NASA CR-66374

STUDY OF CONTROL COMPUTERS FOR CONTROL MOMENT GYRO STABILITY AND CONTROL SYSTEMS

VOLUME II (SIMULATION)

Prepared under Contract No. NAS 1-6698 by
INTERNATIONAL BUSINESS MACHINES CORPORATION

for Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C., March 1967

N67-30137
(ACCESSION NUMBER) (THRU)
263 1
(PAGES) (CODE)
CR-66374 08
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)
FACILITY FORM 602

NASA CR-66374

STUDY OF CONTROL COMPUTERS FOR CONTROL MOMENT GYRO STABILITY AND CONTROL SYSTEMS

By D.P. Bedford, H. Markarian, N.L. Pleszkoch

VOLUME II (SIMULATION)

DISTRIBUTION OF THIS REPORT IS PROVIDED IN THE INTEREST OF INFORMATION EXCHANGE. RESPONSIBILITY FOR THE CONTENTS RESIDES IN THE AUTHOR OR ORGANIZATION THAT PREPARED IT.

Prepared under Contract ~~No~~. NAS1-6698 by
INTERNATIONAL BUSINESS MACHINES CORPORATION

for Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PRECEDING PAGE BLANK NOT FILMED.

ABSTRACT

This 6 months' contractual study effort was performed under NASA Contract NAS1-6698 awarded to the IBM Corporation on 2 September 1966, in response to bids on the Langley Research Center's Statement of Work L-7035, dated 22 June 1966

The study has two objectives. The first objective was to perform an analysis of the control computer computational requirements. This included preliminary hardware sizing for a control moment gyro (CMG) stability and control system used in an Apollo Applications Spacecraft to perform typical control tasks required for experiments such as horizon spectrometry, earth mapping, and solar astronomy. The second objective was to develop a fixed word-length, digital model of the control computer to be incorporated in a sample-data simulation of the integrated CMG control system for manned spacecraft. The final report comprises two volumes. Volume I describes the overall engineering analyses, and Volume II discusses the digital simulation program from a user's viewpoint.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

Section		Page
	ABSTRACT	iii
	SYMBOLS	ix
1	INTRODUCTION	1-1
2	SIMULATION IMPLEMENTATION	2-1
	Program Structure	2-1
	Rigid Body Equations of Motion	2-3
	Fixed Point Programming Techniques	2-6
	Coulomb Friction	2-10
3	PROGRAM DESCRIPTION	3-1
	MAIN Program	3-4
	Subroutine INITIL	3-6
	Subroutine COMP	3-26
	Subroutine RKII	3-81
	Subroutine INPUT	3-86
	Subroutine OUTPUT	3-89
	Subroutine MANUAL	3-96
	Subroutine SUMARY	3-101
	Subroutine DERIV	3-104
	Subroutine TRIG	3-106
	Subroutine RATE	3-108
	Subroutine POSVEL	3-110
	Subroutine EXPER	3-113
	Subroutine DISTRB	3-118
	Subroutine FINDMG	3-121
	Subroutine FINDA	3-124
	Subroutine FINDB	3-127
	Subroutine EVDER	3-133
	Subroutine ENERGY (INDRS)	3-136
	Subroutine STORE (INDEX)	3-140
	Subroutine FANDG	3-143
	Subroutine UPDAT3	3-147

CONTENTS (Continued)

Section		Page
	Function COST (XXXX, YYYY, J)	3-150
	Subroutine SINCC (Y, SINY)	3-154
	Subroutine CCATAN (ANGT, NUM, DEN)	3-156
	Subroutine CCASIN (ANG, SINANG)	3-159
	Function SQRTCC (X)	3-161
	Function DMULT (X, Y)	3-163
	Function QUOTNT (DBLNUM, SGLDEN)	3-165
	Subroutine AINVSE (A, AINV, DETA, X, Y, NSYM)	3-167
	Subroutine RANO (SIG, BIAS, XK, R)	3-169
	Subroutine ZERO	3-174
4	USER'S GUIDE	4-1
	Introduction	4-1
	Input Data	4-1
	Output Data	4-12
Appendix		
A	RIGID BODY EQUATIONS OF MOTION	A-1
B	VEHICLE ATTITUDE EQUATIONS	B-1
C	CMG GIMBAL SERVO EQUATIONS	C-1
D	SENSOR EQUATIONS	D-1
E	VARIABLE NAME LISTS	E-1
F	ANGULAR MOMENTUM TEST	F-1

ILLUSTRATIONS

Figure		Page
1-1	Attitude Control System Functional Diagram	1-2
2-1	CMG Simulation Functional Block Diagram	2-1
2-2	Coulomb Friction Model	2-10
2-3	Model Response	2-11
2-4	Coulomb Friction Math Flow	2-12
3-1	Subroutine Linkage	3-2
3-2	Computation Cycle Timing	3-4
3-3	Initialization Routines	3-6
3-4	Simulated Control Computer	3-27
3-5	Direction Cosine Integration	3-28
3-6	Compute Euler Angles, Navigation, Mode Branch	3-29
3-7	Experiment One, Earth Mapping	3-30
3-8	Experiment Two, Inertial Mode	3-32
3-9	Experiment Three, Horizon Spectrometry	3-33
3-10	Experiment Four, Microwave Transmission	3-36
3-11	Mode 5 — Attitude Hold, Mode 6 — Maneuver, Mode 7 — Manual	3-37
3-12	Compensation and Baseline Control Law	3-39
3-13	Precomputation for Control Laws 2 and 3	3-40
3-14	Algebraic Control Law	3-42
3-15	Iterative Control Law	3-43
3-16	Desaturation Test	3-44
3-17	Cost Function Subroutine	3-150
C-1	Gimbal Servo Loop	C-3
D-1	Star Tracker Servo Loop	D-3

G_α, (G_β)	outer (inner) CMG gimbal drive gear ratio
g	coefficient of the navigation f and g series
H	CMG spin momentum magnitude, lb-ft-sec
h	normalized CMG spin momentum vector
$\underline{i}, \underline{j}, \underline{k}$	unit vectors along vehicle x, y, z axes
$\underline{I}, \underline{J}, \underline{K}$	unit vectors along inertial X, Y, Z axes
$J_{m\alpha}$, $(J_{m\beta})$	outer (inner) gimbal torquer rotor moment of inertia, lb-ft-sec^2
K_c	control law gain
K_e, K_r	multipliers in the baseline control law
K_x, K_y, K_z	attitude control system gain terms, sec^{-1}
K_x'	rate feedback gain
K_s	algebraic control law parameter
$K_\alpha, K_\beta, K_{\alpha\beta}$	cost function constants
$K_{TOLERANCE}$	iterative control law parameter
k_i	linear combination of attitude and rate errors for the baseline control law
L	longitude of spacecraft
$[M_{ij}]$	direction cosine matrix, commanded body to inertial coordinates
$\underline{M}$	external torque vector, ft-lb
M_α	G_α times the torque produced by the outer gimbal torquer, ft-lb
M_β	G_β times the torque produced by the inner gimbal torquer, ft-lb
$M_{T\alpha}$	total torque interaction between CMG and the vehicle along the outer gimbal axis, ft-lb

$M_{T\beta}$	total torque interaction between the inner gimbal assembly and the outer gimbal ring along the inner gimbal axis, ft-lb
m	mass, slugs
$[m_{ij}]$	direction cosine matrix, body to inertial coordinates
Q	constant defining C.M. change, $m_M m_S/(m_M+m_S)$, slugs
$\underline{q}$	position vector of mass element measured from C.M. of the main vehicle, feet
R	gimbal rate $\dot{\alpha}$ or $\dot{\beta}$ in the iterative control law, rad/sec
$\underline{R}$	position vector of target from the geocenter, feet
$\underline{S}$	first moment of mass of the total system about the C.M. of the main vehicle, lb-sec^2
s	Laplace transform variable, rad/sec
$\underline{s}$	slant range vector, feet
$\dot{\underline{s}}_E$	slant range rate vector with respect to the earth, ft/sec
T	sampling interval, seconds
$\underline{T}_b$	torque due to CMG precession at vehicle rates, ft-lb
$\underline{T}_c$	remainder torque, $\underline{T}_d - T_b$, ft-lb
$\underline{T}_d$	desired torque, ft-lb
$\underline{u}$	normalized position vector of the vehicle
V_0	circular velocity at earth's radius, ft/sec
$\underline{V}$	slant range rate vector with respect to the earth, $\dot{\underline{s}}_E$, ft/sec
X, Y, Z	inertial coordinates
x, y, z	vehicle principal axes coordinates
Z_{EI}	local ellipsoid vertical

α	CMG outer gimbal angle
β	CMG inner gimbal angle
$\underline{\epsilon}$	negative of small angle attitude error
$\dot{\eta}$	nominal vehicle rate about the z axis in the horizon spectrometry experiment
θ	Euler angle
λ	vehicle geodedic latitude
λ_R	vehicle geocentric latitude
$\underline{\rho}$	position vector of the vehicle measured from the geocenter, feet
ρ_E	earth's radius, feet
ϕ	Euler angle
ψ	Euler angle
Ω	CMG spin rate, rad/sec
Ω_{EI}	earth's rate in inertial coordinates, rad/sec
$\underline{\omega}$	vehicle angular rate vector, rad/sec
$\underline{\omega}_i$	inertial rate of the ith CMG inner gimbal ring resolved along inner gimbal axes, rad/sec
$\underline{\omega}_i'$	inertial rate of the ith CMG outer gimbal ring resolved along outer gimbal axes, rad/sec

Superscripts:

$'$ (prime)	indicated
$''$ (double prime)	signal at the output of the stabilization compensation
$\cdot$ (dot)	time derivative

Subscripts:

a	outer gimbal
b	inner gimbal
c	command
d	desired
g	gyro rotor
i	ith element
M	crew member
n	step of iteration
r	remainder in the iterative control law
S	total system excluding the moving mass
t	telescope
*	double precision control computer variable

Section 1

INTRODUCTION

A digital computer program that rigorously simulates the system depicted in Figure 1-1 has been developed under contract number NAS 1-6698, issued by NASA's Langley Research Center.

This simulation, a contract delivery item, was required to support Langley's research in Control Moment Gyro (CMG) systems, and can help evaluate the performance of CMGs that are controlled by an on-board digital computer in many attitude control situations. In Volume I of the report, the digital simulation has been used to validate the attitude control loop compensation developed on the basis of a linearized model of the system. Control computer word length requirements to satisfy accuracy specifications in the performance of four space experiments were also established with the aid of the program.

In addition to a detailed description of the simulation, this volume emphasizes the unusual techniques involved to allow effective use of the program.

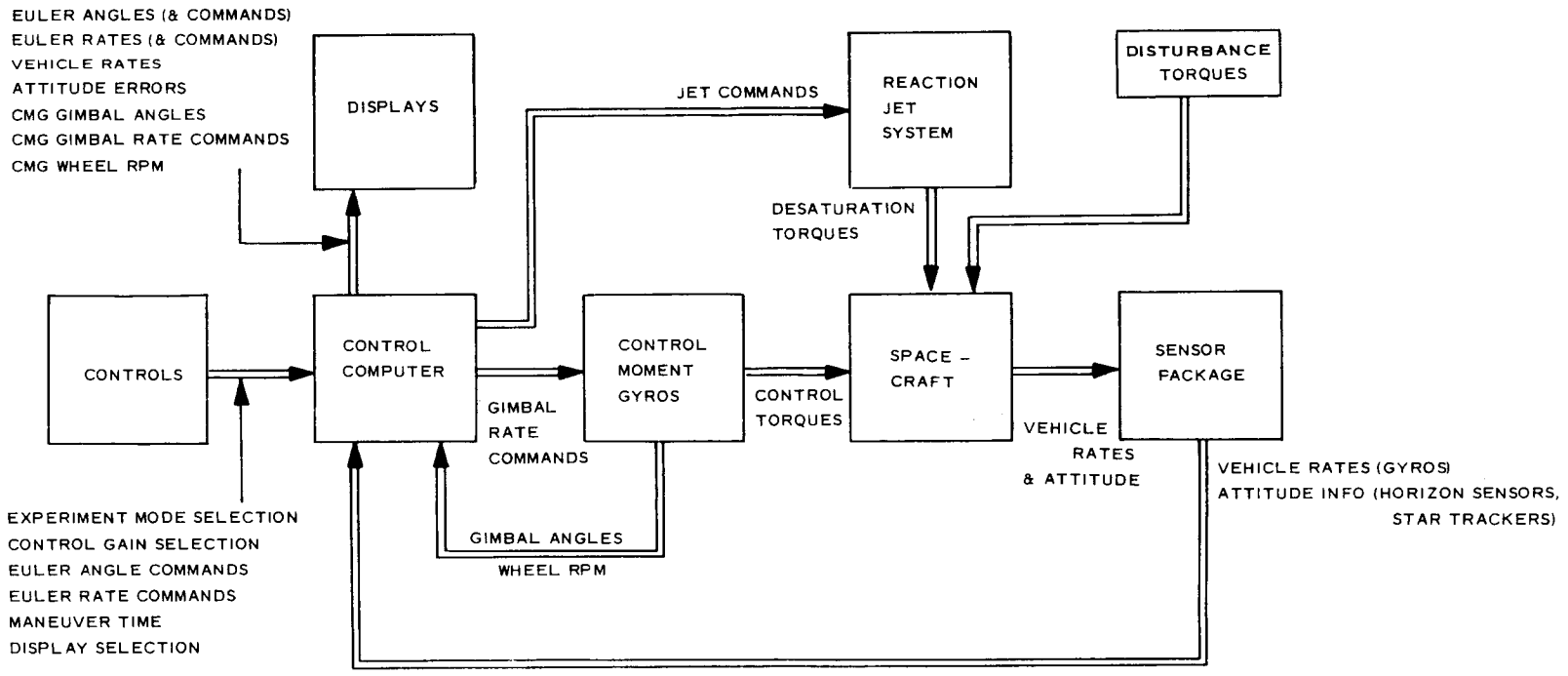


Figure 1-1. Attitude Control System Functional Diagram

Section 2

SIMULATION IMPLEMENTATION

Program Structure

Introduction — The CMG simulation functionally conforms to Figure 2-1. These functions fall into four categories: (1) Input/initialization, (2) Output, (3) Control computer, (4) Environment. This section presents a description of these categories that serve as the program basis.

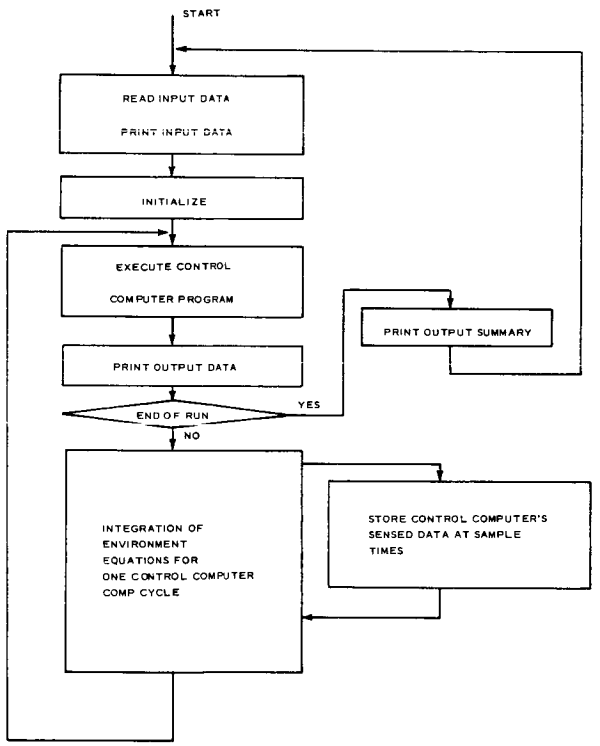


Figure 2-1. CMG Simulation Functional Block Diagram

Input/initialization — Input data is read into an input buffer area. This source data is unaltered so that subsequent runs require only input data changes. After reading data, the entire input buffer is written on the print tape to completely identify the run conditions.

The initialization routine places input data items in the using storage areas. This data is converted, if necessary, to conform to a consistant system of units as follows:

distance	=	foot
force	=	pound
mass	=	slug
time	=	second
angle	=	radian
emf	=	volt

One exception is the star tracker loop with torque in inch-ounces and inertia in inch-ounce-sec^2.

Output. — Output data describing the state of the vehicle, control computer, and sensors can be written on the print tape as often as each control computer cycle. This data is converted radians to degrees (angle), and lb. ft. /sec to watts (power).

At end end of each run a summary is written on the print tape. This output supplies information per CMG gimbal on maximum power, total power, and average power.

Control computer. — This category includes the fixed point, adjustable word length control computer program and its fixed point utility routines (sine, cosine, arc tangent, arc sine, double precision multiply and divide). The cost function used by the iterative control law is programmed as a separate subroutine using floating FORTRAN. This was done to facilitate changes in the function since the user will probably want to use many variations to appreciate the versatility of the iterative control law. This reprogramming capability and the fact that the floating point computation of this function does not increase the accuracy of the control law justify this provision.

A floating point version of the control computer is also supplied to provide reference runs with minimal quantization effects.

Environment. — The operating environment of the control computer includes the vehicle, CMGs, sensors, and disturbing influences.

The basis of this portion of the simulation is an adjustable step, fourth-order Runge-Kutta integration routine. The step size is regulated to keep an analytic approximation of the local truncation error within bounds specified by the input data. To support the integrator, it is necessary to define a state vector and the equations necessary to compute derivative of the state vector. The integration scheme is augmented with the controls needed to store the control computer input data items prior to their use to simulate conversion and computation delays within the control computer.

The equations for computing the state vector derivatives are presented in appendices A, B, C, D.

Rigid Body Equations of Motion

The rigid body equations for the nine angular accelerations of the vehicle and the CMG gimbals are developed in Volume I. These equations can be arranged as:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & 0 & a_{19} \\ a_{12} & a_{22} & a_{23} & 0 & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{13} & a_{23} & a_{33} & a_{34} & a_{35} & 0 & a_{37} & a_{38} & a_{39} \\ a_{14} & 0 & a_{34} & a_{44} & 0 & 0 & 0 & 0 & 0 \\ a_{15} & a_{25} & a_{35} & 0 & a_{55} & 0 & 0 & 0 & 0 \\ a_{16} & a_{26} & 0 & 0 & 0 & a_{66} & 0 & 0 & 0 \\ a_{17} & a_{27} & a_{37} & 0 & 0 & 0 & a_{77} & 0 & 0 \\ 0 & a_{28} & a_{38} & 0 & 0 & 0 & 0 & a_{88} & 0 \\ a_{19} & a_{29} & a_{39} & 0 & 0 & 0 & 0 & 0 & a_{99} \end{pmatrix} \begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ \ddot{\beta}_1 \\ \ddot{\alpha}_1 \\ \ddot{\beta}_2 \\ \ddot{\alpha}_2 \\ \ddot{\beta}_3 \\ \ddot{\alpha}_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \end{pmatrix} \qquad (2\text{-}1)$$

Inspection of Equation 2-1 indicates that solving the equations of motions for the highest derivatives involved the inversion of the 9 x 9 coefficient matrix $\lfloor a_{y}^{\cdot\cdot} \rfloor$. The actual method of solution reduces this requirement to the inversion of a 3 x 3 matrix by taking advantage of the fact that each of the last six equations of motion is a function of the vehicle accelerations, plus only one of the CMG gimbal accelerations. Thus, it is possible to solve each of the last six equations of Equation (2-1) for the respective gimbal acceleration as

$$\left.\begin{aligned}
\ddot{\beta}_1 &= (b_4 - a_{14}\dot{\omega}_x - a_{34}\dot{\omega}_z)/a_{44} \\
\ddot{\alpha}_1 &= (b_5 - a_{15}\dot{\omega}_x - a_{25}\dot{\omega}_y - a_{35}\dot{\omega}_z)/a_{55} \\
\ddot{\beta}_2 &= (b_6 - a_{16}\dot{\omega}_x - a_{26}\dot{\omega}_y)/a_{66} \\
\ddot{\alpha}_2 &= (b_7 - a_{17}\dot{\omega}_x - a_{27}\dot{\omega}_y - a_{37}\dot{\omega}_z)/a_{77} \\
\ddot{\beta}_3 &= (b_8 - a_{28}\dot{\omega}_y - a_{38}\dot{\omega}_z)/a_{88} \\
\ddot{\alpha}_3 &= (b_9 - a_{19}\dot{\omega}_x - a_{29}\dot{\omega}_y - a_{39}\dot{\omega}_z/a_{99}
\end{aligned}\right\} \tag{2-2}$$

Substitution of Equation (2-2) into the first three equations of Equation (2-1) gives

$$\left.\begin{aligned}
&\dot{\omega}_x\left(a_{11} - \frac{a_{14}^2}{a_{44}} - \frac{a_{15}^2}{a_{55}} - \frac{a_{16}^2}{a_{66}} - \frac{a_{17}^2}{a_{77}} - \frac{a_{19}^2}{a_{99}}\right) \\
&+\dot{\omega}_y\left(a_{12} - \frac{a_{15}a_{25}}{a_{55}} - \frac{a_{16}a_{26}}{a_{66}} - \frac{a_{17}a_{27}}{a_{77}} - \frac{a_{19}a_{29}}{a_{99}}\right) \\
&+\dot{\omega}_z\left(a_{13} - \frac{a_{14}a_{34}}{a_{44}} - \frac{a_{15}a_{35}}{a_{55}} - \frac{a_{17}a_{37}}{a_{77}} - \frac{a_{19}a_{39}}{a_{99}}\right)
\end{aligned}\right\} = \left\{\begin{aligned}
&b_1 - \frac{a_{14}}{a_{44}}b_4 \\
&- \frac{a_{15}}{a_{55}}b_5 - \frac{a_{16}}{a_{66}}b_6 \\
&- \frac{a_{17}}{a_{77}}b_7 - \frac{a_{19}}{a_{99}}b_9
\end{aligned}\right. \tag{2-3}$$

$$
\left.\begin{aligned}
&\dot{\omega}_x\left(a_{12} - \frac{a_{15}a_{25}}{a_{55}} - \frac{a_{16}a_{26}}{a_{66}} - \frac{a_{17}a_{27}}{a_{77}} - \frac{a_{19}a_{29}}{a_{99}}\right)\\
&+\dot{\omega}_y\left(a_{22} - \frac{a_{25}^2}{a_{55}} - \frac{a_{26}^2}{a_{66}} - \frac{a_{27}^2}{a_{77}} - \frac{a_{28}^2}{a_{88}} - \frac{a_{29}^2}{a_{99}}\right)\\
&+\dot{\omega}_z\left(a_{23} - \frac{a_{25}a_{35}}{a_{55}} - \frac{a_{27}a_{37}}{a_{77}} - \frac{a_{28}a_{38}}{a_{88}} - \frac{a_{29}a_{39}}{a_{99}}\right)
\end{aligned}\right\} = \left\{\begin{aligned}
&b_2 - \frac{a_{25}}{a_{55}}b_5\\
&-\frac{a_{26}}{a_{66}}b_6 - \frac{a_{27}}{a_{77}}b_7\\
&-\frac{a_{28}}{a_{88}}b_8 - \frac{a_{29}}{a_{99}}b_9
\end{aligned}\right.
$$

(2-3)
(Cont.)

$$
\left.\begin{aligned}
&\dot{\omega}_x\left(a_{13} - \frac{a_{14}a_{34}}{a_{44}} - \frac{a_{15}a_{35}}{a_{55}} - \frac{a_{17}a_{37}}{a_{77}} - \frac{a_{19}a_{39}}{a_{99}}\right)\\
&\dot{\omega}_y\left(a_{23} - \frac{a_{25}a_{35}}{a_{55}} - \frac{a_{27}a_{37}}{a_{77}} - \frac{a_{28}a_{38}}{a_{88}} - \frac{a_{29}a_{39}}{a_{99}}\right)\\
&\dot{\omega}_z\left(a_{33} - \frac{a_{34}^2}{a_{44}} - \frac{a_{35}^2}{a_{55}} - \frac{a_{37}^2}{a_{77}} - \frac{a_{38}^2}{a_{88}} - \frac{a_{39}^2}{a_{99}}\right)
\end{aligned}\right\} = \left\{\begin{aligned}
&b_3 - \frac{a_{34}}{a_{44}}b_4\\
&-\frac{a_{35}}{a_{55}}b_5 - \frac{a_{37}}{a_{77}}b_7\\
&-\frac{a_{38}}{a_{88}}b_8 - \frac{a_{39}}{a_{99}}b_9
\end{aligned}\right.
$$

which can be written as

$$
\begin{pmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{12} & a'_{22} & a'_{23} \\ a'_{13} & a'_{23} & a'_{33} \end{pmatrix} \begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \end{pmatrix} \tag{2-4}
$$

Note that the 3 x 3 matrix is indicated as symmetric as verified by inspection of the corresponding coefficients of $\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$ in Equation (2-3). Therefore, only six a'_{ij} have to be evaluated.

The method used to solve for the derivative equations presented in Appendix A is summarized:

a. Evaluate the a_{ij} and b_i of Equation (2-1)

b. Compute the a'_{ij} and b'_i of Equation (2-4)

c. Solve Equation (2-4) for $\dot{\omega}_i$ by inverting the 3 x 3 coefficient matrix $[a'_{ij}]$

d. Solve Equation (2-2) for $\ddot{\beta}_i$ and $\ddot{\alpha}_i$ using the values of $\dot{\omega}_i$ just computed.

Fixed Point Programming Techniques

Introduction. —In order to simulate the calculations of the fixed point control computer, using FORTRAN IV, it is necessary to treat all variables and parameters as scaled integers. To establish scaling factors, the maximum absolute value of each variable and parameter must be know. The assumed maximum values are presented in the appropriate variable name lists in Appendix E of this report.

The following paragraphs outline the procedures used in scaling and programming the simulated control computer.

Scaling considerations. —The maximum values of all variables have been selected as integer powers of 2. This implies that all arithmetic scaling adjustments reduce to shifting operations and all scaling problems can be reduced to consideration of the associated integer powers of 2 and the number of data bits, n.

Given:

n, the number of magnitude bits of the control computer word

x, a variable such that $|x| \leq x_{max} = 2^{kx}$

X_I, a scaled FORTRAN integer representing x; the following identifications can be made.

Full scale n-bit integer; $FS = 2^n - 1 \cong 2^n = FS'$

K_x is a scale factor such that

$$X_I = K_x x \tag{2-5}$$

Since it is desirable to have X_I approximately equal to FS, when $x = x_{max}$ we can solve for K_x,

$$K_x = \frac{X_I}{x} = \frac{FS'}{x_{max}} = \frac{2^n}{2^{k_x}} = 2^{n-k_x} \tag{2-6}$$

Arithmetic operations. —Let us define the variables x, y, z; their maximum values 2^{kx}, 2^{ky}, 2^{kz}; and their FORTRAN integer representations X_I, Y_I, Z_I; and consider the scaling associated with the following arithmetic operations.

Addition—To program the operation $z = x + y$ using the integers X_I, Y_I, Z_I, first get a common scale factor for X_I and Y_I, then perform the addition, and finally adjust the scale factor of the answer to K_z. Assume $k_x \leq k_y$, then a right shift of $k_y - k_x$ places (i. e., division by $2^{k_y - k_x}$) performed on X_I will produce the value x with a scale factor K_y.

$$\frac{X_I}{2^{k_y-k_x}} = \frac{K_x x}{2^{k_y-k_x}} = \frac{2^{n-k_x}}{2^{k_y-k_x}} x = 2^{n-k_y} x = Kyx$$

The sum $K_y (x + y)$ must be appropriately shifted to yield a scale factor K_z. The three cases are:

$k_z = k_y$

$$Z_I = \frac{X_I}{2^{k_y-k_x}} + Y_I \tag{2-7}$$

$k_z > k_y$

$$Z_I = \frac{\frac{X_I}{2^{k_y-k_x}} + Y_I}{2^{k_z-k_y}} \tag{2-8}$$

$k_z < k_y$

$$Z_I = \left(\frac{X_I}{2^{k_y-k_x}} = Y_I\right) 2^{k_y-k_z} \tag{2-9}$$

Note that the shift factors $2^{k_y-k_x}$, etc., are constants independent of n.

Multiplication—The operation $z = xy$ involves multiplication of X_I, Y_I producing a double register (i. e., 2n bits) product. This result is shifted to yeild a single register answer with a scale factor K_z.

$$X_I \; Y_I = (2^{n-k_x}) \, (2^{n-k_y}) \; xy \tag{2-10}$$

Dividing both sides of Equation (2-10) by $2^{n+k_z-k_x-k_y}$ yields

$$\frac{X_I \, Y_I}{2^{n+k_z-k_x-k_y}} = 2^{n-k_z} xy = K_z xy = Z_I$$

or

$$Z_I = \frac{X_I \, Y_I}{2^{n+kz-kx-ky}} \tag{2-11}$$

Note that this shift factor is a function of n.

Division—An actual fixed point division, $z = x/y$, is performed by placing the divident within a double register so that the properly scaled quotient will appear in a single register (i. e., the least significant n bits of the double register) after the division takes place. Simulation of the dividend placement is accomplished by a left shift of $n-k_z+k_x-k_y$, or

$$X_I \cdot 2^{n-k_z+k_x-k_y} = 2^{n-k_x} \cdot 2^{n-k_z+k_x-k_y} x = 2^{2n-k_z-k_y} x \tag{2-12}$$

dividing both sides of Equation (2-12) by Y_I

$$\frac{X_I \cdot 2^{n-k_z+k_x-k_y}}{Y_I} = \frac{2^{2n-k_z-k_y} x}{2^{n-k_y} y} = 2^{n-k_z} \frac{x}{y} = K_z \frac{x}{y} = Z_I$$

or

$$Z_I = \frac{(X_I \cdot 2^{n-k_z+k_x-k_y})}{Y_I} \tag{2-13}$$

This shifting factor is also a function of n.

Input/output. —When the control computer executes its program, it operates on data supplied by the environmental simulation. This data is generated in floating point form and must be scaled according to its maximum value. The data is converted to a fixed point integer and quantized more if the bit length of the particular converter is less than n.

When the control computer presents data to the environment, these operations are reversed. The n-bit output integer is quantized if the D/A conversion is less than n-bits—then converted to floating point. The scale factor is then removed and the data is ready for use by the environment.

The FORTRAN statements that simulate these operations can be illustrated after defining:

x = A floating point input variable

X_I = The integer version of x

n_x = The number of bits carried in the input conversion of x. $n_x \leq n$

y = A floating point output variable.

Y_I = The integer version of y

n_y = The number of bits carried in the output conversion of y. $n_y \leq n$

The input scaling and conversion statement is

$$X_I = x \cdot 2^{(n-k_x)} \tag{2-14}$$

Note that $2^{(n-k_x)}$ is the scale factor K_x by Equation (2-6). The RHS of Equation (2-14) is the integer X_I in floating form. FORTRAN converts this to a fixed point representation automatically. The additional quantization associated with the A/D conversion is simulated with this all integer statement:

$$X_I = \lfloor X_I/2^{(n-n_x)} \rfloor \cdot 2^{(n-n_x)} \tag{2-15}$$

The division is performed first and causes X_I to be right shifted $n-n_x$ places. As a result the least significant $n-n_x$ bits of X_I are lost. The multiplication shifts this quantity left replacing the lost bits with zeros.

The corresponding statements used for output adjustments are:

$$y = \lfloor Y_I/2^{(n-n_y)} \rfloor \cdot 2^{(n-n_y)} \tag{2-16}$$

$$y = y \cdot 2^{(k_y - n)} \tag{2-17}$$

Coulomb Friction

To maintain a high degree of simulation accuracy, it was necessary to implement a better model for coulomb friction than the standard signum function shown in Figure 2-2.

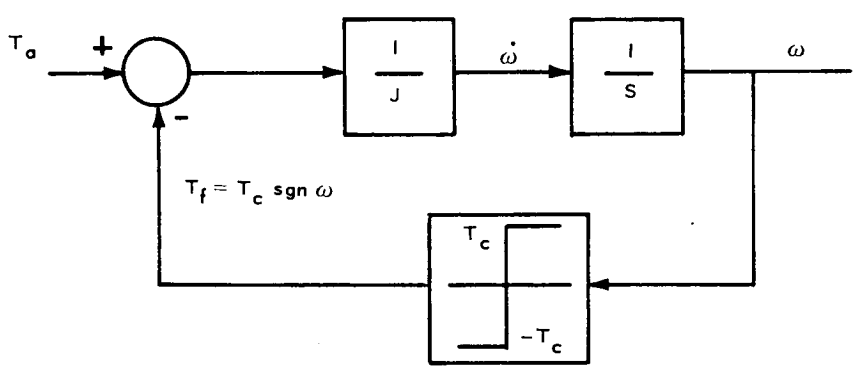


Figure 2-2. Coulomb Friction Model

Using a value of T_f equal to T_c sgn ω can lead to a low amplitude limit cycle because $\dot{\omega}$ is assumed constant over an integration step, Δt. This is best illustrated by considering an example where the system of Figure 2-2 has an initial $\omega = \omega_o$ and an applied torque, T_a, equal to zero. Figure 2-3 shows the desired time histories of ω and T_f. Figure 2-3a corresponds to the signum function model and Figure 2-3b corresponds to the model actually used.

The model for coulomb friction used in this simulation incorporates the logic of Figure 2-4 for computing T_f. Then

$$\dot{\omega} = \frac{T_a - T_f}{J}$$

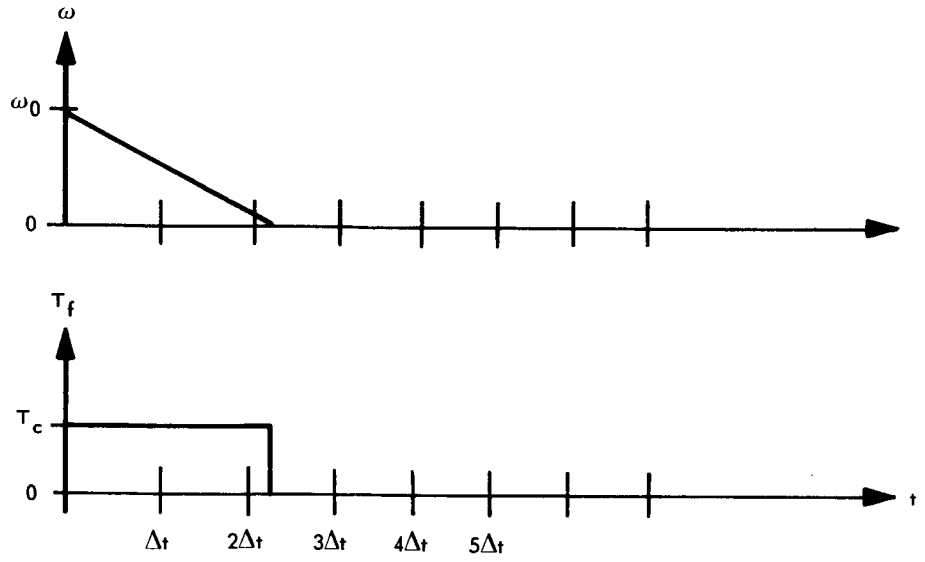


Figure 2-3. Model Response

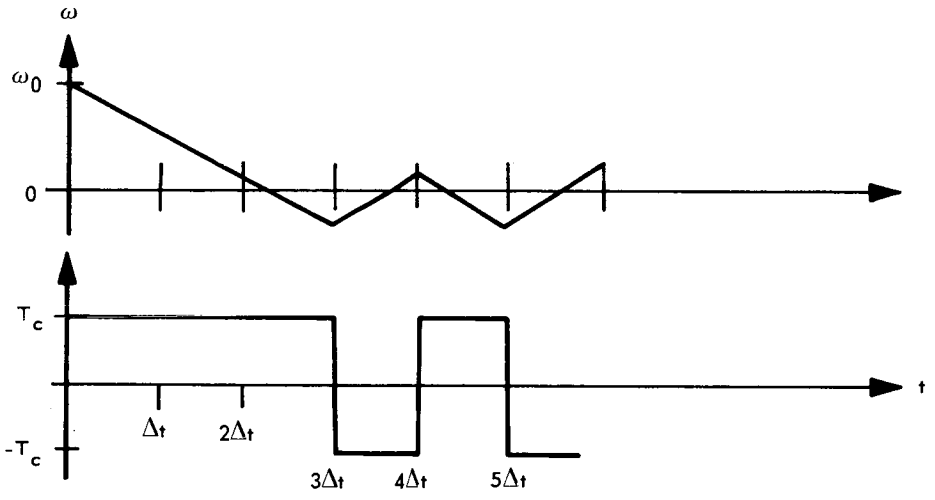


Figure 2-3a

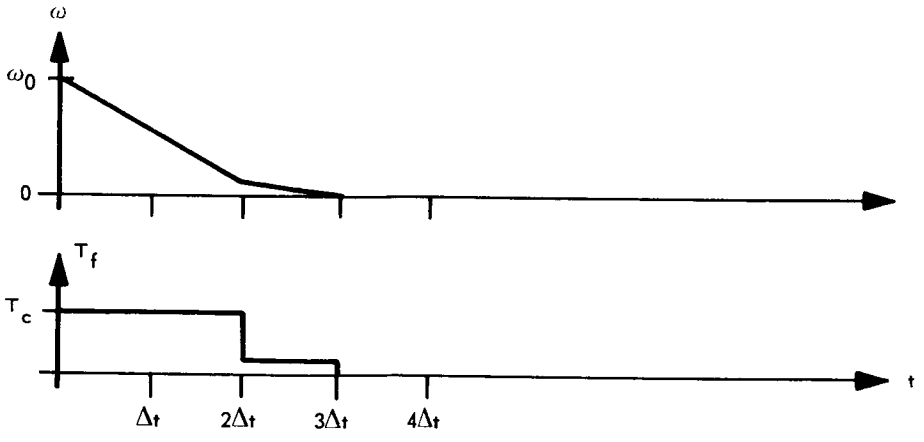
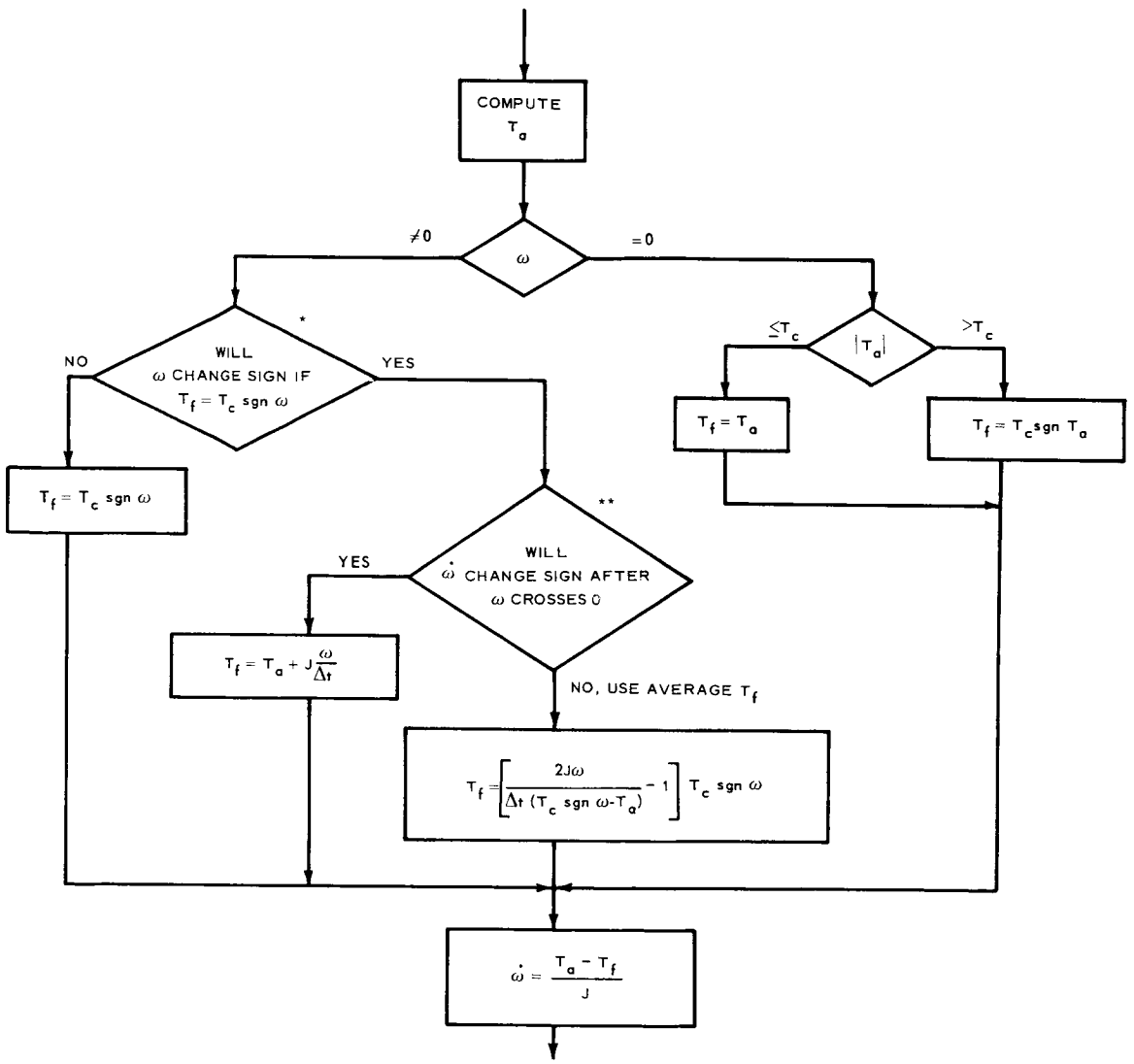


Figure 2-3b



* TAKE NO BRANCH, IF $\frac{T_a - T_c \operatorname{sgn} \omega}{J\omega}\Delta t \geq -1$

** TAKE NO BRANCH, IF $|T_a| > T_c$

Figure 2-4. Coulomb Friction Math Flow

Section 3

PROGRAM DESCRIPTION

The CMG simulation implements the functions of Figure 2-1 using a main program and 31 subroutines. The following sections describe each routine and include listings, routine linkage, purpose and references to source equations, and supporting figures.

For clarification, Figure 3-1 illustrates the major subroutine linkage.

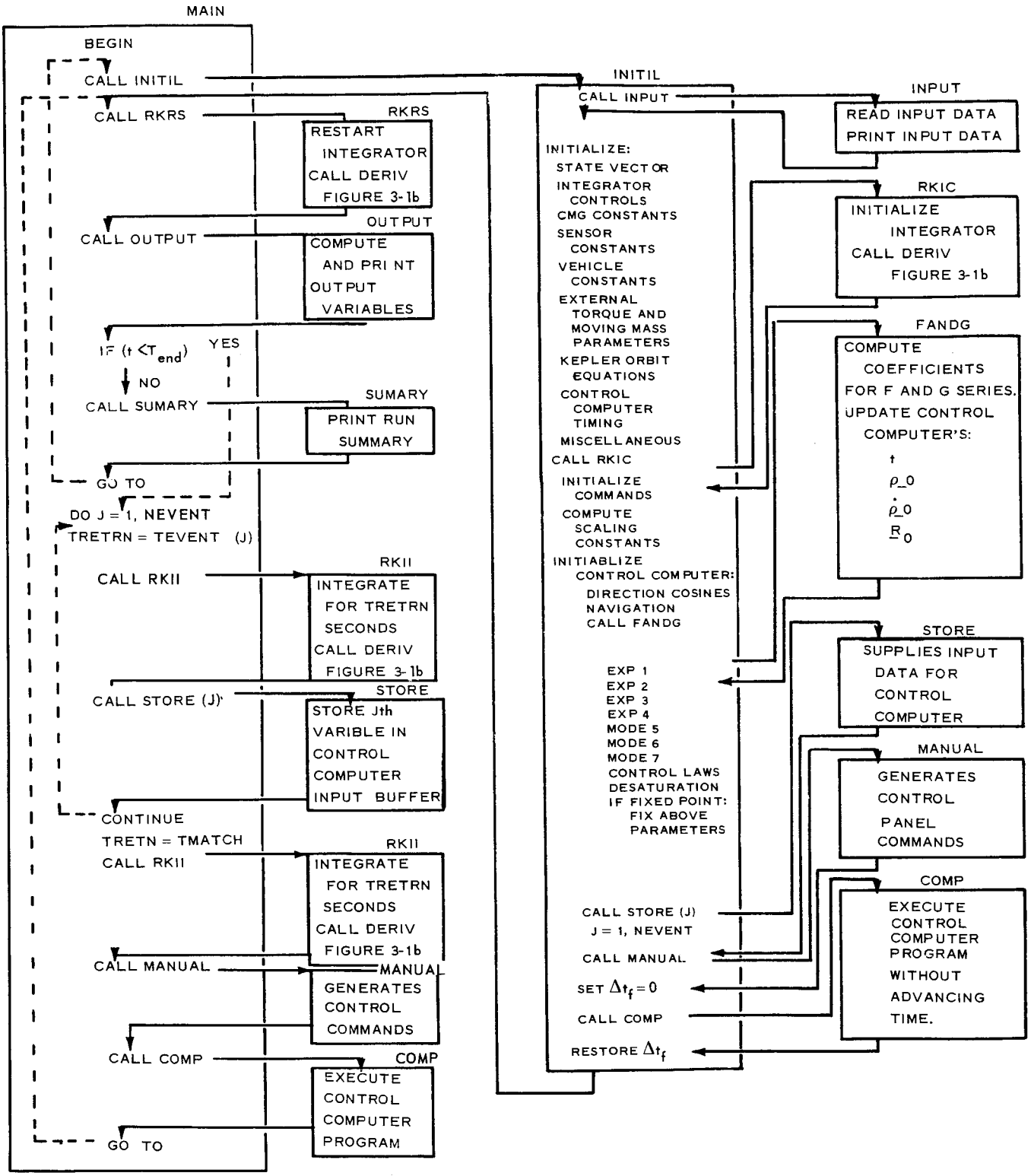


Figure 3-1. Subroutine Linkage (Sheet 1 of 2)

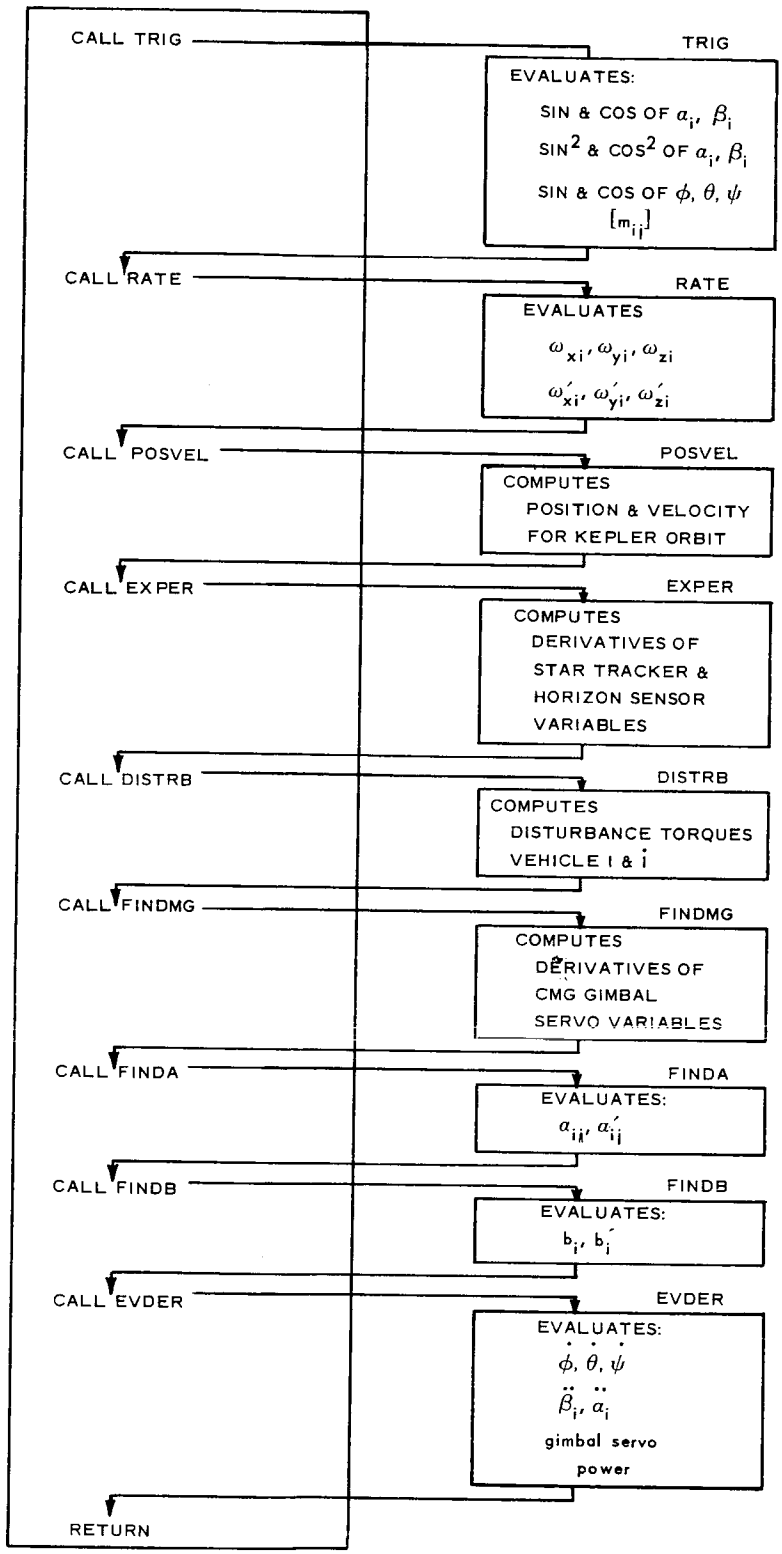


Figure 3-1. Subroutine Linkage (Sheet 2 of 2)

MAIN Program

Called by. — Automatic branch to MAIN after program is loaded on host computer.

