TR-250-1378
DECEMBER 1974

DIGITAL SIMULATION FOR POST-DOCKING RESPONSE

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Science and Engineering Directorate

Under Contract NAS8-29627

NORTHROP SERVICES, INC.
P. O. BOX 1484
HUNTSVILLE, ALABAMA 35807
TELEPHONE (205) 837-0580

TR–250–1378

DIGITAL SIMULATION FOR POST–DOCKING RESPONSE

DECEMBER 1974

by

James R. Roberts
Robert S. Todd

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
SYSTEMS DYNAMICS LABORATORY

Under Contract NAS8-29627

REVIEWED AND APPROVED BY:

J. W. Crenshaw,
Principal Investigator

M. A. Sloan, Jr., Manager
Dynamics and Control Section

NORTHROP SERVICES, INC.
HUNTSVILLE, ALABAMA

FOREWORD

This report presents the results of work performed by Northrop Services, Inc., Huntsville, Alabama in response to the requirements of Contract NAS8-29627 released from the Systems Dynamics Laboratory, Marshall Space Flight Center, Huntsville, Alabama. Technical coordination was maintained through Mr. Mario H. Rheinfurth (S&E-ED15).

ABSTRACT

This report documents the digital program, 2BODY, which simulates the translational and rotational motion of two connected rigid bodies and provides both digital and plot output. Relative rotation of the bodies at the connection is allowed, thereby providing a model suitable for studying system stability and response during a soft-dock regime. This document is to serve both as a users manual for the program as well as to provide all the details and background pertaining to the equations of motion and mathematical models, integration scheme, and input/output routines.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

TEXT	FORTRAN	DEFINITION
C	C	Joint constraint force damping coefficient
C_A	CA	Joint alignment torque damping coefficient
$\underline{d}$	D(3)	Joint or connection vector
$\dot{\underline{d}}$	DD(3)	Time rate of change of $\underline{d}$
$\underline{d}_1, \underline{d}_2$	D1(3),D2(3)	Position vector from the body center of mass to the joint for body 1, 2
$\underline{F}_1, \underline{F}_2$	F1(3),F2(3)	Total external force acting on body 1, 2 (including $\underline{F}_c$)
$\underline{F}_c$	F1(3),F2(3)	Joint constraint force
G	GAIN	Despin torque coefficient or gain
$\underline{\underline{I}}_1, \underline{\underline{I}}_2$	I1(3),I2(3)	Principal moments of inertia of body 1, 2
K	K	Joint constraint force spring or flexibility coefficient
K_A	KA	Joint alignment torque spring or flexibility coefficient
$\underline{L}_1, \underline{L}_2$	TQ1(3),TQ2(3)	Total external torque acting on body 1, 2 (including $\underline{L}_c$, $\underline{L}_A$, and $\underline{L}_D$)
$\underline{L}_A$	(not explicitly used)	Joint alignment torque
$\underline{L}_c$	TQ1(3),TQ2(3)	Joint constraint torque
$\underline{L}_D$	DTQ(3)	Despin torque
m_1, m_2	M1,M2	Mass of body 1,2
$\underline{Q}_1, \underline{Q}_2$	Q1(4),Q2(4)	Quaternions of body 1,2
$\underline{R}_1, \underline{R}_2$	R1(3),R2(3)	Position vector from the inertial frame to the center of mass of body 1,2
$\dot{\underline{R}}_1, \dot{\underline{R}}_2$	R1D(3),R2D(3)	Velocity of the center of mass of body 1,2 relative to inertial space

LIST OF SYMBOLS (Concluded)

TEXT	FORTRAN	DEFINITION
$\ddot{\underline{R}}_1, \ddot{\underline{R}}_2$	(not explicitly used)	Acceleration of the center of mass of body 1,2 relative to inertial space
T_1, T_2	T1(3,3),T2(3,3)	Coordinate transformation matrix resolving body 1,2 coordinates into the inertial frame
T'	TP(3,3)	Coordinate transformation matrix resolving body 2 coordinates into the body 1 frame
T_{LIM}	TLIM	Despin torque limit
$\underline{\nu}$	NU(3)	Error vector based on angular misalignment between the body X-axes
$\dot{\underline{\nu}}$	NUD(3)	Time rate of change of $\underline{\nu}$
$\underline{\omega}_1, \underline{\omega}_2$	W1(3),W2(3)	Angular velocity of body 1, 2

Section I

INTRODUCTION

The program 2BODY simulates the translational and rotational motion of two connected rigid bodies and provides digital output as well as graphical display by either Stromberg Carlson 4020 or online printer plots. The two bodies are assumed to be physically attached at a point; however, relative rotation about this point is possible. In fact, body 2 can be spinning about an axis in the proximity of the docking port axis. Such a vehicle model is analogous to the soft-dock regime (where capture or initial latch has been achieved, but final latch or hard dock has not) and can be used to study post-docking stability and response. Program options relative to vehicle dynamics include ideal attitude control of body 1, a joint or hinge alignment torque, and a despin torque.

This document is to serve both as a users manual for the program as well as to provide all the details and background on equations, models, and routines. Section II covers the equations of motion, coordinate systems, and model for the two-body system including the joint and associated constraint and alignment torques. The despin torquer model and the ideal attitude control scheme for body 1 are also discussed. Section III provides information on QAD2, an executive type, variable step-size integration scheme and the required state vector input. In Section IV, two extremely versatile, user-oriented input/output routines are described, namely CREAD and CRITE. All of the parameters required as input data are defined and discussed in Section V. This discussion is broken down into the categories of program control, case control, vehicle configuration and kinematics control, dynamics options control, and data output control.

Section VI contains a few remarks of interest to the user concerning program construction and improvements relative to data input that could be made. For the user who will be making additions and/or changes to the program, Section VII describes in detail the utility subroutines used in the program along with subroutine dependence and transfer of variables among routines either by argument list or labelled COMMON. Finally, Section VIII covers the applicable program language and computer. Although the Appendixes are all referenced in the body of this document, it might be mentioned that they contain such information as functional flow charts of the main routines, a complete program listing, and the input and output (printout and plots) for a sample case.

Section II

VEHICLE DYNAMICS AND MODELS

2.1 2BODY MODEL AND COORDINATE SYSTEMS

The two coupled bodies are modeled as two separate rigid bodies connected by a massless joint or hinge as illustrated in Figure 2-1. The center of mass (CM) of each body is located relative to an inertial coordinate system by $\underline{R}_1$ and $\underline{R}_2$. Although the simulation assumes that initially (at time = 0), $\underline{R}_1 = \dot{\underline{R}}_1 = 0$, there is no loss of generality. The position vectors $\underline{d}_1$ and $\underline{d}_2$ locate the joint or hinge point of each body relative to the center of mass.

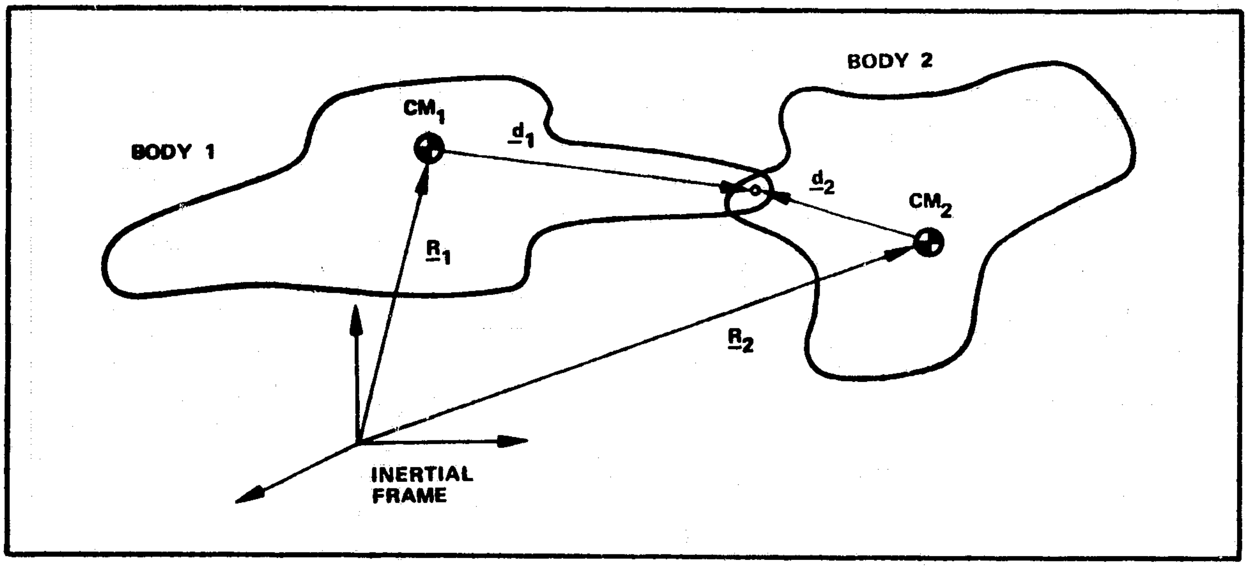


Figure 2-1. TWO COUPLED RIGID BODIES INERTIALLY REFERENCED

The relationship between the inertial and body coordinate frames is as follows:

$$\{X\}_I = T_1 \{X\}_1$$

$$\{X\}_I = T_2 \{X\}_2$$

$$\{X\}_1 = T_1^{-1} T_2\{X\}_2 = T_1^{\ T} T_2 \{X\}_2 = T'\{X\}_2$$

The transformation T_i, (i=1,2), is developed from 3-2-1 Euler angle sequence going from the inertial to the body i reference frame.

The translational and rotational equations of motion are given by

$$\left.\begin{array}{l} \ddot{\underline{R}}_i = \underline{F}_i/m_i \\ \text{and} \\ \underline{\underline{I}}_i \cdot \dot{\underline{\omega}}_i + \underline{\omega}_i \times \underline{\underline{I}}_i \cdot \underline{\omega}_i = \underline{L}_i \end{array}\right\} \quad i = 1,2$$

2.2 JOINT MODEL/CONSTRAINT AND ALIGNMENT TORQUE

The joint connecting the two rigid bodies is treated as an ideal docking mechanism during the soft dock regime (the bodies are physically attached, but not yet hard docked (rigidly attached), angular motion can exist between the two vehicles, and the target vehicle could be spinning about an axis close to or coinciding with the docking port axis).

The connection or joint is modeled as a massless spring and damper. From Figure 2-2, it is seen that

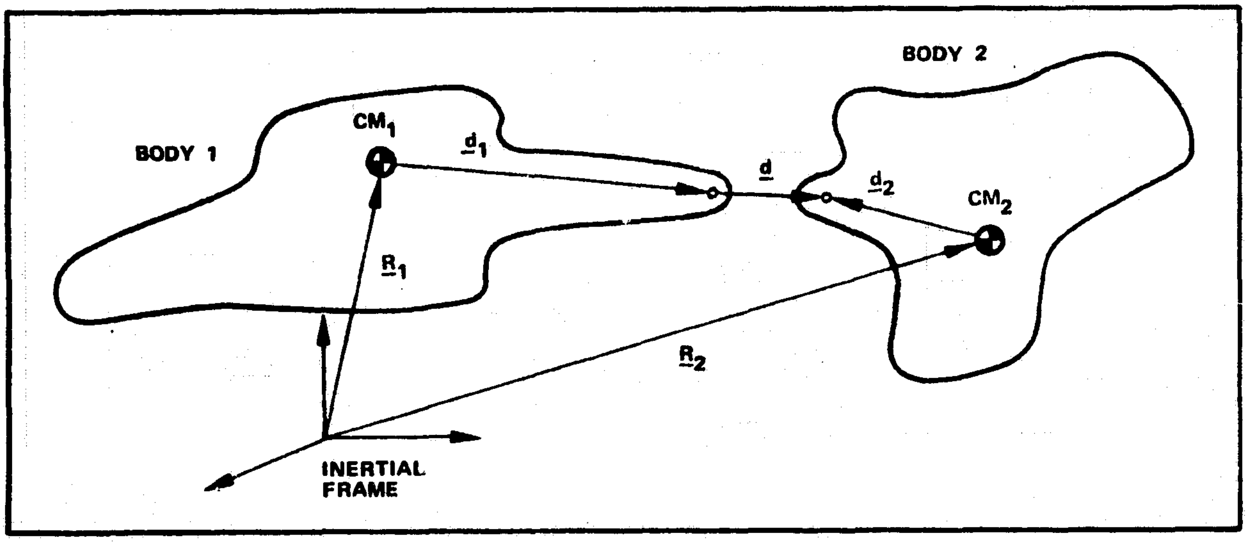


Figure 2-2. TWO COUPLED RIGID BODIES WITH IDEAL JOINT

$$\underline{d} = \underline{R}_2 - \underline{R}_1 + \underline{d}_2 - \underline{d}_1$$

and

$$\dot{\underline{d}} = \dot{\underline{R}}_2 - \dot{\underline{R}}_1 + \underline{\omega}_2 \times \underline{d}_2 - \underline{\omega}_1 \times \underline{d}_1 .$$

where it is assumed that $|\underline{d}| << |\underline{d}_i|$, $i = 1,2$.

Initially (at time = 0), however,

$$\underline{d} = \dot{\underline{d}} = 0$$

since the constraint relations

$$\underline{R}_2 = \underline{R}_1 + \underline{d}_1 - \underline{d}_2$$

and

$$\dot{\underline{R}}_2 = \dot{\underline{R}}_1 + \underline{\omega}_1 \times \underline{d}_1 - \underline{\omega}_2 \times \underline{d}_2$$

are used. The joint structure provides a physical constraint force and torque given by

$$\underline{F}_c = K\,\underline{d} + C\,\dot{\underline{d}}$$

and

$$\underline{L}_c = \underline{d} \times \underline{F}_c$$

where K and C represent the structural flexibility and damping coefficients. To damp the scissoring-type angular motion between the vehicles, an alignment torque is provided, i.e.,

$$\underline{L}_A = K_A\,\underline{\nu} + C_A\,\dot{\underline{\nu}}$$

where K_A and C_A are spring and damper coefficients and $\underline{\nu}$ is an error vector based on the amount of angular misalignment between the docking port axes. The program assumes the x-axes of the vehicles are to be aligned. Thus

$$\underline{\nu} = \hat{i}_1 \times \hat{i}_2$$

where $\hat{i}_j$ is the x-axis unit vector of body j, j = 1,2. Since

$$T' = [i_2 | j_2 | k_2] = [T'_{j1} | T'_{j2} | T'_{j3}]$$

then ν referenced to the body 1 coordinate frame is

$$\nu = \tilde{i}_1\, T'_{j1} = \begin{Bmatrix} 0 \\ -T'_{31} \\ T'_{21} \end{Bmatrix}$$

where the cross product matrix operator of a general vector

$$A = \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \end{Bmatrix}$$

if formed by

$$\tilde{A} = \begin{bmatrix} 0 & -A_3 & A_2 \\ A_3 & 0 & -A_1 \\ -A_2 & A_1 & 0 \end{bmatrix} .$$

Also, then

$$\dot{\nu} = \begin{Bmatrix} 0 \\ -\dot{T}'_{31} \\ \dot{T}'_{21} \end{Bmatrix}$$

where the elements of $\dot{\nu}$ can easily be determined from

$$\dot{T}' = T' \tilde{\omega}_2 - \tilde{\omega}_1 T' .$$

To determine values for K_A and C_A, the alignment torque is assumed of the form (also a typical rate-position feedback control law)

$$L = I \ddot{\theta} = - K_A \ddot{\theta} - C_A \dot{\theta}$$

where

I - an effective inertia of the sytem

θ - a relative angular displacement

$\dot{\theta}$ - a relative angular rate.

Rearranging terms results in

$$\ddot{\theta} + \frac{C_A}{I} \dot{\theta} + \frac{K_A}{I} \theta = 0$$

which is recognized as the equation of motion for a linear harmonic oscillator where

$$\frac{C_A}{I} = 2 \zeta \omega$$

and

$$\frac{K_A}{I} = \omega^2 \quad .$$

Thus

$$K_A = I\,\omega^2 = 4\pi^2\, I\, f^2$$

and

$$C_A = 2\,\zeta\, I\,\omega = 4\pi\quad I\,\zeta\, f$$

where

f - undamped natural frequency

ζ - non-dimensional damping factor.

The undamped natural frequency associated with alignment should usually be chosen to be lower than the precessional frequency of the spinning body. The effective inertia can be approximated by that of the reduced inertia of the system, i.e.,

$$I \cong \frac{I_1\, I_2}{I_1 + I_2}$$

where I_j is the inertia of body j, j = 1,2. Appendix A provides a set of graphs for selecting K_A and C_A on the basis of I, τ, and ζ where τ = 1/f.

2.3 DESPIN TORQUER MODEL

When one body is spinning, a torquer is available for despinning. The despin torque can be applied from either body (reflecting actual hardware considerations), but such application is on only the x-axis. For high relative spin rates, the despin torque is limited, providing a constant output, whereas for low relative spin rates the torquer output is linear as seen by the following equations and illustrated in Figure 2-3.

$$L_{DX} = \text{sgn}(\omega_{2x} - \omega_{1x})\; T_{LIM}\;, \qquad -\Delta\omega > \omega_{2x} - \omega_{1x} > \Delta\omega$$

$$L_{DX} = G(\omega_{2x} - \omega_{1x}) \qquad\qquad , \qquad -\Delta\omega < \omega_{2x} - \omega_{1x} < \Delta\omega$$

The despin torque may be either excluded from the simulation, turned on at time = 0, or activated on an energy condition. The energy condition might

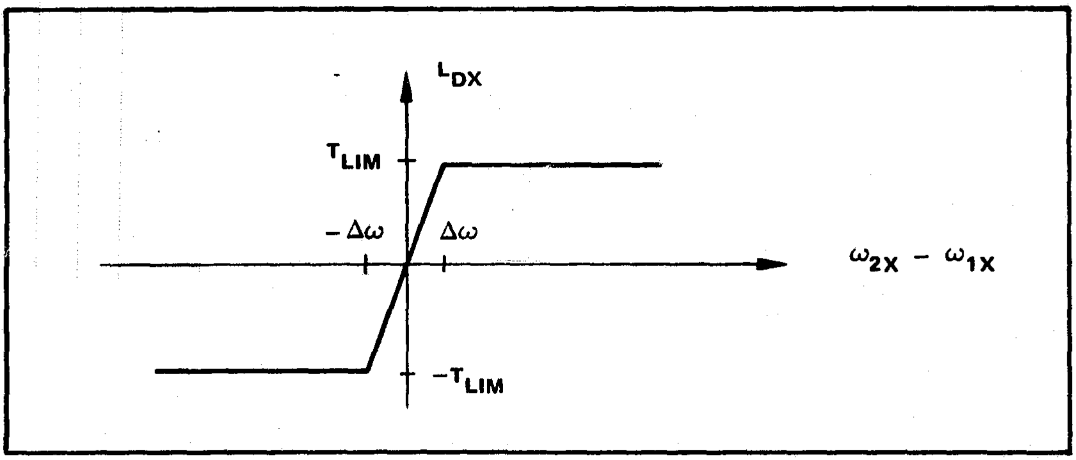


Figure 2-3. DESPIN TORQUER OUTPUT MODEL

be desirable if large initial misalignment conditions were used. Only after a highly transient condition damped out would it be appropriate to despin.

2.4 BODY 1 ATTITUDE CONTROL

If attitude control of body 1 (usually throught of as the chase vehicle, whereas body 2 is usually thought of as the target and possibly spinning vehicle) is desired, an ideal control system is available. The angular rates of body 1 are continuously set to zero to maintain a fixed vehicle attitude; however, the vehicle is still free to translate.

Section III

INTEGRATION SCHEME AND VARIABLES

The QAD2 integration routine employed uses a variable step-size scheme based on a fourth order Merson's method requiring first order differential equations (see reference 1). QAD2 is an executive routine and once called remains in control, calling four user-supplied subroutines (VDYN, AUTO, OUT, and TERM), until integration is complete (when the time reaches a specified value). Provision is made for output at predetermined intervals as well as the initial and final points.

VDYN supplies the differential equations expressing the rates of change for the state vector (SV):

$$SV(26) = \left\{ \begin{array}{l} Q_1(4) \\ \omega_1(3) \\ R_1(3) \\ \dot{R}_1(3) \\ Q_2(3) \\ \omega_2(3) \\ R_2(3) \\ \dot{R}_2(3) \end{array} \right\} .$$

Vehicle attitude is represented by Euler parameters, $Q_i(4)$, $i = 1,2$, during integration; however, input is in terms of Euler angles and output can be in either or both. STEP handles the variable integration step size control whereas TERM is presently a dummy routine. TERM can be used to provide integration termination on conditions other than a maximum value of the time. OUT takes the currently correct values of the time and the state vector

1. *Crenshaw, J. W., "Modified Versions of QUAD2," Northrop Services, Inc. Huntsville, Alabama, Interoffice Memorandum 69-7960-42, 28 May 1969.*

and calculates the desired output quantities. The data are then available for either the digital printout routine (CRITE) or storage in an array for the online printer or SC 4020 plot routines, or storage on tape for later SC 4020 plotting.

Section IV

INPUT/OUTPUT ROUTINES

A data card input routine called CREAD is used (see reference 2). Usage for each set of data to be read is CALL CREAD (DATA,N) which must be preceded by either DIMENSION DATA(N) or COMMON/DATA1/DATA,... . No FORMAT statement is required (an obvious advantage to using CREAD). For each call to CREAD, the data to be read must be separated by commas, but can continue onto additional cards. The field can be variable and all blanks are ignored. CREAD does require input variable data names (such as DATA) to be real; however, the actual numbers punched on the data card for the variable can be either real or integer (integer values being automatically converted to real) thereby eliminating one type of input error. If integer input data is required, it can be obtained from real data already read into the program by CREAD. This is accomplished by a round-and-fix algorithm. An example of data card input for a sample case is given in Appendix D which can be correlated to the calls to CREAD in 2BODY (the main program) and subroutine START, both listed in Appendix C.

A routine called CRITE is used for digital output (see reference 3). The routine provides a tabular format with up to ten columns of printout with multiple rows available for each print cycle. In this manner, vectors may be output in column form rather than in rows. In addition, any MxN ($N \leq 10$) matrix can also be output. Any desired six character name (heading) can be printed at the top of each column and page numbering of the digital output is automatic.

Usage whenever output is to be printed is CALL CRITE (O,KHDR,NA,N,1,IENT) which must be preceded by DIMENSION KHDR(N),NA(N) where

O - the output array

KHDR - the header array

---

2. *Crenshaw, J. W., "CREAD Input Routine," Northrop Interoffice Memorandum 69-7960-76, 11 December 1969.*
3. *Crenshaw, J. W., "Output Routine CRITE," Northrop Memorandum.*

NA - a dimension array

IENT - a control integer.

An example which follows best explains the usage of CRITE and the relationship between O, KHDR, and NA. The integer IENT must be set to 1 on the first call to CRITE in order to initialize the paging logic. IENT is then set to 2 by CRITE and thereafter should not be altered. Example:

```
SUBROUTINE OUT
DIMENSION  O(17), Q1(4), ANG1(3), W1(3), R1(3), R2(3)
DIMENSION  KHDR(5), NA(5)
DATA  KHDR / ' TIME ',' Q1 ',' ANG1 ',' W1 ','R1/R2 ' /
DATA NA / 1, 4, 3,-3, 6/
O(1) = TIME
O(2) = Q1(1)
O(3) = Q1(2)
O(4) = Q1(3)
O(5) = Q1(4)
O(6) = ANG1(1)
O(7) = ANG1(2)
O(8) = ANG1(3)
O(9) = W1(1)
O(10) = W1(2)
O(11) = W1(3)
O(12) = R1(1)
O(13) = R1(2)
O(14) = R1(3)
O(15) = R2(1)
O(16) = R2(2)
O(17) = R2(3)
CALL CRITE (O, KHDR, NA, 5, 1, IENT)
```

results in:

```
                                                      PAGE 1

  TIME       Q1        ANG1       R1/R2

  O(1)       O(2)      O(6)       O(12) ⎫
             O(3)      O(7)       O(13) ⎪
             O(4)      O(8)       O(14) ⎬   (values of)
             O(5)                 O(15) ⎪
                                  O(16) ⎪
                                  O(17) ⎭
```

It should be noted that the order of the variables followed in setting up the output array O must be maintained when setting up the column headings of KHDR. The sequence of integers making up NA corresponds to the dimension or number of entries to be printed in each column. Thus one entry, O(1), is printed in column 1, 4 lines or entries in column 2 for Q1 (O(2) through O(5)), and 3 entries in column 3 for ANG1 (O(6) through O(8)). The 3 entries for W1 will be skipped as implied by the negative sign preceding the corresponding 3 of NA. Then 6 entries (3 for R1 and 3 for R2, i.e. O(12) through O(17)) will be printed in column 4. An example of the resulting printout for another sample case is given in Appendix D which can be correlated to the material found in subroutine OUT listed in Appendix C.

