA (DEGREES),  ---------,  REFERENCED TO CHASE
C
      CALL READ (DATMIN,6,3,6,3)
C
C  ZERO SPACES AND CHANGE DEGREES TO RADIANS.
C
      DO 5 I = 1,NEQTN
    5 Y(I) = 0.0
      DATMIN(6,2) = DATMIN(6,2) * ANUM
      DO 10 I = 1,3
      DO 10 J = 1,3
      TRIT(I,J) = 0.0
      TRIC(I,J) = 0.0
      TRTC(I,J) = 0.0
   10 DATMIN(I,J) = DATMIN(I,J) * ANUM
C
C   COMPUTE TRIT(I,J), TRTC(I,J), TRIC(I,J) ROTATION TRANSFORMATIONS.
C
      CS1IT = COS(DATMIN(1,1))
      CS2IT = COS(DATMIN(1,2))
      CS3IT = COS(DATMIN(1,3))
      SN1IT = SIN(DATMIN(1,1))
      SN2IT = SIN(DATMIN(1,2))
      SN3IT = SIN(DATMIN(1,3))
      CS1TC = COS(DATMIN(2,1))
      CS2TC = COS(DATMIN(2,2))
      CS3TC = COS(DATMIN(2,3))
```

```
      SN1TC = SIN(DATMIN(2,1))
      SN2TC = SIN(DATMIN(2,2))
D
      SN3TC = SIN(DATMIN(2,3))
      TRIT(1,1) = CS1IT * CS2IT
      TRTC(1,1) = CS1TC * CS2TC
      TRIT(2,1) = SN1IT * CS2IT
      TRTC(2,1) = SN1TC * CS2TC
      TRIT(3,1) = - SN2IT
      TRTC(3,1) = - SN2TC
      TRIT(1,2) = CS1IT * SN2IT * SN3IT - SN1IT * CS3IT
      TRTC(1,2) = CS1TC * SN2TC * SN3TC - SN1TC * CS3TC
      TRIT(2,2) = SN1IT * SN2IT * SN3IT + CS1IT * CS3IT
      TRTC(2,2) = SN1TC * SN2TC * SN3TC + CS1TC * CS3TC
      TRIT(3,2) = CS2IT * SN3IT
      TRTC(3,2) = CS2TC * SN3TC
      TRIT(1,3) = CS1IT * SN2IT * CS3IT + SN1IT * SN3IT
      TRTC(1,3) = CS1TC * SN2TC * CS3TC + SN1TC * SN3TC
      TRIT(2,3) = SN1IT * SN2IT * CS3IT - CS1IT * SN3IT
      TRTC(2,3) = SN1TC * SN2TC * CS3TC - CS1TC * SN3TC
      TRIT(3,3) = CS2IT * CS3IT
      TRTC(3,3) = CS2TC * CS3TC
      CALL MULT (TRIT,TRTC,TRIC,3,3,3,3,3)
C
C   FILL IN THE Y(I) STATE VECTOR WITH INITIAL CONDITIONS.
C
      PSP(1) = DLI + DLS + DATMIN(6,1)
      PSP(2) = 0.0
      PSP(3) = 0.0
      Y(NRHO) = DATMIN(6,1)
      DO 25 I = 1,3
      Y(NGAMT + I - 1) = TRIT(I,1)
      Y(NGAMT + I + 2) = TRIT(I,2)
      Y(NUC + I - 1) = DATMIN(4,I)
      Y(NUC + I + 2) = DATMIN(3,I)
      DO 20 J = 1,3
   20 Y(NXC + I - 1) = Y(NXC + I - 1) + TRIT(I,J) * (TDTC(J) + DATMIN
     *(5,J)) - TRIC(I,J) * (CPCO(J) + PSP(J))
      Y(NGAMC + I - 1) = TRIC(I,1)
      Y(NGAMC + I + 2) = TRIC(I,2)
   25 Y(NRHO + I + 2) = DATMIN(6,2)
C
      WRITE (NOT,1001)
      CALL WRITE (Y,1,NEQTN,6HYINITL,1)
C
      RETURN
      END
```

```
      SUBROUTINE MULT3 (A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)
      DIMENSION A(KRA,1), B(KRB,1), Z(KRZ,1)
C
      DO 10 I=1,NRA
      DO 10 J=1,NCB
      Z(I,J) = 0.0
      DO 10 K=1,NRB
   10 Z(I,J) = Z(I,J) + A(I,K) * B(K,J)
C
      RETURN
      END
```

```
      SUBROUTINE MULTAD (A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)
      DIMENSION A(KRA,1), B(KRB,1), Z(KRZ,1)
C
      DO 10 I=1,NRA
      DO 10 J=1,NCB
      DO 10 K=1,NRB
   10 Z(I,J) = Z(I,J) + A(I,K) * B(K,J)
C
      RETURN
      END
```

```
      SUBROUTINE PIERCE (DA,DL,DS,R,ALPHA,SALP,CALP,TAL, ,
     1                       ROTALP,DAB)
C
C          SUBROUTINE TO ESTABLISH IF PIERCE POINTS LIE
C              BELOW DROGUE LIP
C          OR IF PROBE HAS PIERCED DROGUE
C
      DIMENSION DA(3),DL(3), DS(3)
      DIMENSION DB(3),DC(3),DD(3)
      DIMENSION                    PHIPP(2),RPP(2),SPHIPP(2),CPHIPP(2)
C
      DATA   NIT, NOT   /  5,  6  /
C
      IF ((DA(1).LE.0.0) .OR. (DL(1).LE.0.0) .OR.
     1(SQRT(DA(2)*DA(2)+DA(3)*DA(3))/DA(1).GE.TALP)) GO TO 100
      IF ((DS(1).LE.0.0) .OR. (SQRT(DS(2)*DS(2)+DS(3)*DS(3))/DS(1).GE.
     1TALP))  GO TO 110
C
      CALL AVPBV(1.0,DA,DAB,DL,DB)
C
      IF (SQRT(DB(2)*DB(2)+DB(3)*DB(3))/DB(1) .LT. TALP)   RETURN
C
      DD(1) = DA(2)*DB(3) - DA(3)*DB(2)
      DD(2) = DA(3)*DB(1) - DA(1)*DB(3)
      DD(3) = DA(1)*DB(2) - DA(2)*DB(1)
C
      E = -DD(1)/(TALP*SQRT(DD(2)*DD(2)+DD(3)*DD(3)))
      DENOM = SQRT(1.0 - E*E)
      BETA = ATAN2(DD(2),DD(3))
C
      PHIPP(1) = ATAN2(E, DENOM) - BETA
      PHIPP(2) = ATAN2(E,-DENOM) - BETA
C
      DO 12 I=1,2
      SPHIPP(I) = SIN(PHIPP(I))
   12 CPHIPP(I) = COS(PHIPP(I))
C
      CALL AVPBV(1.0,DB,-1.0,DA,DC)
C
      IF (DD(1).EQ.0.0) GO TO 30
      IF (DD(2).EQ.0.0) GO TO 40
C
   30 DO 35 I=1,2
   35 RPP(I) = DD(2)/(DC(1)*SALP*SPHIPP(I) - DC(3)*CALP)
      GO TO 60
C
   40 DO 45 I=1,2
   45 RPP(I) = DD(1)/(SALP*(DC(3)*CPHIPP(I) - DC(2)*SPHIPP(I)))
C
   60 RMAX = AMAX1(RPP(1),RPP(2))
C
      IF (RMAX*CALP .LE. ROTALP)  GO TO 106
C
      RETURN
C
  100 WRITE(NOT,101)
  101 FORMAT(/,10X, 37HPITCH ARM ORIENTATION ERROR, STOP RUN)
      STOP
  106 WRITE(NOT,107)
  107 FORMAT(/,10X,38HPITCH ARM HAS PIERCED DROGUE, STOP RUN)
      STOP
  110 WRITE(NOT,111)


  111 FORMAT(/,10X, 34HPROBE HAS PIERCED DROGUE, STOP RUN)
      STOP
      END
```

```
      PROGRAM PLOTER
C
      COMMON /PLTR  / K1,NCOLS,NTIMES,XDLTA,NPLTS
       COMMON /TAPED/ NTAPE1, NTAPE2, NTAPE3
      COMMON /TIMESS/ STARTT,CELTAT, T, ENDT, TMST, NSECS
C
      DIMENSION Z(500,60), PTITLE(4)
C
      REWIND NTAPE1
      REWIND NTAPE2
C
      NCOLP1 = NCOLS+1
      IZ = 0
      KPL = NTIMES/NPLTS+1
      KCUNT = KPL-1
C
      DO 200 N1=1,NTIMES
      KOUNT = KOUNT+1
      IF (KOUNT .NE. KPL) READ(NTAPE1) DUMMY
      IF (KOUNT .NE. KPL) GO TO 201
      IZ = IZ+1
      READ(NTAPE1) (Z(IZ,K),K=1,NCOLP1)
      KOUNT = 0
  201 CONTINUE
  200 CONTINUE
C
  210 READ  (NTAPE2) NCX,NCY,NXPL,XNAME,YNAME,PTITLE
      IF (NCX.EQ.0) CALL FRAME
      IF (NCX.EQ.0)  RETURN
C
      XSTART = STARTT
      XDELTA = XDLTA
      IF (NXPL .GT. 0)  XDELTA = FLOAT(NXPL)*XDLTA
      IF (NCX .GT. 1) CALL SCALER(Z(1,NCY),Z(1,NCX),IZ,XSTART,XDELTA)
C
      CALL PLOT1 (Z(1,NCX),Z(1,NCY),IZ,     1,XSTART,XDELTA,
     *                  XNAME,YNAME,PTITLE,1,1,0,500)
C
      GO TO 210
C
      END
```

```
      SUBROUTINE PREENT
C
      COMMON /BINDFR/ FI,BINDMU
      COMMON /CHANGE/ Z(150),ZDT(150)
      COMMON /FORMOM/ FANOM(12)
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /LOCLEN/ LOC(12), LEN(12)
      COMMON /LOCPTD/ DSD(3),DHD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /MISCNO/ ANUM
      COMMON /NEQTNS/ NEQTN
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /PUNCHY/ IFPNCH
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TIAG,TSTAR,FRET
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION NAMES(36)
C
      DATA NIT,NOT  /5,6/
      DATA NAMES /6HU - TA,6HRGET  ,6H      ,6HX - TA,6HRGET  ,6H      ,
     *            6HGAMMA ,6HTARGET,6H      ,6HU - CH,6HASE   ,6H      ,
     *            6HX - CH,6HASE   ,6H      ,6HGAMMA ,6HCHASE ,6H      ,
     *            6HRHOS  ,6H      ,6H      ,6HMODE V,6HEL - T,6HARGET ,
     *            6HMODE D,6HISP - ,6HTARGET,6HMODE V,6HEL - C,6HHASE  ,
     *            6HMODE D,6HISP - ,6HCHASE ,6HCON SY,6HS PARA,6HMETERS/
C
      IF(TMST .EQ. 0.0)  KOUNT = NSECS
      IF(TMST .EQ. 0.0) GO TO 10
      KOUNT = KOUNT+1
   10 CONTINUE
C
C       ALTER Y AND YDT FOR PRINTING AND PLOTTING
      DO 12 I=1,NEQTN
      Z(I) = Y(I)
   12 ZDT(I) = YDT(I)
      DO 13 I=1,3
      JI = NUT+2+I
      JJ = NUC+2+I
      Z(JI) = Y(JI)/ANUM
      ZDT(JI) = YDT(JI)/ANUM
      Z(JJ) = Y(JJ)/ANUM
   13 ZDT(JJ) = YDT(JJ)/ANUM
      DO 14 I=1,5
      J = I+NRHO
      Z(J) = Y(J)/ANUM
   14 ZDT(J) = YDT(J)/ANUM
      DO 15 I=1,3
      J = NDELCS+I
      K = NDELCS + I + 18
      Z(K) = Y(K)/ANUM
      ZDT(K) = YDT(K)/ANUM
      Z(J) = Y(J)/ANUM
   15 ZDT(J) = YDT(J)/ANUM
C
      IF(KOUNT .NE. NSECS) GO TO 100
C
      WRITE(NOT,730) T
  730 FORMAT(1H1,///, 37X,17HSIMULATION TIME =, E14.4)
```

```
C
      WRITE(NOT,600)
  600 FORMAT(////,35X,4HY(I),30X,7HYDOT(I))
      DO 5 I = 1,12
      K = (I - 1) * 2 + I
      L = K + 2
      WRITE (NOT,500) (NAMES(J), J = K,L)
  500 FORMAT (1X, 3A6)
      NBEGIN = LOC(I)
      NEND = LOC(I) + LEN(I) - 1
      DO 5 J = NBEGIN,NEND
    5 WRITE (NOT,605) Z(J), ZDT(J)
  605 FORMAT (27X,E17.8,17X,E17.8)
C
      WRITE(NOT,610) (DSD(I), I = 1,3)
  610 FORMAT(//,13HDSD VECTOR = , 3(5X,E17.8))
C
      WRITE(NOT,615)
  615 FORMAT (1H0,16X,3HPHI,22X,5HLAMDA,23X,4HMODE,/)
      WRITE(NOT,620) (I,PHICON(I),ALAM(I),MODE(I),I = 1,10)
  620 FORMAT (3X,I2,5X,E17.8, 10X,E17.3,15X,I3)
C
      WRITE(NOT,630) FRET
  630 FORMAT(///, 10X,16HRETRACT FORCE = , E17.8)
C
      WRITE(NOT,631) FI
  631 FORMAT(/, 10X, 24HBINDING FRICTION FORCE =, E17.8)
C
      WRITE(NOT,640) (FANDM(I),I=1,12)
  640 FORMAT(///,10X,16H FORCE AT PT D =, 3(2X,E17.8),/,
     *            10X,16HMOMENT AT PT D =, 3(2X,E17.8),/,
     *            10X,16H FORCE AT PT P =, 3(2X,E17.8),/,
     *            10X,16HMOMENT AT PT P =, 3(2X,E17.8))
C
      CALL CPWMS (CP)
C
      WRITE (NOT,635) CP
  635 FORMAT(///,30X,18HELAPSED CP TIME = , E17.8)
C
      IF (IFPER .NE. 0) CALL SETPER
C
      KOUNT = 0
  100 CONTINUE
C
      IF (IFPLOT .NE. 0)  CALL SETPLT(CP)
C
      CALL SETTPE
      IF (T.GE.ENDT .AND. IFPNCH.NE.0)
     *                  CALL PUNCH (Y,1,NEQTN,6HYINITL,1)
C
      RETURN
      END
```

```
      SUBROUTINE RETRAK
C
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FRET
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      IF (MODE(8) .EQ. 0)  TSTAR = T + TLAG
      IF (T - TSTAR .LE. 0.0)  GO TO 10
C
      AKD = AKD1
      IF (Y(NRHO) .LT. RHOSO) AKD = AKD2
      FRET = AKD *(T - TSTAR)/ABS(AKD3 + EXTL - Y(NRHO))
      IF (FRET .GT. FRETL)  FRET = FRETL
C
      GO TO 20
C
   10 FRET = 0.0
C
   20 RETURN
C
      END


      SUBROUTINE ROOTS (DA,DL,R,RCTNA,PHIC,FLC,FGC,NRK,NPA)
C
      DIMENSION DA(3), DL(3)
C
      DATA GESLAM, TOLER, MAXITR / 50.0, 0.01, 150 /
      DATA NIT, NOT  /  5, 6  /
C
      FLC = GESLAM
      IC = 0
   50 G = FLC
      PHIC = ATAN2(DA(3)+G*DL(3), DA(2)+G*DL(2))
      RCP = R*COS(PHIC)
      RSP = R*SIN(PHIC)
      G1 = (RCTNA-DA(1))*DL(1)+(RCP-DA(2))*DL(2)+(RSP-DA(3))*DL(3)
      IF (ABS(G-G1) .LT. TOLER)  GO TO 60
      IC = IC + 1
      IF (IC .EQ. MAXITR)  GO TO 200
      FLC = G1
      GO TO 50
C
   60 QR1 = RCTNA - DA(1) - G*DL(1)
      QR2   = RCP - DA(2) - G*DL(2)
      QR3   = RSP - DA(3) - G*DL(3)
      FGC = SQRT(QR1*QR1 + QR2*QR2 + QR3*QR3)
C
      RETURN
C
  200 WRITE(NOT,1001)
 1001 FORMAT(/,10X,45HSUBROUTINE ROOTS  DOES NOT CONVERGE, STOP RUN)
      STOP
C
      END
```

```
      SUBROUTINE RUNKTA
C
C   SUBROUTINE TO NUMERICALLY INTEGRATE A SET OF FIRST ORDER
C   DIFFERENTIAL EQUATIONS
C
C   REFERENCE FOR GILL-RUNGE-KUTTA ALGORITHM  ---
C        MATHEMATICAL METHODS FOR DIGITAL COMPUTERS, ANTHONY RALSTON,
C        HERBERT S. WILF (ART. BY MICHAEL J. ROMANELLI, PAGE 115).
C
C   CODED BY CARL BODLEY, JULY 1969
C
C   SUBROUTINES THAT ARE CALLED ARE --- GEOM, KINMAT, CONTAK AND [illegible]Y.
C
      COMMON /JCOUNT/ JIL
      COMMON /NEQTNS/ NEQTN
      COMMON /QPRKTA/ QRK(150), PRK(4)
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /VECTOR/ Y(150),YDT(150)
C
      DATA NT / 0 /
C
      DO 120 J = 1,4
      JIL = J
      DO 110 I=1,NEQTN
      Z = YDT(I)*DELTAT
      GO TO (103,101,101,105), JIL
  101 R = PRK(JIL)*(Z - QRK(I))
      GO TO 107
  103 R = PRK(JIL)*Z - QRK(I)
      GO TO 107
  105 R = (Z - 2.*QRK(I))/6.
  107 Y(I) = Y(I) + R
  110 QRK(I) = QRK(I) + 3.*R - PRK(JIL)*Z
      IF (JIL .EQ. 1 .OR. JIL .EQ. 3) T = T + DELTAT/2.
      CALL GEOM
      CALL KINMAT
      IF (JIL .EQ. 4) CALL CONTAK
  120 CALL VECTR
C
      NT = NT + 1
      ANT = NT
      TMST = ANT*DELTAT
      T = STARTT + TMST
C
      RETURN
      END
```

```
      SUBROUTINE SCALER (V,H,N, XSTART,SCALE)
C
      DIMENSION V(1),H(1)
C
      DATA   NIT,NOT   /5,  6/
C
 1001 FORMAT (54H1HORIZONTAL SCALE COULD NOT BE SHIFTED IN 10000 MOVES.
     *       //9H HMIN  = E15.8 /
     *         9H HMAX  = E15.8 /
     *         9H SCALE = E15.8 )
C
      VMAX = V(1)
      VMIN = V(1)
      HMAX = H(1)
      HMIN = H(1)
      DO 10 I=1,N
      IF (V(I) .LT. VMIN)  VMIN = V(I)
      IF (V(I) .GT. VMAX)  VMAX = V(I)
      IF (H(I) .LT. HMIN)  HMIN = H(I)
   10 IF (H(I) .GT. HMAX)  HMAX = H(I)
C
      CALL PLOTSS (VMAX,VMIN,VTOP,VBOT)
      CALL PLOTSS (HMAX,HMIN,HTOP,HBOT)
      SCALE = (VTOP-VBOT)/10.000
      HS    = (HTOP-HBOT)/10.000
      IF (HS .GT. SCALE)  SCALE = HS
      XSTART = -5.*SCALE
      XS     = XSTART
C
      I = 0
   15 I = I+1
      IF (I .GT. 10000)  GO TO 999
      IF (HMIN .GE. XSTART)  GO TO 20
      XSTART = XS - FLOAT(I)*SCALE
      GO TO 15
C
   20 I = 0
   25 I = I+1
      IF (I .GT. 10000)  GO TO 999
      IF (HMAX .LE. XSTART+10.*SCALE)  RETURN
      XSTART = XS + FLOAT(I)*SCALE
      GO TO 25
C
C
  999 WRITE(NOT,1001)  HMIN,HMAX,SCALE
      STOP
      END
```

```
      SUBROUTINE SETPER
C
C          SUBROUTINE TO COMPUTE PROBE POINT LOCATIONS AND
C                 PLOT FRAME IN PERSPECTIVE OR STEREO
C
C                 CODED BY BODLEY/PARK    OCTOBER 1969
C
      COMMON /ANGLES/ ALPHAD,SALPD,CALPD,TALPD, SSIGMA(3),CSIGMA(3)
      COMMON /FUNGAM/ SGAMMA(3),CGAMMA(3),SGSS(3),SGCS(3),
     *                                 CGSS(3),CGCS(3)
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTO(3),CPCO(3),RADCON,CSPHER,ROTALP,PEO,FDO
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /JBATCH/ JBA
      COMMON /LOCPTD/ DSD(3),DHD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /LOCPTI/ OTI(3), OCI(3), TCI(3)
      COMMON /LOCPTP/ PSP(3),PAP(3,3),PQP(3,3)
      COMMON /LOCPTT/ THT(3),TCT(3),TST(3),TAT(3,3),TRT(3,3),
     *                CST(3),CAT(3,3),    QRT(3,3), TDT(3),
     *                DHT(3),DGT(3),DST(3), DAT(3,3),DQT(3,3),DRT(3,3)
      COMMON /MISCNO/ ANUM
      COMMON /PERTIT/ PTITL(8),IFFIXD,RANGLE,EED,CANGLE,
     *                COELOC(3),VPLOC(3)
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECTOR/ Y(150),YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION       DPCD(3),DFCD(3),DBBD(3),DDD(3,3),DED(3,3),ODI(3)
      DIMENSION WV(3), VW(3), ILOC(50)
      DIMENSION SHP(186,3),RS(2),HS(2),FINOT(2)
      DIMENSION TRDI(3,3),OVI(3),OEI(3)
      DIMENSION CLOC(333,3),MLOC(345,2)
      DIMENSION COEYLC(3), VPNTLC(3)
C
      DATA   NIT, NOT    / 5, 6  /
      DATA INTL /0/
      DATA   (ILOC(I),I=1,50)
     *   / 1, 4, 4, 4, 6, 7, 8, 6, 7, 8,21,22,23,12,13,14,
     *     9,10,11, 2, 2, 2, 3, 3, 3,
     *     5, 6, 7, 8, 7, 8, 6,21,22,23,18,19,20,15,16,17,
     *    18,19,20, 9,10,11,12,13,14  /
C
      IF (JBA.GT.1) GO TO 98
      IF (INTL .NE. 0) GO TO 98
      READ (NIT,1001) (PTITL (I),I=1,8)
 1001 FORMAT (8A10)
      READ (NIT,1002)  IFFIXD,RANGLE,EED,CANGLE
 1002 FORMAT (I5,5X,6E10.0)
      READ (NIT,1003) (COELOC(I),I=1,3)
      READ (NIT,1003) ( VPLOC(I),I=1,3)
 1003 FORMAT (8E10.0)
   98 CONTINUE
      IF (INTL .NE. 0) GO TO 81
      DO 82 I=1,3
      COEYLC(I) = COELOC(I)
   82 VPNTLC(I) = VPLOC (I)
```