Calls. — INITIL, COMP, RKRS, RKII, OUTPUT, MANUAL, SUMARY, STORE.

Function. — Overall program control and timing.

Comments. — The array TEVENT contains the successive time increments, over which integration must be performed to reach control computer A/D sampling times. Note that

$$\text{TCYCLE} = \text{TMATCH} + \sum_{J=1}^{\text{NEVENT}} \text{TEVENT(J)}$$

A computation cycle timing chart is illustrated in Figure 3-2.

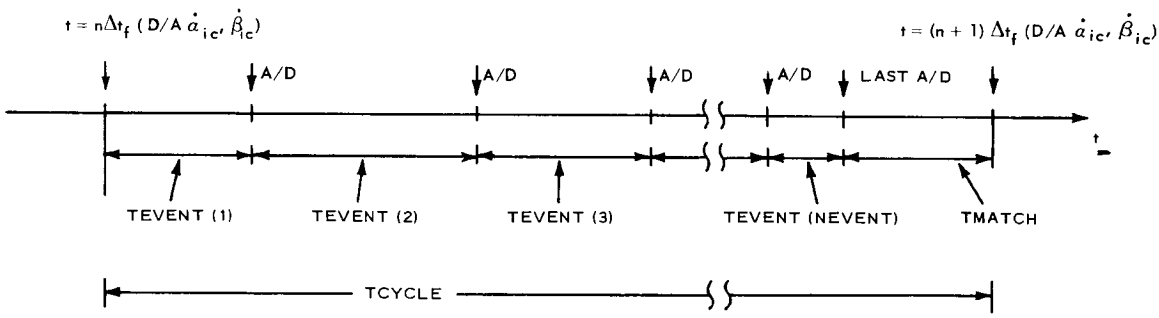


Figure 3-2. Computation Cycle Timing

```
$IBFTC MAIN    LIST
C
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
C
1     CALL INITIL
5     CALL RKRS
      CALL OUTPUT
6     IF(T.GE.TEND)GO TO 4
      DO 3   J=1,NEVENT
      TRETRN=TEVENT(J)
      CALL RKII
3     CALL STORE(J)
      TRETRN=TMATCH
      CALL RKII
      CALL MANUAL
      CALL COMP
      GO TO 5
4     CALL SUMARY
      GO TO 1
      END
```

Called by. — MAIN

Calls. — INPUT, ENERGY, RKIC, FANDG, UPDAT3, STORE, MANUAL, COMP

Function. — Initializes routines in accordance with input data specifications. Performs one time computations such as units conversion and evaluation of constants.

Comments — Because this routine exceeded table space during compilation, it was necessary to divide it into four smaller routines. Since these routines are never used independently, it is convenient to include all four in the verbal label INITIL. Figure 3-3 shows this breakdown.

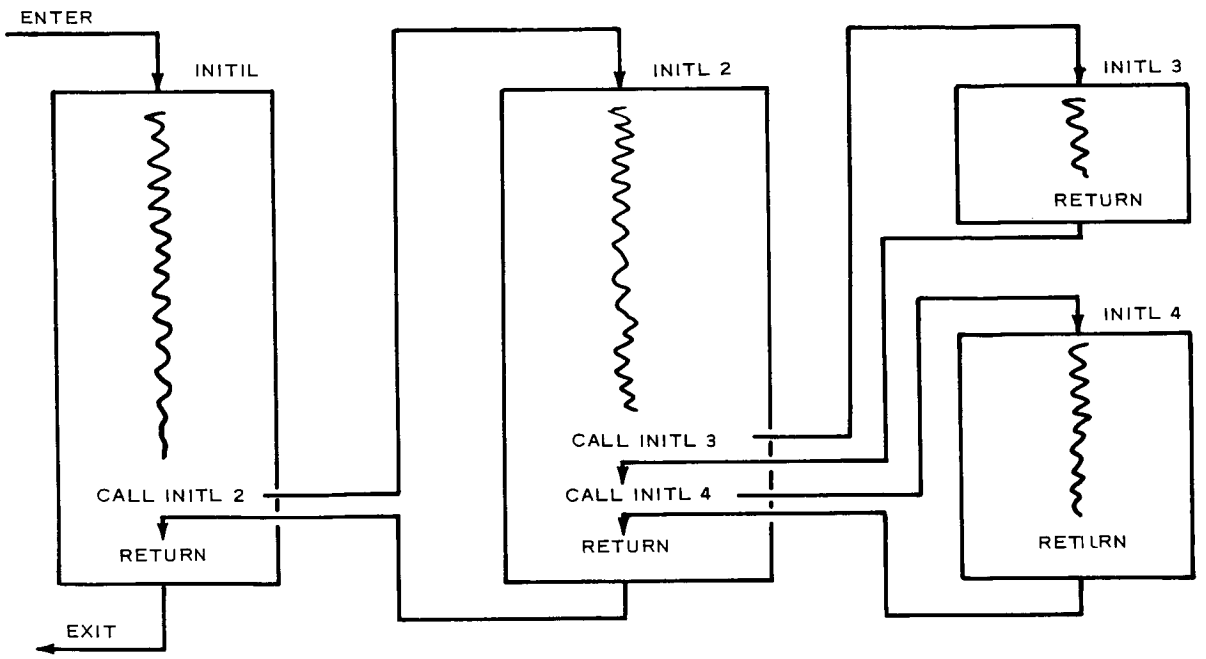


Figure 3-3. Initialization Routines

```
$IBFTC S1P1    LIST
      SUBROUTINE INITIL
      COMMON/RKYV/OMEGAB(79)
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/AGROUP/A(9,9),AP(31)
      COMMON/CGYRO/AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AB AG,BB BG,
     1 CB BG,AA BA,AA CA,BA CA,BB CB,DIF1,DIF2,GRAJMA,SUM1,SUM2,SUM3,
     2 SUM4,SUM5,SUM6,SUM7,SUM8,SUM9,SUM10,AGOMGO(3),AGOMGD(3),AGOMGM,
     3 HNOM
      REAL JMBETA,JMALPH
      DIMENSION GYROI(10)
      EQUIVALENCE (AA,GYROI(1))
      COMMON/CBODY/INERT(21)
      REAL INERT
      COMMON/CSERVO/BINPUT(3),AINPUT(3),K2BETA,K2ALPH,TAUDBE,TAUDAL,
     1 KBETA,KALPH,KSFBE,KSFAL,TAUNBE,TAUNAL,TAUBE,TAUAL,ELIMBE,ELIMAL,
     2 KTBE,KTAL,KBBETA,KBALPH,TAUMBE,TAUMAL,TFBETA,TFALPH,BLIMIT,ALIMIT
     3 ,GRBETA,GRALPH
      REAL K2BETA,K2ALPH,KBETA,KALPH,KSFBE,KSFAL,KTBE,KTAL,KBBETA,KBALPH
      COMMON/TORQUE/MJET(22)
      REAL MJET
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,EVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE(23)
      REAL MU
      COMMON/SENSOR/ESTAR(20),STKR,JINRT(4),ETA(38)
      REAL JINRT
      COMMON/DIST/MDIST(3),MDNOM(3),SPHASE(3,6),CPHASE(3,6),SFREQ(6),
     1 CFREQ(6),MDAMP(3,6),FREQ(6),TQEMM(3)
      REAL MDIST,MDNOM,MDAMP
      COMMON/MOVE/QMASS(41)
      COMMON/BUFFIN/XZ(339),NZ(124),XY( 94),NY( 9)
C
C****  READ INPUT DATA
      CALL INPUT
C****  INITIALIZE STATE VECTOR AND INEGRATOR CONTROLS
```

```
      T=0.
      NINT=79
      DUM1=XZ(159)*DEGTOR
      DUM2=XZ(160)*DEGTOR
      DO 1 J=1,18
      OMEGAB(J)=XZ(J)*DEGTOR
      U(J)=XZ(J+79)*DUM1
1     UMIN(J)=XZ(J+79)*DUM2
      DO 2 J=19,45
      OMEGAB(J)=XZ(J)
      U(J)=XZ(J+79)*XZ(159)
2     UMIN(J)=XZ(J+79)*XZ(160)
      DO 20 J=46,47
      OMEGAB(J)=XZ(J)*DEGTOR
      U(J)=XZ(J+79)*DUM1
20    UMIN(J)=XZ(J+79)*DUM2
      DO 21 J=48,71
      OMEGAB(J)=XZ(J)
      U(J)=XZ(J+79)*XZ(159)
21    UMIN(J)=XZ(J+79)*XZ(160)
      DO 3 J=72,79
      OMEGAB(J)=XZ(J)*DEGTOR
      U(J)=XZ(J+79)*DUM1
3     UMIN(J)=XZ(J+79)*DUM2
      K=1
      DO 4 J=1,79
      IF(NZ(J)    .EQ.0)GO TO 4
      NOINT(K)=J
      K=K+1
4     CONTINUE
      NOINT(K)=0
      DTMIN=XZ(161)
      DTEST=XZ(162)
      NDOUBL=NZ(80)
C****   GYRO DATA INITIALIZATION
      DO 5 J=1,10
5     GYROI(J)=XZ(J+162)
      DO 6 J=1,8
```

```
6       AGOMGO(J)=X7(J+172)
        DO 7 J=1,26
7       BINPUT(J+6)=XZ(J+180)
        BLIMIT=BLIMIT*DEGTOR
        ALIMIT=ALIMIT*DEGTOR
        K2BETA=K2BETA*GRBETA
        K2ALPH=K2ALPH*GRALPH
        KTBE=KTBE*GRBETA
        KTAL=KTAL*GRALPH
        KBBETA=KBBETA*GRBETA
        KBALPH=KBALPH*GRALPH
        TFBETA=TFBETA*GRBETA
        TFALPH=TFALPH*GRALPH
        AB AG=AB+AG
        BB BG=BB+BG
        CB BG=CB+BG
        AA BA=AA-BA
        AA CA=AA-CA
        BA CA=BA-CA
        BB CB=BB-CB
        DIF1=AB AG-BB BG
        DIF2=AB AG-CB BG
        SUM1=CB BG+(GRBETA**2)*JMBETA
        SUM2=BA+(GRALPH**2)*JMALPH
        SUM3=CB BG+GRBETA*JMBETA
        SUM4=BA+GRALPH*JMALPH
        SUM5=CA+CB BG+JMBETA
        SUM6=BA+GRALPH
        SUM7=CA-(GRBETA-1.)*JMBETA
        SUM8=CA+(CB BG*((GRBETA-1.)**2)*JMBETA)/SUM1
        SUM9=((GRALPH-1.)**2)*JMALPH
        SUM10=JMALPH*GRALPH*(GRALPH-1.)
        GRAJMA=JMALPH*GRALPH
        A(4,4)=SUM1
        A(6,6)=SUM1
        A(8,8)=SUM1
C****     SENSOR INITIALIZATION
        DO 8 J=1,4
```

```
8       ESTAR(J)=XZ(J+206)*DEGTOR
        DO 9 J=1,17
9       ESTAR(J+4)=XZ(J+210)
        STKR=STKR*XZ(228)
        DO 10 J=1,2
        JINRT(J)=XZ(229)
10      JINRT(J+2)=XZ(230)
        DO 11 J=1,7
11      ETA(J)=XZ(J+230)
        DO 18 J=1,5
18      ETA(J+7)=XZ(J+313)
        ETA(12)=.5E-3*SQRT(ETA(12)/ESTAR(5))
C****     VEHICLE INITIALIZATION
        DO 12 J=1,6
12      INERT(J+6)=XZ(J+237)
        DO 13 J=1,3
        ETA(J+4)=ETA(J+4)/3600.
13      MJET(J+12)=XZ(J+243)
C****     EXTERNAL TORQUES AND MOVING MASS
        NDUM=0
        DO 15 J=1,3
        MJET(J+15)=0.
        MJET(J+18)=XZ(J+336)*XZ(319)
        FREQ(J)=XZ(J+285)
        FREQ(J+3)=XZ(J+288)
        MDNOM(J)=XZ(J+246)
        DO 15 K=1,6
        NDUM=NDUM+1
        MDAMP(J,K)=XZ(NDUM+249)
        SPHASE(J,K)=SIN(XZ(NDUM+276)*DEGTOR)
15      CPHASE(J,K)=COS(XZ(NDUM+276)*DEGTOR)
        QMASS(1)=XZ(292)
        QMASS(2)=2.*QMASS(1)
        DO 16 J=1,12
16      QMASS(J+11)=XZ(J+292)
        DO 17 J=1,3
        QMASS(J+29)=XZ(J+304)
        QMASS(J+23)=QMASS(J+20)*QMASS(J+29)
```

```
      QMASS(J+26)=QMASS(J+23)*QMASS(J+29)
17    QMASS(J+32)=.5*QMASS(J+17)
C****   KEPLER ORBIT INITIALIZATION
      DUM=0.
      DO 100 J=1,3
100   DUM=DUM+XZ(J+307)*XZ(J+307)
      DUM=SQRT(DUM)
      DUM1=0.
      DO 101 J=1,3
      POSO(J)=XZ(J+307)/DUM
101   DUM1=DUM1+       POSO(J)*XZ(J+310)
      DUM2=0
      DO 102 J=1,3
      VELO(J)=      -DUM1*POSO(J)+XZ(J+310)
102   DUM2=DUM2+VELO(J)*VELO(J)
      DUM2=SQRT(DUM2)
      DUM=DUM*DUM2
      DUM1=DUM/MU
      ECCENT=0.
      CETAO=0.
      SETAO=0.
      DO 103 J=1,3
      VELO(J)=VELO(J)/DUM2
      FVECT(J)=DUM1*XZ(J+310)-VELO(J)
      ECCENT=ECCENT+FVECT(J)*FVECT(J)
      CETAO=CETAO+FVECT(J)*VELO(J)
103   SETAO=SETAO+FVECT(J)*POSO(J)
      PVCON(4)=SQRT(1.-ECCENT)
      ECCENT=SQRT(ECCENT)
      CETAO=CETAO/ECCENT
      SETAO=SETAO/ECCENT
      PVCON(3)=MU**2/((DUM/PVCON(4))**3)
      PVCON(2)=MU/DUM
      PVCON(1)=DUM*DUM/MU
      DUM1=ATAN2(PVCON(4)*SETAO,ECCENT+CETAO)
      MEANAO=DUM1-ECCENT*SIN(DUM1)
      MEANA=MEANAO
      ENOW=DUM1
```

```
      DO 104 J=1,3
      POS(J)=XZ(J+307)
104   VEL(J)=XZ(J+310)
      CALL INITL2
      RETURN
      END
```

```
202     IF(TEVENT(K).LT.TEVENT(MINT))MINT=K
        NORDER(J)=MINT
201     TEVENT(J)=DUMMY
        I=1
        K=1
        DO 203 J=2,15
        L1=NORDER(J-1)
        L2=NORDER(J)
        IF(EVENTT(L2).EQ.EVENTT(L1))GO TO 204
        NUMBER(I)=K
        K=1
        I=I+1
        GO TO 203
204     K=K+1
203     CONTINUE
        NUMBER(I)=K
        NEVENT=I
        L1=NORDER(1)
        TEVENT(1) =EVENTT(L1)
        DUMDBL=TEVENT(1)
        IF(NEVENT.EQ.1)GO TO 206
        K=1
        DO 205 J=2,NEVENT
        K=K+NUMBER(J-1)
        L1=NORDER(K)
        TEVENT(J)=DBLE(EVENTT(L1))-DUMDBL
205     DUMDBL=DUMDBL+TEVENT(J)
206     TMATCH=DBLE(TCYCLE)-DUMDBL
        IF(TMATCH.LT.0.)TMATCH=0.
C****    FINAL INITIALIZATION OF CONTINUOUS EQUATIONS
        DO 19 J=1,19
19      BETADC(J)=0.
        DO 18 J=1,37
18      MODE(J)=0
        MODE(26)=1
        CALL ENERGY(NDUM)
        CALL RKIC
        DO 20 J=1,307
```

```
$IBFTC S1P2    LIST
      SUBROUTINE INITL2
      COMMON/SINCO/SINB(30),DIRCO(3,3)
      COMMON/POWER/KDT(307)
      COMMON/PVDATA/POS(28)
      COMMON/CONSTS/WEARTH(27)
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1 ,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTIN/MODE(37)
      COMMON/FLOOUT/BETADC(19)
      COMMON/QUANT/NBIN(42)
      COMMON/FIXOUT/AC(19)
      COMMON/EXP1V/ACDBL(34),WE(4)
      COMMON/TVECT/H(12),NPASS,NSLOW,XNSLOW
      COMMON/DIRCOS/MS(3,3),MSDOT(3,3),ML(3,3),MLDOT(3,3),DSAVE(3,3),
     CMSDBL(3,3),MLDBL(3,3),XW(3),WPREV(3),TRATIO(3),DELW(3)
      REAL MS,MSDOT,ML,MLDOT,MSDBL,MLDBL
      COMMON/BUFFIN/XZ(339),NZ(124),XY( 94),NY( 9)
      DOUBLE PRECISION DUMDBL
C
C****   COMPUTER SPEC. INITIALIZATION
      TCYCLE=XZ(319)
      NSLOW=NZ(124)
      NPASS =NSLOW
      XNSLOW=NSLOW
      DUMMY=1.+TCYCLE
      DO 200 J=1,15
      TEVENT(J)=    XZ(J+319)
      IF(TEVENT(J).LT.0.)TEVENT(J)=0.
      IF(TEVENT(J).GT.TCYCLE)TEVENT(J)=TCYCLE
      EVENTT(J)=TEVENT(J)
200   NUMBER(J)=0
      DO 207 J=1,3
      TRATIO(J)=(TCYCLE-TEVENT(J))/TCYCLE
207   IF(NY(9).EQ.0)TRATIO(J)=0.
      DO 201 J=1,15
      MINT=1
      DO 202 K=2,15
```

```
20      KDT(J)=0
C****     INITIALIZE ON-BOARD COMPUTER CONTROLS
        TEND=XZ(336)
        DO 21 J=1,5
21      MODE(J+25)=NY(J)
        MODE(31)=NY(8)
        NBIN(42)=NZ(123)
        IF(NZ(123).NE.0)CALL INITL3
        NCOST1=1
        NMAN1=1
        NPRINT=(XZ(335))/TCYCLE +.5
        NPRCTL=NPRINT
        LNECNT=100
C****     DIRECTION COSINE INITIALIZATION
400     DO 401 J=1,3
        DO 401 K=1,3
        MS(J,K)=DIRCO(J,K)
401     MSDBL(J,K)=DIRCO(J,K)
C****     POSITION AND VELOCITY INITIALIZATION (PLUS EXP. 1  TARGET POS.)
        DO 402 J=1,3
402     POS(J+25)=XY(J)
        CALL FANDG
C****     EXPERIMENT ONE  (EARTH MAPPING)
        WE(1)=WEARTH(27)
        DO  410  J=2,4
410     WE(J)=WE(1)*WE(J-1)
        ACDBL(9)=0.
        ACDBL(10)=WE(1)
        ACDBL(11)=0.
        NDUM=0
        DO 411 J=1,3
        DO 411 K=1,3
        NDUM=NDUM+1
        MLDBL(J,K)=XY(NDUM+4)
411     ML(J,K)=MLDBL(J,K)
        AC(7)=XY(4)*WEARTH(2)
        ACDBL(1)=AC(7)
        CALL INITL4
```

```
$IBFTC S1P3    LIST
      SUBROUTINE INITL3
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VEO,FIXWO(10),WEARTH
      REAL MU
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/SCALER/CSCALE,DSCALE,LSCALE,LLSCAL,MSCALE,NSCALE,PSCALE,
     CQSCALE,RSCALE,TSCALE,VSCALE
      INTEGER CSCALE,DSCALE,PSCALE,QSCALE,RSCALE,TSCALE,VSCALE
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NP0,FL2NP1
      DIMENSION NM7P9(17),FNM7P9(17)
      EQUIVALENCE (NM7P9(1),NM7),(FNM7P9(1),FLNM7)
      COMMON/MISCEL/FS,DBLFS,NBIT  ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/QUANT/NBIN(42)
      COMMON/EXP3V/ANGLE(11),KC,PREV(25)
      COMMON/SINCOS/ASC,B,C,HALFPI,PI,HAFPI
      INTEGER ASC,B,C,HALFPI,PI,HAFPI
      COMMON/ASINC/AAS,BAS,CAS,DAS
      INTEGER AAS,BAS,CAS,DAS
      COMMON/ATANC/AAT,BAT,CAT,DAT,QUARPI
      INTEGER AAT,BAT,CAT,DAT,QUARPI
      COMMON/BUFFIN/XZ(339),NZ(124),XY( 94),NY( 9)
C
C****   COMPUTE SCALING CONSTANTS
      NBIT=NZ(81)
      DO 18 J=1,41
      NBIN(J)=  2**(NZ(81)-NZ(J+81))
18    IF(NBIN(J).LT.1)NBIN(J)=1
      NM7=2**(NZ(81)-7)
      FLNM7=NM7
      DO 301 J=2,17
      NM7P9(J)=2*NM7P9(J-1)
301   FNM7P9(J)=NM7P9(J)
      XNH=ALOG(XZ(319))/.69314
```

```
      NH=XNH
      DUM=NH
      IF(DUM.LT.XNH)NH=NH+1
      IF(NH.GT.5)CALL EXIT
      FLNM8=FLNM7/2.
      FLNM10=FLNM8/4.
      FLNM11=FLNM8/8.
      FLNP12=FLNP9*8.
      FL2NP0=FLNP0*FLNP0
      FL2NP1=FL2NP0*2.
      FL2NM1=FL2NP0/2.
      FL2NM2=FL2NM1/2.
      F2NM10=FL2NP0/1024.
      F2NM15=F2NM10/32.
      F2NM25=F2NM15/1024.
      FS=NP0-1
      HALFFS=NM1
      DBLFS=NP0*NP0-1
C****   SPECIAL SCALING CONSTANTS
      CSCALE=2**(5-NH)
      DSCALE=2*CSCALE
      LLSCAL=2*DSCALE
      NDUM=NZ(81)-11
      LSCALE=2**NDUM
      MSCALE=2**(-NDUM)
      NDUM=NDUM-1
      PSCALE=2**NDUM
      NSCALE=2**(-NDUM)
      NDUM=NDUM-2
      TSCALE=2**NDUM
      VSCALE=2**(-NDUM)
      NDUM=NDUM-1
      RSCALE=2**NDUM
      QSCALE=2**(-NDUM )
C****   FIXED POINT CONSTANTS
      PI=PIE*FLNM2
      HALFPI=PI
      HAFPI=PI/2
```

```
QUARPI=PI
ASC=-.166656*FLNP2
B=   .83119E-2*FLNP6
C= -.184882E-3*FLNP12
AAT= .999215*FLNP0
BAT= -.3211819*FLNP1
CAT= .1462766*FLNP2
DAT= -.389929E-1*FLNP4
AAS=  1.5707288E0*FLNM1
BAS= -.2121144*FLNP2
CAS=  .074261*FLNP3
DAS= -.0187293*FLNP5
KC=   .003373*FLNP8
RETURN
END
```

```
$IBFTC S1P4     LIST
      SUBROUTINE INITL4
      COMMON/CONSTS/RTODEG,DEGTOR,RE(24),WEARTH
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
     1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKP,
     2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DFLW(3),STBIAS(4),STSIG,
     3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),FLBORE(2),AZBORE(2),QST(4)
     4 ,WNG(6)
      REAL JINRT
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NP0,FL2NP1
      COMMON/MISCEL/FS,DBLFS,NBIT   ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/QUANT/NBIN(25),NBOUT(16),NEXPNT
      COMMON/FIXOUT/BDOTC(3),ADOTC(3),AC,ED(3),EP(3),WC(3),NJET(3)
      COMMON/EXP1V/ACDBL,ACDOT,ADSAVE,COSAC,COSWT,        ZOMEGA(3),
     COMEGAE(3),RO(3),S,SDOT,SDUM,SINAC,SINWT,SPRIME,SREL(3),SSQ,TANAC,
     CV(3),VC(3),VDOUB(3),WE,WE2,WE3,WE4
      COMMON/EXP2V/DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,
     CU2X,U1Z,U2Z
      COMMON/EXP3V/ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,XXKC,PREV,PSP(4),   SINDUM,SINL,SINLR,SIN2LR,SINTH,
     CSPX(2),SPY(2),SPZ(2),S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,
     CZEYI,ZEZI,EPZDOT
      COMMON/MOD567/DELE(3),ECOM(3),EDCOM(3),TENDM,ECOMDP(3),DELTAE(3),
     CMAX,MAXRT ,TP,NHHH
      REAL MAXRT
      COMMON/CONTL1/EPPREV(3),GAIN(6,5),GAINP(6,5),EPP(3),HNOMP,KLCL(6),
     C    MAGA(3),MAGB(3),MCA(3),MCB(3),TRQC(3),TRQCP(3),UNITVA(3,3),
     CUNITVB(3,3),ZO(5)
      COMMON/CONTL2/DELA(3),DELB(3),DOT1(3),DOT2(3),DOT3(3),XXX1,XXX2,
     CKSAVE,MAGASQ(3),MAGBSQ(3),TREM(3),TRQPRD(3),UNITVH(3,3)
```

```
      C  ,BHOLD,NBSELF,NBDTDS,MBMAX
       REAL KSAVE
       COMMON/CONTL3/CGYRO(3),COSTA(3),COSTB(3),RTEST,RUSE,TDESSQ,
      CTDOTA(3),TDOTB(3),TREMSQ,XKEND,ITER,NOITER
       COMMON/DESAT/ERRLIM(3),JETCT(3),TJCNT(3),GIMLIM(3),BDOTMX,ADOTMX
       INTEGER TJCNT
       COMMON/TVECT/DELT,H,TIME(10),NPASS,NSLOW,XNSLOW
       COMMON/DIRCOS/MS(3,3),MSDOT(3,3),ML(3,3),MLDOT(3,3),DSAVE(3,3),
      CMSDBL(3,3),MLDBL(3,3),XW(3),WPREV(3),TRATIO(3),XDELW(3)
       REAL MS,MSDOT,ML,MLDOT,MSDBL,MLDBL
       COMMON/BUFFIN/XZ(339),NZ(124),XY( 94),NY( 9)
       DIMENSION XXWE(4)
       DIMENSION COSST(4),SINST(4),XXS1X(6),XDEL(5)
       INTEGER NMS(3,3),NMSDBL(3,3),NNWE(4),NML(3,3),NMLDBL(3,3),NNS1X(6)
      1 ,NDEL(5),NNZO(5),NGAIN(6,5),NGAINP(6,5),NERLIM(3),FIXDT,NTRATO(3)
       EQUIVALENCE (XXWE(1),WE,NNWE(1))
       EQUIVALENCE (NHH,H),(NMS(1,1),MS(1,1)),(NMSDBL(1,1),MSDBL(1,1)),
      1 (NOMGAE,OMEGAE(2)),(NML(1,1),ML(1,1)),(NMLDBL(1,1),MLDBL(1,1)),
      2 (NACDBL,ACDBL),(NAC,AC),(NNS1X(1),S1X,XXS1X(1)),(NETADT,ETADOT),
      3 (NDEL(1),DEL,XDEL(1)),(NDANG,DELANG),(NWBAR,WBAR),(NNZO(1),ZO(1))
      4 ,(NGAIN(1,1),GAIN(1,1)),(NGAINP(1,1),GAINP(1,1)),(NHNOM,HNOMP),
      5 (NKSAVE,KSAVE),(NXKEND,XKEND),(NERLIM(1),ERRLIM(1)),(DELT,FIXDT),
      6 (MXBD,BDOTMX),(MXAD,ADOTMX),(NMAXRT,MAXRT),(NTRATO(1),TRATIO(1))
C
C****     EXPERIMENT TWO  (INERTIAL MODE)  OR    EXPERIMENT THREE
C                                                (HORIZON  SPECTROMETRY)
       SINDIF=SIN(ASTAR(1)-ASTAR(2))
       DO 420 J=1,4
       COSST(J)=COS(ESTAR(J))
420    SINST(J)=SIN(ESTAR(J))
       IF(NY(6).GE.3)GO TO 430
       DO 421 J=1,2
       SXI(J)=SINST(J)
       SYI(J)=-COSST(J)*SINST(J+2)
421    SZI(J)=COSST(J)*COSST(J+2)
       DEL=COSST(1)*COSST(1)*SINST(3)*COSST(2)*SINDIF
       DELP=-COSST(1)*COSST(2)*SINDIF
       DEL1=-SINST(1)*COSST(2)*SINST(4)
```

```
      DEL2=-DELP
      DEL3=SINST(1)*COSST(1)*SINST(3)
      GO TO 432
430   DO 431 J=1,2
      SXI(J)=-COSST(J)*SINST(J+2)
      SYI(J)=COSST(J)*COSST(J+2)
431   SZI(J)=-SINST(J)
      DEL=-COSST(1)*COSST(1)*COSST(2)*COSST(3)*SINDIF
      DELP=-COSST(1)*COSST(2)*SINDIF
      DEL1= DELP
      DEL2=-COSST(2)*COSST(4)*SINST(1)
      DEL3=COSST(1)*COSST(3)*SINST(1)
432   DO 433 J=1,6
      ECOM(J)=0.
433   XXS1X(J)=SXI(J)
      ETADOT=XY(14)*DEGTOR
      DELANG=ETADOT*TCYCLE*XNSLOW
      WBAR=XY(15)
      IF(NFXPNT.EQ.0)GO TO 434
      NETADT=ETADOT*FLNP5
      NDANG=DELANG*FL2NM2
434   CALL UPDAT3
C****    EXPERIMENT FOUR (MICROWAVE TRANSMISSION)
C
C              NO INITIALIZATION NEEDED
C
C****    MODE FIVE (ATTITUDE HOLD)
C
C              NO INITIALIZATION NEEDED
C
C****    MODE SIX  (MANUEVER)
C
      MAXRT =XY(16)*DEGTOR
C
C
C****    MODE SEVEN (MANUAL)
C
C             NO INITIALIZATION NEEDED
```

```
C
C****     BODY RATE SELECTION AND BASELINE CONTROL LAW
C
      DO 440 J=1,3
440   EPPREV(J)=0.
      NDUM=0
      DO 441 J=1,6
      DO 441 K=1,5
      NDUM=NDUM+1
      GAIN(J,K)=XY(NDUM+21)
      GAINP(J,K)=XY(NDUM+51)
441   ZO(K)=XY(K+16)
C****     CONTROL LAWS  2 AND 3  (HMCL AND ITERATIVE)
      HNOMP=XZ(180)
      KSAVE=XY(82)
      DO 450 J=1,3
      TRQPRD(J+12)=XY(J+91)*DEGTOR
      ADOTC(J)=0.
450   BDOTC(J)=0.
      XKEND=XY(83)*XY(83)
      ITER=NY(7)
C****     DESATURATION
      BDOTMX=XY(90)*DEGTOR
      ADOTMX=XY(91)*DEGTOR
      DO 460 J=1,3
      NJET(J)=0
      ERRLIM(J)=XY(J+83)*DEGTOR
      JETCT (J)=XY(J+86)/TCYCLE +.99
460   TJCNT(J)=0.
      DO 470 J=1,NEVENT
470   CALL STORE(J)
      CALL MANUAL
      IF(NFXPNT.NE.0)GO TO 500
      XXKC=.003373
      H=0.
      CALL COMP
      H=TCYCLE
      RETURN
```

```
C****    CONVERSION TO SCALED FIXPOINT
C
C
500     NHHH=TCYCLE*2.**(NBIT-NH)
        DO 501 J=1,3
        NTRATO(J)=TRATIO(J)*FLNPO
        DO 501 K=1,3
        NMSDBL(J,K)=MS(J,K)*FL2NPO
501     NMS(J,K)=NMSDBL(J,K)/NPO
        WE=WE*FL2NPO*8192.
        WE2=WE*WEARTH*16384.
        WE3=WE2*WEARTH*16384.
        WE4=WE3*WEARTH*8192.
        DO 510 J=1,4
510     NNWE(J)=XXWE(J)
        NOMGAE=NNWE(1)/NPO
        DO 511 J=1,3
        DO 511 K=1,3
        NMLDBL(J,K)=ML(J,K)*FL2NPO
511     NML(J,K)=NMLDBL(J,K)/NPO
        NACDBL=AC*FL2NPO
        NAC=NACDBL/NPO
        DO 533 J=1,5
        NDEL(J)=XDEL(J)*FLNPO
533     NNS1X(J)=XXS1X(J)*FLNPO
        NNS1X(6)=XXS1X(6)*FLNPO
        NWBAR=WBAR*FLNP5
        NMAXRT=MAXRT*FLNP5
        DUM=FLNM10/8.
        DO 541 K=1,5
        NNZO(K)=ZO(K)*FLNPO
        DO 541 J=1,6
        NGAIN(J,K)=GAIN(J,K)*DUM
541     NGAINP(J,K)=GAINP(J,K)*FLNM10
        NHNOM=HNOMP*FLNM11
        NKSAVE=KSAVE*FLNPO
        NXKEND=XKEND*FLNPO
        MXBD=BDOTMX*FLNP2
```

```
      MXAD=ADOTMX*FLNP2
      NBSELF=TRQPRD(14)*FLNM2
      NBDTDS=TRQPRD(15)*FLNP2
      DO 560 J=1,3
560   NERLIM(J)=ERRLIM(J)*FL2NP1
      NHH=0
      CALL COMP
      NHH=NHHH
      RETURN
      END
```

Subroutine COMP

Called by. — MAIN, INITIL

Calls. — DMULT, QUOTNT, SQRTCC, COST, CCATAN, CCASIN, SINCC, COSCC.

Function. — Simulates control computer program.

Comments. — This subroutine is programmed in two versions — fixed point and floating point. Due to compiler limitations, the fixed point version is split into two subroutines: COMP and COMPX.

Figure 3-4 is an overall block diagram of the control computer. Figures 3-5 through 3-16 are the math flow for the control computer program. For these math flows the following notation is defined for previous values of variables;

Subscript ($_p$) — Used when a storage location is allocated for the previous value. For example,

$$X_p = X$$

$$X = a + b$$

$$z = X + X_p \quad .$$

Superscript($\sim$) — Used to indicate the previous value of a variable in an equation solving for its new value. ($t = \tilde{t} + \Delta t$).

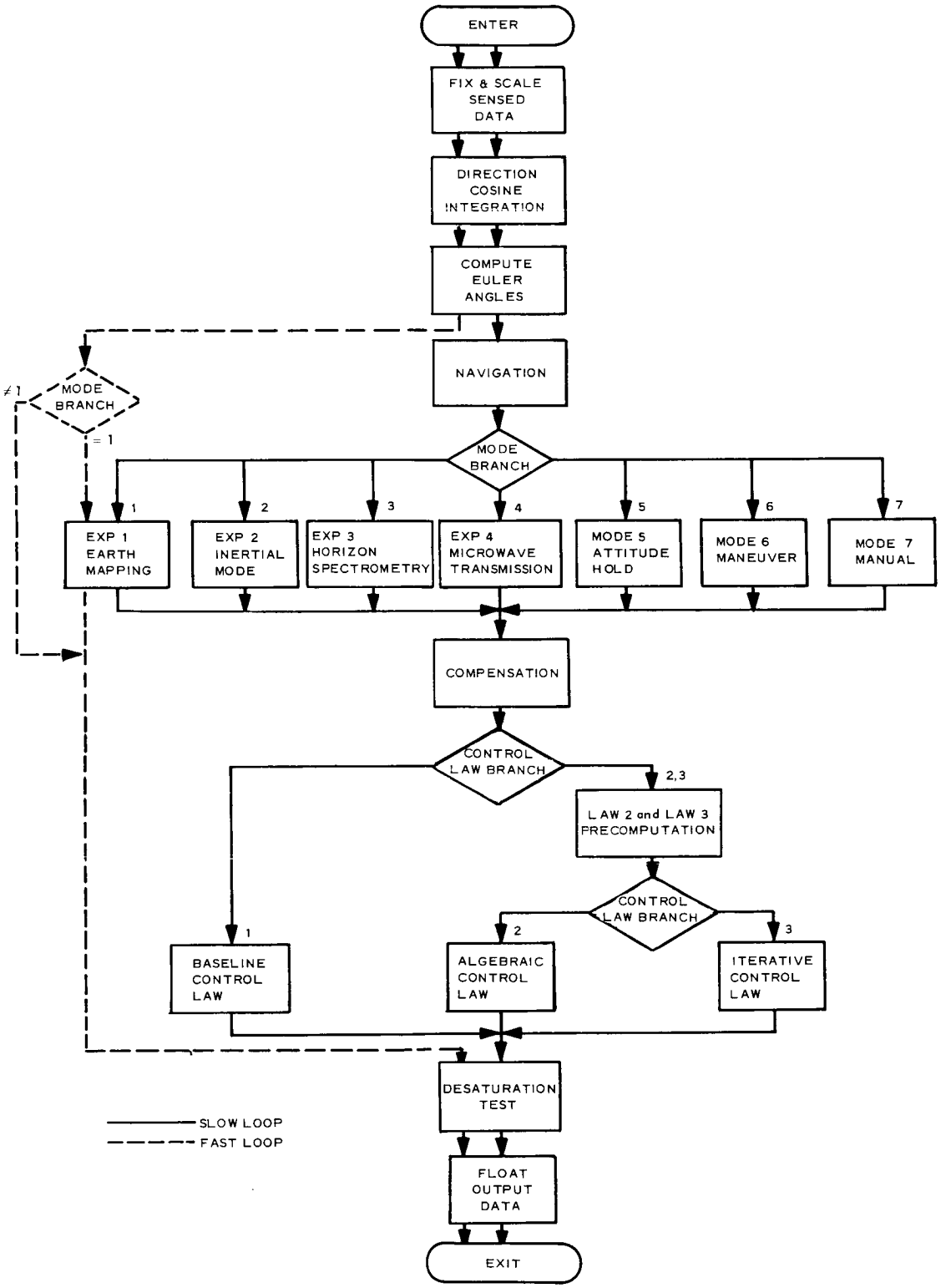


Figure 3-4. Simulated Control Computer

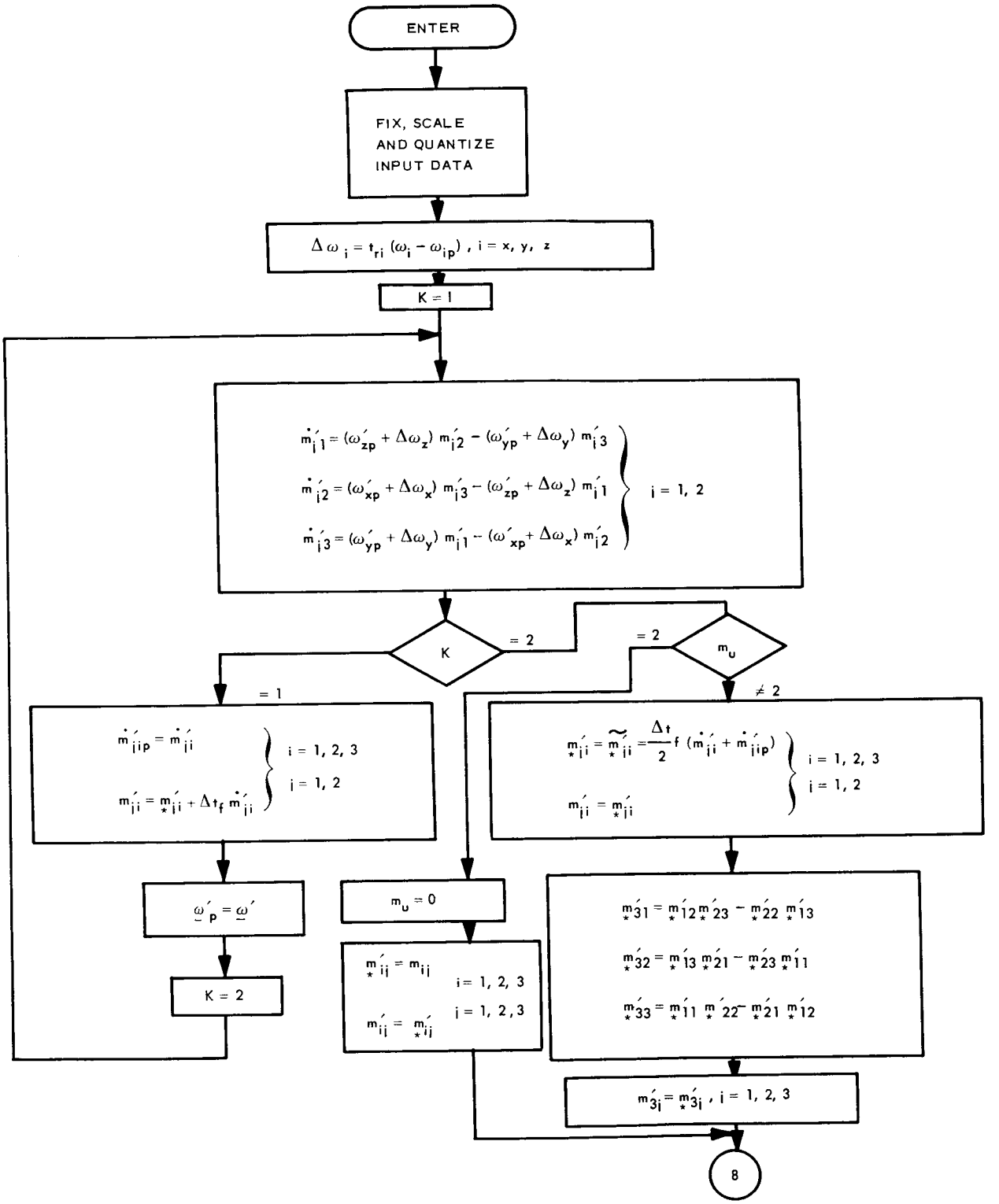


Figure 3-5. Control Computer Math Flow Direction Cosine Integration

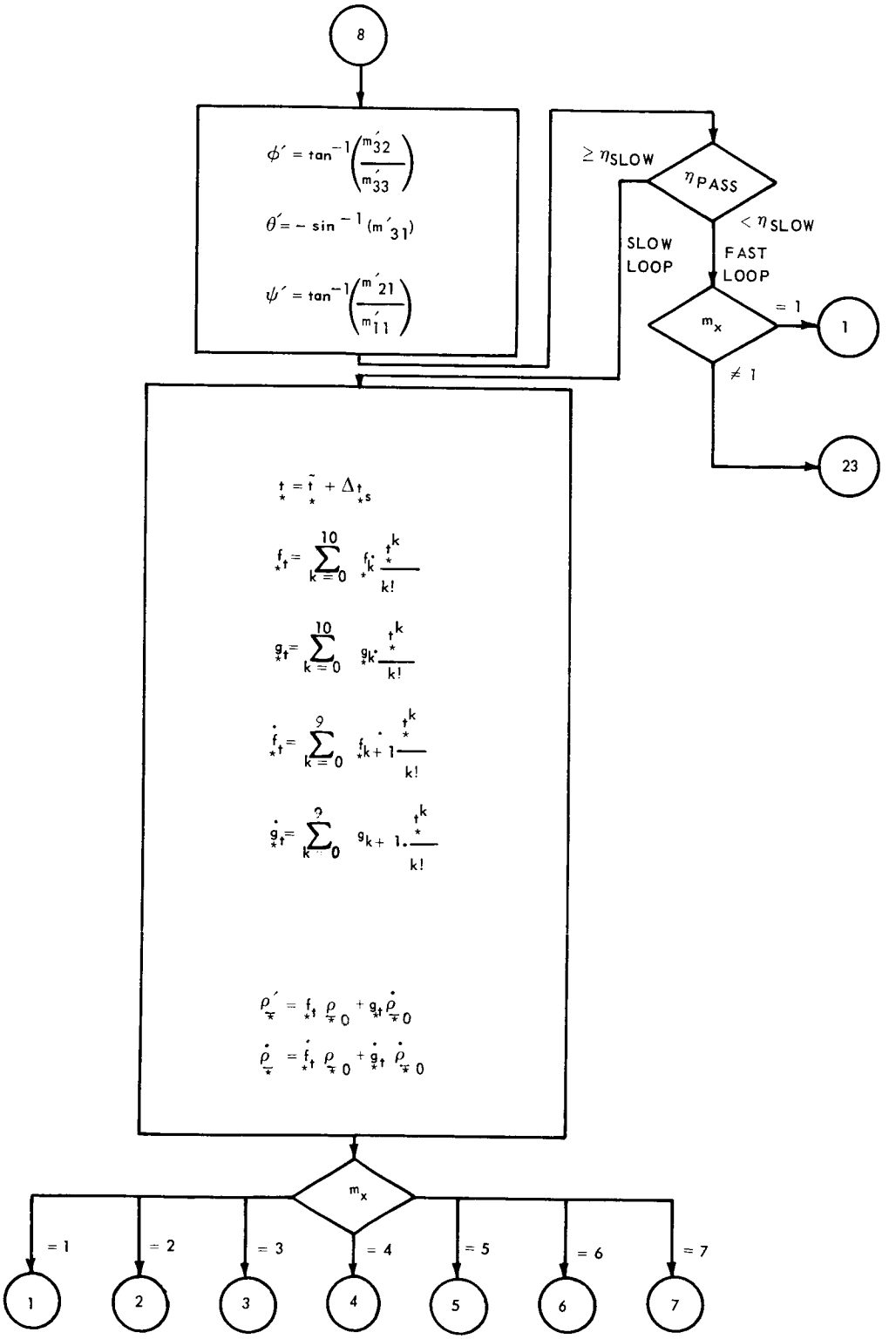


Figure 3-6. Control Computer Math Flow Compute Euler Angles, Navigation, Mode Branch