Section V

DESCRIPTION OF PARAMETERS

The following provides a description of the parameters required as input data for the 2BODY simulation. The sequence in which the data are to be read is indicated in the comment card documentation in the MAIN program of the 2BODY listing (see Appendix C). In addition, reference can be made to the sample case in Appendix D.

5.1 PROGRAM CONTROL

DT ..... Initial integration step size

TMAX ... Cutoff or terminal value of time

ISTEP .. Print occurs every ISTEP integration steps

PTIME .. Print occurs each PTIME interval of time

DT is not a critical parameter and is set under QAD2 initialization in the main program. Either ISTEP or PTIME can be used to control output of digital printout. ISTEP is useful when a wide range of frequencies occur, since a good distribution of points is provided for plotting. The influence of either ISTEP or PTIME is suppressed by using a value above the expected or normal range thereby allowing the other of the two to predominate. Use of ISTEP only is usually more efficient.

5.2 CASE CONTROL

CASES ... Number of cases to process

As long as CASES is equal to the actual number of cases of data to be processed, the program will execute all the data and provide a normal exit. If CASES is less than the actual number of cases for which data has been submitted, only the number of cases indicated by CASES will be processed. If CASES is larger than the number of cases to be run, all the data will be executed and the program will error off while trying to read more data.

5.3 VEHICLE CONFIGURATION AND KINEMATICS CONTROL

M1,M2 ... Mass of body 1,2

I1,I2 ... Principal moments of inertia of body 1,2

ANG1,ANG2 Attitude of body 1,2

W1,W2 ... Angular velocity of body 1,2

D1,D2 ... Position vector from the body center of mass to the hinge or joint for body 1,2

5.4 DYNAMICS OPTIONS CONTROL

ICTRL ... = 0,1 .. Body 1 ideal control is inactive, active

IALIGN .. = 0,1 .. X-axis alignment torque inactive, active

IDSPIN .. = 0 .. Despin torque inactive

= 1 .. Despin torque active at time = 0

= 2 .. Despin torque activated on energy condition (the value of the energy flag (EPS) is set in a DATA statement of subroutine OUT)

IDTQRF .. = 1,2 .. Despin torque reference frame in body 1,2

KA ...... Alignment torque spring coefficient

CA ...... Alignment torque damping coefficient

GAIN .... Despin torque gain

TLIM .... Despin torque limit

5.5 DATA OUTPUT CONTROL

IPRNT ... = 0 .. Digital printout is suppressed

= 1 .. Digital printout is requested

ITAPE ... = 0 .. No output tape is created

= 1 .. A single precision output data tape is created

= 2 .. A double precision output data tape is created

IPLOT ... = 0 .. Plot output is suppressed

= 1 .. Online printer plots are requested

= 2 .. SC 4020 plots are requested

= 3 .. Both printer and 4020 plots are requested

When IPRNT = 1, specific data will be output by CRITE (see Section IV). When ITAPE = 1 or 2, data to be stored on tape is determined by the variables listed after the tape WRITE statement in subroutine OUT. For a double precision data tape, the variables to be listed are the elements of O, the double precision output array. For a single precision data tape, the variables to be listed are the elements of OP, the single precision version of the O array. When IPLOT = 1, 2, or 3, then two data cards are required for each plot requested. On the first card, the location or dimension number of the variables in the OP array which are to be independent and dependent variable respectively are listed in 2I5 format. The second card is to contain the plot title in 18A4 format.

Section VI

REMARKS

The program itself is constructed in a modular fashion and consequently will easily yield to future additions and/or changes to more sophisticated math models. Although the program has been cleaned up and streamlined whenever possible, there are still two items which could be changed adding a slight benefit to the user. These include the means of data input relative to an energy level that could be used to trigger the despin torquer and the alignment torque spring and damper coefficients.

First, the energy threshold (EPS) on which the despin torque is activated is input by a DATA statement in subroutine OUT. The need for such an input could be eliminated by developing and incorporating into the simulation an appropriate energy expression. Secondly, there is an alternative to determining the values of K_A and C_A from the graphs of Appendix A-1 and inputing on cards. If appropriate expressions were developed and included in the simulation for the effective inertia I and the precessional frequency f_p along with the previously given equations for K_A and C_A, then two different but more desirable quantities could be input by cards. These would be the damping factor, ζ, and the frequency ratio, f_r, between alignment and precession. Generally it would be desirable to have an alignment frequency lower than the precessional frequency.

Section VII

SUBPROGRAMS REQUIRED/INTERRELATIONSHIP

A functional diagram indicating the overall logic flow in the simulation along with a functional flow chart of the main routines of the simulation are given in Appendix B. In the following, four tables are shown. Table 7-1 is a matrix of routines indicating the dependence of a particular subroutine on other subroutines. Table 7-2 lists the utility subroutines giving the name, mathematical operation, usage, and dimensioning whereas Table 7-3 lists the Univac 1108 System Routines that are used in the program. Finally, Table 7-4 lists the main routines of the 2BODY program showing both the usage and the transfer of variables either through an argument list or labelled COMMON.

Table 7-1. SUBROUTINE DEPENDENCE

THESE ROUTINES REQUIRE THESE ROUTINES (AS INDICATED BY AN X)	1108 SYSTEM ROUTINES							UTILITY SUBROUTINES											PROGRAM SUBROUTINES																	
	DTAN	DSQRT	DSIN	DCOS	DATAN2	DASIN	DACOS	VMAGN	ULR6	ULR5	ULR2	ULR1	TWST	SUBV	TPRD	MPRD	CROSS	CMOD	XL8	VDYN	TERM	STEP	START	ROT8	RFILT	QAD2	PACK	OUT	GRAPH2	GRAPH1	GSTOR	GSPEC	DPLOTM	DOCK	CRITE	CREAD
2BODY																							X			X										X
DOCK														X	X	X	X																			
GRAPH1																																	X			
GRAPH2*																											X									
OUT	X	X	X	X	X	X	X				X				X	X	X			X											X				X	
QAD2																		X		X	X	X						X								
START									X			X				X	X								X				X	X		X				X
STEP		X						X																												
VDYN										X						X	X		X					X										X		
TWST			X	X																																
ULR1													X			X																				
ULR2					X		X																													
ULR6		X																																		
VMAGN		X																																		

**GRAPH2 also requires the following SC 4020 plot routines: IDENT, GRID1V, PRINTV, QUIK3V, QUIK3L, and ENDJOB.*

Table 7-2. UTILITY SUBROUTINES

NAME	OPERATION	USAGE/DIMENSION
CMOD	Replaces library modulo function {Z = X - Y INT(X/Y)}	Z = CMOD(X,Y)
CROSS	Vector cross product $\underline{C} = \underline{A} \times \underline{B}$	CALL CROSS(A,B,C) A(3), B(3), C(3)
MPRD	Matrix product C = AB	CALL MPRD(A,B,C,L,M,N) A(L,M), B(M,N), C(L,N)
TPRD	Transpose matrix product $C = A^T B$	CALL TPRD(A,B,C,L,M,N) A(L,M), B(L,N), C(M,N)
SUBV	Vector subtraction $\underline{C} = \underline{A} - \underline{B}$	CALL SUBV(A,B,C) A(3), B(3), C(3)
TWST	Develops a rotation matrix T for a single rotation of THETA radians about the N-axis measured from the unprimed frame, i.e., x = Tx'	CALL TWST(THETA,T,N) T(3,3)
ULR1	Develops a rotation matrix T for an L-M-N Euler rotation sequence for angles Q(1), Q(2), and Q(3) respectively with Q(1) measured from the unprimed frame, i.e., $x = [Q(1)]_L [Q(2)]_M [Q(3)]_N x' = Tx'$	CALL ULR1(Q,T,L,M,N) Q(3), T(3,3)
ULR2	Generates the Euler angles Q(1), Q(2), and Q(3) from a rotation matrix T for an L-M-N Euler rotation sequence and Q(1) measured from the unprimed frame, i.e., x = Tx', (inverse of ULR1)	CALL ULR2(T,Q,L,M,N) T(3,3), Q(3)
ULR5	Develops a rotation matrix T from Euler parameters Z	CALL ULR5(Z,T) Z(4), T(3,3)
ULR6	Develops Euler parameters Z from a rotation matrix T	CALL ULR6(T,Z) T(3,3), Z(4)
VMAGN	N-dimensional vector magnitude $\lvert\underline{A}\rvert = \sqrt{A_1^2 + A_2^2 + \ldots + A_N^2}$	VMAGN(A,N) A(N)

Table 7-3. UNIVAC 1108 SYSTEM ROUTINES

NAME	OPERATION	USAGE
DACOS	Double precision arc cosine $\cos^{-1}(X)$	DACOS(X)
DASIN	Double precision arc sine $\sin^{-1}(X)$	DASIN(X)
DATAN2	Double precision arc tangent of a ratio (X/Y) $\tan^{-1}(X/Y)$	DATAN2(X,Y)
DCOS	Double precision cosine $\cos(X)$	DCOS(X)
DSIN	Double precision sine $\sin(X)$	DSIN(X)
DSQRT	Double precision square root $\sqrt{X}$	DSQRT(X)
DTAN	Double precision tangent $\tan(X)$	DTAN(X)

Table 7-4. TRANSFER OF VARIABLES AMONG ROUTINES

ROUTINE (USAGE)	LABELLED COMMON	
	INPUT	OUTPUT
1 2BODY (MAIN PROGRAM)	NTGRAT/TMAX,ISTEP,PTIME	None
2 CALL START(SV,K) (1)(1)	None	ALGNTQ/KA,CA DSPNTQ/GAIN,TLIM D/D1(3),D2(3) FLAGS/ICTRL,IALIGN,ITQON,IDTQRF NTGRAT/TMAX,ISTEP,PTIME OUTPUT/IPRNT,ITAPE,IPLOT PLOT/NPP,INUSE(MV),GDATA(MPP,MV) VEH/M1,I1(3),M2,I2(3)
3 CALL QAD2(T,SV,26,KEY,DT,TMAX,ISTEP,PTIME (1)(1) (1)(1) (1) (1) (1) VDYN,STEP,OUT TERM) (5) (SUBROUTINES)	None	None
4 CALL OUT(T,SV,J) (3)(3)(3)	ALGNTQ/KA,CA VEH/M1,I1(3),M2,I2(3) VDYN1/W1(3),W2(3),T1(3,3)T2(3,3),D(3),DD(3) VDYN2/R1(3),R1D(3),R2(3),R2D(3) DOCK1/F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3) DOCK2/NU1(3),NUD1(3),TP(3,3) FLAGS/ICTRL,IALIGN,ITQON,IDTQRF OUTPUT/IPRNT,ITAPE,IPLOT PLOT/NPP,INUSE(MV),GDATA(MPP,MV)	None
5 CALL VDYN(T,SV,SVD) (4)(4)(3)	VEH/M1,I1(3),M2,I2(3) D/D1(3),D2(3) DOCK1/F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3) FLAGS/ICTRL,IALIGN,ITQON,IDTQRF	VDYN1/W1(3),W2(3),T1(3,3),T2(3,3),D(3),DD(3) VDYN2/R1(3),R1D(3),R2(3),R2D(3)

Table 7-4. TRANSFER OF VARIABLES AMONG ROUTINES (Concluded)

ROUTINE (USAGE)	LABELLED COMMON	
	INPUT	OUTPUT
6 CALL DOCK	ALGNTQ/KA,CA DSPNTQ/GAIN,TLIM D/D1(3),D2(3) FLAGS/ICTRL,IALIGN,ITQON,IDTQRF VDYN1/W1(3),W2(3),T1(3,3),T2(3,3),D(3),DD(3)	DOCK1/F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3) DOCK2/NU1(3),NUD1(3),TP(3,3)
7 CALL XL8(SV(J),M1,FKI,SVD(L)) ↑ (5) ↑ (5) ↑ (5) ↓ (3) CALL for body 1: J=11, K=1, L=8 CALL for body 2: J=24, K=2, L=21	None	None
8 CALL ROT8(SV(J),IK,TQK,SVD(J)) ↑ (5) ↑ (5) ↑ (5) ↓ (3) CALL for body 1: J=1, K=1 CALL for body 2: J=14, K=2	None	None

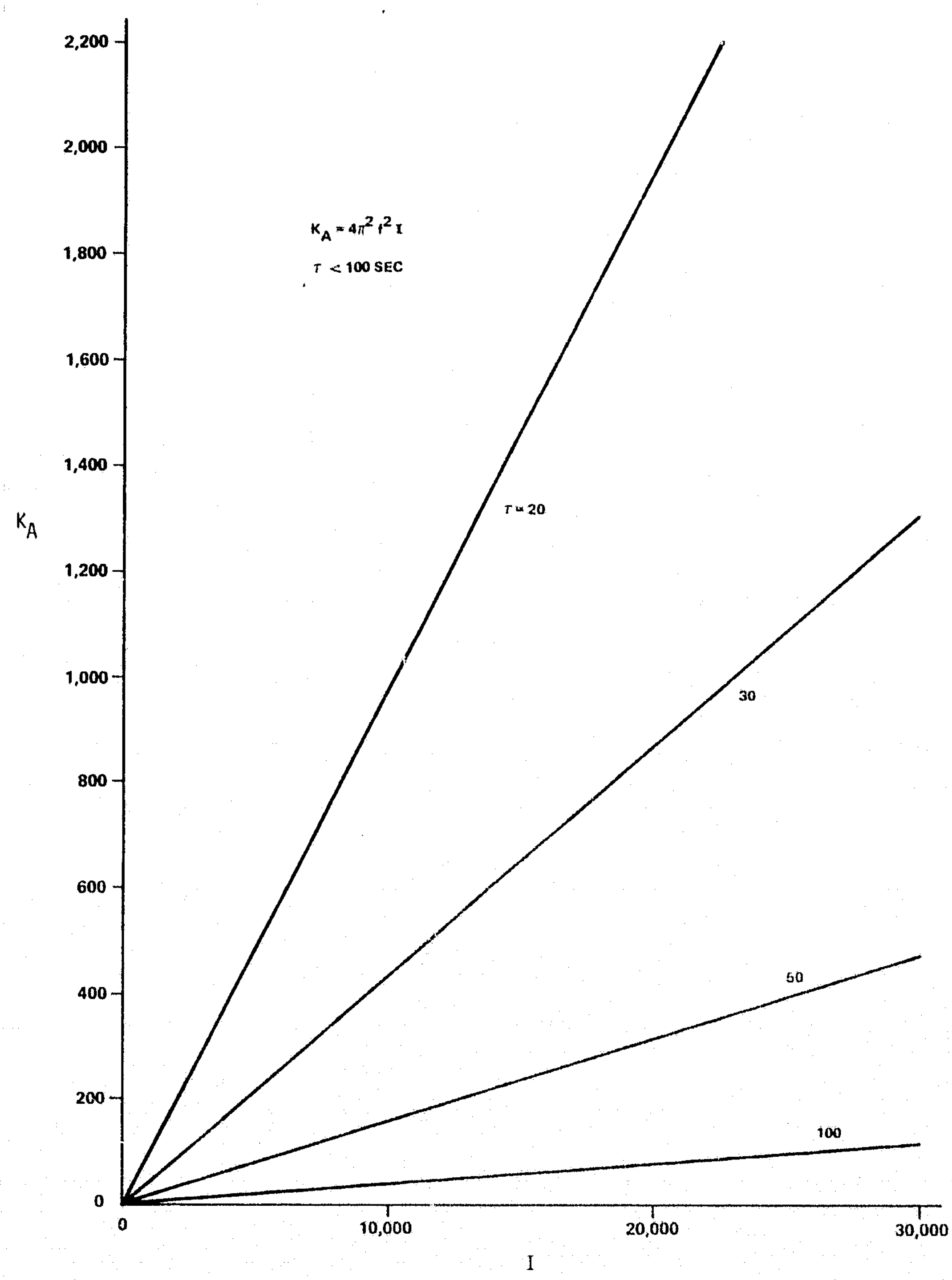
Section VIII

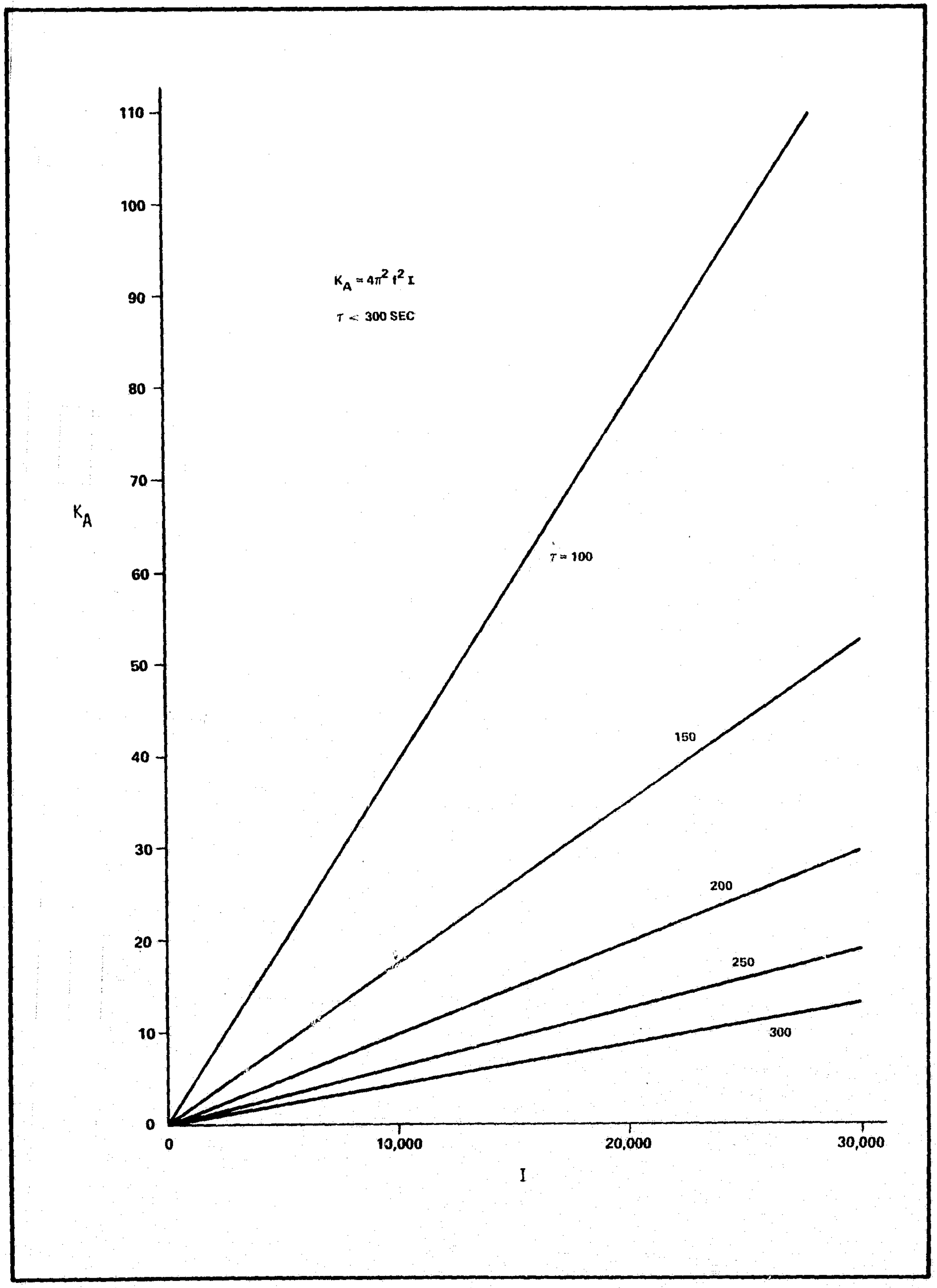
LANGUAGE/MACHINE

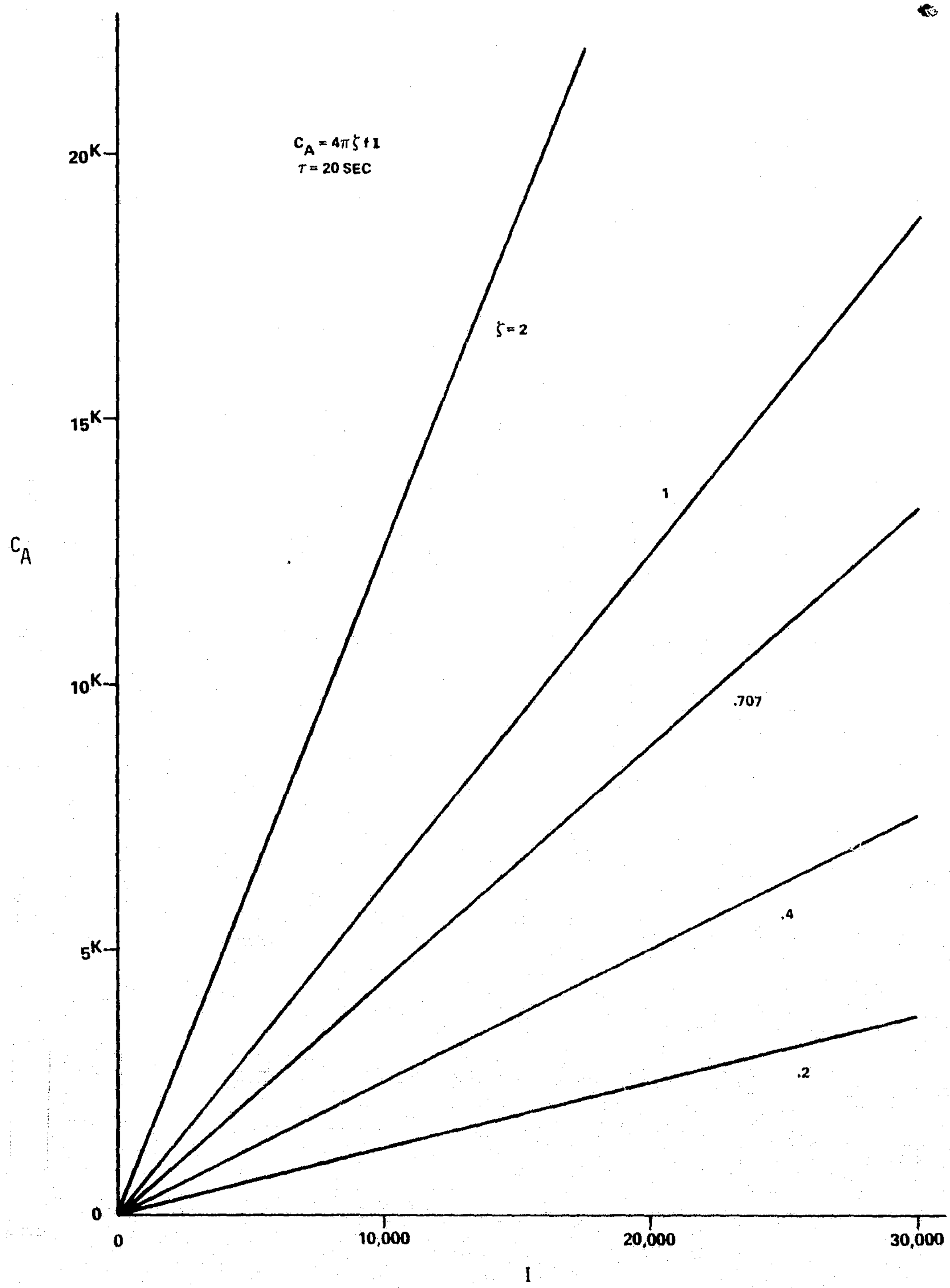
The 2BODY digital simulation program, a complete listing of which is given in Appendix C, is written in Fortran for the Univac 1108. The program is in double precision throughout except for the online and SC4020 plot routines.