```
C
   81 IF (IFFIXD .NE. 0) GO TO 97
C ESTABLISH ODI, TRDI
      CALL ATXBV (TRTI,TDT,WV)
      CALL AVPBV (1.0,OTI,1.0,WV,ODI)
      CALL AXB3  (TRDT,TRTI,TRDI)
   97 CONTINUE
C
      IF (INTL .NE. 0) GO TO 99
C
      DO 90 I=1,25
      MLOC(I,1) = ILOC(I)
   90 MLOC(I,2) = ILOC(I+25)
C
      IF (IFFIXD .NE. 0) GO TO 96
C ESTABLISH OVI, OEI
      CALL ATXBV (TRDI,COEYLC,WV)
      CALL AVPBV (1.0,ODI,1.0,WV,OEI)
      CALL ATXBV (TRDI,VPNTLC,WV)
      CALL AVPBV (1.0,ODI,1.0,WV,OVI)
   96 CONTINUE
C
C   COMPUTE PROBE HEAD COORDINATES
      DLFI = ANUM*6.
      DO 10 L=1,2
      L1 = 1 + 31*(L-1)
      L2 = L1 + 30
      AL = L
      FI = ANUM*180.*(AL-1.)
      DO 10 I=L1,L2
      SHP(I,1) = 0.
      SHP(I,2) = CSPHER*COS(FI)
      SHP(I,3) = CSPHER*SIN(FI)
      FI = FI + DLFI
   10 CONTINUE
      L2 = 62
      FI =-135.*ANUM
      DO 20 J=1,4
      DO 20 L=1,2
      DLFI = 180*(L-1) - 135*(2-L)
      FI = FI + DLFI*ANUM
      L1 = L2 + 1
      L2 = L1 + 16 - L
      AL = 6*(1 + 14*(2-L))
      AL = ANUM*AL
      DLAL = 12*(L-1) - 6
      DLAL = ANUM*DLAL
      DO 20 I=L1,L2
      SHP(I,1) = CSPHER*COS(AL)
      SHP(I,2) = CSPHER*COS(FI)*SIN(AL)
      SHP(I,3) = CSPHER*SIN(FI)*SIN(AL)
      AL = AL + DLAL
   20 CONTINUE
C
C   COMPUTE DROGUE-CONE COORDINATES
      RS(1) = CSPHER*CALPD
      HS(1) = RS(1)/TALPD
      RS(2) = RADCON
      HS(2) = ROTALP
      FINOT(1) = 0.
      FINOT(2) = ANUM*180.
      DLFI = ANUM*6.
```

```
      L2 = 209
      DO 30 L=1,2
      DO 30 J=1,2
      FI = FINOT(J)
      L1 = L2 + 1
      L2 = L1 + 30
      DO 30 I=L1,L2
      CLOC(I,1) = HS(L)
      CLOC(I,2) = RS(L)*COS(FI)
      CLOC(I,3) = RS(L)*SIN(FI)
      FI = FI + DLFI
   30 CONTINUE
C
C   COMPUTE MLOC ARRAY
      IR = 25
      IS = -8
      DO 40 L=1,10
      IS = IS+31
      IT = IS
      DO 40 J=1,30
      IT = IT+1
      IR = IR+1
      MLOC(IR,1) = IT
      MLOC(IR,2) = IT+1
   40 CONTINUE
      L2 = 11
      IR = 325
      IS1 = 210
      IS2 = 272
      DO 50 L=1,2
      IS1 = IS1+34*(L-1)
      IS2 = IS2+34*(L-1)
      L2 = L2-2*(L-1)
      DO 50 I=1,L2
      I1 = IS1+3*(I-1)
      I2 = IS2+3*(I-1)
      IR = IR+1
      MLOC(IR,1) = I1
      MLOC(IR,2) = I2
   50 CONTINUE
C
      INTL = 1
   99 CONTINUE
C
      IF (IFFIXD .NE. 0) GO TO 95
C MODIFY COEVLC, VPNTLC
      CALL AVPBV (1.0,OEI,-1.0,ODI,WV)
      CALL AXBV  (TRDI,WV,COEVLC)
      CALL AVPBV (1.0,OVI,-1.0,ODI,WV)
      CALL AXBV  (TRDI,WV,VPNTLC)
   95 CONTINUE
C
C ESTABLISH DHPD
      CALL ATXBV (TRPD,PSP,WV)
      CALL AVPBV (1.0,DSD,-1.0,WV,DHPD)
C
      WV(2) = 0.0
      WV(3) = 0.0
C
C ESTABLISH DPCD
      WV(1) = DLP
      CALL ATXBV (TRPD,WV,VW)
```

```
      CALL AVPBV (1.0,DHPD,1.C,VW,DPCD)
C
C ESTABLISH DFCD
      WV(1) = DLF
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.0,VW,DFCD)
C
C ESTABLISH DBBD
      WV(1) = DLI+Y(NRHO)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.C,VW,DBBD)
C
C DSD COMPUTED BY GEOM
C DAD COMPUTED BY GEOM
C
      DO 100 K=1,3
C
C ESTABLISH DPD
      WV(1) = DLP
      WV(2) = -AP*SSIGMA(K)
      WV(3) =  AP*CSIGMA(K)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.C,VW,DPD(1,K))
C
C ESTABLISH DFD
      WV(1) = DLF
      WV(2) = -AF*SSIGMA(K)
      WV(3) =  AF*CSIGMA(K)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.0,VW,DFD(1,K))
C
C ESTABLISH DDD
      TRM = AA+DLD*SGAMMA(K)-AD*CGAMMA(K)
      WV(1) = DLI+Y(NRHO)-DLD*CGAMMA(K)-AD*SGAMMA(K)
      WV(2) = -TRM*SSIGMA(K)
      WV(3) =  TRM*CSIGMA(K)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.0,VW,DDD(1,K))
C
C ESTABLISH DED
      TRM = AA+DLE*SGAMMA(K)-AE*CGAMMA(K)
      WV(1) = DLI+Y(NRHO)-DLE*CGAMMA(K)-AE*SGAMMA(K)
      WV(2) = -TRM*SSIGMA(K)
      WV(3) =  TRM*CSIGMA(K)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.0,VW,DED(1,K))
C
C ESTABLISH DBD
      TRM = AA+DLB*SGAMMA(K)
      WV(1) = DLI+Y(NRHO)-DLB*CGAMMA(K)
      WV(2) = -TRM*SSIGMA(K)
      WV(3) =  TRM*CSIGMA(K)
      CALL ATXBV (TRPD,WV,VW)
      CALL AVPBV (1.0,DHPD,1.0,VW,DBD(1,K))
C
  100 CONTINUE
C
C ROW 1-23 OF CLOC
      DO 200 J=1,3
      CLOC( 1, J) = DHPD(J)
      CLOC( 2, J) = DPCD(J)
      CLOC( 3, J) = DFCD(J)
```

```
      CLOC( 4, J) = DBBD(J)
      CLOC( 5, J) =  DSD(J)
      DO 200 I=1,3
      CLOC( 5+I, J) = DAD(J,I)
      CLOC( 8+I, J) = DPD(J,I)
      CLOC(11+I, J) = DFD(J,I)
      CLOC(14+I, J) = DDD(J,I)
      CLOC(17+I, J) = DED(J,I)
      CLOC(20+I, J) = DBD(J,I)
  200 CONTINUE
C
C ROW 24-209 OF CLOC
      DO 210 I=1,186
      II = I+23
      DO 210 J=1,3
      CLOC(II,J) = DSD(J)
      DO 210 K=1,3
  210 CLOC(II,J)    CLOC(II,J) + SHP(I,K)*TRPD(K,J)
C
      CALL PLOT3 (CLOC,MLOC,COEYLC,VPNTLC,RANGLE,IFPER,0,IFPER,
     *            EED,CANGLE,PTITL ,333,345,333,345)
C
      RETURN
      END
```

```
      SUBROUTINE SETPLT(CPTNE)
C
C          SUBROUTINE TO WRITE DATA ON NTAPE1 SO THAT IT CAN
C          BE RECALLED FOR PLOTTING
C
C                CODED BY A C PARK    AUG 1969
C
      COMMON /BINDFR/ FI,BINDMU
      COMMON /CHANGE/ Z(150),ZDT(150)
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /FORMOM/ FANDM(12)
      COMMON /LATCHL/ VLAM(6)
      COMMON /LOCPTD/ DSD(3),DMD(3),DGD(3),DAD(3,3),DRD(3,3),
     *                DQD(3,3),QRD(3,3)
     *                ,DHPD(3),DBD(3,3),DFD(3,3),DPD(3,3)
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FRET
      COMMON /TAPED/ NTAPE1, NTAPE2, NTAPE3
      COMMON /TIMESS/ STARTT,CELTAT, T, ENDT, TMST, NSECS
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      IF (TMST .GT. 0.0)  GO TO 10
      LS1 = NUT
      LE1 = LS1+5
      LS2 = NUC
      LE2 = LS2+5
      LS3 = NRHO
      LE3 = LS3+5
      N1 = NDELCS+1
      N2 = NDELCS+2
      N3 = NDELCS+3
      N4 = NDELCS+19
      N5 = NDELCS+20
      N6 = NDELCS+21
   10 CONTINUE
C
      WRITE (NTAPE1) T,
     *                  (Z(I),I=LS1,LE1),(Z(I),I=LS2,LE2),
     *                  (Z(I),I=LS3,LE3),
     *                  (DSD(I),I=1,3),
     *                  (ALAM(I),I=1,10), (VLAM(I),I=1,6),
     *                  (FANDM(I),I=1,6),
     *                   Z(N1),Z(N2),Z(N3),Z(N4),Z(N5),Z(N6),
     *                  (FORCS(I),I=2,10)
C
      RETURN
      END
      SUBROUTINE SETTPE
C
C               CODED BY A C PARK     SEPT 1969
C
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /FORMOM/ FANDM(12)
      COMMON /LATCHL/ VLAM(6)
      COMMON /TAPED/ NTAPE1, NTAPE2, NTAPE3
      COMMON /TIMESS/ STARTT,CELTAT, T, ENDT, TMST, NSECS
C
      WRITE (NTAPE3)  T, (FANDM(I),I=1,6), (FORCS(I),I=2,7),
     *                (VLAM(I),I=1,6), (FANDM(I),I=7,12),
     *                FORCS(1),  (FORCS(I),I=8,10)
C
      IF (T.GE.ENDT) ENDFILE NTAPE3
C
      RETURN
      END
```

```
      PROGRAM SETUP
C
C   CODED BY CARL BODLEY, FRED NERZ AND COLT PARK, JULY 1969.
C
      DIMENSION PTITLE(4), SIGMA(3)
C
      COMMON /ANGLES/ ALPHAD,SALPD,CALPD,TALPD, SSIGMA(3),CSIGMA(3)
      COMMON /BINDFR/ FI,BINDMU
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /GEOMTY/ AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,
     *                TDTD(3),CPCD(3),RADCON,CSPHER,ROTALP,PED,FDD
      COMMON /IFOUTS/ MINIC,IFPLOT,IFPER
      COMMON /JBATCH/ JBA
      COMMON /JCOUNT/ JIL
      COMMON /MASS/  AMT, AMC, AIT(3,3), AIC(3,3)
      COMMON /MFREQD/ FREQT(26), FREQC(26), ZETAT, ZETAC
      COMMON /MISCNO/ ANUM
      COMMON /NCNTRL/ NCNSYS
      COMMON /NEQTNS/ NEQTN
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PLTR  / K1,NCOLS,NTIMES,XDLTA,NPLTS
      COMMON /PUNCHY/ IFPNCH
      COMMON /QPRKTA/ QRK(150), PRK(4)
      COMMON /RETRCT/ AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,TLAG,TSTAR,FRET
      COMMON /SLIDFR/ HDMU, ARMMU, BFV(8,39)
      COMMON /SPRDMP/ FSPRNG(2,20,4),FDAMP(2,20,4)
      COMMON  /TAPED/ NTAPE1, NTAPE2, NTAPE3
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /VECTOR/ Y(150), YDT(150)
C
      DATA NIT,NOT/5,6/
      DATA NTMHST  /58/
      DATA  NPLTS /500/
      DATA KTABLE  /20/
C
 1001 FORMAT (16I5)
 1003 FORMAT (2I5,2X,I1,2X,2(A6,4X),4A10)
 1004 FORMAT (8E10.0)
 2001 FORMAT (////15X,40H THE INPUT SCALARS TO PROGRAM DOCKEL ARE ,
     *         //23X,       10H STARTT = F10.6,
     *          /23X,       10H DELTAT = F10.6,
     *          /23X,       10H ENDT   = F10.6,
     *          /23X,       10H XDLTA  = F10.6,
     *          /23X,       10H XPRNT  = F10.6,
     *          /23X,       10H ZETAT  = F10.6,
     *          /23X,       10H ZETAC  = F10.6,
     *          /23X,       10H IFELAS = I5,
     *          /23X,       10H IFPLOT = I5,
     *          /23X,       10H IFPER  = I5,
     *          /23X,       10H JBATCH = I5,
     *          /23X,       10H MINIC  = I5,
     *          /23X,       10H IFPNCH = I5     )
 2005 FORMAT (     23X,     10H NCOLS  = I5,
     *          /23X,       10H NCNSYS = I5,
     *          /23X,       10H NFORCS = I5,
     *          /23X,       10H NEQTN  = I5,
     *          /23X,       10H NMODET = I5,
     *          /23X,       10H NMODEC = I5    )
 2002 FORMAT  (// 23X,
     * 10H AA      = E15.8,/23X,10H AB      = E15.8,/23X,
     * 10H AD      = E15.8,/23X,10H AE      = E15.8,/23X,
     * 10H AF      = E15.8,/23X,10H AP      = E15.8,/23X,
```

```
     * 10H DLB    = E15.8,/23X,10H DLD    = E15.8,/23X,
     * 10H DLE    = E15.8,/23X,10H DLF    = E15.8,/23X,
     * 10H DLI    = E15.8,/23X,10H DLP    = E15.8,/23X,
     * 10H DLQ    = E15.8,/23X,
     * 10H DLS    = E15.8,/23X,10H FDO    = E15.8,/23X,
     * 10H PEO    = E15.8,/23X,10H ALPHAD = E15.8,/23X,
     * 10H CSPHER = E15.8,/23X,10H RADCON = E15.8,/23X,
     * 10H AKD1   = E15.8,/23X,10H AKD2   = E15.8,/23X,
     * 10H AKD3   = E15.8,/23X,10H RHOSO  = E15.8,/23X,
     * 10H FRETL  = E15.8,/23X,10H EXTL   = E15.8,/23X,
     * 10H TLAG   = E15.8,/23X,10H TSTAR  = E15.8,/23X,
     * 10H BINDMU = E15.8,/23X,
     * 10H HDMU   = E15.8,/23X,10H ARMMU  = E15.8          )
 2003 FORMAT (////// 25X,30HTHE TARGET VEHICLES MASS IS   E15.8)
 2004 FORMAT (////// 25X,30HTHE CHASE  VEHICLES MASS IS   E15.8)
C
 3001 FORMAT (24H1ERROR CHECK AT NERROR = I3, 8H FAILED,
     *        /17H0PROGRAM STOPPED.)
 3002 FORMAT (52H1MECHANISM SPRING,DAMPING OR FRICTION NOT COMPATIBLE/
     *          17H0PROGRAM STOPPED.)
C
      REWIND NTAPE1
C
      IF (JBA .GT. 1) GO TO 50
C
      REWIND NTAPE2
      REWIND NTAPE3
C
      READ (NIT,1001) MINIC,IFPLOT,IFPER,IFPNCH
      READ (NIT,1001) NCNSYS,NFORCS
C
      READ (NIT,1004) STARTT,DELTAT,ENDT,XDLTA,XPRNT
      NSECS = XPRNT
      NTIMES = (ENDT-STARTT)/DELTAT + 1.1
C
      READ (NIT,1004)
     *  AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,FDO,PEO,
     *  ALPHAD,CSPHER,RADCON,AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,
     *  TLAG,TSTAR,BINDMU,HDMU,ARMMU
C
      CALL READ   (TDTO ,N1,   N2, 1, 3)
      CALL READ   (CPCO ,N1,   N2, 1, 3)
C
      ANUM = ATAN2(1.,1.)/45.
      SALPD = SIN(ANUM*ALPHAD)
      CALPD = COS(ANUM*ALPHAD)
      TALPD = SALPD/CALPD
      ROTALP = RADCON/TALPD
      CALL READ   (SIGMA,N1,N2,1,3)
      DO 10 I=1,3
      SSIGMA(I) = SIN(ANUM*SIGMA(I))
   10 CSIGMA(I) = COS(ANUM*SIGMA(I))
C
      READ (NIT,1004)  AMT
      CALL READ (AIT, N1, N2, 3, 3)
      WRITE (NOT,2003) AMT
      READ (NIT,1004)  AMC
      CALL READ (AIC, N1, N2, 3, 3)
      WRITE (NOT,2004) AMC
C
      DO 15 I=1,4
      CALL READ   (FSPRNG(1,1,I),N1,N2,2,KTABLE)
```

```
      NFSPR = FSPRNG(1,1,I) + 1.1
      IF (NFSPR .NE. N2) GO TO 900
      DO 17 J=3,N2
      JM1 = J - 1
      IF (FSPRNG(1,J,I) .LE. FSPRNG(1,JM1,I)) GO TO 900
   17 CONTINUE
      CALL READ   (FDAMP(1,1,I),N1,N2,2,KTABLE)
      NFDMP = FDAMP(1,1,I) + 1.1
      IF (NFDMP .NE. N2) GO TO 900
      DO 18 J=3,N2
      JM1 = J - 1
      IF (FDAMP(1,J,I) .LE. FDAMP(1,JM1,I)) GO TO 900
      IF (FDAMP(2,J,I) .LE. FDAMP(2,JM1,I)) GO TO 900
   18 CONTINUE
   15 CONTINUE
C
   25 CALL HSIG
C
      NCOLS = NTHHST
      NCOLP1 = NCOLS + 1
C
      CALL LOCATE
C
   50 IF (MINIC .EQ. 0)  CALL READ(Y,N1,N2,1,150)
      IF (MINIC .EQ. 1)  CALL MINYS
C
      IF (JBA.GT.1) GO TO 90
      IF (IFPLOT.EQ.0)  GO TO 90
      NERROR = 0
      LIMPLT = NPLTS *NCOLP1
      IF (LIMPLT .GT. K1) GO TO 901
  100 READ (NIT,1003) NCX,NCY,NXPL,XNAME,YNAME,PTITLE
      NERROR = NERROR + 1
      IF (NCX .GT. NCOLP1 .OR. NCY .GT. NCOLP1 )  GO TO 901
      WRITE (NTAPE2) NCX,NCY,NXPL,XNAME,YNAME,PTITLE
      IF (NCX.NE.0)  GO TO 100
C
C  PRINT INPUT SCALARS.
   90 CALL PAGEHD
      WRITE (NOT,2001) STARTT,DELTAT,ENDT,XDLTA,XPRNT,ZETAT,ZETAC,
     *                 IFELAS,IFPLOT,IFPER,JBA,MINIC,IFPNCH
      WRITE (NOT,2005) NCOLS,NCNSYS,NFORCS,NEQTN,NMODET,NMODEC
      WRITE (NOT,2002)
     *  AA,AB,AD,AE,AF,AP,DLB,DLD,DLE,DLF,DLI,DLP,DLQ,DLS,FDO,PEO,
     *  ALPHAD,CSPHER,RADCON,AKD1,AKD2,AKD3,RHOSO,FRETL,EXTL,
     *  TLAG,TSTAR,BINDMU,HDMU,ARMMU
C
C  SET INTEGRATION CONSTANTS.
      PRK(1) = .5
      PRK(2) = 1. - SQRT(.5)
      PRK(3) = 1. + SQRT(.5)
      PRK(4) = .5
C
C  SET INITIAL VALUES.
      JIL = 4
      T = STARTT
      TMST = 0.0
      DO 30 I=1,NEQTN
   30 QRK(I) = 0.
C
      RETURN
  900 WRITE (NOT,3002)

      STOP
  901 WRITE (NOT,3001) NERROR
      STOP
      END
```

```
      SUBROUTINE  SKEWV (VEC,SKV)
      DIMENSION   VEC(3), SKV(3,3)
C
C   CODED BY CARL BODLEY   JUNE  1968
C
      SKV(2,3) =  VEC(1)
      SKV(3,1) =  VEC(2)
      SKV(1,2) =  VEC(3)
      SKV(3,2) = -SKV(2,3)
      SKV(1,3) = -SKV(3,1)
      SKV(2,1) = -SKV(1,2)
      SKV(1,1) = 0.0
      SKV(2,2) = 0.0
      SKV(3,3) = 0.0
      RETURN
      END
```