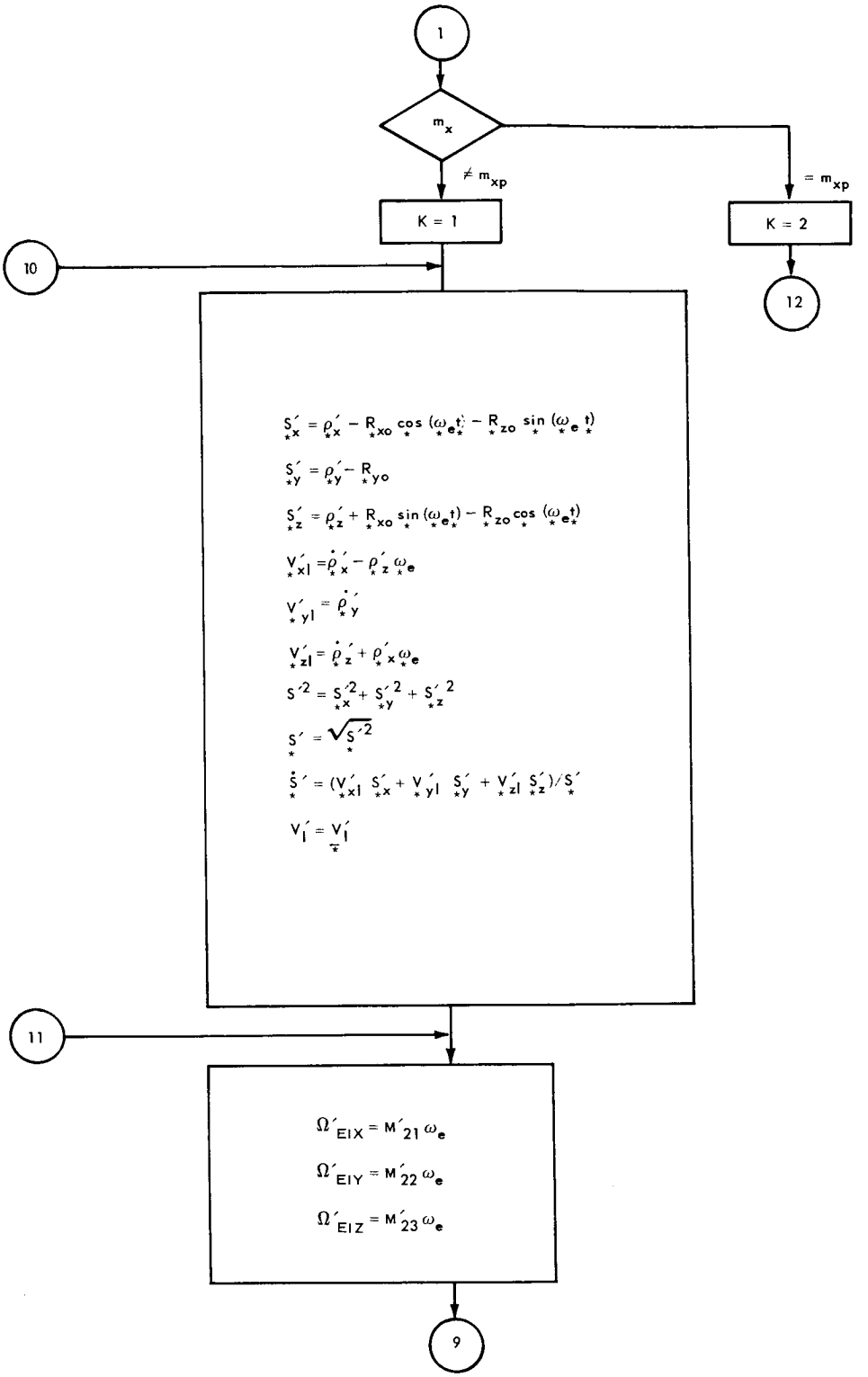


Figure 3-7. Control Computer Math Flow Experiment One, Earth Mapping (Sheet 1 of 2)

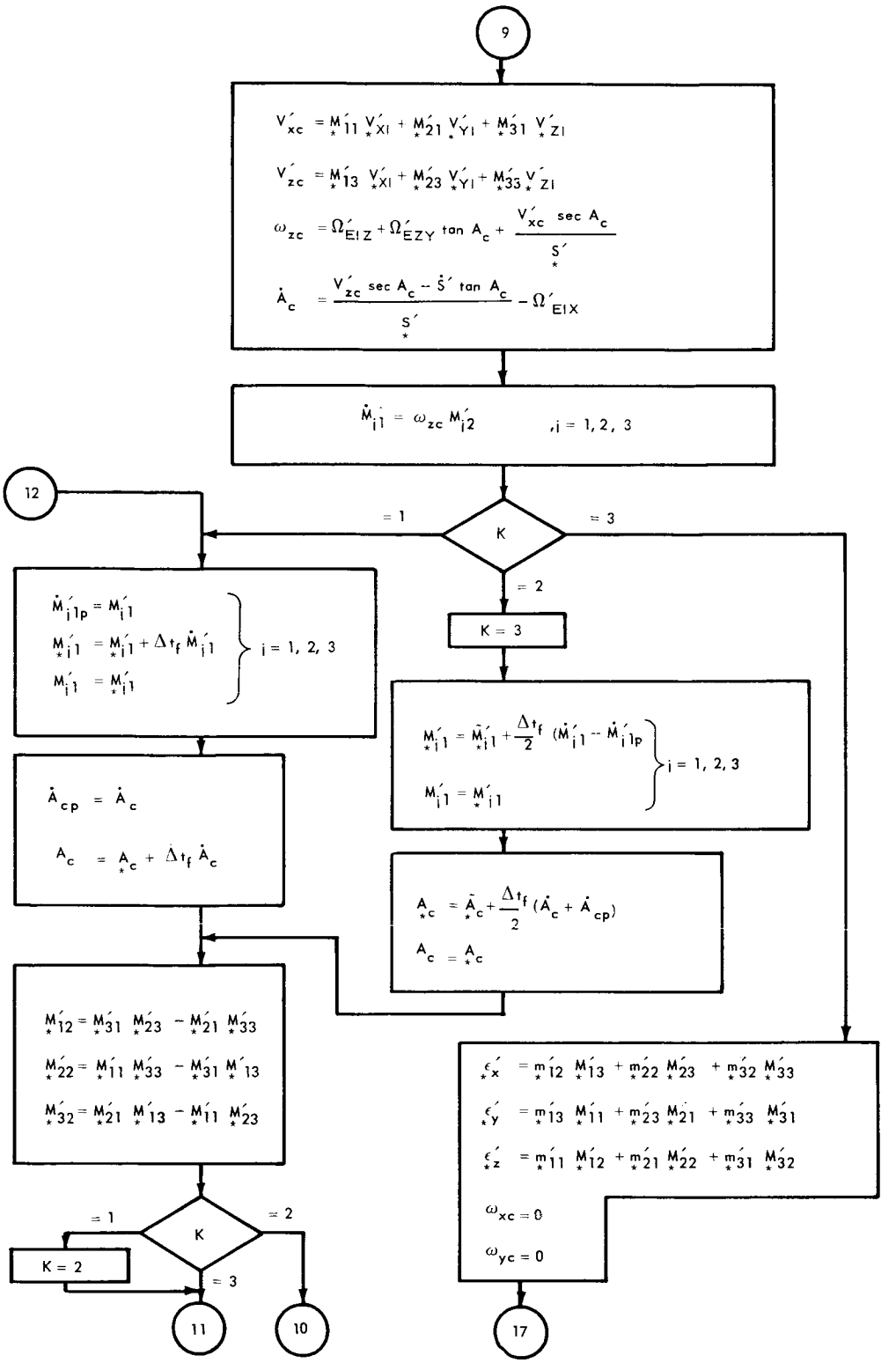


Figure 3-7. Control Computer Math Flow Experiment One, Earth Mapping (Sheet 2 of 2)

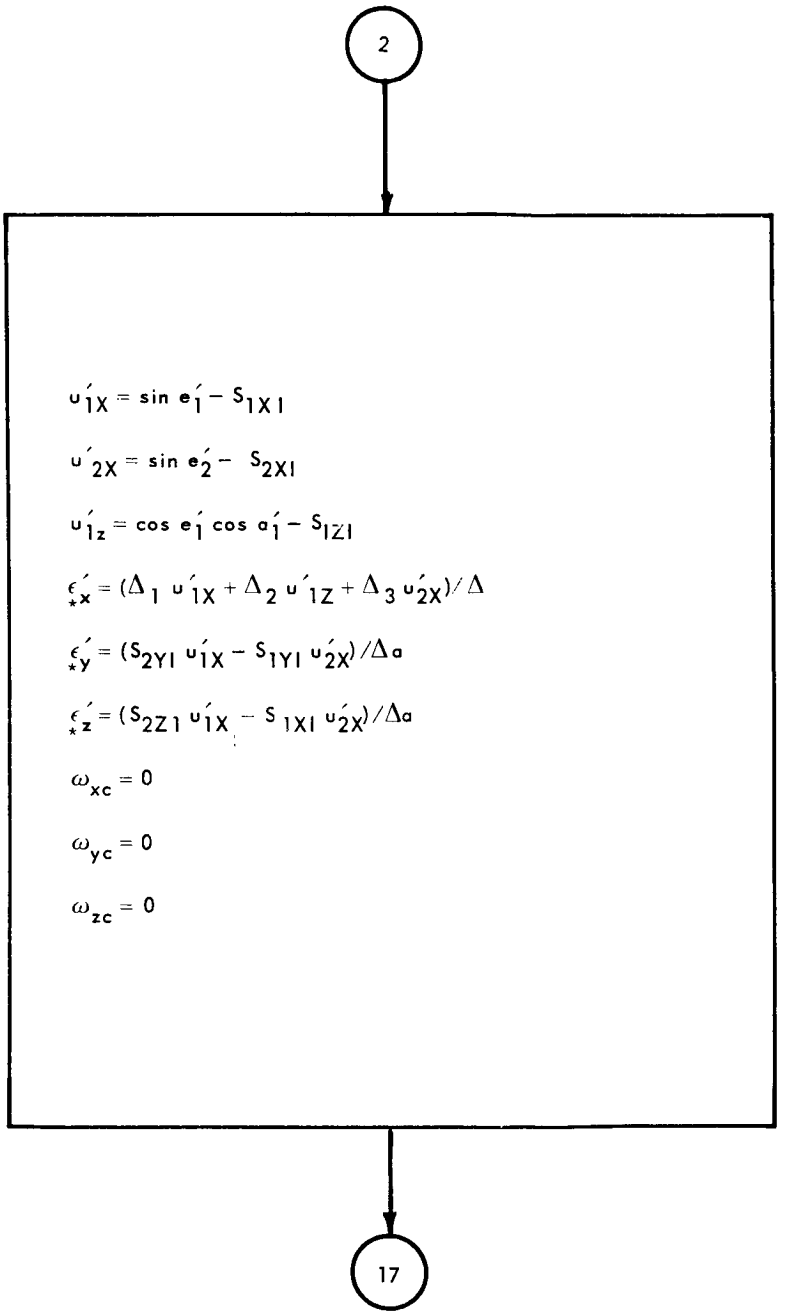


Figure 3-8. Control Computer Math Flow Experiment Two, Inertial Mode

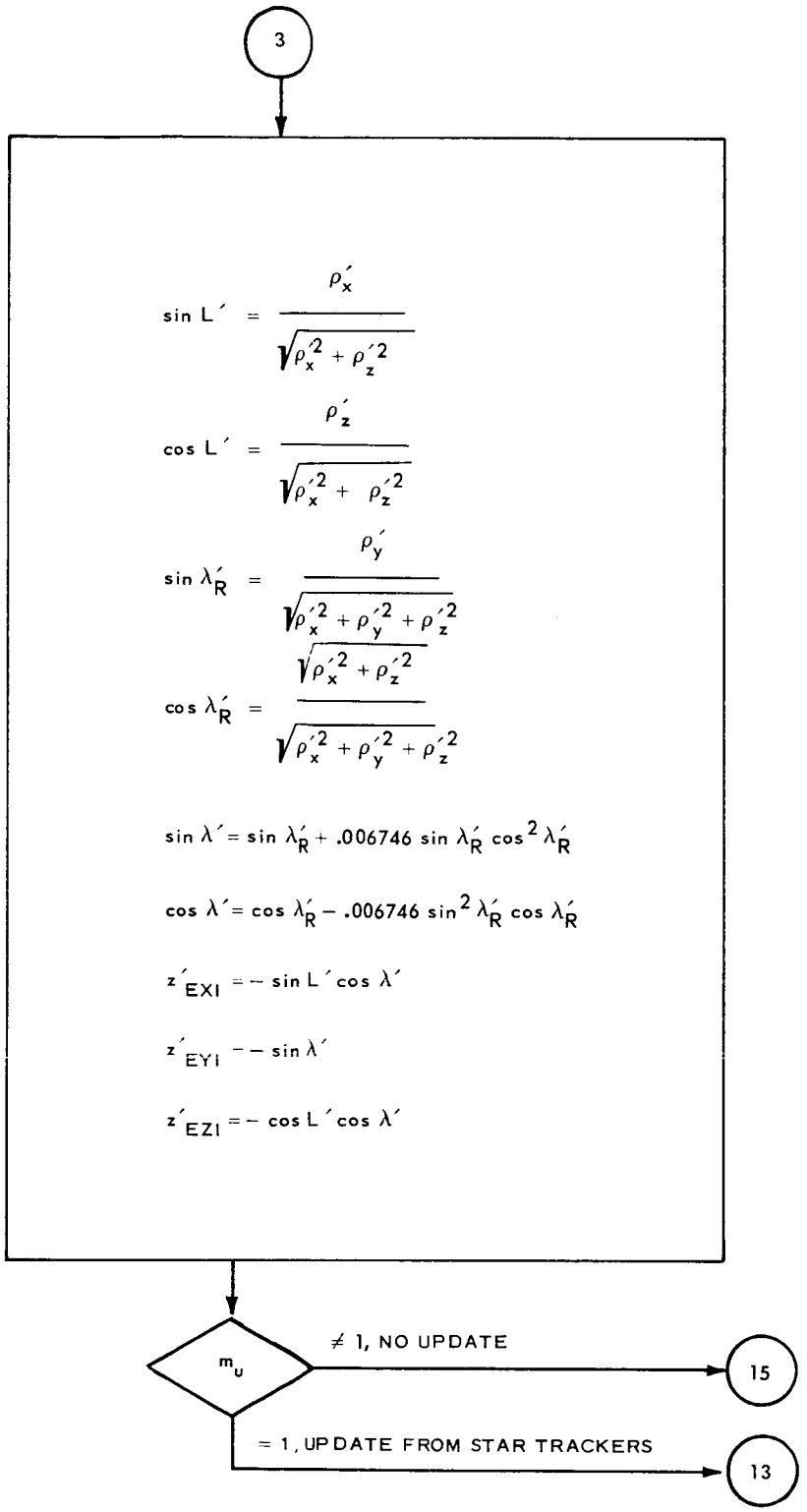


Figure 3-9. Control Computer Math Flow Experiment Three, Horizon Spectrometry (Sheet 1 of 3)

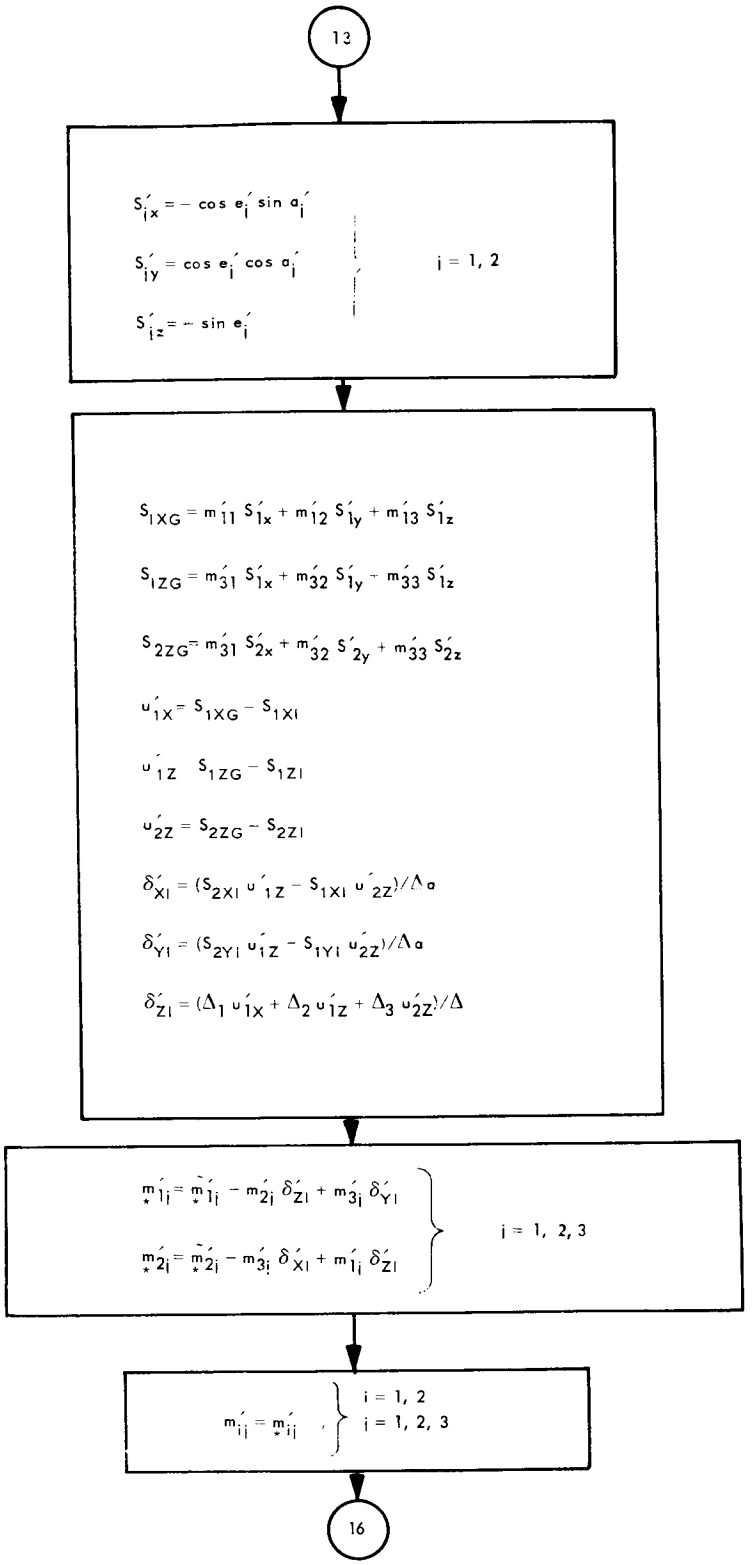


Figure 3-9. Control Computer Math Flow Experiment Three, Horizon Spectrometry (Sheet 2 of 3)

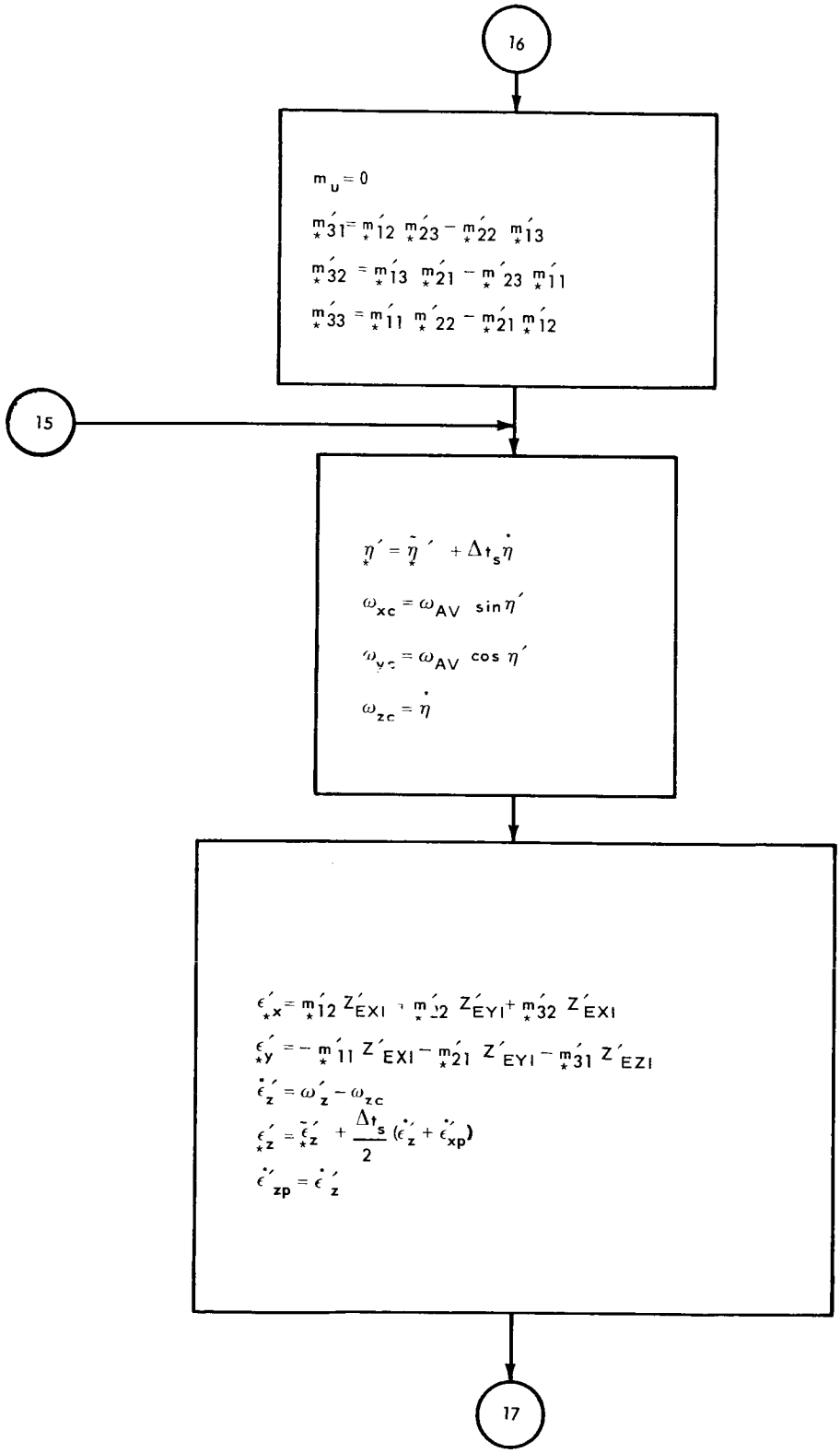


Figure 3-9. Control Computer Math Flow Experiment Three, Horizon Spectrometry (Sheet 3 of 3)

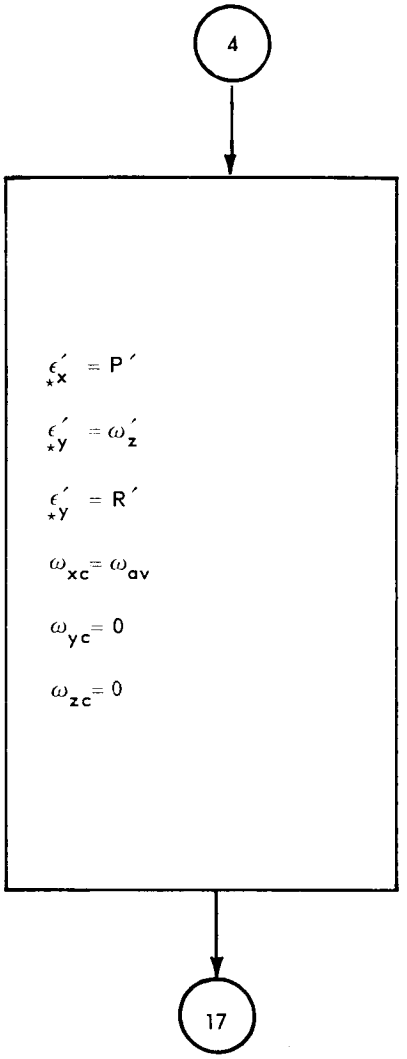


Figure 3-10. Control Computer Math Flow Experiment Four, Microwave Transmission

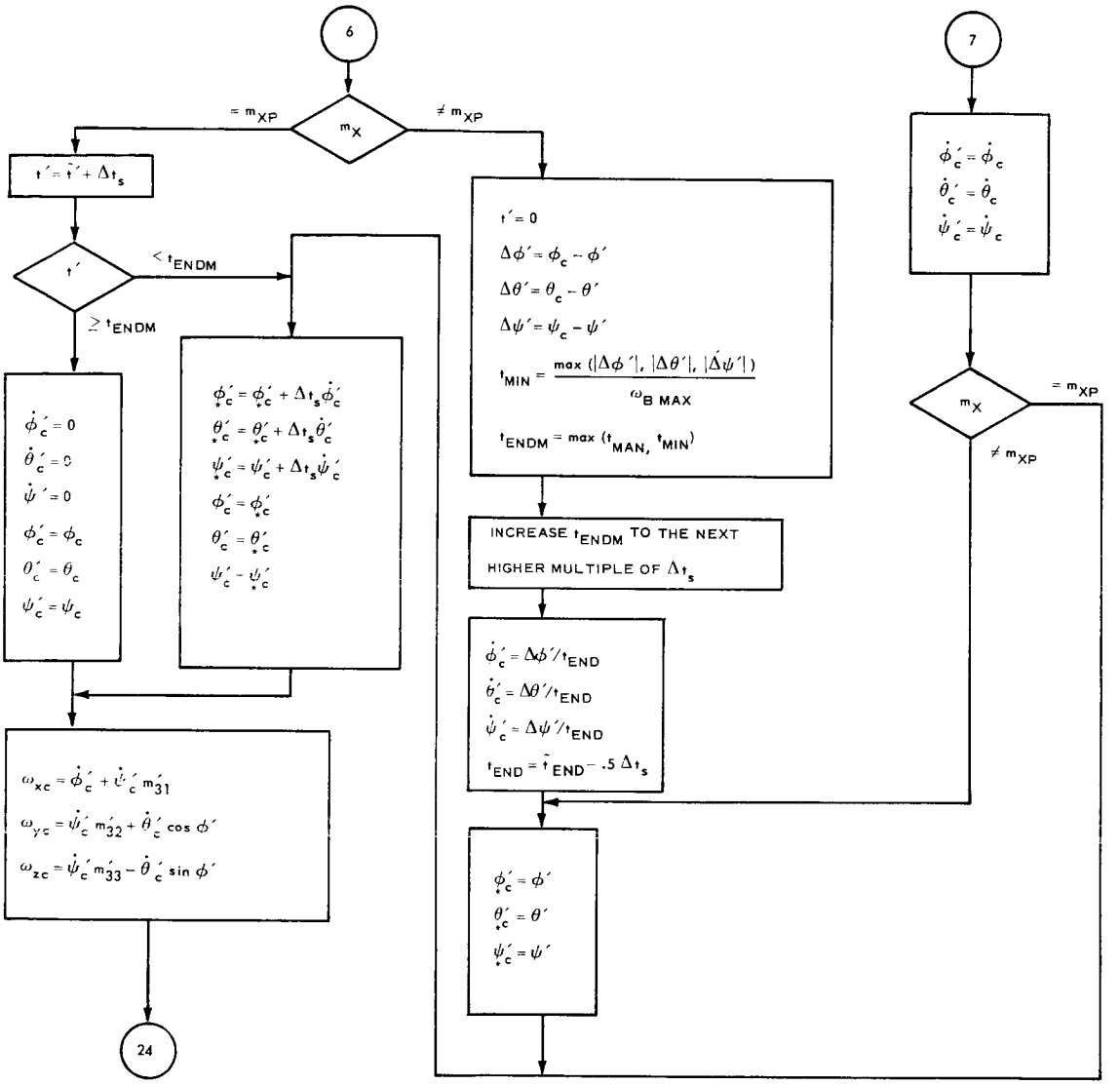


Figure 3-11. Control Computer Math Flow Mode 5, Attitude Hold; Mode 6, Maneuver; Mode 7, Manual (Sheet 1 of 2)

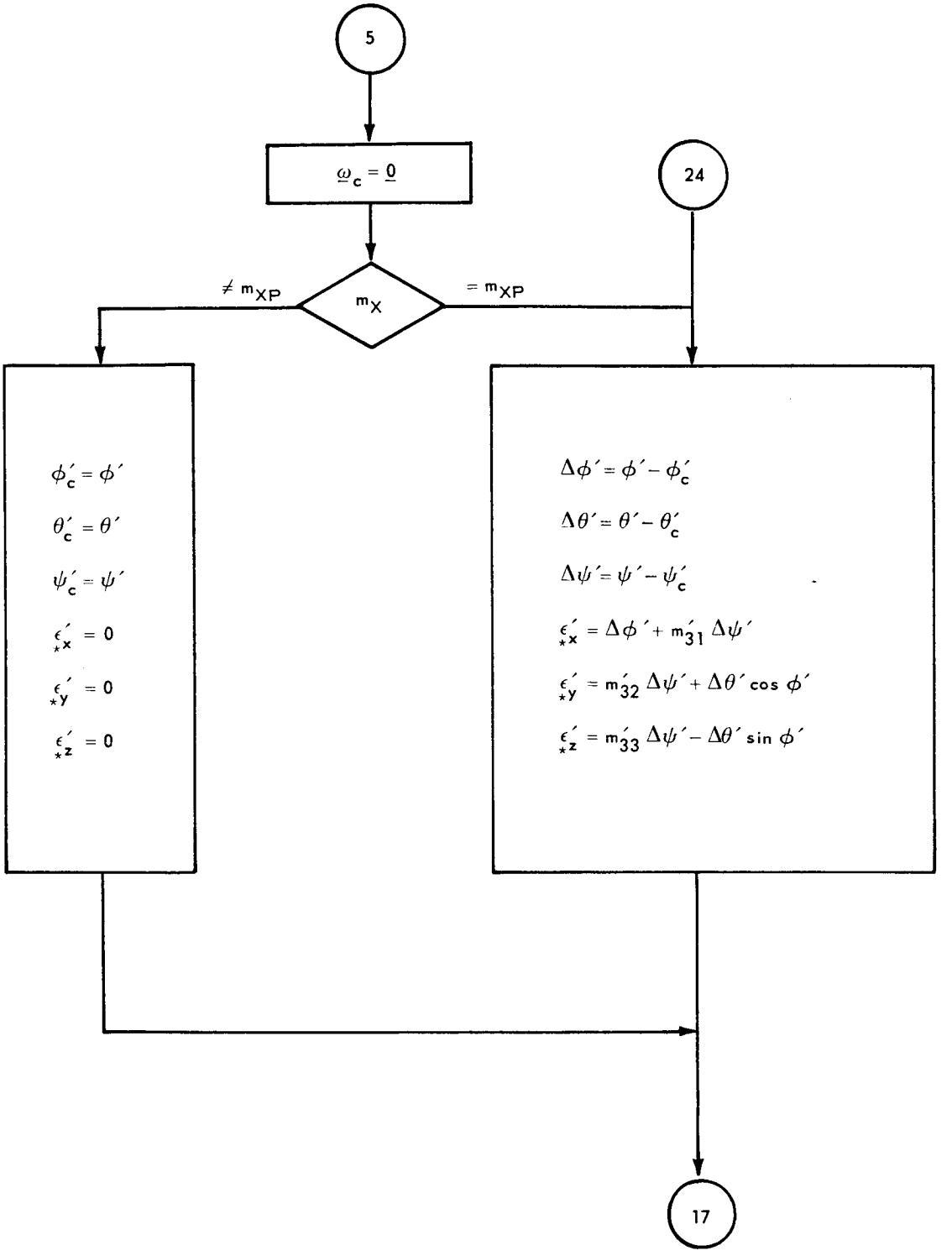


Figure 3-11. Control Computer Math Flow Mode 5, Attitude Hold; Mode 6, Maneuver; Mode 7, Manual (Sheet 2 of 2)

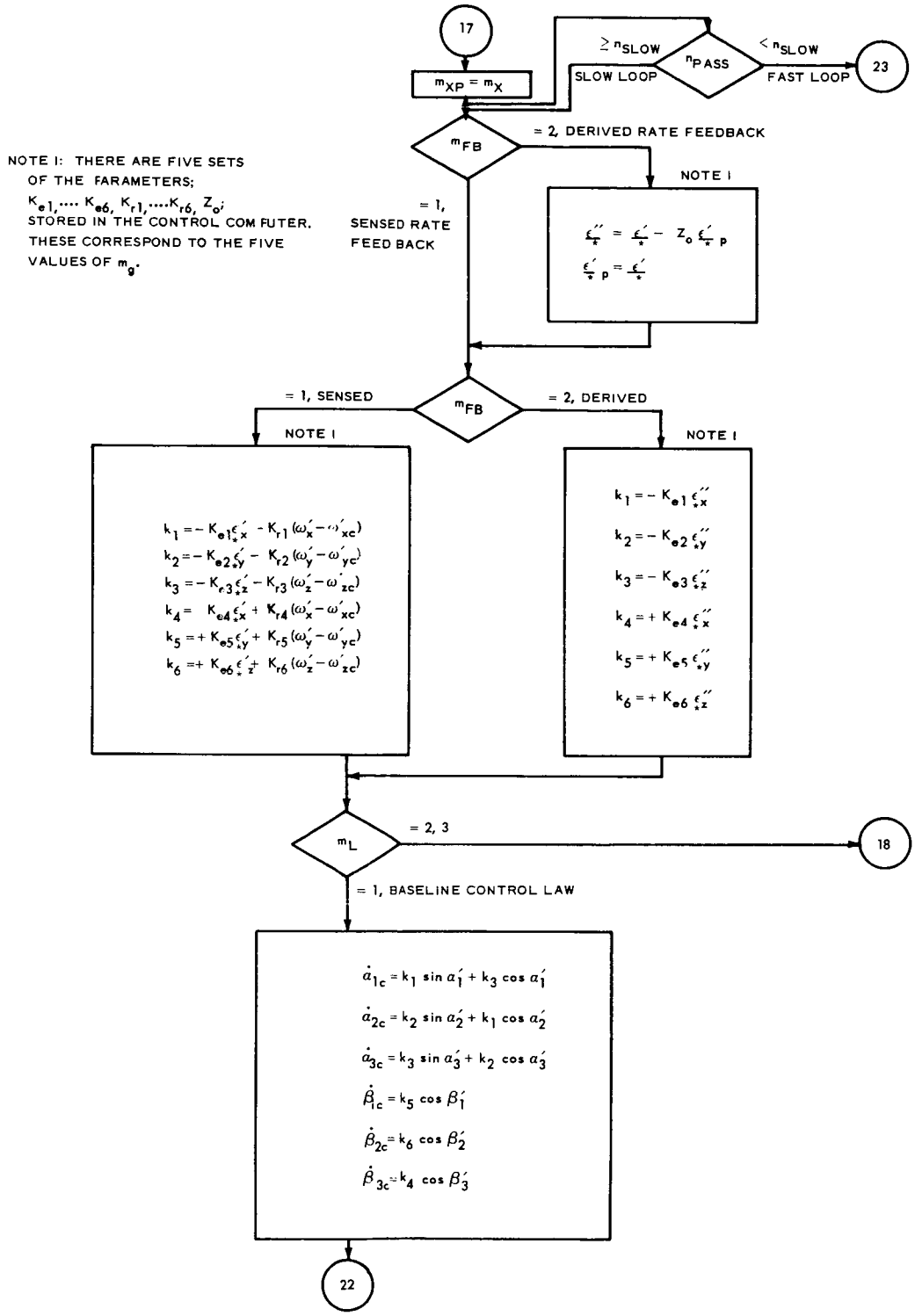


Figure 3-12. Control Computer Math Flow Compensation and Baseline Control Law

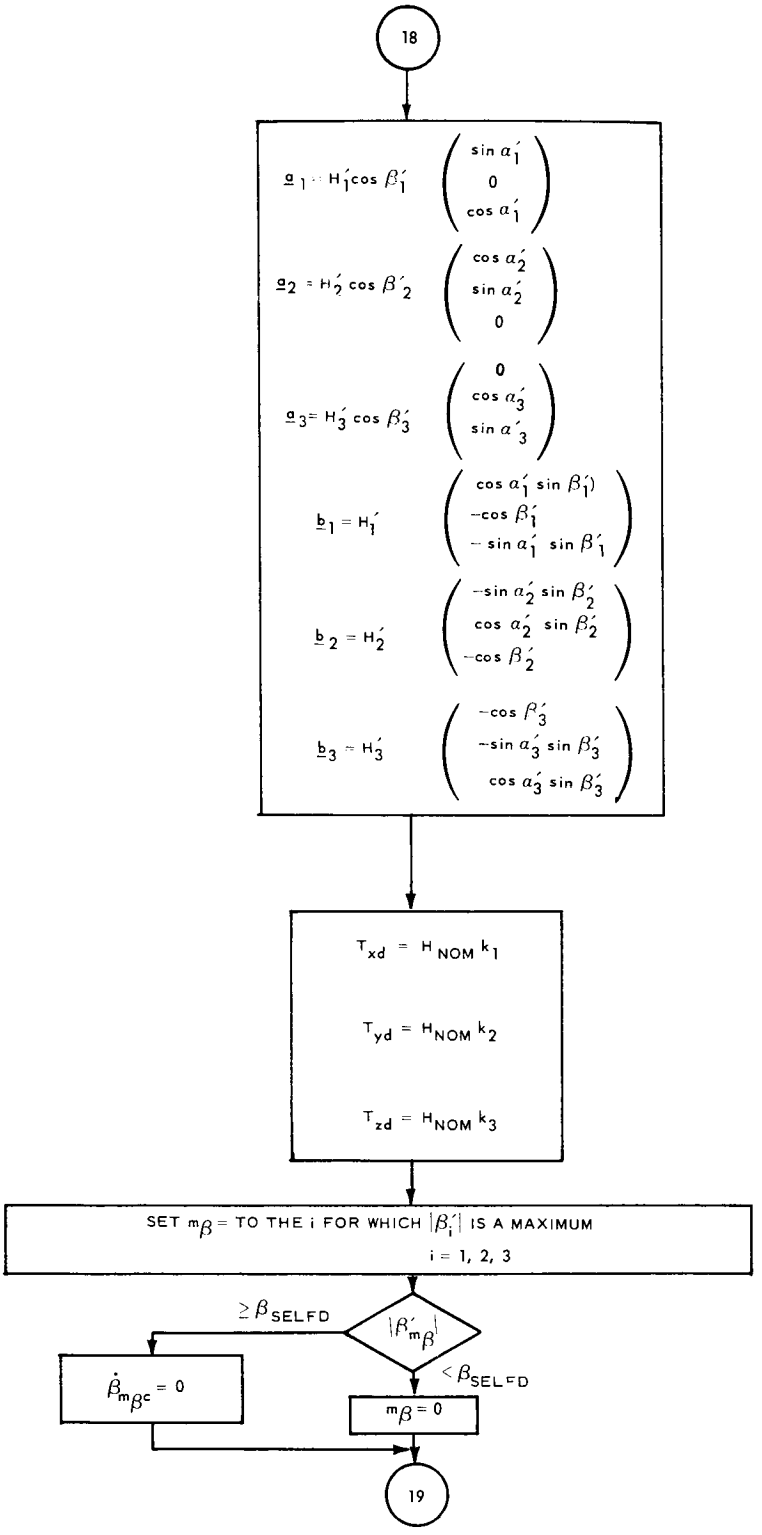


Figure 3-13. Control Computer Math Flow Precomputation for Control Laws 2 and 3 (Sheet 1 of 2)

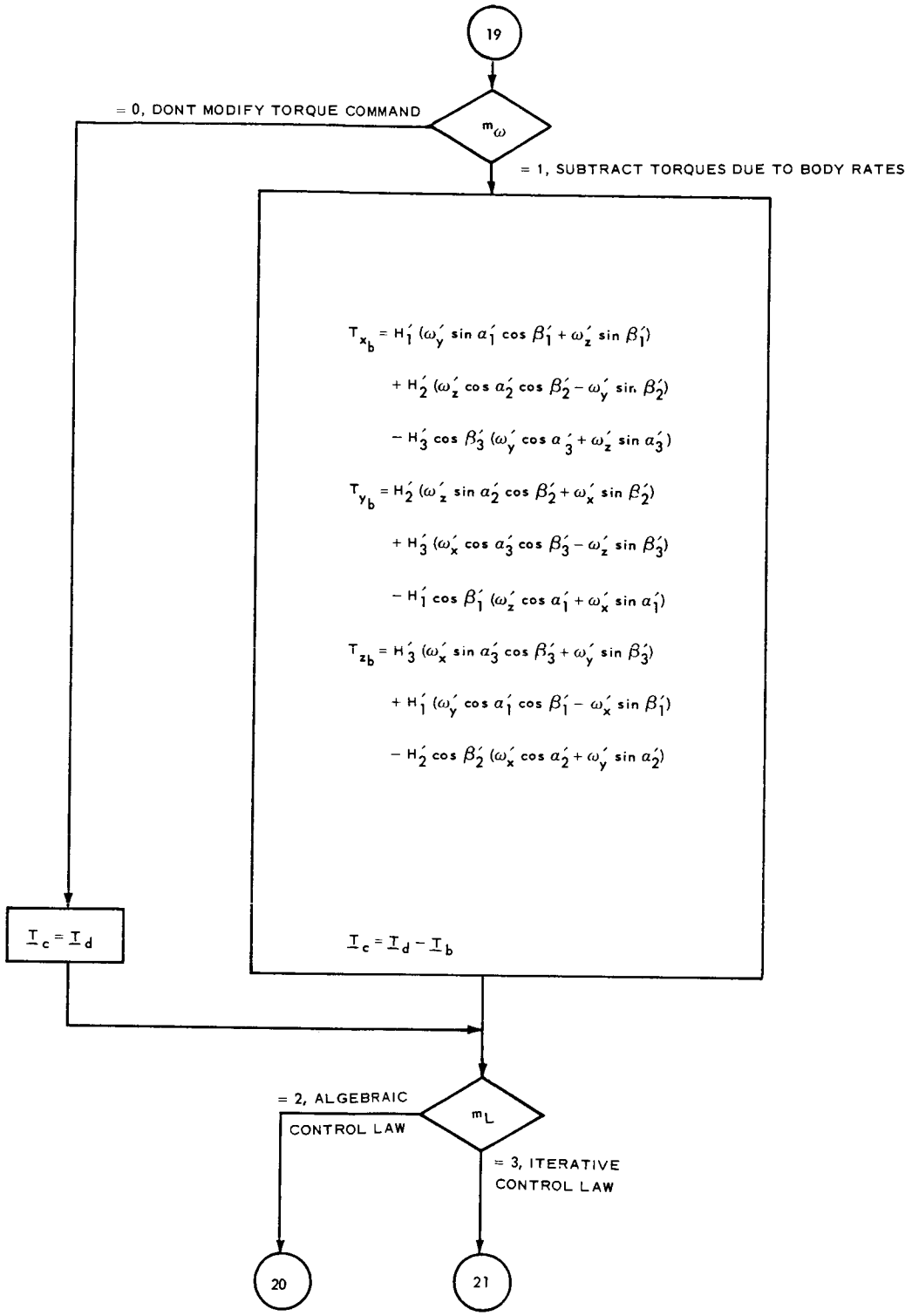


Figure 3-13. Control Computer Math Flow Precomputation for Control Laws 2 and 3 (Sheet 2 of 2)