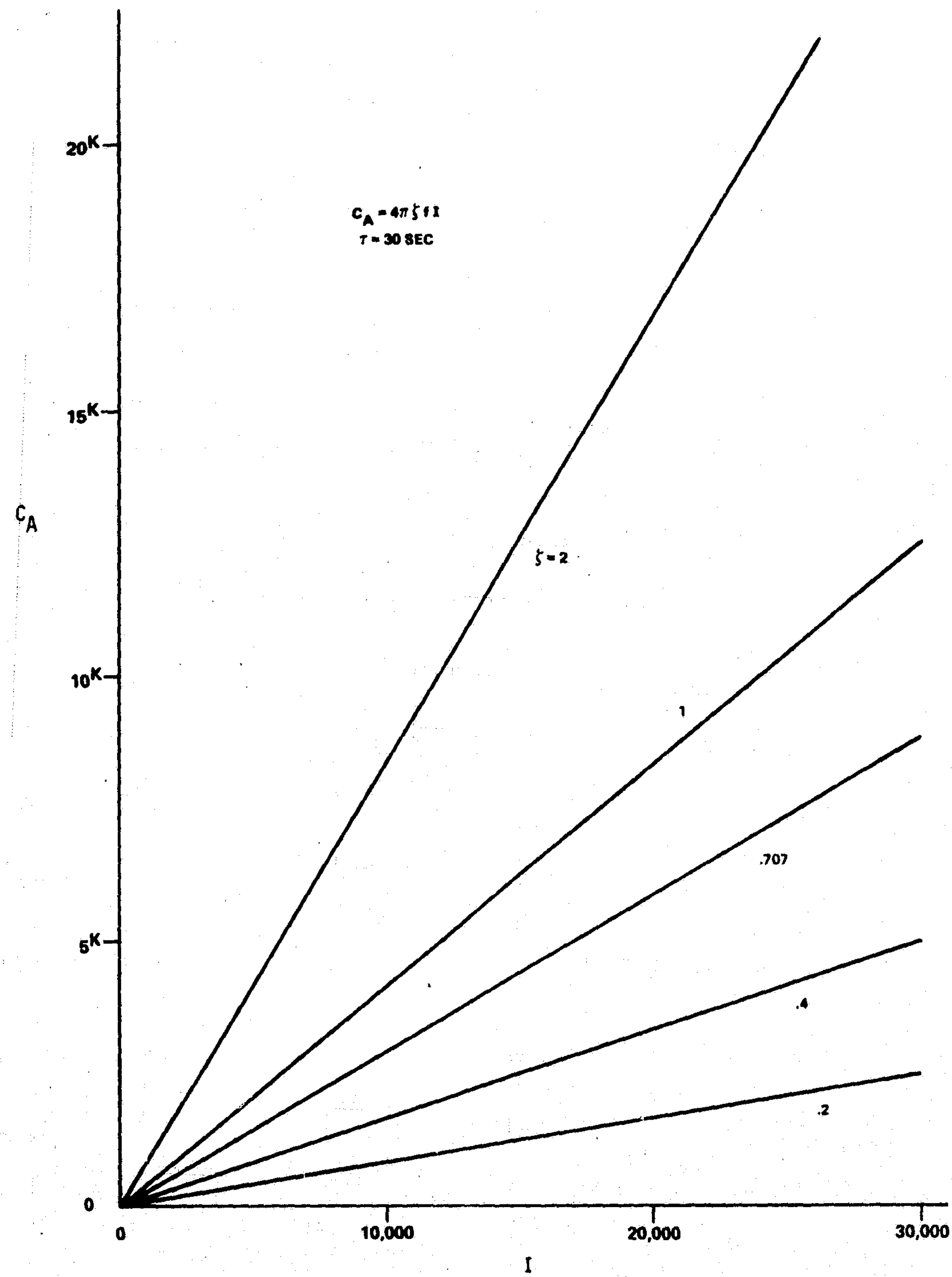
Appendix A

GRAPHS OF K_A AND C_A FOR I, τ, AND ζ

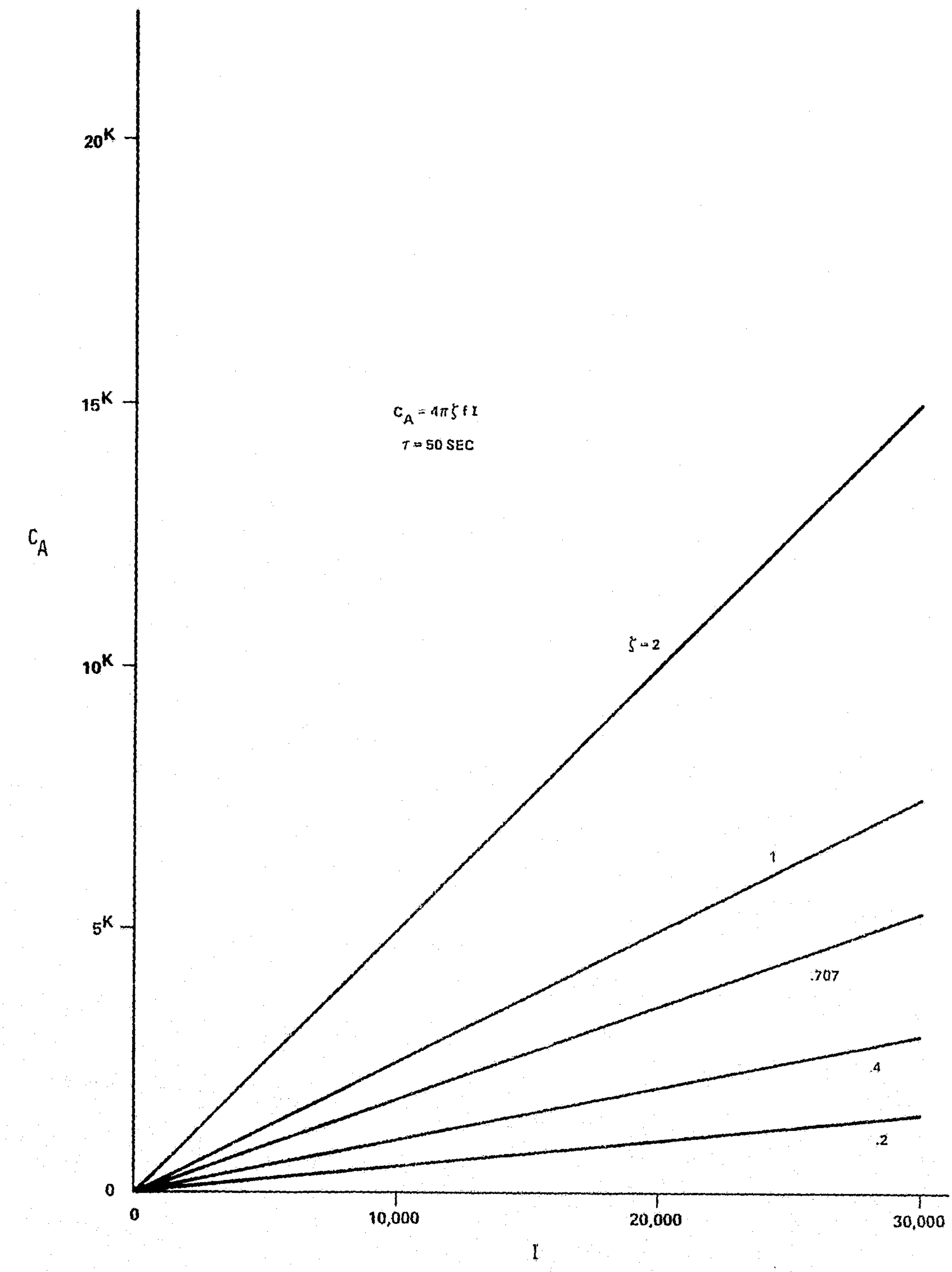

2,200
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
K_A
$K_A = 4\pi^2 f^2 I$
τ < 100 SEC
τ = 20
30
50
100
0
10,000
20,000
30,000
I

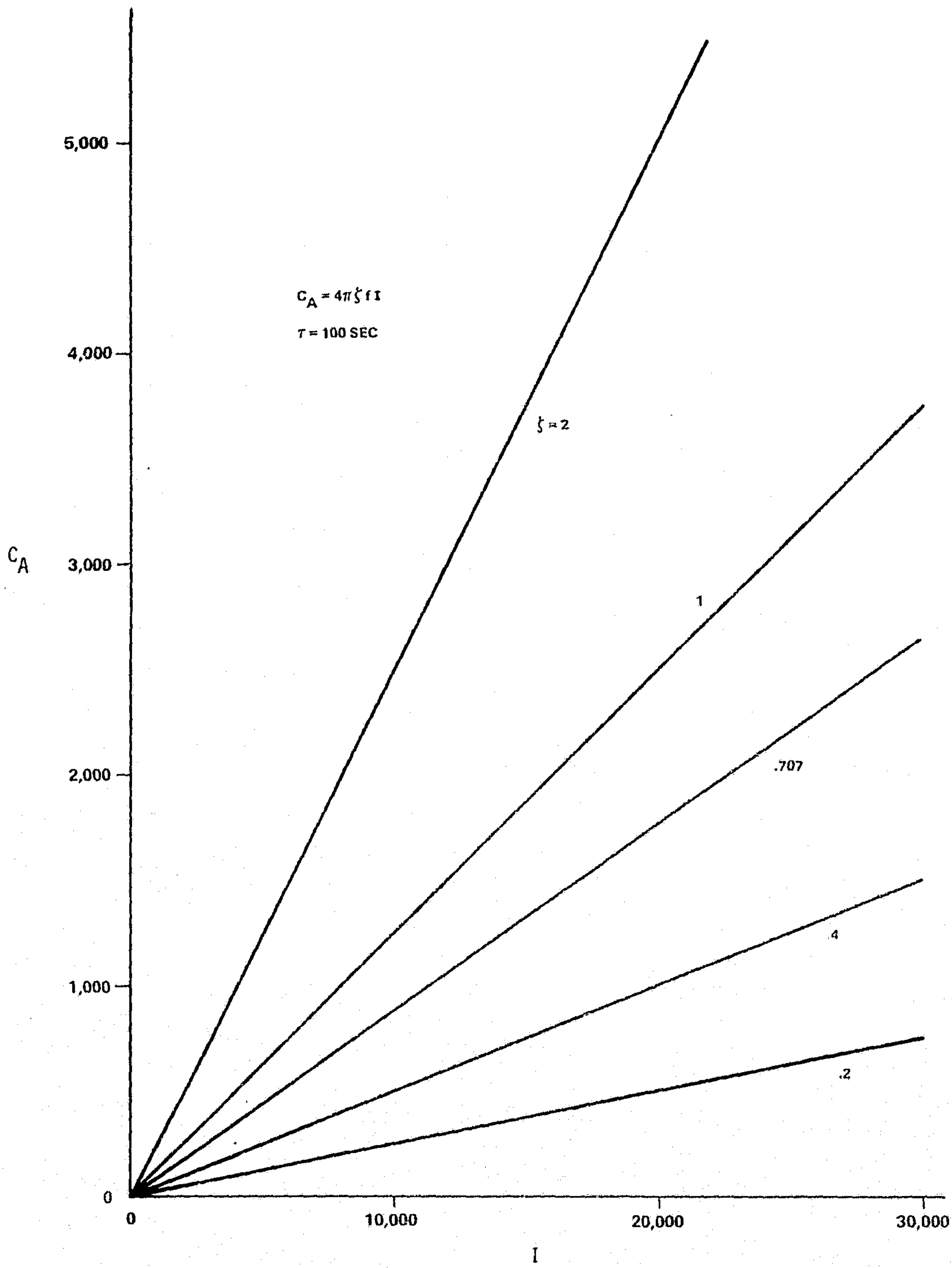

110
100
90
80
70
60
50
40
30
20
10
0
K_A
$K_A = 4\pi^2 f^2 I$
$\tau <$ 300 SEC
τ = 100
150
200
250
300
0
10,000
20,000
30,000
I


$C_A = 4\pi\zeta f I$
τ = 20 SEC
ζ = 2
1
.707
.4
.2
20K
15K
10K
5K
0
C_A
0
10,000
20,000
30,000
I

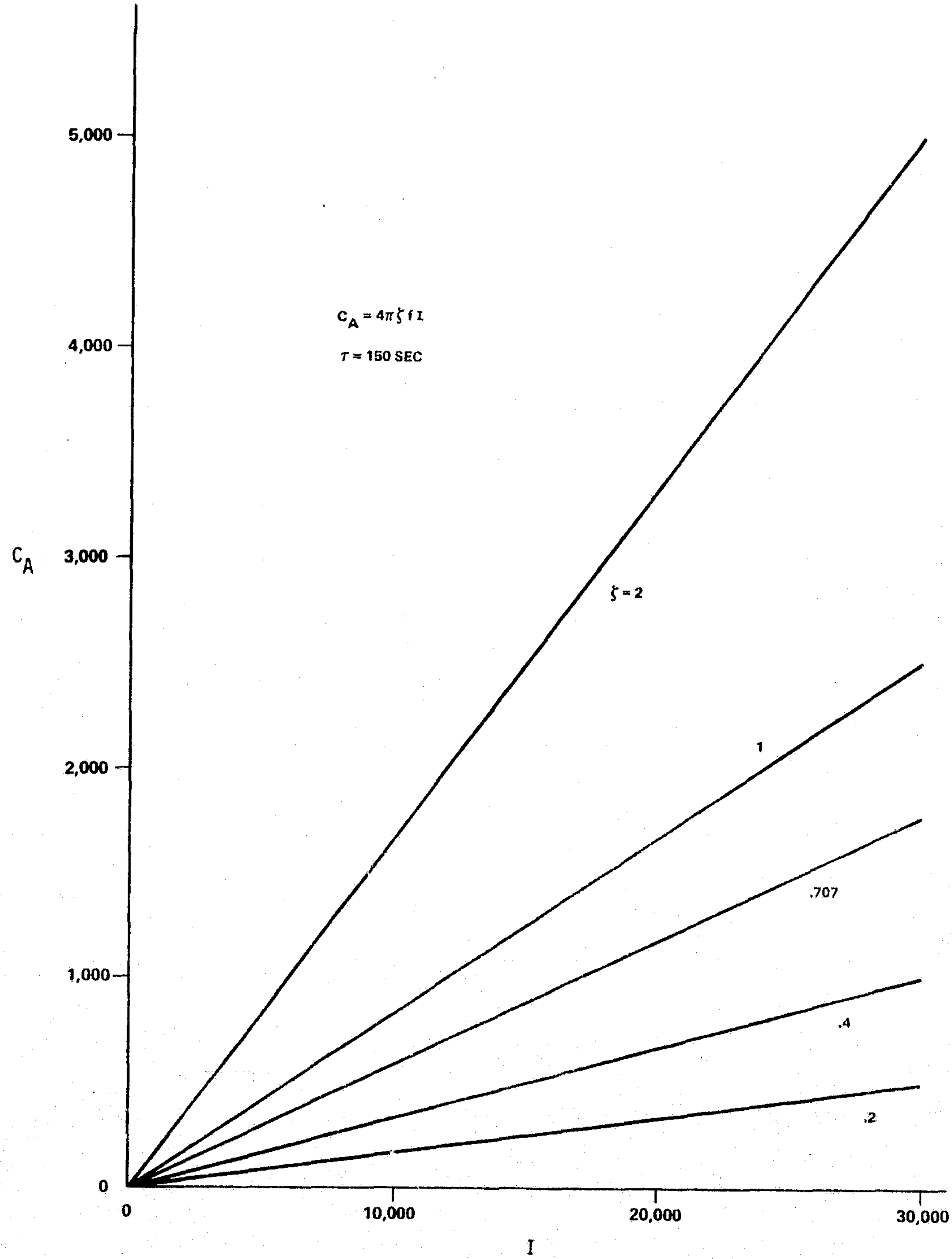


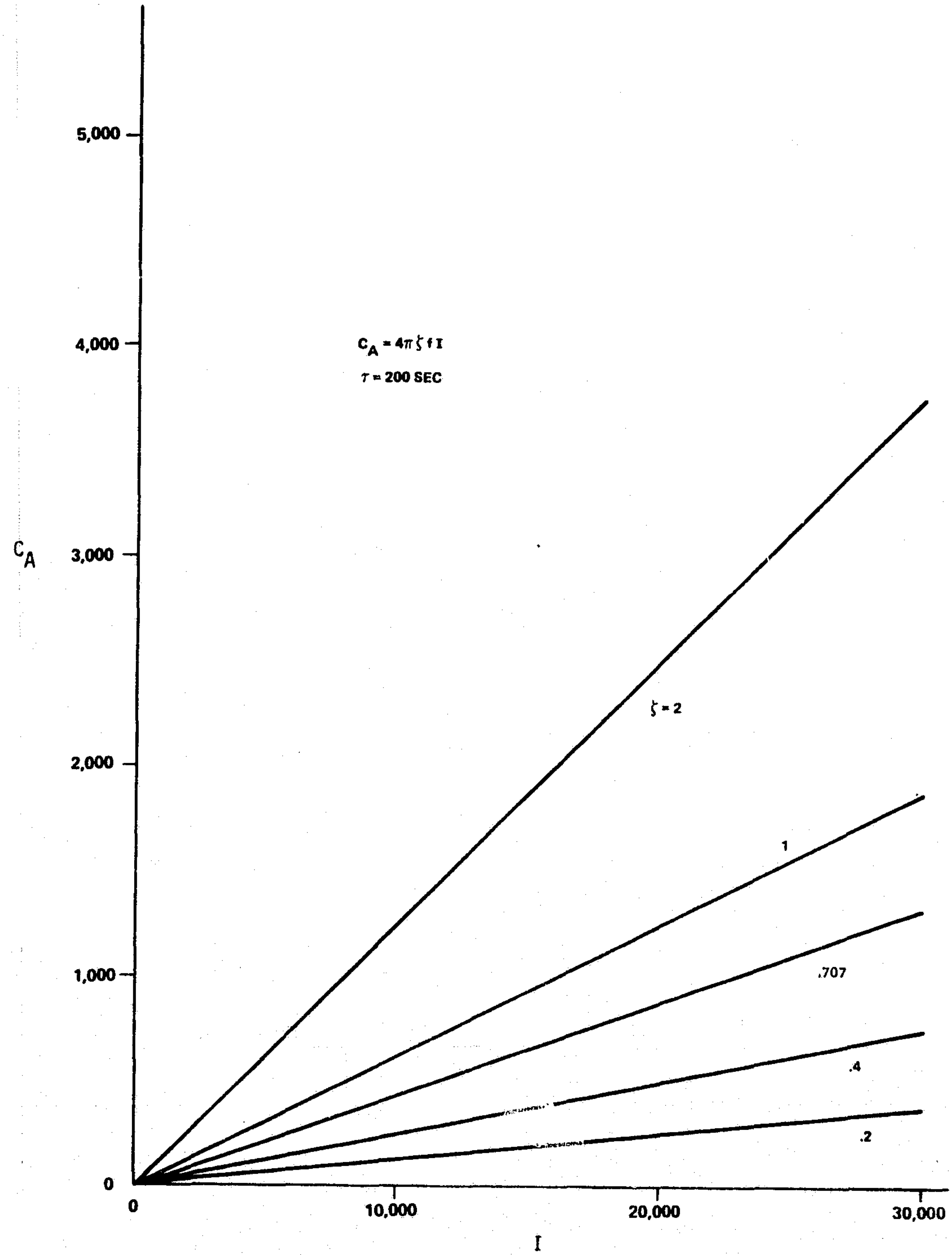
$C_A = 4\pi \zeta f I$
$\tau = 30$ SEC
20K
15K
10K
5K
0
C_A
$\zeta = 2$
1
.707
.4
.2
0
10,000
20,000
30,000
I

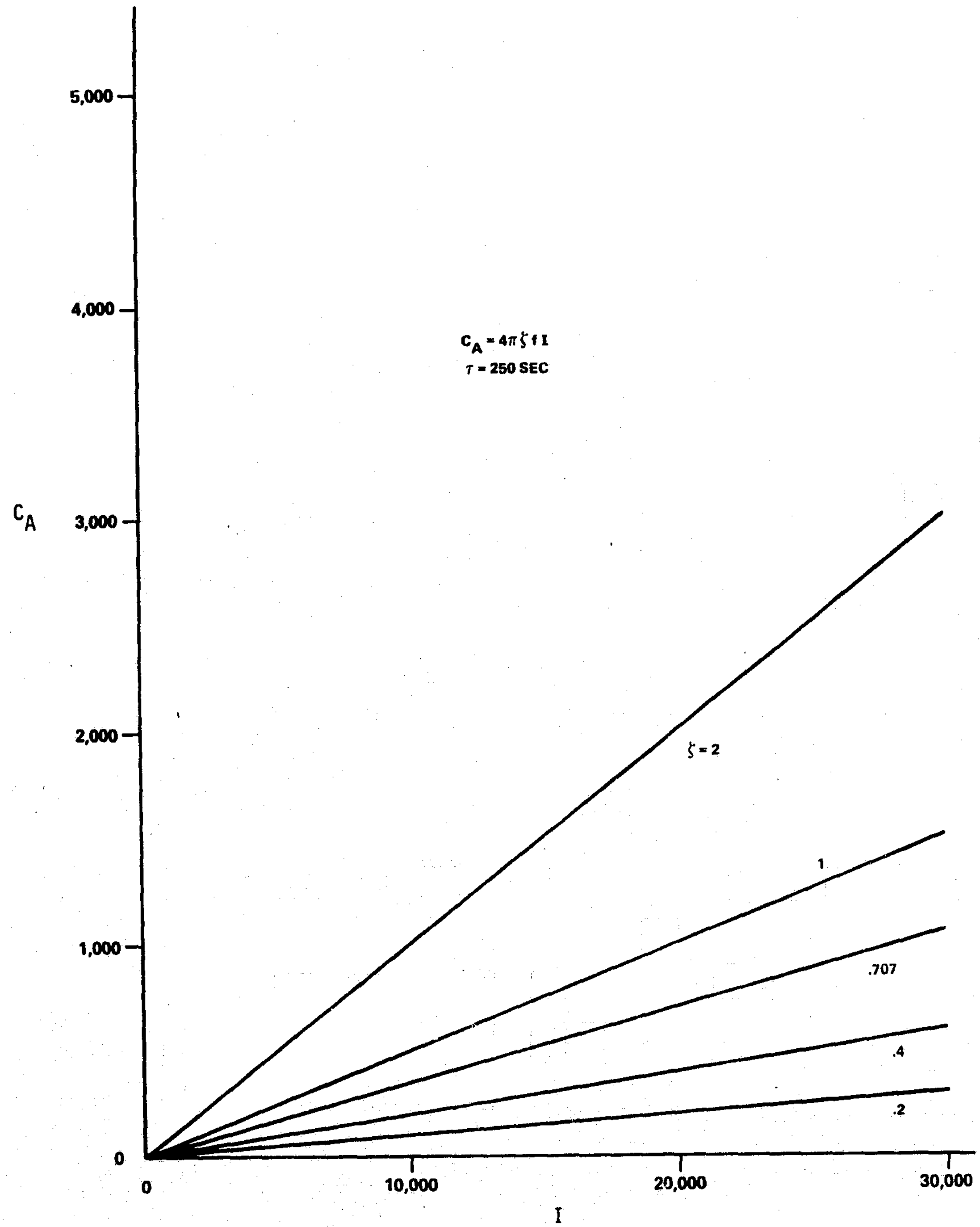

$C_A = 4\pi\zeta f I$
τ = 50 SEC
C_A
20K
15K
10K
5K
0
ζ = 2
1
.707
.4
.2
0
10,000
20,000
30,000
I