```
      PROGRAM TIMHST
C
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
C
      CALL GEOM
      CALL KINMAT
      CALL CONTAK
      CALL VECTR
      CALL FORFUN
      CALL PREENT
    1 CALL RUNKTA
      CALL FORFUN
      CALL PREENT
      IF (T .GE. ENDT) RETURN
      GO TO 1
      END
```

```
      SUBROUTINE VECTR
C
C                 COMPUTES TIME DERIVATIVE OF STATE VECTOR
C
      COMMON /BC1234/ BC1(10,6), BC2(10,6), BC3(10,26), BC4(10,26)
      COMMON /BTDATA/ TRHO(9,6), BL5(10,6), BLV(7,39)
      COMMON /CSFORC/ FORCS(10), NFORCS
      COMMON /JCOUNT/ JIL
      COMMON /MASS/  AMT, AMC, AIT(3,3), AIC(3,3)
      COMMON /MFREQD/ FREQT(26), FREQC(26), ZETAT, ZETAC
      COMMON /MISCNO/ ANUM
      COMMON /NEQTNS/ NEQTN
      COMMON /NMODES/ NMODET, NMODEC, IFELAS
      COMMON /PHILAM/ PHICON(10),ALAM(10), MODE(10)
      COMMON /SLIDFR/ HDMU, ARMMU, BFV(8,39)
      COMMON /TIMESS/ STARTT,DELTAT, T, ENDT, TMST, NSECS
      COMMON /TRNFMS/ TRTI(3,3),TRCI(3,3),TRTC(3,3),TRHT(3,3),TRHC(3,3),
     *                TRRT(3,3,3),TRRC(3,3,3),TRPC(3,3),TRDT(3,3),
     *                TRPD(3,3),TRHD(3,3),TRRD(3,3,3),TRPT(3,3)
     *                ,TRHP(3,3),TRTP(3,3),TRRP(3,3,3)
      COMMON /VECTOR/ Y(150), YDT(150)
      COMMON /YPSTNS/ NUT,NXT,NGAMT,NUC,NXC,NGAMC,NRHO,
     *                NXITDT,NXIT,NXICDT,NXIC, NDELCS
C
      DIMENSION GAMDTT(3,3),GAMDTC(3,3),OMEGT(3,3),OMEGC(3,3)
      DIMENSION  AITI(3,3), AICI(3,3), TZOMT(26), OMSQT(26),
     1                                 TZOMC(26), OMSQC(26)
      DIMENSION  V1(6), V2(6), V3(6), V4A(3), V4B(3), V5A(3), V5B(3),
     1           V26A(26), V26B(26), T33(3,3)
      DIMENSION VEL(8),FOR(8),BIGF(39)
      DATA ILATCH,ROV/ 0, 1.0 /
C
      IF (TMST .GT. 0.0) GO TO 5
C
      CALL INV2NP(AIT, AITI, 3, 3)
      CALL INV2NP(AIC, AICI, 3, 3)
      CALL ZERO (BIGF,39,1,39)
C
      PI    = 180.*ANUM
      FPI   = 4.*PI
      FPISQ = 4.*PI**2
C
      FPIZT = FPI*ZETAT
      FPIZC = FPI*ZETAC
C
      DO 2 I=1,NMODET
      TZOMT(I) = FPIZT*FREQT(I)
    2 OMSQT(I) = FPISQ*FREQT(I)**2
C
      DO 4 I=1,NMODEC
      TZOMC(I) = FPIZC*FREQC(I)
    4 OMSQC(I) = FPISQ*FREQC(I)**2
C
      IST = 36+2*NMODET
      KD = 12+NMODET+NMODEC
C
C
      NOMT = NUT + 3
      NOMC = NUC + 3
    5 CONTINUE
C
      IF (Y(NRHO) .LE. ROV .AND. JIL .EQ. 4 .AND.
```

```
     *                   MODE(8) .EQ. 1) ILATCH = 1
      IF (ILATCH .EQ. 1) CALL HLATCH(1)
      CALL SKEWV (Y(NOMT), OMEGT)
      CALL SKEWV (Y(NOMC), OMEGC)
C
      CALL ATXBV (TRTI, Y(NUT), YDT(NXT))
      CALL ATXBV (TRCI, Y(NUC), YDT(NXC))
C
      CALL AXB3 (OMEGT, TRTI, GAMDTT)
      CALL AXB3 (OMEGC, TRCI, GAMDTC)
C
      DO 30 I=1,2
      IT = 3*I+NGAMT-4
      IC = 3*I+NGAMC-4
      DO 30 J=1,3
      YDT(IT+J) = GAMDTT(I,J)
   30 YDT(IC+J) = GAMDTC(I,J)
C
      CALL LAMRHO
C
      IF (HDMU.EQ.0.0 .AND. ARMMU.EQ.0.0) GO TO 90
C
      DO 100 I=1,8
      VEL(I) = 0.0
      CALL MULTAD(BFV(I,1),Y(NUT),VEL(I),1,6,1,8,150,8)
      CALL MULTAD(BFV(I,7),Y(NUC),VEL(I),1,6,1,8,150,8)
      CALL MULTAD(BFV(I,13),Y(NXITDT),VEL(I),1,NMODET,1,8,150,8)
      CALL MULTAD(BFV(I,13+NMODET),Y(NXICDT),VEL(I),1,NMODEC,1,8,150,8)
  100 CONTINUE
C
      NLAM = 3
      DO 200 I=1,7,2
      I1 = I+1
      NLAM = NLAM+1
      COF = HDMU
      IF (I.GT.1) COF = ARMMU
      VELMAG = SQRT(VEL(I)*VEL(I)+VEL(I1)*VEL(I1))
      FOR( I) = -COF*VEL( I)*ALAM(NLAM)/VELMAG
  200 FOR(I1) = -COF*VEL(I1)*ALAM(NLAM)/VELMAG
C
      CALL MULT (FOR,BFV,BIGF,1,8,KD,1,8)
C
   90 CONTINUE
C
      CALL CONSYS
C
      CALL MULT (FORCS,BC1,V1,1,NFORCS,6,1,10)
      CALL MULT (ALAM(4),BLV(1,1),V3,1,7,6,1,7)
      CALL AXB3 (OMEGT,AIT,T33)
      CALL AXBV (OMEGT,Y(NUT),V2(1))
      CALL AXBV (T33,Y(NOMT),V2(4))
C
      DO 60 I=1,3
      I3 = I+3
      V4A(I) = V2(I) + (V1(I)+V3(I)+BIGF(I))/AMT
   60 V4B(I) = V1(I3) + V2(I3) + V3(I3) + BIGF(I3)
C
      CALL AXBV (AITI,V4B,V5B)
C
      DO 61 I=1,3
      I3 = I+3
      YDT( I) = V4A(I)
```

```
   51 YDT(I3) = V5B(I)
C
      CALL MULT (FORCS,BC3,V26A,1,NFORCS,NMODET,1,10)
      CALL MULT (ALAM(4),BLV(1,13),V26B,1,7,NMODET,1,7)
C
      DO 62 I=1,NMODET
      II = I+36
      IJ = II+NMODET
      YDT(II) = V26A(I) + V26B(I) - TZOMT(I)*Y(II) - OMSQT(I)*Y(IJ)
     *          + BIGF(I+12)
   62 YDT(IJ) = Y(II)
C
      CALL MULT (FORCS,BC2,V1,1,NFORCS,6,1,10)
      CALL MULT (ALAM(4),BLV(1,7),V3,1,7,6,1,7)
      CALL AXB3 (OMEGC,AIC,T33)
      CALL AXBV (OMEGC,Y(NUC),V2(1))
      CALL AXBV (T33,Y(NOMC),V2(4))
C
      DO 63 I=1,3
      I3 = I+3
      V4A(I) = V2(I) + (V1(I) + V3(I)+BIGF(I+6))/AMC
   63 V4B(I) = V1(I3) + V2(I3) + V3(I3) + BIGF(I+9)
C
      CALL AXBV (AICI,V4B,V5B)
C
      DO 64 I=1,3
      I15 = I+15
      I18 = I+18
      YDT(I15) = V4A(I)
   64 YDT(I18) = V5B(I)
C
      CALL MULT (FORCS,BC4,V26A,1,NFORCS,NMODEC,1,10)
      CALL MULT (ALAM(4),BLV(1,13+NMODET),V26B,1,7,NMODEC,1,7)
C
      DO 65 I=1,NMODEC
      II = I+IST
      IJ = II+NMODEC
      YDT(II) = V26A(I) + V26B(I) - TZOMC(I)*Y(II) - OMSQC(I)*Y(IJ)
     *          + BIGF(I+12+NMODET)
   65 YDT(IJ) = Y(II)
C
      IF (ILATCH .EQ. 1) CALL HLATCH(2)
      RETURN
      END
```

APPENDIX D

PROGRAM DOCKEL INPUT DATA

This Appendix presents a listing of required DOCKEL input data. These data are for the Baseline Configuration and were used to generate the results presented in Chapter II. Numerical output obtained from these data are shown in Appendix E.

```
SC-1
NAS8-21280
BASELINE CONFIGURATION
    1    1    1    0
   34   10
    0.        0.01       10.       0.5       50.
   2.25       .85        .708      .965      3.095     4.25     14.48     12.90
   17.59     4.75       14.10      3.30      16.53     2.055     11.5      10.50
    42.75     2.58       13.56     0.0        3.0      5.41      2.08    1000.0
   11.12     10.15        0.0      0.28       .26       .26
TOTO      1    3
    1    1            407.88            0.8309            36.167
0000000000
CPCO      1    3
    1    1     134.7           -7.7                   -7.7
0000000000
SIGMA     1    3
    1    1     180.            300.              60.
0000000000
 296.4000
AIT       3    3              INERTIA FOR CLUSTER ONLY
    1    1        +6.428E+06       +1.622E+04        +3.528E+06
    2    1        +1.622E+04       +2.474E+07        -1.575E+05
    3    1        +3.528E+06       -1.575E+05        +2.512E+07
0000000000
  78.8577
AIC       3    3
    1    1     195240.          16956.           16932.
    2    1      16956.         539016.          -79212.
    3    1      16932.         -79212.          539028.
0000000000
FSPROX    2    8
    1    1      7.              0.               0.32             .87
    1    5      2.62            8.12             11.1            11.12
    2    1      0.          -2000.             -125.           -118.
    2    5    -106.2           -81.             -69.            731.0
0000000000
FDMROX    2    5
    1    1    4.0              -12.              -0.1             0.1
    1    5    12.0
    2    1    0.0              -92.              -91.             91.
    2    5    92.
0000000000
FSPALL    2    19
    1    1    19.0              -0.0708          -0.06605         -0.0587
    1    5    -0.05135          -0.0367          -0.02936         -0.02202
    1    9    -0.01468          -0.00734          0.00734          0.01468
    1   13     0.02202           0.02936          0.03670          0.05135
    1   17     0.05870           0.06605          0.07080
    2    1     0.0              -81.75     +03   -72.3      +03  -59.15     +03
    2    5   -49.1        +03   -33.22     +03   -26.17     +03  -19.5      +03
    2    9   -12.94       +03    -6.256    +03     6.256    +03   12.94     +03
    2   13    19.5        +03    26.17     +03    33.22     +03   49.10     +03
    2   17    59.15       +03    72.30     +03    81.75     +03
0000000000
FDMALY    2    3
    1    1    2.0               0.0              1.0
    2    1    0.0               0.0              1.0       -06
0000000000
FSPALU    2    19
    1    1    18.0              -0.0708          -0.06605         -0.0587
    1    5    -0.05135          -0.0367          -0.02936         -0.02202
```

```
    1    9   -0.01468            -0.00734             0.00734             0.01468
    1   13    0.02202             0.02936             0.0367              0.05135
    1   17    0.05870             0.06605             0.0708
    2    1    0.0                -81.75     +03      -65.15      +03    -47.0       +03
    2    5  -36.0       +03      -22.2      +03      -17.18      +03    -12.28      +03
    2    9   -8.31      +03      -4.085     +03       4.085      +03      8.31      +03
    2   13   12.28      +03       17.18     +03       22.2       +03     36.0       +03
    2   17   47.0       +03       65.15     +03       81.75      +03
0000000000
FOMALZ  2    3
    1    1    2.0                 0.0                 1.0
    2    1    0.0                 0.0                 1.0       -06
0000000000
FSPATT  2   10
    1    1    9.0                 8.43                8.5                 9.0
    1    5    9.5                10.0                10.5                11.0
    1    9   11.25               11.50
    2    1    0.0               -360.0              -305.0              -185.0
    2    5 -123.5               -86.5               -62.5               -45.5
    2    9  -38.5               -32.5
0000000000
FDMATT  2   16 ATTEN. DAMPING (T=TEMP= +70 THRU +250 DEG F )
    1    1   15.0               -19.2               -18.25              -16.8
    1    5  -14.4               -12.0                -9.6                -7.2
    1    9   -4.8                -2.76               -1.20               -0.60
    1   13   -0.24               -0.12                0.12                1.32
    2    1    0.               -2980.              -2685.              -2450.
    2    5 -2170.              -1920.              -1680.              -1435.
    2    9 -1175.               -910.               -630.               -405.
    2   13  -210.                -18.                 18.                 78.
0000000000
    1   14    1      0.01        0.01
FREQ     1   96
    1    5   0.                0.                2.54996194E-01    2.56654233E-01
    1    9   2.59230144E-01    2.63558032E-01    2.66530243E-01    3.16363216E-01
    1   13   4.08638136E-01    4.08671735E-01    4.08675934E-01    4.08678470E-01
    1   17   4.08678774E-01    4.08875139E-01    5.10929768E-01    5.13945264E-01
    1   21   5.37976914E-01    5.38156352E-01    5.74922575E-01    5.81649117E-01
    1   25   5.86156855E-01    5.86943106E-01    6.17050186E-01    6.18100481E-01
    1   29   6.18829615E-01    6.18840578E-01    7.15880147E-01    8.57194549E-01
    1   33   8.76752892E-01    9.04248565E-01    9.05857860E-01    9.20249224E-01
    1   37   9.88036352E-01    1.02627491E+00    1.06010342E+00    1.09757768E+00
    1   41   1.11870753E+00    1.11875386E+00    1.11876104E+00    1.11876459E+00
    1   45   1.11876506E+00    1.11889359E+00    1.25392463E+00    1.259090 2E+00
    1   49   1.30060590E+00    1.30091512E+00    1.36216324E+00    1.36281889E+00
    1   53   1.36672613E+00    1.37554721E+00    1.37958881E+00    1.73465346E+00
    1   57   1.81008596E+00    1.81094561E+00    1.81107947E+00    1.82252375E+00
    1   61   2.11352028E+00    2.13162065E+00    2.13181872E+00    2.13471946E+00
    1   65   2.14928675E+00    2.78612394E+00    2.88460197E+00    2.90610104E+00
    1   69   3.31360816E+00    3.85170624E+00    4.12990059E+00    4.14176135E+00
    1   73   4.47363853E+00    4.68029637E+00    6.12480919E+00    6.33019624E+00
    1   77   6.65320423E+00    7.50817221E+00    7.95936849E+00    8.30724494E+00
    1   81   8.94704006E+00    9.38847045E+00    9.79588804E+00    1.00894674E+01
    1   85   1.03479378E+01    1.11819072E+01    1.17654872E+01    1.22070518E+01
    1   89   1.23384923E+01    1.25791755E+01    1.47431405E+01    1.75212818E+01
    1   93   2.52185445E+01    4.39829177E+01    5.28482614E+01    8.88177531E+01
0000000000
RTMODE   6   96
    1    1   3.37096783E-02    1.23539660E-02   -1.59449945E-02   -4.34111732E-04
    1    5   3.57812083E-02   -2.43781212E-02    2.35864798E-03    5.93297350E-04
    1    9  -4.80510784E-04    2.86124847E-06    5.95135453E-04   -7.72296766E-05
    1   13   7.54610732E-07    1.13132248E-06    1.95837124E-07    2.09693285E-07
```

1	17	7.93955160E-08	1.92548456E-05	6.17025205E-05	1.24977151E-04
1	21	2.01805904E-05	2.91987366E-05	5.86912477E-03	7.94923731E-05
1	25	2.46479748E-06	2.10084411E-04	2.30859386E-03	1.09895480E-04
1	29	3.06015576E-05	1.52113891E-06	2.02330550E-02	4.81375751E-03
1	33	3.97618063E-03	6.17778992E-06	7.71207804E-04	6.67019316E-03
1	37	4.81624098E-03	2.75076466E-03	5.85791268E-04	4.82502094E-04
1	41	5.95146487E-06	2.23030764E-06	4.75653357E-07	3.03218252E-07
1	45	1.25060419E-07	2.71882806E-05	8.50025176E-05	1.17011315E-04
1	49	-9.62123889E-07	-4.12844677E-06	1.07137173E-05	1.38935231E-04
1	53	2.60864398E-03	1.74962268E-04	6.04076638E-04	4.28118908E-03
1	57	3.15612193E-05	7.03016400E-06	4.08273876E-04	1.42170985E-05
1	61	1.68447020E-03	1.17703399E-03	3.47239825E-04	2.45452095E-03
1	65	7.24354186E-04	3.06177881E-05	-5.79596544E-04	2.26075309E-04
1	69	6.06747606E-06	-4.21192149E-04	5.05314330E-05	2.23933093E-04
1	73	-3.78903618E-04	1.02142122E-04	-2.56064876E-04	5.71295238E-05
1	77	-2.59390862E-04	-2.65493979E-04	8.67187204E-04	2.84514565E-03
1	81	4.75644775E-05	9.00861691E-04	7.73672058E-04	-6.13299226E-04
1	85	-1.98280055E-03	6.17674589E-02	4.15375434E-03	3.37380770E-04
1	89	2.04984887E-03	4.42937788E-04	2.78800581E-03	-2.08751501E-03
1	93	1.28418761E-03	7.42829220E-04	3.30279460E-04	1.27781896E-04
2	1	5.12305849E-02	-9.38024557E-02	3.02482483E-03	1.13541500E-02
2	5	-4.77465311E-03	1.85895868E-02	-7.12965330E-03	-6.77275097E-03
2	9	7.05480541E-03	-2.81927842E-06	-7.57665369E-03	-8.18171770E-03
2	13	-4.02767476E-05	-8.44746263E-05	-1.23409744E-05	-1.52574010E-05
2	17	-5.31044093E-06	-1.50324529E-03	1.57474057E-03	-1.28803286E-03
2	21	-1.06291004E-04	1.65694373E-04	-5.04402141E-05	-4.56511191E-03
2	25	4.74056828E-07	-2.48415143E-03	-3.15687166E-04	-1.63810216E-03
2	29	-7.52210968E-05	-9.35274382E-06	4.45609851E-03	8.98498502E-03
2	33	-2.27800082E-02	9.07101958E-07	-3.08334165E-03	9.25577209E-04
2	37	-1.38663972E-02	4.57833566E-02	1.64751749E-02	-1.54098939E-02
2	41	-8.01084343E-05	-2.16464105E-04	-2.56562958E-05	-3.64312972E-05
2	45	-1.13274068E-05	-1.72692110E-03	6.98141186E-04	-4.93337745E-04
2	49	-3.22884382E-05	4.00262597E-05	3.12055788E-05	-1.54029390E-03
2	53	4.64187617E-04	1.25919799E-02	2.05443035E-03	-6.26650314E-03
2	57	9.27369053E-05	1.19602774E-03	1.33604028E-03	-2.99914585E-04
2	61	9.38544874E-03	-7.50081000E-04	-2.44707813E-04	7.64415830E-04
2	65	1.39814868E-02	1.42695203E-04	-2.11763155E-03	6.86906506E-04
2	69	3.63629420E-03	-1.50813437E-02	1.82028141E-01	6.35892015E-02
2	73	-4.41588856E-03	-1.32801069E-02	-6.45130556E-02	-1.49810078E-01
2	77	7.31403070E-02	-3.95898658E-02	-3.28627534E-02	1.08226046E-02
2	81	-2.12644730E-02	3.16180286E-04	-7.44669028E-03	-2.23929926E-03
2	85	1.18435727E-02	-6.31011855E-04	1.28152957E-02	4.67712916E-03
2	89	-3.40669164E-03	-3.72410981E-02	4.36326396E-04	-7.24800995E-03
2	93	3.26548153E-03	-3.91671271E-04	-3.13290351E-04	1.00457022E-04
3	1	-2.11585247E-02	-1.46326584E-02	-9.31244414E-02	-2.01675988E-03
3	5	-3.60273501E-02	-8.58750492E-03	3.17591518E-03	6.77441643E-04
3	9	2.81078096E-03	-6.63618234E-06	-8.67791524E-03	2.05171220E-03
3	13	9.39534805E-06	2.45356634E-06	1.38114974E-06	[illegible].05995360E-07
3	17	2.79969117E-07	4.83999644E-05	-2.17516492E-04	-1.80126598E-04
3	21	-1.77733337E-05	-5.64411234E-05	5.21399193E-03	9.02113934E-04
3	25	-5.99149926E-06	-3.38981921E-03	2.19304817E-03	1.62777600E-04
3	29	-4.23221267E-04	-1.00276510E-05	1.82785419E-02	-2.80294086E-03
3	33	1.73807862E-03	-1.18390157E-05	-4.20914862E-03	5.91410353E-03
3	37	3.87151615E-03	-2.02442955E-03	5.71566533E-05	2.55949822E-03
3	41	1.30523402E-05	2.86476085E-05	3.38002610E-06	4.70551521E-06
3	45	1.43176900E-06	2.45000441E-04	-2.41877551E-04	1.56480255E-04
3	49	-1.29462024E-05	-5.17450306E-05	-4.44193955E-05	-1.70417615E-03
3	53	5.22249285E-04	5.47823067E-05	1.21926002E-02	-6.17484979E-03
3	57	-6.[illegible]2031620E-05	-4.52914706E-04	5.80743529E-05	-7.47069084E-04
3	61	4.23715659E-03	-5.84963413E-03	-1.88011752E-03	4.72411822E-03
3	65	3.91863542E-03	-1.28264833E-04	3.28906411E-03	7.26718985E-04
3	69	2.57811648E-03	8.73291314E-02	1.32490167E-02	5.52847365E-03

3	73	-3.47557168E-02	2.08432504E-01	-7.26947275E-02	-3.65157930E-02
3	77	-1.39919243E-01	-2.07244730E-02	-9.99155049E-04	1.42645257E-02
3	81	-3.53340564E-03	-1.97234480E-02	-5.38536789E-03	-5.58437218E-04
3	85	1.23432852E-03	-5.92216145E-04	-8.87614287E-04	8.76745164E-05
3	89	3.34002997E-02	-3.81946911E-03	-2.48992180E-03	-1.09488268E-02
3	93	-2.80716917E-02	-1.03347817E-02	-6.25110713E-03	-2.50590566E-03
4	1	-8.93415952E-05	1.55241722E-04	-7.89469238E-06	4.05163293E-04
4	5	-3.45607720E-07	-5.30425150E-05	1.56814891E-06	1.44918000E-05
4	9	-8.61620263E-05	1.75803543E-08	-1.03938572E-05	1.02789918E-04
4	13	4.33068938E-07	9.06960409E-07	1.32282933E-07	1.63396906E-07
4	17	5.68267388E-08	1.62046315E-05	-1.53817201E-05	1.20564434E-05
4	21	8.70371213E-07	-1.50909450E-06	1.02755587E-06	-2.05353464E-05
4	25	1.66511971E-08	-8.65963986E-07	1.32326402E-06	-8.72062029E-06
4	29	-7.16749529E-08	-4.33837206E-08	-6.97736674E-06	7.94473003E-05
4	33	-6.06633040E-05	3.35099469E-08	-6.67214830E-07	-7.87958766E-07
4	37	-9.54997450E-05	9.11613406E-05	7.52470318E-05	-1.51295750E-04
4	41	-1.01314596E-06	-2.16815703E-06	-2.73362939E-07	-3.60627943E-07
4	45	-1.14215576E-07	-1.81152441E-05	2.38784693E-05	-2.26492917E-05
4	49	-1.74198528E-06	2.65528163E-06	6.20921358E-08	1.21554186E-07
4	53	-2.26817197E-06	5.11630313E-05	-4.32653442E-06	-7.33685352E-06
4	57	1.42294181E-07	-1.95153930E-06	7.71425467E-07	-1.18421115E-07
4	61	-1.19561416E-06	6.74967040E-07	-2.55157711E-07	-1.08855286E-06
4	65	3.76019298E-05	-7.06738308E-07	-1.45321163E-05	-5.33550108E-05
4	69	-2.78048204E-06	-2.25906648E-05	-1.60498447E-04	3.92336142E-04
4	73	8.96926103E-06	-1.94132105E-06	6.44527863E-05	1.12072829E-04
4	77	-1.58839741E-05	-3.24326203E-04	-9.92925184E-04	4.06000168E-04
4	81	6.69216094E-04	8.60146264E-05	-7.27628659E-05	-6.11310125E-06
4	85	-2.88796664E-04	-3.28890383E-06	-2.13124281E-05	-1.40758312E-03
4	89	-6.55122870E-06	-1.42473673E-04	-2.89849238E-04	-3.22982410E-04
4	93	1.31602059E-04	-2.50006673E-05	-1.15230946E-05	3.10234628E-05
5	1	1.02183012E-04	6.14775207E-05	1.64926485E-04	1.02137621E-06
5	5	-2.15430359E-05	-1.53093243E-05	-7.48192471E-06	1.88631953E-06
5	9	-1.65605342E-05	5.70711648E-09	6.59998080E-06	-1.42711977E-05
5	13	-1.53522060E-08	-5.77712911E-09	-2.06530253E-09	-8.40538883E-10
5	17	-3.83000922E-10	-1.29362846E-07	1.36751055E-06	1.68919011E-06
5	21	2.26327656E-07	4.48352372E-07	-1.20642163E-05	-1.73584909E-06
5	25	5.02943364E-09	3.32046519E-06	-4.98332774E-06	-3.93624537E-07
5	29	8.36185062E-07	1.97771122E-08	-4.10665447E-05	6.63712519E-06
5	33	-2.87508835E-06	1.28069355E-08	4.86933640E-06	-1.24403550E-05
5	37	-8.84079359E-06	2.71912021E-06	5.96268958E-06	-2.66516232E-06
5	41	-1.99116097E-10	-5.03258206E-08	-3.97403024E-09	-8.75488731E-09
5	45	-2.53344827E-09	-3.99690263E-07	-1.80274203E-07	-1.19289719E-06
5	49	-1.54388979E-07	-1.82861751E-07	1.11692905E-07	1.50935382E-06
5	53	-8.57796957E-07	1.22314384E-07	-2.17772680E-05	9.79375070E-06
5	57	9.37232854E-08	4.32177631E-07	-1.64465741E-07	5.64589302E-07
5	61	-5.76687833E-06	1.26649631E-05	4.03392305E-06	-5.82954152E-06
5	65	-7.58710747E-06	-1.87042565E-06	-3.10103897E-05	-1.02125718E-05
5	69	-1.65755093E-05	-5.03973732E-04	-6.78329340E-05	-3.01052375E-05
5	73	9.21420522E-05	-8.60706584E-04	3.40795306E-04	1.66454611E-04
5	77	6.72030479E-04	9.71086634E-05	-1.15828719E-05	-1.23063476E-04
5	81	1.86838462E-05	1.52246500E-04	4.47292369E-05	1.47849693E-04
5	85	-1.88613366E-06	1.58469387E-05	-2.53133594E-05	-4.33034942E-06
5	89	-4.52437275E-04	4.07025564E-05	3.25524272E-05	6.93614015E-05
5	93	2.37830910E-04	1.18757856E-04	7.32310828E-05	2.63028255E-05
6	1	4.76396242E-05	-1.92321805E-04	3.42858309E-05	1.64299459E-05
6	5	-1.62114099E-05	-6.83422694E-05	-1.80612386E-05	-1.12956142E-05
6	9	1.16881227E-05	-1.05065077E-08	-1.88619584E-05	-1.22076663E-05
6	13	-6.19088436E-08	-1.29501851E-07	-1.89421645E-08	-2.34219179E-08
6	17	-8.15008043E-09	-2.28900753E-06	2.30175076E-06	-1.86806692E-06
6	21	-1.62428558E-07	2.53865465E-07	-3.45668679E-07	-6.30310881E-06
6	25	-2.75438674E-09	-5.84902704E-06	-8.75246359E-07	-2.35431010E-06
6	29	-1.63774916E-07	-1.51471046E-08	9.75208860E-06	-1.70103600E-06