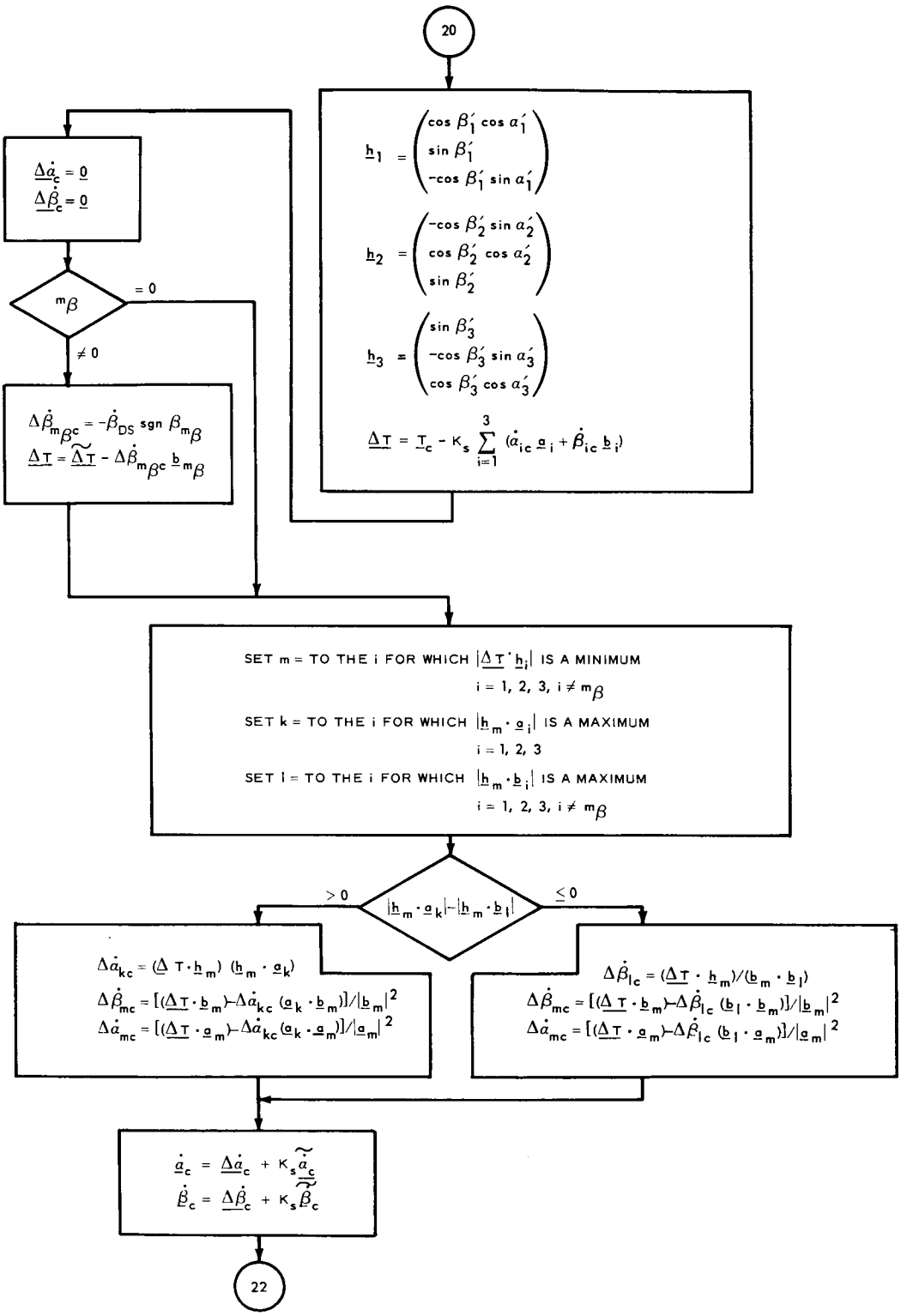


Figure 3-14. Control Computer Math Flow Algebraic Control Law

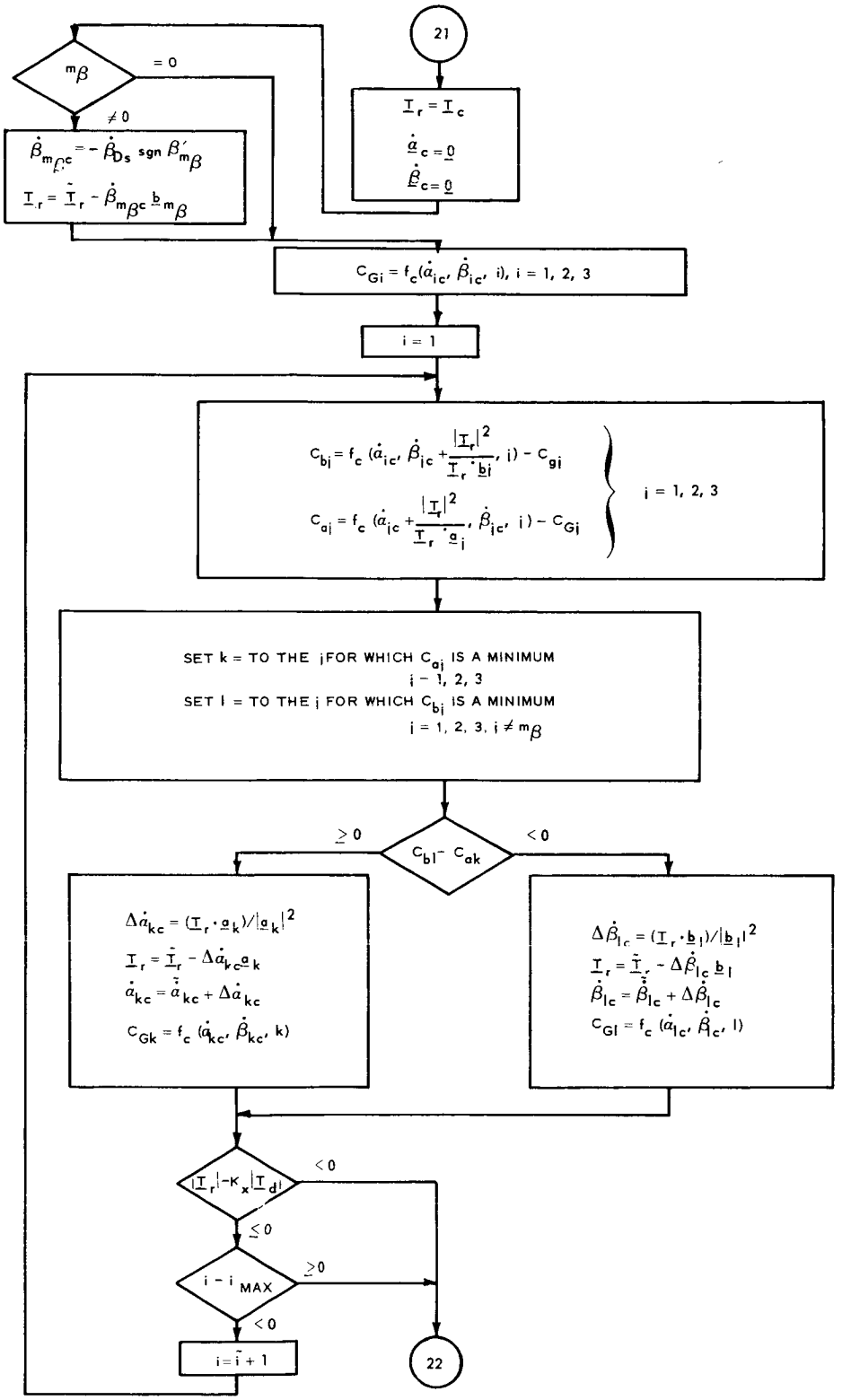


Figure 3-15. Control Computer Math Flow Iterative Control Law

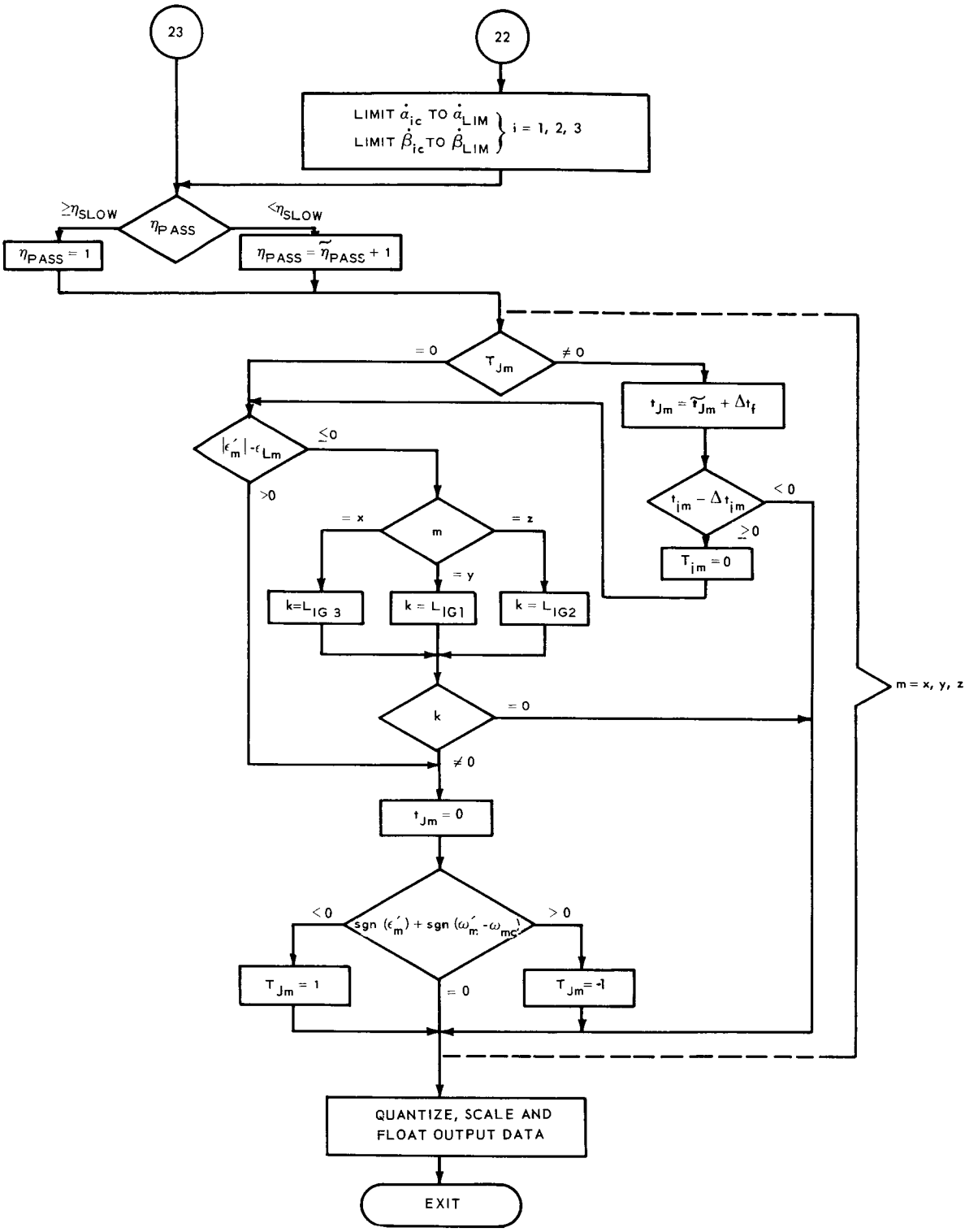


Figure 3-16. Control Computer Math Flow Desaturation Test

```
$IBFTC S2       LIST
      SUBROUTINE COMP
C
C
      COMMON/SINCO/SPACE(30),DIRCO(3,3)
      COMMON/FLOTIN/XIN(25),NXIN(12)
      COMMON/FLOOUT/XOUT(16),NXOUT(3)
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/SCALER/CSCALE,DSCALE,LSCALE,LLSCAL,MSCALE,NSCALE,PSCALE,
     CQSCALE,RSCALE,TSCALE,VSCALE
      INTEGER CSCALE,DSCALE,PSCALE,QSCALE,RSCALE,TSCALE,VSCALE
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NP0,FL2NP1
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/FIXIN/W(3),E(2),A(2),PH,RH,BETA(3),ALPHA(3),EDOTC(3),EC(3),
     CTM,HCL(3),MODE,RATEFB,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER A,ALPHA,BETA,E,EC,EDOTC,HCL    ,PH,RATEFB,RH,TM,UPDATE,W,
      COMMON/QUANT/QIN(25),QOUT(16),NEXPNT
      INTEGER QIN,QOUT
      COMMON/FIXOUT/BDOTC(3),ADOTC(3),AC,ED(3),EP(3),WC(3),JET(3)
      INTEGER AC,ADOTC,BDOTC,ED,EP,WC
      COMMON/EXP1V/ACDBL,ACDOT,ADSAVE,COSAC,COSWT,          OMEGA(3),
     COMEGAE(3),RO(3),S,SDOT,SDUM,SINAC,SINWT,SPRIME,SREL(3),SSQ,TANAC,
     CV(3),VC(3),VDOUB(3),WE,WE2,WE3,WE4
      INTEGER ACDBL,ACDOT,ADSAVE,COSAC,COSWT,      OMEGA,OMEGAE,RO,S,
     CSDOT,SDUM,SINAC,SINWT,SPRIME,SREL,SSQ,TANAC,V,VC,VDOUB,WE,WE2,WE3,
     CWE4
      COMMON/EXP2V/DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,
     CU2X,U1Z,U2Z
      INTEGER DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,U2X,
     CU1Z,U2Z
      COMMON/EXP3V/ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,KC,PREV,PSP(3),R,SINDUM,SINL,SINLR,SIN2LR,SINTH,
     CSPX(2),SPY(2),SPZ(2),S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,
```

```
CZEYI,ZEZI,EPZDOT
 INTEGER ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
CDBLPI,ETADOT,    PREV,PSP,R,SINDUM,SINL,SINLR,SIN2LR,SINTH,SPX,SPY,
CSPZ,S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,ZEYI,ZEZI,EPZDOT
 COMMON/MOD567/DELE(3),ECOM(3),EDCOM(3),TENDM,ECOMDP(3),DELTAE(3),
CMAX,MAXRT ,TP,NHHH
 INTEGER DELE,ECOM,EDCOM,TENDM,ECOMDP,DELTAE,TP
 COMMON/CONTL1/EPPREV(3),GAIN(6,5),GAINP(6,5),EPP(3),HNOM ,KLCL(6),
C     MAGA(3),MAGB(3),MCA(3),MCB(3),TRQC(3),TRQCP(3),UNITVA(3,3),
CUNITVB(3,3),ZO(5)
 INTEGER EPPREV,GAIN,GAINP,EPP,HNOM,TRQC,TRQCP,UNITVA,UNITVB,ZO
 COMMON/CONTL2/DELA(3),DELB(3),DOT1(3),DOT2(3),DOT3(3),DUM1,DUM2,
CKSAVE,MAGASQ(3),MAGBSQ(3),TREM(3),TRQPRD(3),UNITVH(3,3)
C ,BHOLD,BSELFD,BDOTDS,MBMAX
 INTEGER DELA,DELB,DOT1,DOT2,DOT3,DUM1,DUM2,TREM,TRQPRD,UNITVH
C ,BHOLD,BSELFD,BDOTDS
 INTEGER TRQCPP(3)
 EQUIVALENCE(TRQCPP(1),TRQPRD(1))
 COMMON/CONTL3/CGYRO(3),COSTA(3),COSTB(3),RTEST,RUSE,TDESSQ,
CTDOTA(3),TDOTB(3),TREMSQ,XKEND,ITER,NOITER
 INTEGER CGYRO,COSTA,COSTB,RTEST,RUSE,TDESSQ,TDOTA,TDOTB,TREMSQ,
CXKEND
 COMMON/DESAT/ERRLIM(3),JETCT(3),TJCNT(3),GIMLIM(3),BDOTMX,ADOTMX
 INTEGER TJCNT,ERRLIM,GIMLIM,BDOTMX,ADOTMX
 COMMON/TVECT/DELT,H,T(10),NPASS,NSLOW,XNSLOW
 INTEGER DELT,H,T
 COMMON/NAV/F(10),FDOT,FDUM(10),FTOT,G(10),GDOT,GDUM(10),GTOT,P(3),
CPO(3),PDOT(3),PDOTO(3)
 INTEGER F,FDOT,FDUM,FTOT,G,GDOT,GDUM,GTOT,P,PO,PDOT,PDOTO
 COMMON/DIRCOS/MS(3,3),MSDOT(3,3),ML(3,3),MLDOT(3,3),DSAVE(3,3),
CMSDBL(3,3),MLDBL(3,3),XW(3),WPREV(3),TRATIO(3),DELW(3)
 INTEGER DSAVE,XW,WPREV,TRATIO,DELW
 COMMON/NTRIG/COSE(2),COSA(2),CBETA(3),CALPHA(3),SINE(2),SINA(2),
CSBETA(3),SALPHA(3),SINED(1),COSED(1)
 INTEGER COSA,COSE,CALPHA,CBETA,SINA,SINE,SALPHA,SBETA,SINED,COSED
 INTEGER DMULT,QUOTNT,SQRTCC,DUM
 INTEGER SCORR
C
```

```
C FIX AND SCALE SENSED DATA
C
      DO 140 I = 1,3
      W(I) = FLNP5 * XIN(I)
      E(I)=FLNM2*XIN(I+3)
      A(I+1)=FLNP1*XIN(I+6)
      EDOTC(I) = FLNP5 * XIN(I+15)
      EC(I)=FLNM2*XIN(I+18)
      BETA(I) = FLNM2 * XIN(I+9)
      ALPHA(I)= FLNM2 * XIN(I+12)
      HCL(I) = FLNM11 * XIN(I+22)
      HCL(I+3)=NXIN(I)
      HCL(I+6)=NXIN(I+3)
      LIMIG(I) =NXIN(I+6)
  140 LIMOG(I) =NXIN(I+9)
      A(2)=A(2)/8
      EC(2)=EC(2)*2
      TM = FLNM8 * XIN(22)
C
C INPUT DATA QUANTIZED
C
      DO 147 I = 1,25
      IF(IABS(W(I)).GT.FS)W(I)=ISIGN(FS,W(I))
  147 W(I) = (W(I)/QIN(I))*QIN(I)
C
C DIRECTION COSINE INTEGRATION
C
      K= 1
      DO 179 J=1,3
      DELW(J)=(TRATIO(J)*(W(J)-WPREV(J)))/NPO
179   XW(J)=WPREV(J)+DELW(J)
   34 DO 19 J = 1,2
      MSDOT(J,1) =(XW(3)*MS(J,2))/NPO -(XW(2)*MS(J,3))/NPO
      MSDOT(J,2) =(XW(1)*MS(J,3))/NPO -(XW(3)*MS(J,1))/NPO
   19 MSDOT(J,3) =(XW(2)*MS(J,1))/NPO -(XW(1)*MS(J,2))/NPO
      IF(K.GE.2)GO TO 31
      DO 33 J=1,3
      WPREV(J)=W(J)
```

```
      XW(J)=W(J)+DELW(J)
      DO 33 I = 1,2
      DSAVE(I,J) = MSDOT(I,J)
   33 MS(I,J) = (MSDBL(I,J) +(H*MSDOT(I,J))/CSCALE)/NPO
      K = 2
      GO TO 34
   31 IF(UPDATE.NE.2)GO TO 32
      DO 191 J=1,3
      DO 191 I=1,3
      MSDBL(I,J)=DIRCO(I,J)*FL2NPO
  191 MS(I,J)=MSDBL(I,J)/NPO
      NXIN(3)=0
      GO TO 190
   32 DO 35 I = 1,2
      DO 35 J = 1,3
      MSDBL(I,J) =  MSDBL(I,J) +(H*(MSDOT(I,J) + DSAVE(I,J)))/DSCALE
   35 MS(I,J) = MSDBL(I,J)/NPO
      MSDBL(3,1)=DMULT(MSDBL(1,2),MSDBL(2,3))-DMULT(MSDBL(2,2),MSDBL(1,3
     1 ))
      MSDBL(3,2)=DMULT(MSDBL(1,3),MSDBL(2,1))-DMULT(MSDBL(2,3),MSDBL(1,1
     1 ))
      MSDBL(3,3)=DMULT(MSDBL(1,1),MSDBL(2,2))-DMULT(MSDBL(2,1),MSDBL(1,2
     1 ))
      DO 180 J=1,3
180   MS(3,J)=MSDBL(3,J)/NPO
C
C EULER ANGLES AND TRIG FUNCTIONS
C
  190 CALL CCATAN(ED(1),MS(3,2),MS(3,3))
      CALL CCASIN(ED(2),-MS(3,1))
      CALL CCATAN(ED(3),MS(2,1),MS(1,1))
      DO 27 I = 1,4
      CALL SINCC(E(I),SINE(I))
   27 CALL COSCC(E(I),COSE(I))
      DO 28 I = 1,6
      CALL SINCC(BETA(I),SBETA(I))
   28 CALL COSCC(BETA(I),CBETA(I))
      CALL SINCC(ED(1),SINED(1))
```

```
      CALL COSCC(ED(1),COSED(1))
      IF(NPASS.GE.NSLOW)GO TO 170
      IF(MODE-1)171,101,171
C
C POSITION AND VELOCITY
C
C
C T ARRAY IS DOUBLE PRECISION
C
170   T(1)=T(1)+DELT
      IF(T(1).GT.DBLFS)T(1)=DBLFS
      T(2)=DMULT(T(1),T(1))
      T(3)=(DMULT(T(1),T(2))*2)/3
      T(4)=DMULT(T(1),T(3))
      T(5)=(DMULT(T(1),T(4))/5)*4
      T(6)=(DMULT(T(1),T(5))/3)*4
      T(7)=(DMULT(T(1),T(6))/7)*8
      T(8)=DMULT(T(1),T(7))
      T(9)=(DMULT(T(1),T(8))/9)*8
      T(10)=(DMULT(T(1),T(9))/5)*4
      DO 142 J=1,10
      FDUM(J)=DMULT(F(J),T(J))
  142 GDUM(J)=DMULT(G(J),T(J))
      FTOT=       FDUM(2)+FDUM(3)+(((((( FDUM(10)/8+FDUM(9))/4+FDUM(8))/
     1 8+FDUM(7))/8+FDUM(6))/4+FDUM(5))/4+FDUM(4))/4+DBLFS/2
      GTOT=((((((( GDUM(10)/4+GDUM(9))/8+GDUM(8))/8+GDUM(7))/4+GDUM(6))/
     1 8+GDUM(5))/4+GDUM(4))/2+GDUM(3))/2+GDUM(1)
      DO 143 J=2,10
      FDUM(J)=DMULT(F(J),T(J-1))
  143 GDUM(J)=DMULT(G(J),T(J-1))
      FDOT=(((((((( FDUM(10)/8+FDUM(9))/4+FDUM(8))/8+FDUM(7))/8+FDUM(6))/
     1 2+FDUM(5))/4+FDUM(4))/2+FDUM(3)+FDUM(2))*2
      GDOT=  ((((( GDUM(10)/4+GDUM(9))/8+GDUM(8))/8+GDUM(7))/4+GDUM(6))/
     1 4+GDUM(5))/4+GDUM(4)+GDUM(3)+G(1)
      DO 144 J=1,3
      P(J)=(DMULT(FTOT,PO(J))+DMULT(GTOT,PDOTO(J)))*2
 144  PDOT(J)=(DMULT(FDOT,PO(J))+DMULT(GDOT,PDOTO(J)))*2
C
```

```
C SELECT MODE ON MODE BRANCH
C
      GO TO (101,102,103,104,105,106,107),MODE
C
C EXPERIMENT ONE
C
  101 IF(MODE.EQ.MDLAST)GO TO 20
      K=1
  155 COSWT=(DMULT(WE4,T(4))/1024-DMULT(WE2,T(2)))/256+DBLFS
      SINWT=DMULT(WE,T(1))-DMULT(WE3,T(3))/1024
      SREL(1)=P(1)-DMULT(RO(1),COSWT)-DMULT(RO(3),SINWT)/8
      SREL(2)=P(2)-RO(2)
      SREL(3)=P(3)-DMULT(RO(3),COSWT)+DMULT(RO(1),SINWT)/8
      VDOUB(1)=PDOT(1)-DMULT(P(3),WE)/8
      VDOUB(2)=PDOT(2)
      VDOUB(3)=PDOT(3)+DMULT(P(1),WE)/8
      SDUM=0
      SSQ=0
      DO 141 J=1,3
      V(J) = VDOUB(J)/NPO
      SSQ=SSQ+DMULT(SREL(J),SREL(J))
  141 SDUM=SDUM+DMULT(SREL(J),VDOUB(J))
      S=SQRTCC(SSQ/NM4)
      SCORR=((SSQ*16-(S*S))*128)/S
      IF(IABS(SCORR).GT.FS)SCORR=ISIGN(FS,SCORR)
      SDOT=(SDUM/S)*4
      SDUM=(SCORR*NPO)/S
      SDOT=SDOT-(SDUM*SDOT)/NP8
  162 DUM=AC/4
      CALL SINCC(DUM,SINAC)
      CALL COSCC(DUM,COSAC)
      SPRIME =(S*COSAC)/NPO
      DO 185 J=1,3
      OMEGA(J)=(ML(2,J)*OMEGAE(2))/NPO
      VC(J) = 0
      DO 10 K=1,3
10    VC(J)=VC(J)+DMULT(MLDBL(K,J),VDOUB(K))
185   VC(J)=VC(J)/NPO
```

```
      TANAC=(SINAC*NM1)/COSAC
      WC(3)=(VC(1)*NM3)/SPRIME
      WC(3)=WC(3)-(SDUM*WC(3))/NP8
      WC(3) = OMEGA(3)/256 +(OMEGA(2)*TANAC)/NP7+WC(3)
      ACDOT =((VC(3)-(SDOT*SINAC)/NPO)*NM3)/SPRIME
      ACDOT=ACDOT-(ACDOT*SDUM)/NP8
      ACDOT=ACDOT-OMEGA(1)/256
      DO 11 J= 1,3
   11 MLDOT(J,1) = (WC(3) * ML(J,2))/NPO
      IF(K-2)166,21,26
   20 K=2
166   DO 22 J = 1,3
      DSAVE(J,1) = MLDOT(J,1)
   22 MLDBL(J,1) =  MLDBL(J,1) + (H*MLDOT(J,1))/CSCALE
      ADSAVE = ACDOT
      AC = (ACDBL + (H*ACDOT)/CSCALE)/NPO
      GO TO 23
   21 K=3
      DO 24 J = 1,3
24    MLDBL(J,1) = MLDBL(J,1) + (H*(MLDOT(J,1)-DSAVE(J,1)))/DSCALE
      ACDBL = ACDBL + (H*(ACDOT + ADSAVE))/DSCALE
      AC = ACDBL/NPO
 23   MLDBL(1,2)=DMULT(MLDBL(3,1),MLDBL(2,3))-DMULT(MLDBL(2,1),MLDBL(3,3
     1 ))
      MLDBL(2,2)=DMULT(MLDBL(1,1),MLDBL(3,3))-DMULT(MLDBL(3,1),MLDBL(1,3
     1 ))
      MLDBL(3,2)=DMULT(MLDBL(2,1),MLDBL(1,3))-DMULT(MLDBL(1,1),MLDBL(2,3
     1 ))
      DO 181 J=1,3
      ML(J,1)=MLDBL(J,1)/NPO
181   ML(J,2)=MLDBL(J,2)/NPO
      IF(K-2)167,155,162
167   K=2
      GO TO 162
 26   EP(1)=(DMULT(MSDBL(1,2),MLDBL(1,3))+DMULT(MSDBL(2,2),MLDBL(2,3))+
     1 DMULT(MSDBL(3,2),MLDBL(3,3)))*2
      EP(2)=(DMULT(MSDBL(1,3),MLDBL(1,1))+DMULT(MSDBL(2,3),MLDBL(2,1))+
     1 DMULT(MSDBL(3,3),MLDBL(3,1)))*2
```

```
      EP(3)=(DMULT(MSDBL(1,1),MLDBL(1,2))+DMULT(MSDBL(2,1),MLDBL(2,2))+
     1 DMULT(MSDBL(3,1),MLDBL(3,2)))*2
      WC(1) = 0
      WC(2) = 0
      GO TO 100
C
C EXPERIMENT TWO
C
  102 U1X = (SINE(1)-S1X)
      U2X = (SINE(2)-S2X)
      U1Z =((COSE(1) *COSA(1))/NPO-S1Z)
      EP(1) = QUOTNT(2*(DEL1*U1X+DEL2*U1Z+DEL3*U2X),DEL)
      EP(2) = QUOTNT(2*(S2Y*U1X-S1Y*U2X),DELP)
      EP(3) = QUOTNT(2*(S2Z*U1X-S1Z*U2X),DELP)
      DO 161 J=1,3
161   WC(J)=0
      GO TO 100
C
C EXPERIMENT THREE
C
C
C CONVERT P VECTOR TO SINGLE PRECISION
C
103   DO 14 I = 1,3
   14 PSP(I) = P(I)/NPO
      Z = SQRTCC((PSP(1)**2 + PSP(3)**2)/NPO)
      R = SQRTCC((PSP(1)**2+PSP(2)**2+PSP(3)**2)/NPO)
      SINTH =(PSP(1)*NPO)/Z
      COSTH =(PSP(3)*NPO)/Z
      SINLR =(PSP(2)*NPO)/R
      COSLR =(Z*NPO)/R
      SIN2LR=(SINLR*COSLR)/NM1
      SINL = SINLR +(((SIN2LR*COSLR)/NPO)*KC)/NP8
      COSL = COSLR -(((SIN2LR*SINLR)/NPO)*KC)/NP8
      ZEXI =(-SINTH * COSL)/NPO
      ZEYI = -SINL
      ZEZI =(-COSTH * COSL)/NPO
C
```

```
C MAKE DECISION TO UPDATE
C
      IF(UPDATE.NE.1)GO TO 15
C
C UPDATE SECTION
C
      DO 29 I = 1,2
      SPX(I) = (-COSE(I) * SINA(I))/NPO
      SPY(I) = (COSE(I) * COSA(I))/NPO
   29 SPZ(I) = -SINE(I)
      S1XG =(MS(1,1)*SPX(1)+MS(1,2)*SPY(1)+MS(1,3)*SPZ(1))/NPO
      S1ZG =(MS(3,1)*SPX(1)+MS(3,2)*SPY(1)+MS(3,3)*SPZ(1))/NPO
      S2ZG =(MS(3,1)*SPX(2)+MS(3,2)*SPY(2)+MS(3,3)*SPZ(2))/NPO
       U1X =(S1XG - S1X)
       U1Z =(S1ZG - S1Z)
       U2Z = (S2ZG - S2Z)
      DELX = QUOTNT(2*(S2X*U1Z-S1X*U2Z),DELP)/NM7
      IF(IABS(DELX).GT.FS)DELX = ISIGN(FS,DELX)
      DELY = QUOTNT(2*(S2Y*U1Z-S1Y*U2Z),DELP)/NM7
      IF(IABS(DELY).GT.FS)DELY = ISIGN(FS,DELY)
      DELZ = QUOTNT(2*(DEL1*U1X+DEL2*U1Z+DEL3*U2Z),DEL)/NM7
      IF(IABS(DELZ).GT.FS)DELZ = ISIGN(FS,DELZ)
      DO 17 I = 1,3
      MSDBL(1,I) = MSDBL(1,I) + (-MS(2,I)*DELZ+MS(3,I)*DELY)/256
      MSDBL(2,I) = MSDBL(2,I) + (MS(1,I)*DELZ-MS(3,I)*DELX)/256
      DO 17 K = 1,2
   17 MS(K,I) = MSDBL(K,I)/NPO
      MSDBL(3,1)=DMULT(MSDBL(1,2),MSDBL(2,3))-DMULT(MSDBL(2,2),MSDBL(1,3
     1 ))
      MSDBL(3,2)=DMULT(MSDBL(1,3),MSDBL(2,1))-DMULT(MSDBL(2,3),MSDBL(1,1
     1 ))
      MSDBL(3,3)=DMULT(MSDBL(1,1),MSDBL(2,2))-DMULT(MSDBL(2,1),MSDBL(1,2
     1 ))
      DO 182 J=1,3
182   MS(3,J)=MSDBL(3,J)/NPO
      NXIN(3)=0
   15 ANGLE = ANGLE + DELANG
      IF(ANGLE.GE.DBLFS)ANGLE=ANGLE-DBLPI-DBLPI
```

```
      IF(ANGLE.LE.-DBLFS)ANGLE=ANGLE+DBLPI+DBLPI
      DUM = ANGLE/NPO
      CALL SINCC(DUM ,SINDUM)
      CALL COSCC(DUM ,COSDUM)
      WC(1) = ( WBAR*SINDUM)/NPO
      WC(2) = (WBAR*COSDUM)/NPO
      WC(3) = ETADOT
      EP(1)=(DMULT(MSDBL(1,2),ZEXI*NPO)+DMULT(MSDBL(2,2),ZEYI*NPO)+
     1 DMULT(MSDBL(3,2),ZEZI*NPO))*2
      EP(2)=-(DMULT(MSDBL(1,1),ZEXI*NPO)+DMULT(MSDBL(2,1),ZEYI*NPO)+
     1 DMULT(MSDBL(3,1),ZEZI*NPO))*2
      EPZDOT = W(3) - WC(3)
      EP(3) =((H*(EPZDOT+PREV))/CSCALE)*NSLOW+EP(3)
      PREV = EPZDOT
      GO TO 100
C
C EXPERIMENT FOUR
C
  104 EP(1) = PH*NPO
      EP(2)=W(3)*NM4
      EP(3) = RH*NPO
      WC(1) = WBAR
      WC(2) = 0
      WC(3) = 0
      GO TO 100
C
C MODE 5
C
  105 DO 50 I = 1,3
   50 WC(I) = 0
      IF(MODE-MDLAST)51,52,51
   51 DO 53 I = 1,3
      ECOM(I) = ED(I)
   53 EP(I) = 0
      GO TO 100
   52 DO 55 I = 1,3
   55 DELE(I) =-ECOM(I) + ED(I)
      EP(1) = (MS(3,1)*DELE(3))*8 + DELE(1)*NP8
```

```
      EP(2) = (MS(3,2)*DELE(3))*8 + (COSED(1)*DELE(2))*4
      EP(3) = (MS(3,3)*DELE(3))*8 - (SINED(1)*DELE(2))*4
      GO TO 100
C
C MODE 6
C
  106 IF(MODE-MDLAST)56,57,56
   56 DO 149 I = 1,3
  149 DELTAE(I) = EC(I) - ED(I)
      DELTAE(2) = DELTAE(2)/2
      MAX = IABS(DELTAE(1))
      DO 150 I = 2,3
150   IF(MAX-IABS(DELTAE(I)).LT.0)MAX = IABS(DELTAE(I))
      TP = (MAX*NM1)/MAXRT
      TENDM=TM
      IF(TM.LT.TP)TENDM = TP
      TP=0
      TENDM=((((TENDM*NM2)/DELT)+1)*DELT)/NM2
      DO 58 I = 1,3
      EDCOM(I)=(DELTAE(I)*NM1)/TENDM
   58 ECOMDP(I) = ED(I)*NPO
      TENDM=TENDM*NM2-DELT/2
      GO TO 67
57    TP=TP+DELT
      IF(TP  -TENDM)67,59,59
   67 ECOMDP(1) = ECOMDP(1) +((NHHH*EDCOM(1))/LLSCAL)*NSLOW
      ECOMDP(2) = ECOMDP(2) +((NHHH*EDCOM(2))/DSCALE)*NSLOW
      ECOMDP(3) = ECOMDP(3) +((NHHH*EDCOM(3))/LLSCAL)*NSLOW
      DO 60 I = 1,3
   60 ECOM(I) = ECOMDP(I)/NPO
      GO TO 61
   59 DO 62 I= 1,3
      EDCOM(I) = 0
   62 ECOM(I) = EC(I)
   61 WC(1) = EDCOM(1) + (EDCOM(3)*MS(3,1))/NPO
      WC(2) =(EDCOM(3)*MS(3,2))/NPO + (EDCOM(2)*COSED(1))/NPO
      WC(3) =(EDCOM(3)*MS(3,3))/NPO - (EDCOM(2)*SINED(1))/NPO
      GO TO 52
```

```
C
C MODE 7
C
  107 IF(MODE-MDLAST)63,64,63
   63 DO 65 I = 1,3
   65 ECOMDP(I) = ED(I)*NPO
   64 DO 66 I = 1,3
   66 EDCOM(I) = EDOTC(I)
      GO TO 67
C
  100 DO 12 J = 1,3
   12 IF(IABS(EP(J)).GT.DBLFS)EP(J) = ISIGN(DBLFS,EP(J))
      MDLAST=MODE
      IF(NPASS.GE.NSLOW)CALL COMPX
171   NPASS=NPASS+1
      IF(NPASS.GT.NSLOW)NPASS=1
      GIMLIM(1)=LIMIG(3)
      GIMLIM(2)=LIMIG(1)
      GIMLIM(3)=LIMIG(2)
      DO 701 J=1,3
      IF(JET(J).EQ.0)GO TO 711
      TJCNT(J)=TJCNT(J)+1
      IF(TJCNT(J).LT.JETCT (J))GO TO 701
      JET(J)=0
711   IF(IABS(EP(J)).GT.ERRLIM(J))GO TO 721
      IF(GIMLIM(J).EQ.0)GO TO 701
721   IDUMMY=ISIGN(1,EP(J))+ISIGN(1,W(J)-WC(J))
      TJCNT(J)=0
      IF(IDUMMY)731,701,741
731   JET(J)=1
      GO TO 701
741   JET(J)=-1
701   CONTINUE
      DO 148 I = 1,16
148   XOUT(I)=(BDOTC(I)/QOUT(I))*QOUT(I)
      DO 145 I = 1,3
      XOUT(I)=XOUT(I)/FLNP2
      XOUT(I+3)=XOUT(I+3)/FLNP2
```

```
      XOUT(I+7)=XOUT(I+7)/FLNM2
      XOUT(I+10)=XOUT(I+10)/FL2NP1
      XOUT(I+13)=XOUT(I+13)/FLNP5
145   NXOUT(I)=JET(I)
      XOUT(7)=XOUT(7)/FLNPO
      XOUT(9)=XOUT(9)/2.
      RETURN
      END
```

```
$IBFTC S2X      LIST
      SUBROUTINE COMPX
C
      COMMON/SINCO/SPACE(30),DIRCO(3,3)
      COMMON/FLOTIN/XIN(25),NXIN(12)
      COMMON/FLOOUT/XOUT(16),NXOUT(3)
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/SCALER/CSCALE,DSCALE,LSCALE,LLSCAL,MSCALE,NSCALE,PSCALE,
     CQSCALE,RSCALE,TSCALE,VSCALE
      INTEGER CSCALE,DSCALE,PSCALE,QSCALE,RSCALE,TSCALE,VSCALE
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NP0,FL2NP1
      COMMON/MISCFL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/FIXIN/W(3),E(2),A(2),PH,RH,BETA(3),ALPHA(3),EDOTC(3),EC(3),
     CTM,HCL(3),MODE,RATEFB,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER A,ALPHA,BETA,E,EC,EDOTC,HCL    ,PH,RATEFB,RH,TM,UPDATE,W,
      COMMON/QUANT/QIN(25),QOUT(16),NEXPNT
      INTEGER QIN,QOUT
      COMMON/FIXOUT/BDOTC(3),ADOTC(3),AC,ED(3),EP(3),WC(3),JET(3)
      INTEGER AC,ADOTC,BDOTC,ED,EP,WC
      COMMON/EXP1V/ACDBL,ACDOT,ADSAVE,COSAC,COSWT,          OMEGA(3),
     COMEGAE(3),RO(3),S,SDOT,SDUM,SINAC,SINWT,SPRIME,SREL(3),SSQ,TANAC,
     CV(3),VC(3),VDOUB(3),WE,WE2,WE3,WE4
      INTEGER ACDBL,ACDOT,ADSAVE,COSAC,COSWT,      OMEGA,OMEGAE,RO,S,
     CSDOT,SDUM,SINAC,SINWT,SPRIME,SREL,SSQ,TANAC,V,VC,VDOUB,WE,WE2,WE3,
     CWE4
      COMMON/EXP2V/DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,
     CU2X,U1Z,U2Z
      INTEGER DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,U2X,
     CU1Z,U2Z
      COMMON/EXP3V/ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,KC,PREV,PSP(3),R,SINDUM,SINL,SINLR,SIN2LR,SINTH,
     CSPX(2),SPY(2),SPZ(2),S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,
     CZEYI,ZEZI,EPZDOT
```

```
      INTEGER ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,    PREV,PSP,R,SINDUM,SINL,SINLR,SIN2LR,SINTH,SPX,SPY,
     CSPZ,S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,ZEYI,ZEZI,EPZDOT
      COMMON/MOD567/DELE(3),ECOM(3),EDCOM(3),TENDM,ECOMDP(3),DELTAE(3),
     CMAX,MAXRT ,TP,NHHH
      INTEGER DELE,ECOM,EDCOM,TENDM,ECOMDP,DELTAE,TP
      COMMON/CONTL1/EPPREV(3),GAIN(6,5),GAINP(6,5),EPP(3),HNOM ,KLCL(6),
     C     MAGA(3),MAGB(3),MCA(3),MCB(3),TRQC(3),TRQCP(3),UNITVA(3,3),
     CUNITVB(3,3),ZO(5)
      INTEGER EPPREV,GAIN,GAINP,EPP,HNOM,TRQC,TRQCP,UNITVA,UNITVB,ZO
      COMMON/CONTL2/DELA(3),DELB(3),DOT1(3),DOT2(3),DOT3(3),DUM1,DUM2,
     CKSAVE,MAGASQ(3),MAGBSQ(3),TREM(3),TRQPRD(3),UNITVH(3,3)
     C ,BHOLD,BSELFD,BDOTDS,MBMAX
      INTEGER DELA,DELB,DOT1,DOT2,DOT3,DUM1,DUM2,TREM,TRQPRD,UNITVH
     C ,BHOLD,BSELFD,BDOTDS
      INTEGER TRQCPP(3)
      EQUIVALENCE(TRQCPP(1),TRQPRD(1))
      COMMON/CONTL3/CGYRO(3),COSTA(3),COSTB(3),RTEST,RUSE,TDESSQ,
     CTDOTA(3),TDOTB(3),TREMSQ,XKEND,ITER,NOITER
      INTEGER CGYRO,COSTA,COSTB,RTEST,RUSE,TDESSQ,TDOTA,TDOTB,TREMSQ,
     CXKEND
      COMMON/DESAT/ERRLIM(3),JETCT(3),TJCNT(3),GIMLIM(3),BDOTMX,ADOTMX
      INTEGER TJCNT,ERRLIM,GIMLIM,BDOTMX,ADOTMX
      COMMON/TVECT/DELT,H,T(10),NPASS,NSLOW,XNSLOW
      INTEGER DELT,H,T
      COMMON/NAV/F(10),FDOT,FDUM(10),FTOT,G(10),GDOT,GDUM(10),GTOT,P(3),
     CPO(3),PDOT(3),PDOTO(3)
      INTEGER F,FDOT,FDUM,FTOT,G,GDOT,GDUM,GTOT,P,PO,PDOT,PDOTO
      COMMON/DIRCOS/MS(3,3),MSDOT(3,3),ML(3,3),MLDOT(3,3),DSAVE(3,3),
     CMSDBL(3,3),MLDBL(3,3),XW(3),WPREV(3),TRATIO(3),DELW(3)
      INTEGER DSAVE,XW,WPREV,TRATIO,DELW
      COMMON/NTRIG/COSE(2),COSA(2),CBETA(3),CALPHA(3),SINE(2),SINA(2),
     CSBETA(3),SALPHA(3),SINED(1),COSED(1)
      INTEGER COSA,COSE,CALPHA,CBETA,SINA,SINE,SALPHA,SBETA,SINED,COSED
      INTEGER DMULT,COST,DUM
C
C BODY RATE SELECTION AND BASELINE CONTROL LAW
C
```

```
      NFB=0
      IF(RATEFB.EQ.1)NFB=1
      DO 71 I = 1,3
      EPP(I) = EP(I) - DMULT(ZO(NGAIN)*NPO,EPPREV(I))*(1-NFB)
   71 EPPREV(I) = EP(I)
      IF(N.GE.14)GO TO 94
      DO 74 I = 1,3
      KLCL(I) = DMULT(-GAIN(I,NGAIN)*NPO,EPP(I))*VSCALE - (GAINP(I,NGAIN
     C)*(W(I)-WC(I))*NFB)/NM7
   74 KLCL(I+3) = DMULT(GAIN(I+3,NGAIN)*NPO,EPP(I))*VSCALE + (GAINP(I+3,
     CNGAIN)*(W(I)-WC(I))*NFB)/NM7
      GO TO 82
   94 DO 95 I = 1,3
      KLCL(I) = DMULT(-GAIN(I,NGAIN)*NPO,EPP(I))/TSCALE - (GAINP(I,NGAIN
     C)*(W(I)-WC(I))*NFB)/NM7
   95 KLCL(I+3) = DMULT(GAIN(I+3,NGAIN)*NPO,EPP(I))/TSCALE + (GAINP(I+3,
     CNGAIN)*(W(I)-WC(I))*NFB)/NM7
   82 DO 96 J=1,6
   96 IF(IABS(KLCL(J)).GT.FS)KLCL(J)=ISIGN(FS,KLCL(J))
      IF(LAW.GE.2)GO TO 73
      ADOTC(1) =(KLCL(1)*SALPHA(1))/NPO + (KLCL(3)*CALPHA(1))/NPO
      ADOTC(2) =(KLCL(2)*SALPHA(2))/NPO + (KLCL(1)*CALPHA(2))/NPO
      ADOTC(3) =(KLCL(3)*SALPHA(3))/NPO + (KLCL(2)*CALPHA(3))/NPO
      BDOTC(1) =(KLCL(5)*CBETA(1))/NPO
      BDOTC(2) =(KLCL(6)*CBETA(2))/NPO
      BDOTC(3) =(KLCL(4)*CBETA(3))/NPO
      GO TO 108
C
C PRELIMINARIES TO CONTROL LAWS 2 AND 3
C
   73 DO 75 I = 1,3
      UNITVA(I,I) = SALPHA(I)
   75 UNITVB(I,I) = (CALPHA(I)*SBETA(I))/NPO
      UNITVA(1,2) = 0
      UNITVA(1,3) = CALPHA(1)
      UNITVA(2,1) = CALPHA(2)
      UNITVA(2,3) = 0
      UNITVA(3,1) = 0
```

```
      UNITVA(3,2) = CALPHA(3)
      UNITVB(1,2) =-CBETA(1)
      UNITVB(1,3) =(-SALPHA(1)*SBETA(1))/NPO
      UNITVB(2,1) =(-SALPHA(2)*SBETA(2))/NPO
      UNITVB(2,3) =-CBETA(2)
      UNITVB(3,1) =-CBETA(3)
      UNITVB(3,2) =-(SALPHA(3)*SBETA(3))/NPO
      MBMAX=1
      DO 79 I = 1,3
      MAGA(I) = (HCL(I)*CBETA(I))/NPO
      MAGB(I) = HCL(I)
      TRQC(I)=(HNOM*KLCL(I))/NM1
      IF(IABS(TRQC(I)).GT.FS)TRQC(I) = ISIGN(FS,TRQC(I))
      IF(IABS(BETA(I)).GT.IABS(BETA(MBMAX)))MBMAX=I
   79 TRQCP(I)=TRQC(I)
      IF(IABS(BETA(MBMAX)).GE.BSELFD)GO TO 174
      MBMAX=0
      GO TO 173
174   BDOTC(MBMAX)=0
173   IF(MODCOM.EQ.0)GO TO 163
      MCB(1)=(HCL(1)*((((W(2)*SALPHA(1))/NPO)*CBETA(1))/NPO+(W(3)*SBETA
     1 (1))/NPO))/NPO+(HCL(2)*((((W(3)*CALPHA(2))/NPO)*CBETA(2))/NPO-(
     2 W(2)*SBETA(2))/NPO))/NPO-(((HCL(3)*CBETA(3))/NPO)*((W(2)*CALPHA
     3 (3))/NPO+(W(3)*SALPHA(3))/NPO))/NPO
      MCB(2)=(HCL(2)*((((W(3)*SALPHA(2))/NPO)*CBETA(2))/NPO+(W(1)*SBETA
     1 (2))/NPO))/NPO+(HCL(3)*((((W(1)*CALPHA(3))/NPO)*CBETA(3))/NPO-(
     2 W(3)*SBETA(3))/NPO))/NPO-(((HCL(1)*CBETA(1))/NPO)*((W(3)*CALPHA
     3 (1))/NPO+(W(1)*SALPHA(1))/NPO))/NPO
      MCB(3)=(HCL(3)*((((W(1)*SALPHA(3))/NPO)*CBETA(3))/NPO+(W(2)*SBETA
     1 (3))/NPO))/NPO+(HCL(1)*((((W(2)*CALPHA(1))/NPO)*CBETA(1))/NPO-(
     2 W(1)*SBETA(1))/NPO))/NPO-(((HCL(2)*CBETA(2))/NPO)*((W(1)*CALPHA
     3 (2))/NPO+(W(2)*SALPHA(2))/NPO))/NPO
      DO 85 I = 1,3
   85 TRQCP(I)=TRQCP(I)-MCB(I)/4
C
C CONTROL LAW 2 (ALGEBRAIC )
C
163   IF(LAW.NE.2)GO TO 109
```

```
      DO 110 I = 1,3
      DELB(I) = 0
      DELA(I) = 0
  110 UNITVH(I,I) = (CBETA(I)*CALPHA(I))/NPO
      UNITVH(1,2) = SBETA(1)
      UNITVH(1,3) = -(CBETA(1)*SALPHA(1))/NPO
      UNITVH(2,1) = -(CBETA(2)*SALPHA(2))/NPO
      UNITVH(2,3) = SBETA(2)
      UNITVH(3,1) = SBETA(3)
      UNITVH(3,2) = -(CBETA(3)*SALPHA(3))/NPO
      DO 111 I = 1,3
      TRQPRD(I) = 0
      DO 112 J = 1,3
  112 TRQPRD(I) = TRQPRD(I) +(((UNITVB(J,I)*BDOTC(J))/NPO)*MAGB(J))/NM1+
     C(((UNITVA(J,I)*ADOTC(J))/NPO)*MAGA(J))/NM1
  111 TRQCPP(I) = TRQCP(I) - (KSAVE*TRQPRD(I))/NPO
      IF(MBMAX.EQ.0)GO TO 175
      DELB(MBMAX)=ISIGN(BDOTDS,-BETA(MBMAX))
      DUM=(DELB(MBMAX)*MAGB(MBMAX))/NM1
      DO 176 J=1,3
176   TRQCPP(J)=TRQCPP(J)-(DUM*UNITVB(MBMAX,J))/NPO
175   DO 113 J = 1,3
      ADOTC(J) = (KSAVE*ADOTC(J))/NPO
      BDOTC(J) = (KSAVE*BDOTC(J))/NPO
      DOT1(J) = 0
      DO 113 I = 1,3
  113 DOT1(J) = DOT1(J) + (TRQCPP(I)*UNITVH(J,I))/NPO
      M1 = 1
      IF(MBMAX.EQ.1)M1=2
      DO 114 J = 2,3
114   IF(IABS(DOT1(J)).LT.IABS(DOT1(M1)).AND.J.NE.MBMAX)M1=J
      DO 115 J= 1,3
      DOT2(J) = 0
      DOT3(J) = 0
      DO 116 I = 1,3
      DOT2(J) = DOT2(J) + (UNITVH(M1,I)*UNITVA(J,I))/NPO
  116 DOT3(J) = DOT3(J) + (UNITVH(M1,I)*UNITVB(J,I))/NPO
      DOT2(J) = (MAGA(J)*DOT2(J))/NPO
```

```
  115 DOT3(J) = (MAGB(J)*DOT3(J))/NPO
      DOT2(M1) = 0