$C_A = 4\pi\zeta f I$
τ = 100 SEC
ζ = 2
1
.707
.4
.2
C_A
5,000
4,000
3,000
2,000
1,000
0
0
10,000
20,000
30,000
I


5,000
4,000
3,000
2,000
1,000
0
C_A
$C_A = 4\pi \zeta f I$
τ = 150 SEC
ζ = 2
1
.707
.4
.2
0
10,000
20,000
30,000
I

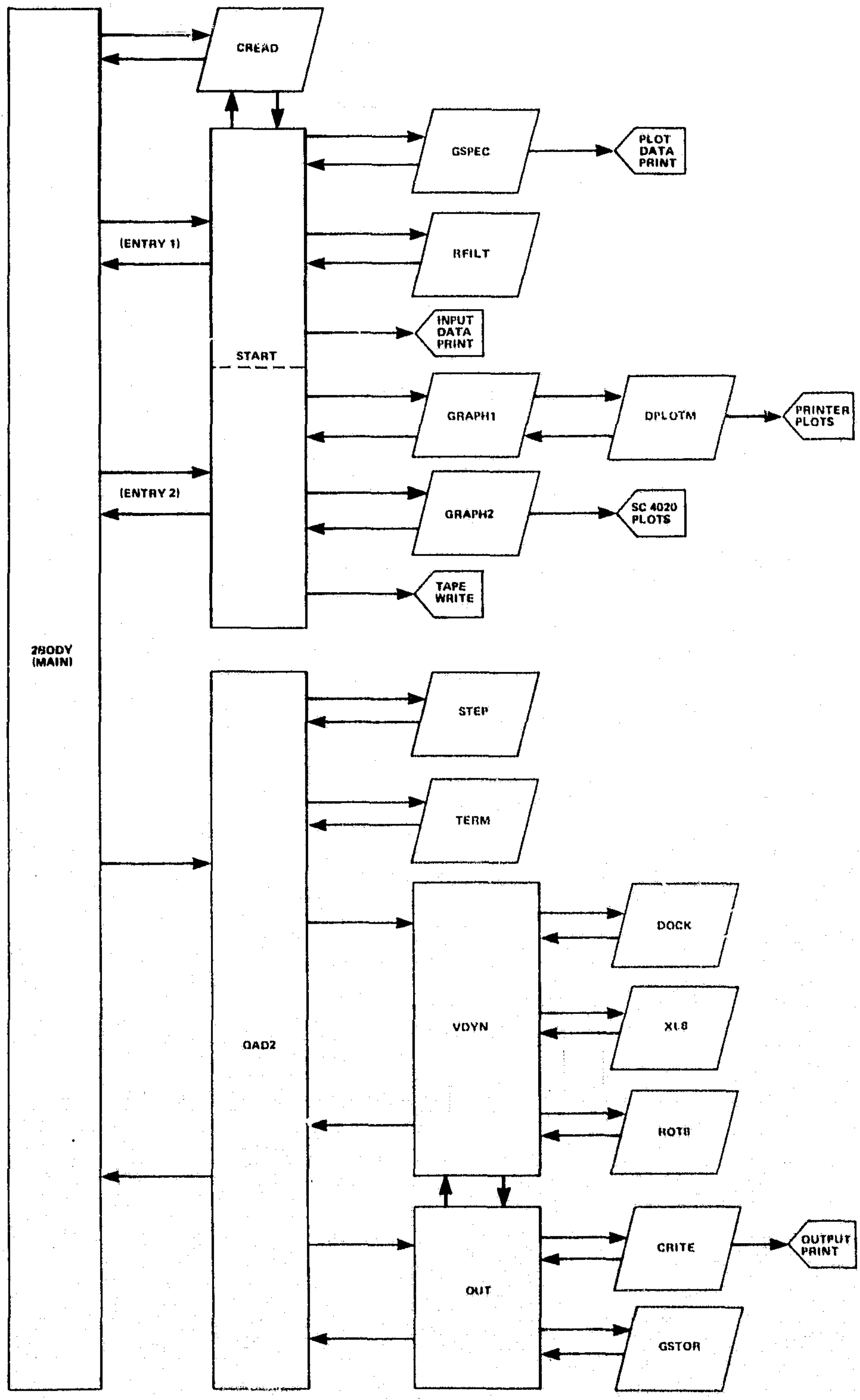

$C_A = 4\pi \zeta f I$
τ = 200 SEC
C_A
5,000
4,000
3,000
2,000
1,000
0
0
10,000
20,000
30,000
I
ζ = 2
1
.707
.4
.2


$C_A = 4\pi \zeta f I$
τ = 250 SEC
C_A
5,000
4,000
3,000
2,000
1,000
0
ζ = 2
1
.707
.4
.2
0
10,000
20,000
30,000
I

Appendix B

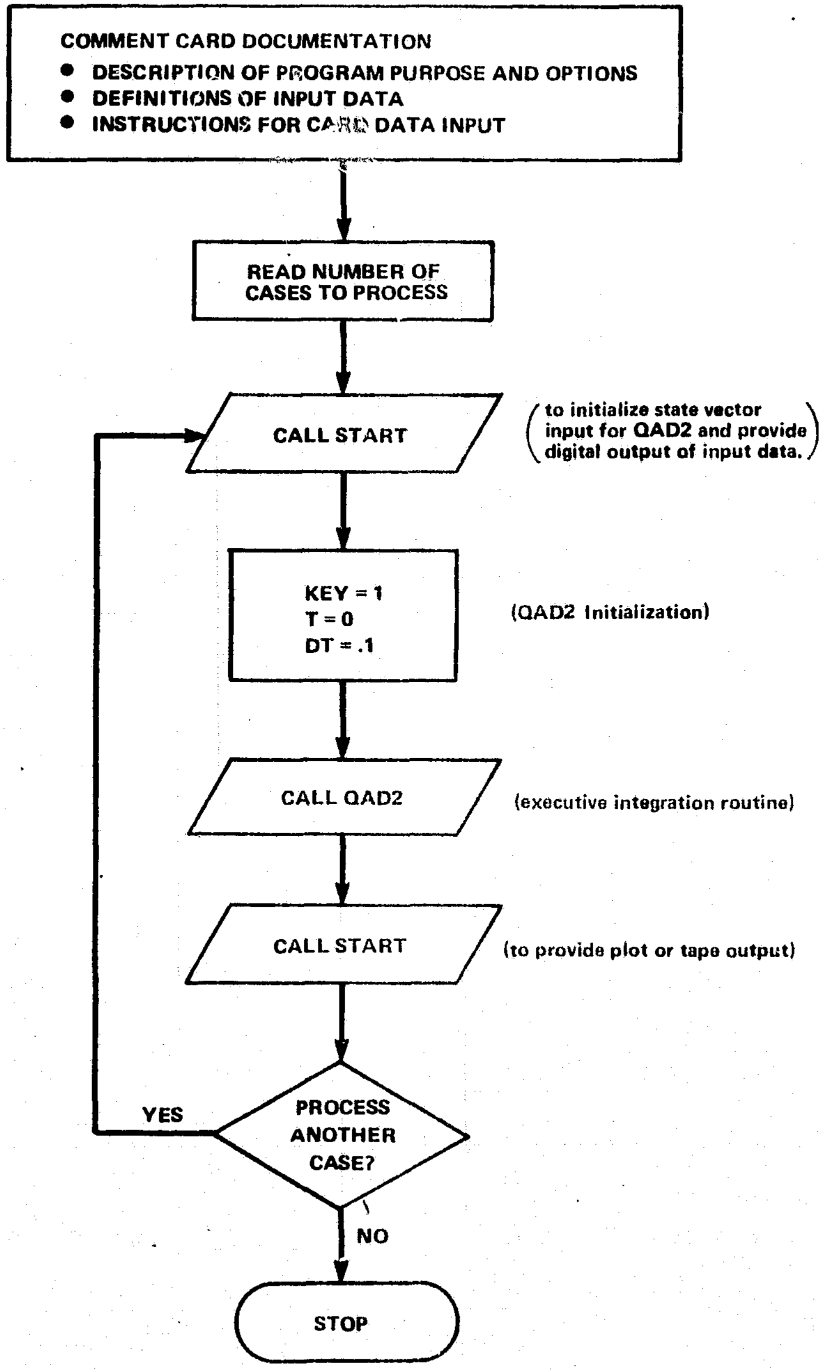
PROGRAM STRUCTURE/FUNCTIONAL FLOW CHART

2BODY SIMULATION

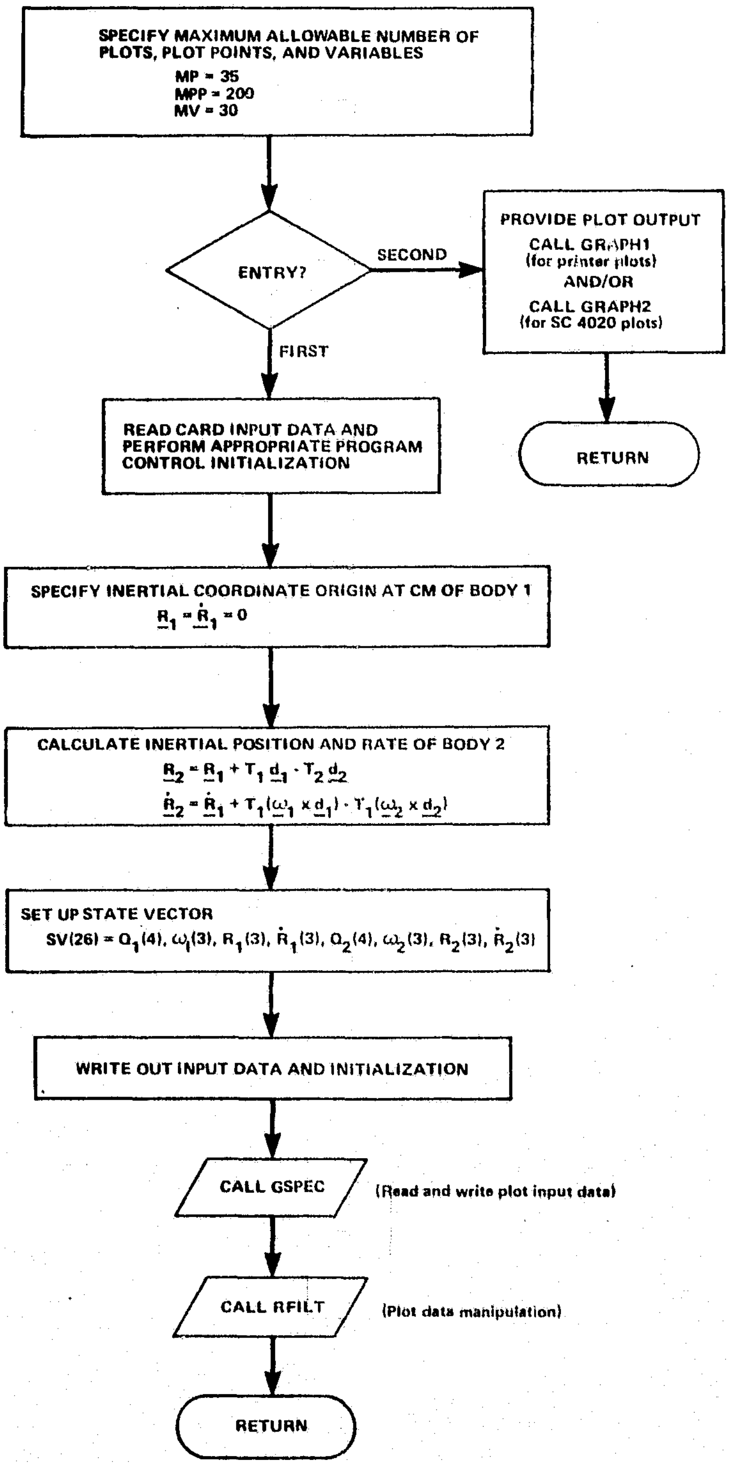


2BODY

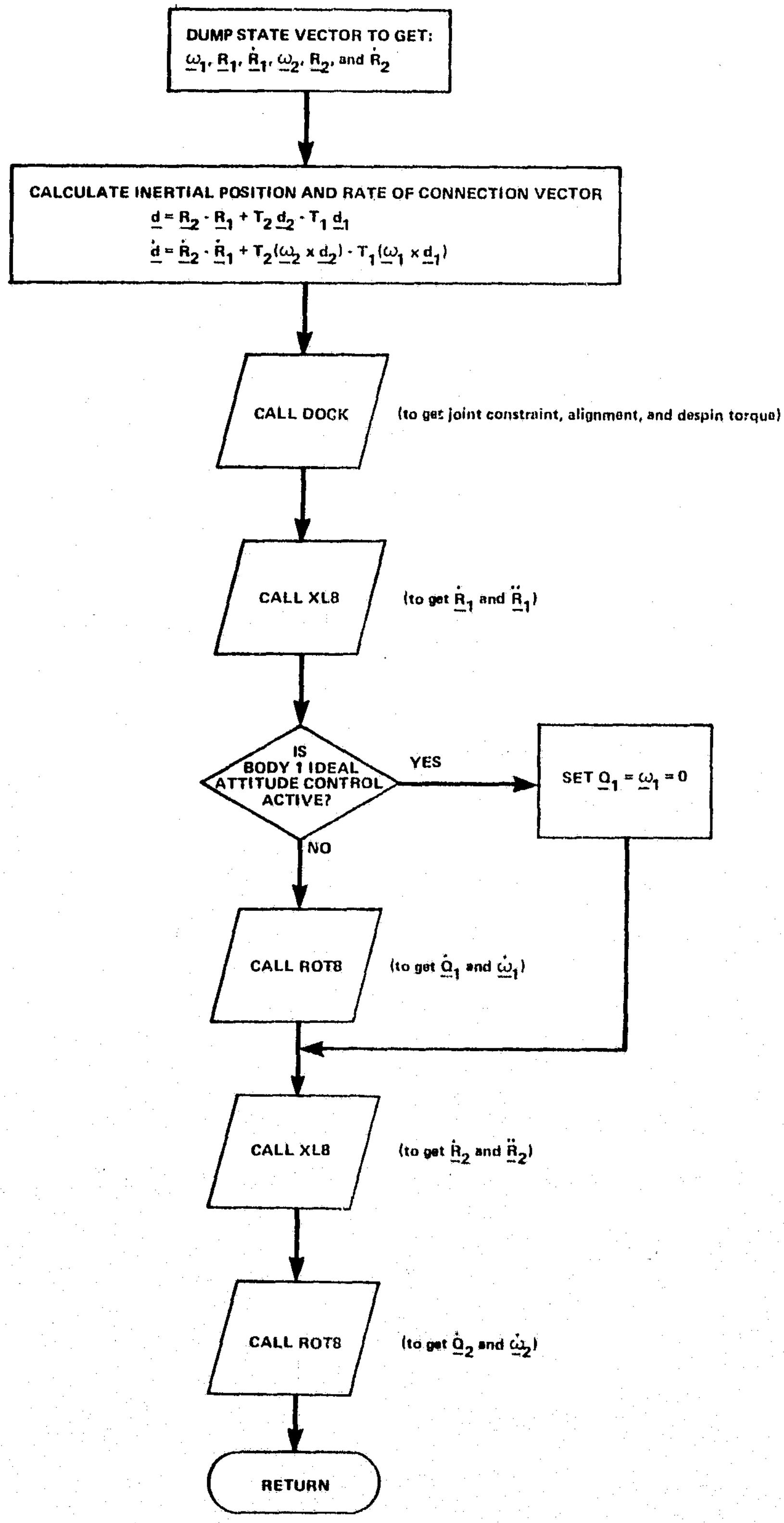
(MAIN PROGRAM)



START
(PROGRAM INITIALIZATION)

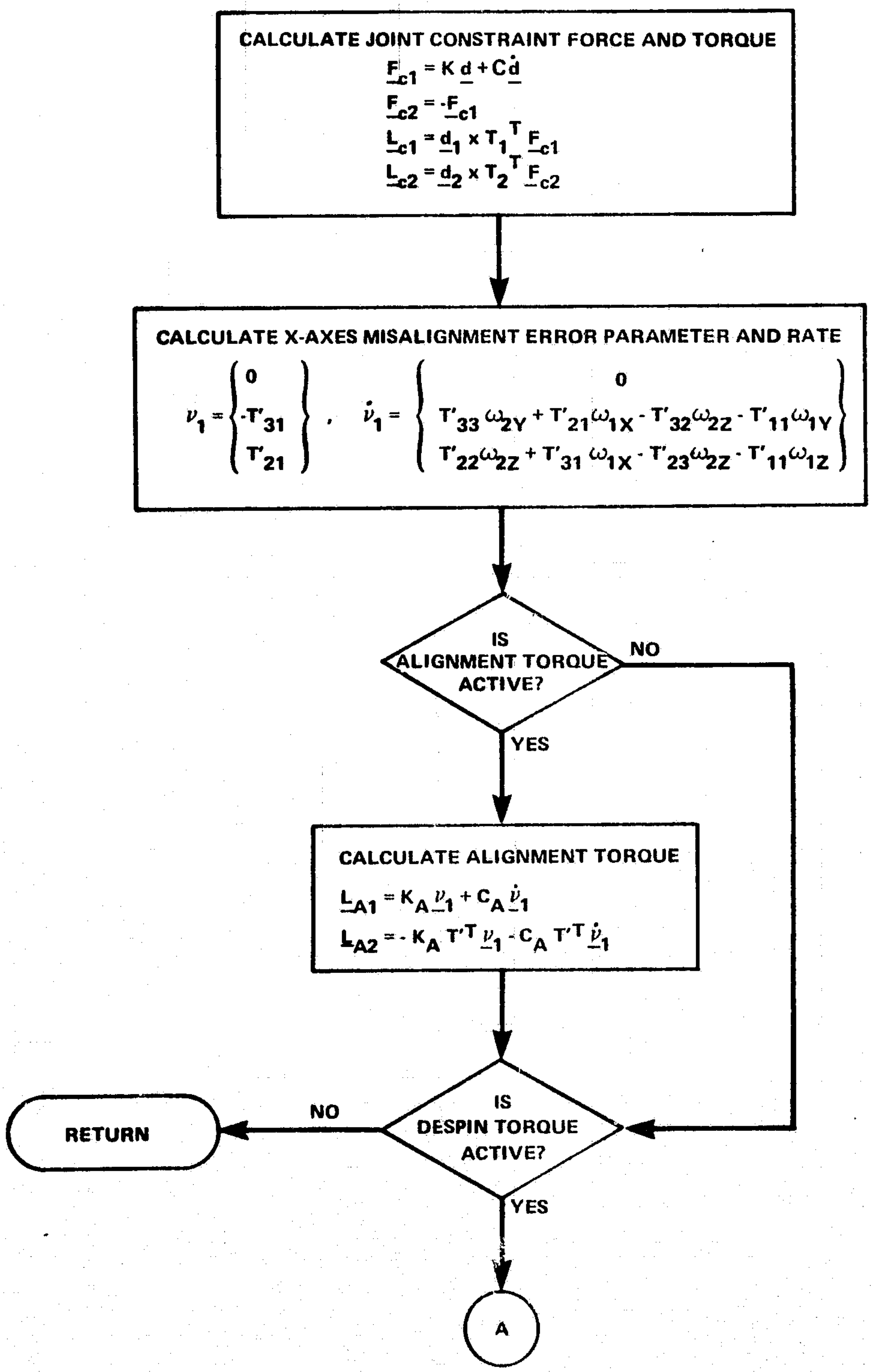


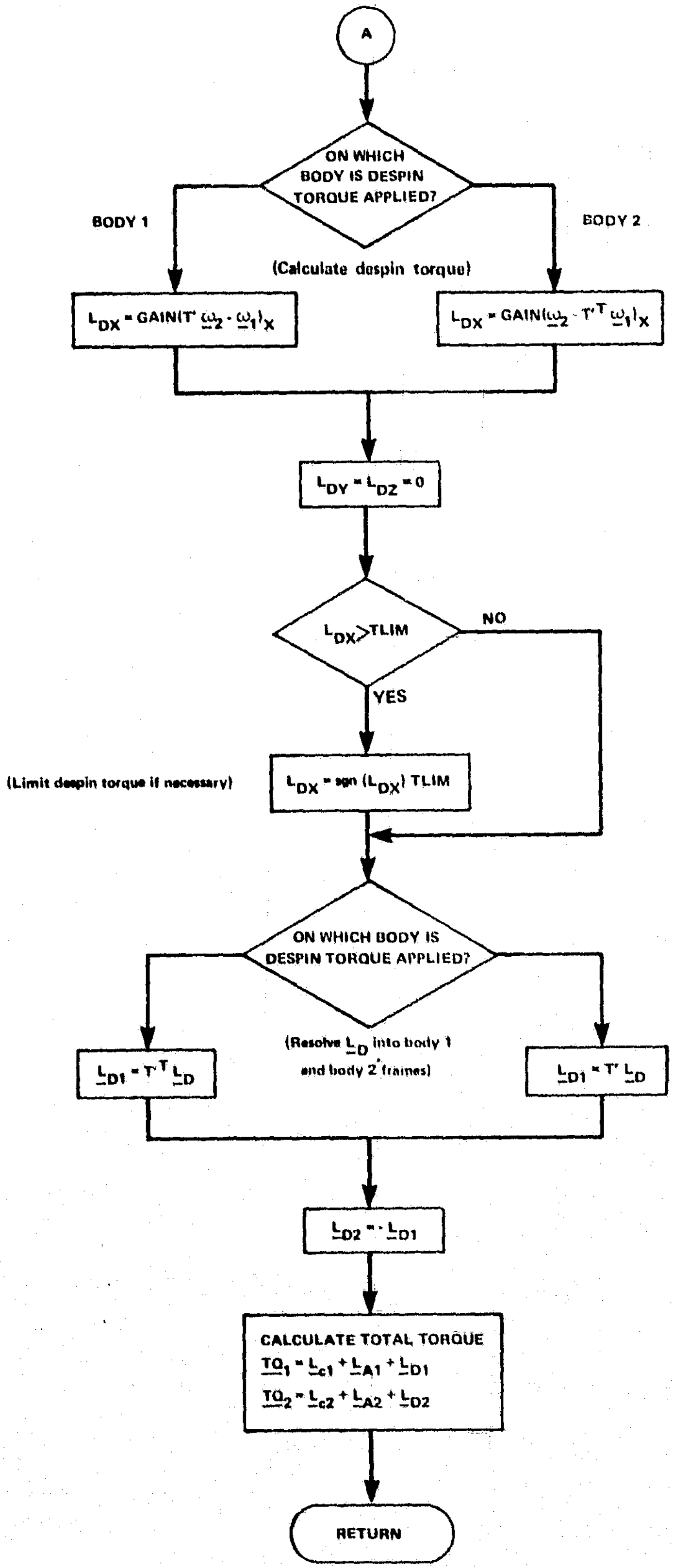
VDYN
(VEHICLE DYNAMICS)