```
     6   33  -4.30797683E-05   -6.27490920E-C9   -7.37695132E-06    6.38298151E-C7
     6   37  -5.77514648E-06    8.31166651E-05    2.73673951E-05   -2.04650660E-05
     6   41  -9.88046685E-08   -3.19709432E-C7   -3.6566722CE-08   -5.44178057E-08
     6   45  -1.67650292E-08   -2.47715272E-06    7.24731882E-07   -2.57722296E-07
     6   49  -4.0190676CE-08    3.79463867E-C8    5.98114400E-08   -3.163C3218E-C6
     6   53   9.85115469E-07    1.95231868E-05    3.99452721E-06   -9.90464587E-06
     6   57   1.46514564E-C7    2.07342643E-C6    1.6967617CE-C6   -3.589286 8E-C7
     6   61   1.49506284E-05    2.34629505E-06    7.94984771E-07   -1.85423516E-06
     6   65   2.01834373E-C5   -3.49299599E-C7   -1.33288665E-05    2.04597369E-C5
     6   69   2.06284689E-05   -7.26241060E-05    8.43336884E-04    2.95370420E-04
     6   73  -2.42927781E-05   -5.33902708E-C5   -2.91409169E-04   -6.992100 76E-C4
     6   77   3.57077277E-04   -1.99783042E-04   -1.70726467E-04    4.24958510E-05
     6   81  -8.80366283E-05    1.20824504E-C5   -1.38882395E-04   -2.198278C8E-C5
     6   85   1.73717650E-04   -2.90861643E-06    1.45813036E-04    6.27014304E-05
     6   89  -4.363242C2E-05   -4.63522378E-C4    6.05549064E-05   -9.79977440E-C5
     6   93  -9.59324927E-06   -1.51363965E-05   -4.75292573E-06    2.56011385E-05
 OCCCOCCCOO
 JT        1    96
     1   31    1    C    2
     1   37    3    4    5    6
     1   54    7    8
     1   65    9
     1   70   10   11   12   13   14
 0000000000
 IHT       1    6
     1    1    1    2    3    0    0    0
 000C0C0000
 ISIGT     1    6
     1    1    0    0    0    1    2    3
 0000000000
 FREQSC    1    1
     1    1      0.0
 000000C000
 MODESC    6    1
     1    1      0.0
     2    1      0.0
     3    1      0.0
     4    1      0.0
     5    1      0.0
     6    1      0.0
 000C00CC00
 JC        1    1
     1    1    1
 0000000000
 IHC       1    6
     1    1    1    2    3    0    0    0
 000C000000
 ISIGC     1    6
     1    1    0    0    0    1    2    3
 0000000000
 INCOND    6    3
     2    1    180.                3.                180.
     3    1    0.                C.1                C.
     4    1    6.                0.                 -2.
     5    1    13.8               0.                  9.15
     6    1   11.12               36.38               0.0
 C00C0CC000
     1    2   2   T, SEC    UT        TARGET LIN VEL   (I COMP/T TRIAD), IPS
     1    3   2   T, SEC    VT        TARGET LIN VEL   (J COMP/T TRIAD), IPS
     1    4   2   T, SEC    WT        TARGET LIN VEL   (K COMP/T TRIAD), IPS
     1    5   2   T, SEC    OMXT      TARGET ROT VEL   (I COMP/T TRIAD), DPS
     1    6   2   T, SEC    OMYT      TARGET ROT VEL   (J COMP/T TRIAD), DPS
```

```
1    7  2  T, SEC    OMZT      TARGET ROT VEL   (K COMP/T TRIAD), DPS
1    8  2  T, SEC    UC        CHASE LIN VEL    (I COMP/C TRIAD), IPS
1    9  2  T, SEC    VC        CHASE LIN VEL    (J COMP/C TRIAD), IPS
1   10  2  T, SEC    WC        CHASE LIN VEL    (K COMP/C TRIAD), IPS
1   11  2  T, SEC    OMXC      CHASE ROT VEL    (I COMP/C TRIAD), DPS
1   12  2  T, SEC    OMYC      CHASE ROT VEL    (J COMP/C TRIAD), DPS
1   13  2  T, SEC    OMZC      CHASE ROT VEL    (K COMP/C TRIAD), DPS
1   14  2  T, SEC    ROX       PROBE EXTENSION  (P TRIAD), IN.
1   15  2  T, SEC    ALY       PROBE ROTATION ABOUT Y  (P TRIAD), DEG
1   16  2  T, SEC    ALZ       PROBE ROTATION ABOUT Z  (P TRIAD), DEG
1   17  2  T, SEC    GAM1      PITCH ARM 1 ROTATION  (P TRIAD), DEG
1   18  2  T, SEC    GAM2      PITCH ARM 2 ROTATION  (P TRIAD), DEG
1   19  2  T, SEC    GAM3      PITCH ARM 3 ROTATION  (P TRIAD), DEG
1   20  2  T, SEC    DSDX      VECTOR DSD  (I COMP/ D TRIAD), IN.
1   21  2  T, SEC    DSDY      VECTOR DSD  (J COMP/ D TRIAD), IN.
1   22  2  T, SEC    DSDZ      VECTOR DSD  (K COMP/ D TRIAD), IN.
1   23  2  T, SEC    LAM 1     CONSTRAINT FORCE, TENS LINK 1, LBS
1   24  2  T, SEC    LAM 2     CONSTRAINT FORCE, TENS LINK 2, LBS
1   25  2  T, SEC    LAM 3     CONSTRAINT FORCE, TENS LINK 3, LBS
1   26  2  T, SEC    LAM 4     CONSTRAINT FORCE, PROBE HD CONTACT, LBS
1   27  2  T, SEC    LAM 5     CONSTRAINT FORCE, PITCH ARM 1, LBS
1   28  2  T, SEC    LAM 6     CONSTRAINT FORCE, PITCH ARM 2, LBS
1   29  2  T, SEC    LAM 7     CONSTRAINT FORCE, PITCH ARM 3, LBS
1   30  2  T, SEC    LAM 8     CONSTRAINT FORCE, PROBE HD CAPTURE, LBS
1   31  2  T, SEC    LAM 9     CONSTRAINT FORCE, PROBE HD CAPTURE, LBS
1   32  2  T, SEC    LAM 10    CONSTRAINT FORCE, PROBE HD CAPTURE, LBS
1   33  2  T, SEC    LAMLFX    LATCH CONSTRAINT FORCE  (FX) LB.
1   34  2  T, SEC    LAMLFY    LATCH CONSTRAINT FORCE  (FY) LB.
1   35  2  T, SEC    LAMLFZ    LATCH CONSTRAINT FORCE  (FZ) LB.
1   36  2  T, SEC    LAMLMX    LATCH CONSTRAINT MOMENT (MX) IN-LB.
1   37  2  T, SEC    LAMLMY    LATCH CONSTRAINT MOMENT (MY) IN-LB.
1   38  2  T, SEC    LAMLMZ    LATCH CONSTRAINT MOMENT (MZ) IN-LB.
1   39  2  T, SEC    FXD        FORCE AT D, X COMP, LBS
1   40  2  T, SEC    FYD        FORCE AT D, Y COMP, LBS
1   41  2  T, SEC    FZD        FORCE AT D, Z COMP, LBS
1   42  2  T, SEC    MXD       MOMENT AT D, X COMP, IN. LBS
1   43  2  T, SEC    MYD       MOMENT AT D, Y COMP, IN. LBS
1   44  2  T, SEC    MZD       MOMENT AT D, Z COMP, IN. LBS
1   45  2  T, SEC    PHI       EULER ANGLE ROLL, DEG     CMG STATION
1   46  2  T, SEC    THETA     EULER ANGLE PITCH, DEG    CMG STATION
1   47  2  T, SEC    IOMEGZ    INTEGRAL OF OMEGA Z, DEG CMG STATION
1   48  2  T, SEC    PHICSM    EULER ANGLE ROLL  CSM ATTIT. ERROR  DEG.
1   49  2  T, SEC    THECSM    EULER ANGLE PITCH CSM ATTIT. ERROR  DEG.
1   50  2  T, SEC    PSICSM    EULER ANGLE YAW   CSM ATTIT. ERROR  DEG.
1   51  2  T, SEC    TORQX     CMG TORQUE, X COMP, IN. LBS
1   52  2  T, SEC    TORQY     CMG TORQUE, Y COMP, IN. LBS
1   53  2  T, SEC    TORQZ     CMG TORQUE, Z COMP, IN. LBS
1   54  2  T, SEC    TXTACS    TACS CONTROL TORQUE X IN-LB.
1   55  2  T, SEC    FYTACS    TACS CONTROL FORCE  Y  LB.
1   56  2  T, SEC    FZTACS    TACS CONTROL FORCE  Z  LB.
1   57  2  T, SEC    TXRCS     RCS CONTROL TORQUE X  IN-LB.
1   58  2  T, SEC    TYRCS     RCS CONTROL TORQUE Y  IN-LB.
1   59  2  T, SEC    TZRCS     RCS CONTROL TORQUE Z  IN-LB.
000C000000
   400.      0.0        0.0        0.0        0.0        0.0        0.0          2.2
    1.25        1.9        0.5        0.5        0.5       0.61       0.51       C.61
     0.24       0.24       0.24     1440.0
CMGSLP   6    14
   1    1  -7.26375859E-05   1.14622589E-04   4.27000728E-06   2.73991503E-04
   1    5  -5.58963318E-04   2.72508185E-C4   1.31209158E-05   3.11969468E-C5
   1    9  -2.09105069E-04  -6.34475302E-05   2.57131685E-04   4.79394787E-05
   1   13  -4.86035021E-04  -5.91527509E-C4
```