      DOT3(M1) = 0
      M2A = 1
      M2B = 1
      IF(MBMAX.EQ.1)M2B=2
      DO 117 J = 2,3
      IF(IABS(DOT2(J)).GT.IABS(DOT2(M2A)))M2A = J
117   IF(IABS(DOT3(J)).GT.IABS(DOT3(M2B)).AND.J.NE.MBMAX)M2B=J
      IF(IABS(DOT2(M2A))-IABS(DOT3(M2B)))118,118,119
  119 DELA(M2A) = (DOT1(M1)*NM1)/DOT2(M2A)
      DUM1 = 0
      DUM2 = 0
      DO 120 J = 1,3
      DELB(M1) = DELB(M1) + (TRQCPP(J)*UNITVB(M1,J))/NPO
      DUM1 = DUM1 + (UNITVA(M2A,J)*UNITVB(M1,J))/NPO
      DELA(M1) = DELA(M1) + (TRQCPP(J)*UNITVA(M1,J))/NPO
  120 DUM2 = DUM2 + (UNITVA(M2A,J)*UNITVA(M1,J))/NPO
      DELB(M1) = ((DELB(M1)/2 - (((DELA(M2A)*DUM1)/NPO)*MAGA(M2A))/NPO)*
     CNPO)/MAGB(M1)
      DELA(M1) = ((DELA(M1)/2 - (((DELA(M2A)*DUM2)/NPO)*MAGA(M2A))/NPO)*
     CNPO)/MAGA(M1)
      GO TO 121
  118 DELB(M2B) = (DOT1(M1)*NM1)/DOT3(M2B)
      DUM1 = 0
      DUM2 = 0
      DO 122 J = 1,3
      DELB(M1) = DELB(M1) + (TRQCPP(J)*UNITVB(M1,J))/NPO
      DUM1 = DUM1 + (UNITVB(M2B,J)*UNITVB(M1,J))/NPO
      DELA(M1) = DELA(M1) + (TRQCPP(J)*UNITVA(M1,J))/NPO
  122 DUM2 = DUM2 + (UNITVB(M2B,J)*UNITVA(M1,J))/NPO
      DELB(M1) = ((DELB(M1)/2 - (((DELB(M2B)*DUM1)/NPO)*MAGB(M2B))/NPO)*
     CNPO)/MAGB(M1)
      DELA(M1) = ((DELA(M1)/2 - (((DELB(M2B)*DUM2)/NPO)*MAGB(M2B))/NPO)*
     CNPO)/MAGA(M1)
  121 DO 123 J = 1,3
      ADOTC(J) = ADOTC(J) + DELA(J)
  123 BDOTC(J) = BDOTC(J) + DELB(J)
```

```
      GO TO 108
C
C CONTROL LAW 3 (ITERATIVE)
C
  109 TREMSQ = 0
      TDESSQ=0
      DO 125 J=1,3
      ADOTC(J) = 0
      BDOTC(J) = 0
      TREM(J) = TRQCP(J)
      MAGASQ(J) =(MAGA(J)**2)/NPO
      MAGBSQ(J) =(MAGB(J)**2)/NPO
      DO 125 I = 1,3
      UNITVA(J,I) = (MAGA(J)*UNITVA(J,I))/NPO
  125 UNITVB(J,I) = (MAGB(J)*UNITVB(J,I))/NPO
      IF(MBMAX.EQ.0)GO TO 177
      BDOTC(MBMAX)=ISIGN(BDOTDS,-BETA(MBMAX))
      DO 178 J=1,3
178   TREM(J)=TREM(J)-(BDOTC(MBMAX)*UNITVB(MBMAX,J))/NM1
177   DO 124 J=1,3
      TREMSQ = TREMSQ + (TREM(J)**2)/NPO
      TDESSQ=TDESSQ+(TRQC(J)**2)/NPO
  124 CGYRO(J) = COST(ADOTC(J),BDOTC(J),J)
      DO 126 K = 1,ITER
      DO 127 J = 1,3
      TDOTA(J) = 0
      TDOTB(J) = 0
      DO 127 I = 1,3
      TDOTA(J) = TDOTA(J) + (TREM(I)*UNITVA(J,I))/NPO
  127 TDOTB(J) = TDOTB(J) + (TREM(I)*UNITVB(J,I))/NPO
      MINA = 1
      MINB = 1
      IF(MBMAX.EQ.1)MINB=2
      DO 128 J = 1,3
      IF(TDOTA(J))129,130,129
  129 RTEST = (TREMSQ*NM1)/TDOTA(J)
      COSTA(J) = COST(ADOTC(J)+RTEST,BDOTC(J),J) - CGYRO(J)
      IF(COSTA(J).LT.COSTA(MINA))MINA = J
```

```
      GO TO 131
  130 COSTA(J) = FS
  131 IF(TDOTB(J))132,133,132
  132 RTEST = (TREMSQ*NM1)/TDOTB(J)
      COSTB(J) = COST(ADOTC(J),BDOTC(J)+RTEST,J) - CGYRO(J)
      IF(COSTB(J).LT.COSTB(MINB).AND.J.NE.MBMAX)MINB=J
      GO TO 128
  133 COSTB(J) = FS
  128 CONTINUE
      IF(COSTA(MINA)-COSTB(MINB))134,134,135
  134 RUSE = (TDOTA(MINA)*NM1)/MAGASQ(MINA)
      DO 136 J = 1,3
  136 TREM(J) = TREM(J) - (RUSE*UNITVA(MINA,J))/NM1
      ADOTC(MINA) = ADOTC(MINA) + RUSE
      CGYRO(MINA) = COST(ADOTC(MINA),BDOTC(MINA),MINA)
      GO TO 137
  135 RUSE = (TDOTB(MINB)*NM1)/MAGBSQ(MINB)
      DO 138 J = 1,3
  138 TREM(J) = TREM(J) - (RUSE*UNITVB(MINB,J))/NM1
      BDOTC(MINB) = BDOTC(MINB) + RUSE
      CGYRO(MINB) = COST(ADOTC(MINB),BDOTC(MINB),MINB)
  137 TREMSQ = 0
      DO 139 J = 1,3
  139 TREMSQ = TREMSQ + (TREM(J)**2)/NPO
      IF(TREMSQ.LT.(XKEND*TDESSQ)/NPO)GO TO 154
  126 CONTINUE
      NOITER = ITER
      GO TO 108
  154 NOITER = J
C
C DESATURATION
C
108   DO 160 J=1,3
      IF(IABS(BDOTC(J)).GT.BDOTMX)BDOTC(J)=ISIGN(BDOTMX,BDOTC(J))
160   IF(IABS(ADOTC(J)).GT.ADOTMX)ADOTC(J)=ISIGN(ADOTMX,ADOTC(J))
      RETURN
      END
```

```
$IBFTC S3       LIST
      SUBROUTINE COMP
C
C
      COMMON/SINCO/SPACE(30),DIRCO(3,3)
      COMMON/FLOTIN/XIN(25),NXIN(12)
      COMMON/FLOOUT/XOUT(16),NXOUT(3)
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      COMMON/FIXIN/W(3),E(2),A(2),PH,RH,BETA(3),ALPHA(3),EDOTC(3),EC(3),
     CTM,HCL(3),MODE,RATEFB,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER RATEFB,UPDATE
      COMMON/FIXOUT/BDOTC(3),ADOTC(3),AC,ED(3),EP(3),WC(3),JET(3)
      COMMON/EXP1V/ACDBL,ACDOT,ADSAVE,COSAC,COSWT,          OMEGA(3),
     COMEGAE(3),RO(3),S,SDOT,SDUM,SINAC,SINWT,SPRIME,SREL(3),SSQ,TANAC,
     CV(3),VC(3),VDOUB(3),WE,WE2,WE3,WE4
      COMMON/EXP2V/DEL,DELP,DEL1,DEL2,DEL3,S1X,S2X,S1Y,S2Y,S1Z,S2Z,U1X,
     CU2X,U1Z,U2Z
      COMMON/EXP3V/ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,KC,PREV,PSP(3),R,SINDUM,SINL,SINLR,SIN2LR,SINTH,
     CSPX(2),SPY(2),SPZ(2),S1XG,S1ZG,S2ZG,                 WBAR,Z,ZEXI,
     CZEYI,ZEZI,EPZDOT
      REAL KC
      COMMON/MOD567/DELF(3),ECOM(3),EDCOM(3),TENDM,ECOMDP(3),DELTAF(3),
     CMAX,MAXRT ,TP,NHHH
      REAL MAX,MAXRT
      COMMON/CONTL1/EPPREV(3),GAIN(6,5),GAINP(6,5),EPP(3),HNOM ,KLCL(6),
     C     MAGA(3),MAGB(3),MCA(3),MCB(3),TRQC(3),TRQCP(3),UNITVA(3,3),
     CUNITVB(3,3),ZO(5)
      REAL KLCL,MAGA,MAGB,MCA,MCB
      COMMON/CONTL2/DELA(3),DELB(3),DOT1(3),DOT2(3),DOT3(3),DUM1,DUM2,
     CKSAVE,MAGASQ(3),MAGBSQ(3),TREM(3),TRQPRD(3),UNITVH(3,3)
     C ,BHOLD,BSELFD,BDOTDS,MBMAX
      REAL KSAVE,MAGASQ,MAGBSQ
      DIMENSION TRQCPP(3)
      EQUIVALENCE(TRQCPP(1),TRQPRD(1))
      COMMON/CONTL3/CGYRO(3),COSTA(3),COSTB(3),RTEST,RUSE,TDESSQ,
     CTDOTA(3),TDOTB(3),TREMSQ,XKEND,ITER,NOITER
      COMMON/DESAT/ERRLIM(3),JETCT(3),TJCNT(3),GIMLIM(3),BDOTMX,ADOTMX
```

```
      INTEGER TJCNT,GIMLIM
      COMMON/TVECT/DELT,H,T(10),NPASS,NSLOW,XNSLOW
      COMMON/NAV/F(10),FDOT,FDUM(10),FTOT,G(10),GDOT,GDUM(10),GTOT,P(3),
     CPO(3),PDOT(3),PDOTO(3)
      COMMON/DIRCOS/MS(3,3),MSDOT(3,3),ML(3,3),MLDOT(3,3),DSAVE(3,3),
     CMSDBL(3,3),MLDBL(3,3),XW(3),WPREV(3),TRATIO(3),DELW(3)
      REAL MS,MSDOT,ML,MLDOT,MSDBL,MLDBL
      COMMON/NTRIG/COSE(2),COSA(2),CBETA(3),CALPHA(3),SINE(2),SINA(2),
     CSBETA(3),SALPHA(3),SINED(1),COSED(1)
      DATA TWOPI/6.2831853E0/
C
C EQUATE THE INPUT VECTORS
C
      DO 140 I = 1,37
  140 W(I) = XIN(I)
C
C DIRECTION COSINE INTEGRATION
C
      K=1
      DO 179 J=1,3
      DELW(J)= TRATIO(J)*(W(J)-WPREV(J))
179   XW(J)=WPREV(J)+DELW(J)
   34 DO 19 J = 1,2
      MSDOT(J,1) =XW(3)*MS(J,2) -XW(2)*MS(J,3)
      MSDOT(J,2) =XW(1)*MS(J,3) -XW(3)*MS(J,1)
   19 MSDOT(J,3) =XW(2)*MS(J,1) -XW(1)*MS(J,2)
      IF(K.GE.2)GO TO 31
      DO 33 J=1,3
      WPREV(J)=W(J)
      XW(J)=W(J)+DELW(J)
      DO 33 I = 1,2
      DSAVE(I,J) = MSDOT(I,J)
   33 MS(I,J) = MSDBL(I,J) + H*MSDOT(I,J)
      K=2
      GO TO 34
   31 IF(UPDATE.NE.2)GO TO 32
      DO 191 J=1,3
      DO 191 I=1,3
```

```
      MSDBL(I,J)=DIRCO(I,J)
  191 MS(I,J)=MSDBL(I,J)
      NXIN(3)=0
      GO TO 190
   32 DO 35 I = 1,2
      DO 35 J = 1,3
      MSDBL(I,J) =  MSDBL(I,J) +(H/2.)*(MSDOT(I,J)+DSAVE(I,J))
   35 MS(I,J) = MSDBL(I,J)
      MS(3,1) = MS(1,2)*MS(2,3) - MS(2,2)*MS(1,3)
      MS(3,2) = MS(1,3)*MS(2,1) - MS(2,3)*MS(1,1)
      MS(3,3) = MS(1,1)*MS(2,2) - MS(2,1)*MS(1,2)
C
C EULER ANGLES AND TRIG FUNCTIONS
C
  190 ED(1)=ATAN2(MS(3,2),MS(3,3))
      ED(2)=ASIN(-MS(3,1))
      ED(3)=ATAN2(MS(2,1),MS(1,1))
      DO 27 I = 1,4
      SINE(I) = SIN(E(I))
   27 COSE(I) = COS(E(I))
      DO 28 I = 1,6
      SBETA(I) = SIN(BETA(I))
   28 CBETA(I) = COS(BETA(I))
      SINED(1) = SIN(ED(1))
      COSED(1) = COS(ED(1))
      IF(NPASS.GE.NSLOW)GO TO 170
      IF(MODE-1)171,101,171
C
C POSITION AND VELOCITY
C
170   T(1)=T(1)+DELT
      FTOT=0.
      GTOT=0.
      FDOT=0.
      GDOT=0.
      DO 142 J=2,10
142   T(J)=T(1)*T(J-1)/FLOAT(J)
      DO 165 K=2,10
```

```
      J=12-K
      FDOT=FDOT+F(J)*T(J-1)
      GDOT=GDOT+G(J)*T(J-1)
      FTOT=FTOT+F(J)*T(J)
  165 GTOT=GTOT+G(J)*T(J)
      FTOT=FTOT+1.
      GTOT=GTOT+T(1)
      GDOT=GDOT+1.
      DO 144 J=1,3
      P(J)=FTOT*PO(J)+GTOT*PDOTO(J)
  144 PDOT(J)=FDOT*PO(J)+GDOT*PDOTO(J)
C
C SELECT MODE ON MODE BRANCH
C
      GO TO (101,102,103,104,105,106,107),MODE
C
C EXPERIMENT ONE
C
  101 IF(MODE.EQ.MDLAST)GO TO 20
      K=1
  155 COSWT=WE4*T(4)-WE2*T(2)+1.
      SINWT=-WE3*T(3)+WE*T(1)
      SREL(1)=P(1)-RO(1)*COSWT-RO(3)*SINWT
      SREL(2)=P(2)-RO(2)
      SREL(3)=P(3)-RO(3)*COSWT+RO(1)*SINWT
      VDOUB(1)=PDOT(1)-P(3)*WE
      VDOUB(2)=PDOT(2)
      VDOUB(3)=PDOT(3)+P(1)*WE
      SDUM=0.
      SSQ=0.
      DO 141 J=1,3
      V(J) = VDOUB(J)
      SSQ=SSQ+SREL(J)**2
  141 SDUM=SDUM+SREL(J)*VDOUB(J)
      S=SQRT(SSQ)
      SDOT=SDUM/S
  162 SPRIME = S*COS(AC)
      DO 10 J = 1,3
```

```
      OMEGA(J)=(ML(2,J)*OMEGAE(2))
      VC(J) = 0.
      DO 10 K = 1,3
   10 VC(J) = VC(J) + ML(K,J) * V(K)
      TANAC = SIN(AC)/COS(AC)
      WC(3) = OMEGA(3) + OMEGA(2) * TANAC + VC(1)/SPRIME
      ACDOT = (VC(3)-SDOT*SIN(AC))/SPRIME - OMEGA(1)
      DO 11 J = 1,3
   11 MLDOT(J,1) = (WC(3) * ML(J,2))
      IF(K-2)166,21,26
   20 K=2
166   DO 22 J = 1,3
      DSAVE(J,1) = MLDOT(J,1)
   22 ML(J,1) =  MLDBL(J,1) + H*MLDOT(J,1)
      ADSAVE = ACDOT
      AC = ACDBL + H*ACDOT
      GO TO 23
   21 K=3
      DO 24 J = 1,3
      MLDBL(J,1) = MLDBL(J,1) + H*(MLDOT(J,1)+ DSAVE(J,1))/2.
   24 ML(J,1) = MLDBL(J,1)
      ACDBL = ACDBL + H*(ACDOT + ADSAVE)/2.
      AC = ACDBL
   23 ML(1,2) = ML(3,1)*ML(2,3) - ML(2,1)*ML(3,3)
      ML(2,2) = ML(1,1)*ML(3,3) - ML(3,1)*ML(1,3)
      ML(3,2) = ML(2,1)*ML(1,3) - ML(1,1)*ML(2,3)
      IF(K-2)167,155,162
167   K=2
      GO TO 162
   26 EP(1) = MS(1,2)*ML(1,3)+MS(2,2)*ML(2,3)+MS(3,2)*ML(3,3)
      EP(2) = MS(1,3)*ML(1,1)+MS(2,3)*ML(2,1)+MS(3,3)*ML(3,1)
      EP(3) = MS(1,1)*ML(1,2)+MS(2,1)*ML(2,2)+MS(3,1)*ML(3,2)
      WC(1) = 0.
      WC(2) = 0.
      GO TO 100
C
C EXPERIMENT TWO
C
```

```
      U1X = S1XG - S1X
      U1Z = S1ZG - S1Z
      U2Z = S2ZG - S2Z
     DELX = (S2X*U1Z - S1X*U2Z)/DELP
     DELY = (S2Y*U1Z - S1Y*U2Z)/DELP
     DELZ = (DEL1*U1X + DEL2*U1Z + DEL3*U2Z)/DEL
     DO 17 I = 1,3
     MSDBL(1,I)=MSDBL(1,I)-MS(2,I)*DELZ+MS(3,I)*DELY
     MSDBL(2,I)=MS(1,I)*DELZ+MSDBL(2,I)-MS(3,I)*DELX
     DO 17 K = 1,2
   17 MS(K,I) = MSDBL(K,I)
     MS(3,1) = MS(1,2)*MS(2,3) - MS(2,2)*MS(1,3)
     MS(3,2) = MS(1,3)*MS(2,1) - MS(2,3)*MS(1,1)
     MS(3,3) = MS(1,1)*MS(2,2) - MS(2,1)*MS(1,2)
     NXIN(3)=0
   15 ANGLE = ANGLE + DELANG
     IF(ANGLE.GE.4.)ANGLE=ANGLE-TWOPI
     IF(ANGLE.LE.-4.)ANGLE=ANGLE+TWOPI
     WC(1) =  WBAR*SIN(ANGLE)
     WC(2) = WBAR*COS(ANGLE)
     WC(3) = ETADOT
     EP(1) = MS(1,2)*ZEXI + MS(2,2)*ZEYI + MS(3,2)*ZEZI
     EP(2) =-MS(1,1)*ZEXI - MS(2,1)*ZEYI - MS(3,1)*ZEZI
     EPZDOT = W(3) - WC(3)
     EP(3) = H*(EPZDOT + PREV)*XNSLOW/2. +EP(3)
     PREV = EPZDOT
     GO TO 100
C
C EXPERIMENT FOUR
C
104  EP(1) = PH
     EP(2)=W(3)
     EP(3) = RH
     WC(1) = WBAR
     WC(2) = 0.
     WC(3) = 0.
     GO TO 100
C
```

```
C MODE 5
C
  105 DO 50 I = 1,3
   50 WC(I) = 0.
      IF(MODE-MDLAST)51,52,51
   51 DO 53 I = 1,3
      ECOM(I) = ED(I)
   53 EP(I) = 0.
      GO TO 100
   52 DO 55 I = 1,3
   55 DELE(I) =-ECOM(I) + ED(I)
      EP(1) = MS(3,1)*DELE(3) + DELE(1)
      EP(2) = MS(3,2)*DELE(3) + COSED(1)*DELE(2)
      EP(3) = MS(3,3)*DELE(3) - SINED(1)*DELE(2)
      GO TO 100
C
C MODE 6
C
  106 IF(MODE-MDLAST)56,57,56
   56 DO 149 I = 1,3
  149 DELTAE(I) = EC(I) - ED(I)
      MAX = ABS(DELTAE(1))
      DO 150 I = 2,3
      IF(MAX- ABS(DELTAE(I)).LT.0.)MAX = ABS(DELTAE(I))
  150 CONTINUE
      TP = MAX/MAXRT
      TENDM = TM
      IF(TM.LT.TP)TENDM = TP
      TP=0.
      TENDM=(FLOAT(IFIX(TENDM/DELT))+1.)*DELT
      DO 58 I = 1,3
      EDCOM(I)=DELTAE(I)/TENDM
   58 ECOMDP(I) = ED(I)
      TENDM=TENDM-.5*DELT
      GO TO 67
57    TP=TP+DELT
      IF(TP  -TENDM)67,59,59
   67 DO 60 I = 1,3
```

```
      ECOMDP(I) = ECOMDP(I) + DELT*EDCOM(I)
   60 ECOM(I) = ECOMDP(I)
      GO TO 61
   59 DO 62 I = 1,3
      EDCOM(I) = 0.
   62 ECOM(I) = EC(I)
   61 WC(1) = EDCOM(1) + EDCOM(3)*MS(3,1)
      WC(2) = EDCOM(3)*MS(3,2) + EDCOM(2)*COSED(1)
      WC(3) = EDCOM(3)*MS(3,3) - EDCOM(2)*SINED(1)
      GO TO 52
C
C MODE 7
C
  107 IF(MODE-MDLAST)63,64,63
   63 DO 65 I = 1,3
   65 ECOMDP(I) = ED(I)
   64 DO 66 I = 1,3
   66 EDCOM(I) = EDOTC(I)
      GO TO 67
C
C BODY RATE SELECTION AND BASELINE CONTROL LAW
C
100   MDLAST=MODE
      IF(NPASS.LT.NSLOW)GO TO 171
      XFB=0.
      IF(RATEFB.EQ.1)XFB=1.
      DO 74 I = 1,3
      EPP(I) = EP(I) - ZO(NGAIN)*EPPREV(I)*(1.-XFB)
   71 EPPREV(I) = EP(I)
      KLCL(I) = -GAIN(I,NGAIN)*EPP(I)-GAINP(I,NGAIN)*(W(I)-WC(I))*XFB
   74 KLCL(I+3)=GAIN(I+3,NGAIN)*EPP(I)+GAINP(I+3,NGAIN)*(W(I)-WC(I))*XFB
      IF(LAW.GE.2)GO TO 73
      ADOTC(1) = KLCL(1) * SALPHA(1) + KLCL(3) * CALPHA(1)
      ADOTC(2) = KLCL(2) * SALPHA(2) + KLCL(1) * CALPHA(2)
      ADOTC(3) = KLCL(3) * SALPHA(3) + KLCL(2) * CALPHA(3)
      BDOTC(1) = KLCL(5) * CBETA(1)
      BDOTC(2) = KLCL(6) * CBETA(2)
      BDOTC(3) = KLCL(4) * CBETA(3)
```

```
      GO TO 108
C
C PRELIMINARIES TO CONTROL LAWS 2 AND 3
C
   73 DO 75 I = 1,3
      UNITVA(I,I) = SALPHA(I)
   75 UNITVB(I,I) = CALPHA(I)*SBETA(I)
      UNITVA(1,2) = 0.
      UNITVA(1,3) = CALPHA(1)
      UNITVA(2,1) = CALPHA(2)
      UNITVA(2,3) = 0.
      UNITVA(3,1) = 0.
      UNITVA(3,2) = CALPHA(3)
      UNITVB(1,2) = -CBETA(1)
      UNITVB(1,3) = -SALPHA(1)*SBETA(1)
      UNITVB(2,1) = -SALPHA(2)*SBETA(2)
      UNITVB(2,3) = -CBETA(2)
      UNITVB(3,1) = -CBETA(3)
      UNITVB(3,2) = -SALPHA(3)*SBETA(3)
      MBMAX=1
      DO 79 I = 1,3
      MAGA(I) = HCL(I)*CBETA(I)
      MAGB(I) = HCL(I)
      TRQC(I)=HNOM*KLCL(I)
      IF( ABS(BETA(I)).GT. ABS(BETA(MBMAX)))MBMAX=I
   79 TRQCP(I) = TRQC(I)
      IF( ABS(BETA(MBMAX)).GE.BSELFD)GO TO 174
      MBMAX=0
      GO TO 173
174   BDOTC(MBMAX)=0.
173   IF(MODCOM.EQ.0)GO TO 163
      MCB(1)=HCL(1)*(W(2)*SALPHA(1)*CBETA(1)+W(3)*SBETA(1))+HCL(2)*(W(3)
     1 *CALPHA(2)*CBETA(2)-W(2)*SBETA(2))-HCL(3)*CBETA(3)*(W(2)*CALPHA
     2 (3)+W(3)*SALPHA(3))
      MCB(2)=HCL(2)*(W(3)*SALPHA(2)*CBETA(2)+W(1)*SBETA(2))+HCL(3)*(W(1)
     1 *CALPHA(3)*CBETA(3)-W(3)*SBETA(3))-HCL(1)*CBETA(1)*(W(3)*CALPHA
     2 (1)+W(1)*SALPHA(1))
      MCB(3)=HCL(3)*(W(1)*SALPHA(3)*CBETA(3)+W(2)*SBETA(3))+HCL(1)*(W(2)
```

```
     1 *CALPHA(1)*CBETA(1)-W(1)*SBETA(1))-HCL(2)*CBETA(2)*(W(1)*CALPHA
     2 (2)+W(2)*SALPHA(2))
      DO 85 I = 1,3
   85 TRQCP(I)=TRQCP(I)-MCB(I)
C
C CONTROL LAW 2 (ALGEBRAIC )
C
163   IF(LAW.NE.2)GO TO 109
      DO 110 I = 1,3
      DELB(I) = 0.
      DELA(I) = 0.
  110 UNITVH(I,I) = CBETA(I)*CALPHA(I)
      UNITVH(1,2) = SBETA(1)
      UNITVH(1,3) =-CBETA(1)*SALPHA(1)
      UNITVH(2,1) =-CBETA(2)*SALPHA(2)
      UNITVH(2,3) = SBETA(2)
      UNITVH(3,1) = SBETA(3)
      UNITVH(3,2) =-CBETA(3)*SALPHA(3)
      DO 111 I = 1,3
      TRQPRD(I) = 0.
      DO 112 J = 1,3
  112 TRQPRD(I) = TRQPRD(I) + UNITVB(J,I)*BDOTC(J)*MAGB(J) + UNITVA(J,I)
     C*ADOTC(J)*MAGA(J)
  111 TRQCPP(I) = TRQCP(I) - KSAVE*TRQPRD(I)
      IF(MBMAX.EQ.0)GO TO 175
      DELB(MBMAX)= SIGN(BDOTDS,-BETA(MBMAX))
      DUMMY=DELB(MBMAX)*MAGB(MBMAX)
      DO 176 J=1,3
176   TRQCPP(J)=TRQCPP(J)-DUMMY*UNITVB(MBMAX,J)
175   DO 113 J = 1,3
      ADOTC(J) = KSAVE*ADOTC(J)
      BDOTC(J) = KSAVE*BDOTC(J)
      DOT1(J) = 0.
      DO 113 I = 1,3
  113 DOT1(J) = DOT1(J) + TRQCPP(I)*UNITVH(J,I)
      M1 = 1
      IF(MBMAX.EQ.1)M1=2
      DO 114 J = 2,3
```

```
114   IF(ABS(DOT1(J)).LT.ABS(DOT1(M1)).AND.J.NE.MBMAX)M1 =J
      DO 115 J = 1,3
      DOT2(J) = 0.
      DOT3(J) = 0.
      DO 116 I = 1,3
      DOT2(J) = DOT2(J) + UNITVH(M1,I)*UNITVA(J,I)
  116 DOT3(J) = DOT3(J) + UNITVH(M1,I)*UNITVB(J,I)
      DOT2(J) = MAGA(J)*DOT2(J)
  115 DOT3(J) = MAGB(J)*DOT3(J)
      DOT2(M1) = 0.
      DOT3(M1) = 0.
      M2A = 1
      M2B = 1
      IF(MBMAX.EQ.1)M2B=2
      DO 117 J = 2,3
      IF(ABS(DOT2(J)).GT.ABS(DOT2(M2A)))M2A = J
117   IF(ABS(DOT3(J)).GT.ABS(DOT3(M2B)).AND.J.NE.MBMAX)M2B=J
      IF(ABS(DOT2(M2A))-ABS(DOT3(M2B)))118,118,119
  119 DELA(M2A) = DOT1(M1)/DOT2(M2A)
      DUM1 = 0.
      DUM2 = 0.
      DO 120 J = 1,3
      DELB(M1) = DELB(M1) + TRQCPP(J)*UNITVB(M1,J)
      DUM1 = DUM1 + UNITVA(M2A,J)*UNITVB(M1,J)
      DELA(M1) = DELA(M1) + TRQCPP(J)*UNITVA(M1,J)
  120 DUM2 = DUM2 + UNITVA(M2A,J)*UNITVA(M1,J)
      DELB(M1)= (DELB(M1)-DELA(M2A)*DUM1*MAGA(M2A))/MAGB(M1)
      DELA(M1)= (DELA(M1)-DELA(M2A)*DUM2*MAGA(M2A))/MAGA(M1)
      GO TO 121
  118 DELB(M2B) = DOT1(M1)/DOT3(M2B)
      DUM1 = 0.
      DUM2 = 0.
      DO 122 J = 1,3
      DELB(M1) = DELB(M1) + TRQCPP(J)*UNITVB(M1,J)
      DUM1 = DUM1 + UNITVB(M2B,J)*UNITVB(M1,J)
      DELA(M1) = DELA(M1) + TRQCPP(J)*UNITVA(M1,J)
  122 DUM2 = DUM2 + UNITVB(M2B,J)*UNITVA(M1,J)
      DELB(M1) =(DELB(M1)-DELB(M2B)*DUM1*MAGB(M2B))/MAGB(M1)
```

```
      DELA(M1) =(DELA(M1)-DELB(M2B)*DUM2*MAGB(M2B))/MAGA(M1)
  121 DO 123 J = 1,3
      ADOTC(J) = ADOTC(J) + DELA(J)
  123 BDOTC(J) = BDOTC(J) + DELB(J)
      GO TO 108
C
C CONTROL LAW 3 (ITERATIVE)
C
  109 TREMSQ = 0.
      TDESSQ=0.
      DO 125 J=1,3
      ADOTC(J) = 0.
      BDOTC(J) = 0.
      TREM(J) = TRQCP(J)
      MAGASQ(J) = MAGA(J)**2
      MAGBSQ(J) = MAGB(J)**2
      DO 125 I = 1,3
      UNITVA(J,I) = MAGA(J)*UNITVA(J,I)
  125 UNITVB(J,I) = MAGB(J)*UNITVB(J,I)
      IF(MBMAX.EQ.0)GO TO 177
      BDOTC(MBMAX)= SIGN(BDOTDS,-BETA(MBMAX))
      DO 178 J=1,3
178   TREM(J)=TREM(J)- BDOTC(MBMAX)*UNITVB(MBMAX,J)
177   DO 124 J=1,3
      TREMSQ = TREMSQ + TREM(J)**2
      TDESSQ=TDESSQ+(TRQC(J)**2)
  124 CGYRO(J) = COST(ADOTC(J),BDOTC(J),J)
      DO 126 K = 1,ITER
      DO 127 J = 1,3
      TDOTA(J) = 0.
      TDOTB(J) = 0.
      DO 127 I = 1,3
      TDOTA(J) = TDOTA(J) + TREM(I)*UNITVA(J,I)
  127 TDOTB(J) = TDOTB(J) + TREM(I)*UNITVB(J,I)
      MINA = 1
      MINB = 1
      IF(MBMAX.EQ.1)MINB=2
      DO 128 J = 1,3
```

```
      IF(TDOTA(J))129,130,129
  129 RTEST = TREMSQ/TDOTA(J)
      COSTA(J)=COST(ADOTC(J)+RTEST,BDOTC(J),J)- CGYRO(J)
      IF(COSTA(J).LT.COSTA(MINA))MINA = J
      GO TO 131
  130 COSTA(J) = 1.E10
  131 IF(TDOTB(J))132,133,132
  132 RTEST = TREMSQ/TDOTB(J)
      COSTB(J) = COST(ADOTC(J),BDOTC(J)+RTEST,J)- CGYRO(J)
      IF(COSTB(J).LT.COSTB(MINB).AND.J.NE.MBMAX)MINB=J
      GO TO 128
  133 COSTB(J) = 1.E10
  128 CONTINUE
      IF(COSTA(MINA)-COSTB(MINB))134,134,135
  134 RUSE = TDOTA(MINA)/MAGASQ(MINA)
      DO 136 J = 1,3
  136 TREM(J) = TREM(J) - RUSE*UNITVA(MINA,J)
      ADOTC(MINA) = ADOTC(MINA) + RUSE
      CGYRO(MINA) = COST(ADOTC(MINA),BDOTC(MINA),MINA)
      GO TO 137
  135 RUSE = TDOTB(MINB)/MAGBSQ(MINB)
      DO 138 J = 1,3
  138 TREM(J) = TREM(J) - RUSE*UNITVB(MINB,J)
      BDOTC(MINB) = BDOTC(MINB) + RUSE
      CGYRO(MINB) = COST(ADOTC(MINB),BDOTC(MINB),MINB)
  137 TREMSQ = 0.
      DO 139 J = 1,3
  139 TREMSQ = TREMSQ + TREM(J)**2
      IF(TREMSQ.LT.(XKEND*TDESSQ))GO TO 154
  126 CONTINUE
      NOITER = ITER
      GO TO 108
  154 NOITER = J
C
C DESATURATION
C
108   DO 160 J=1,3
      IF( ABS(BDOTC(J)).GT.BDOTMX)BDOTC(J)= SIGN(BDOTMX,BDOTC(J))
```

```
160    IF( ABS(ADOTC(J)).GT.ADOTMX)ADOTC(J)= SIGN(ADOTMX,ADOTC(J))
171    NPASS=NPASS+1
       IF(NPASS.GT.NSLOW)NPASS=1
       GIMLIM(1)=LIMIG(3)
       GIMLIM(2)=LIMIG(1)
       GIMLIM(3)=LIMIG(2)
       DO 701 J = 1,3
       IF(JET(J).EQ.0)GO TO 711
       TJCNT(J)=TJCNT(J)+1
       IF(TJCNT(J).LT.JETCT (J))GO TO 701
       JET(J) = 0
711    IF(ABS(EP(J)).GT.ERRLIM(J))GO TO 721
       IF(GIMLIM(J).EQ.0)GO TO 701
721    DUMMY = SIGN(1.,EP(J)) + SIGN(1.,W(J)-WC(J))
       TJCNT(J) = 0
       IF(DUMMY)731,701,741
731    JET(J) = 1
       GO TO 701
741    JET(J)= -1
701    CONTINUE
C
C EQUATE THE OUTPUT VECTORS
C
       DO 145 I = 1,19
  145  XOUT(I) = BDOTC(I)
       RETURN
       END
```

Subroutine RKII

Entry points. — RKII, RKRS, RKIC.

Called by. — MAIN, INITIL.

Calls. — EXPER, FINDBP, EVDERP, DERIV, ENERGY, FINDMG.

Function. — Fourth order Runge-Kutta integrator.

Comments. — Calls to EXPER, FINDBP, EVDERP are to recompute coulomb friction when step size changes. A call to ENERGY performs computations at the end of each complete integration interval H. A call to FINDMG recomputes gimbal servo inputs if there has been a change in the gimbal limit situation.

```
$IBFTC S4       LIST
      SUBROUTINE RKII
      COMMON/RKYV/ SV(79)
      COMMON/RKYVDB/TSTOP,DBLT,DSV(79)
      DOUBLE PRECISION DSV
      COMMON/RKYDV/SVDER(79)
      COMMON/RKC/U(79),UMIN(79),HMIN,HUSE,H,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      DOUBLE PRECISION ERROR(79),SVINC(79),         XK(79),TSTOP,DBLT
      IF(TRETRN.LE.0.)RETURN
      TSTOP=DBLT+TRETRN
      NEND=0
13    ODD=TSTOP-DBLT
      IF(ODD.GT.HUSE)GO TO 1
      H=ODD
      NEND=1
      GO TO 3
1     H=HUSE
3     HALFH=.5*H
      IF(H.EQ.HPREV)GO TO 103
      HPREV=H
      IF(NINT.GT.47)CALL EXPER
      CALL FINDBP
      CALL EVDERP
      DO 104 J=1,NINT
      ERROR(J)=HALFH*SVDER(J)
104   SVINC(J)=ERROR(J)
C****  COMPUTE K2
103   T=DBLT+HALFH
      DO 4 J=1,NINT
4     SV(J)=DSV(J)+ERROR(J)
C****                          ,WHERE ERROR(J)=K1(J)/2
      CALL DERIV
      DO 5 J=1,NINT
      XK(J)=H*SVDER(J)
      SVINC(J)=SVINC(J)+XK(J)
C****  COMPUTE K3
5     SV(J)=DSV(J)+.5*XK(J)
```

```
      CALL DERIV
      DO 6 J=1,NINT
      XK(J)=H*SVDER(J)
      SVINC(J)=SVINC(J)+XK(J)
      ERROR(J)=ERROR(J)-XK(J)
C**** COMPUTE K4
6     SV(J)=DSV(J)+XK(J)
      T=DBLT+H
      CALL DERIV
      DO 7 J=1,NINT
      XK(J)=HALFH*SVDER(J)
      SVINC(J)=(SVINC(J)+XK(J))/3.
      ERROR(J)=ERROR(J)-2.*XK(J)
C**** COMPUTE K5 = K1NEXT
7     SV(J)=DSV(J)+SVINC(J)
      CALL DERIV
      MININD=0
      DO 8 J=1,NINT
      XK(J)=HALFH*SVDER(J)
      ERROR(J)=(ERROR(J)+3.*XK(J))*2.
      IF(U(J).LE.0.)GO TO 8
      IF(ABS(SNGL(ERROR(J))).GE.U(J))GO TO 100
      IF(MININD.GT.0)GO TO 8
      MININD=-1
      IF(ABS(SNGL(ERROR(J))).GE.UMIN(J))MININD=1
8     CONTINUE
      IF(H.LT.HUSE)GO TO 9
      IF(MININD.GE.0)GO TO 9
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.NDOUBL)GO TO 9
      HUSE=2.*HUSE
      NCOUNT=0
9     DBLT=DBLT+H
      IF(H.EQ.HUSE)GO TO 11
      ODD=HUSE/H
      DO 10 J=1,NINT
10    XK(J)=ODD*XK(J)
11    DO 12 J=1,NINT
```

```
      DSV(J)=DSV(J)+SVINC(J)
      SVINC(J)=XK(J)
12    ERROR(J)=XK(J)
      CALL ENERGY(INDRS)
      LIM1=48
      LIM2=51
      HALFH=.5*HUSE
17    DO 15 J=LIM1,LIM2
      ERROR(J)=HALFH*SVDER(J)
15    SVINC(J)=ERROR(J)
      IF(INDRS.EQ.0)GO TO 16
      CALL FINDMG
      INDRS=0
      LIM1=25
      LIM2=30
      GO TO 17
16    IF(NEND.EQ.1)RETURN
      GO TO 13
C****  H TOO BIG
100   NCOUNT=0
      NEND=0
      HUSE=.5*HUSE
      IF(HUSE.LT.HMIN)GO TO 200
      H=HUSE
      HPREV=H
      HALFH=.5*H
      T=DBLT
      DO 101 J=1,NINT
101   SV(J)=DSV(J)
C****  COMPUTE K1
      CALL DERIV
      DO 102 J=1,NINT
      ERROR(J)=HALFH*SVDER(J)
102   SVINC(J)=ERROR(J)
      GO TO 13
200   WRITE(6,201)
201   FORMAT(50H0INTEGRATION INCREMENT TOO SMALL,RUN TERMINATED.    )
      CALL EXIT
```

```
      ENTRY RKRS
301   H=HUSE
      HPREV=H
      HALFH=.5*H
      CALL DERIV
      DO 300 J=1,NINT
      ERROR(J)=HALFH*SVDER(J)
300   SVINC(J)=ERROR(J)
      RETURN
      ENTRY RKIC
      NCOUNT=0
      DO 401 J=1,NINT
401   DSV(J)=SV(J)
      DBLT=T
      GO TO 301
      END
```

Subroutine INPUT

Called by. — INITIL

Function. — Reads input buffer changes as specified on data cards. Prints entire input buffer.