DOCK

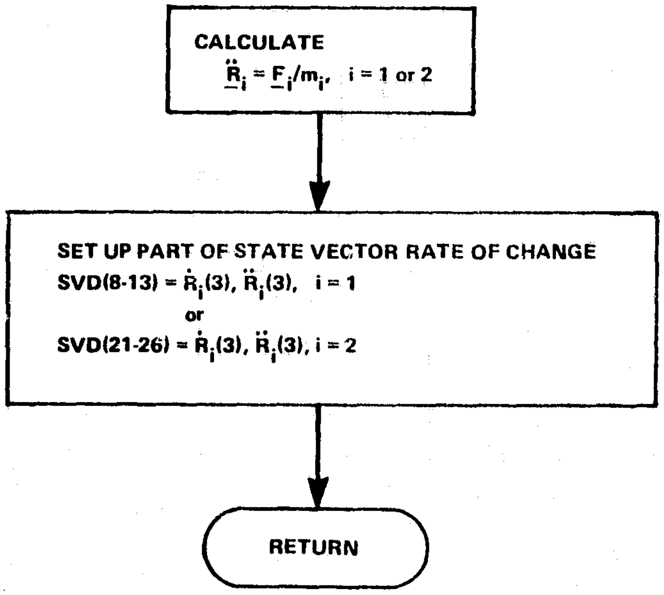
(JOINT DYNAMICS)




A
ON WHICH BODY IS DESPIN TORQUE APPLIED?
BODY 1
BODY 2
(Calculate despin torque)
L_DX = GAIN(T' ω_2 - ω_1)_X
L_DX = GAIN(ω_2 - T'^T ω_1)_X
L_DY = L_DZ = 0
L_DX>TLIM
NO
YES
(Limit despin torque if necessary)
L_DX = sgn (L_DX) TLIM
ON WHICH BODY IS DESPIN TORQUE APPLIED?
(Resolve L_D into body 1 and body 2 frames)
L_D1 = T'^T L_D
L_D1 = T' L_D
L_D2 = - L_D1
CALCULATE TOTAL TORQUE
TQ_1 = L_c1 + L_A1 + L_D1
TQ_2 = L_c2 + L_A2 + L_D2
RETURN

XL8

(TRANSLATIONAL EQUATIONS)



ROT8

(ROTATIONAL EQUATIONS)

CALCULATE

$$\dot{Q}_i = \tfrac{1}{2}\,\tilde{\tilde{Q}}_i\,\omega_i, \quad i = 1 \text{ or } 2$$

ie,

$$\begin{Bmatrix} \dot{Q}_i(1) \\ \dot{Q}_i(2) \\ \dot{Q}_i(3) \end{Bmatrix} = \tfrac{1}{2} \begin{bmatrix} Q_i(4) & -Q_i(3) & Q_i(2) \\ Q_i(3) & Q_i(4) & -Q_i(1) \\ -Q_i(2) & Q_i(1) & Q_i(4) \end{bmatrix} \begin{Bmatrix} \omega_i(1) \\ \omega_i(2) \\ \omega_i(3) \end{Bmatrix}$$

$$\dot{\omega}_i = I_i^{-1}(TQ_i - \tilde{\omega}_i\, I_i\, \omega_i)$$

ie,

$$\begin{Bmatrix} \dot{\omega}_i(1) \\ \dot{\omega}_i(2) \\ \dot{\omega}_i(3) \end{Bmatrix} = \begin{Bmatrix} [TQ_i(1) + (I_i(2) - I_i(3))\,\omega_i(2)\,\omega_i(3)]\,/I_i(1) \\ [TQ_i(2) + (I_i(3) - I_i(1))\,\omega_i(3)\,\omega_i(1)]\,/I_i(2) \\ [TQ_i(3) + (I_i(1) - I_i(2))\,\omega_i(1)\,\omega_i(2)]\,/I_i(3) \end{Bmatrix}$$

↓

SET UP PART OF THE STATE VECTOR RATE OF CHANGE

$SVD(1\text{-}7) = \dot{Q}_i(4), \dot{\omega}_i(3), \quad i = 1$

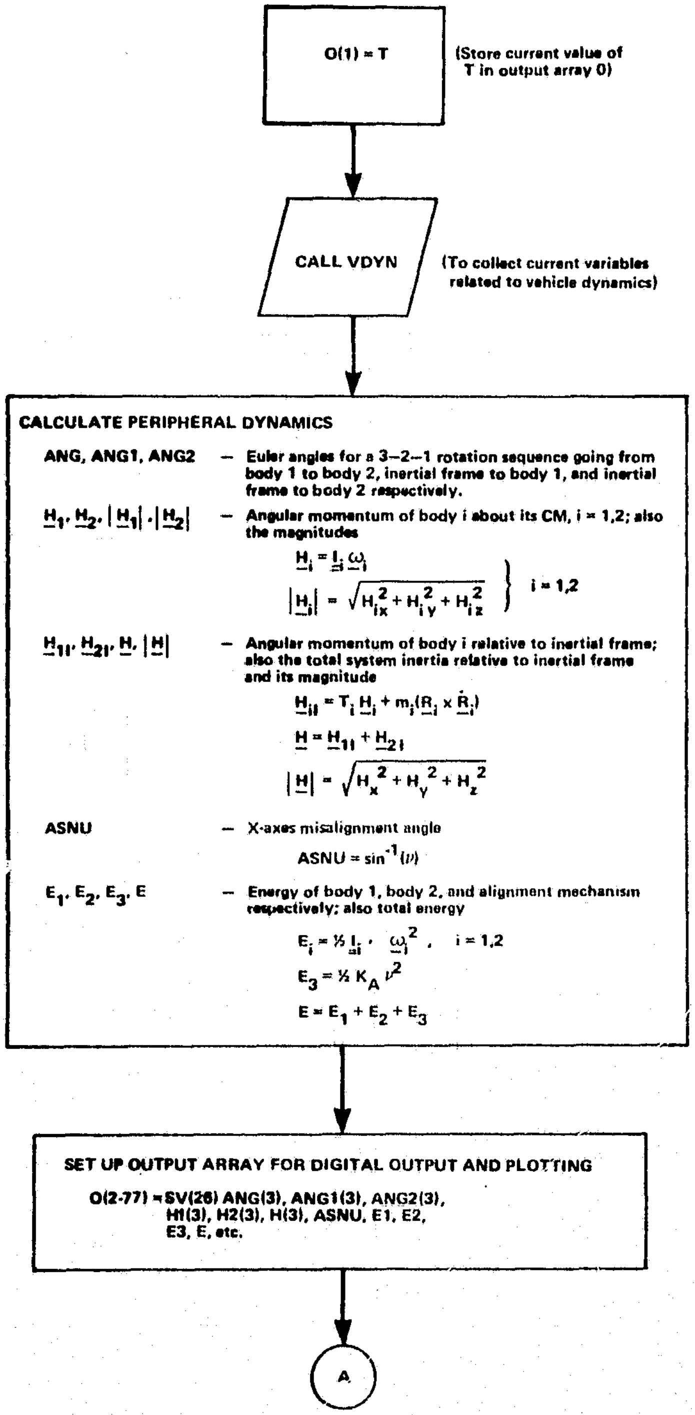
or

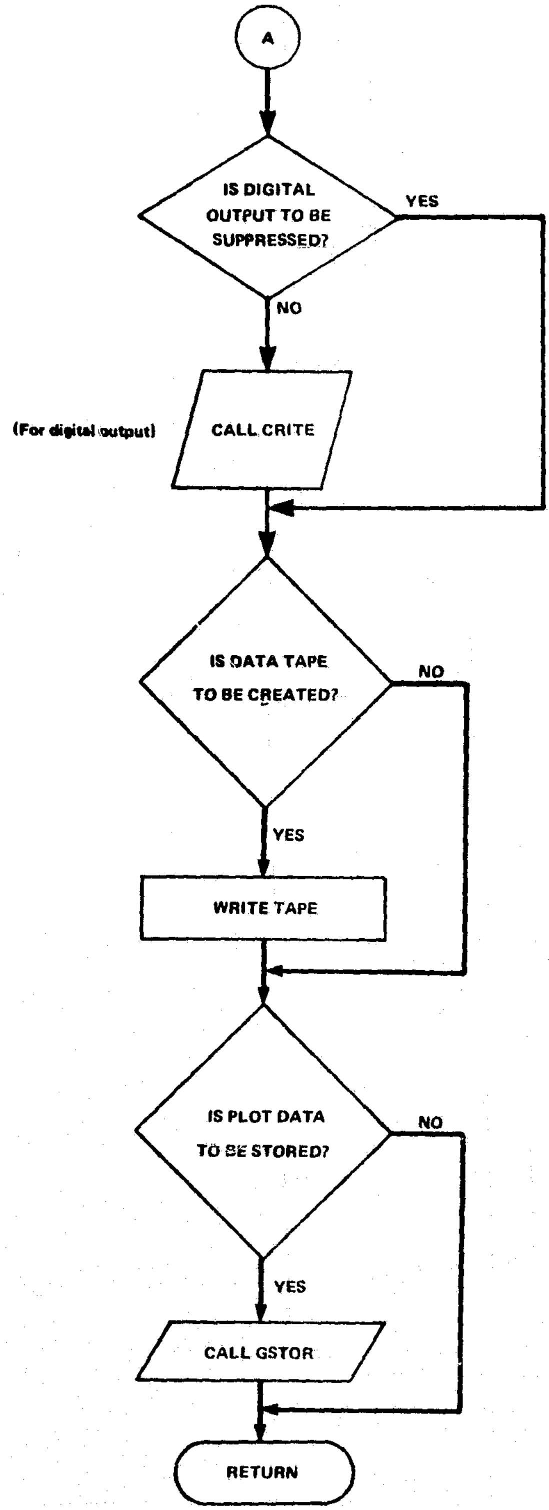
$SVD(14\text{-}20) = \dot{Q}_i(4), \dot{\omega}_i(3), \quad i = 2$

↓

RETURN

OUT
(PERIPHERAL DYNAMICS/DIGITAL OUTPUT)




A
IS DIGITAL OUTPUT TO BE SUPPRESSED?
YES
NO
(For digital output)
CALL CRITE
IS DATA TAPE TO BE CREATED?
NO
YES
WRITE TAPE
IS PLOT DATA TO BE STORED?
NO
YES
CALL GSTOR
RETURN