```
    2    1  -2.91474528E-05  -1.52660247E-05   7.94852302E-04   2.66242646E-05
    2    5   5.20392374E-05   2.77615483E-07   2.35689672E-05   2.17862838E-04
    2    9   3.44388233E-05   1.09246492E-05  -5.56375182E-04  -2.35773047E-04
    2   13   1.41511359E-05   1.27635645E-05
    3    1   1.21136805E-05  -1.15872227E-05   1.94109146E-05  -3.99939026E-05
    3    5   7.79460617E-05  -1.54271345E-05  -7.81754333E-04  -2.26373116E-04
    3    9   2.34721117E-04   7.93266973E-05  -9.32890174E-06  -9.90988984E-05
    3   13   2.84309935E-05   2.11170183E-04
    4    1  -7.17712332E-06  -6.13312115E-05  -9.90614064E-05   9.73504723E-05
    4    5   7.47866263E-05  -1.44918745E-04   5.24854656E-05  -4.54864112E-06
    4    9   4.08274598E-05  -2.74983309E-05   8.83878113E-05  -4.09595955E-04
    4   13   4.06192304E-05  -5.57115100E-06
    5    1  -5.07791649E-03   1.43951491E-02  -7.92058377E-03  -2.61401214E-02
    5    5  -7.15160868E-03   2.25925616E-03  -4.69875981E-03  -3.31425635E-03
    5    9  -6.28561112E-03  -4.30982556E-03   3.16609280E-02   1.07251359E-02
    5   13  -9.10841690E-04  -1.35164385E-04
    6    1  -2.14941189E-02  -1.30899041E-03  -5.04784146E-03   1.52679139E-03
    6    5   5.75175350E-03  -9.86934598E-04   8.60929157E-04  -1.74451642E-02
    6    9   2.33170332E-04   4.06231986E-02   3.43828044E-03   1.59705874E-03
    6   13   5.12837618E-03   5.56540263E-03
0000000000
CVEC      6   12
    1    1              0.0              2.0             20.0            100.0
    1    5          28320.0              2.0             20.0              0.0
    1    9              0.0              0.0              0.5           -484.0
    2    1              0.0              2.0             20.0            100.0
    2    5            100.0              2.0             20.0              0.0
    2    9              0.0              0.0              0.5            0.831
    3    1              0.0              2.0             20.0            100.0
    3    5           -200.0              2.0             20.0              0.0
    3    9              0.0              0.0              0.5            36.17
    4    1              0.0              4.0             20.0            100.0
    4    5          15360.0              0.5              1.0              0.0
    4    9            0.007              1.0              1.0
    5    1             -3.0              4.0             20.0            100.0
    5    5          15360.0              0.5              1.0              0.0
    5    9            0.007              1.0              1.0
    6    1              0.0              4.0             20.0            100.0
    6    5          15360.0              0.5              1.0              0.0
    6    9            0.007              1.0              1.0
0000000000
SC-1
    0         0.0         2.5         30.
  100.       300.          0.
   12.         0.          0.
STOP
```

APPENDIX E

PROGRAM DOCKEL OUTPUT

Appendix E presents sample outp . from DOCKEL. This output was obtained using the input data of Appendix D. Although the program is capable of printing output at each integration interval, only a small percentage of that information is presented here.

RUN NO. SC-1 DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX TDTO (1 X 3)

1	1	4.07880000E+02	8.30900000E-01	3.61670000E+01

END OF READ.

CARD INPUT MATRIX CPCO (1 X 3)

1	1	1.34700000E+02	-7.70000000E+00	-7.70000000E+00

END OF READ.

CARD INPUT MATRIX SIGMA (1 X 3)

1	1	1.80000000E+02	3.00000000E+02	6.00000000E+01

END OF READ.

CARD INPUT MATRIX AIT (3 X 3) INERTIA FOR CLUSTER ONLY

1	1	6.42800000E+06	1.62200000E+04	3.52800000E+06
2	1	1.62200000E+04	2.47400000E+07	-1.57500000E+05
3	1	3.52800000E+06	-1.57500000E+05	2.51200000E+07

END OF READ.

THE TARGET VEHICLES MASS IS 2.96400000E+02

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX AIC (3 X 3)

1	1	1.95240000E+05	1.69560000E+04	1.69320000E+04
2	1	1.69560000E+04	5.39016000E+05	-7.92120000E+04
3	1	1.69320000E+04	-7.92120000E+04	5.39028000E+05

END OF READ.

THE CHASE VEHICLES MASS IS 7.88577000E+01

CARD INPUT MATRIX FSPROX (2 X 8)

1	1	7.00000000E+00	0.	3.20000000E-01	8.70000000E-01
1	5	2.62000000E+00	8.12000000E+02	1.11000000E+01	1.11200000E+01
2	1	0.	-2.00000000E+03	-1.25000000E+02	-1.18000000E+02
2	5	-1.06200000E+02	-8.10000000E+01	-6.90000000E+01	7.31000000E+02

END OF READ.

CARD INPUT MATRIX FDMROX (2 X 5)

1	1	4.00000000E+00	-1.20000000E+01	-1.00000000E-01	1.00000000E-01
1	5	1.20000000E+01			
2	1	0.	-9.20000000E+01	-9.10000000E+01	9.10000000E+01
2	5	9.20000000E+01			

END OF READ.

RUN NO. SC-1 DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX FSPALL (2 X 19)

1	1	1.80000000E+01	-7.08000000E-02	-6.60500000E-02	-5.87000000E-02
1	5	-5.13500000E-02	-3.67000000E-02	-2.93600000E-02	-2.20200000E-02
1	9	-1.46800000E-02	-7.34000000E-03	7.34000000E-03	1.46800000E-02
1	13	2.20200000E-02	2.93600000E-02	3.67000000E-02	5.13500000E-02
1	17	5.87000000E-02	6.60500000E-02	7.08000000E-02	
2	1	0.	-8.17500000E+04	-7.23000000E+04	-5.91500000E+04
2	5	-4.91000000E+04	-3.32200000E+04	-2.61700000E+04	-1.95000000E+04
2	9	-1.29400000E+04	-6.25600000E+03	6.25600000E+03	1.29400000E+04
2	13	1.95000000E+04	2.61700000E+04	3.32200000E+04	4.91000000E+04
2	17	5.91500000E+04	7.23000000E+04	8.17500000E+04	

END OF READ.

CARD INPUT MATRIX FDMALY (2 X 3)

1	1	2.00000000E+00	0.	1.00000000E+00
2	1	0.	0.	1.00000000E-06

END OF READ.

CARD INPUT MATRIX FSPALU (2 X 19)

1	1	1.80000000E+01	-7.08000000E-02	-6.60500000E-02	-5.87000000E-02
1	5	-5.13500000E-02	-3.67000000E-02	-2.93600000E-02	-2.20200000E-02
1	9	-1.46800000E-02	-7.34000000E-03	7.34000000E-03	1.46800000E-02
1	13	2.20200000E-02	2.93600000E-02	3.67000000E-02	5.13500000E-02
1	17	5.87000000E-02	6.60500000E-02	7.08000000E-02	
2	1	0.	-8.17500000E+04	-6.51500000E+04	-4.70000000E+04
2	5	-3.60000000E+04	-2.22000000E+04	-1.71800000E+04	-1.22800000E+04
2	9	-8.31000000E+03	-4.08500000E+03	4.08500000E+03	8.31000000E+03
2	13	1.22800000E+04	1.71800000E+04	2.22000000E+04	3.60000000E+04
2	17	4.70000000E+04	6.51500000E+04	8.17500000E+04	

END OF READ.

CARD INPUT MATRIX FDMALZ (2 X 3)

1	1	2.00000000E+00	0.	1.00000000E+00
2	1	0.	0.	1.00000000E-06

END OF READ.

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX FSPATT (2 X 10)

1	1	9.00000000E+00	8.43000000E+00	8.50000000E+00	9.00000000E+00
1	5	9.50000000E+00	1.00000000E+01	1.05000000E+01	1.10000000E+01
1	9	1.12500000E+01	1.15000000E+01		
2	1	0.	-3.60000000E+02	-3.05000000E+02	-1.85000000E+02
2	5	-1.23500000E+02	-8.65000000E+01	-6.25000000E+01	-4.55000000E+01
2	9	-3.85000000E+01	-3.25000000E+01		

END OF READ.

CARD INPUT MATRIX FDMATT (2 X 16) ATTEN. DAMPING (T=TEMP= +70 THRU +250 DEG F)

1	1	1.50000000E+01	-1.92000000E+01	-1.82500000E+01	-1.68000000E+01
1	5	-1.44000000E+01	-1.20000000E+01	-9.60000000E+00	-7.20000000E+00
1	9	-4.80000000E+00	-2.76000000E+00	-1.20000000E+00	-6.00000000E-01
1	13	-2.40000000E-01	-1.20000000E-01	1.20000000E-01	1.32000000E+00
2	1	0.	-2.98000000E+03	-2.68500000E+03	-2.45000000E+03
2	5	-2.17000000E+03	-1.92000000E+03	-1.68000000E+03	-1.43500000E+03
2	9	-1.17500000E+03	-9.10000000E+02	-6.30000000E+02	-4.05000000E+02
2	13	-2.10000000E+02	-1.80000000E+01	1.80000000E+01	7.80000000E+01

END OF READ.

CARD INPUT MATRIX FREQ (1 X 96)

1	5	0.	0.	2.54996194E-01	2.56654233E-01
1	9	2.59230144E-01	2.63558032E-01	2.665302243E-01	3.16363216E-01
1	13	4.08638136E-01	4.08671735E-01	4.08675934E-01	4.08678470E-01
1	17	4.08678774E-01	4.08875139E-01	5.10929768E-01	5.13945264E-01
1	21	5.37976914E-01	5.38156352E-01	5.74922575E-01	5.81649117E-01
1	25	5.86156855E-01	5.86943106E-01	6.17050186E-01	6.18100481E-01
1	29	6.18829615E-01	6.18840578E-01	7.15880147E-01	8.57194549E-01
1	33	8.76752892E-01	9.04248565E-01	9.05857860E-01	9.20249224E-01
1	37	9.88036352E-01	1.02627491E+00	1.06018342E+00	1.09757768E+00
1	41	1.11870753E+00	1.11875386E+00	1.11876104E+00	1.11876459E+00
1	45	1.11876506E+00	1.11889359E+00	1.25392463E+00	1.25909002E+00
1	49	1.30060590E+00	1.30091512E+00	1.36216324E+00	1.36281889E+00
1	53	1.36672613E+00	1.37554721E+00	1.37958881E+00	1.73465346E+00
1	57	1.81008596E+00	1.81094561E+00	1.81107947E+00	1.82252375E+00
1	61	2.11352028E+00	2.13162065E+00	2.13181872E+00	2.13471946E+00
1	65	2.14928675E+00	2.78612394E+00	2.88460197E+00	2.90610104E+00
1	69	3.31360816E+00	3.85170624E+00	4.12990059E+00	4.14176135E+00
1	73	4.47363853E+00	4.68029637E+00	6.12480919E+00	6.33019624E+00
1	77	6.65320423E+00	7.50817221E+00	7.95936849E+00	8.30724494E+00
1	81	8.94704806E+00	9.38847045E+00	9.79588804E+00	1.00894674E+01
1	85	1.03479378E+01	1.11819072E+01	1.17654872E+01	1.22070518E+01
1	89	1.23384923E+01	1.25791755E+01	1.47431405E+01	1.75212818E+01
1	93	2.52185445E+01	4.39829177E+01	5.28482614E+01	8.88177531E+01

END OF READ.

RUN NO. SC-1 DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX RTMODE (6 X 96)

1	1	3.37096783E-02	1.23539660E-02	-1.59449945E-02	-4.34111732E-04
1	5	3.57812083E-02	-2.43781212E-02	2.35864798E-03	5.93297350E-04
1	9	-4.80510784E-04	2.86124847E-06	5.95135453E-04	-7.72296766E-05
1	13	7.54610732E-07	1.13132248E-06	1.95837124E-07	2.09693285E-07
1	17	7.93955160E-08	1.92548456E-05	6.17025205E-05	1.24977151E-04
1	21	2.01805904E-05	2.91987366E-05	5.86912477E-03	7.94923731E-05
1	25	2.46479748E-06	2.10084411E-04	2.30859386E-03	1.09899480E-04
1	29	3.06015576E-05	1.52113891E-06	2.02338550E-02	4.81375751E-03
1	33	3.97618063E-03	6.17778992E-06	7.71207804E-04	6.67019316E-03
1	37	4.81624098E-03	2.75076466E-03	5.85791268E-04	4.82502094E-04
1	41	5.95146487E-06	2.23030764E-06	4.75653357E-07	3.03218252E-07
1	45	1.25060419E-07	2.71882806E-05	8.50025176E-05	1.17011315E-04
1	49	-9.62123889E-07	-4.12844677E-06	1.07137173E-05	1.38935231E-04
1	53	2.60864398E-03	1.74962268E-04	6.04076638E-04	4.28118908E-03
1	57	3.15612193E-05	7.03016400E-06	4.08273876E-04	1.42170985E-05
1	61	1.68447020E-03	1.17703399E-03	3.47239825E-04	2.45452095E-03
1	65	7.24354186E-04	3.06177881E-05	-5.79596544E-04	2.26075309E-04
1	69	6.06747606E-06	-4.21192149E-04	5.05314330E-05	2.23933093E-04
1	73	-3.78903618E-04	1.02142122E-04	-2.56064876E-04	5.71295238E-05
1	77	-2.59390862E-04	-2.65493979E-04	8.67187204E-04	2.84514565E-03
1	81	4.75644775E-05	9.00861691E-04	7.73672058E-04	-6.13299226E-04
1	85	-1.98280055E-03	6.17674589E-02	4.15375434E-03	3.37380770E-04
1	89	2.04984887E-03	4.42937788E-04	2.78800581E-03	-2.08751501E-03
1	93	1.28418761E-03	7.42829220E-04	3.30279460E-04	1.27781896E-04
2	1	5.12305849E-02	-9.38024557E-02	3.02482483E-03	1.13541500E-02
2	5	-4.77465311E-03	1.85895868E-02	-7.12965330E-03	-6.77275097E-03
2	9	7.05480541E-03	-2.81927842E-06	-7.57665369E-03	-8.18171770E-03
2	13	-4.02767476E-05	-8.44746263E-05	-1.23409744E-05	-1.52574010E-05
2	17	-5.31044093E-06	-1.50324529E-03	1.57474857E-03	-1.28803286E-03
2	21	-1.06291004E-04	1.65694373E-04	-5.04402141E-05	-4.56511191E-03
2	25	4.74056828E-07	-2.48415143E-03	-3.15687166E-04	-1.63810216E-03
2	29	-7.52210968E-05	-9.35274382E-06	4.45609851E-03	8.98498502E-03
2	33	-2.27800082E-02	9.07101958E-07	-3.08334165E-03	9.25577209E-04
2	37	-1.38663972E-02	4.57833566E-02	1.64751749E-02	-1.54098939E-02
2	41	-8.01084343E-05	-2.16464105E-04	-2.56562958E-05	-3.64312972E-05
2	45	-1.13274068E-05	-1.72692110E-03	6.98141186E-04	-4.93337745E-04
2	49	-3.22884382E-05	4.00262597E-05	3.12055788E-05	-1.54029390E-03
2	53	4.64187617E-04	1.25919799E-02	2.05443035E-03	-6.26650314E-03
2	57	9.27369053E-05	1.19602774E-03	1.33604028E-03	-2.99914585E-04
2	61	9.38544874E-03	-7.50081000E-04	-2.44707813E-04	7.64415830E-04
2	65	1.39814868E-02	1.42695203E-04	-2.11763155E-03	6.86936506E-04
2	69	3.63629420E-03	-1.50813437E-02	1.82028141E-01	6.35892015E-02
2	73	-4.41588856E-03	-1.32801069E-02	-6.45130556E-02	-1.49810078E-01
2	77	7.31403070E-02	-3.95398658E-02	-3.28627534E-02	1.08226846E-02
2	81	-2.12644730E-02	3.16180286E-04	-7.44669028E-03	-2.23929026E-03
2	85	1.18435727E-02	-6.31011855E-04	1.28152957E-02	4.67712916E-03
2	89	-3.40669164E-03	-3.72410981E-02	4.36326396E-04	-7.24800995E-03

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX RTMODE (6 X 96) CONTINUED

2	93	3.26548153E-03	-3.91671271E-04	-3.13290351E-04	1.00457022E-04
3	1	-2.11585247E-02	-1.46326584E-02	-9.31244414E-02	-2.01675988E-03
3	5	-3.60273501E-02	-8.58750492E-03	3.17591518E-03	6.77441643E-04
3	9	2.81078096E-03	-6.63618234E-06	-8.67791524E-03	2.05171220E-03
3	13	9.39534805E-06	2.45356634E-06	1.38114974E-06	4.05995360E-07
3	17	2.79969117E-07	4.83999644E-05	-2.17516492E-04	-1.80126598E-04
3	21	-1.77733337E-05	-5.64411234E-05	5.21399193E-03	9.02113934E-04
3	25	-5.99149926E-06	-3.38981921E-03	2.19304817E-03	1.62777600E-04
3	29	-4.23221267E-04	-1.00276510E-05	1.82785419E-02	-2.80294086E-03
3	33	1.73807862E-03	-1.18396157E-05	-4.20914862E-03	5.91410353E-03
3	37	3.87151615E-03	-2.02442955E-03	5.71566533E-05	2.55949822E-03
3	41	1.30523402E-05	2.86476085E-05	3.38002610E-06	4.70551521E-06
3	45	1.43176900E-06	2.45000441E-04	-2.41877551E-04	1.56480255E-04
3	49	-1.29462024E-05	-5.17450306E-05	-4.44193955E-05	-1.70417615E-03
3	53	5.22249285E-04	5.47823067E-05	1.21926002E-02	-6.17484979E-03
3	57	-6.72031620E-05	-4.52914706E-04	5.80743529E-05	-7.47069084E-04
3	61	4.23715659E-03	-5.84963413E-03	-1.88011752E-03	4.72411822E-03
3	65	3.91863542E-03	-1.28264833E-04	3.28906411E-03	7.26718985E-04
3	69	2.57811648E-03	8.73291314E-02	1.32490167E-02	5.52847365E-03
3	73	-3.47557168E-02	2.08432504E-01	-7.26947275E-02	-3.65157930E-02
3	77	-1.39919243E-01	-2.07244730E-02	-9.99155049E-04	1.42645257E-02
3	81	-3.53340564E-03	-1.97234480E-02	-5.38535789E-03	-5.58437218E-04
3	85	1.23432852E-03	-5.92216145E-04	-8.87614287E-04	8.76749164E-05
3	89	3.34002997E-02	-3.81946911E-03	-2.48992180E-03	-1.09488268E-02
3	93	-2.80716917E-02	-1.03347817E-02	-6.25110713E-03	-2.50590566E-03
4	1	-8.93415952E-05	1.55241722E-04	-7.89459238E-06	4.05163293E-04
4	5	-3.45607720E-07	-5.30425150E-05	1.56814891E-06	1.44918000E-05
4	9	-8.61620263E-05	1.75803543E-08	-1.03938572E-05	1.02789918E-04
4	13	4.33068938E-07	9.06960409E-07	1.32282933E-07	1.63396906E-07
4	17	5.68267388E-08	1.62046315E-05	-1.53817201E-05	1.20564434E-05
4	21	8.70371213E-07	-1.50909450E-06	1.02755587E-06	-2.05353464E-05
4	25	1.66511971E-08	-8.65963986E-07	1.32326402E-06	-8.72062029E-06
4	29	-7.16749529E-08	-4.33837206E-08	-6.97736674E-06	7.94473003E-05
4	33	-6.06633040E-05	3.35099469E-08	-6.67214830E-07	-7.87958766E-07
4	37	-9.54997450E-05	9.11613486E-05	7.52470318E-05	-1.51295750E-04
4	41	-1.01314596E-06	-2.16815703E-06	-2.73362939E-07	-3.60627943E-07
4	45	-1.14215576E-07	-1.81152441E-05	2.38784693E-05	-2.26492917E-05
4	49	-1.74198528E-06	2.65528163E-06	6.20921358E-08	1.21554186E-07
4	53	-2.26817197E-06	5.11630313E-05	-4.32653442E-06	-7.33685352E-06
4	57	1.42294181E-07	-1.95153930E-06	7.71425467E-07	-1.18421115E-07
4	61	-1.19561416E-06	6.74967040E-07	-2.55157711E-07	-1.08855286E-06
4	65	3.76019298E-05	-7.06738308E-07	-1.45321163E-05	-5.33550108E-05
4	69	-2.78048204E-06	-2.25906648E-05	-1.60498447E-04	3.92336142E-04
4	73	8.96926103E-06	-1.94132105E-06	6.44527863E-05	1.12072829E-04
4	77	-1.58839741E-05	-3.24326283E-04	-9.92925184E-04	4.06000168E-04
4	81	6.69216094E-04	8.60146264E-05	-7.27628659E-05	-6.11310125E-06
4	85	-2.88796664E-04	-3.28890383E-06	-2.13124281E-05	-1.40758312E-03

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX RTMODE (6 X 96) CONTINUED

4	89	-6.55122870E-06	-1.42473673E-04	-2.89849238E-04	-3.22982410E-04
4	93	1.31602059E-04	-2.50006673E-05	-1.15230946E-05	3.10234628E-05
5	1	1.02183012E-04	6.14775207E-05	1.64926485E-04	1.02137621E-06
5	5	-2.15430359E-05	-1.53093243E-05	-7.48192471E-06	1.88631953E-06
5	9	-1.65605342E-05	5.70711648E-09	6.59998080E-06	-1.42711977E-05
5	13	-1.53522060E-08	-5.77712911E-09	-2.06530253E-09	-8.40538883E-10
5	17	-3.83800922E-10	-1.29362846E-07	1.36751055E-06	1.68919011E-06
5	21	2.26327656E-07	4.48352372E-07	-1.20642163E-05	-1.73584909E-06
5	25	5.02943364E-09	3.32046519E-06	-4.98332774E-06	-3.93624537E-07
5	29	8.36185062E-07	1.97771122E-08	-4.10665447E-05	6.63712519E-06
5	33	-2.87508835E-06	1.28069355E-08	4.86933648E-06	-1.24403550E-05
5	37	-8.84079359E-06	2.71912021E-06	5.96268958E-06	-2.66516232E-06
5	41	-1.99116097E-10	-5.03258206E-08	-3.97403024E-09	-8.75488731E-09
5	45	-2.53344827E-09	-3.99690263E-07	-1.80274203E-07	-1.19289719E-06
5	49	-1.54388979E-07	-1.82861751E-07	1.11692985E-07	1.50935382E-06
5	53	-8.57796957E-07	1.22314384E-07	-2.17772680E-05	9.79375070E-06
5	57	9.37232854E-08	4.32177631E-07	-1.64465741E-07	5.64589302E-07
5	61	-5.76687833E-06	1.26649631E-05	4.03392305E-06	-5.82954152E-06
5	65	-7.58710747E-06	-1.87042565E-06	-3.10103897E-05	-1.02125718E-05
5	69	-1.65755093E-05	-5.03973732E-04	-6.78329340E-05	-3.01052375E-05
5	73	9.21420522E-05	-8.60706584E-04	3.40795300E-04	1.66454611E-04
5	77	6.72030479E-04	9.71086634E-05	-1.15828719E-05	-1.23063476E-04
5	81	1.86838462E-05	1.52246500E-04	4.47292369E-05	1.47849693E-04
5	85	-1.88613366E-06	1.58469387E-05	-2.53133594E-05	-4.33034942E-06
5	89	-4.52437275E-04	4.07025564E-05	3.25524272E-05	6.93614015E-05
5	93	2.37830910E-04	1.18757856E-04	7.32310828E-05	2.63028255E-05
6	1	4.76396242E-05	-1.92321805E-04	3.42858309E-05	1.64299459E-05
6	5	-1.62114099E-05	-6.83422694E-05	-1.80612386E-05	-1.12956142E-05
6	9	1.16881227E-05	-1.05065077E-08	-1.88619584E-05	-1.22076663E-05
6	13	-6.19088436E-08	-1.29501851E-07	-1.89421645E-08	-2.34219179E-08
6	17	-8.15080043E-09	-2.28900753E-06	2.30175076E-06	-1.86806692E-06
6	21	-1.62428558E-07	2.53865465E-07	-3.45668679E-07	-6.30310881E-06
6	25	-2.75438674E-09	-5.84902704E-06	-8.75246359E-07	-2.35431010E-06
6	29	-1.63774916E-07	-1.51471046E-08	9.75208860E-06	-1.70103600E-06
6	33	-4.30797683E-05	-6.27490920E-09	-7.37695132E-06	6.38298151E-07
6	37	-5.77514648E-06	8.31166651E-05	2.73673951E-05	-2.04650660E-05
6	41	-9.88046685E-08	-3.19709432E-07	-3.65667220E-08	-5.44178057E-08
6	45	-1.67650292E-08	-2.47715272E-06	7.24731882E-07	-2.57722296E-07
6	49	-4.01906760E-08	3.79463867E-08	5.98114400E-08	-3.16303218E-06
6	53	9.85115469E-07	1.95231868E-05	3.99452721E-06	-9.90464587E-06
6	57	1.46514564E-07	2.07342643E-06	1.69676170E-06	-3.58928608E-07
6	61	1.49506284E-05	2.34629505E-06	7.94984771E-07	-1.85423516E-06
6	65	2.01834373E-05	-3.49299599E-07	-1.33288665E-05	2.84597369E-05
6	69	2.06284689E-05	-7.26241860E-05	8.43336884E-04	2.95370420E-04
6	73	-2.42927781E-05	-5.33902708E-05	-2.91409169E-04	-6.99210076E-04
6	77	3.57077277E-04	-1.99783042E-04	-1.70726467E-04	4.24958510E-05
6	81	-8.80366283E-05	1.20824504E-05	-1.38882395E-04	-2.19827808E-05
6	85	1.73717650E-04	-2.90861643E-06	1.45813036E-04	6.27014304E-05
6	89	-4.36324202E-05	-4.63522378E-04	6.05549064E-05	-9.79977440E-05
6	93	-9.59324927E-06	-1.51363965E-05	-4.75292573E-06	2.56011385E-05

END OF READ.

RUN NO. SC-1

DATE 1FFE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT INTEGER MATRIX JT (1 X 96)

1	31	1	0	2	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
1	37	3	4	5	6	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
1	54	7	8	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
1	65	9	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
1	70	10	11	12	13	14	-0	-0	-0	-0	-0	-0	-0	-0	-0

END OF READIM.

CARD INPUT INTEGER MATRIX IHT (1 X 6)

1	1	1	2	3	0	0	0

END OF READIM.

CARD INPUT INTEGER MATRIX ISIGT (1 X 6)

1	1	0	0	0	1	2	3

END OF READIM.

CARD INPUT MATRIX FREQSC (1 X 1)

1	1	0.

END OF READ.

CARD INPUT MATRIX MODESC (6 X 1)

1	1	0.
2	1	0.
3	1	0.
4	1	0.
5	1	0.
6	1	0.

END OF READ.