Comments. — Input data items are listed in the User's Guide section of this report.

```
$IBFTC S5       LIST
      SUBROUTINE INPUT
C
      COMMON/BUFFIN/OMGABO(3),BETADO(3),ALPHDO(3),BETAO(3),ALPHAO(3),
     1 PHIO,THETAO,PSIO,TLDBO(3),TLDAO(3),EMBO(3),EMAO(3),EFBO(3),
     2 EFAO(3),HEXTO(3),ENRGBO(3),ENRGAO(3),HORIZO(2),NOISEO(4),STV1O(4)
     3 ,STV2O(4),STV3O(4),STV3PO(4),STV5O(4),SQTIO(4),QTBO(4),U(79),
     4 EMAX,EMIN,DTMIN,DTEST,AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,
     5 AGOMGO(3),AGOMGD(3),AGOMGM,HNOM,K2(2),TAUD(2),K(2),KSF(2),TAUN(2)
     6 ,TAU(2),ELIM(2),KT(2),KB(2),TAUM(2),TF(2),LIMIT(2),GRATIO(2),
     7 ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,TAU3D,TAU3NP,
     8 TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,STG5,JINRTE,
     9 JINRTA,ETAP,ETAR,TAUHRZ,GHRZ,DELW(3),IXX,IYY,IZZ,IXY,IYZ,IXZ
      COMMON/BUFFIN/JETTQE(3),MDNOM(3),MDX(6),MDY(6),MDZ(6),PHASEX(6),
     1 PHASEY(6),PHASEZ(6),FREQ(6),QMASS,PMO(3),VMO(3),AMO(3),R(3),
     2 OMGAMM(3),POS(3),VEL(3),STBIAS(4),PSD,TCYCLE,TEVENT(15),TPRINT,
     3 TRUN,FLOWRT(3)
      COMMON/BUFFIN/NOINT(79),NDOUBL,NBIT,NBIN(25),NBOUT(16),NFXPNT
     1 ,NSLOW
      COMMON/BUFFIN/PTARGT(3),AZCO,CAPM1(3),CAPM2(3),CAPM3(3),ETADOT,
     1 WOAVE,WBMAX,ZO(5),KE1(5),KE2(5),KE3(5),KE4(5),KE5(5),KE6(5),
     2 KR1(5),KR2(5),KR3(5),KR4(5),KR5(5),KR6(5),KSAVE,XKREM,ERRLIM(3),
     3 DTJET(3),MAXRTC(2),BHOLD,BSELFD,BDOTDS
      COMMON/BUFFIN/MODE,NFBSEL,NUPDAT,NGAIN,LAW,NMOUNT,ITER,MODCOM
     1 ,NXTRAP
      REAL NOISEO,JMBETA,JMALPH,K2,K,KSF,KT,KB,LIMIT,JINRTE,JINRTA,IXX,
     1 IYY,IZZ,IXY,IYZ,IXZ,JETTQE,MDNOM,MDX,MDY,MDZ,KE1,KE2,KE3,KE4,KE5,
     2 KE6,KR1,KR2,KR3,KR4,KR5,KR6,KSAVE,MAXRTC
C
C
      NAMELIST/DATA/NBIT,NBIN,NBOUT,NFXPNT,PTARGT,AZCO,CAPM1,CAPM2,
     1 CAPM3,ETADOT,WOAVE,WBMAX,NMOUNT,NUPDAT,MODE,ZO,KE1,KE2,KE3,KE4,
     2 KE5,KE6,KR1,KR2,KR3,KR4,KR5,KR6,NFBSEL,NGAIN,KSAVE,XKREM,ITER,
     X LAW,MODCOM,BHOLD,BSELFD,BDOTDS,ERRLIM,DTJET,
     3 TCYCLE,NSLOW,TEVENT,NXTRAP, TRUN,TPRINT,OMGABO,BETADO,ALPHDO,
     4 BETAO,ALPHAO,PHIO,THETAO,PSIO,TLDBO,TLDAO,EMBO,EMAO,EFBO,EFAO,
     5 HEXTO,ENRGBO,ENRGAO,HORIZO,NOISEO,STV1O,STV2O,STV3O,STV3PO,STV5O,
     6 SQTIO,QTBO,POS,VEL,AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AGOMGO,
```

```
     7 AGOMGD,AGOMGM,HNOM,IXX,IYY,IZZ,IXY,IYZ,IXZ,JETTOE,FLOWRT,
     X KSF,K,K2,KT,KB,
     8 TAU,TAUM,TAUN,TAUD,ELIM,TF,GRATIO,LIMIT,MAXRTC,ESTAR,ASTAR,TAUWNG
     9 ,TAU1,TAU2N,TAU2D,TAU3N,TAU3D,TAU3NP,TAU3DP,TAU5,STG2,STG3,STG5,
     1 STKT,STKM,STKV,STKR,STAL,STTF,JINRTE,JINRTA,PSD,STBIAS,ETAP,ETAR,
     2 GHRZ,TAUHRZ,DELW,MDNOM,MDX,MDY,MDZ,PHASEX,PHASEY,PHASEZ,FREQ,
     3 QMASS,PMO,VMO,AMO,R,OMGAMM,DTEST,DTMIN,U,EMAX,EMIN,NDOUBL,NOINT
C
C
      READ(5,DATA)
C
      WRITE(6,DATA)
C
      RETURN
      END
```

Subroutine OUTPUT

Called by. — MAIN.

Function. — Prepares and prints output data, computes momentum check numbers, and computes theoretical vehicle torques.

Comments. — The output buffer XLNE is used for temporary storage of intermediate results while computing momentum check and vehicle torques. A detailed list of output variables is contained in the User's Guide section.

```
$IBFTC S6       LIST
      SUBROUTINE OUTPUT
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/RATES/OMEGA(3,3),OMEGAP(3,3),OMGASQ(3),OMGAXP(3)
      COMMON/CGYRO/AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AB AG,BB BG,
     1 CB BG,AA BA,AA CA,BA CA,BB CB,DIF1,DIF2,GRAJMA,SUM1,SUM2,SUM3,
     2 SUM4,SUM5,SUM6,SUM7,SUM8,SUM9,SUM10,AGOMGO(3),AGOMGD(3),AGOMGM,
     3 HNOM
      REAL JMBETA,JMALPH
      COMMON/CBODY/IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,
     1 IXZO,IXXD,IYYD,IZZD,IXYD,IYZD,IXZD,ITERM(3)
      REAL IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,IXZO,IXXD,
     1 IYYD,IZZD,IXYD,IYZD,IXZD,ITERM,INERT(6),INERTO(6),INERTD(6)
      EQUIVALENCE(IXX,INERT(1)),(IXXO,INERTO(1)),(IXXD,INERTD(1))
      COMMON/CSERVO/BINPUT(3),AINPUT(3),K2BETA,K2ALPH,TAUDBE,TAUDAL,
     1 KBETA,KALPH,KSFBE,KSFAL,TAUNBE,TAUNAL,TAUBE,TAUAL,ELIMBE,ELIMAL,
     2 KTBE,KTAL,KBBETA,KBALPH,TAUMBE,TAUMAL,TFBETA,TFALPH,BLIMIT,ALIMIT
     3 ,GRBETA,GRALPH
      REAL K2BETA,K2ALPH,KBETA,KALPH,KSFBE,KSFAL,KTBE,KTAL,KBBETA,KBALPH
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
      REAL MJET,MTOT,MB,MA,MAGJET
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
```

```
1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
 REAL MEANA,MEANAO
 COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VEO,FIXWO(10),WEARTH
 COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
 COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
 INTEGER FBSEL,UPDATE
 COMMON/FLOOUT/BETADC(3),ALPHDC(3),TELAZC,EULER(3),EPSLON(3),
1 OMGABC(3),JET(3)
 COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,
2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DELW(3),STBIAS(4),STSIG,
3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),ELBORE(2),AZBORE(2),QST(4)
4 ,WNG(4),HRZACT(2)
 COMMON/DIST/MDIST(3),MDNOM(3),SPHASE(3,6),CPHASE(3,6),SFREQ(6),
1 CFREQ(6),MDAMP(3,6),FREQ(6),TQFMM(3)
 REAL MDIST,MDNOM,MDAMP
 COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
C FL2NP0,FL2NP1
 COMMON/FIXIN/W(37)
 COMMON/QUANT/NBIN(25),NBOUT(16),NEXPNT
 COMMON/MOD567/DELE(3),ECOM(3),EDCOM(3),TENDM,ECOMDP(3),DELTAE(3),
CMAX,MAXRT ,TP,NHHH
 COMMON/CONTL1/EPPREV(3),GAIN(6,5),GAINP(6,5),EPP(3),HNOMP,KLCL(6),
C     MAGA(3),MAGB(3),MCA(3),MCB(3),TRQC(3),TRQCP(3),UNITVA(3,3),
CUNITVB(3,3),ZO(5)
 REAL KLCL,MAGA,MAGB,MCA,MCB
 COMMON/CONTL3/CGYRO(3),COSTA(3),COSTB(3),RTEST,RUSF,TDESSQ,
CTDOTA(3),TDOTB(3),TREMSQ,XKEND,ITER,NOITER
 COMMON/NAV/F(10),FDOT,FDUM(10),FTOT,G(10),GDOT,GDUM(10),GTOT,P(3),
CPO(3),PDOT(3),PDOTO(3)
 DIMENSION XLNE1(6),XLNE2(9),XLNE3(9),XLNE4(9),XLNE5(9),XLNE6(9),
1 XLNE7(9),XLNE8(9),XLNE9(9),XLNE10(8),XLNE11(5),XLNE(91)
 EQUIVALENCE(XLNE1(1),XLNE(1)),(XLNE2(1),XLNE(7)),(XLNE3(1),
1 XLNE(16)),(XLNE4(1),XLNE(25)),(XLNE5(1),XLNE(34)),(XLNE6(1),
```

```
     2 XLNE(43)),(XLNE7(1),XLNE(52)),(XLNE8(1),XLNE(61)),(XLNE9(1),
     3 XLNE(70)),(XLNE10(1),XLNE(79)),(XLNE11(1),XLNE(87))
      DIMENSION NNW(3),NEDCOM(3),NECOM(3),NTRQC(3),NTRQCP(3),NKLCL(6),
     1 NNP(3),NPDOT(3)
      EQUIVALENCE (NNW(1),W(1)),(NEDCOM(1),EDCOM(1)),(NECOM(1),ECOM(1)),
     1 (NTRQC(1),TRQC(1)),(NTRQCP(1),TRQCP(1)),(NKLCL(1),KLCL(1)),
     2 (NNP(1),P(1)),(NPDOT(1),PDOT(1))
      IF(NPRCTL.GE.NPRINT)GO TO 1
      NPRCTL=NPRCTL+1
      RETURN
1     NPRCTL=1
      FUELT=0.
      LNECNT=LNECNT+12
      IF(LNECNT.LE.60)GO TO 2
      LNECNT=12
      WRITE(6,1000)
1000  FORMAT(1H1)
2     DO 3 J=1,3
      IF(NFXPNT.EQ.0)GO TO 8
      XLNE4(J+6)=FLOAT(NNW(J))/FLNP5
      XLNE5(J)=FLOAT(NEDCOM(J))/FLNP5
      XLNE5(J+3)=FLOAT(NECOM(J))/FLNM2
      XLNE10(J)=FLOAT(NNP(J))/F2NM25
      XLNE10(J+3)=FLOAT(NPDOT(J))/F2NM15
      IF(LAW.EQ.1)GO TO 4
      XLNE7(J)=FLOAT(NTRQC(J))/FLNM8
      XLNE7(J+3)=FLOAT(NTRQCP(J))/FLNM8
      GO TO 9
4     XLNE7(J)=HNOM*FLOAT(NKLCL(J)-NKLCL(J+3))/FLNP2
      GO TO 5
8     XLNE4(J+6)=W(J)
      XLNE5(J)=EDCOM(J)
      XLNE5(J+3)=ECOM(J)
      XLNE10(J)=P(J)
      XLNE10(J+3)=PDOT(J)
      IF(LAW.EQ.1)GO TO 6
      XLNE7(J)=TRQC(J)
      XLNE7(J+3)=TRQCP(J)
```

```
      GO TO 9
6     XLNE7(J)=HNOM*(KLCL(J)-KLCL(J+3))
5     XLNE7(J+3)=XLNE7(J)
9     XLNE8(J)=MJET(J)
      XLNE6(J)=DRALPH(J+3)
      XLNE6(J+3)=ALPHA(J+3)
      XLNE6(J+6)=EULER(J)
      XLNE5(J+6)=EPSLON(J)
      XLNE9(J+6)=OMGABD(J)*RTODEG
      XLNE3(J+6)=0.
      XLNE2(J)=0.
      FUELT=FUELT+FUEL(J)
      XLNE2(J+6)=OMGABC(J)-OMEGAB(J)
3     XLNE8(J+3)=MDIST(J)+TQEMM(J)
      IF(NFXPNT.NE.0)XLNE5(5)=XLNE5(5)/2.
      K=0
7     DO 10 J=1,3
      INDEX=K+J+6
      IF(J-2)11,12,13
11    JY=2
      JZ=3
      GO TO 20
12    JY=3
      JZ=1
      GO TO 20
13    JY=1
      JZ=2
20    XLNE7(INDEX)=AGOMGA(J)*((ALPHDC(J)+XLNE3(JY+6))*SINA(J)*COSB(J)+
     1 (BETADC(J)*COSA(J)+XLNE3(JZ+6))*SINB(J))+AGOMGA(JY)*((ALPHDC(JY)+
     2 XLNE3(JZ+6))*COSA(JY)*COSB(JY)-(BETADC(JY)*SINA(JY)+XLNE3(JY+6))*
     3 SINB(JY))-AGOMGA(JZ)*COSB(JZ)*(BETADC(JZ)+XLNE3(JY+6)*COSA(JZ)+
     4 XLNE3(JZ+6)*SINA(JZ))
      IF(K.NE.0)GO TO 10
      XLNE3(J)=INERT(J)*OMEGAB(J)-INERT(J+3)*OMEGAB(JY)-INERT(JZ+3)*
     1 OMEGAB(JZ)
      XLNE4(1)=SUM5*OMEGAP(JZ,J)+SUM3*BETAD(J)
      XLNE4(2)= AB AG*OMEGA(J,J)+AGOMGA(J)
      XLNE4(3)=BB BG*OMEGA(JY,J)
```

```
      XLNE9(1)=AA*OMEGAP(J,J)+XLNE4(2)*COSB(J)-XLNE4(3)*SINB(J)
      XLNE2(J+3)=XLNE4(1)*SINA(J)+XLNE9(1)*COSA(J)
      XLNE3(J+3)=XLNE4(2)*SINB(J)+XLNE4(3)*COSB(J)+BA*OMEGAP(JY,J)
     1 +JMALPH*(OMEGAB(JY)+GRALPH*ALPHAD(J))
      XLNE4(J+3)=XLNE4(1)*COSA(J)-XLNE9(1)*SINA(J)
10    CONTINUE
      IF(K.NE.0)GO TO 14
      K=9
      XLNE9(4)=XLNE2(4)+XLNE3(6)+XLNE4(5)+XLNE3(1)
      XLNE9(5)=XLNE2(5)+XLNE3(4)+XLNE4(6)+XLNE3(2)
      XLNE9(6)=XLNE2(6)+XLNE3(5)+XLNE4(4)+XLNE3(3)
      DO 15 J=1,3
      XLNE1(J+1)=-HEXT(J)
      XLNE3(J+6)=OMEGAB(J)
      XLNE2(1)=XLNE2(1)+XLNE7(J+3)**2
15    XLNE2(2)=XLNE2(2)+XLNE7(J+3)*XLNE7(J+6)
      XLNE1(5)=XLNE2(2)/XLNE2(1)
      DO 16 J=1,3
      XLNE2(3)=XLNE2(3)+(XLNE7(J+6)-XLNE1(5)*XLNE7(J+3))**2
      DO 16 KK=1,3
16    XLNE1(J+1)=XLNE1(J+1)+DIRCO(J,KK)*XLNE9(KK+3)
      XLNE1(6)=SQRT(XLNE2(3)/XLNE2(1))
      GO TO 7
14    XLNE1(1)=0.
      DO 17 J=1,6
      XLNE1(1)=XLNE1(1)+ENRGYB(J)
      XLNE2(J)=BETADC(J)
      XLNE3(J)=BETAD(J)
      XLNE4(J)=BETA(J)
      XLNE9(J)=POS(J)
17    XLNE10(J)=XLNE10(J)-XLNE9(J)
      DO 18 J=1,45
18    XLNE(J+6)=XLNE(J+6)*RTODEG
      DO 19 J=1,2
      XLNE10(J+6)=HRZACT(J)*RTODEG
      XLNE11(J)=QST(J)*RTODEG
19    XLNE11(J+2)=(AZBORE(J)-QTB(J+2))*RTODEG
      XLNE11(5)=TELAZC*RTODEG
```

```
      XLNE1(1)=1.356*XLNE1(1)
      WRITE(6,51)T,FUELT,XLNE1,NOITER
      WRITE(6,52)XLNE2
      WRITE(6,53)XLNE3
      WRITE(6,54)XLNE4
      WRITE(6,55)XLNE5
      WRITE(6,56)XLNE6
      WRITE(6,57)XLNE7
      WRITE(6,58)XLNE8
      WRITE(6,59)XLNE9
      WRITE(6,60)XLNE10
      WRITE(6,61)XLNE11
51    FORMAT(3HOT=F8.2,3H,W=E11.4,3H,E=E11.4,7H,HTEST=E14.7,2E15.7,
     1 7H,TGAIN=2F8.5,6H,ITER=I2)
52    FORMAT(5H  BDC 3E13.5,5H ,ADC 3E13.5,5H,WERR 3E13.5)
53    FORMAT(5H   BD 3E13.5,5H  ,AD 3E13.5,5H   ,W 3E13.5)
54    FORMAT(5H    B 3E13.5,5H   ,A 3E13.5,5H ,WOB 3E13.5)
55    FORMAT(5H ATDC 3E13.5,5H ,ATC 3E13.5,5H,EPOB 3E13.5)
56    FORMAT(5H  ATD 3E13.5,5H  ,AT 3E13.5,5H,ATOB 3E13.5)
57    FORMAT(5H   TC 3E13.5,5H ,TCP 3E13.5,5H,TGIM 3E13.5)
58    FORMAT(5H TJET 3E13.5,5H ,TEX 3E13.5,5H,TCTL 3E13.5)
59    FORMAT(5H    P 3E13.5,5H   ,V 3E13.5,5H  ,WD 3E13.5)
60    FORMAT(5H OBPE 3E13.5,5H,OBVE 3E13.5,11H,HRZ ERR,P=F12.5,3H,R=
     1 E12.5)
61    FORMAT(24H STAR TRACKER ERR,DELE1=E12.5,7H,DELE2=E12.5,7H,DELA1=
     1 E12.5,7H,DELA2=E12.5,14H  ,TEL AZ COM=E12.5)
      RETURN
      END
```

Subroutine MANUAL

Called by. — MAIN, INITIL.

Calls. — Optionally; FANDG, UPDAT3.

Function. — This subroutine generates all external communication with the control computer. This routine is written to supply maneuver and manual mode commands.

Comments. — It is intended that this routine will be reprogrammed to suit the needs of the users. The commentary in the listing describes the capabilities of this routine.

```
$IBFTC S7      LIST
      SUBROUTINE MANUAL
C
C
C****  THIS SUBROUTINE SIMULATES ASTRONAUT ACTIONS. IT CAN BE PROGRAMMED
C      TO SET THE INTEGERS LISTED.
C
C        SYMBOL  (RANGE) (MEANING)  COMMENTS
C
C      1. MODE    (1-7)  (EXP1-EXP4,HOLD,MANEUVER,MANUAL)  EXP2 AND EXP3
C               CANNOT BE PERFORMED IN THE SAME RUN BECAUSE OF STAR
C               TRACKER MOUNTING DIFFERENCES.
C
C      2. MDLAST  (0-7)  (MODE DURING LAST PASS OF CONTROL COMPUTER)
C               CHANGING MODE AUTOMATICALLY SIMULATES AN INSERT COMMAND.
C               TO SIMULATE AN INSERT COMMAND WITHOUT CHANGING MODE ,SET
C               MDLAST TO 0  .
C
C      3. NGAIN   (1-5)  (SELECTS ONE OF THE FIVE SETS OF CONTROL LAW
C                                                              GAINS)
C
C      4. UPDATE  (0-2)  (IF SET TO 1, WHILE IN EXP3 MODE, ONBOARD
C               ATTITUDE INTEGRATION IS UPDATED FROM STAR TRACKERS. IF
C               SET TO 2, IN ANY MODE, UPDATE FROM ATTITUDE COMPUTATIONS.
C               IN BOTH CASES THE CONTROL COMPUTER RESETS UPDATE TO ZERO)
C
C      5. MODCOM  (0-1)  (IF SET TO 1, CONTROL LAWS 2 AND 3 USE TORQUE
C               COMMAND PRIME, WHICH CONSIDERS BODY RATE COUPLING)
C
C      6. FBSEL   (1-2)  (IF SET TO 1, CONTROL LAWS USE SENSED RATE
C               FEEDBACK, IF SET TO 2, THEY USE DERIVED RATE FEEDBACK)
C               IF FBSEL IS CHANGED IT WILL PROBABLY BE NECESSARY TO
C               CHANGE NGAIN.
C
C      7. LAW     (1-3)  (SELECTS CONTROL LAW 1,2, OR 3.
C
C
C****  THIS SUBROUTINE CAN ALSO BE PROGRAMMED TO SUPPLY THE FOLLOWING
```

```
C       VARIABLES.
C
C       1. EULRDC(3) RAD./SEC. ,  (PHIDOTC,THETADOTC,PSIDOTC) THESE ARE
C               THE STICK COMMANDS DURING MANUAL MODE (7).
C               MAXIMUM VALUE = 2**(-5)
C
C       2. EULRC(3) RAD. , (PHIC,THETAC,PSIC) ATTITUDE COMMANDS DURING
C               MANEUVER MODE (6).  MAXIMUM VALUE = PI,PI/2,PI
C
C       3. TMAN  SEC. , (TIME TO COMPLETE MANEUVER,MODE 6)
C               MAXIMUM VALUE = 2**8
C
C
C****   THIS SUBROUTINE CAN CALL THE FOLLOWING OTHER SUBROUTINES.
C
C       1. SUBROUTINE FANDG ,   SIMULATES TRNSMISSION OF NEW
C               COEFFICIENTS FOR THE F AND G SERIES (NAVIGATION).
C
C       2. SUBROUTINE UPDAT3,   SIMULATES TRANSMISSION OF UPDATED PHASE
C               ANGLE FOR EXPERIMENT 3  OMEGAXC AND OMEGAYC. (SINUSOIDAL
C               TO MAINTAIN THE Z AXIS ALONG THE LOCAL VERTICAL).
C
C
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      REAL NOISED
      EQUIVALENCE(OMGAXD,OMGABD(1)),(OMGAYD,OMGABD(2)),(OMGAZD,OMGABD
     1 (3))
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
```

```
     1 NOINT(79)
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
      REAL MJET,MTOT,MB,MA,MAGJET
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VFO,FIXWO(10),WEARTH
      REAL MU
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/FLOOUT/BETADC(3),ALPHDC(3),TELAZC,EULER(3),EPSLON(3),
     1 OMGABC(3),JET(3)
      COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
     1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,
     2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DELW(3),STBIAS(4),STSIG,
     3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),ELBORE(2),AZBORE(2),QST(4)
     4 ,WNG(4),HRZACT(2)
      REAL JINRT
      COMMON/MISCEL/FS,DBLFS,NBIT   ,NH,MDLAST,HALFFS
C
C
      DIMENSION PHIC(4),THETAC(4),PSIC(4),TSTART(4),TMANUV(4)
      DATA  PHIC,THETAC,PSIC,TSTART,TMANUV /0.,0.,.1,0.,0.,.1,.1,0.,
     1 .1,.1,.1,0.,-1.,6.,12.,18.,4*0./
      NAMELIST/COMAND/PHIC,THETAC,PSIC,TSTART,TMANUV
C
      DO 999 J=1,3
999   IF(JET(J).NE.0)FUEL(J)=FUEL(J)+FLOWRT(J)
      IF(NMAN1.EQ.0)GO TO 1000
      MDLAST=0
C
```

```
C****  THIS SECTION (THRU STATEMENT 1000) IS EXECUTED ON FIRST PASS
C      THRU THIS ROUTINE (EACH RUN). USE THIS SECTION TO READ IN ANY
C      NEEDED DATA.
C
C
      READ(5,COMAND)
      WRITE(6,COMAND)
      NINSRT=1
C
C
C****  END OF SINGLE PASS SECTION
C
C
C****  THE NEXT SECTION SIMULATES OPERATOR ACTIONS
C
C
1000  IF(MODE.LT.6)GO TO 1003
      IF(NINSRT.GE.5)GO TO 1003
      IF(T.LT.TSTART(NINSRT))GO TO 1003
      IF(MODE.EQ.7)GO TO 1001
      EULRC(1) =  PHIC(NINSRT)*DEGTOR
      EULRC(2) =THETAC(NINSRT)*DEGTOR
      EULRC(3) =  PSIC(NINSRT)*DEGTOR
      TMAN=TMANUV(NINSRT)
      MDLAST=0
      GO TO 1002
1001  EULRDC(1) =  PHIC(NINSRT)*DEGTOR
      EULRDC(2) =THETAC(NINSRT)*DEGTOR
      EULRDC(3) =  PSIC(NINSRT)*DEGTOR
1002  NINSRT=NINSRT+1
1003  NMAN1=0
      RETURN
      END
```

Subroutine SUMARY

Called by. — MAIN.

Function. — Prints power, energy, and fuel consumed at the end of each run.

```
$IBFTC S8       LIST
      SUBROUTINE SUMARY
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
      REAL MJET,MTOT,MB,MA,MAGJET
      COMMON/POWER/PMAXB(3),PMAXA(3),DTLIST(300),KDT
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VEO,FIXWO(10),WEARTH
      WRITE(6,1)
      ENRGYT=0.
      FUELT=FUEL(1)+FUEL(2)+FUEL(3)
      DUM=1.356/T
      DO 10 J=1,6
      PMAXB(J)=1.356*PMAXB(J)
      ENRGYB(J)=DUM*ENRGYB(J)
10    ENRGYT=ENRGYT+ENRGYB(J)
      WRITE(6,2)PMAXB,PMAXA
      WRITE(6,3)ENRGYB,ENRGYA,ENRGYT
      DO 11 J=1,6
11    ENRGYB(J)=1.333333*ENRGYB(J)
      ENRGYT=1.333333*ENRGYT+90.
      WRITE(6,4)ENRGYB,ENRGYA,ENRGYT
      WRITE(6,5)FUEL,FUELT
      NIPSEC=FLOAT(KDT)/T    +.5
      WRITE(6,6)KDT,NIPSEC,DTLIST
1     FORMAT(1H1,30X,6HBETA 1,9X,6HBETA 2,9X,6HBETA 3,8X,7HALPHA 1,8X,
     1 7HALPHA 2,8X,7HALPHA 3,9X,5HTOTAL )
2     FORMAT(25H0MAX OUTPUT POWER (WATTS)  6E15.7 )
3     FORMAT(25H AVE OUTPUT POWER (WATTS)  7E15.7 )
4     FORMAT(25H THEOR. AVE POWER (WATTS)  7E15.7 )
5     FORMAT(25H0JET FUEL (LBS) X,Y,Z,TOT  3E15.7,45X,F15.7)
6     FORMAT(1H0,I8,24H INTEGRATION STEPS, AVE  I4,25H/SEC. THE FIRST 30
     10 WERE,  /(6E20.7))
```

```
RETURN
END
```

Subroutine DERIV

Called by. — RKII.

Calls. — TRIG, RATE, POSVEL, EXPER, DISTRB, FINDMG, FINDA, FINDB, EVDER.

Function. — Serves as a sequencer for computing derivatives.

```
$IBFTC SEQ     LIST
      SUBROUTINE DERIV
      CALL TRIG
      CALL RATE
      CALL POSVEL
      CALL EXPER
      CALL DISTRB
      CALL FINDMG
      CALL FINDA
      CALL FINDB
      CALL EVDER
      RETURN
      END
```

Subroutine TRIG

Called by. — DERIV.

Function. — Evaluates sines and cosines of Euler angles and CMG gimbal angles. Evaluates direction cosine matrix.

```
$IBFTC S9       LIST
      SUBROUTINE TRIG
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      DO 1 J=1,9
      SINB(J)=SIN(BETA(J))
1     COSB(J)=COS(BETA(J))
      DO 2 J=1,6
      SINBSQ(J)=SINB(J)**2
2     COSBSQ(J)=COSB(J)**2
      DIRCO (1,1)=COSPSI*COSTHE
      DUM=COSPSI*SINTHE
      DIRCO (1,2)=DUM*SINPHI-SINPSI*COSPHI
      DIRCO (1,3)=DUM*COSPHI+SINPSI*SINPHI
      DIRCO (2,1)=SINPSI*COSTHE
      DUM=SINPSI*SINTHE
      DIRCO (2,2)=DUM*SINPHI+COSPSI*COSPHI
      DIRCO (2,3)=DUM*COSPHI-COSPSI*SINPHI
      DIRCO (3,1)=-SINTHE
      DIRCO (3,2)=COSTHE*SINPHI
      DIRCO (3,3)=COSTHE*COSPHI
      RETURN
      END
```

Subroutine RATE

Called by. — DERIV.

Function. — Evaluates equations (A - 7) to (A - 24).

```
$IBFTC S10     LIST
      SUBROUTINE RATE
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/RATES/OMEGA(3,3),OMEGAP(3,3),OMGASQ(3),OMGAXP(3)
      OMEGAP(1,1)=OMEGAX*COSA(1)-OMEGAZ*SINA(1)
      OMEGAP(2,1)=OMEGAY+ALPHAD(1)
      OMEGAP(3,1)=OMEGAX*SINA(1)+OMEGAZ*COSA(1)
C
      OMEGAP(1,2)=OMEGAX*COSA(2)+OMEGAY*SINA(2)
      OMEGAP(2,2)=-OMEGAX*SINA(2)+OMEGAY*COSA(2)
      OMEGAP(3,2)=OMEGAZ+ALPHAD(2)
C
      OMEGAP(1,3)=OMEGAX+ALPHAD(3)
      OMEGAP(2,3)=OMEGAY*COSA(3)+OMEGAZ*SINA(3)
      OMEGAP(3,3)=-OMEGAY*SINA(3)+OMEGAZ*COSA(3)
C
      OMEGA(1,1)=OMEGAP(1,1)*COSB(1)+OMEGAP(2,1)*SINB(1)
      OMEGA(2,1)=-OMEGAP(1,1)*SINB(1)+OMEGAP(2,1)*COSB(1)
      OMEGA(3,1)=OMEGAP(3,1)+BETAD(1)
C
      OMEGA(1,2)=OMEGAP(1,2)+BETAD(2)
      OMEGA(2,2)=OMEGAP(2,2)*COSB(2)+OMEGAP(3,2)*SINB(2)
      OMEGA(3,2)=-OMEGAP(2,2)*SINB(2)+OMEGAP(3,2)*COSB(2)
C
      OMEGA(1,3)=OMEGAP(1,3)*COSB(3)-OMEGAP(3,3)*SINB(3)
      OMEGA(2,3)=OMEGAP(2,3)+BETAD(3)
      OMEGA(3,3)=OMEGAP(1,3)*SINB(3)+OMEGAP(3,3)*COSB(3)
      RETURN
      END
```

Subroutine POSVEL

Called by. –DERIV.

Function. –Performs Kepler orbit position and velocity determination.

Comments. –The algorithm used is presented below. The following symbols are defined for this section only.

$$\hat{\rho}_0 = \frac{\underline{\rho}_0}{|\underline{\rho}_0|}$$

$$\underline{V}_s = \dot{\underline{\rho}}_0 - (\dot{\underline{\rho}}_0 \cdot \hat{\rho}_0)\hat{\rho}_0$$

$$H = |\underline{\rho}_0| \cdot |\underline{V}_s|$$

$$\hat{S}_0 = \frac{\underline{V}_s}{|\underline{V}_s|}$$

$$\underline{f} = \frac{H}{\mu}\dot{\underline{\rho}}_0 - \hat{S}_0$$

$$e = |\underline{f}|$$

$$E_0 = \tan^{-1} \frac{(\underline{f} \cdot \hat{\rho}_0)\sqrt{1-e^2}}{e^2 + (\underline{f} \cdot \hat{S}_0)}$$

$$M = E_0 - e \sin E_0$$

$$a = \frac{H^2}{\mu(1-e^2)}$$

The procedure:

1) Solve $M + \frac{\sqrt{\mu}}{a^{3/2}} t = E - e \sin E$

for E

2) Compute

$$b = \frac{\cos E - e}{e - e^2 \cos E}$$

$$c = \frac{\sqrt{1-e^2}\ \sin E}{e - e^2 \cos E}$$

$$\hat{\rho} = \hat{\rho}_0 \left[b(\underline{f}\cdot\hat{S}_0) + c(\underline{f}\cdot\hat{\rho}_0)\right] + \hat{S}_0 \left[c(\underline{f}\cdot\hat{S}_0) - b(\underline{f}\cdot\hat{\rho}_0)\right]$$

$$\hat{S} = -\hat{\rho}_0 \left[c(\underline{f}\cdot\hat{S}_0) - b(\underline{f}\cdot\hat{\rho}_0)\right] + \hat{S}_0 \left[b(\underline{f}\cdot\hat{S}_0) + c(\underline{f}\cdot\hat{\rho}_0)\right]$$

$$\underline{\rho}(t) = \frac{H^2}{\mu(1 + \hat{S}\cdot\underline{f})}\ \hat{\rho}$$

$$\dot{\underline{\rho}}(t) = \frac{\mu}{H}(\hat{S} + \underline{f})$$

```
$IBFTC S11      LIST
      SUBROUTINE POSVEL
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      MEANA=MEANAO+PVCON(3)*T
1     EPREV=ENOW
      ENOW=MEANA+ECCENT*SIN(EPREV)
      IF(ABS(ENOW-EPREV).GT.(1.E-7*ABS(ENOW)+1.E-10))GO TO 1
5     CENOW=COS(ENOW)
      DUM=1.-ECCENT*CENOW
      COSETA=(CENOW-ECCENT)/DUM
      SINETA=SIN(ENOW)*PVCON(4)/DUM
      DUM=COSETA*CETAO+SINETA*SETAO
      DUM1=SINETA*CETAO-COSETA*SETAO
      DUM2=1.
      DO 2 J=1,3
      POS(J)=POSO(J)*DUM+VELO(J)*DUM1
      DUM3=-POSO(J)*DUM1+VELO(J)*DUM
      VEL(J)=PVCON(2)*(DUM3+FVECT(J))
2     DUM2=DUM2+DUM3*FVECT(J)
      DUM2=PVCON(1)/DUM2
      DO 3 J=1,3
3     POS(J)=POS(J)*DUM2
      RETURN
      END
```

Subroutine EXPER

Called by. — DERIV, RKII.

Function. — This routine solves for the derivatives of the variables related to star tracker and horizon sensor dynamics.

Comments. — The applicable equations are listed in Appendix D.

```
$IBFTC S12      LIST
      SUBROUTINE EXPER
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      REAL NOISED
      EQUIVALENCE(OMGAXD,OMGABD(1)),(OMGAYD,OMGABD(2)),(OMGAZD,OMGABD
     1 (3))
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
     1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,
     2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DELW(3),STBIAS(4),STSIG,
     3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),ELBORE(2),AZBORE(2),QST(4)
     4 ,WNG(4),HRZACT(2)
      REAL JINRT
      REAL LAMDA
      NINT=79
      GO TO (1,2,3,1,1,1,1),MODE
1     NDUM=1
      NINT=45
```

```
      GO TO 4
2     NDUM=4
      NSTAR=3
      QTBD(1)=SQTI(1)-OMEGAY*COS(QTB(3))-OMEGAZ*SIN(QTB(3))
      QTBD(3)=SQTI(3)-OMEGAX
      DO 5 J=1,2
      SX(J)=DIRCO (1,1)*SXI(J)+DIRCO (2,1)*SYI(J)+DIRCO (3,1)*SZI(J)
      SY(J)=DIRCO (1,2)*SXI(J)+DIRCO (2,2)*SYI(J)+DIRCO (3,2)*SZI(J)
      SZ(J)=DIRCO (1,3)*SXI(J)+DIRCO (2,3)*SYI(J)+DIRCO (3,3)*SZI(J)
      ELBORE(J)=ASIN(SX(J))
      AZBORE(J)=ATAN2(-SY(J),SZ(J))
5     QST(J)=ELBORE(J)-QTB(J)
      QTBD(2)=SQTI(2)-OMEGAY*COS(AZBORE(2))-OMEGAZ*SIN(AZBORE(2))
      QST(3)=(AZBORE(1)-QTB(3))*COS(ELBORE(1))
4     DO 6 J=NDUM,4
      NOISED(J)=0.
      STV1D(J)=0.
      STV2D(J)=0.
      STV3D(J)=0.
      STV3PD(J)=0.
      STV5D(J)=0.
      SQTID(J)=0.
6     QTBD(J)=0.
      GO TO (7,8,3,11,7,7,7),MODE
3     NSTAR=4
      DO 9 J=1,2
      SX(J)=DIRCO (1,1)*SXI(J)+DIRCO (2,1)*SYI(J)+DIRCO (3,1)*SZI(J)
      SY(J)=DIRCO (1,2)*SXI(J)+DIRCO (2,2)*SYI(J)+DIRCO (3,2)*SZI(J)
      SZ(J)=DIRCO (1,3)*SXI(J)+DIRCO (2,3)*SYI(J)+DIRCO (3,3)*SZI(J)
      ELBORE(J)=-ASIN(SZ(J))
      AZBORE(J)=ATAN2(-SX(J),SY(J))
      QST(J)=ELBORE(J)-QTB(J)
      QST(J+2)=(AZBORE(J)-QTB(J+2))*COS(ELBORE(J))
      QTBD(J)=SQTI(J)+OMEGAX*COS(QTB(J+2))+OMEGAY*SIN(QTB(J+2))
9     QTBD(J+2)=SQTI(J+2)-OMEGAZ
8     DO 10  J=1,NSTAR
      NOISED(J)=(WNG(J)-NOISE(J))/TAUWNG
      STV1D(J)=(QST(J)*STKT-STV1(J))/TAU1
```

```
      STV2D(J)=(STG2*(STV1(J)+NOISE(J))-STV2(J))/TAU2D
      STV3D(J)=(STG3*(TAU2N*STV2D(J)+STV2(J)-STV5(J))-STV3(J))/TAU3D
      STV3PD(J)=(TAU3N*STV3D(J)+STV3(J)-STV3P(J))/TAU3DP
      DUM    =TAU3NP*STV3PD(J)+STV3P(J)
      IF(ABS(DUM    ).GT.STAL)DUM    =SIGN(STAL,DUM    )
      STV5D(J)=(STKR*QTBD(J)-STV5(J))/TAU5
      IF(QTBD(J).NE.0.)GO TO 20
      SQTID(J)=STKM*DUM
      IF(ABS(SQTID(J)).GT.STTF)GO TO 21
      SQTID(J)=0.
      GO TO 10
21    SQTID(J)=(SQTID(J)-SIGN(STTF,SQTID(J)))/JINRT(J)
      GO TO 10
20    STCOUL=SIGN(STTF,QTBD(J))/JINRT(J)
      SQTID(J)=STKM*(DUM-STKV*QTBD(J))/JINRT(J)-STCOUL
      DUM1=DELTAT*SQTID(J)
      IF(DUM1/QTBD(J).GE.-1.)GO TO 10
      DUM2=SQTID(J)+STCOUL+STCOUL
      IF(DUM2/SQTID(J).LE.0.)GO TO 22
      SQTID(J)=DUM2+QTBD(J)*(DUM2-SQTID(J))/DUM1
      GO TO 10
22    SQTID(J)=-QTBD(J)/DELTAT
10    CONTINUE
7     HORIZD(1)=0.
      HORIZD(2)=0.
      RETURN
11    NINT=47
      DUM1=POS(1)**2+POS(3)**2
      DUM2=SQRT(DUM1)
      DUM1=SQRT(DUM1+POS(2)**2)
      LAMDA=ASIN(POS(2)/DUM1)
      CLONG=POS(3)/DUM2
      SLONG=POS(1)/DUM2
      LAMDA=LAMDA+.003373*SIN(2.*LAMDA)+.000006*SIN(4.*LAMDA)
      HRZACT(1)=          DIRCO (1,2)*CLONG-DIRCO (3,2)*SLONG
      HRZACT(2)=          (DIRCO (1,1)*SLONG+DIRCO (3,1)*CLONG)*
     1 COS(LAMDA)+DIRCO (2,1)*SIN(LAMDA)
      HORIZD(1)=(GHORIZ*HRZACT(1)-HORIZ(1))/TAUHRZ
```

```
HORIZD(2)=(GHORIZ*HRZACT(2)-HORIZ(2))/TAUHRZ
RETURN
END
```

Subroutine DISTRB

Called by. — DERIV.

Function. — Computes disturbance torques acting on the vehicle, including time functions approximating gravity gradient, and aerodynamic torques plus internal moving mass effects.

Comments. — The equations used in this routine are those supplied in the RFP.

```
$IBFTC S13      LIST
      SUBROUTINE DISTRB
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/CGYRO/AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AB AG,BB BG,
     1 CB BG,AA BA,AA CA,BA CA,BB CB,DIF1,DIF2,GRAJMA,SUM1,SUM2,SUM3,
     2 SUM4,SUM5,SUM6,SUM7,SUM8,SUM9,SUM10,AGOMGO(3),AGOMGD(3),AGOMGM,
     3 HNOM
      REAL JMBETA,JMALPH
      COMMON/CBODY/IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,
     1 IXZO,IXXD,IYYD,IZZD,IXYD,IYZD,IXZD,ITERM(3)
      REAL IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,IXZO,IXXD,
     1 IYYD,IZZD,IXYD,IYZD,IXZD,ITERM,INERT(6),INERTO(6),INERTD(6)
      EQUIVALENCE(IXX,INERT(1)),(IXXO,INERTO(1)),(IXXD,INERTD(1))
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/DIST/MDIST(3),MDNOM(3),SPHASE(3,6),CPHASE(3,6),SFREQ(6),
     1 CFREQ(6),MDAMP(3,6),FREQ(6),TQEMM(3)
      REAL MDIST,MDNOM,MDAMP
      COMMON/MOVE/QMASS,QMASS2,PMASS(3),VMASS(3),AMASS(3),PMASSO(3),
     1 VMASSO(3),AMASSO(3),R(3),ROMGA(3),ROMGA2(3),OMGAMM(3),HALFAM(3),
     2 COSMM(3),SINMM(3)
      DO 1 J=1,6
      DUMMY=FREQ(J)*T
      SFREQ(J)=SIN(DUMMY)
1     CFREQ(J)=COS(DUMMY)
      DO 3 J=1,3
      MDIST(J)=MDNOM(J)
      DO 2 K=1,6
2     MDIST(J)=MDIST(J)+MDAMP(J,K)*(SFREQ(K)*CPHASE(J,K)+CFREQ(K)*
     1 SPHASE(J,K))
      DUMMY=OMGAMM(J)*T
      SINMM(J)=SIN(DUMMY)
      COSMM(J)=COS(DUMMY)
      PMASS(J)=(HALFAM(J)*T+VMASSO(J))*T+PMASSO(J)
3     VMASS(J)=AMASSO(J)*T+VMASSO(J)
      PMASS(1)=PMASS(1)+R(3)*COSMM(3)+R(2)*COSMM(2)
```

```
      PMASS(2)=PMASS(2)+R(3)*SINMM(3)+R(1)*COSMM(1)
      PMASS(3)=PMASS(3)+R(2)*SINMM(2)+R(1)*SINMM(1)
      VMASS(1)=VMASS(1)-ROMGA(3)*SINMM(3)-ROMGA(2)*SINMM(2)
      VMASS(2)=VMASS(2)+ROMGA(3)*COSMM(3)-ROMGA(1)*SINMM(1)
      VMASS(3)=VMASS(3)+ROMGA(2)*COSMM(2)+ROMGA(1)*COSMM(1)
      AMASS(1)=AMASSO(1)-ROMGA2(3)*COSMM(3)-ROMGA2(2)*COSMM(2)
      AMASS(2)=AMASSO(2)-ROMGA2(3)*SINMM(3)-ROMGA2(1)*COSMM(1)
      AMASS(3)=AMASSO(3)-ROMGA2(2)*SINMM(2)-ROMGA2(1)*SINMM(1)
      DO 9 J=1,3
      IF(J-2)4,5,6
4     JY=2
      JZ=3
      GO TO 7
5     JY=3
      JZ=1
      GO TO 7
6     JY=1
      JZ=2
7     INERT(J)=INERTO(J)+QMASS*(PMASS(JY)*PMASS(JY)+PMASS(JZ)*PMASS(JZ))
      INERT(J+3)=INERTO(J+3)+QMASS*PMASS(J)*PMASS(JY)
      INERTD(J)=QMASS2*(PMASS(JY)*VMASS(JY)+PMASS(JZ)*VMASS(JZ))
      INERTD(J+3)=QMASS*(PMASS(J)*VMASS(JY)+PMASS(JY)*VMASS(J))
      AGOMGA(J)=AGOMGO(J)+AGOMGD(J)*T
      IF(AGOMGA(J).GE.0.)GO TO 8
      AGOMGA(J)=0.
      GO TO 9
8     IF(AGOMGA(J).LE.AGOMGM)GO TO 9
      AGOMGA(J)=AGOMGM
9     CONTINUE
      RETURN
      END
```

Subroutine FINDMG

Called by. — DERIV, RKII.

Function. — Computes derivatives of the variables associated with the CMG gimbal servo dynamics.

Comments. — The equations are listed in Appendix C.

```
$IBFTC S14      LIST
      SUBROUTINE FINDMG
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      REAL NOISED
      EQUIVALENCE(OMGAXD,OMGABD(1)),(OMGAYD,OMGABD(2)),(OMGAZD,OMGABD
     1 (3))
      COMMON/CSERVO/BINPUT(3),AINPUT(3),K2BETA,K2ALPH,TAUDBE,TAUDAL,
     1 KBETA,KALPH,KSFBE,KSFAL,TAUNBE,TAUNAL,TAUBE,TAUAL,ELIMBE,ELIMAL,
     2 KTBE,KTAL,KBBETA,KBALPH,TAUMBE,TAUMAL,TFBETA,TFALPH,BLIMIT,ALIMIT
     3 ,GRBETA,GRALPH
      REAL K2BETA,K2ALPH,KBETA,KALPH,KSFBE,KSFAL,KTBE,KTAL,KBBETA,KBALPH
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
      REAL MJET,MTOT,MB,MA,MAGJET
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/FLOOUT/BETADC(3),ALPHDC(3),TELAZC,EULER(3),EPSLON(3),
     1 OMGABC(3),JET(3)
      DO 1 J=1,3
      BINPUT(J)=BETADC(J)
      AINPUT(J)=ALPHDC(J)
      IF(LIMIG(J))2,3,4
2     BINPUT(J)=AMAX1(0.,BINPUT(J))
      GO TO 3
4     BINPUT(J)=AMIN1(0.,BINPUT(J))
3     IF(LIMOG(J))5,6,7
5     AINPUT(J)=AMAX1(0.,AINPUT(J))
```

```
      GO TO 6
7     AINPUT(J)=AMIN1(0.,AINPUT(J))
6     EFBDOT(J)=(BETAD(J)*K2BETA-EFB(J))/TAUDBE
      EMBDOT(J)=(KBETA*(KSFBE*BINPUT(J)-EFB(J)-TAUNBE*EFBDOT(J))-EMB(J)
     1 )/TAUBE
      DUM=EMB(J)
      IF(ABS(DUM).GT.ELIMBE)DUM=SIGN(ELIMBE,DUM)
      TLDDB(J)=(KTBE*(DUM-KBBETA*BETAD(J))-TLOADB(J))/TAUMBE
      EFADOT(J)=(ALPHAD(J)*K2ALPH-EFA(J))/TAUDAL
      EMADOT(J)=(KALPH*(KSFAL*AINPUT(J)-EFA(J)-TAUNAL*EFADOT(J))-EMA(J)
     1 )/TAUAL
      DUM=EMA(J)
      IF(ABS(DUM).GT.ELIMAL)DUM=SIGN(ELIMAL,DUM)
1     TLDDA(J)=(KTAL*(DUM-KBALPH*ALPHAD(J))-TLOADA(J))/TAUMAL
      RETURN
      END
```

Subroutine FINDA

Called by. — DERIV.