Appendix C

PROGRAM LISTING

```
C
C          ***   2 - B O D Y   S I M U L A T I O N   ***
C
C
C
C     PROGRAM .... SIMULATES THE TRANSLATIONAL AND ROTATIONAL MOTION
C                  OF TWO CONNECTED RIGID BODIES
C     OPTIONS .... 1) IDEAL ATTITUDE CONTROL OF BODY 1
C                  2) JOINT ALIGNMENT TORQUE
C                  3) DESPIN TORQUE (FOR ONE BODY SPINNING)
C                  4) OUTPUT BY DIGITAL PRINTOUT OR TAPE
C                  5) GRAPHICAL DISPLAY BY ON-LINE PRINTER OR SC 4020
C                     PLOTS
C
C                          ** DEFINITIONS **
C
C     CASES ...... NUMBER OF CASES
C     M1,M2 ...... MASS OF BODY 1,2
C     I1,I2 ...... PRINCIPAL MOMENTS OF INERTIA OF BODY 1,2
C     ANG1,ANG2 .. ATTITUDE OF BODY 1,2
C     W1,W2 ...... ANGULAR VELOCITY OF BODY 1,2
C     D1,D2 ...... HINGE OR JOINT POSITION VECTOR IN BODY 1,2
C     ICTRL ...... =0,1 -- BODY 1 IDEAL ATTITUDE CONTROL INACTIVE,ACTIVE
C     IALIGN ..... =0,1 -- X-AXIS ALIGNMENT TORQUE INACTIVE, ACTIVE
C     IDSPIN ..... =0 -- DESPIN TORQUE INACTIVE
C                  =1 -- DESPIN TORQUE ACTIVE AT TIME=0
C                  =2 -- DESPIN TORQUE ACTIVATED ON ENERGY CONDITION
C     IDTQRF ..... =1,2 -- DESPIN TORQUE REFERENCE FRAME IN BODY 1,2
C     KA ......... ALIGNMENT TORQUE SPRING COEFFICIENT
C     CA ......... ALIGNMENT TORQUE DAMPING COEFFICIENT
C     GAIN ....... DESPIN TORQUE GAIN
C     TLIM ....... DESPIN TORQUE LIMIT
C     TMAX ....... CUTOFF OR TERMINAL VALUE OF TIME
C     ISTEP ...... OUTPUT PRINT OCCURS EVERY 'ISTEP' INTEGRATION STEP
C     PTIME ...... OUTPUT PRINT OCCURS EACH 'PTIME' INTERVAL OF TIME
C     IPRNT ...... =0 -- NO DIGITAL PRINTOUT
C                  =1 -- DIGITAL PRINTOUT
C     ITAPE ...... =0 -- NO OUTPUT DATA TAPE
C                  =1 -- SINGLE PRECISION OUTPUT DATA TAPE
C                  =2 -- DOUBLE PRECISION OUTPUT DATA TAPE
C     IPLOT ...... =0 -- NO PLOT OUTPUT
C                  =1 -- PRINTER PLOTS
C                  =2 -- SC 4020 PLOTS
C                  =3 -- BOTH PRINTER AND SC 4020 PLOTS
C     O .......... DOUBLE PRECISION OUTPUT DATA ARRAY
C     OP ......... SINGLE PRECISION OUTPUT DATA ARRAY
C     MP ......... DIMENSION FOR MAXIMUM NUMBER OF PLOTS
C     MPP ........ DIMENSION FOR MAXIMUM NUMBER OF PLOT POINTS
C     MV ......... DIMENSION FOR MAXIMUM NUMBER OF VARIABLES
C         *** MP,MPP,MV ARE SET BY PARAMETER STATEMENTS
C                         IN BOTH SUBROUTINE START AND OUT ***
```

```
C
C
C                           ** CARD DATA INPUT **
C
C     DATA CARD                   PARAMETERS                      FORMAT
C
C         1                          CASES
C         2             M1,I1(3),ANG1(3),W1(3),D1(3)            'CREAD'
C         3             M2,I2(3),ANG2(3),W2(3),D2(3)        (REAL OR INTEGER
C         4              ICTRL,IALIGN,IDSPIN,IDTQRF          DATA SEPARATED
C         5                          KA,CA                   BY COMMA'S)
C         6                        GAIN,TLIM
C         7                    TMAX,ISTEP,PTIME
C         8                    IPRNT,ITAPE,IPLOT
C      9,11,...                    OP( ),OP( )                   2I5
C     10,12,...                    'PLOT TITLE'                  18A4
C
C     *** FOR EACH CASE, REPEAT CARDS 2 THRU 8 PLUS 9,10,11,12,... ***
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION SV(156)
      COMMON /NTGRAT/ TMAX,ISTEP,PTIME
      EXTERNAL VDYN,STEP,OUT,TERM
      DATA ICASE /0/
      INTGR(X)=X+0.5
C
C *** READ NUMBER OF CASES
C
      CALL CREAD (CASES,1)
      NOC=INTGR(CASES)
C
    1 K=1
      CALL START (SV,K)
C
C *** QAD2 INITIALIZATION
C
      KEY=1
      T=0.D0
      DT=.1D0
C
      CALL QAD2(T,SV,26,KEY,DT,TMAX,ISTEP,PTIME,VDYN,STEP,OUT,TERM)
C         T    - INDEPENDENT VARIABLE
C         DT   - INITIAL INTEGRATION STEP SIZE (NOT CRITICAL)
C         SV   - STATE VECTOR ARRAY DIMENSIONED 6N
C                THE FIRST N ELEMENTS CONTAIN THE INDEPENDENT VARIABLES
C
      K=2
      CALL START (SV,K)
      ICASE=ICASE+1
      IF (ICASE.LT.NOC) GO TO 1
      END
```

```
      SUBROUTINE START (SV,K)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KA
      DOUBLE PRECISION M1,I1,M2,I2
      PARAMETER MP=35
      PARAMETER MPP=200
      PARAMETER MV=30
      DIMENSION ANG1(3),W1(3),ANG2(3),W2(3)
      DIMENSION D1I(3),D2I(3)
      DIMENSION DYNOPT(4)
      DIMENSION OUTPUT(3)
      DIMENSION PFLAG(2)
      DIMENSION R1(3),R1D(3),R2(3),R2D(3)
      DIMENSION SV(26)
      DIMENSION T1(3,3),T2(3,3)
      DIMENSION TIME(3)
      DIMENSION VEH1(13),VEH2(13)
      DIMENSION W1XD(3),W1XDI(3),W2XD(3),W2XDI(3)
      DIMENSION IGVAR(MP,2),GMULT(2,MP),GSCAL(4,MP),ISQUA(MP),ROWS(MP),
     *          COLS(MP),GNAME(18,MP),ISCAL(MP),ISORT(MP),XY(MPP,2)
      COMMON /ALGNTQ/ KA,CA
      COMMON /DSPNTQ/ GAIN,TLIM
      COMMON /D/ D1(3),D2(3)
      COMMON /FLAGS/ ICTRL,IALIGN,ITQON,IDTQRF
      COMMON /NTGRAT/ TMAX,ISTEP,PTIME
      COMMON /OUTPUT/ IPRNT,ITAPE,IPLOT
      COMMON /PLOT/ NPP,INUSE(MV),GDATA(MPP,MV)
      COMMON /VEH/ M1,I1(3),M2,I2(3)
      DATA NR,NW /5,6/
      DATA IT /0/
      DATA PFLAG /' NO    ',' YES   '/
      DATA IBLANK,IHIT,IDASH,II /1H ,1H*,1H-,1HI/
      INTGR(X)=X+0.5
      GO TO (101,102),K
C
C *** READ INPUT DATA
C
  101 CALL CREAD (VEH1,13)
      CALL CREAD (VEH2,13)
      M1=VEH1(1)
      M2=VEH2(1)
      DO 1 I=1,3
      I1(I)=VEH1(I+1)
      I2(I)=VEH2(I+1)
      ANG1(I)=VEH1(I+4)
      ANG2(I)=VEH2(I+4)
      W1(I)=VEH1(I+7)
      W2(I)=VEH2(I+7)
      D1(I)=VEH1(I+10)
    1 D2(I)=VEH2(I+10)
      CALL CREAD (DYNOPT,4)
      ICTRL = INTGR(DYNOPT(1))
```

```
      IALIGN = INTGR(DYNOPT(2))
      IDSPIN = INTGR(DYNOPT(3))
      IDTQRF = INTGR(DYNOPT(4))
      CALL CREAD (KA,2)
      CALL CREAD (GAIN,2)
      CALL CREAD (TIME,3)
      TMAX=TIME(1)
      ISTEP=INTGR(TIME(2))
      PTIME=TIME(3)
      CALL CREAD (OUTPUT,3)
      IPRNT = INTGR(OUTPUT(1))
      ITAPE = INTGR(OUTPUT(2))
      IPLOT = INTGR(OUTPUT(3))
C
C *** INITIALIZATION
C
      NPP=0
      IF (ICTRL.EQ.0) GO TO 3
      DO 2 I=1,3
    2 W1(I)=0.D0
      PCTRL=PFLAG(2)
      GO TO 4
    3 PCTRL=PFLAG(1)
    4 IF (IALIGN.EQ.0) GO TO 5
      PALIGN=PFLAG(2)
      GO TO 6
    5 PALIGN=PFLAG(1)
    6 ITQON=IDSPIN+1
      IPLOT=IPLOT+1
      ITAPE=ITAPE+1
      IF (ITAPE.EQ.1) GO TO 7
      IF (IT.EQ.1) GO TO 7
      REWIND 8
      IT=1
C
C *** INERTIAL COORDINATE ORIGIN LOCATED AT CM OF BODY 1
C
    7 DO 8 I=1,3
      R1(I)=0.D0
    8 R1D(I)=0.D0
C
C *** CALCULATE INERTIAL POSITION (R2) AND RATE (R2D) OF BODY 2
C
      CALL ULR1 (ANG1,T1,3,2,1)
      CALL ULR1 (ANG2,T2,3,2,1)
      CALL MPRD (T1,D1,D1I,3,3,1)
      CALL MPRD (T2,D2,D2I,3,3,1)
      CALL CROSS (W1,D1,W1XD)
      CALL CROSS (W2,D2,W2XD)
      CALL MPRD (T1,W1XD,W1XDI,3,3,1)
      CALL MPRD (T2,W2XD,W2XDI,3,3,1)
```

```
      DO 9 I=1,3
      R2(I)=R1(I)+D1I(I)-D2I(I)
    9 R2D(I)=R1D(I)+W1XDI(I)-W2XDI(I)
C
C *** SET UP STATE VECTOR,  SV(26) = (Q1(4),W1(3),R1(3),R1D(3),Q2(4),....)
C
      CALL ULR6 (T1,SV)
      CALL ULR6 (T2,SV(14))
      DO 10 I=1,3
      SV(I+4 )=W1(I)
      SV(I+7 )=R1(I)
      SV(I+10)=R1D(I)
      SV(I+17)=W2(I)
      SV(I+20)=R2(I)
   10 SV(I+23)=R2D(I)
C
C *** WRITE INPUT DATA AND INITIAL CONDITIONS
C
      WRITE(NW,200)
  200 FORMAT(1H1,24X,'***  TWO-BODY SIMULATION  ***',///
     A           /,14X,'BODY 1  (TUG)',26X,'BODY 2  (TARGET)',/
     B           /,36X,'UNITS',/)
      WRITE(NW,201) M1,M2,(I1(I),I2(I),I=1,3)
  201 FORMAT(1H ,13X,'M1     ',F8.1,7X,'  KG   ',11X,'M2     ',F8.1,/
     C           /,14X,'I1X    ',F8.1,7X,'       ',11X,'I2X    ',F8.1,
     D           /,14X,'I1Y    ',F8.1,7X,' KG-M2 ',11X,'I2Y    ',F8.1,
     E           /,14X,'I1Z    ',F8.1,7X,'       ',11X,'I2Z    ',F8.1,/)
      WRITE(NW,202) (ANG1(I),ANG2(I),I=1,3),(W1(I),W2(I),I=1,3)
  202 FORMAT(1H ,13X,'ANG1X ',F8.3,7X,'       ',11X,'ANG2X ',F8.3,
     F           /,14X,'ANG1Y ',F8.3,7X,'  RAD  ',11X,'ANG2Y ',F8.3,
     G           /,14X,'ANG1Z ',F8.3,7X,'       ',11X,'ANG2Z ',F8.3,/
     H           /,14X,'W1X    ',F8.3,7X,'       ',11X,'W2X    ',F8.3,
     I           /,14X,'W1Y    ',F8.3,7X,'RAD/SEC',11X,'W2Y    ',F8.3,
     J           /,14X,'W1Z    ',F8.3,7X,'       ',11X,'W2Z    ',F8.3,/)
      WRITE(NW,203) (D1(I),D2(I),I=1,3)
  203 FORMAT(1H ,13X,'D1X    ',F8.3,7X,'       ',11X,'D2X    ',F8.3,
     K           /,14X,'D1Y    ',F8.3,7X,'   M   ',11X,'D2Y    ',F8.3,
     L           /,14X,'D1Z    ',F8.3,7X,'       ',11X,'D2Z    ',F8.3,/)
      WRITE(NW,204) (R1(I),R2(I),I=1,3),(R1D(I),R2D(I),I=1,3)
  204 FORMAT(1H ,13X,'R1X    ',F8.3,7X,'       ',11X,'R2X    ',F8.3,
     M           /,14X,'R1Y    ',F8.3,7X,'   M   ',11X,'R2Y    ',F8.3,
     N           /,14X,'R1Z    ',F8.3,7X,'       ',11X,'R2Z    ',F8.3,/
     O           /,14X,'R1DX   ',F8.3,7X,'       ',11X,'R2DX   ',F8.3,
     P           /,14X,'R1DY   ',F8.3,7X,'M/SEC  ',11X,'R2DY   ',F8.3,
     Q           /,14X,'R1DZ   ',F8.3,7X,'       ',11X,'R2DZ   ',F8.3,/)
      WRITE(NW,205) PCTRL
  205 FORMAT(1H0,//,19X,'***  IDEAL CONTROL ON BODY 1 ...',A6,/)
      IF (IALIGN) 20,103,104
  103 WRITE(NW,215) PALIGN
  215 FORMAT(1H ,   18X,'***  ALIGNMENT TORQUE ..........',A6,/)
      GO TO 105
```

```
104 WRITE(NW,225) PALIGN,KA,CA
225 FORMAT(1H ,   18X,'***  ALIGNMENT TORQUE ..........',A6
   R                 ,'WITH KA =',F6.0,' AND CA =',F7.0,/)
105 GO TO (106,107,108),ITQON
106 WRITE(NW,206)
206 FORMAT(1H ,   18X,'***  DESPIN TORQUE ............. NO',//)
    GO TO 11
107 WRITE(NW,207) IDTQRF,GAIN,TLIM
207 FORMAT(1H ,   18X,'***  DESPIN TORQUE ............. ACTIVE AT T=0'
   S                 ,' IN BODY',I2
   T                 ,' WITH GAIN =',F8.0,' AND TLIM =',F4.0,//)
    GO TO 11
108 WRITE(NW,208) IDTQRF,GAIN,TLIM
208 FORMAT(1H ,   18X,'***  DESPIN TORQUE ............. ACTIVATED ON '
   U                 ,'ENERGY CONDITION IN BODY',I2
   V                 ,' WITH GAIN =',F8.0,' AND TLIM =',F4.0,//)
 11 WRITE(NW,213) TMAX,ISTEP,PTIME
213 FORMAT(1H0,   18X,'***  CUTOFF TIME .....',F5.1,' SECONDS',/
   W           /,19X,'***  OUTPUT PRINT TO OCCUR EVERY',I5,
   X       ' INTEGRATION STEPS AND AT EACH',F5.1,' SECOND INTERVAL',//)
    GO TO (12,12,109,109),IPLOT
109 WRITE(NW,209)
209 FORMAT(1H0,   18X,'***  SC 4020 PLOTS REQUESTED')
 12 GO TO (15,13,14),ITAPE
 13 WRITE(NW,210)
210 FORMAT(1H0,   18X,'***  SINGLE PRECISION DATA TAPE REQUESTED')
    GO TO 15
 14 WRITE(NW,211)
211 FORMAT(1H0,   18X,'***  DOUBLE PRECISION DATA TAPE REQUESTED')
 15 IF(IPRNT.EQ.1) GO TO 16
    WRITE(NW,212)
212 FORMAT(1H0,   18X,'***  DIGITAL PRINTOUT IS SUPPRESSED')
 16 IF (IPLOT.EQ.1) GO TO 20
C
C *** READ AND WRITE PLOT INPUT DATA AND INITIALIZATION
C
    CALL GSPEC (NR,NW,IGVAR,GMULT,GSCAL,ISQUA,ROWS,COLS,GNAME,
   *            ISCAL,ISORT,MP,NC)
    CALL RFILT (NC,IGVAR,MP,INUSE,MV)
    GO TO 20
102 GO TO (19,17,18,17),IPLOT
C     (FOR PRINTER PLOTS)
 17 CALL GRAPH1 (NC,NPP,IGVAR,GDATA,GMULT,XY,MPP,MP,MV,
   *              NW,GNAME,ISORT,ISCAL,GSCAL,ISQUA,ROWS,COLS,
   *              IBLANK,IHIT,IDASH,II)
    GO TO (19,19,18,18),IPLOT
C     (FOR SC 4020 PLOTS)
 18 CALL GRAPH2 (NC,NPP,IGVAR,GDATA,GMULT,XY,MPP,MP,MV,
   *              GNAME,ISCAL,GSCAL,IHIT)
 19 IF (ITAPE.EQ.1) GO TO 20
    END FILE 8
 20 RETURN
    END
```

```
      SUBROUTINE OUT(T,SV,J)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KA
      DOUBLE PRECISION M1,I1,M2,I2
      DOUBLE PRECISION NU1,NUSQ,NUD1,NUDSQ
      REAL OP
      PARAMETER IH=19
      PARAMETER IO=77
      PARAMETER MP=35
      PARAMETER MPP=200
      PARAMETER MV=30
      DIMENSION ANG(3),ANG1(3),ANG2(3)
      DIMENSION H1(3),H1I(3),H2(3),H2I(3)
      DIMENSION H(3),HT1(3),HT2(3)
      DIMENSION NA(IH),KHDR(IH)
      DIMENSION O(IO),OP(IO)
      DIMENSION R1XD(3),R2XD(3)
      DIMENSION SV(26) ,SVD(26)
      COMMON /ALGNTQ/ KA,CA
      COMMON /VEH/ M1,I1(3),M2,I2(3)
      COMMON /VDYN1/ W1(3),W2(3),T1(3,3),T2(3,3),D(3),DD(3)
      COMMON /VDYN2/ R1(3),R1D(3),R2(3),R2D(3)
      COMMON /DOCK1/ F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3)
      COMMON /DOCK2/ NU1(3),NUD1(3),TP(3,3)
      COMMON /FLAGS/ ICTRL,IALIGN,ITQON,IDTQRF
      COMMON /OUTPUT/ IPRNT,ITAPE,IPLOT
      COMMON /PLOT/ NPP,INUSE(MV),GDATA(MPP,MV)
      DATA DPR/57.2957795131D0/
      DATA EPS /9.D3/
      DATA NA / 1,-4, 3, 3,-6,-4, 3, 3,-6, 4,-6,-6,-6, 4, 4, 4, 4,-2,-4/
      DATA KHDR /' TIME ',' Q1  ',' ANG1 ','  W1  ','R1/R1D',' Q2  ',
     *           ' ANG2 ','  W2  ','R2/R2D','ANG/NU',' D/DD ','F1/F2 ',
     *           'TQ1TQ2','H1 TOT','H2 TOT','H TOT ','F123T ',' CONE ',
     *           'X12YZI'/
C
C *** PAGING LOGIC FOR CRITE
C
      GO TO (1,2,2,2),J
    1 IENT=1
      GO TO 3
    2 IENT=2
C
    3 O(1)=T
C
      CALL VDYN(T,SV,SVD)
```

```
C *** VDYN IN CONJUNCTION WITH DOCK, XL8, AND ROT8 HANDLES VEH. DYNAMICS
C *** VDYN RETURNS SVD ARRAY FOR Q1D,Q2D,W1D,W2D,R1D,R1DD,R2D,R2DD
C
      CALL ULR2 (TP,ANG,3,2,1)
      CALL ULR2 (T1,ANG1,3,2,1)
      CALL ULR2 (T2,ANG2,3,2,1)
      X1=DACOS(DCOS(ANG1(1))*DCOS(ANG1(2)))
      X2=DACOS(DCOS(ANG2(1))*DCOS(ANG2(2)))
      DEL1=-DATAN2(DTAN(ANG1(2)),DSIN(ANG1(1)))
      DEL2=-DATAN2(DTAN(ANG2(2)),DSIN(ANG2(1)))
      X1YI=X1*DCOS(DEL1)
      X1ZI=X1*DSIN(DEL1)
      X2YI=X2*DCOS(DEL2)
      X2ZI=X2*DSIN(DEL2)
C
C *** CALCULATE BODY 1,2 AND TOTAL ANGULAR MOMENTUM AND CONE ANGLES
C
      DO 4 I=1,3
      H1(I)=I1(I)*W1(I)
    4 H2(I)=I2(I)*W2(I)
      H1MAG=DSQRT(H1(1)*H1(1)+H1(2)*H1(2)+H1(3)*H1(3))
      H2MAG=DSQRT(H2(1)*H2(1)+H2(2)*H2(2)+H2(3)*H2(3))
      CALL MPRD (T1,H1,H1I,3,3,1)
      CALL MPRD (T2,H2,H2I,3,3,1)
      CALL CROSS (R1,R1D,R1XD)
      CALL CROSS (R2,R2D,R2XD)
      DO 6 I=1,3
      H1I(I)=H1I(I)+M1*R1XD(I)
      H2I(I)=H2I(I)+M2*R2XD(I)
    6 H(I)=H1I(I)+H2I(I)
      HMAG=DSQRT(H(1)*H(1)+H(2)*H(2)+H(3)*H(3))
      IF (DABS(HMAG)) 8,8,7
    7 CALL TPRD (T1,H,HT1,3,3,1)
      CALL TPRD (T2,H,HT2,3,3,1)
      CONE1=DASIN(DSQRT(HT1(2)*HT1(2)+HT1(3)*HT1(3))/HMAG)
      CONE2=DASIN(DSQRT(HT2(2)*HT2(2)+HT2(3)*HT2(3))/HMAG)
C
C***  CALCULATION OF X-AXES MISALIGNMENT
C
    8 NUSQ=NU1(2)*NU1(2)+NU1(3)*NU1(3)
      NUDSQ=NUD1(2)*NUD1(2)+NUD1(3)*NUD1(3)
      ASNU=DASIN(DSQRT(NUSQ))
C
C *** CALCULATE HINGE,RIGID BODY, AND TOTAL ENERGY
C
      E1=.5D0*(I1(1)*W1(1)**2 + I1(2)*W1(2)**2 + I1(3)*W1(3)**2)
      E2=.5D0*(I2(1)*W2(1)**2 + I2(2)*W2(2)**2 + I2(3)*W2(3)**2)
      E=E1+E2
      IF (IALIGN.EQ.0) GO TO 9
      E3=.5D0*KA*NUSQ
      E=E+E3
```

```
C *** IS DESPIN TORQUE ON / ENERGY TEST
C
    9 GO TO (11,11,10),ITQON
   10 IF (E.GT.EPS) GO TO 11
      ITQON=2
C
C *** SET UP OUTPUT ARRAY FOR CRITE AND PLOTTING
C         NOTE THAT O(1)=T AND O(I+1)=SV(I) FOR I=1,26
C
   11 DO 12 I=1,4
      O(I+1)=SV(I)                                                Q1
   12 O(I+17)=SV(I+13)                                            Q2
      DO 13 I=1,3
      O(I+5)=ANG1(I)*DPR
      O(I+8)=SV(I+4)                                              W1
      O(I+11)=SV(I+7)                                             R1
      O(I+14)=SV(I+10)                                            R1D
      O(I+21)=ANG2(I)*DPR
      O(I+24)=SV(I+17)                                            W2
      O(I+27)=SV(I+20)                                            R2
      O(I+30)=SV(I+23)                                            R2D
      O(I+33)=ANG(I)*DPR
      O(I+37)=D(I)
      O(I+40)=DD(I)
      O(I+43)=F1(I)
      O(I+46)=F2(I)
      O(I+49)=TQ1(I)
      O(I+52)=TQ2(I)
      O(I+55)=H1(I)
      O(I+59)=H2(I)
      O(I+63)=H(I)
   13 CONTINUE
      O(37)=ASNU*DPR
      O(59)=H1MAG
      O(63)=H2MAG
      O(67)=HMAG
      O(68)=E1
      O(69)=E2
      O(70)=E3
      O(71)=E
      O(72)=CONE1*DPR
      O(73)=CONE2*DPR
      O(74)=X1YI*DPR
      O(75)=X1ZI*DPR
      O(76)=X2YI*DPR
      O(77)=X2ZI*DPR
```

```
      IF (IPRNT.EQ.0) GO TO 14
      CALL CRITE (O,KHDR,NA,IH,1,IENT)
   14 GO TO (18,15,17),ITAPE
C
C *** SINGLE PRECISION OUTPUT ARRAY (O) FOR PLOT ARRAY (OP)
C
   15 DO 16 I=1,IO
   16 OP(I)=O(I)
      WRITE(8) OP(1)
      GO TO 18
   17 WRITE (8) T
   18 IF (IPLOT.EQ.1) GO TO 21
      IF (NPP.GT.MPP) GO TO 21
      IF (ITAPE.EQ.2) GO TO 20
      DO 19 I=1,IO
   19 OP(I)=O(I)
   20 NPP=NPP+1
      CALL GSTOR (NPP,INUSE,OP,IO,GDATA,MPP,MV)
   21 RETURN
      END
```

```
      SUBROUTINE VDYN(T,SV,SVD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M1,I1,M2,I2
      DIMENSION SV(26),SVD(26)
      DIMENSION D1I(3),D2I(3)
      DIMENSION W1XD(3),W1XDI(3),W2XD(3),W2XDI(3)
      COMMON /VDYN1/ W1(3),W2(3),T1(3,3),T2(3,3),D(3),DD(3)
      COMMON /VDYN2/ R1(3),R1D(3),R2(3),R2D(3)
      COMMON /VEH/ M1,I1(3),M2,I2(3)
      COMMON /D/ D1(3),D2(3)
      COMMON /DOCK1/ F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3)
      COMMON /FLAGS/ ICTRL,IALIGN,ITQON,IDTQRF
      DO 1 I=1,3
      W1(I)=SV(I+4)
      R1(I)=SV(I+7)
      R1D(I)=SV(I+10)
      W2(I)=SV(I+17)
      R2(I)=SV(I+20)
    1 R2D(I)=SV(I+23)
C
C *** CALCULATE INERTIAL POSITION AND RATE OF CONNECTION VECTOR D
C
      CALL ULR5 (SV(1),T1)
      CALL ULR5 (SV(14),T2)
      CALL MPRD (T1,D1,D1I,3,3,1)
      CALL MPRD (T2,D2,D2I,3,3,1)
      CALL CROSS (W1,D1,W1XD)
      CALL CROSS (W2,D2,W2XD)
      CALL MPRD (T1,W1XD,W1XDI,3,3,1)
      CALL MPRD (T2,W2XD,W2XDI,3,3,1)
      DO 2 I=1,3
      D(I)=R2(I)-R1(I)+D2I(I)-D1I(I)
    2 DD(I)=R2D(I)-R1D(I)+W2XDI(I)-W1XDI(I)
C
C *** DOCK PROVIDES JOINT CONSTRAINT, ALIGNMENT, AND DESPIN TORQUES
C
      CALL DOCK
C
C *** XL8 SETS UP SVD ARRAY FOR R1D,R1DD,R2D,R2DD
C *** ROT8 SETS UP SVD ARRAY FOR Q1D,Q2D,W1D,W2D
C
      CALL XL8 (SV(11),M1,F1I,SVD(8))
      IF (ICTRL.EQ.0) GO TO 4
      DO 3 I=1,7
    3 SVD(I)=0.D0
      GO TO 5
    4 CALL ROT8(SV(1),I1,TQ1,SVD(1))
    5 CALL XL8 (SV(24),M2,F2I,SVD(21))
      CALL ROT8(SV(14),I2,TQ2,SVD(14))
      RETURN
      END
```

```
      SUBROUTINE DOCK
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRFCISION K,KA
      DOUBLE PRFCISION NU1,NU2,NUD1,NUD2
      DIMENSION DTQ(3)
      DIMENSION NU2(3),NUD2(3)
      DIMENSION TEMP(3)
      COMMON /ALGNTQ/ KA,CA
      COMMON /DSPNTQ/ GAIN,TLIM
      COMMON /D/ D1(3),D2(3)
      COMMON /DOCK1/ F1(3),F1I(3),TQ1(3),F2(3),F2I(3),TQ2(3)
      COMMON /DOCK2/ NU1(3),NUD1(3),TP(3,3)
      COMMON /FLAGS/ ICTRL,IALIGN,ITQON,IDTQRF
      COMMON /VDYN1/ W1(3),W2(3),T1(3,3),T2(3,3),D(3),DD(3)
      DATA K,C /4.12D3,2.91D2/
C
C *** CALCULATE JOINT CONSTRAINT FORCE AND TORQUE
C
      DO 1 I=1,3
      F1I(I)=K*D(I)+C*DD(I)
    1 F2I(I)=-F1I(I)
      CALL TPRD (T1,F1I,F1,3,3,1)
      CALL TPRD (T2,F2I,F2,3,3,1)
      CALL CROSS (D1,F1,TQ1)
      CALL CROSS (D2,F2,TQ2)
C
C *** CALCULATE X-AXES MISALIGNMENT ERROR PARAMETER AND RATE
C
      CALL TPRD (T1,T2,TP,3,3,3)
      NU1(1)=0.D0
      NU1(2)=-TP(3,1)
      NU1(3)=TP(2,1)
      NUD1(1)=0.D0
      NUD1(2)=TP(3,3)*W2(2)+TP(2,1)*W1(1)-TP(3,2)*W2(3)-TP(1,1)*W1(2)
      NUD1(3)=TP(2,2)*W2(3)+TP(3,1)*W1(1)-TP(2,3)*W2(2)-TP(1,1)*W1(3)
C
      IF (IALIGN.EQ.0) GO TO 3
C
C *** CALCULATE ALIGNMENT TORQUE
C
      CALL TPRD (TP,NU1,NU2,3,3,1)
      CALL TPRD (TP,NUD1,NUD2,3,3,1)
      DO 2 I=1,3
      TQ1(I)=TQ1(I)+KA*NU1(I)+CA*NUD1(I)
    2 TQ2(I)=TQ2(I)-KA*NU2(I)-CA*NUD2(I)
```

```
    3 GO TO (11,4,11),ITQON
C
C *** CALCULATE DESPIN TORQUE IN BODY 1 OR 2 REFERENCE FRAME
C
    4 IF (IDTQRF.EQ.1) GO TO 5
      CALL TPRD (TP,W1,TEMP,3,3,1)
      CALL SUBV (W2,TEMP,TEMP)
      GO TO 6
    5 CALL MPRD (TP,W2,TEMP,3,3,1)
      CALL SUBV (TEMP,W1,TEMP)
    6 DTQ(1)=GAIN*TEMP(1)
      DTQ(2)=0.D0
      DTQ(3)=0.D0
      IF (DABS(DTQ(1)).GT.TLIM)  DTQ(1) = DSIGN(TLIM,DTQ(1))
      GO TO (7,9),IDTQRF
    7 CALL TPRD (TP,DTQ,TEMP,3,3,1)
      DO 8 I=1,3
      TQ1(I)=TQ1(I)+DTQ(I)
    8 TQ2(I)=TQ2(I)-TEMP(I)
      GO TO 11
    9 CALL MPRD (TP,DTQ,TEMP,3,3,1)
      DO 10 I=1,3
      TQ1(I)=TQ1(I)+TEMP(I)
   10 TQ2(I)=TQ2(I)-DTQ(I)
   11 RETURN
      END
```

```
      SUBROUTINE XL8(SV,M,F,SVD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      DIMENSION SV(26),F(3),SVD(26)
      DO 1 I=1,3
C
C *** SET UP R1D OR R2D
C
      SVD(I)=SV(I)
C
C *** SET UP R1DD OR R2DD
C
1     SVD(I+3)=F(I)/M
      RETURN
      END
```

```
      SUBROUTINE ROT8(SV,I,TQ,SVD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION I(3)
      DIMENSION SV(26),TQ(3),SVD(26)
C
C *** SET UP Q1D OR Q2D
C
      SVD(1)= .5D0*( SV(4)*SV(5)-SV(3)*SV(6)+SV(2)*SV(7))