```
RUN NO. SC-1                                              DATE 16FE70
                                                         RUN BY

        NAS8-21280
        BASELINE CONFIGURATION

CARD INPUT INTEGER MATRIX JC       (   1 X   1 )

   1    1    1

END OF READIM.


CARD INPUT INTEGER MATRIX IHC      (   1 X   6 )

   1    1    1    2    3    0    0    0

END OF READIM.


CARD INPUT INTEGER MATRIX ISIGC    (   1 X   6 )

   1    1    0    0    0    1    2    3

END OF READIM.


OUTPUT MATRIX    LOC  (   1 X  12 )

   1    1         1    7   10   16   22   25   31   37   51   65   66   67

END OF WRITIM.


OUTPUT MATRIX    LEN  (   1 X  12 )

   1    1         6    3    6    6    3    6    6   14   14    1    1   [illegible]4

END OF WRITIM.
```

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX INCOND (6 X 3)

2	1	1.80000000E+02	3.00000000E+00	1.80000000E+02
3	1	0.	1.00000000E-01	0.
4	1	6.00000000E+00	0.	-2.00000000E+00
5	1	1.38000000E+01	0.	9.15000000E+00
6	1	1.11200000E+01	3.63800000E+01	0.

END OF READ.

THE DATMIN ARRAY IS AS FOLLOWS --

1 TARGET EULER ANGLES - PSI, THETA, PHI (DEGREES)
2 CHASE / TARGET EULER ANGLES - PSI, THETA, PHI (DEGREES)
3 CHASE ANGULAR VELOCITY - OMEGA X, OMEGA Y, OMEGA Z - (DEGREES/SEC)
4 CHASE C.G. VELOCITY IN X,Y,Z DIRECTIONS REFERENCED TO CHASE (LENGTH/SEC)
5 D TO S DISTANCE IN X,Y,Z DIRECTIONS REFERENCED TO TARGET (LENGTH)
6 RHO(X), (LENGTH), GAMMA(1,2,3), (DEGREES), --------

RUN NO. SC-1 DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

OUTPUT MATRIX YINITL (1 X 100)

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	1	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.000E+00
1	11	0.	-0.	0.	1.000E+00	0.	6.000E+00	0.	-2.000E+00	0.	1.745E-03
1	21	0.	5.038E+02	0.531E+00	4.610E+01	-9.986E-01	-3.426E-15	-5.234E-02	-3.251E-15	1.000E+00	-3.426E-15
1	31	1.112E+01	0.	0.	6.350E-01	6.350E-01	6.350E-01	0.	0.	0.	0.

END OF WRITE.

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

THE INPUT SCALARS TO PROGRAM DOCKEL ARE

STARTT	=	0.
DELTAT	=	.010000
ENDT	=	5.000000
XDLTA	=	.500000
XPRNT	=	50.000000
ZETAT	=	.010000
ZETAC	=	.010000
IFELAS	=	1
IFPLOT	=	1
IFPER	=	1
JBATCH	=	1
MINIC	=	1
IFPNCH	=	0
NCOLS	=	58
NCNSYS	=	34
NFORCS	=	10
NEQTN	=	100
NMODET	=	14
NMODEC	=	1

AA	=	2.25000000E+00
AB	=	8.50000000E-01
AD	=	7.08000000E-01
AE	=	9.65000000E-01
AF	=	3.08500000E+00
AP	=	4.25000000E+00
DLB	=	1.44800000E+01
DLD	=	1.29000000E+01
DLE	=	1.75900000E+01
DLF	=	4.75000000E+00
DLI	=	1.41000000E+01
DLP	=	3.30000000E+00
DLQ	=	1.65300000E+01
DLS	=	2.05500000E+00
FD0	=	1.15000000E+01
PE0	=	1.05000000E+01
ALPHAD	=	4.27500000E+01
CSPHER	=	2.58000000E+00
RADCON	=	1.35600000E+01
AKD1	=	0.
AKD2	=	0.
AKD3	=	5.41000000E+00
RHOSO	=	2.08000000E+00
FRETL	=	1.00000000E+03
EXTL	=	1.11200000E+01
TLAG	=	1.01500000E+01
TSTAR	=	0.
BINDMU	=	2.80000000E-01
HDMU	=	2.60000000E-01
ARMMU	=	2.60000000E-01

RUN NO. SC-1

DATE 16FE70
RUN BY

NAS8-21280
BASELINE CONFIGURATION

CARD INPUT MATRIX CMGSLP (6 X 14)

1	1	-7.26375859E-05	1.14622589E-04	4.27000728E-06	2.73991503E-04
1	5	-5.58963318E-04	2.72508185E-04	1.31209158E-05	3.11969468E-05
1	9	-2.09105069E-04	-6.34475302E-05	2.57131685E-04	4.79394787E-05
1	13	-4.86035021E-04	-5.91527509E-04		
2	1	-2.91474528E-05	-1.52660247E-05	7.94852302E-04	2.66242646E-05
2	5	5.20392374E-05	2.77615483E-07	2.35689672E-05	2.17862838E-04
2	9	3.44388233E-05	1.09246492E-05	-5.56375182E-04	-2.35773047E-04
2	13	1.41511359E-05	1.27635645E-05		
3	1	1.21136805E-05	-1.15872227E-05	1.94109146E-05	-3.99939026E-05
3	5	7.79460617E-05	-1.54271345E-05	-7.81754333E-04	-2.26373116E-04
3	9	2.34721117E-04	7.93266979E-05	-9.32898174E-06	-9.90988984E-05
3	13	2.84309935E-05	2.11170183E-04		
4	1	-7.17712332E-06	-6.13312115E-05	-9.90614064E-05	9.73504723E-05
4	5	7.47866263E-05	-1.44918745E-04	5.24854656E-05	-4.54864112E-06
4	9	4.08274598E-05	-2.74983309E-05	8.83878113E-05	-4.09595955E-04
4	13	4.06192304E-05	-5.57115100E-06		
5	1	-5.07791649E-03	1.43951491E-02	-7.92058377E-03	-2.61401214E-02
5	5	-7.15160868E-03	2.25925616E-03	-4.69875981E-03	-3.31425635E-03
5	9	-6.28561112E-03	-4.30982556E-03	3.16609280E-02	1.07251359E-02
5	13	-9.10841690E-04	-1.35164385E-04		
6	1	-2.14941189E-02	-1.30899041E-03	-5.04784146E-03	1.52679139E-03
6	5	5.75175350E-03	-9.86934598E-04	8.60929157E-04	-1.74451642E-02
6	9	2.33170332E-04	4.06231986E-02	3.43828044E-03	1.59705874E-03
6	13	5.12837618E-03	5.56540263E-03		

END OF READ.

CARD INPUT MATRIX CVEC (6 X 12)

1	1	0.	2.00000000E+00	2.00000000E+01	1.00000000E+02
1	5	2.83200000E+04	2.00000000E+00	2.00000000E+01	0.
1	9	0.	0.	5.00000000E-01	-4.84000000E+02
2	1	0.	2.00000000E+00	2.00000000E+01	1.00000000E+02
2	5	1.00000000E+02	2.00000000E+00	2.00000000E+01	0.
2	9	0.	0.	5.00000000E-01	8.31000000E-01
3	1	0.	2.00000000E+00	2.00000000E+01	1.00000000E+02
3	5	-2.00000000E+02	2.00000000E+00	2.00000000E+01	0.
3	9	0.	0.	5.00000000E-01	3.61700000E+01
4	1	0.	4.00000000E+00	2.00000000E+01	1.00000000E+02
4	5	1.53600000E+04	5.00000000E-01	1.00000000E+00	0.
4	9	7.00000000E-03	1.00000000E+00	1.00000000E+00	-0.
5	1	-3.00000000E+00	4.00000000E+00	2.00000000E+01	1.00000000E+02
5	5	1.53600000E+04	5.00000000E-01	1.00000000E+00	0.
5	9	7.00000000E-03	1.00000000E+00	1.00000000E+00	-0.
6	1	0.	4.00000000E+00	2.00000000E+01	1.00000000E+02
6	5	1.53600000E+04	5.00000000E-01	1.00000000E+00	0.
6	9	7.00000000E-03	1.00000000E+00	1.00000000E+00	-0.

END OF READ.

SIMULATION TIME = 0.

	Y(I)	YDOT(I)
U - TARGET		
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
X - TARGET		
	0.	0.
	0.	0.
	0.	0.
GAMMA TARGET		
	1.00000000E+00	0.
	0.	0.
	0.	0.
	0.	0.
	1.00000000E+00	0.
	0.	0.
U - CHASE		
	6.00000000E+00	3.49065850E-03
	0.	0.
	-2.00000000E+00	1.04719755E-02
	0.	7.07047821E-05
	1.00000000E-01	-1.78223662E-06
	0.	3.00732500E-06
X - CHASE		
	5.83836006E+02	-6.09644912E+00
	8.53090000E+00	-1.40518405E-14
	4.61046691E+01	1.68324333E+00
GAMMA CHASE		
	-9.98629535E-01	-9.13434754E-05
	-3.42554796E-15	5.67358305E-18
	-5.23359562E-02	1.74293734E-03
	-3.25072364E-15	0.
	1.00000000E+00	0.
	-3.42554796E-15	0.
RHOS		
	1.11200000E+01	0.
	0.	0.
	0.	0.
	3.63623627E+01	0.
	3.63623627E+01	0.
	3.63623627E+01	0.
MODE VEL - TARGET		
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
MODE DISP - TARGET		
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.
	0.	0.

```
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
MODE VEL - CHASE
                                   0.                         0.
MODE DISP - CHASE
                                   0.                         0.
CON SYS PARAMETERS
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                  -3.00000000E+00             1.00000000E-01
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.
                                   0.                         0.

DSD VECTOR =        1.38000000E+01       -1.84741111E-13       9.15000000E+00

                PHI                        LAMDA                   MODE

   1        4.99981392E-03           2.20643694E+01              1
   2        4.99981392E-03           2.20643694E+01              1
   3        4.99981392E-03           2.20643694E+01              1
   4        6.83993270E-02           0.                          0
   5        1.41487526E+00           0.                          0
   6        3.96042096E+00           0.                          0
   7        3.96042096E+00           0.                          0
   8        9.99917919E+00           0.                          0
   9       -1.84741111E-13           0.                          0
  10        9.15000000E+00           0.                          0


        RETRACT FORCE =    0.

        BINDING FRICTION FORCE =    0.


         FORCE AT PT O =     0.             0.             0.
        MOMENT AT PT O =     0.             0.             0.
         FORCE AT PT P =     0.             0.             0.
        MOMENT AT PT P =     0.             0.             0.


                         ELAPSED CP TIME =     9.98900000E+00
```

SIMULATION TIME = 1.0000E+00

	Y(I)	YDOT(I)
U - TARGET		
	-6.53045006E-01	1.04489443E-03
	-2.15296265E-02	-2.93016913E-05
	4.21112870E-01	1.61808245E-03
	8.07355482E-03	9.53540921E-06
	-1.41768653E-01	-6.05199704E-07
	-7.77701543E-03	1.61410641E-05
X - TARGET		
	-5.21302477E-01	-6.53877036E-01
	-1.53549278E-02	-2.15089943E-02
	3.34713360E-01	4.19820841E-01
GAMMA TARGET		
	9.99998047E-01	-4.89715037E-06
	-9.81799367E-05	-1.35990022E-04
	1.97379776E-03	2.47431090E-03
	9.79759316E-05	1.35456131E-04
	9.99999990E-01	-2.78767074E-08
	1.03453033E-04	1.41177751E-04
U - CHASE		
	8.58759832E+00	5.05363808E+00
	1.15379688E-01	5.57880229E-02
	-4.86358729E-01	-3.18536303E-01
	-4.40033010E-02	3.09762860E-02
	-2.12584033E+00	-1.43167297E-03
	-3.69721167E-01	7.28028290E-04
X - CHASE		
	5.77267510E+02	-8.59916630E+00
	8.58538920E+00	6.95529936E-02
	4.63857310E+01	-2.14950036E-01
GAMMA CHASE		
	-9.96660732E-01	3.05757580E-03
	-5.28755205E-03	-6.42089657E-03
	-8.14826775E-02	-3.69822766E-02
	-5.33994329E-03	-6.49388014E-03
	9.99985652E-01	-3.47793193E-05
	4.25067028E-04	2.39652990E-04
RHOS		
	1.10982264E+01	5.44338967E-03
	-3.74396055E-02	9.35990137E-03
	-3.78674898E-02	9.46687245E-03
	3.64116305E+01	-1.22738947E-02
	3.64116305E+01	-1.22738947E-02
	3.64116305E+01	-1.22738947E-02
MODE VEL - TARGET		
	1.06025829E+00	-2.62472549E+00
	2.30658132E-02	-1.69005590E+00
	-1.11416716E-01	-1.29515334E+00
	-2.94527082E-01	-3.72909725E+00
	-6.81618420E-02	-7.20357223E-01
	1.37123926E-01	8.19891655E-01
	-3.31043875E-02	1.09194079E-01
	3.82133007E-01	-5.55055817E+00
	8.94793303E-02	1.06823524E+00
	-1.12393872E+00	-2.62309029E+01
	4.14524264E-02	4.46126669E+00
	9.79290972E-03	1.66462476E+00
	-1.97302413E-01	-2.43849375E+00
	5.08066735E-01	-1.51734524E+01
MODE DISP - TARGET		
	1.25016579E-01	1.06025829E+00
	5.56075022E-02	2.30658132E-02
	3.39648506E-02	-1.11416716E-01
	9.05978327E-02	-2.94527082E-01
	1.64411383E-02	-6.81618420E-02
	-1.76372356E-02	1.37123926E-01
	-2.45617058E-03	-3.31043875E-02
	7.29899271E-02	3.82133007E-01
	-5.99010196E-03	8.94793303E-02

4.57153438E-02	-1.12393872E+00
-6.65745831E-03	4.14524264E-02
-2.46555221E-03	9.79290972E-03
3.22669945E-03	-1.97302413E-01
1.72004539E-02	5.08066735E-01

MODE VEL - CHASE

0.	0.

MODE DISP - CHASE

0.	0.

CON SYS PARAMETERS

3.92000000E+02	4.00000000E+02
5.65873300E-03	-3.91342984E-03
-1.10428639E-01	-1.46326856E-01
-6.12930683E-03	-8.56164437E-03
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
-2.44358025E-02	-1.38510451E-02
-4.67379518E+00	-2.12599782E+00
-3.03966600E-01	-3.70044983E-01
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.

DSD VECTOR = 7.98777814E+00 -7.57267305E-01 3.50936341F+00

	PHI	LAMDA	MODE
1	4.96503952E-03	2.20013015E+01	1
2	4.96503952E-03	2.20013015E+01	1
3	4.96503952E-03	2.20013015E+01	1
4	2.04232503E-01	0.	0
5	1.74266263E+00	0.	0
6	7.17706798E+00	0.	0
7	6.02814051E+00	0.	0
8	4.18695733E+00	0.	0
9	-7.67267305E-01	0.	0
10	3.50936341E+00	0.	0

RETRACT FORCE = 0.

BINDING FRICTION FORCE = 0.

FORCE AT PT D =	0.	0.	0.
MOMENT AT PT D =	0.	0.	0.
FORCE AT PT P =	0.	0.	0.
MOMENT AT PT P =	0.	0.	0.

ELAPSED CP TIME = 1.15869000E+02

SIMULATION TIME = 2.0000E+00

	Y(I)	YDOT(I)
U - TARGET		
	-2.92500222E+00	-4.33237595E+00
	-6.21314193E-02	8.32196263E-02
	4.88168815E-03	1.76284667E+00
	1.62787922E-02	-1.11940089E-02
	-8.39953601E-02	-6.28986819E-01
	-1.84191350E-02	2.25111596E-02
X - TARGET		
	-1.83395271E+00	-2.92504233E+00
	-1.19475142E-01	-6.01361284E-02
	5.25535528E-01	-5.75612441E-03
GAMMA TARGET		
	9.99993202E-01	-5.53155289E-06
	-6.83185793E-04	-3.22392058E-04
	3.62349628E-03	1.46578365E-03
	6.80913035E-04	3.20442742E-04
	9.99999571E-01	-3.97471657E-07
	6.28424448E-04	2.85281463E-04
U - CHASE		
	4.99497071E+00	-1.05079835E+01
	2.10683254E-01	-3.60982604E-01
	-1.40371155E+00	8.11409280E+00
	-8.57522487E-02	-3.99648551E-01
	-4.11412849E-01	-9.12065775E+00
	-4.99507286E-01	-2.91502002E+00
X - CHASE		
	5.68619289E+02	-5.11133096E+00
	8.94968806E+00	1.85372586E-01
	4.67962378E+01	8.96916267E-01
GAMMA CHASE		
	-9.94968176E-01	7.60701381E-04
	-4.73674784E-03	-8.70951159E-03
	-1.00079427E-01	-7.15050959E-03
	-4.83047918E-03	-8.82395718E-03
	9.99988092E-01	-4.30525708E-05
	6.94264842E-04	6.16647538E-04
RHOS		
	8.28638651E+00	-2.05405689E+00
	-7.58542247E-01	1.03977408E+00
	-1.56254770E-01	1.11011733E+00
	4.10404460E+01	8.02809176E-01
	4.20344247E+01	3.40905873E+00
	4.20344247E+01	3.40905873E+00
MODE VEL - TARGET		
	-4.43161843E-01	4.09308494E+01
	-9.12136389E-01	9.48114313E+00
	2.34052872E-01	1.10555415E+01
	7.67400712E-01	-2.40921923E+01
	2.92187539E-01	-7.94078383E+00
	5.19550647E-01	1.01759708E+01
	-5.98328746E-01	-5.24617638E+00
	1.66697353E+00	2.02652587E+00
	7.73582901E-01	1.15723822E+01
	-1.72509866E+00	-1.60914665E+02
	-2.95066004E+00	6.81719773E+01
	-9.69809800E-01	2.54266010E+01
	4.64002381E-01	9.89043244E+00
	1.31710399E-01	-7.32339196E+01
MODE DISP - TARGET		
	-2.80359424E+00	-4.43161843E-01
	-4.65378222E-01	-9.12136389E-01
	-4.02792647E-01	2.34052872E-01
	4.93559802E-01	7.67400712E-01
	1.69878518E-01	2.92187539E-01
	-2.08999272E-01	5.19550647E-01
	7.33686795E-02	-5.98328746E-01
	4.77347900E-02	1.66697353E+00
	-5.60180494E-02	7.73582901E-01

```
                                        3.65051683E-01              -1.72509866E+00
                                       -5.13327322E-02              -2.95066004E+00
                                       -2.99780800E-02              -9.69809800E-01
                                       -3.67700381E-02               4.84002381E-01
                                        2.22264970E-01               1.31710399E-01
MODE VEL - CHASE
                                        0.                           0.
MODE DISP - CHASE
                                        0.                           0.
CON SYS PARAMETERS
                                        7.92000000E+02               4.00000000E+02
                                        3.52701194E-02              -4.15467504E-02
                                       -2.15662279E-01               5.83503235E-02
                                       -4.15467460E-02              -1.56628252E-03
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                       -3.99791660E-02              -3.55384869E-02
                                       -5.74374428E+00              -4.11761289E-01
                                       -2.72766121E-01              -5.01739099E-01
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.
                                        0.                           0.

DSD VECTOR =          3.80411719E+00        6.41747786E-03        3.74669515E-03

                  PHI                        LAMDA                        MODE

     1        3.07473578E-01          0.                                   0
     2        4.99953621E-03          3.76651241E+02                       1
     3        4.99953621E-03          3.76651241E+02                       1
     4       -3.21926415E-03          0.                                   0
     5        3.88747767E-03          6.46690023E+02                       1
     6        2.36642718E+00          0.                                   0
     7        2.63340055E+00          0.                                   0
     8        3.29637900E-03          6.95229810E+02                       1
     9        6.41747786E-03         -5.31265046E+01                       1
    10        3.74669515E-03         -2.85737772E+02                       1


         RETRACT FORCE =     0.