Function. — Computes a_{ij} and a'_{ij} using equations (A - 25) to (A - 51).

```
$IBFTC S15      LIST
      SUBROUTINE FINDA
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/AGROUP/A(9,9),AP(3,3),APINV(3,3),DETAP,DIF1XB(3),AAXB(3),
     1 XBSQ(3),BAXB(3)
      COMMON/CGYRO/AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AB AG,BB BG,
     1 CB BG,AA BA,AA CA,BA CA,BB CB,DIF1,DIF2,GRAJMA,SUM1,SUM2,SUM3,
     2 SUM4,SUM5,SUM6,SUM7,SUM8,SUM9,SUM10,AGOMGO(3),AGOMGD(3),AGOMGM,
     3 HNOM
      REAL JMBETA,JMALPH
      COMMON/CBODY/IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,
     1 IXZO,IXXD,IYYD,IZZD,IXYD,IYZD,IXZD,ITERM(3)
      REAL IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,IXZO,IXXD,
     1 IYYD,IZZD,IXYD,IYZD,IXZD,ITERM,INERT(6),INERTO(6),INERTD(6)
      EQUIVALENCE(IXX,INERT(1)),(IXXO,INERTO(1)),(IXXD,INERTD(1))
      DO 1 J=1,3
      XBSQ(J)=(AB AG)*SINBSQ(J)+(BB BG)*COSBSQ(J)
      AAXB(J)=AA+(AB AG)*COSBSQ(J)+(BB BG)*SINBSQ(J)
      BAXB(J)=SUM2+XBSQ(J)
1     DIF1XB(J)=DIF1*SINB(J)*COSB(J)
      DO 6 J=1,3
      IF(J-2)2,3,4
2     J2=2
      J3=3
      GO TO 5
3     J2=3
      J3=1
      GO TO 5
4     J2=1
      J3=2
5     A(J,2*J+2)=SINA(J)*SUM3
      A(J,2*J2+2)=COSA(J2)*SUM3
      A(J,2*J+3)=COSA(J)*DIF1XB(J)
      A(J2,2*J3+3)=-SINA(J3)*DIF1XB(J3)
      A(J2,2*J+3)=SUM4+XBSQ(J)
6     A(2*J+3,2*J+3)=SUM2+XBSQ(J)
```

```
      DO 11 J=1,3
      IF(J-2)7,8,9
7     J2=2
      J3=3
      GO TO 10
8     J2=3
      J3=1
      GO TO 10
9     J2=1
      J3=2
10    AP(J,J)=INERT(J)+COSASQ(J)*AAXB(J)+SINASQ(J2)*AAXB(J2)+SUM8*
     1 (SINASQ(J)+COSASQ(J2))-A(J,2*J+3)**2/A(2*J+3,2*J+3)-A(J,2*J2+3)**
     2 2/A(2*J2+3,2*J2+3)+(SUM9/BAXB(J3))*(XBSQ(J3)+BA)
      AP(J,J2)=-INERT(J+3)+SINA(J2)*COSA(J2)*(SUM8-AAXB(J2)+DIF1XB(J2)**
     1 2/BAXB(J2))+SUM10*(A(J,2*J+3)/BAXB(J)+A(J2,2*J3+3)/BAXB(J3))
11    AP(J2,J)=AP(J,J2)
      RETURN
      END
```

Subroutine FINDB

Entry points. — FINDB, FINDBP.

Called by. — DERIV, RKII.

Function. — Computes b_i and b_i' by equations (A-52) to (A-69).

```
$IBFTC S16      LIST
      SUBROUTINE FINDB
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/RATES/OMEGA(3,3),OMEGAP(3,3),OMGASQ(3),OMGAXP(3)
      COMMON/AGROUP/A(9,9),AP(3,3),APINV(3,3),DETAP,DIF1XB(3),AAXB(3),
     1 XBSQ(3),BAXB(3)
      COMMON/BGROUP/BAUX1(3),BAUX2(3),BAUX3(3),BAUX4(3),BAUX5(3),
     1 BAUX6(3),BAUX7(3),BAUX8(3),BAUX9(3),BAUX10(3),BAUX11(3),
     2 BAUX12(3),BAUX13(3),BAUX14(3),BAUX15(3),BAUX16(3),BAUX17(3),
     3 BAUX18(3),BAUX19(3),BSMALL(9),BLARGE(9),BPRIME(9)
      COMMON/CGYRO/AA,BA,CA,AB,BB,CB,AG,BG,JMBETA,JMALPH,AB AG,BB BG,
     1 CB BG,AA BA,AA CA,BA CA,BB CB,DIF1,DIF2,GRAJMA,SUM1,SUM2,SUM3,
     2 SUM4,SUM5,SUM6,SUM7,SUM8,SUM9,SUM10,AGOMGO(3),AGOMGD(3),AGOMGM,
     3 HNOM
      REAL JMBETA,JMALPH
      COMMON/CBODY/IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,
     1 IXZO,IXXD,IYYD,IZZD,IXYD,IYZD,IXZD,ITERM(3)
      REAL IXX,IYY,IZZ,IXY,IYZ,IXZ,IXXO,IYYO,IZZO,IXYO,IYZO,IXZO,IXXD,
     1 IYYD,IZZD,IXYD,IYZD,IXZD,ITERM,INERT(6),INERTO(6),INERTD(6)
      EQUIVALENCE(IXX,INERT(1)),(IXXO,INERTO(1)),(IXXD,INERTD(1))
      COMMON/CSERVO/BINPUT(3),AINPUT(3),K2BETA,K2ALPH,TAUDBE,TAUDAL,
     1 KBETA,KALPH,KSFBE,KSFAL,TAUNBE,TAUNAL,TAUBE,TAUAL,ELIMBE,ELIMAL,
     2 KTBE,KTAL,KBBETA,KBALPH,TAUMBE,TAUMAL,TFBETA,TFALPH,BLIMIT,ALIMIT
     3 ,GRBETA,GRALPH
      REAL K2BETA,K2ALPH,KBETA,KALPH,KSFBE,KSFAL,KTBE,KTAL,KBBETA,KBALPH
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
```

```
      REAL MJET,MTOT,MB,MA,MAGJET
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/FLOOUT/BETADC(3),ALPHDC(3),TELAZC,EULER(3),EPSLON(3),
     1 OMGABC(3),JET(3)
      COMMON/DIST/MDIST(3),MDNOM(3),SPHASE(3,6),CPHASE(3,6),SFREQ(6),
     1 CFREQ(6),MDAMP(3,6),FREQ(6),TQEMM(3)
      REAL MDIST,MDNOM,MDAMP
      COMMON/MOVE/QMASS,QMASS2,PMASS(3),VMASS(3),AMASS(3),PMASSO(3),
     1 VMASSO(3),AMASSO(3),R(3),ROMGA(3),ROMGA2(3),OMGAMM(3),HALFAM(3),
     2 COSMM(3),SINMM(3)
      DO 5 J=1,3
      DUM=JET(J)
      MJET(J)=DUM*MAGJET(J)
      MTOT(J)=MJET(J)+MDIST(J)
      IF(J-2)1,2,3
1     JP1=2
      JP2=3
      GO TO 4
2     JP1=3
      JP2=1
      GO TO 4
3     JP1=1
      JP2=2
4     BAUX1(J)=OMEGAP(J,J)*ALPHAD(J)
      BAUX2(J)=OMEGAP(JP2,J)*ALPHAD(J)
      BAUX3(J)=OMEGA(JP2,J)*OMEGA(JP1,J)*(BB CB)
      BAUX4(J)=SINB(J)*BAUX3(J)
      BAUX5(J)=COSB(J)*BAUX3(J)
      BAUX6(J)=OMEGA(JP2,J)*(OMEGA(J,J)*DIF2+AG OMGA(J))
      BAUX7(J)=SINB(J)*BAUX6(J)
      BAUX8(J)=COSB(J)*BAUX6(J)
      BAUX9(J)=OMEGAP(JP2,J)*OMEGAP(J,J)*(AA CA)
      BAUX10(J)=AAXB(J)*BAUX2(J)
      BAUX11(J)=OMEGA(JP1,J)*(AB AG)
```

```
      BAUX12(J)=OMEGA(J,J)*(BB BG)
      BAUX13(J)=(COSB(J)*BAUX11(J)+SINB(J)*BAUX12(J))*BETAD(J)
      BAUX14(J)=SUM7*BAUX1(J)
      BAUX15(J)=OMEGAP(J,J)*OMEGAP(JP1,J)*(AA BA)
      BAUX16(J)=OMEGAP(JP1,J)*OMEGAP(JP2,J)*(BA CA)
      BAUX17(J)=(OMEGAP(JP2,J)+GRBETA*BETAD(J))*JMBETA
      BAUX18(J)=OMEGAP(J,J)*BAUX17(J)
      BAUX19(J)=OMEGAP(JP1,J)*BAUX17(J)
      OMGASQ(J)=OMEGAB(J)**2
      OMGAXP(J)=OMEGAB(J)*OMEGAB(JP1)
      BSMALL(2*J+2)=OMEGA(JP1,J)*(OMEGA(J,J)*DIF1+AG OMGA(J))
     1 -(SUM 3)*BAUX1(J)
5     BSMALL(2*J+3)=SINB(J)*COSB(J)*DIF1*(BAUX2(J)-BETAD(J)* OMEGAP(JP1,
     1 J))          +OMEGAP(J,J)*BETAD(J)*XBSQ(J)+BAUX4(J)-BAUX8(J)
     2 -BAUX9(J)+BAUX18(J)-AGOMGD(J)*SINB(J)
      DO 10 J=1,3
      IF(J-2)6,7,8
6     JP1=2
      JP2=3
      GO TO 9
7     JP1=3
      JP2=1
      GO TO 9
8     JP1=1
      JP2=2
9     TQEMM(J)=      INERTD(J+3)*OMEGAB(JP1)-(INERTD(J)*OMEGAB(J)-
     1 INERTD(JP2+3)*OMEGAB(JP2)+QMASS*(OMEGAB(JP1)*(PMASS(J)*VMASS(JP1)
     2 -PMASS(JP1)*VMASS(J))+OMEGAB(JP2)*(PMASS(J)*VMASS(JP2)-PMASS(JP2)
     3 *VMASS(J))+PMASS(JP1)*AMASS(JP2)-PMASS(JP2)*AMASS(JP1)))
      ITERM(J)=(OMGASQ(JP2)-OMGASQ(JP1))*INERT(JP1+3)+OMGAXP(JP1)*(
     1 INERT(JP2)-INERT(JP1))+OMGAXP(JP2)*INERT(J+3)-OMGAXP(J)*
     2 INERT(JP2+3)-TQEMM(J)
10    BSMALL(J)=COSA(J)*(BAUX10(J)+BAUX5(J)+BAUX7(J)-BAUX13(J)
     1 +BAUX16(J)-BAUX19(J)-AGOMGD(J)*COSB(J))
     2 -SINA(JP1)*(BAUX10(JP1)+BAUX5(JP1)+BAUX7(JP1)-BAUX13(JP1)
     3 +BAUX16(JP1)-BAUX19(JP1)-AGOMGD(JP1)*COSB(JP1))
     4 +SINA(J)*(BSMALL(2*J+2)-BAUX14(J)+BAUX15(J))
     5 +COSA(JP1)*(BSMALL(2*JP1+2)-BAUX14(JP1)+BAUX15(JP1))
```

```
     6 +DIF1XB(JP2)*BAUX2(JP2)+BAUX4(JP2)-BAUX8(JP2)-BAUX9(JP2)
     7 -BETAD(JP2)*(BAUX11(JP2)*SINB(JP2)-BAUX12(JP2)*COSB(JP2))
     8 -ITERM(J)+BAUX18(JP2)+GRAJMA*(ALPHAD(J)*OMEGAB(JP2)-
     9 ALPHAD(JP1)*OMEGAB(JP1))-AGOMGD(JP2)*SINB(JP2)
      ENTRY FINDBP
12    DO 20 J=1,6
      K=J
      I=J+3
      JP2=2
      IF(K.GE.4)GO TO 21
      DUM=1.
      GO TO 22
21    K=J-3
      I=K
      JP2=3
      DUM=-1.
22    SUMTQE=TLOADB(J)+BETAD(I)*AGOMGA(K)*COSB(K)*DUM
      COUL=BINPUT(JP2+25)
      IF(BETAD(J).NE.0.)GO TO 23
      IF(ABS(SUMTQE).LE.COUL)GO TO 24
      MB(J)=TLOADB(J)-SIGN(COUL,SUMTQE)
      GO TO 20
24    MB(J)=TLOADB(J)-SUMTQE
      GO TO 20
23    NDUM=JP2+2*K
      ZINRT=A(NDUM,NDUM)
      COUL=SIGN(COUL/ZINRT,BETAD(J))
      DUM1=SUMTQE/ZINRT-COUL
      DUMDEL=DUM1*DELTAT
      IF(DUMDEL/BETAD(J).GE.-1.)GO TO 25
      DUM2=DUM1+COUL+COUL
      IF(DUM2/DUM1.LE.0.)GO TO 26
      DUM1=DUM2+BETAD(J)*(DUM2-DUM1)/DUMDEL
      GO TO 25
26    DUM1=-BETAD(J)/DELTAT
25    MB(J)=TLOADB(J)-SUMTQE+DUM1*ZINRT
20    CONTINUE
      DO 13 J=1,3
```

```
      BLARGE(J)=BSMALL(J)+MTOT(J)
      BLARGE(2*J+2)=BSMALL(2*J+2)+MB(J)
13    BLARGE(2*J+3)=BSMALL(2*J+3)+MA(J)
      DO 11 J=4,9
11    BPRIME(J)=BLARGE(J)/A(J,J)
      BPRIME(1)=BLARGE(1)-(A(1,4)*BPRIME(4)+A(1,5)*BPRIME(5)
     1 +A(1,6)*BPRIME(6)+A(1,7)*BPRIME(7)+A(1,9)*BPRIME(9))
      BPRIME(2)=BLARGE(2)-(A(2,5)*BPRIME(5)+A(2,6)*BPRIME(6)
     1 +A(2,7)*BPRIME(7)+A(2,8)*BPRIME(8)+A(2,9)*BPRIME(9))
      BPRIME(3)=BLARGE(3)-(A(3,4)*BPRIME(4)+A(3,5)*BPRIME(5)
     1 +A(3,7)*BPRIME(7)+A(3,8)*BPRIME(8)+A(3,9)*BPRIME(9))
      RETURN
      END
```

Subroutine EVDER

Entry points. — EVDER, EVDERP.

Called by. — DERIV, RKII.

Calls. — SOLVEX.

Function. — Computes Euler rates (Appendix B) and $\dot{\alpha}_i$ and $\dot{\beta}_i$, equations (A-79) to (A-85).

Comments. — This routine can set any selected state vector derivative(s) to zero. This will "freeze" the associated state vector element at its initial value.

```
$IBFTC S17      LIST
      SUBROUTINE EVDER
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      REAL NOISED
      EQUIVALENCE(OMGAXD,OMGABD(1)),(OMGAYD,OMGABD(2)),(OMGAZD,OMGABD
     1 (3))
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/RATES/OMEGA(3,3),OMEGAP(3,3),OMGASQ(3),OMGAXP(3)
      COMMON/AGROUP/A(9,9),AP(3,3),APINV(3,3),DETAP,DIF1XB(3),AAXB(3),
     1 XBSQ(3),BAXB(3)
      COMMON/BGROUP/BAUX1(3),BAUX2(3),BAUX3(3),BAUX4(3),BAUX5(3),
     1 BAUX6(3),BAUX7(3),BAUX8(3),BAUX9(3),BAUX10(3),BAUX11(3),
     2 BAUX12(3),BAUX13(3),BAUX14(3),BAUX15(3),BAUX16(3),BAUX17(3),
     3 BAUX18(3),BAUX19(3),BSMALL(9),BLARGE(9),BPRIME(9)
      COMMON/TORQUE/MJET(3),MTOT(3),MB(3),MA(3),MAGJET(3),FUEL(3),
     1 FLOWRT(3),FUELT
      REAL MJET,MTOT,MB,MA,MAGJET
      PSID=(SINPHI*OMEGAY+COSPHI*OMEGAZ)/COSTHE
      THETAD=COSPHI*OMEGAY-SINPHI*OMEGAZ
      PHID=OMEGAX+PSID*SINTHE
      DO 5 J=1,3
      DRBETA(J)=BETAD(J)
      DRALPH(J)=ALPHAD(J)
      HEXTD(J)=0.
```

```
      DO 5 K=1,3
5     HEXTD(J)=HEXTD(J)+DIRCO(J,K)*MTOT(K)
      ENRGBD(1)=ABS(TLOADB(1)* OMEGA(3,1))
      ENRGBD(2)=ABS(TLOADB(2)* OMEGA(1,2))
      ENRGBD(3)=ABS(TLOADB(3)* OMEGA(2,3))
      ENRGAD(1)=ABS(TLOADA(1)*OMEGAP(2,1))
      ENRGAD(2)=ABS(TLOADA(2)*OMEGAP(3,2))
      ENRGAD(3)=ABS(TLOADA(3)*OMEGAP(1,3))
      ENTRY EVDERP
      CALL SOLVEX(AP,APINV,DETAP,OMGABD,BPRIME,1)
      BETADD(1)=BPRIME(4)-(A(1,4)*OMGAXD+A(3,4)*OMGAZD)/A(4,4)
      BETADD(2)=BPRIME(6)-(A(1,6)*OMGAXD+A(2,6)*OMGAYD)/A(6,6)
      BETADD(3)=BPRIME(8)-(A(2,8)*OMGAYD+A(3,8)*OMGAZD)/A(8,8)
      DO 2 J=1,3
2     ALPHDD(J)=BPRIME(2*J+3)-(A(1,2*J+3)*OMGAXD+A(2,2*J+3)*OMGAYD
     1 +A(3,2*J+3)*OMGAZD)/A(2*J+3,2*J+3)
      DO 3 J=1,79
      IF(NOINT(J).EQ.0) GO TO 4
      K=NOINT(J)
3     OMGABD(K)=0.
4     RETURN
      END
```

Subroutine ENERGY (INDRS)

Called by. — RKII, INITIL.

Calls. — RANO.

Function. — This routine performs computations needed once each complete integration step.

Comments — Certain items used in the environment integration cannot be computed every time the derivatives are evaluated (at least four times each integration step). These include:

a. CMG gimbal limit detection

b. Star tracker noise computation

c. Peak power detection.

All gimbals are checked for limit crossings. If any gimbals change their limit status, the argument INDRS is set to one.

```
$IBFTC S18     LIST
      SUBROUTINE ENERGY(INDRS)
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/RKYDV/OMGABD(3),BETADD(3),ALPHDD(3),DRBETA(3),DRALPH(3),
     1 PHID,THETAD,PSID,TLDDB(3),TLDDA(3),EMBDOT(3),EMADOT(3),EFBDOT(3),
     2 EFADOT(3),HEXTD(3),ENRGBD(3),ENRGAD(3),HORIZD(2),NOISED(4),
     3 STV1D(4),STV2D(4),STV3D(4),STV3PD(4),STV5D(4),SQTID(4),QTBD(4)
      REAL NOISED
      EQUIVALENCE(OMGAXD,OMGABD(1)),(OMGAYD,OMGABD(2)),(OMGAZD,OMGABD
     1 (3))
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/CSERVO/BINPUT(3),AINPUT(3),K2BETA,K2ALPH,TAUDBE,TAUDAL,
     1 KBETA,KALPH,KSFBE,KSFAL,TAUNBE,TAUNAL,TAUBF,TAUAL,ELIMBE,ELIMAL,
     2 KTBE,KTAL,KBBETA,KBALPH,TAUMBE,TAUMAL,TFBETA,TFALPH,BLIMIT,ALIMIT
     3 ,GRBETA,GRALPH
      REAL K2BETA,K2ALPH,KBETA,KALPH,KSFBE,KSFAL,KTBE,KTAL,KBBETA,KBALPH
      COMMON/POWER/PMAXB(3),PMAXA(3),DTLIST(300),KDT
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
     1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,
     2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DELW(3),STBIAS(4),STSIG,
     3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),ELBORE(2),AZBORE(2),QST(4)
     4 ,WNG(4),HRZACT(2)
      REAL JINRT
      DATA XG/0377777777777/
      KDT=KDT+1
      IF(KDT.LE.300)DTLIST(KDT)=DELTAT
      DO 13 J=1,3
      POWR=ABS(ENRGBD(J))
```

```
      IF(POWR.GT.PMAXB(J))PMAXB(J)=POWR
      POWR=ABS(ENRGAD(J))
13    IF(POWR.GT.PMAXA(J))PMAXA(J)=POWR
      INDRS=0
      DO 10  J=1,3
      IF(LIMIG(J))1,2,3
1     IF(BETA(J).LE.-BLIMIT) GO TO 4
5     LIMIG(J)=0.
      INDRS=1
      GO TO 2
3     IF(BETA(J)-BLIMIT) 5,4,4
2     IF(ABS(BETA(J)).LE.BLIMIT)GO TO 4
      INDRS=1
      IF(BETA(J).GT.0.)GO TO 6
      LIMIG(J)=-1
      GO TO 4
6     LIMIG(J)=1
4     IF(LIMOG(J))7,8,9
7     IF(ALPHA(J).LE.-ALIMIT)GO TO 10
11    LIMOG(J)=0.
      INDRS=1
      GO TO 8
9     IF(ALPHA(J)-ALIMIT)11,10,10
8     IF(ABS(ALPHA(J)).LE.ALIMIT)GO TO 10
      INDRS=1
      IF(ALPHA(J).GT.0.)GO TO 12
      LIMOG(J)=-1
      GO TO 10
12    LIMOG(J)=1
10    CONTINUE
      GO TO (14,15,16,14,14,14,14),MODE
15    NDUM=3
      GO TO 17
16    NDUM=4
17    DO 18 J=1,NDUM
      CALL RANO(STSIG,STBIAS(J),XG,WNG(J))
18    NOISED(J)=(WNG(J)-NOISE(J))/TAUWNG
14    RETURN
```

END

Subroutine STORE (INDEX)

Called by. — MAIN, INITIL.

Function. — Stores the sensed variable(s) corresponding to the event time specified in the argument, INDEX, in the control computer input buffer.

```
$IBFTC S19      LIST
      SUBROUTINE STORE(INDEX)
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VFO,FIXWO(10),WEARTH
      REAL MU
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/SENSOR/ESTAR(2),ASTAR(2),TAUWNG,TAU1,TAU2N,TAU2D,TAU3N,
     1 TAU3D,TAU3NP,TAU3DP,TAU5,STKT,STG2,STG3,STAL,STKM,STKV,STTF,STKR,
     2 JINRT(4),ETA(2),TAUHRZ,GHORIZ,DELW(3),STBIAS(4),STSIG,
     3 SXI(2),SYI(2),SZI(2),SX(2),SY(2),SZ(2),ELBORE(2),AZBORE(2),QST(4)
     4 ,WNG(4),HRZACT(2)
      REAL JINRT
      DO 1 J=1,3
      ZREAL(J)=OMEGAB(J)+DELW(J)
      ZREAL(J+9)=BETA(J)
1     ZREAL(J+12)=ALPHA(J)
      DO 2 J=1,2
      ZREAL(J+3)=QTB(J)
      ZREAL(J+5)=QTB(J+2)
2     ZREAL(J+7)=HORIZ(J)+ETA(J)
      DO 3 J=1,6
      IF(ABS(ZREAL(J+9)).LE.PIE)GO TO 3
      DUMMY=   ABS(ZREAL(J+9))/PIE
      IDUM=DUMMY
      IDUM=((IDUM+1)/2)*2
      DUMMY=IDUM
      ZREAL(J+9)=ZREAL(J+9)-SIGN(DUMMY,ZREAL(J+9))*PIE
3     CONTINUE
```

```
      DO 4 J=1,4
      IF(ABS(ZREAL(J+3)).LE.PIE) GO TO 4
      DUMMY=ABS(ZREAL(J+3))/PIE
      IDUM=DUMMY
      IDUM=((IDUM+1)/2)*2
      DUMMY=IDUM
      ZREAL(J+3)=ZREAL(J+3)-SIGN(DUMMY,ZREAL(J+3))*PIE
4     CONTINUE
      IF(INDEX.EQ.1)NLOWER=1
      NUPPER=NLOWER+NUMBER(INDEX)-1
      DO 5  K=NLOWER,NUPPER
      I=NORDER(K)
5     ZLATE(I)=ZREAL(I)
      NLOWER=NUPPER+1
      RETURN
      END
```

Subroutine FANDG

Called by. — INITIL; Optionally, MANUAL.

Function. — Computes coefficients f_i and g_i of the power series for position and velocity.

Comments. — The f and g series are discussed in Volume I.

```
$IBFTC S20      LIST
      SUBROUTINE FANDG
C  THIS ROUTINE SIMULATES GROUND TRANSMISSION OF NAVIGATION UPDATE INFO
      COMMON/RKC/U(79),UMIN(79),DTMIN,DTEST,DELTAT,NDOUBL,NINT,T,TRETRN,
     1 NOINT(79)
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      COMMON/CONSTS/RTODEG,DEGTOR,RE,MU,PIE,WO(10),VEO,FIXWO(10),WEARTH
      REAL MU
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNPO,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NPO,FL2NP1
      COMMON/QUANT/NBIN(25),NBOUT(16),NFXPNT
      COMMON/EXP1V/ACDBL,ACDOT,ADSAVE,COSAC,COSWT,        ZOMEGA(3),
     COMEGAE(3),RO(3),S,SDOT,SDUM,SINAC,SINWT,SPRIME,SREL(3),SSQ,TANAC,
     CV(3),VC(3),VDOUB(3),WE,WE2,WE3,WE4
      COMMON/TVECT/DELT,H,TIME(10),NPASS,NSLOW,XNSLOW
      COMMON/NAV/F(10),FDOT,FDUM(10),FTOT,G(10),GDOT,GDUM(10),GTOT,P(3),
     CPO(3),PDOT(3),PDOTO(3)
      INTEGER   NPO(3),NPDOTO(3),FIXF(10),FIXG(10),FIXT,FIXDT,FPTGT(3)
      EQUIVALENCE (NPO(1),PO(1)),(NPDOTO(1),PDOTO(1)),(FIXF(1),F(1)),
     1 (FIXG(1),G(1)),(FIXT,TIME(1)), (FIXDT,DELT),(FPTGT(1),RO(1))
      DUM1=WEARTH*T
      DUM2=SIN(DUM1)
      DUM3=COS(DUM1)
      RO(1)=PTARGT(1)*DUM3+PTARGT(3)*DUM2
      RO(2)=PTARGT(2)
      RO(3)=PTARGT(3)*DUM3-PTARGT(1)*DUM2
      DELT=TCYCLE*XNSLOW
      TIME(1)=-DELT
      DUM1=0.
      DUM2=0.
      DUM3=0.
      DO 1 J=1,3
```

```
      PO(J)=POS(J)
      DUM1=DUM1+POS(J)*VEL(J)
      DUM2=DUM2+POS(J)*POS(J)
      DUM3=DUM3+VEL(J)*VEL(J)
1     PDOTO(J)=VEL(J)
      DUM1=DUM1/(RE*VEO)
      DUM2=DUM2/(RE*RE)
      DUM4=SQRT(DUM2)
      DUM3=DUM3/(VEO*VEO)-1./DUM4
      ZU=1./(DUM2*DUM4)
      ZS=DUM1/DUM2
      ZE=DUM3/DUM2
      F(1)=0.
      F(2)=-ZU
      F(3)=3.*ZS*ZU
      F(4)=(-15.*ZS**2*ZU+3.*ZE*ZU+ZU**2)
      F(5)=105.*ZS**3*ZU+ZS*ZU*(-45.*ZE-15.*ZU)
      F(6)=-945.*ZS**4*ZU+ZS**2*ZU*(630.*ZE+210.*ZU)+ZE*ZU*(-24.*ZU-45.*
     1 ZE)-ZU**3
      F(7)=ZS**3*ZU*(10395.*ZS**2-9450.*ZE-3150.*ZU)+ZS*(ZE*ZU*(882.*
     1 ZU+1575.*ZE)+63.*ZU**3)
      F(8)=ZS**4*ZU*(-135135.*ZS**2+155925.*ZE+51975.*ZU)+ZS**2*(ZE*ZU*
     1 (-24570.*ZU-42525.*ZE)-2205.*ZU**3)+ZE*ZU*(117.*ZU**2+1575.*ZE**2
     2 +1107.*ZE*ZU)+ZU**4
      F(9)=ZS**5*ZU*(2027025.E0*ZS**2-2837835.E0*ZE-945945.*ZU)+ZS**3*
     1 (ZE*ZU*(644490.*ZU+1091475.E0*ZE)+65835.*ZU**3)+ZS*(ZE*ZU*
     2 (-10935.*ZU**2-99225.*ZE**2-74385.*ZE*ZU)-255.*ZU**4)
      F(10)=ZS**6*ZU*(-34459425.E0*ZS**2+567567.E2*ZE+189189.E2*ZU)+
     1 ZS**4*(ZE*ZU*(-1702701.E1*ZU-2837835.E1*ZE)-189189.E1*ZU**3)+
     2 ZS**2*(ZE*ZU*(599940.*ZU**2+43659.E2*ZE**2+342144.E1*ZE*ZU)+
     3 21120.*ZU**4)+ZE*ZU*(-498.*ZU**3-99225.*ZE**3+ZE*ZU*(-85410.*ZE
     4 -15066.*ZU))-ZU**5
      G(1)=1.
      G(2)=0.
      G(3)=-ZU
      G(4)=6.*ZS*ZU
      G(5)=ZU*(-45.*ZS**2+9.*ZE+ZU)
      G(6)=ZS*ZU*(420.*ZS**2-180.*ZE-30.*ZU)
```

```
      G(7)=ZS**2*ZU*(-4725.*ZS**2+3150.*ZE+630.*ZU)+ZE*ZU*(-54.*ZU-225.*
     1 ZE)-ZU**3
      G(8)=ZS**3*ZU*(62370.*ZS**2-56700.*ZE-12600.*ZU)+ZS*(ZE*ZU*(3024.*
     1 ZU+9450.*ZE)+126.*ZU**3)
      G(9)=ZS**4*ZU*(-945945.*ZS**2+1091475.E0*ZE+259875.*ZU)+ZS**2*(
     1 ZE*ZU*(-111510.*ZU-297675.*ZE)-6615.*ZU**3)+ZE*ZU*(243.*ZU**2+
     2 11025.*ZE**2+4131.*ZE*ZU)+ZU**4
      G(10)=ZS**5*ZU*(162162.E2*ZS**2-2270268.E1*ZE-567567.E1*ZU)+ZS**3*
     1 (ZE*ZU*(361746.E1*ZU+87318.E2*ZE)+263340.*ZU**3)+ZS*(ZE*ZU*(
     2 -35100.*ZU**2-793800.*ZE**2-371790.*ZE*ZU)-510.*ZU**4)
      IF(NFXPNT.NE.0)GO TO 3
      DO 2 J=2,10
      F(J)=WO(J)*F(J)
2     G(J)=WO(J-1)*G(J)
      RETURN
3     FIXDT=F2NM10*DELT
      FIXT=-FIXDT
      DO 4 J=2,10
      FIXF(J)=F(J)*FIXWO(J)*FL2NPO
4     FIXG(J)=G(J)*FIXWO(J-1)*FL2NPO
      FIXG(1)=G(1)*FL2NM1
      DO 5 J=1,3
      NPO(J)=PO(J)*F2NM25
      NPDOTO(J)=PDOTO(J)*F2NM15
5     FPTGT(J)=RO(J)*F2NM25
      RETURN
      END
```

Subroutine UPDAT3

Called by. — INITIL; Optionally, MANUAL.

Function. — Computes phase angle, η, used for transforming orbital rate to body commands ω_{xc}, ω_{yc}.

```
$IBFTC S21      LIST
      SUBROUTINE UPDAT3
C  UPDATE FOR EXPERIMENT  THREE
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/PVDATA/POS(3),VEL(3),POSO(3),VELO(3),ECCENT,ENOW,MEANA,
     1 MEANAO,FVECT(3),CETAO,SETAO,PVCON(4),PTARGT(3)
      REAL MEANA,MEANAO
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNPO,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NPO,FL2NP1
      COMMON/QUANT/NBIN(25),NBOUT(16),NFXPNT
      COMMON/FIXOUT/BDOTC(3),ADOTC(3),AC,ED(3),EP(3),WC(3),NJET(3)
      COMMON/EXP3V/ANGLE,COSDUM,COSL,COSLR,COSTH,DELANG,DELX,DELY,DELZ,
     CDBLPI,ETADOT,KC,PREV,PSP(3),R,SINDUM,SINL,SINLR,SIN2LR,SINTH,
     CSPX(2),SPY(2),SPZ(2),S1XG,S1ZG,S2ZG,                WBAR,Z,ZEXI,
     CZEYI,ZEZI,EPZDOT
      EQUIVALENCE (NANGLE,ANGLE),(NPREV,PREV),(NETADT,ETADOT),
     1 (NDANG,DELANG)
      DIMENSION VMOD(3)
      EP(3)=0.
      DUM1=0.
      DUM2=0.
      DO 1 J=1,3
      DUM1=DUM1+POS(J)*VEL(J)
1     DUM2=DUM2+POS(J)*POS(J)
      DUM1=DUM1/DUM2
      DUM2=0.
      DUM3=0.
```

```
      DO 2 J=1,3
      VMOD(J)=VEL(J)-DUM1*POS(J)
      DUM2=DUM2-VMOD(J)*DIRCO(J,1)
2     DUM3=DUM3+VMOD(J)*DIRCO(J,2)
      ANGLE=ATAN2(DUM3,DUM2)
      IF(NFXPNT.NE.0) GO TO 3
      ANGLE=ANGLE-DELANG
      PREV=OMEGAZ-ETADOT
      RETURN
3     NANGLE=ANGLE*FL2NM2
      NPREV=IFIX(OMEGAZ*FLNP5)-NETADT
      NANGLE=NANGLE-NDANG
      RETURN
      END
```

Function COST (XXXX, YYYY, J)

Called by. — COMP.

Function. — Iterative control law cost function.

Comments. — Figure 3-17 is the math flow of this subroutine. It is anticipated that the user may wish to alter this routine. To easily make these changes, the subroutine is programmed in floating-point FORTRAN.

The listing commentary supplies details.

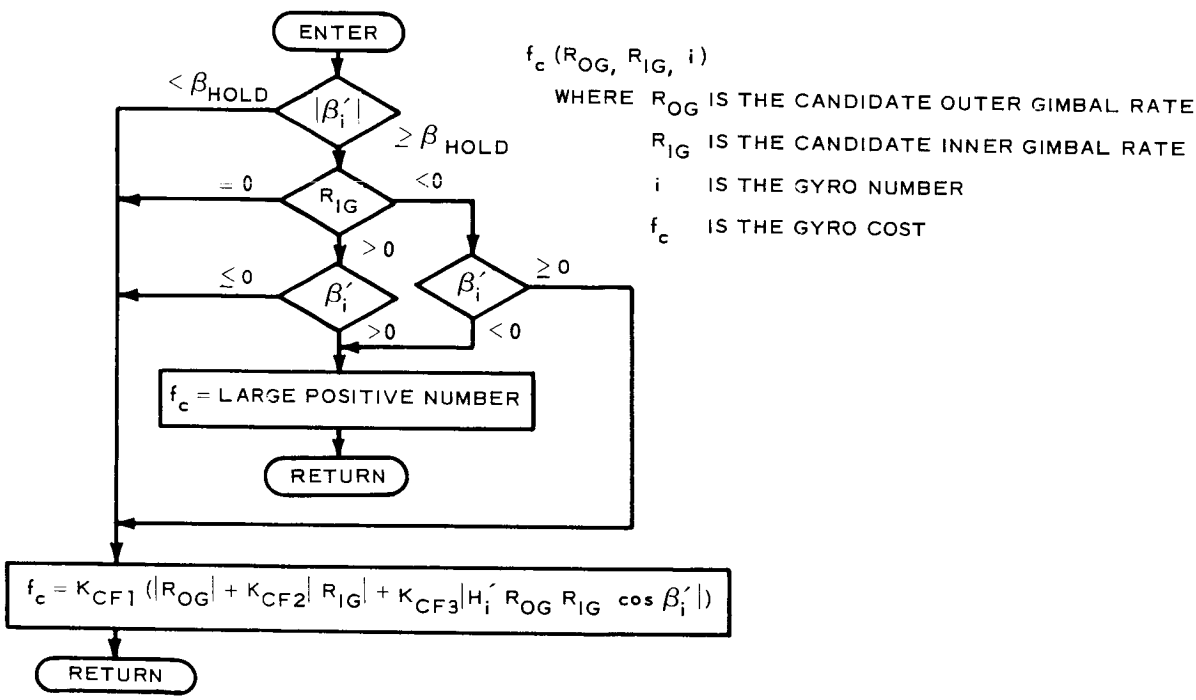


Figure 3-17. Control Computer Math Flow Cost Function Subroutine

```
$IBFTC S22     LIST
      FUNCTION COST(XXXX,YYYY,J)
C
C****  THIS ROUTINE REPRESENTS THE COST FUNCTION FOR THE ITERATIVE
C      CONTROL LAW. TO EASE THE PROGRAMMING TASK INVOLVED IN CHANGING
C      THIS FUNCTION, IT IS WRITTEN IN FLOATING POINT FORTRAN,FOR BOTH
C      FLOATING AND FIXED POINT CONTROL COMPUTERS.
C
      COMMON/RKYV/OMEGAB(3),BETAD(3),ALPHAD(3),BETA(3),ALPHA(3),PHI,
     1 THETA,PSI,TLOADB(3),TLOADA(3),EMB(3),EMA(3),EFB(3),EFA(3),HEXT(3)
     2 ,ENRGYB(3),ENRGYA(3),HORIZ(2),NOISE(4),STV1(4),STV2(4),STV3(4),
     3 STV3P(4),STV5(4),SQTI(4),QTB(4)
      REAL NOISE
      EQUIVALENCE(OMEGAX,OMEGAB(1)),(OMEGAY,OMEGAB(2)),(OMEGAZ,OMEGAB
     1 (3))
      COMMON/SINCO/SINB(3),SINA(3),SINPHI,SINTHE,SINPSI,COSB(3),COSA(3),
     1 COSPHI,COSTHE,COSPSI,SINBSQ(3),SINASQ(3),COSBSQ(3),COSASQ(3),
     2 DIRCO(3,3)
      COMMON/IOCONT/ZREAL(15),NUMBER(15),NORDER(15),TEVENT(15),TMATCH,
     1 NEVENT,EVENTT(15),TCYCLE,NCOST1,NMAN1,NPRINT,NPRCTL,TEND,LNECNT
      COMMON/FLOTIN/ZLATE(15),EULRDC(3),EULRC(3),TMAN,AGOMGA(3),MODE,
     1 FBSEL,UPDATE,NGAIN,LAW,MODCOM,LIMIG(3),LIMOG(3)
      INTEGER FBSEL,UPDATE
      COMMON/FLOTSC/FLNM7,FLNM6,FLNM5,FLNM4,FLNM3,FLNM2,FLNM1,FLNP0,
     CFLNP1,FLNP2,FLNP3,FLNP4,FLNP5,FLNP6,FLNP7,FLNP8,FLNP9,
     C FLNM11,FLNM10,FLNM8,FLNP12,F2NM25,F2NM15,F2NM10,FL2NM2,FL2NM1,
     C FL2NP0,FL2NP1
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/QUANT/NBIN(25),NBOUT(16),NFXPNT
      COMMON/CONTL2/DELA(3),DELB(3),DOT1(3),DOT2(3),DOT3(3),DUM1,DUM2,
     CKSAVE,MAGASQ(3),MAGBSQ(3),TREM(3),TRQPRD(3),UNITVH(3,3)
     C ,BHOLD,BSELFD,BDOTDS,MBMAX
      REAL KSAVE,MAGASQ,MAGBSQ
      EQUIVALENCE (ARATE,NARATE),(BRATE,NBRATE),(XCOST,NCOST)
      DIMENSION CFK(3)
      DATA CFK /.016,1.,1. /
      NAMELIST/COSTK/ CFK
```

```
      ARATE=XXXX
      BRATE=YYYY
      IF(NFXPNT.EQ.0)GO TO 97
      ARATE=FLOAT(NARATE)/FLNP2
      BRATE=FLOAT(NBRATE)/FLNP2
   97 IF(NCOST1.EQ.0)GO TO 1
      NCOST1=0
C
C
C****  THE NEXT SECTION (THRU STATEMENT 1) IS EXECUTED ON FIRST PASS
C      THRU THIS ROUTINE (EACH RUN). USE THIS SECTION TO READ ANY
C      NEEDED DATA.
C
C
      READ(5,COSTK)
      WRITE(6,COSTK)
C
C
C****  END OF SINGLE PASS SECTION
C
C
C****  THE NEXT SECTION IS THE COST FUNCTION. CONSIDER THESE GUIDELINES.
C
C         1. THE FIRST STATEMENT IS GIVEN THE STATEMENT NUMBER 1
C         2. THERE SHOULD BE A STATEMENT OF THE FORM,
C
C                XCOST = EXPRESSION   , WHERE EXPRESSION INVOLVES THE
C                                       PSEUDO-ARGUMENTS (ARATE,BRATE,J)
C                                       AND ANY OTHER APPROPRIATE
C                                       VARIABLES AND CONSTANTS.
C
C         3. TO SCALE THE COST FUNCTION, BE AWARE THAT
C
C                ARATE AND BRATE HAVE A MAXIMUM ABSOLUTE VALUE OF
C                   .25 RAD./SEC.
C
C                XCOST MUST RANGE FROM  0. TO 1.
C
```

```
C
  1    IF((ABS(BETA(J)).LT.BHOLD).OR.(BRATE.EQ.0.).OR.(BRATE.GT.0..AND.
     X BETA(J).LT.0.).OR.(BRATE.LT.0..AND.BETA(J).GT.0.))GO TO 2
       XCOST=100.
       GO TO 98
  2    XCOST=CFK(1)*(ABS(ARATE)+CFK(2)*ABS(BRATE)+CFK(3)*ABS(AGOMGA(J)*
     X   COSB(J)*ARATE*BRATE))
C
C
C****   END OF COST FUNCTION
C
C
  98   IF(NFXPNT.EQ.0)GO TO 99
       IF(XCOST.GT.1.)XCOST=1.
       NCOST=XCOST*FLNPO
       IF(NCOST.GE.FS)NCOST=FS-1
  99   COST=XCOST
       RETURN
       END
```

Subroutine SINCC (Y, SINY)

Entry points. — SINCC, COSCC.

Called by. — COMP.

Function. — Fixed point sin/cos utility routine.

Comments. — Uses polynomial approximation

$$\text{siny} = (1 + a_{sc}y^2 + b_{sc}y^4 + c_{sc}y^6)y, \quad |y| \leq \frac{\pi}{2}$$

and the trig. identity, $\cos y = \sin(\pi/2 \pm y)$.

Scaling:

a_{sc}	=	-0.166656,	$-a_{sc}$	$\leq 2^{-2}$
b_{sc}	=	0.0083119,	b_{sc}	$\leq 2^{-6}$
c_{sc}	=	-0.000184882,	$-c_{sc}$	$\leq 2^{-12}$
argument Y	=	angle,	$\lvert Y \rvert$	$\leq 2^{2}$
argument SINY	=	sin(y) or cos(y),	$\lvert \text{SINY} \rvert$	$\leq 2^{0}$

```
$IBFTC S23     LIST
      SUBROUTINE SINCC(Y,SINY)
C
C SINE AND COSINE SUBROUTINE
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/SINCOS/ASC,B,C,HALFPI,PI,HAFPI
      INTEGER ASC,B,C,HALFPI,PI,HAFPI
      INTEGER X,Y,SINY,X2,X4,X6,Z
      X = Y
      GO TO 9
      ENTRY COSCC(Y,SINY)
      IF(Y.LT.0)X = Y+HAFPI
      IF(Y.GE.0)X = HAFPI -Y
    9 NEG = 0
      IF(X)1,2,2
    1 X=-X
      NEG=-1
    2 IF(X-PI)3,3,4
    3 IF(X.GT.HAFPI)X=PI-X
      GO TO 5
    4 NEG = NEG+1
      X=X-PI
    5 X2 =(X**2)/NM2
      X4 =(X2**2)/NM1
      X6 =(X4*X2)/NM1
      Z =((FS+(ASC*X2)/NP0 + (B*X4)/NP3 + (C*X6)/NP8)*X)/NM2
      IF(NEG)6,7,6
    6 SINY =-Z
      GO TO 8
    7 SINY = Z
    8 RETURN
      END
```

Subroutine CCATAN (ANGT, NUM, DEN)

Called by. –COMP.

Function. –Fixed point arc tangent utility routine.

Comments. –Uses polynomial approximation

$$\text{arc tan } x = \left\{[(dx^2 + c)x^2 + b]x^2 + a\right\} x, \quad |x| \leq 1$$

Scaling:

$a = 0.999215,$	$a \leq 2^0$
$b = -0.3211819,$	$-b \leq 2^{-1}$
$c = 0.1462766,$	$c \leq 2^{-2}$
$d = -0.0389929,$	$-d \leq 2^{-4}$
argument NUM,	$\lvert NUM \rvert \leq 2^0$
argument DEN,	$\lvert DEN \rvert \leq 2^0$
argument ANGT = arc tan $\left(\frac{NUM}{DEN}\right)$,	$\lvert ANGT \rvert \leq 2^2$

```
$IBFTC S24      LIST
      SUBROUTINE CCATAN(ANGT,NUM,DEN)
C
C ARC TANGENT SUBROUTINE
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/SINCOS/ASC,B,C,HALFPI,PI,HAFPI
      INTEGER ASC,B,C,HALFPI,PI,HAFPI
      COMMON/ATANC/AAT,BAT,CAT,DAT,QUARPI
      INTEGER AAT,BAT,CAT,DAT,QUARPI
      INTEGER ANGT,DEN,TAN,ATAN,TAN2
      IF(IABS(NUM)-IABS(DEN))45,46,47
   45 J = 1
      TAN =(NUM*NP0)/DEN
      GO TO 48
   47 J = 3
      TAN =(DEN*NP0)/NUM
      GO TO 48
   46 J = 2
      ATAN = QUARPI
      IF(NUM.LT.0)ATAN=-ATAN
      IF(DEN.LT.0)ATAN=-ATAN
      GO TO 49
   48 TAN2 =(TAN**2)/NP0
      ATAN = (((((((DAT*TAN2)/NP2 + CAT)*TAN2)/NP1 + BAT)*TAN2)/NP1 +
     CAAT)*TAN)/NP0
   49 IF(NUM)50,50,51
   50 IF(DEN)52,52,53
   52 IF(J-3)55,54,55
   54 ANGT= (-HALFPI/2 - ATAN/4)
      RETURN
   55 ANGT= (-PI + ATAN/4)
      RETURN
   53 IF(J-3)56,54,56
   56 ANGT= ATAN/4
```

```
      RETURN
   51 IF(DEN)57,57,58
   57 IF(J-3)59,60,59
   59 ANGT= (PI + ATAN/4)
      RETURN
   60 ANGT= (HALFPI/2 - ATAN/4)
      RETURN
   58 IF(J-3)56,60,56
      END
```

Subroutine CCASIN (ANG, SINANG)

Called by. –COMP.