      SVD(2)= .5D0*( SV(3)*SV(5)+SV(4)*SV(6)-SV(1)*SV(7))
      SVD(3)= .5D0*(-SV(2)*SV(5)+SV(1)*SV(6)+SV(4)*SV(7))
      SVD(4)=-.5D0*( SV(1)*SV(5)+SV(2)*SV(6)+SV(3)*SV(7))
C
C *** SET UP W1D OR W2D
C
      SVD(5)=(TQ(1)+(I(2)-I(3))*SV(6)*SV(7))/I(1)
      SVD(6)=(TQ(2)+(I(3)-I(1))*SV(7)*SV(5))/I(2)
      SVD(7)=(TQ(3)+(I(1)-I(2))*SV(5)*SV(6))/I(3)
      RETURN
      END
```

```
      SUBROUTINE  QAD2(X,Y,N,KEY,HZ,XM,NP,XP,DERIV,AUTO,OUT,TERM)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION Y(1)
      DIMENSION C1(4),C2(3),C3(5)
      EQUIVALENCE        (C1(1),THIRD),(C1(2),SIXTH),(C1(4),HALF)
      EQUIVALENCE (C2(2),THREE),(C3(1),TWO),(C3(2),ZERO)
      EQUIVALENCE (C2(1),ONE)
      DATA C1,C2,C3,THIRTY,RATIO
     X   /.33333333333333333D0,.16666666666666667D0,.125D0,.5D0,1.D0,3.D0,
     X   4.D0,2.D0,0.D0,-9.D0,8.D0,-1.D0,30.D0,1.D-4/
      DATA IOUT/6/
      XMOD(X,Y)=CMOD(X,Y)
      SSIGN(X,Y)=DSIGN(X,Y)
      AABS(X)=DABS(X)
      GO TO (1,2),KEY
1     N2=  N
      N3=2*N
      N4=3*N
      N5=4*N
      N6=5*N
      XLAST=X-XMOD(X,SSIGN(XP,XM-X))
      H1=SSIGN(HZ,XM-X)
      CALL OUT(X,Y,1)
2     KSRCH=0
      IND=0
      KNTRL=-1
      LOOK=0
10    KSKIP=1
      CALL TERM(X,Y,G1,LOOK)
20    H=SSIGN(H1,XM-X)
40    XMAX=XLAST+SSIGN(XP,H )
      DH=XM-XLAST
      IF(AABS(DH/XP)-ONE)30,30,50
30    XMAX=XM
50    DO 390 K=1,10
      DO 290 J=1,NP
      IF(KSRCH)70,55,70
55    DH=XMAX-X
      IF(AABS(DH/H)-ONE)60,60,70
60    H1=H
      H=DH
      IND=1
70    DO 160 M=1,10
      DO 90 I=1,N
      IX =N3+I
      Y(IX )=ZERO
      IX5=N5+I
```

```
80      Y(IX5)=Y(I)
        DO 115 KPASS=1,5
        GO TO(81,82,83,81,85),KPASS
81      DH=X
        IF(KPASS-1)85,83,85
82      DH=X+THIRD*H
        GO TO 83
85      DH=DH+HALF*H
83      CALL DERIV(DH,Y(N5+1),Y(N6+1))
        DO 115 I=1,N
        IX =N6+I
        FI=H*Y(IX )
        GO TO (91,92,93,94,94),KPASS
91      IX=N2+I
        GO TO 95
93      IX=N4+I
95      Y(IX)=FI
94      IX =N3+I
        Y(IX )=Y(IX )+C3(KPASS)*FI
92      IX =N2+I
        XNEW=Y(IX )
        GO TO (101,102,102,104,105),KPASS
104     IX5=N4+I
        XNEW=XNEW-THREE*Y(IX5)
102     XNEW=XNEW+C2(KPASS-1)*FI
101     IX5=N5+I
        Y(IX5)=Y(I)+C1(KPASS)*XNEW
        IF(KPASS-4)115,110,105
110     Y(IX )=Y(IX )+THREE*FI
105     Y(IX )=Y(IX )+FI
115     CONTINUE
        DO 120 I=1,N
        IX =N2+I
        Y(IX )=SIXTH*Y(IX )
        IX =N3+I
120     Y(IX )=AABS(Y(IX )/THIRTY)
        IF(KNTRL)140,170,140
140     KNTRL=0
        CALL AUTO(X,Y,Y(N3+1 ),KNTRL)
        IF(KNTRL-2)170,150,170
150     H=HALF*H
160     IND=0
        KPASS=2
        WRITE(IOUT,1000)
1000     FORMAT(21H1STEP HALVED 10 TIMES)
```

```
      KEY=3
      GO TO 325
170   X=X+H
      DO 180 I=1,N
      IX =N2+I
180   Y(I)=Y(I)+Y(IX )
      IF(LOOK)190,225,190
190   CALL TERM(X,Y,G ,LOOK)
      IF(KSKIP)196,191,196
191   IF(KSRCH)300,195,300
195   IF(G*G1)200,400,220
196   KSKIP=0
220   G1=G
225   IF(IND)230,260,230
230   IF(XMAX-XM)241,250,241
241   XLAST=XMAX
242   CALL OUT(X,Y,2)
      IND=0
      GO TO 20
250   KPASS=3
      KEY=1
      GO TO 325
260   IF(KNTRL)270,290,270
270   IF(KNTRL-3)290,280,290
280   H=TWO*H
290   CONTINUE
      CALL OUT(X,Y,2)
      GO TO 50
200   EPS=RATIO*AABS(H)
      KSRCH=1
      IND=0
      GO TO 370
300   IF(IND)320,370,320
370   H=G*H/(G1-G)
      G1=G
      IF(AABS(H)-EPS)380,380,390
400   H=ZERO
      KSRCH=1
380   IND=1
390   CONTINUE
      KPASS=2
      WRITE(IOUT,2000)
2000  FORMAT(15H1CAN'T CONVERGE )
      KEY=4
      GO TO 325
320   KPASS=4
      KEY=2
325   CALL OUT(X,Y,KPASS)
      RETURN
      END
```

```
      SUBROUTINE STEP(T,SV,E,KNTRL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION SV(26),E(26)
      DATA EPS/1.D-7/
      C1=4.D2
      EQ1=VMAGN(E( 1),4)
      EQ2=VMAGN(E(14),4)
      EW1=VMAGN(E( 5),3)
      EW2=VMAGN(E(18),3)
      E1 =VMAGN(E( 8),3)
      E2 =VMAGN(E(21),3)
      ED1=VMAGN(E(11),3)
      ED2=VMAGN(E(24),3)
      DEL=DSQRT((E1*E1+E2*E2)/C1+ED1*ED1+ED2*ED2+EW1*EW1+EW2*EW2+EQ1*EQ1
     1+EQ2*EQ2)
      KNTRL=1
      IF(DEL-EPS)1,1,2
    1 IF(64.D0*DEL-EPS)3,4,4
    2 KNTRL=2
      GO TO 4
    3 KNTRL=3
    4 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TERM(T,SV,G,K)
      DIMENSION SV(26)
      G=1.
      RETURN
      END
```

```
      SUBROUTINE CREAD(X,NWORD)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION K(80),X(1)
      DATA IBLNK,IDOT,IZERO,NINE,IPLUS,MINUS,IE,ICOM
     X    /1H ,1H.,1H0,1H9,1H+,1H-,1HE,1H,/
      DATA IN,IOUT
     X    /5,6/
      DATA KSHFT,KADD
     X    /1073741824,15/
      DATA ZERO,ONE,HALF,TIN
     X /0.D0,1.D0,.5D0,10.D0/
      SSIGN(X,Y)=DSIGN(X,Y)
      I=80
      DO 33 JJ=1,NWORD
      IF(I-80) 2,1,1
    1 READ(IN,100)K
  100 FORMAT(80A1)
      I=0
    2 KHALF=0
      N=0
      KE=0
    3 I=I+1
      FLOT=ZERO
      L1=0
      LNEG=0
      LDOT=0
      DO 19 INDEX=I,80
      KI=K(INDEX)
      IF(KI-IBLNK)4,19,4
    4 KCON=KI/KSHFT
      KCON=KCON+KADD
      IF(KCON)8,6,5
5     IF(KCON-9)6,6,8
    6 L1=1
      XCON=KCON
      FLOT=TIN*FLOT+XCON
      IF(LDOT)7,19,7
    7 N=N-1
      GO TO 19
```

```
    8 IF(KI-IPLUS)9,19,9
    9 IF(KI-MINUS)11,10,11
   10 LNEG=1
      GO TO 19
   11 IF(KHALF)18,12,18
   12 IF(KI-IDOT)14,13,14
   13 LDOT=1
      GO TO 19
   14 IF(KI-IE)18,15,18
15    IF(INDEX-80)16,23,23
   16 KE=1
      GO TO 20
   18 IF(KI-ICOM)34,20,34
   19 CONTINUE
      INDEX=81
20    I=INDEX
      IF(LNEG)21,22,21
   21 FLOT=-FLOT
   22 IF(KHALF)27,23,27
   23 Z=FLOT
      IF(KE)24,26,24
   24 KHALF=1
      IF(L1)3,25,3
   25 Z=ONE
      GO TO 3
   26 IF(L1)28,33,28
   27 NFLOT=FLOT+SSIGN(HALF,FLOT)
      N=N+NFLOT
   28 TEN=TIN
      IF(N)29,32,30
   29 TEN=ONE/TEN
      N=-N
   30 DO 31 J=1,N
   31 Z=Z*TEN
   32 X(JJ)=Z
   33 CONTINUE
      RETURN
   34 WRITE(IOUT,200)K
200   FORMAT(12H1INPUT ERROR//1H0,80A1)
      CALL EXIT
      END
```

```
      SUBROUTINE CRITE(A,KHDR,NA,NVAR,NWORD,IENT)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(1),KHDR(1),NA(1)
      DIMENSION K(121)
      DATA LHDR
     X  / 6  /
      DATA KDOT,KZERO,KBLNK,KSIGN,KE,MINUS
     X    /1H.,1H0,1H ,1H ,1HE,1H-/
      DATA IOUT
     X    /6/
      DATA KSHFT ,KADD
     X    /1073741824,-15/
      DATA ONE,HALF,TEN,TENTH,ROUND,EPSLN
     X /1.D0,.5D0,1.D1,.1D0,.5D-5,1.D-18 /
      GO TO (1,2),IENT
1     IENT=2
      NWMAX=12/LHDR
      IF(NWORD-NWMAX)111,112,112
  111 NWMAX=NWORD
112   INDNT=(12-LHDR*NWMAX)/2
      NPAGE=0
      MAX=0
11    NPAGE=NPAGE+1
      WRITE(IOUT,200)NPAGE
200   FORMAT(1H1,110X,4HPAGE,I4/)
      LINE=3
      GO TO 3
2     IF(LINE-58+MAX)3,11,11
3     MSAVE=1
4     DO 92 M=MSAVE,1000
      DO 5 I=1,121
5     K(I)=KBLNK
      LOC=M
      LVOID=1
      KOL=1
      DO 6 J=1,NVAR
      NJ=NA(J)
      IF(NJ)6,6,7
7     IF(NJ-M)90,8,8
8     LVOID=0
      I1=12*KOL-10
```

```
      IF(LINE-3)9,88,9
9     X=A(LOC)
      N=0
      IF(X-EPSLN) 12, 20, 20
   12 IF(X+EPSLN) 15, 10, 10
10    K(I1+5)=KZERO
      GO TO 90
15    X=-X
      KSIGN=MINUS
20    IF(X-(ONE-ROUND))30,25,25
25    X=X/TEN
      N=N+1
      GO TO 20
30    IF(X-TENTH)35,40,40
35    X=TEN*X
      N=N-1
      GO TO 30
40    X=X+ROUND
      NX=TEN*X
      IF(NX-9)41,41,71
71    X=X/TEN
      N=N+1
41    IF(N)50,55,45
45    IF(N-5)55,55,50
50    LDOT=I1+2
      K(I1+7)=KE
      N=N-1
      GO TO 60
55    I1=I1+2
      LDOT=I1+N+1
      N=0
60    K(LDOT)=KDOT
      I2=I1+6
65    K(I1)=KSIGN
      KSIGN=KBLNK
      I1=I1+1
```

```
      DO 75 I=I1,I2
      IF(I-LDOT)70,75,70
70    X=TEN*X
      NX=X
      KK=NX+KADD
      K(I)=KSHFT*KK
      XNX=NX
      X=X-XNX
75    CONTINUE
      IF(N)80,90,85
80    N=-N
      KSIGN=MINUS
85    I1=I1+7
      I2=I1+2
      X=N
      X=(X+HALF)*TENTH*TENTH
      N=0
      GO TO 65
88    I1=I1+INDNT-1
      DO 89 IWORD=1,NWMAX
      I2=(J-1)*NWORD+IWORD
      KK=KHDR(I2)
      DO 89 I=1,LHDR
      I2=I1+I+LHDR*(IWORD-1)
      K(I2)= KK
 89   KK=KK*64
90    KOL=KOL+1
      IF(KOL-10)6,6,91
6     LOC=LOC+IABS(NJ)
91    WRITE(IOUT,100)K
  100 FORMAT(121A1)
      LINE=LINE+1
      IF(LINE-5)93, 3,94
94    IF(LVOID)95,92,95
92    CONTINUE
93    MSAVE=999
      GO TO 4
95    MAX=M-1
      RETURN
      END
```

```
      SUBROUTINE GSPEC (NR,NW,IGVAR,GMULT,GSCAL,ISQUA,ROWS,COLS,GNAME,
     *                  ISCAL,ISORT,M,NG)
      DIMENSION IGVAR(M,2),GMULT(2,M),GSCAL(4,M),ISQUA(M),ROWS(M),
     *          COLS(M),GNAME(18,M),ISCAL(M),ISORT(M)
      NG= 0
      WRITE (NW,500)
      DO 12 I=1,M
      READ (NR,501) (IGVAR(I,J),J=1,2),(GMULT(J,I),J=1,2),(GSCAL(J,I),
     *              J=1,4),ISQUA(I),ROWS(I),COLS(I)
      IF (IGVAR(I,1)) 1,15,1
    1 WRITE (NW,601) (IGVAR(I,J),J=1,2),(GMULT(J,I),J=1,2),(GSCAL(J,I),
     *              J=1,4),ISQUA(I),ROWS(I),COLS(I)
      NG= NG+1
      READ (NR,502) (GNAME(J,I),J=1,18)
      DO 5 J=1,2
      IF (GMULT(J,I)) 3,2,3
    2 GMULT(J,I)= 1.
      GO TO 5
    3 IF (ABS(GMULT(J,I)-1.)-.00001) 4,4,5
    4 GMULT(J,I)= 57.29578
    5 CONTINUE
      ISCAL(I)= 0
      DO 6 J=1,4
      IF (GSCAL(J,I)) 7,6,7
    6 CONTINUE
      ISCAL(I)= 1
    7 IF (ROWS(I)) 9,8,9
    8 ROWS(I)= 50.
    9 IF (COLS(I)) 11,10,11
   10 COLS(I)= 100.
   11 WRITE (NW,601) (IGVAR(I,J),J=1,2),(GMULT(J,I),J=1,2),(GSCAL(J,I),
     *              J=1,4),ISQUA(I),ROWS(I),COLS(I)
      WRITE (NW,602) (GNAME(J,I),J=1,18)
      ISORT(I)= 0
      IF (ROWS(I)-50.) 12,12,111
  111 ISORT(I)= 1
   12 CONTINUE
   13 READ (NR,501) IDATA
      IF (IDATA) 14,15,14
   14 READ (NR,502) ADATA
      GO TO 13
   15 WRITE (NW,603)
      RETURN
  500 FORMAT (1H1,26X,'***  PLOT DATA  ***',///)
  501 FORMAT (2I5,6F10.0,I2,2F4.0)
  502 FORMAT (18A4)
  601 FORMAT (1H ,2I10,6F10.0,I2,2F5.0)
  602 FORMAT (1H ,18A4)
  603 FORMAT (1H1)
      END
```

```
      SUBROUTINE RFILT (NG,IGVAR,M,INUSE,N)
      DIMENSION IGVAR(M,2),INUSE(N)
   15 K= 1
      INUSE(1)= IGVAR(1,1)
      DO 16 I=2,N
   16 INUSE(I)= 0
      DO 21 J=1,2
      DO 21 I=1,NG
      L= 1
   17 IF (INUSE(L)-IGVAR(I,J)) 18,21,18
   18 L= L+1
      IF (INUSE(L)) 17,19,17
   19 K= K+1
      IF (K-N+1) 20,20,22
   20 INUSE(K)= IGVAR(I,J)
   21 IGVAR(I,J)= L
   22 RETURN
      END
```

```
      SUBROUTINE GSTOR (NPP,INUSE,C,L,GDATA,M,N)
      DIMENSION INUSE(N),C(L),GDATA(M,N)
      DO 1 I=1,N
      J= INUSE(I)
      IF (J) 1,2,1
    1 GDATA(NPP,I)= C(J)
    2 RETURN
      END
```

```
      SUBROUTINE GRAPH1 (NG,NPP,IGVAR,GDATA,GMULT,XY,L,M,N,
     *                   NW,GNAME,ISORT,ISCAL,GSCAL,ISQUA,ROWS,COLS,
     *                   IB,IH,ID,II)
      DIMENSION IGVAR(M,2),GMULT(2,M),GSCAL(4,M),ISQUA(M),ROWS(M),
     *          COLS(M),GNAME(18,M),ISCAL(M),ISORT(M),GDATA(L,N),XY(L,2)
      DO 4 I=1,NG
      IX= IGVAR(I,1)
      IY= IGVAR(I,2)
      IF (IX-N) 1,1,5
    1 IF (IY-N) 2,2,5
    2 DO 3 J=1,NPP
      XY(J,1)= GDATA(J,IX)*GMULT(1,I)
    3 XY(J,2)= GDATA(J,IY)*GMULT(2,I)
    4 CALL DPLOTM (XY(1,1),XY(1,2),NPP,GNAME(1,I),ISQUA(I),ISCAL(I),
     *             ISORT(I),GSCAL(1,I),ROWS(I),COLS(I),NW,IB,IH,ID,II)
    5 RETURN
      END
```

```
      SUBROUTINE PACK (DATA)
      DIMENSION IS(6),TEMP(12),DATA(18)
      DATA IS/0,6,12,18,24,30/
      I=1
      L=0
      DO 10 J=1,18
      DO 10 K=1,4
      L=L+1
      IF (L.LE.6) GO TO 10
      L=1
      I=I+1
   10 FLD(IS(L),6,TEMP(I))=FLD(IS(K),6,DATA(J))
      DO 11 J=1,12
   11 DATA(J)=TEMP(J)
      RETURN
      END
```

```
      SUBROUTINE GRAPH2 (NG,NPP,IGVAR,GDATA,GMULT,XY,L,M,N,
     *                   GNAME,ISCAL,GSCAL,IH)
      DIMENSION BLANK(12),OPDATA(22)
      DIMENSION IGVAR(M,2),GDATA(L,N),GMULT(2,M),XY(L,2),
     *          GNAME(18,M),ISCAL(M),GSCAL(4,M)
      DATA BLANK /12*6H      /
      DATA OPDATA /132H SC-4020 DATA
     *                  NINE INCH CAMERA          HARD COPY OUTPUT
     *                 /
      CALL IDENT (9,OPDATA)
      DO 5 I=1,NG
      IX= IGVAR(I,1)
      IY= IGVAR(I,2)
      IF (IX-N) 1,1,6
    1 IF (IY-N) 2,2,6
    2 XL= GDATA(1,IX)*GMULT(1,I)
      XR= XL
      YB= GDATA(1,IY)*GMULT(2,I)
      YT= YB
      DO 3 J=1,NPP
      XY(J,1)= GDATA(J,IX)*GMULT(1,I)
      XY(J,2)= GDATA(J,IY)*GMULT(2,I)
      XL= AMIN1 (XL,XY(J,1))
      XR= AMAX1 (XR,XY(J,1))
      YB= AMIN1 (YB,XY(J,2))
    3 YT= AMAX1 (YT,XY(J,2))
      IF (ABS(YT-YB).GE..00001) GO TO 31
      BIAS= (YT-YB)/2.
      YT= BIAS+.00001
      YB= BIAS-.00001
   31 DX= (XR-XL)/10.
      DY= (YT-YB)/10.
      CALL GRID1V (1,XL,XR,YB,YT,DX,DY,-0,-0,-1,-1,7,7)
      CALL PACK (GNAME(1,I))
      CALL PRINTV (72,GNAME(1,I),24,0)
      IF (ISCAL(I).EQ.1) GO TO 4
      CALL QUIK3V (0,IH,BLANK,BLANK,-NPP,XY(1,1),XY(1,2))
      GO TO 5
    4 CALL QUIK3L (0,GSCAL(1,I),GSCAL(2,I),GSCAL(3,I),GSCAL(4,I),
     *             IH,BLANK,BLANK,-NPP,XY(1,1),XY(1,2))
    5 CONTINUE
    6 CALL ENDJOB
      RETURN
      END
```

```
      SUBROUTINE DPLOTM (X,Y,NPP,TITLE,ISQUA,ISCAL,ISORT,SCALE,ROWS,
     X                  COLS,NRITE,BLANK,HIT,DASH,IALPH)
      REAL LINE
      INTEGER GRID(121),OUT(121)
      INTEGER BLANK,HIT,DASH
      DIMENSION X(1),Y(1),TITLE(18),SCALE(4),AXIS(12),DIGIT(2)
      IF (COLS-120.) 1,1,2
2     COLS=120.
1     WRITE (NRITE,601) TITLE
      JSORT= 1
      IF (ISORT) 202,201,202
 201  JSORT= -1
 202  CONTINUE
      DO 5 I=1,NPP
      DO 5 J=1,NPP
      IF ((Y(J)-Y(I))*FLOAT(JSORT)) 5,5,3
3     TEMP=Y(I)
      Y(I)=Y(J)
      Y(J)=TEMP
      TEMP=X(I)
      X(I)=X(J)
      X(J)=TEMP
5     CONTINUE
      IF (ISCAL) 6,13,6
6     SCALE(4)=Y(1)
      SCALE(3)=Y(NPP)
      SCALE(2)=X(1)
      SCALE(1)=X(1)
      DO 10 I=1,NPP
      IF (SCALE(1)-X(I)) 8,8,7
7     SCALE(1)=X(I)
8     IF (SCALE(2)-X(I)) 9,10,10
9     SCALE(2)=X(I)
10    CONTINUE
      XM=10.*(SCALE(2)-SCALE(1))/COLS
      YM=10.*(SCALE(4)-SCALE(3))/ROWS
      SCALE(1)=SCALE(1)-XM
      SCALE(2)=SCALE(2)+XM
      SCALE(3)=SCALE(3)-YM
      SCALE(4)=SCALE(4)+YM
13    IF (ISQUA) 14,15,14
14    SCALE(3)=SCALE(1)*(ROWS/COLS)*8.5/5.
      SCALE(4)=SCALE(2)*(ROWS/COLS)*8.5/5.
15    CONTINUE
```

```
      DO 152 I=1,3,2
      J= I/2+1
      DIGIT(J)= 1.
      IF (ABS(SCALE(I+1)-SCALE(I))-10.) 150,150,151
150   DIGIT(J)= .001
151   IP1=I+1
      DO 152 K=I,IP1
      ITEMP= SCALE(K)/DIGIT(J)
152   SCALE(K)= FLOAT(ITEMP)*DIGIT(J)
      DO 12 I=1,3,2
      IF (ABS(SCALE(I+1)-SCALE(I))-.00001) 11,11,12
11    TEMP= (SCALE(I+1)-SCALE(I))/2.
      SCALE(I)= TEMP+.0001*FLOAT(JSORT)
      SCALE(I+1)= TEMP-.0001*FLOAT(JSORT)
12    CONTINUE
      YSTEP= (SCALE(4)-SCALE(3))/ROWS
      XSTEP= (SCALE(2)-SCALE(1))/COLS
      YHALF= YSTEP/2.
      XHALF= XSTEP/2.
      NP=1
      ICOLS=COLS+1.001
      IROWS=ROWS+1.001
      DO 16 I=1,ICOLS
16    GRID(I)=DASH
      DO 17 I=1,ICOLS,10
17    GRID(I)=IALPH
      WRITE (NRITE,602) (GRID(I),I=1,ICOLS)
      DO 263 L=1,IROWS
      LINE= L-1
      DO 18 I=2,ICOLS
18    OUT(I)= BLANK
      OUT(1)= DASH
      OUT(ICOLS)= DASH
      YPOS= SCALE(4)-LINE*YSTEP
      TOP= YPOS + YHALF
      BOT= YPOS - YHALF
```

```
 181 IF (ISORT) 182,19,182
 182 IF (BOT-Y(NP)) 251,251,183
 183 IF (TOP-Y(NP)) 21,21,25
19   IF (Y(NP)-BOT) 251,20,20
20   IF (Y(NP)-TOP) 21,25,25
21   TEMP= COLS*(X(NP)-SCALE(1))/(SCALE(2)-SCALE(1))+1.
     ICOL=TEMP
     IF (TEMP-FLOAT(ICOL)-.5) 225,22,22
22   ICOL= ICOL+1
225  IF (ICOL) 25,25,23
23   IF (ICOL-121) 24,24,25
24   OUT(ICOL)= HIT
25   NP=NP+1
     IF (NP-NPP) 181,181,251
251  IF (DIGIT(2)-.999) 261,26,26
261  WRITE (NRITE,604) YPOS, (OUT(I),I=1,ICOLS)
     GO TO 263
26   WRITE (NRITE,603) YPOS, (OUT(I),I=1,ICOLS)
263  CONTINUE
262  NIS= COLS/10.+1.
     DO 27 I=1,NIS
27   AXIS(I)= SCALE(1)+(SCALE(2)-SCALE(1))*FLOAT(I-1)*10./COLS
     WRITE (NRITE,602) (GRID(I),I=1,ICOLS)
     IF (DIGIT(1)-1.) 29,28,28
28   WRITE (NRITE,605) (AXIS(I),I=1,NIS)
     GO TO 30
29   WRITE (NRITE,606) (AXIS(I),I=1,NIS)
30   RETURN
601  FORMAT (1H1,18A4/)
602  FORMAT (1H ,10X,121A1)
603  FORMAT (1H ,F9.2,1X,121A1)
 604 FORMAT (1H ,F9.4,1X,121A1)
605  FORMAT (1H ,3X,12F10.2)
606  FORMAT (1H ,3X,12F10.5)
     END
```

```
      DOUBLE PRECISION FUNCTION CMOD(X,Y)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      CMOD=X
      IF(Y)1,5,1
    1 Z=X/Y
      N=Z
      XN=N
      IF(Z-XN)2,4,2
    2 IF(X*Y)3,5,4
    3 XN=N-1
    4 CMOD=CMOD-XN*Y
    5 RETURN
      END
```

```
      SUBROUTINE CROSS(A,B,C)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(3),B(3),C(3)
      C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END
```

```
      SUBROUTINE MPRD(A,B,R,M,N,L)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(1),B(1),R(1)
      DATA ZERO
     X /0.D0/
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=ZERO
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
      SUBROUTINE SUBV(A,B,C)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(3),B(3),C(3)
      C(1)=A(1)-B(1)
      C(2)=A(2)-B(2)
      C(3)=A(3)-B(3)
      RETURN
      END
```

```
      SUBROUTINE TPRD(A,B,R,M,N,L)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(1),B(1),R(1)
      DATA ZERO
     X /0.D0/
      IR=0
      IK=-N
      DO 10 K=1,L
      IJ=0
      IK=IK+N
      DO 10 J=1,M
      IB=IK
      IR=IR+1
      R(IR)=ZERO
      DO 10 I=1,N
      IJ=IJ+1
      IB=IB+1
   10 R(IR)=R(IR)+A(IJ)*B(IB)
      RETURN
      END
```

```
      SUBROUTINF TWST(THETA,T,N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION T(3,3)
      DATA ZERO,ONE/0.D0,1.D0/
      SSIN(X)=DSIN(X)
      CCOS(X)=DCOS(X)
      CTHET=CCOS(THETA)
      DO 90 I=1,3
      DO 90 J=I,3
      IF(I-J)10,60,10
   10 IF((I-N)*(J-N))30,20,30
20    X=ZERO
      GO TO 50
   30 X=SSIN(THETA)
      IF(N-2)50,40,50
   40 X=-X
   50 T(I,J)=-X
      GO TO 90
   60 IF(I-N)70,80,70
   70 X=CTHET
      GO TO 90
   80 X=ONE
   90 T(J,I)=X
      RETURN
      END
```

```
      SUBROUTINE  ULR1(Q,T,L,M,N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION Q(3),T(3,3),A(3,3),B(3,3)
      CALL  TWST(Q(3),T,N)
      CALL  TWST(Q(2),A,M)
      CALL MPRD(A,T,B,3,3,3)
      CALL  TWST(Q(1),A,L)
      CALL MPRD(A,B,T,3,3,3)
      RETURN
      END
```

```
      SUBROUTINF ULR5(Z,T)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION Z(4),T(3,3)
      DATA TWO,HALF
     X  /2.D0,.5D0/
      Q=Z(4)*Z(4)-HALF
      DO 50 I=1,3
      DO 50 J=1,3
      X=Z(I)*Z(J)
      KGN=2*(J-I)
      IF(KGN)10,40,20
   10 SGN=KGN+3
      GO TO 30
   20 SGN=KGN-3
   30 KGN=6-I-J
      X=X+SGN*Z(KGN)*Z(4)
      GO TO 50
   40 X=X+Q
   50 T(I,J)=TWO*X
      RETURN
      END
```

```
      SUBROUTINE  ULR2(T,Q,L,M,N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION T(3,3),Q(3)
      DATA IOUT/6/
      DATA ONE,ZERO,HLFPI
     X   /1.D0,0.D0,1.5707963268D0/
       AACOS(X)=DACOS(X)
      AATAN2(X,Y)=DATAN2(X,Y)
      AABS(X)=DABS(X)
      IF(M-N)5,99,5
    5 IF(L-2)1,2,3
    1 IF(M-2)99,12,13
   12 K=3
      GO TO 40
   13 K=2
      GO TO 50
    2 IF(M-2)21,99,23
   21 K=3
      GO TO 50
   23 K=1
      GO TO 40
    3 IF(M-2)31,32,99
   31 K=2
   40 E=ONE
      GO TO 60
   32 K=1
50    E=-ONE
   60 IF(AABS(T(L,N)-ONE))70,70,80
   70 Q(2)=ZERO
      Q(3)=ZERO
      SQ1=E*T(K,M)
      CQ1=T(M,M)
      IF(N-L)95,110,95
   80 Q(2)=AACOS(T(L,N))
      IF(N-L)90,100,90
   90 Q(3)=AATAN2(-E*T(L,M),T(L,L))
      SQ1=-E*T(M,N)
      CQ1=T(K,N)
   95 Q(2)=E*(HLFPI-Q(2))
      GO TO 110
  100 Q(3)=AATAN2(T(L,M),E*T(L,K))
      SQ1=T(M,N)
      CQ1=-E*T(K,N)
  110 Q(1)=AATAN2(SQ1,CQ1)
      RETURN
   99 WRITE(IOUT,1000)
1000  FORMAT(5H1ULR2)
      CALL EXIT
      END
```

```
      SUBROUTINE ULR6(T,Z)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION T(9),Z(4)
      DATA ZERO,ONE,TWO
     X /0.D0,1.D0,2.D0/
      SSQRT(X)=DSQRT(X)
      X=T(1)+T(5)+T(9)+ONE
      IF(X)10,10,60
   10 D=ZERO
      X=ONE+T(1)-X
      IF(X)20,20,50
   20 A=ZERO
      X=T(5)+ONE
      IF(X)30,30,40
30    B=ZERO
      C=TWO
      GO TO 70
   40 B=SSQRT(TWO*X)
      C=(T(6)+T(8))/B
      GO TO 70
   50 A=SSQRT(TWO*X)
      B=(T(2)+T(4))/A
      C=(T(3)+T(7))/A
      GO TO 70
   60 D=SSQRT(X)
      A=(T(6)-T(8))/D
      B=(T(7)-T(3))/D
      C=(T(2)-T(4))/D
   70 Z(1)=A/TWO
      Z(2)=B/TWO
      Z(3)=C/TWO
      Z(4)=D/TWO
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION VMAGN(X,N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION X(1)
      Y=0.D0
      DO 1 I=1,N
1     Y=Y+X(I)*X(I)
      VMAGN=DSQRT(Y)
      RETURN
      END
```