         BINDING FRICTION FORCE =  -4.74062187E+01


          FORCE AT PT D =     -1.17208861E+03      5.66752393E+01       6.42545632E+02
         MOMENT AT PT D =     -9.28042973E-01     -1.28730567E+04       1.03585982E+02
          FORCE AT PT P =     -1.10430733E+03     -6.31890289E+01       7.52582981E+02
         MOMENT AT PT P =     -8.61847180E+01     -7.48037862E+03      -1.51873024E+03


                         ELAPSED CP TIME =     2.18033000E+02
```

SIMULATION TIME = 3.0000E+00

	Y(I)	YDOT(I)
U - TARGET		
	-3.62879327E+00	2.70251822E-02
	-1.75869214E-01	5.16717120E-02
	-3.49752468F-02	-3.70089240E-01
	4.68304968E-02	-1.63764519E-02
	-9.29079562E-02	1.08454253E-01
	-5.32215661E-02	1.82182053E-02
X - TARGET		
	-5.32837713E+00	-3.62879586E+00
	-2.44423909E-01	-1.70883862E-01
	4.85512855E-01	-5.41588080E-02
GAMMA TARGET		
	9.99985412E-01	-9.73769205E-06
	-1.36196067E-03	-9.30873558E-04
	5.22699523F-03	1.62038350E-03
	1.35554715E-03	9.24604383E-04
	9.99998324E-01	-2.26492818E-06
	1.23034784E-03	8.22189088E-04
U - CHASE		
	2.73072373E+00	-4.53267738E-02
	6.30877772E-01	-1.98772924E-01
	-1.33118432E+00	-1.45516564E+00
	-6.70385066E-02	5.30127509E-02
	-1.48337410E-01	1.93569898E+00
	-1.07332872E-01	5.87835153E-03
X - CHASE		
	5.65268230E+02	-2.86147718E+00
	9.37671927E+00	5.99604441E-01
	4.77835806E+01	1.03890974E+00
GAMMA CHASE		
	-9.94453609E-01	2.90496108E-04
	-1.02263966E-02	-1.86676505E-03
	-1.04677795E-01	-2.57738160E-03
	-1.04307166E-02	-1.98537578E-03
	9.99944612E-01	-2.20691216E-05
	1.40463039E-03	9.67519348E-04
RHOS		
	8.37660599E+00	1.03794276E+09
	1.88430323E-01	7.02326801E-01
	-2.67195890E-02	-1.68203285E+00
	4.17529626E+01	-1.75578215E+00
	4.17492037E+01	-1.75572786E+00
	4.17492037E+01	-1.75572786E+00
MODE VEL - TARGET		
	-8.48283400E+00	-2.38460302E+01
	-1.73306068E+00	4.47581672E+00
	2.54073656E-01	9.89725666E+00
	-1.79007689E+00	-1.86999728E+01
	-8.60054031E-01	-6.61594444E+00
	1.69674172E+00	2.12469009E+00
	-2.67295307E-01	1.26195109E+01
	-2.95725081E+00	-8.58976865E+00
	1.48549871E+00	-5.03248224E+00
	2.78579583E+00	-7.44506274E+01
	1.43460451E+00	1.30094067E+01
	6.30906045E-01	4.20387732E+00
	-3.73477736E-01	3.28533826E+00
	5.09202102E-01	-5.32548574E+01
MODE DISP - TARGET		
	1.12492445E+00	-8.48283400E+00
	-1.58933329E-01	-1.73306068E+00
	-2.73954888E-01	2.54073656E-01
	4.79335407E-01	-1.79007689F+00
	1.57722253E-01	-8.60054031E-01
	-6.09390816E-02	1.69674172E+00
	-1.65634717E-01	-2.67205307E-01
	1.03154881E-01	-2.95725081E+09
	2.42287191E-02	1.48549871E+00

1.07390255E-01	2.78579583E+00
-1.82560721E-02	1.43460451E+00
-6.08773430E-03	6.30906845E-01
1.02606917E-03	-3.73477736E-01
3.35090330E-02	5.09202102E-01

MODE VEL - CHASE

0.	0.

MODE DISP - CHASE

0.	0.

CON SYS PARAMETERS

8.23333333E+02	0.
6.41124241E-02	7.92134363E-02
-3.10641358E-01	-1.57736440E-01
-7.03388826E-02	2.52527251E-02
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
-8.09240010E-02	-5.57631357E-02
-6.00860329E+00	-1.48488858E-01
-5.89176516E-01	-1.07715021E-01
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.

DSD VECTOR = 3.80622547E+00 5.78355412E-03 1.62547141E-03

	PHI	LAMDA	MODE
1	3.85436505E-03	2.34957342E+01	1
2	4.99999726E-03	2.35153451E+01	1
3	4.99999726E-03	2.35153451E+01	1
4	-7.42852841E-04	0.	0
5	6.05731769E-02	0.	0
6	2.28278025E+00	0.	0
7	2.66874148E+00	0.	0
8	5.40466020E-03	-7.9786531?E+00	1
9	5.78355412E-03	-1.6323060?E+01	1
10	1.62547141E-03	1.14000554E+02	1

RETRACT FORCE = 0.

BINDING FRICTION FORCE = 1.95888442E+01

FORCE AT PT D =	7.97865314E+00	1.63230605E+01	-1.14000554E+02
MOMENT AT PT D =	0.	0.	0.
FORCE AT PT P =	-3.2093931?E+00	-1.62009962E+01	-1.14251718E+02
MOMENT AT PT P =	-9.10489462E-11	2.80303706E+03	-3.97473172E+02

ELAPSED CP TIME = 3.20106000E+02

SIMULATION TIME = 4.0000E+00

	Y(I)	YDOT(I)
U - TARGET		
	-3.63513070E+00	-1.59850724E-02
	-1.19115700E-01	1.48101404E-01
	-4.23676731E-02	1.93284810E-02
	3.08857080E-02	-4.01679320E-02
	-8.92711047E-02	-3.89352592E-03
	-3.45333407E-02	4.71212358E-02
X - TARGET		
	-8.95941947E+00	-3.63499696E+00
	-3.75745536E-01	-1.11490042E-01
	4.19867796E-01	-6.72743896E-02
GAMMA TARGET		
	9.99974787E-01	-1.18302356E-05
	-2.07605079E-03	-6.05604303E-04
	6.79079997E-03	1.55691113E-03
	2.06342020E-03	5.99042596E-04
	9.99996129E-01	-2.24981479E-06
	1.86643464E-03	5.43136883E-04
U - CHASE		
	2.69960005E+00	-5.07114744E-02
	4.23458183E-01	-5.45114168E-01
	-1.38343445E+00	5.02415244E-02
	-7.53528338E-02	-2.71163593E-02
	-1.26881752E-01	-1.86776523E-01
	-4.04383186E-01	-7.89263142E-01
X - CHASE		
	5.62415196E+02	-2.83816931E+00
	9.82762236E+00	3.76362724E-01
	4.88656633E+01	1.08820687E+00
GAMMA CHASE		
	-9.94164229E-01	3.47870608E-04
	-1.54914376E-02	-7.04856351E-03
	-1.06759076E-01	-2.21665587E-03
	-1.58057664E-02	-7.15697436E-03
	9.99872878E-01	-1.14299010E-04
	2.09873806E-03	5.54149785E-04
RHOS		
	9.29282526E+00	7.80345663E-01
	-1.15015680E-02	1.12586096E+00
	-1.16949325E-01	1.15759078E+00
	4.01234558E+01	-1.45667304E+00
	4.01197275E+01	-1.45666210E+00
	4.01197275E+01	-1.45666210E+00
MODE VEL - TARGET		
	6.38192317E+00	-3.06627026E+01
	-2.54148473E+00	-6.75041479E+00
	-5.53676645E-01	8.69022484E+00
	1.85215515E-01	-1.34458243E+01
	-2.46999759E-01	-3.50563153E+00
	6.90697636E-01	-3.79254521E+00
	1.44267708E+00	-1.16787748E+01
	1.63113727E+00	2.09245574E+01
	3.46604368E-01	-1.46421920E+01
	3.14909536E+00	-2.95373693E+00
	2.01430591E-01	1.20114253E+01
	1.53180756E-01	1.60560681E+00
	2.43184266E-01	1.02561961E+00
	8.10344627E-01	-9.85685043E-01
MODE DISP - TARGET		
	1.49588748E+00	6.38192317E+00
	1.97834414E-01	-2.54148473E+00
	-2.39922056E-01	-5.53676645E-01
	3.71343186E-01	1.85215515E-01
	9.62664440E-02	-2.46999759E-01
	6.34740962E-02	6.98697636E-01
	1.60509754E-01	1.44267708E+00
	-2.80411338E-01	1.63113727E+00
	8.32598117E-02	3.46404368E-01

```
                                   1.91430918E-03                  3.14999536E+00
                                  -5.93205931E-03                  2.81430591E-01
                                   1.72618161E-03                  1.53188756E-01
                                  -1.89518660E-03                  2.43184266E-01
                                   8.75546279E-04                  8.10344627E-01
MODE VEL - CHASE
                                   0.                              0.
MODE DISP - CHASE
                                   0.                              0.
CON SYS PARAMETERS
                                   8.23333333E+02                  0.
                                   1.04272757E-01                 -3.15406827E-02
                                  -4.04773933E-01                 -1.08489390E-01
                                  -1.20964483E-01                 -8.87898593E-02
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                  -1.20940185E-01                 -3.19619646E-02
                                  -6.12852397E+00                 -1.27735042E-01
                                  -8.92731941E-01                 -4.86437285E-01
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.
                                   0.                              0.

USD VECTOR =          3.80630450E+00        8.48132556E-03        -4.05936459E-03

                 PHI                        LAMDA                      MODE

    1        3.85436506E-03             1.79562685E+01                 1
    2        4.99999727E-03             1.79851363E+01                 1
    3        4.99999727E-03             1.79851363E+01                 1
    4       -3.18230618E-03             0.                             0
    5        3.98380408E-01             0.                             0
    6        2.58092571E+00             0.                             0
    7        3.20572533E+00             0.                             0
    8        5.48368553E-03             4.73968908E+00                 1
    9        8.48132556E-03            -4.45534282E+01                 1
   10       -4.05936459E-03            -4.03117808E+00                 1


          RETRACT FORCE =     0.

          BINDING FRICTION FORCE =    8.84832552E+00


           FORCE AT PT D =    -4.73968908E+00     4.45534282E+01      4.83117808E+00
          MOMENT AT PT D =     0.                 0.                  0.
           FORCE AT PT P =    -3.52171880E+00    -4.46324239E+01      4.38944726E+00
          MOMENT AT PT P =     3.59698777E-11    -1.11719877E+02     -1.13598185E+03