Appendix D

SAMPLE CASE (INPUT AND OUTPUT)

TYPICAL DATA CARD INPUT

FIRST CARD

```
1
3222,  8418,31000,31000,  .1,.1,.1,  0,0,0,  5,0,0
4500,  43400,43400,11732,  0,0,0,  .099619,.008716,0,  -2.3,0,0
1,1,1,2
435,2962
100000,10
65,1000,.5
1,0,3
   74   75
BODY 2 X-AXIS TRACE IN INERTIAL SPACE    (DEGS)
NU - RELATIVE MISALIGNMENT BETWEEN X-AXIS    (DEG)
```

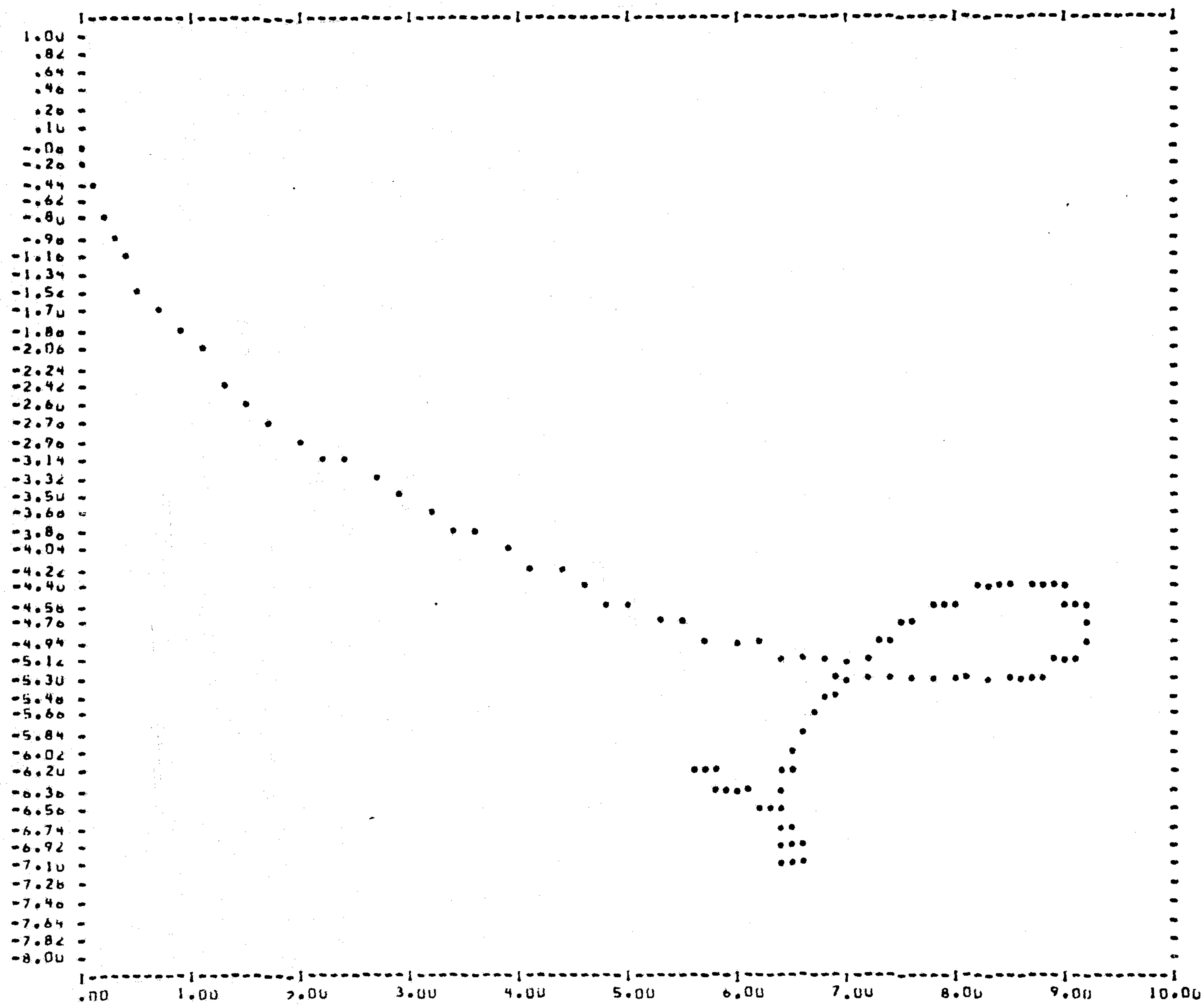
LAST CARD

TYPICAL DIGITAL PRINTOUT

TIME	ANG1	W1	ANG2	W2	ANG/NU	H1 TOT	H2 TOT	H TOT	E123T
0	5.7296	0	0	9.9619E-02	-5.1608	0	4323.5	4278.7	0
	5.7296	0	-2.9240E-10	8.7160E-03	-6.2464	0	378.27	1032.3	217.00
	5.7296	0	0	0	-5.1608	0	0	0	4.3139
					8.0961	0	4340.0	4401.4	221.31
.50000	5.7296	0	2.9418E-02	9.9533E-02	-5.1541	0	4319.7	4275.4	0
	5.7296	0	.25143	8.8846E-03	-5.9934	0	385.60	1041.0	216.70
	5.7296	0	2.8527	1.4736E-03	-2.3089	0	17.286	25.607	4.1074
					7.8986	0	4336.9	4400.4	220.81
1.0000	5.7296	0	.16013	9.9445E-02	-5.1061	0	4315.9	4273.8	0
	5.7296	0	.50293	8.9696E-03	-5.7365	0	390.15	1049.4	216.37
	5.7296	0	5.7033	1.9784E-03	.53636	0	23.211	67.742	3.8786
					7.6742	0	4333.6	4401.3	220.25
1.5000	5.7296	0	.18677	9.9356E-02	-5.0420	0	4312.0	4273.6	0
	5.7296	0	.75269	9.0607E-03	-5.4800	0	393.24	1056.6	216.01
	5.7296	0	8.5519	1.8041E-03	3.3777	0	21.165	123.06	3.6482
					7.4414	0	4330.0	4404.0	219.66
2.0000	5.7296	0	.28010	9.9265E-02	-4.9712	0	4308.1	4273.3	0
	5.7296	0	1.0009	9.1346E-03	-5.2244	0	396.45	1061.6	215.65
	5.7296	0	11.398	1.6734E-03	6.2161	0	19.632	174.59	3.4230
					7.2068	0	4326.3	4406.7	219.07
2.5000	5.7296	0	.38924	9.9173E-02	-4.8843	0	4304.1	4271.5	0
	5.7296	0	1.2457	9.2367E-03	-4.9707	0	400.87	1064.1	215.30
	5.7296	0	14.243	2.0524E-03	9.0508	0	24.078	208.53	3.1978
					6.9645	0	4322.8	4407.0	218.50
3.0000	5.7296	0	.52790	9.9079E-02	-4.7673	0	4300.0	4268.2	0
	5.7296	0	1.4835	9.3666E-03	-4.7214	0	406.51	1063.6	214.97
	5.7296	0	17.086	2.8092E-03	11.881	0	32.958	225.14	2.9657
					6.7058	0	4319.3	4404.5	217.94
3.5000	5.7296	0	.69954	9.8984E-02	-4.6167	0	4295.9	4264.7	0
	5.7296	0	1.7109	9.5005E-03	-4.4792	0	412.32	1059.4	214.64
	5.7296	0	19.926	3.4466E-03	14.706	0	40.438	235.80	2.7271
					6.4291	0	4315.8	4400.6	217.37
4.0000	5.7296	0	.89450	9.8886E-02	-4.4420	0	4291.7	4262.0	0
	5.7296	0	1.9287	9.6045E-03	-4.2447	0	416.84	1052.0	214.28
	5.7296	0	22.768	3.6019E-03	17.527	0	42.258	251.54	2.4890
					6.1410	0	4312.1	4397.1	216.77
4.5000	5.7296	0	1.0989	9.8790E-02	-4.2573	0	4287.5	4260.0	0
	5.7296	0	2.1395	9.6556E-03	-4.0160	0	419.05	1044.2	213.87
	5.7296	0	25.607	3.3347E-03	20.345	0	39.123	274.33	2.2596
					5.8501	0	4308.1	4394.7	216.13

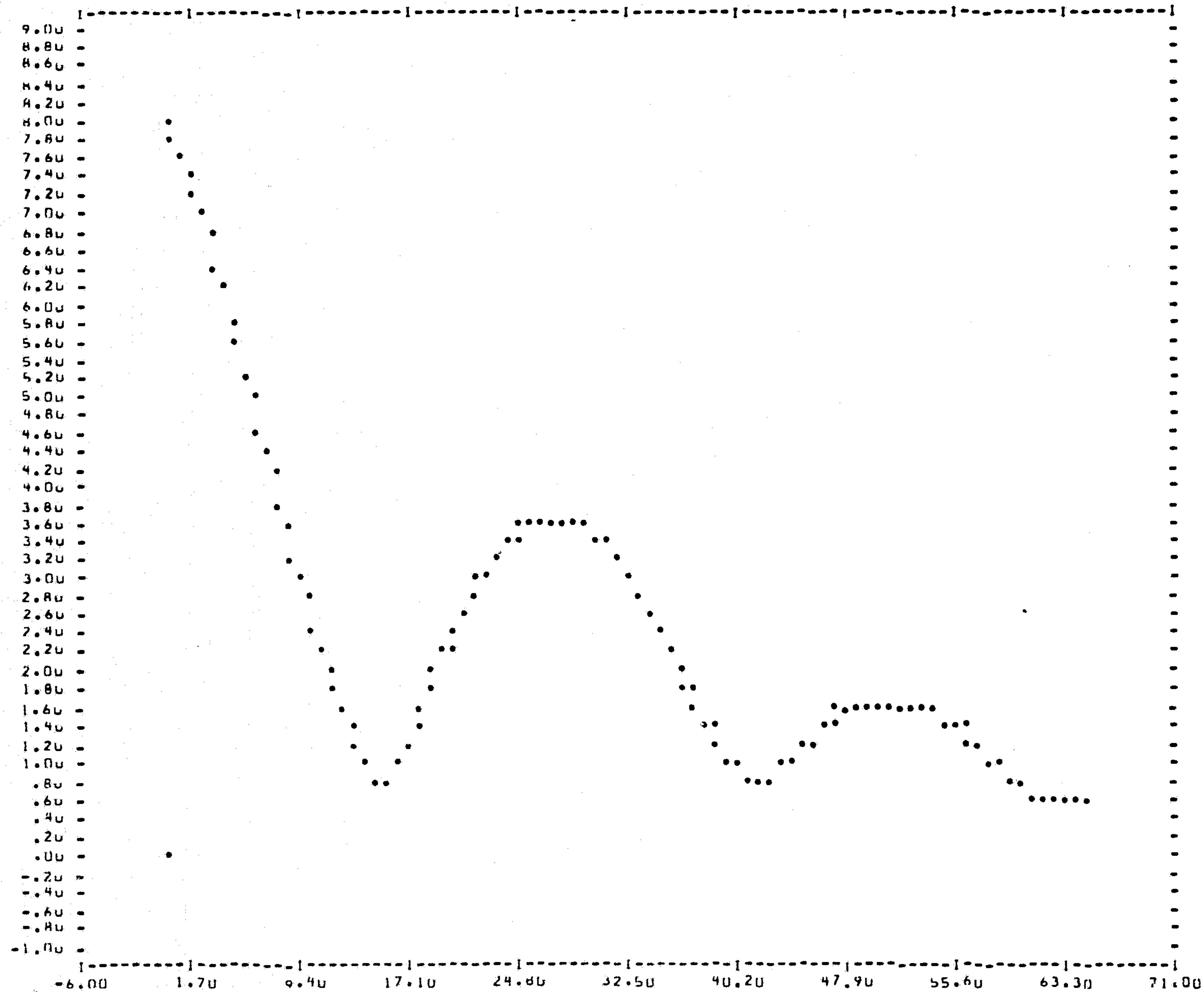
TYPICAL PRINTER PLOTS

BODY 2 X-AXIS TRACE IN INERTIAL SPACE (DEGS)

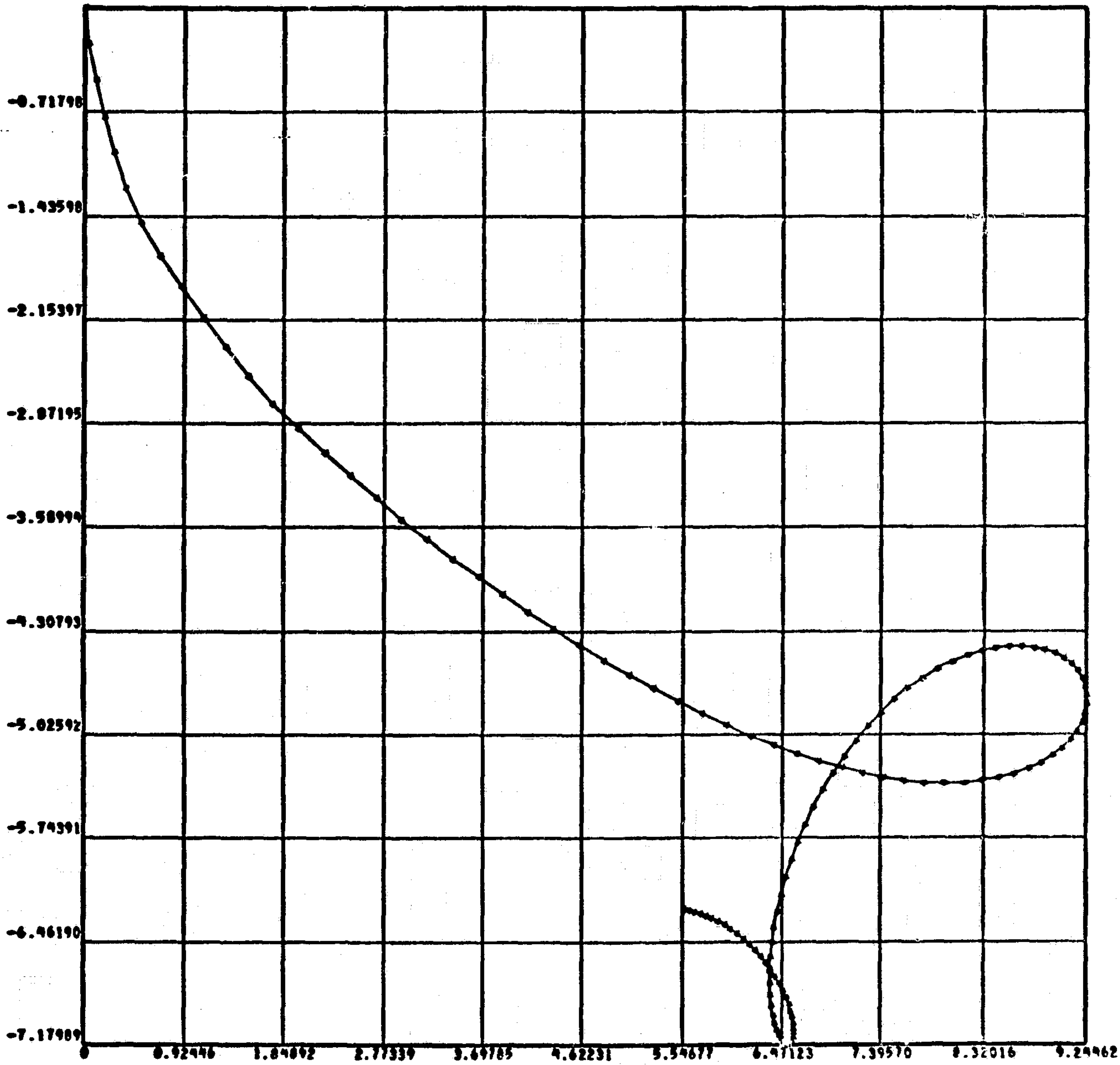


TYPICAL PRINTER PLOTS (Concluded)

NU - RELATIVE MISALIGNMENT BETWEEN X-AXIS (DEG)

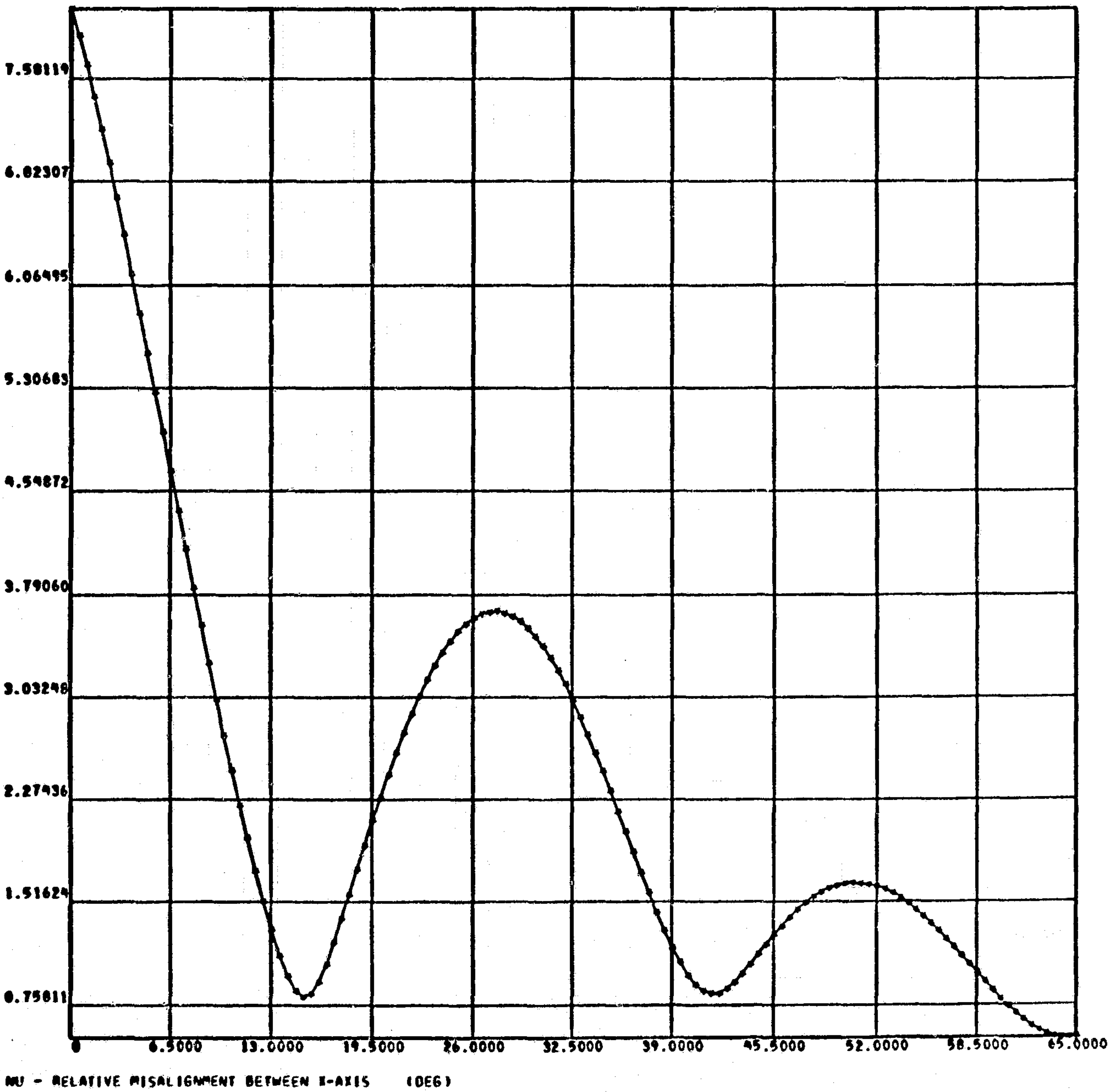


TYPICAL SC 4020 PLOTS



BODY 2 X-AXIS TRACE IN INERTIAL SPACE (DEGS)

TYPICAL SC 4020 PLOTS (Concluded)



End