                          ELAPSED CP TIME =     4.21651000E+02
```

SIMULATION TIME = 5.0000E+00

	Y(I)	YDOT(I)
U - TARGET		
	-3.61778613E+00	1.22062597E-02
	-1.48646556E-01	-4.42768697E-02
	-4.05588650E-02	-8.63708772E-05
	3.81064695E-02	1.10883335E-02
	-8.77866404E-02	1.79324042E-03
	-4.29508000E-02	-1.29544181E-02
X - TARGET		
	-1.25901880E+01	-3.61771304E+00
	-5.02332046E-01	-1.38544389E-01
	3.59314570E-01	-7.12397133E-02
GAMMA TARGET		
	9.99961074E-01	-1.49053278E-05
	-2.76451757E-03	-7.53394641E-04
	8.37902250E-03	1.53024710E-03
	2.74381182E-03	7.44026423E-04
	9.99993156E-01	-3.70750892E-06
	2.48162950E-03	6.71339155E-04
U - CHASE		
	2.75428341E+00	3.51061527E-02
	5.43896005E-01	1.71208112E-01
	-1.41126653E+00	-2.87300756E-02
	-7.08268001E-02	7.09483171E-03
	-7.96960278E-02	6.48559926E-02
	-2.54976518E-01	2.24187312E-01
X - CHASE		
	5.59561134E+02	-2.90313383E+00
	1.02607277E+01	4.78051063E-01
	4.99286076E+01	1.10311121E+00
GAMMA CHASE		
	-9.93776021E-01	2.48417986E-04
	-2.12363496E-02	-4.44199743E-03
	-1.09353722E-01	-1.39492305E-03
	-2.16686897E-02	-4.55755570E-03
	9.99761378E-01	-1.00833440E-04
	2.76663758E-03	7.42099013E-04
RHOS		
	1.02862428E+01	7.54185667E-01
	3.58392749E-03	1.03300166E-01
	3.37410710E-02	-2.86908528E-01
	3.81659166E+01	-1.56117499E+00
	3.81659166E+01	-1.56117499E+00
	3.81659166E+01	-1.56117499E+00
MODE VEL - TARGET		
	4.92428797E+00	3.26271309E+01
	-1.01583405E+00	-1.20396352E+01
	-5.81462339E-01	9.44762534E+00
	-4.82587239E-01	-1.72688116E+01
	-5.63955707E-01	-4.03357421E+00
	4.22686815E-01	-3.95456432E+00
	-2.01302575E+00	5.02221007E-01
	4.43095861E-01	-2.19992791E+01
	-4.95937024E-01	-1.13423527E+01
	1.73644194E+00	3.86531306E+01
	5.92913919E-01	-4.29851166E+00
	1.72191023E-01	-4.[illegible]85901E+00
	-1.67267085E-01	-2.[illegible]557083E+00
	1.17750559E-01	1.[illegible]5622357E+01
MODE DISP - TARGET		
	-1.63518153E+00	4.92428797E+00
	4.10038619E-01	-1.01583405E+00
	-2.3[illegible]558864E-01	-5.8146233[illegible]E-01
	4.03574258E-01	-4.82587239E-01
	3.80814874E-02	-5.63955707E-01
	8.58597709E-02	4.22686815E-01
	-4.13328757E-03	-2.01302575E+00
	2.91179860E-01	4.43095861E-[illegible]1
	6.19632979E-02	-4.95937024E[illegible]01

-6.73696955E-02	1.73644194E+00
2.56537833E-03	5.92913919E-01
4.65839625E-03	1.72191023E-01
2.92573057E-03	-1.67267085E-01
-1.34517425E-02	1.17750559E-01

MODE VEL - CHASE

0.	0.

MODE DISP - CHASE

0.	0.

CON SYS PARAMETERS

5.23333333E+02	0.
1.57186380E-01	3.62969420E-02
-4.84128762E-01	-1.42489829E-01
-1.63557733E-01	4.42073901E-02
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
-1.59473238E-01	-4.28004576E-02
-6.27806185E+00	-8.04054027E-02
-1.22418736E+00	-2.56290703E-01
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.
0.	0.

DSD VECTOR = 3.80583337E+00 6.17789527E-03 2.40133614E-03

	PHI	LAMDA	MODE
1	4.99999026E-03	9.87387283E+00	1
2	4.99998952E-03	9.87387282E+00	1
3	4.99998952E-03	9.87387282E+00	1
4	-1.46469479E-03	0.	0
5	8.63436518E-01	0.	0
6	2.99601116E+00	0.	0
7	3.73557109E+00	0.	0
8	5.01255920E-03	-3.60332659E+00	1
9	6.17789527E-03	1.23118286E+01	1
10	2.40133614E-03	1.69786273E[illegible]	1

RETRACT FORCE = 0.

BINDING FRICTION FORCE = 0.

FORCE AT PT D =	3.60332659E+00	-1.23118286E+01	-1.69786273E+00
MOMENT AT PT D =	0.	0.	0.
FORCE AT PT P =	3.11405985E+00	1.23969433E+01	-2.01649594E+00
MOMENT AT PT P =	-1.08952344E-02	5.33135082E+01	3.27742100E+02

ELAPSED CP TIME = 5.23678000E+02

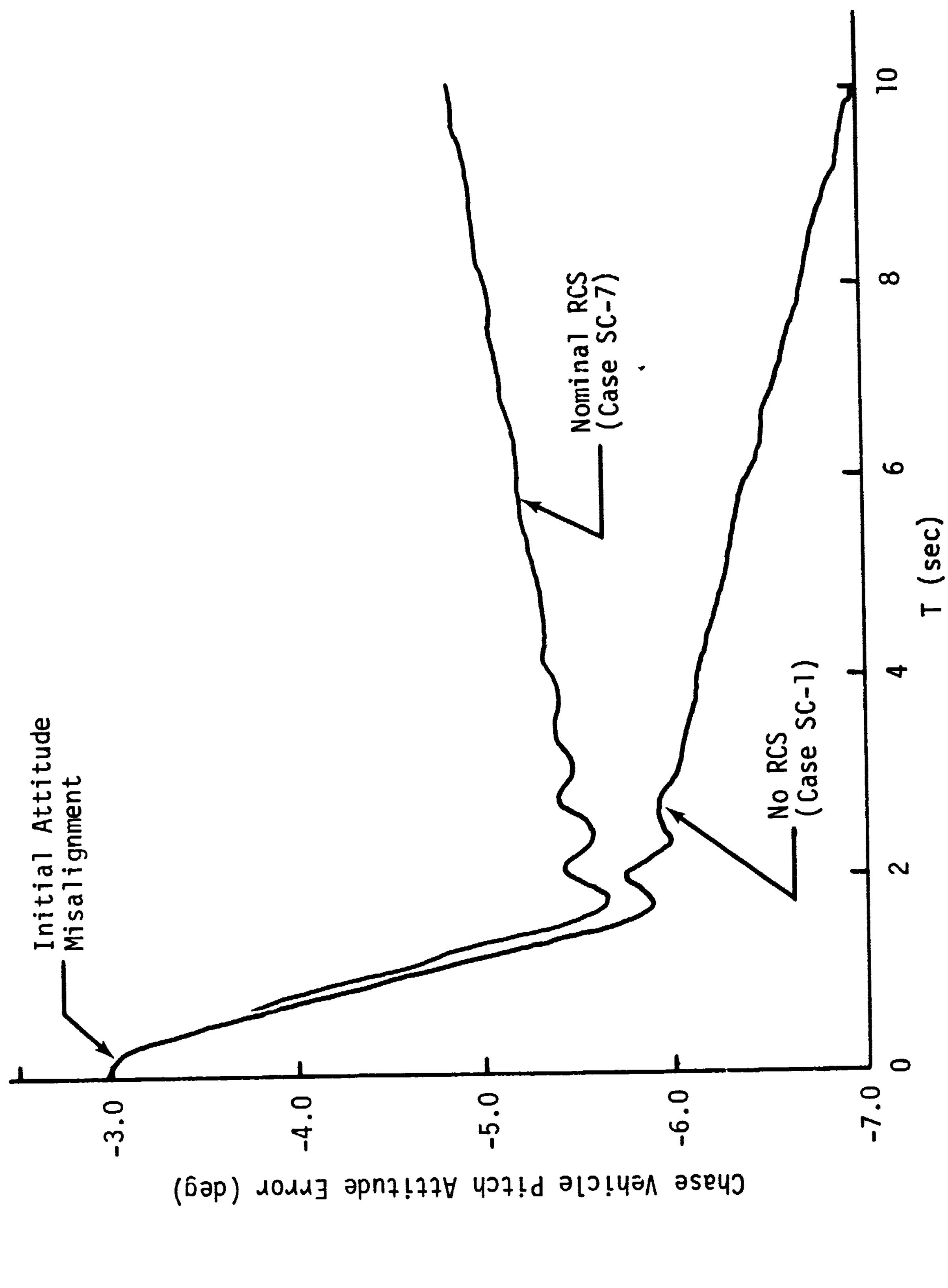


Fig. II-17 RCS Effectivity

III. DIGITAL COMPUTER PROGRAM

The digital computer program for the docking simulation of two elastic vehicles (PROGRAM DOCKEL) was developed to provide numerical evaluation of the analytical formulation presented in Volume I. It was used to generate the results presented in this volume. A summary description of this program is presented in the following sections.

A. APPLICABILITY AND LIMITATIONS

Although the program has been verified through an extensive analysis of a typical Apollo Applications Program (AAP) configuration, it is not restricted to any particular orbital configuration. The program was designed with complete versatility to the user as a primary objective and therefore many options have been included. These options may be exercised at the user's discretion as best benefits his requirements. Also, because of the great complexity of the overall dynamics program, it was necessary to impose several limitations on the program.

The first limitation is strictly a hardware-imposed restraint; the program was developed on, and as such is limited to, the Control Data Corporation 6400/6500 series computers. This is a relatively minor restriction in that modification to any other current digital computer could be made with a minimum of effort. Primarily, the uniqueness lies in the use of the plotting subroutines for the response time histories and the optional perspective/stereo pairs plots.

A far more severe restriction is the assumption of a known and relatively invariant description of the probe and drogue docking mechanism. Although consistent variations in geometry and stiffness or damping characteristics are permitted, no deviation from either the overall geometrical configuration or the basic kinematical relationships of the probe-drogue are permissible. However, the general description of both the target and chase vehicles are quite arbitrary. Input parameters describing the location of the drogue cone on the target vehicle and the probe on the chase vehicle are arbitrary as are the inertial characteristics of both vehicles.

Several of the user selected input options are itemized:

1) The location of the drogue cone apex (D) with respect to the target vehicle center of mass (T) is defined with an input vector;
2) The location of the probe hard-point (P) with respect to the chase vehicle center of mass (C) is defined with an input vector;
3) The orientation of the probe pitch arms with respect to the drogue at the initiation of the simulation are defined with an input vector;
4) The target vehicle total mass and inertial properties about the center of mass are defined with an input scalar and input matrix, respectively;
5) The chase vehicle total mass and inertial properties about the center of mass are defined with an input scalar and input matrix, respectively;
6) Stiffness and damping characteristics of the probe barrel and the three pitch arms are defined by input matrices;
7) Modal characteristics for the target and chase vehicles are defined by an input vector of structural frequencies and input matrix of normal modes. Additional input data required here are three integer vectors that select the desired frequencies and modes from the complete arrays;
8) Control system parameters are defined at the user's option. As many as three control systems can be included. Chase vehicle axial thrust can be varied through the input data.

B. OVERLAY STRUCTURE AND LOGIC FLOW

The requirement for a large amount of core storage locations, especially to store the time history data for plot output, dictated the necessity to use an overlay structure in the formulation of the program. The program comprises four overlays (Fig. III-1).

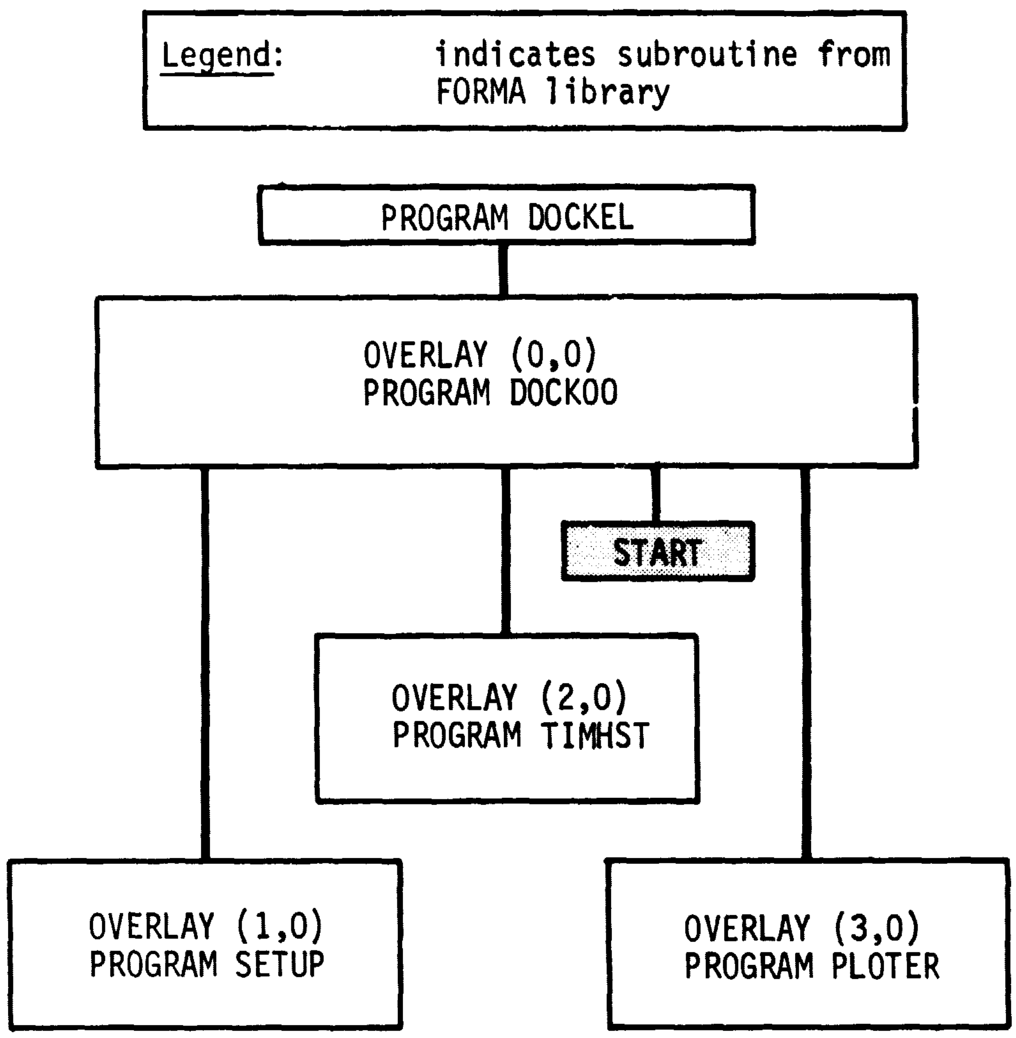


Fig. III-1 PROGRAM DOCKEL Overlay Structure

1. OVERLAY (0,0) PROGRAM DOCKOO

This overlay (Fig. III-2) is the control overlay for the entire program. Its basic function is to allocate sufficient storage locations for program variables through definition of several COMMON blocks. These COMMON blocks are then available to the other three overlays which are called, in turn, by this overlay.

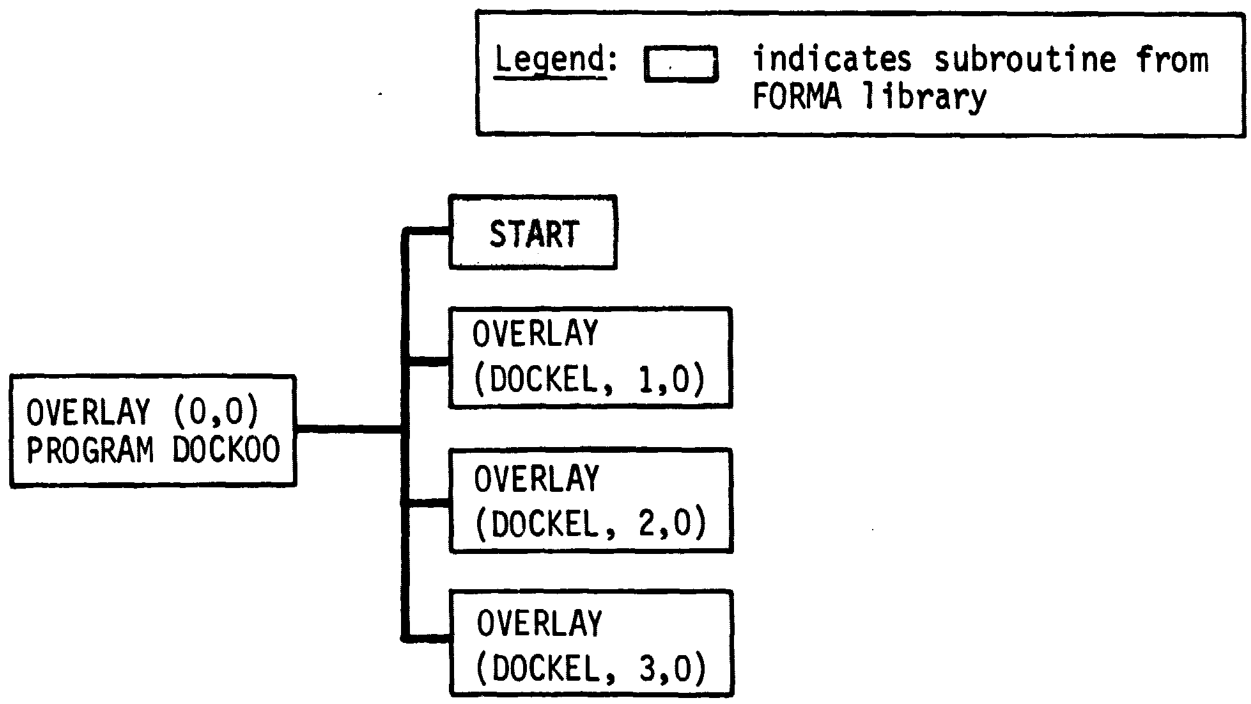


Fig. III-2 PROGRAM DOCKOO Calling Structure

DOCKOO also initializes several program variables with DATA statements, initializes systems plotting subroutines, calls a FORMA subroutine to read a program run number and two title cards, and increments a batch indicator that allows the operator to run several jobs under the same run without the necessity of rereading an entire new set of input data. The overlay (Fig. III-3) then calls the overlay whose function is to read input data and compute program variables (SETUP), the overlay that computes the docking simulation time history (TIMHST), and, if so indicated by the plot option indicator, the overlay that plots docking program variable time histories (PLOTER). Control then returns to the beginning of the overlay and, if data are available, begins another simulation run. If no more data are available, the run is terminated.

2. OVERLAY (1,0) PROGRAM SETUP

The main function of this overlay (Fig. III-4) is to read a majority of required program input data, compute several variables that will be required in the time history evaluation, read and write the title cards on tape for the plot overlay and set integration constants for use in the time history integration subroutine. The subroutine also prints input data. The logic flow for the program is shown in Fig. III-5.

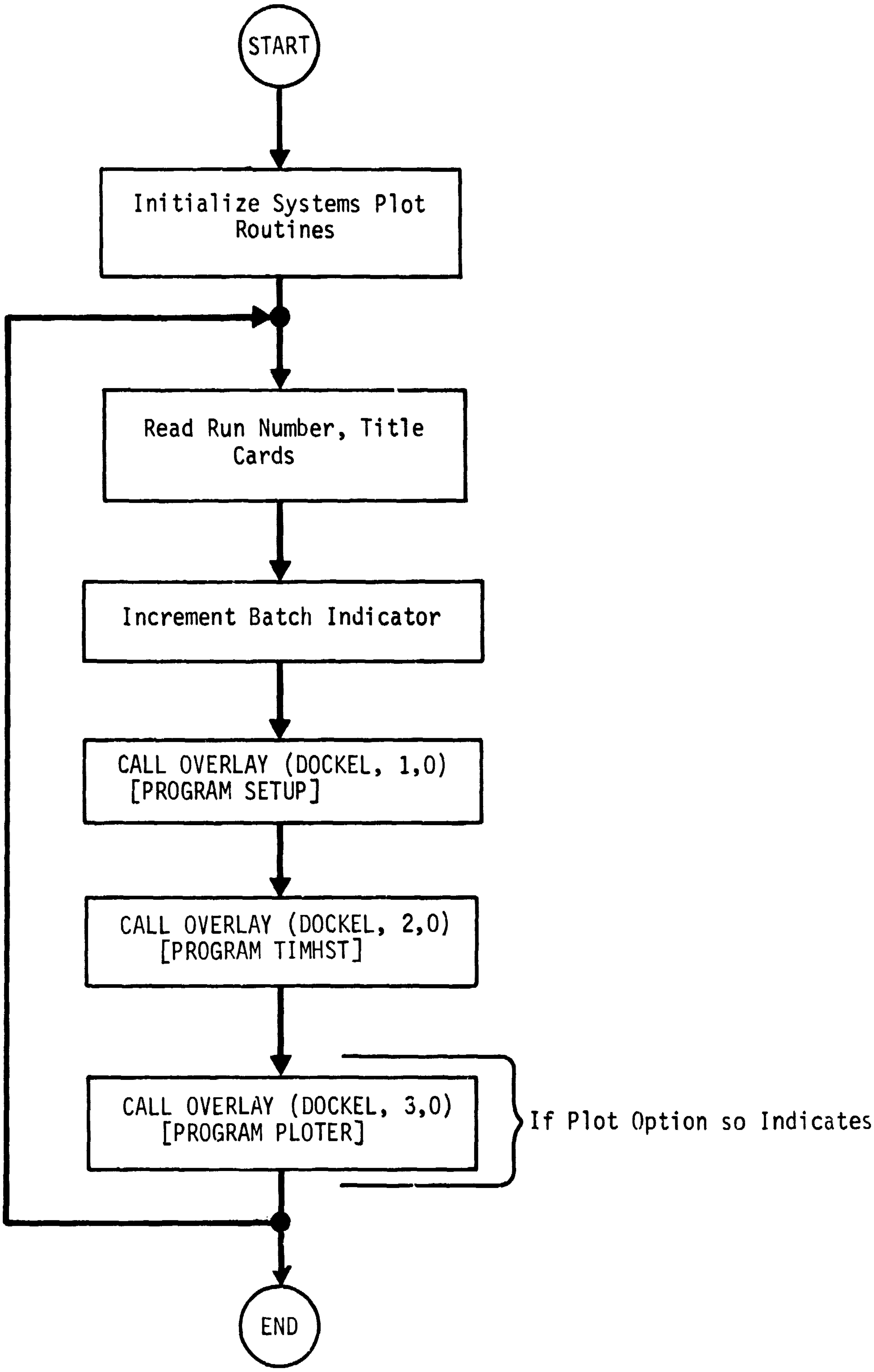


Fig. III-3 PROGRAM DOCK00 Logic Flow

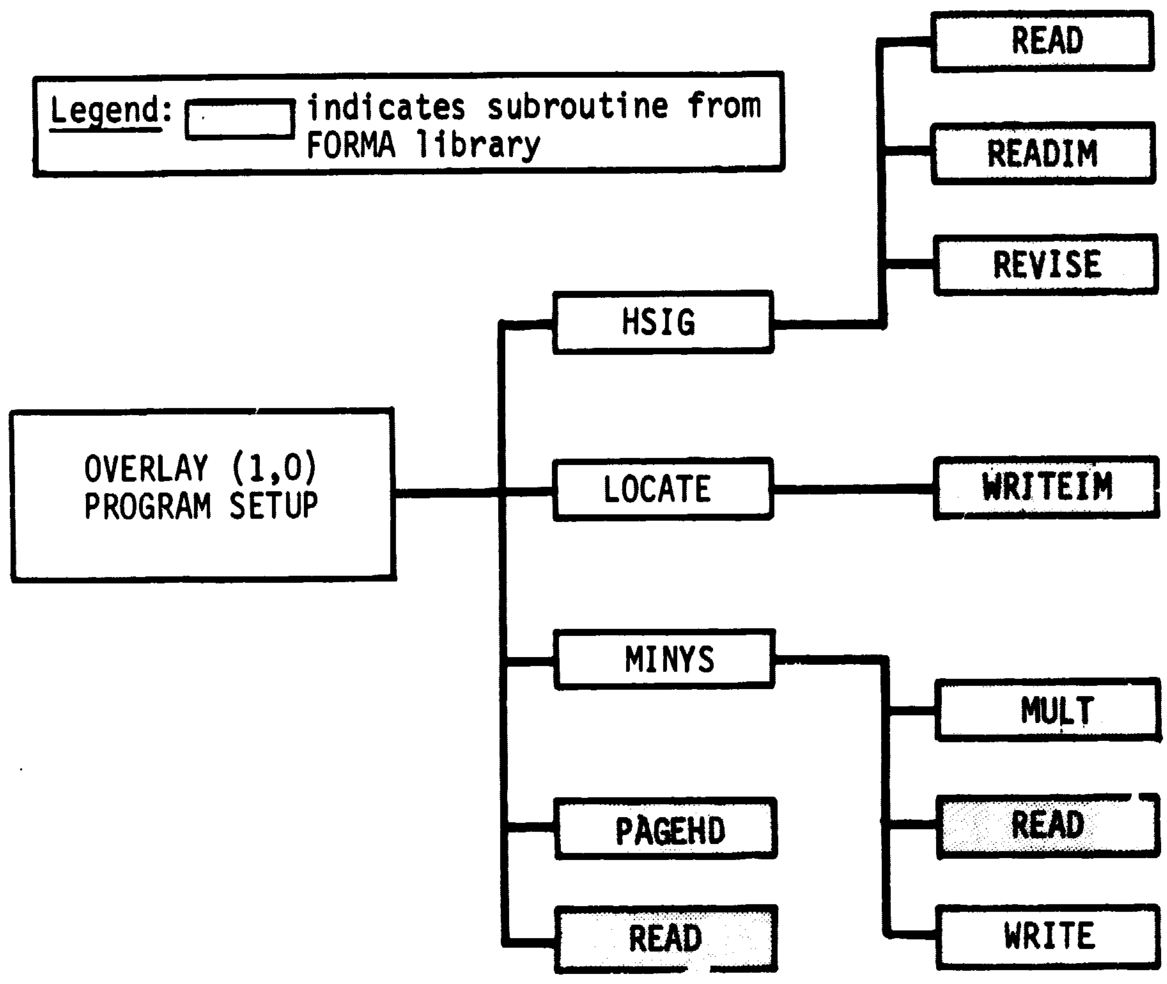


Fig. III-4. PROGRAM SETUP Calling Structure

3. OVERLAY (2,0) PROGRAM TIMHST

PROGRAM TIMHST, as the name indicates, is the overlay that computes the docking simulation time history. The program calling structure is indicated in Fig. III-6. The overlay begins by computing and printing the state vector and all other pertinent variables for the initial time. It then proceeds through the time history integration, using the Runge-Kutta numerical technique, until time reaches the specified final time. The time history variables are written on a tape so as to be available for plotting. The output may be printed at each integration interval if desired. This logic is indicated in Fig. III-7.

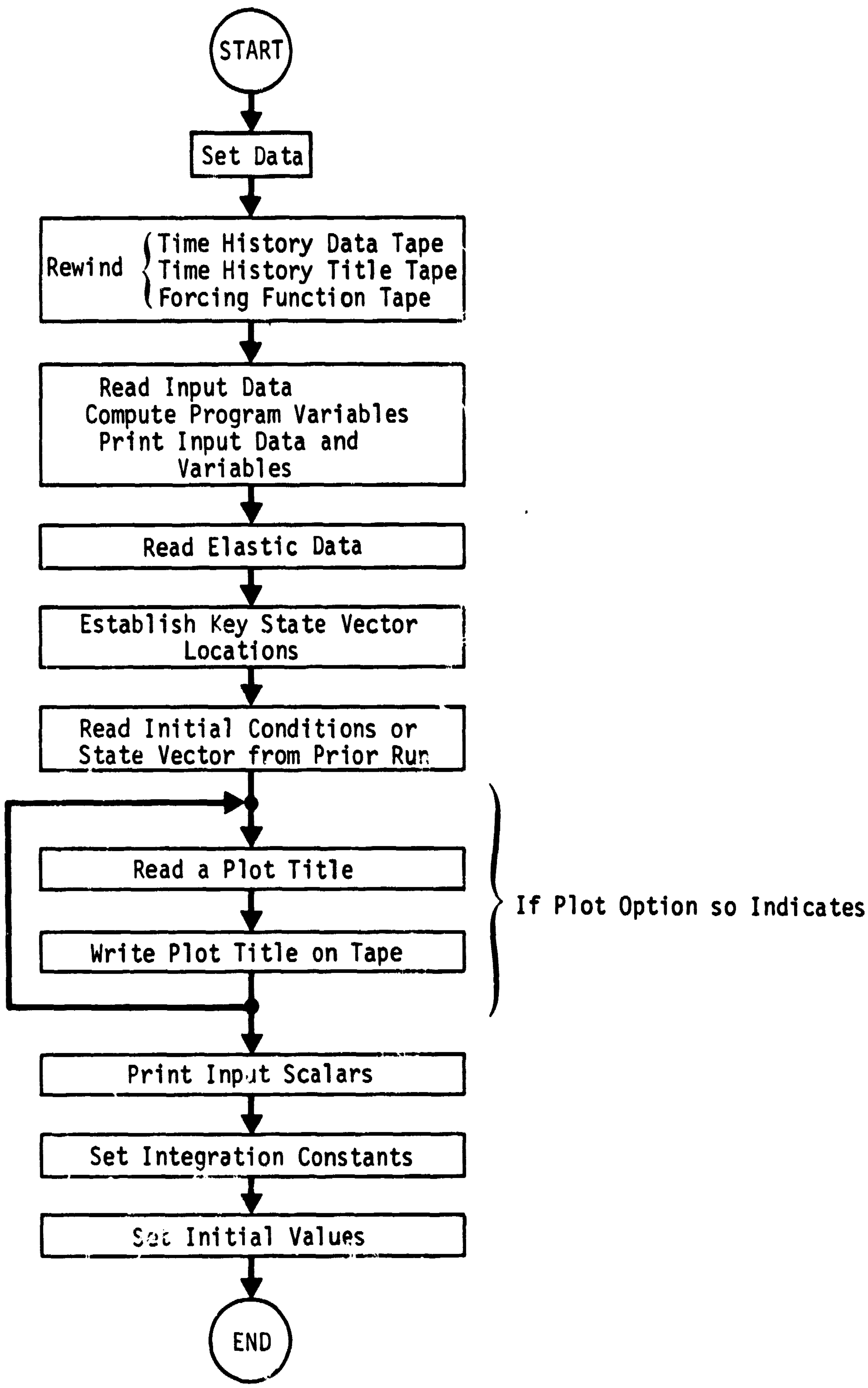


Fig. III-5 PROGRAM SETUP Logic Flow

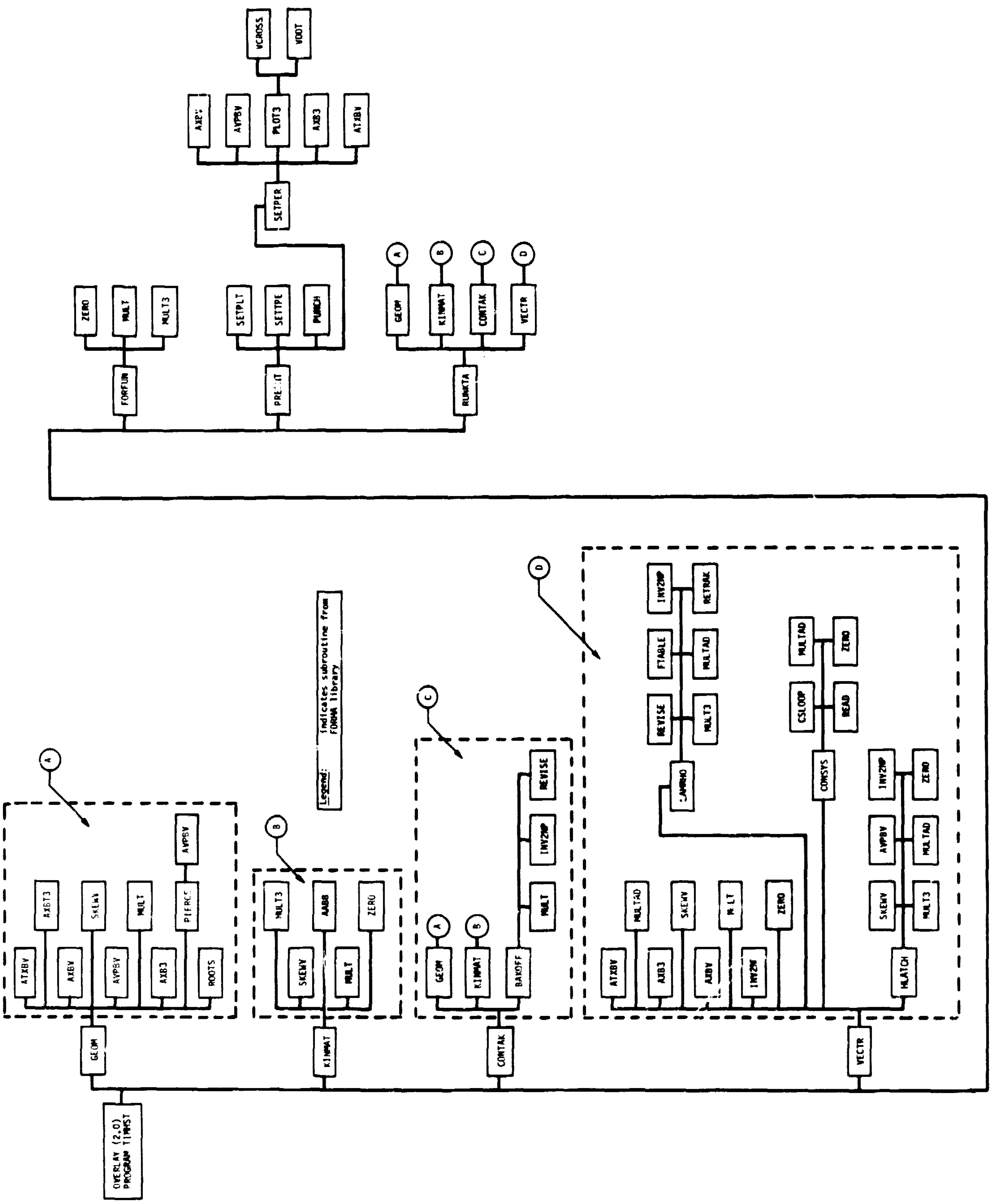


Fig. III-6 PROGRAM TIMHST Calling Structure