Calls. –SQRTCC.

Function. –Fixed point arc sin utility routine.

Comments. –Uses approximation

$$\text{arc sin } x = \frac{\pi}{2} - \left\{ [(dx + c)x + b]x + a \right\} \sqrt{1-x}$$

for $0 \leq x \leq 1$

Scaling:

$a = 1.5707288,$	$a \leq 2^{1}$
$b = -0.2121144,$	$-b \leq 2^{-2}$
$c = 0.0742610,$	$c \leq 2^{-3}$
$d = -0.0187293,$	$-d \leq 2^{-5}$
argument ANG = arc sin (SINANG),	$\lvert ANG \rvert \leq 2^{1}$
argument SINANG,	$\lvert SINANG \rvert \leq 2^{0}$

```
$IBFTC S25      LIST
      SUBROUTINE CCASIN(ANG,SINANG)
C
C ARC SINE SUBROUTINE
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      COMMON/SINCOS/ASC,B,C,HALFPI,PI,HAFPI
      INTEGER ASC,B,C,HALFPI,PI,HAFPI
      COMMON/ASINC/AAS,BAS,CAS,DAS
      INTEGER AAS,BAS,CAS,DAS
      INTEGER ANG,SINANG,XAS
      INTEGER SQRTCC
      IF(SINANG)36,37,38
   36 XAS = -SINANG
      GO TO 39
   37 ANG = 0
      RETURN
   38 XAS = SINANG
   39 MIDTM1 = (((((DAS*XAS)/NP2+CAS)*XAS)/NP1+BAS)*XAS)/NP3+AAS
      ANG = HALFPI - (SQRTCC(FS-XAS) * MIDTM1)/NP0
      IF(SINANG.LT.0)ANG=-ANG
      RETURN
      END
```

Function SQRTCC(X)

Called by. –COMP, CCASIN.

Function. –Fixed point square root routine.

Comments. –Uses three iterations of

$$(\sqrt{X})_n = \frac{(\sqrt{X})}{2}_{n-1} + \frac{X}{2(\sqrt{X})_{n-1}}$$

Scaling. –

$$\left.\begin{array}{ll} \text{argument } X, & 0 \le X \le 2^k \\ \text{SQRTCC} = \sqrt{X}, & 0 \le \text{SQRTCC} \le 2^{\frac{k}{2}} \end{array}\right\} \text{ k even}$$

```
$IBFTC S26      LIST
      INTEGER FUNCTION SQRTCC(X)
C
C SQUARE ROOT FUNCTION
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      INTEGER X,Z,TRY
      IF(X.GT.0) GO TO 1
      SQRTCC=0
      RETURN
1     K=1
      Z=X
2     TRY=Z*4
      IF(TRY.GT.FS)GO TO 3
      Z=TRY
      K=2*K
      GO TO 2
3     TRY=HALFFS+Z/2
      DO 4 J=1,3
4     TRY=TRY/2+(Z*NM1)/TRY
      SQRTCC=TRY/K
      RETURN
      END
```

Function DMULT(X,Y)

Called by. –COMP

Function. –Returns double precision product of two double precision numbers.

Comments. –If x is double precision $\leq 2^{k_x}$ and y is double precision $\leq 2^{k_y}$

and, $X = 2^n A + B$

$Y = 2^n C + D$

where A,B,C,D are FORTRAN integers $\leq 2^n$,

then the desired product is $XY/2^{2n}$ for a product maximum value of $2^{k_x + k_y}$.

Note that $\dfrac{XY}{2^{2n}} = \dfrac{2^{2n} AC + 2^n (AD + BC) + BD}{2^{2n}}$

or as programmed $\dfrac{XY}{2^{2n}} = AC + \dfrac{(AD + BC)}{2^n} + \cancel{\dfrac{BD}{2^{2n}}}$

Summary. –

argument =	X, double precision,	$\lvert x \rvert$	$\leq 2^{k_x}$
argument =	Y, double precision,	$\lvert y \rvert$	$\leq 2^{k_y}$
DMULT =	XY, double precision,	$\lvert xy \rvert$	$\leq 2^{k_x + k_y}$

```
$IBFTC S27      LIST
      INTEGER FUNCTION DMULT(X,Y)
C
C DOUBLE PRECISION MULTIPLY FUNCTION
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      INTEGER X,Y,MSX,LSX,MSY,LSY
      IF(X.GT.DBLFS)X=DBLFS
      MSX=X/NPO
      LSX=X-MSX*NPO
      IF(Y.GT.DBLFS)Y=DBLFS
      MSY=Y/NPO
      LSY=Y-MSY*NPO
      DMULT=MSX*MSY+(MSX*LSY+LSX*MSY)/NPO
      RETURN
      END
```

Function QUOTNT (DBLNUM, SGLDEN)

Called by. –COMP.

Function. –Returns the double precision quotient of a double precision dividend and a single precision divisor.

Comments. –Assume x, the double precision numerator, is represented by the FORTRAN integer $X = 2^n A + B$, and y is the single precision denominator represented by Y. Let $Q = 2^n C + D$ be the answer.

Where A, B, C, D, Y are $\leq 2^n$

and $A \leq Y$

then $$\frac{X}{Y} = \frac{Q}{2^n} \text{ TRUNCATED } = C + \cancelto{0}{\frac{D}{2^n}} = C$$

since $$\frac{D}{2^n} < 1$$

It remains to find D. Note that

$$\frac{Q \cdot Y}{2^n} = CY + \frac{DY}{2^n} = X$$

or $$D = \frac{(X - CY)\, 2^n}{Y}$$

Summary. –Compute:

$$C = \frac{X}{Y}$$

and

$$D = \frac{(X - CY)\, 2^n}{Y}$$

then $$\text{QUOTNT} = 2^n C + D$$

where $$|x| \leq 2^{k_x}, \quad |y| \leq 2^{k_y}, \quad \left|\frac{x}{y}\right| \leq 2^{k_x - k_y}$$

```
$IBFTC S28      LIST
      INTEGER FUNCTION QUOTNT(DBLNUM,SGLDEN)
C
C DOUBLE PRECISION DIVIDED BY A SINGLE PRECISION NUMBER
C
C****   DBLNUM , 2**N  --- SGLDEN , 2**D  --- QUOTNT , 2**(N-D)
C
      COMMON/NPSCAL/NM7,NM6,NM5,NM4,NM3,NM2,NM1,NP0,NP1,NP2,NP3,NP4,NP5,
     CNP6,NP7,NP8,NP9
      COMMON/MISCEL/FS,DBLFS,N      ,NH,MDLAST,HALFFS
      INTEGER FS,DBLFS,HALFFS
      INTEGER DBLNUM,SGLDEN
C
      IF(IABS(DBLNUM).GT.DBLFS)DBLNUM=ISIGN(DBLFS,DBLNUM)
      MSIG=DBLNUM/SGLDEN
      LSIG=((DBLNUM-MSIG*SGLDEN)*NP0)/SGLDEN
      QUOTNT=MSIG*NP0+LSIG
      IF(IABS(MSIG).GT.FS)QUOTNT=ISIGN(DBLFS,MSIG)
      RETURN
      END
```

Subroutine AINVSE (A, AINV, DETA, X, Y, NSYM)

Entry points. –AINVSE, SOLVEX.

Called by. –EVDER.

Function. –Computes $\underline{X} = [A]^{-1} \underline{Y}$, order 3.

Comments. –Arguments.

A	=	3 x 3 matrix
AINV	=	$[A]^{-1}$
DETA	=	$\lvert A \rvert$
X	=	3 vector,
Y	=	3 vector
NSYM	=	1 if A symmetric

$\left.\begin{matrix} X \\ Y \end{matrix}\right\} \underline{X} = [A]^{-1} \underline{Y}$

```
$IBFTC S29      LIST
      SUBROUTINE AINVSE(A,AINV,DETA,X,Y,NSYM)
C
C     TO FIND  B=A**-1 AND DET(A), WHERE A AND B ARE 3X3 MATRICES,
C       CALL AINVSE(A,B,DETA,DUMMY,DUMMY,NSYM)
C
C     TO FIND  X=(A**-1)*Y, B=A**-1 AND DET(A),(X AND Y ARE 3-VECTORS),
C       CALL SOLVEX(A,B,DETA,X,Y,NSYM)
C
C     IF A IS SYMMETRIC,NSYM=1    OTHERWISE,NSYM=0
C
      DIMENSION A(3,3),AINV(3,3),X(3),Y(3)
      NOSOL=1
      GO TO 1
      ENTRY SOLVEX(A,AINV,DETA,X,Y,NSYM)
      NOSOL=0
1     DETA=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(3,1)*A(2,3)+A(1,3)*A(2,1)*
     1 A(3,2)-A(3,1)*A(2,2)*A(1,3)-A(1,1)*A(2,3)*A(3,2)-A(1,2)*A(2,1)*
     2 A(3,3)
      AINV(1,1)=(A(2,2)*A(3,3)-A(2,3)*A(3,2))/DETA
      AINV(2,2)=(A(1,1)*A(3,3)-A(1,3)*A(3,1))/DETA
      AINV(3,3)=(A(1,1)*A(2,2)-A(1,2)*A(2,1))/DETA
      AINV(1,2)=(A(1,3)*A(3,2)-A(1,2)*A(3,3))/DETA
      AINV(1,3)=(A(1,2)*A(2,3)-A(2,2)*A(1,3))/DETA
      AINV(2,3)=(A(1,3)*A(2,1)-A(1,1)*A(2,3))/DETA
      IF(NSYM.EQ.1)GO TO 3
      AINV(2,1)=(A(3,1)*A(2,3)-A(2,1)*A(3,3))/DETA
      AINV(3,1)=(A(2,1)*A(3,2)-A(2,2)*A(3,1))/DETA
      AINV(3,2)=(A(3,1)*A(1,2)-A(1,1)*A(3,2))/DETA
4     IF(NOSOL.NE.0)RETURN
      DO 2 J=1,3
2     X(J)=AINV(J,1)*Y(1)+AINV(J,2)*Y(2)+AINV(J,3)*Y(3)
      RETURN
3     AINV(2,1)=AINV(1,2)
      AINV(3,1)=AINV(1,3)
      AINV(3,2)=AINV(2,3)
      GO TO 4
      END
```

Subroutine RANO (SIG, BIAS, XK, R)

Called by. — ENERGY.

Function. — Produces normally distributed random numbers.

Comments. — This routine is written in MAP (IBSYS assembly language), and is included for completeness only. It will be necessary to replace this routine with a Langley Research Center utility routine.

Arguments:

SIG	=	Standard deviation of distribution
BIAS	=	Mean value of distribution
XK	=	Kicker, initially set equal to 377 777 777 777 octal
R	=	Generated random number

```
$IBMAP S30      FULIST,REF
       ENTRY    RANO
 RANO  SAVE     (1,2)
       CLA      3,4
       STA      PZTE
       CLA      4,4
       ALS      18
       STD      PZTE
       CLA      5,4
       STA      PZTE+1
       CLA      6,4
       STA      PZTE+2
       TSX      JRDY,4
 PZTE  PZE      0
       PZE      0
       STO      **
       RETURN   RANO
 CAML  PZE      0
       PZE      0
       PZE      0
       STO      STP
       SXD      STP-1,1
       TZE      2,4
       TMI      1,4
       ANA      STP-4
       LRS      1
       ADD      STP
       LRS      1
       ADD      STP-3
       LXA      STP-2,1
 JBM   STO      STP+1
       CLA      STP
       FDP      STP+1
       STQ      STP+2
       CLA      STP+2
       FAD      STP+1
       SUB      STP-4
       TIX      JBM,1,1
```

```
        LXD     STP-1,1
        TRA     2,4
        OCT     1000000000,100400000000,3,0
STP     OCT     0,0,0
PTG     TZE     1,4
        TMI     1,4
        LRS     27
        STQ     CAML
        ALS     19
        ORA     JAE
        FSB     JAE+1
        STO     CAML+1
        CLA     CAML
        LRS     8
        ORA     JAE+2
        FAD     JAE+3
        STO     CAML
        FSB     JAE+4
        FDH     CAML
        STQ     CAML
        FMP     CAML
        STO     CAML+2
        LDQ     JRDY-2
        FMP     CAML+2
        FAD     JRDY-3
        LRS     35
        FMP     CAML+2
        FAD     JAE+5
        LRS     35
        FMP     CAML
        FAD     CAML+1
        LRS     35
        FMP     JRDY-1
        TRA     2,4
JAE     OCT     210000000000,210401000000,200000000000
        OCT     200552023631,201552023631,202561251001
        OCT     200754213603,200462532521,200542710277
JRDY    SXD     DDCK,4
```

```
CLA     1,4
STA     RDCO
ARS     18
STA     RDCO+1
CLA     2,4
STA     *+3
STA     *+4
STA     *+4
LDQ     **
MPY     RDCO+4
STQ     **
CLA     **
ARS     8
ADD     RDCO+5
FAD     RDCO+6
STO     XYZ+3
LRS     35
FMP     XYZ-6
FAD     XYZ-5
SSM
FAD     XYZ-5
LRS     35
FMP     XYZ-4
TSX     PTG,4
HTR     *
LRS     35
FMP     XYZ-6
STO     DDCK-4
TSX     CAML+3,4
HTR     *
STO     XYZ+4
LRS     35
FMP     DDCK-5
STO     DDCK-3
CLA     XYZ-5
STO     DDCK-2
AXT     3,4
LDQ     XYZ,4
```

```
        FMP     DDCK-2,4
        FAD     DDCK-2
        STO     DDCK-2
        TIX     *-4,4,1
        CLA     XYZ
        STO     DDCK-1
        AXT     2,4
        LDQ     XYZ+3,4
        FMP     DDCK-3,4
        FAD     DDCK-1
        STO     DDCK-1
        TIX     *-4,4,1
        FDP     DDCK-2
        STQ     DDCK-2
        CLA     DDCK-2
        CHS
        FAD     XYZ+4
        STO     DDCK-1
        CLA     XYZ+3
        FSB     XYZ-4
        TPL     *+3
        CLS     DDCK-1
        TRA     *+2
        CLA     DDCK-1
        LRS     35
RDCD    FMP     **
        FAD     **
        LXD     DDCK,4
        TRA     3,4
        OCT     11060471625,200000000000,0
        DEC     -2.,1.,.5
        OCT     201556626307,176603476734,167526704623
XYZ     OCT     202501770730,200633037061,172522333054
        OCT     0,0,0,0,0,0
DDCK    OCT     0
        END
```

Subroutine ZERO

Function. — Stores data at program load time into common blocks /CONSTS/, /AGROUP/, /BUFFIN/.

Comments. — This is a block data subroutine. The data loaded into /CONSTS/ and /AGROUP/ should remain unchanged, but the data put into the input buffer should be altered to reflect that data most frequently used by LRC. This minimizes the number of data cards that must be prepared for a run, since these cards need contain only changed data.

```
$IBFTC ZERO    LIST
      BLOCK DATA
      COMMON/CONSTS/X(27)  /AGROUP/A(112)
      COMMON/BUFFIN/SVIC(79),U(83),AA(18),SRVO(26),STTRKR(31),BODYI(9),
     1 TQDIST(61),POSIC(6),TDATA(26),NOINT(124),DATACC(16),GAINS(65),
     2 OTHER(13),NCONT(9)
      DATA X /57.2957795E0,.174532925E-1,2.0925741E7,1.407654E16,
     1 3.14159265E0,1.2394434E-3,1.5362199E-6,1.9040576E-9,
     2 2.3599717E-12,2.9250513E-15,3.6254356E-18,4.4935222E-21,
     3 5.5694665E-24,6.9030385E-27,8.5559255E-30,2.59362716E4,
     4 .317297510E0,.402710840E0,.255558294E0,.324352042E0,.411664382E0,
     5 .261240167E0,.331563418E0,.420816989E0,.267048366E0,.338935126E0,
     6 .72722052E-4/, A /112*0./
      DATA SVIC/79*0./, U / 9*.01,9*.001,12*20.,6*.02,9*0.,2*.001,4*0.,
     1 4*.001,4*.005,4*.5,4*10.,4*.005,4*.05,4*.005,1.,.01,.1E-7,
     2 .15625E-2 /,AA / 4.1,3.8,.7,1.06,1.12,1.02,.837,.43,2*.005,
     3 3*1000.,3*0.,2*1000./, SRVO /2*.427,2*.00667,2*1840.,2*28.8652,
     4 2*.0182,2*6.667,2*21.,2*.16,2*1.53,2*.00305,2*.15,80.,1.E5,
     5 2*67.6/,STTRKR/  4*0.,                          .025,.016,1.1,168.,
     6 100.,5.3,.05,2.,.016,578.,4.,248.,20.,.754,1.,3.8,.8,3.8,5.32,
     7 9.33, 2*6.4E-4,.06,1.008,3*5.25E-3/, BODYI/.151E6,.15E6,.3E5,
     8 3*0.,2*1000.,500./,POSIC/0.,0.,.22141741E8,.2521402E5,0.,0./,
     9 TDATA/4*0.,420.5,17*.1,5.99,3*.1667/,NOINT/79*0,4,42*15,0,1/,DATA
     1 CC/0.,0.,.20925741E8,0.,1.,3*0.,1.,3*0.,1.,1.,.1138751E-2,1./,
     2 OTHER/ .5,.1,3*5.,2*.3,.5,2*10.,90.,78.,1./,NCONT/6,2,0,4,1,3,10,
     3 1,1/
      DATA TQDIST/61*0./
      DATA GAINS/ 5*.90484,2*4737.,3*2369.,2*4706.,3*2353.,2*941.1,
     1 3*470.6,2*4737.,3*2369.,2*4706.,3*2353.,2*941.1,3*470.6,30*0./
      END
```

Section 4

USERS GUIDE

Introduction

A familiarity with the system features and characteristics is necessary for using the CMG simulation. This guide is primarily a tabulation of input/output data and formats, containing explanations of options and references to other sections of this report.

Data input is by NAMELIST, where only data changes must be supplied; thus, the value that each data item will assume is given. These are not recommended values and are subject to change by LRC.

All data used by the control computer is subject to the maximum values specified in Appendix E.

Source decks for two subroutines should be made available to the user. These routines can be easily reprogrammed to suit individual requirements. They are:

a. MANUAL — Supplies desired commands to the system.

b. COST — The iterative control law's cost function.

Modification instructions for these routines are contained in the Program Description section.

Input Data

There are three input lists. These are to be specified in the following order:

$DATA

$COMAND

$COSTK

<u>$DATA</u> - This data supplies the main input buffer, /BUFFIN/, as follows.

Name	Symbol	Description	Value
NBIT	n	Control computer word length is sign + n bits, $7 \leq n \leq 17$	15
NFXPNT		Indicator; 0, floating point 1, fixed point Must correspond with COMP routine fixed or floating	0
PTARGT(3)	$\underline{R}_o$	Inertial coordinates of target position at t = 0. (Earth Mapping)	$\begin{pmatrix} 0. \\ 0. \\ 20\ 925\ 741. \end{pmatrix}$ ft.
AZCO	A_{c_o}	Telescope azimuth command at t = 0 (Earth Mapping)	0.°
CAPM1(3) CAPM2(3) CAPM3(3)	M_{1i} M_{2i} M_{3i}	Row 1 Row 2 Row 3 Direction cosines commanded body to inertial (Exp. 1)	$\begin{bmatrix} 1., 0., 0. \\ 0., 1., 0. \\ 0., 0., 1. \end{bmatrix}$
ETADOT	$\dot{\eta}$	z-axis rate command (Horizon Spectrometry)	1.°/sec.
WOAVE	ω_{av}	Average orbital rate (Horizon Spectrometry, Microwave Transmission)	0.0011387551 rod/sec
WBMAX	ω_{BMAX}	Maximum body rate (Maneuver Mode)	1.°/sec
NMOUNT		Star tracker mounting indicator 2, Inertial; 3, Horizon Spectrometry	3
NUPDAT	m_u	Attitude update command; 0, no update; 1, update from star trackers; 2, update from environment (Horizon Spectromety Spectrometry)	0
MODE	m_x	Mode selector 1, Earth Mapping; 2, Inertial; 3, Horizon Spectrometry; 4, Microwave Transmission; 5, Attitude Hold; 6, Maneuver; 7, Manual	6

Name	Symbol	Description	Value
ZO(5)	z_o, m_G	Five values of compensation z_o	5*.90484
KE1(5)	K_{e1}, m_G	Five sets of attitude error gains	2*4809., 3*2405.
KE2(5)	K_{e2}, m_G		2*4777., 3*2389.
KE3(5)	K_{e3}, m_G		2*955., 3*478.
KE4(5)	K_{e4}, m_G		2*4809., 3*2405.
KE5(5)	K_{e5}, m_G		2*4777., 3*2389.
KE6(5)	K_{e6}, m_G		2*955., 3*478.
KR1(5)	K_{r1}, m_G	Five sets of attitude error gains	5*0.
KR2(5)	K_{r2}, m_G		5*0.
KR3(5)	K_{r3}, m_G		5*0.
KR4(5)	K_{r4}, m_G		5*0.
KR5(5)	K_{r5}, m_G		5*0.
KR6(5)	K_{r6}, m_G		5*0.
NFBSEL	m_{FB}	Feedback selector 1, sensed rate; 2, derived rate	2
NGAIN	m_G	Gain selector, 1 through 5	4
KSAVE	K_s	Algebraic Control Law parameter (Volume I)	0.5
XKREM	K_x	Iterative Control Law parameter (Volume I)	0.1
ITER	i_{max}	Maximum number of iterations ICL	10
LAW	m_L	Control Law selector 1, Baseline C.L.; 2, Algebraic C.L., 3, Iterative C.L.	1
MODCOM	m_ω	Indicator, For control laws 2 and 3; 0, $\underline{T}_c = \underline{T}_d$; 1, $\underline{T}_c = \underline{T}_d - \underline{T}_b$ (Algebraic & Iterative)	1

Name	Symbol	Description	Value
BHOLD	β_{HOLD}	Value of $\lvert\beta\rvert$, for which iterative control law will not select $\dot{\beta}$ of same polarity as β	90°
BSELFD	β_{SELFD}	$\lvert\beta\rvert > \beta_{SELFD}$ initiates self desaturation in Control Laws 2 and 3 (Algebraic & Iterative)	78°
BDOTDS	$\dot{\beta}_{DS}$	Self desaturation rate	1.°/sec.
ERRLIM(3)	$\underline{\epsilon}_L$	Jet desaturation error limits	3*5.°
DTJET(3)	$\underline{\Delta t}_J$	Jet pulse width	$\begin{pmatrix} 0.3 \\ 0.3 \\ 0.5 \end{pmatrix}$ sec.
TCYCLE	Δt_f	Control computer comp period	0.1 sec.
NSLOW	n_{SLOW}	$= \dfrac{\Delta t_s}{\Delta t_f}$, an integer ≥ 1	1
NXTRAP		Indicator, 1, Extrapolate $\underline{\omega}$; 0, use late $\underline{\omega}$.	1
TRUN	t_{end}	Run duration	6. sec
TPRINT		Print out interval	0.1 sec
POS(3)	$\underline{\rho}_0$	Vehicle inertial position @ t = 0	$\begin{pmatrix} 0. \\ 0. \\ 22\,141\,741. \end{pmatrix}$ ft.
VEL(3)	$\underline{\dot{\rho}}_0$	Vehicle inertial velocity @ t = 0	$\begin{pmatrix} 22\,214.02 \\ 0. \\ 0. \end{pmatrix}$ ft./sec.
AA	A_a	CMG Inertias (ft.-lb.-sec.^{2})	4.1
BA	B_a		3.8
CA	C_a		0.7
AB	A_b		1.06
BB	B_b		1.12
CB	C_b		1.02
AG	A_g		0.837
BG	B_g		0.43
JMBETA	J_M		0.005
JMALPH	J_M		0.005

Name	Symbol	Description	Value
AGOMGO(3)	$H_i(o)$	CMG spin momenta @ t = 0	3*1000. ft.-lb.-sec.
AGOMGD(3)	$\dot{H}_i$	CMG spin momenta rate of change	3*0. ft.-lb.
AGOMGM	H_{max}	Maximum value of H_i	1000. ft.-lb.-sec.
HNOM	H_{nom}	Nominal value of H_i	1000. ft.-lb.-sec.
IXX	I_{xx}	Vehicle Inertias (ft.-lb.-sec.^{2})	151 000.
IYY	I_{yy}		150 000.
IZZ	I_{zz}		30 000.
IXY	I_{xy}		0.
IYZ	I_{yz}		0.
IXZ	I_{xz}		0.
JETTQE(3)		Reaction jet torques	2*1000., 500. ft.-lb.
FLOWRT(3)		Reaction jet fuel rates	3*0.1667 lbs/sec.
KSF(2)	K_{SFi}	CMG Gimbal Servo parameters subscript 1 - β 2 - α See Appendix C.	2*28.8652 v/rad/sec
K(2)	K_i		2*1840.
K2(2)	$K2_i$		2*0.47 v/rad/sec
KT(2)	K_{Ti}		2*0.16 ft.-lb/amp
KB(2)	K_{Bi}		2*1.53 v/rad/sec
TAU(2)	τ_i		2*6.667 sec
TAUM(2)	τ_{Mi}		2*0.00305 sec
TAUN(2)	τ_{Ni}		2*0.0182 sec
TAUD(2)	τ_{Di}		2*0.00667 sec
ELIM(2)	e_{Li}		2*21. volts
TF(2)	T_{ci}/G_i		2*0.15 ft-lb
GRATIO(2)	G_i		2*67.6
LIMIT(2)	β_{LIM}, α_{LIM}		80.°, 10^5.°
MAXRTC(2)	$\dot{\beta}_{cmax}$, $\dot{\alpha}_{cmax}$		2* 10. °/sec.
ESTAR	E_1, E_2	Inertial Location of Stars 1 and 2	0., 0. °
ASTAR	A_1, A_2		0., 0. °

Name	Symbol	Description	Value
TAUWNG	τ_{WNG}	Star Tracker parameters, Appendix D.	0.025 sec
TAU1	τ_1		0.016 sec
TAU2N	τ_{2N}		1.1 sec
TAU2D	τ_{2D}		168. sec
TAU3N	τ_{3N}		100. sec
TAU3D	τ_{3D}		5.3 sec
TAU3NP	τ'_{3N}		0.05 sec
TAU3DP	τ'_{3D}		2. sec
TAU5	τ_5		0.016 sec
STG2	K_2		4. sec
STG3	K_3		248. sec
STG5	K_5		3.8 sec
STKT	K_T		578. v/rad
STKM	K_M		0.754 in.-oz./v
STKV	K_V		1. v/rad/sec
STKR	K_r		0.8 v/rad/sec
STAL	A_L		20. volts
STTF	T_{cst}		3.8 in.oz.
JINRTE	J_e		5.32 in.oz-sec^2
JINRTA	J_a		9.33 in.oz-sec^2
PSD		Noise characteristics	420.5 (mv)^2/cps
STBIAS(4)			4*0. volts
ETAP	η_p	Horizon Sensor parameters, Appendix D	0.00064 rad
ETAR	η_r		0.00064 rad
GHRZ	G_H		1.008
TAUHRZ	τ_H		0.06 sec
DELW(3)	ΔW	Rate sensor bias	3*0.00525 rad/hr.
MDNOM(3)	A_{xo}, A_{yo}, A_{zo}	Disturbance torque parameters, Eqns. (A-12) of the RFP Baseline	3*0. ft.-lb.
MDX(6)	A_{xi}		6*0. ft.-lb.
MDY(6)	A_{yi}		6*0. ft.-lb.
MDZ(6)	A_{zi}		6*0. ft.-lb.
PHASEX(6)	B_{xi}		6*0. deg.
PHASEY(6)	B_{yi}		6*0. deg.
PHASEZ(6)	B_{zi}		6*0. deg.
FREQ(6)	ω_i		6*0. rad/sec.

Name	Symbol	Description	Value
QMASS	Q	Moving Mass parameters, Eqns. (A-5) through (A-9) of RFP Baseline	0. slugs
PMO(3)	x_o, y_o, z_o		3*0. ft.
VMO	$\dot{x}_o, \dot{y}_o, \dot{z}_o$		3*0. ft/sec.
AMO(3)	$\ddot{x}_o, \ddot{y}_o, \ddot{z}_o$		3*0. ft/sec^2
R(3)	R_1, R_2, R_3		3*0. ft.
OMGAMM(3)	$\Omega_1, \Omega_2, \Omega_3$		3*0. rad/sec.
DTEST		Step size estimate for integration routine	0.0015625 sec.
DTMIN		Minimum allowable integration step size	10^{-6} sec.
EMAX		Integrator error bound multipliers; if all integrators errors $<$EMIN*U_i; step size double command. if any integrators error $\geq$EMAX*U_i; halve step size	1.
EMIN			0.01
NDOUBL		Number of double commands before step size is doubled	4
OMGAB0(3)	$\underline{\omega}_0$	Vehicle body rates	3*0. °/sec
BETAD0(3)	$\dot{\underline{\beta}}_0$	Inner gimbal rates	3*0. °/sec
ALPHD0(3)	$\dot{\underline{\alpha}}_0$	Outer gimbal rates	3*0. °/sec
BETA0(3)	$\underline{\beta}_0$	Inner gimbal angles	3*0. °
ALPHA0(3)	$\underline{\alpha}_0$	Outer gimbal angles	3*0. °
PHI0	ϕ_0	Euler angles	0. °
THETA0	θ_0		0. °
PSI0	ψ_0		
TLDB0(3)	$T_{\beta i_0}$	CMG gimbal servo variables	3*0. ft-lb
TLDA0(3)	$T_{\alpha i_0}$		3*0. ft-lb
EMB0(3)	$e'_{m\beta i_0}$		3*0. volts
EMA0(3)	$e'_{m\alpha i_0}$		3*0. volts
EFB0(3)	$e'_{f\beta i_0}$		3*0. volts
EFA0(3)	$e'_{f\alpha i_0}$		3*0. volts

State Vector Initial Conditions

Name	Symbol	Description	Value
HEXT0(3)	$\underline{H}_{ex_0}$	Momentum due to external torques	3*0. ft-lb-sec
ENRGB0(3)		Mechanical output energy of CMG gimbals	3*0. lb-ft
ENRGA0(3)			3*0. lb-ft
HORIZ0(2)	P_o, R_o	Horizon sensor outputs	2*0. °
NOISE0(4)	nvi_0	Star tracker variables	4*0. volts
STV10(4)	λ_{1i_0}		4*0. volts
STV20(4)	λ_{2i_0}		4*0. volts
STV30(4)	λ_{3i_0}		4*0. volts
STV3P0(4)	λ'_{3i_0}		4*0. volts
STV50(4)	λ_{5i_0}		4*0. volts
SQTI0(4)	λ_{6i_0}		4*0. °/sec
QTB0(4)	e'_{1_0}, e'_{2_0}, a'_{1_0}, a'_{2_0}		4*0. °
U(79)		Integration error bounds. One to one correspondence with state vector. See state vector initial conditions for units. Note: if $U_i = 0$, then the ith integrator is not considered for step size adjustment.	9*0.01 9*0.001 12*20. 6*0.02 9*0. 2*0.01 4*0. 4*0.001 4*0.005 4*0.5 4*10 4*0.005 4*0.05 4*0.005
NOINT(79)		Integration control, normally NOINT(I) = 0. If NOINT(I) = 1 the ith state variable is frozen at its initial value.	79*0.

Name	Symbol	Description	Value
NBOUT(16)		Number of bits carried in the output conversion of the 16 control computer output variables. See Table 4-2.	16*15
NBIN(25)		Number of bits carried in the input conversion of the first 25 control computer input variables. See Table 4-1.	25*15
TEVENT(15)		The time during a control computer comp cycle when each of the first 15 input variables are sampled. $0 \leq$ TEVENT(I) $\leq$ TCYCLE Note: Ideal case (no delay) when TEVENT(I) = TCYCLE Worst case (on period delay) when TEVENT(I) = 0. See Table 4-1.	15*0.1

Table 4-1. Control Computer Input Variables

Array Order	Variable	Source		
1	ω'_x	Rate Gyros	TEVENT (15)	NBIT (25)
2	ω'_y			
3	ω'_z			
4	e'_1	Star Trackers		
5	e'_2			
6	a'_1			
7	a'_2			
8	P'	Horizon Sensors		
9	R'			
10	β_1	CMG Gimbals		
11	β_2			
12	β_3			
13	α_1			
14	α_2			
15	α_3			
16	$\dot{\phi}_c$	Control Panel or Stick		
17	$\dot{\theta}_c$			
18	$\dot{\psi}_c$			
19	ϕ_c			
20	θ_c			
21	ψ_c			
22	t_m			
23	H_1	CMG Spin Tachs		
24	H_2			
25	H_3			

Table 4-2. Control Computer Output Variables

NBIT (16)	Variables	Use
1	$\dot{\beta}_{1c}$	CMG Commands
2	$\dot{\beta}_{2c}$	
3	$\dot{\beta}_{3c}$	
4	$\dot{\alpha}_{1c}$	
5	$\dot{\alpha}_{2c}$	
6	$\dot{\alpha}_{3c}$	
7	A_c	Experiment one telescope command (Earth Mapping)
8	ϕ'	Control Panel Display
9	θ'	
10	ψ'	
11	ϵ_x	
12	ϵ_y	
13	ϵ_z	
14	ω'_{xc}	
15	ω'_{yc}	
16	ω'_{zc}	

$COMAND - This NAMELIST supplies data to subroutine MANUAL.

Name	Value	
PHIC(4)	0., 0., 0. 1, 0.	deg. or deg./sec.
THETAC(4)	0., 0.1, 0.1, 0.	deg. or deg./sec.
PSIC(4)	0.1, 0.1, 0.1, 0.	deg. or deg./sec.
TMANUV(4)	0., 0., 0., 0.	sec.
TSTART(4)	-1., 6., 12., 18.	sec.

This data can be used in Modes 6 or 7. Otherwise it is ignored. Whenever t exceeds TSTART(I) a command and insert are initiated. If in Mode 6;

$$\begin{pmatrix} \phi_c \\ \theta_c \\ \psi_c \end{pmatrix} = \begin{pmatrix} \text{PHIC(I)} \\ \text{THETAC(I)} \\ \text{PSIC(I)} \end{pmatrix}$$

and t_{man} = TMANUV(I)

If in Mode 7;

$$\begin{pmatrix} \dot{\phi}_c \\ \dot{\theta}_c \\ \dot{\psi}_c \end{pmatrix} = \begin{pmatrix} \text{PHIC(I)} \\ \text{THETAC(I)} \\ \text{PSIC(I)} \end{pmatrix}$$

$COSTK - This NAMELIST supplies data to subroutine COST, the cost function used with the iterative control law. The data supplied is

CFK(3) 0.016, 1., 1.

and is used in this function

$$\text{COST=CFK(1)*} \left\{ |\dot{\alpha}| + \text{CFK (2)*}|\dot{\beta}| + \text{CFK (3)*}|H\dot{\alpha}\dot{\beta}\cos\beta| \right\}$$

Output Data

A block of output data is printed at the specified print intervals. The following is a model format showing the labels as printed and an X for data item.

T = X ,W = X ,E = X ,HTEST = X X X, TGAIN = X X , ITER = X

BDC	X	X	X	,ADC	X	X	X	,WERR	X	X	X
BD	X	X	X	,AD	X	X	X	,W	X	X	X
B	X	X	X	,A	X	X	X	,WOB	X	X	X
ATDC	X	X	X	,ATC	X	X	X	,EPOB	X	X	X
ATD	X	X	X	,AT	X	X	X	,ATOB	X	X	X
TC	X	X	X	,TCP	X	X	X	,TGIM	X	X	X
TJET	X	X	X	,TEX	X	X	X	,TCTL	X	X	X
P	X	X	X	,V	X	X	X	,WD	X	X	X
OBPE	X	X	X	,OBVE	X	X	X	,HRZ ERR, P = X , R = X			

STAR TRACKER ERR, DELE1 = X, DELE2=X, DELA1=X, DELA2=X, TEL AZ COM=X

Where

T	= time (sec.)
W	= weight of reaction jet fuel consumed (pounds)
E	= actual output power of CMG torquers (watts)
HTEST	= components of the angular momentum test (ft. -lb. -sec.)
TGAIN	= The projections of $\underline{TGIM}$ on and normal to $\underline{TCP}$, normalized by $\lvert \underline{TCP} \rvert$.
ITER	= the number of iterations taken in the last execution of the iterative control law
BDC	$= \dot{\beta}_{ci}$ (°/sec.)
ADC	$= \dot{\alpha}_{ci}$ (°/sec.)
WERR	= components of $\underline{\omega}_c - \underline{\omega}$ (°/sec.)
BD	$= \dot{\beta}_i$ (°/sec.)
AD	$= \dot{\alpha}_i$ (°/sec.)
W	$= \underline{\omega}$ (°/sec.)
ATDC	$= \dot{\phi}_{c'}, \dot{\theta}_{c'}, \dot{\psi}_{c'}$ (°/sec.)
ATC	$= \phi_{c'}, \theta_{c'}, \psi_{c'}$ (°)
EPOB	$= \underline{\epsilon}'$ (°)
ATD	$= \dot{\phi}, \dot{\theta}, \dot{\psi}$ (°/sec.)
AT	$= \phi, \theta, \psi$ (°)

See figure 3-11

ATOB	$= \phi', \theta', \psi'$ (°)
TC	$= \underline{T}_d$ (ft. -lb.)
TCP	$= \underline{T}_c$ (ft. -lb.)
TGIM	= theoretical torque produced as a result of $\dot{\alpha}_{ic}$, $\dot{\beta}_{ic}$ only (ft. -lb.)
TJET	= reaction jet torques (ft. -lb.)
TEX	= disturbance torques, external plus moving mass (ft. -lb.)
TCTL	= theoretical torque produced as a result of $\dot{\alpha}_{ic}$, $\dot{\beta}_{ic}$, $\underline{\omega}$.. (ft. -lb.)
P	$= \underline{\rho}$ (ft.)
V	$= \dot{\underline{\rho}}$ (ft./sec.)
WD	$= \dot{\underline{\omega}}$ (°/sec.^{2})
OBPE	$= \underline{\rho}' - \underline{\rho}$ (ft.)
OBVE	$= \dot{\underline{\rho}}' - \dot{\underline{\rho}}$ (ft./sec.)
P	$= P_a$ (°)
R	$= R_a$ (°)
DELE1	$= e_1 - e'_1$ (°)
DELE2	$= e_2 - e'_2$ (°)
DELA 1	$= a_1 - a'_1$ (°)
DELA 2	$= a_2 - a'_2$ (°)
TELAZCOM	$= A_c$

At the end of each run, a summary is printed out. The data items are identified on the output as:

Peak torquer output power (watts)

Average torquer output power (watts)

Theoretical average torquer output power (watts)

Jet Fuel (pounds)

Number of integration steps taken and the values of first 300 of these (sec.).

Appendix A

RIGID BODY EQUATIONS OF MOTION

The equations in this appendix yield the accelerations $\dot{\omega}_i$, $\ddot{\beta}_i$, $\ddot{\alpha}_i$. Many are cyclic and are programmed in DO loops. A group of three equations is cyclic when the second (or third) equation is produced by cycling all subscripts of the first (or second) equation in the following manner:

$x \rightarrow y$

$y \rightarrow z$

$z \rightarrow x$

$1 \rightarrow 2$

$2 \rightarrow 3$

$3 \rightarrow 1$

All a_{ij} not listed are equal to zero. $(i = 1, 9; j = 1, 9)$

(A-1) → (A-2) → (A-3)

$$\frac{d}{dt}\beta_1 = \dot{\beta}_1 \tag{A-1}$$

$$\frac{d}{dt}\beta_2 = \dot{\beta}_2 \tag{A-2}$$

$$\frac{d}{dt}\beta_3 = \dot{\beta}_3 \tag{A-3}$$

(A-4) → (A-5) → (A-6)

$$\frac{d}{dt}\alpha_1 = \dot{\alpha}_1 \tag{A-4}$$

$$\frac{d}{dt}\alpha_2 = \dot{\alpha}_2 \tag{A-5}$$

$$\frac{d}{dt}\alpha_3 = \dot{\alpha}_3 \tag{A-6}$$

(A-7) → (A-8) → (A-9)

$$\omega'_{x1} = \omega_X \cos\alpha_1 - \omega_Z \sin\alpha_1 \tag{A-7}$$

$$\omega'_{y2} = \omega_Y \cos\alpha_2 - \omega_X \sin\alpha_2 \tag{A-8}$$

$$\omega'_{z3} = \omega_Z \cos\alpha_3 - \omega_Y \sin\alpha_3 \tag{A-9}$$

(A-10) → (A-11) → (A-12)

$$\omega'_{y1} = \omega_Y + \dot{\alpha}_1 \tag{A-10}$$

$$\omega'_{z2} = \omega_Z + \dot{\alpha}_2 \tag{A-11}$$

$$\omega'_{x3} = \omega_X + \dot{\alpha}_3 \tag{A-12}$$

(A-13) → (A-14) → (A-15)

$$\omega'_{z1} = \omega_X \sin\alpha_1 + \omega_Z \cos\alpha_1 \tag{A-13}$$

$$\omega'_{x2} = \omega_Y \sin\alpha_2 + \omega_X \cos\alpha_2 \tag{A-14}$$

$$\omega'_{y3} = \omega_Z \sin\alpha_3 + \omega_Y \cos\alpha_3 \tag{A-15}$$

(A-16) → (A-17) → (A-18)

$$\omega_{x1} = \omega'_{x1} \cos\beta_1 + \omega'_{y1} \sin\beta_1 \tag{A-16}$$

$$\omega_{y2} = \omega'_{y2} \cos\beta_2 + \omega'_{z2} \sin\beta_2 \tag{A-17}$$

$$\omega_{z3} = \omega'_{z3} \cos\beta_3 + \omega'_{x3} \sin\beta_3 \tag{A-18}$$

(A-19) → (A-20) → (A-21)

$$\omega_{y1} = \omega'_{y1} \cos\beta_1 - \omega'_{x1} \sin\beta_1 \tag{A-19}$$

$$\omega_{z2} = \omega'_{z2} \cos\beta_2 - \omega'_{y2} \sin\beta_2 \tag{A-20}$$

$$\omega_{x3} = \omega'_{x3} \cos\beta_3 - \omega'_{z3} \sin\beta_3 \tag{A-21}$$

(A-22) → (A-23) → (A-24)

$$\omega_{z1} = \omega'_{z1} + \dot{\beta}_1 \tag{A-22}$$

$$\omega_{x2} = \omega'_{x2} + \dot{\beta}_2 \tag{A-23}$$

$$\omega_{y3} = \omega'_{y3} + \dot{\beta}_3 \tag{A-24}$$

$$\left.\begin{aligned} a_{44} &= C_b + B_g + J_{m\beta} G_\beta^2 \\ a_{66} &= C_b + B_g + J_{m\beta} G_\beta^2 \\ a_{88} &= C_b + B_g + J_{m\beta} G_\beta^2 \end{aligned}\right\}$$

Listed for completeness only. These are constant and are computed at initialization only.

(A-25) → (A-26) → (A-27)

$$\begin{aligned} a_{11} = {} & I_{xx} + [A_a + (A_b+A_g)\cos^2\beta_1 + (B_b+B_g)\sin^2\beta_1]\cos^2\alpha_1 \\ & + (C_a + C_b + B_g + J_{m\beta})(\sin^2\alpha_1 + \cos^2\alpha_2) \\ & + [A_a + (A_b+A_g)\cos^2\beta_2 + (B_b+B_g)\sin^2\beta_2]\sin^2\alpha_2 \\ & + J_{m\alpha} + B_a + (A_b+A_g)\sin^2\beta_3 + (B_b+B_g)\cos^2\beta_3 \end{aligned} \tag{A-25}$$

$$\begin{aligned} a_{22} = {} & I_{yy} + B_a + J_{m\alpha} + (A_b+A_g)\sin^2\beta_1 + (B_b+B_g)\cos^2\beta_1 \\ & + (C_a + C_b + B_g + J_{m\beta})(\sin^2\alpha_2 + \cos^2\alpha_3) \\ & + [A_a + (A_b+A_g)\cos^2\beta_2 + (B_b+B_g)\sin^2\beta_2]\cos^2\alpha_2 \\ & + [A_a + (A_b+A_g)\cos^2\beta_3 + (B_b+B_g)\sin^2\beta_3]\sin^2\alpha_3 \end{aligned} \tag{A-26}$$

$$\begin{aligned} a_{33} = {} & I_{zz} + B_a + J_{m\alpha} + (C_a + C_b + B_g + J_{m\beta})(\cos^2\alpha_1 + \sin^2\alpha_3) \\ & + (A_b+A_g)\sin^2\beta_2 + (B_b+B_g)\cos^2\beta_2 \\ & + [A_a + (A_b+A_g)\cos^2\beta_1 + (B_b+B_g)\sin^2\beta_1]\sin^2\alpha_1 \\ & + [A_a + (A_b+A_g)\cos^2\beta_3 + (B_b+B_g)\sin^2\beta_3]\cos^2\alpha_3 \end{aligned} \tag{A-27}$$

(A-28) → (A-29) → (A-30)

$$a_{12} = -I_{xy} - (A_b + A_g - B_b - B_g)(\sin\alpha_3 \sin\beta_3 \cos\beta_3 - \cos\alpha_1 \sin\beta_1 \cos\beta_1)$$

$$+ [C_a - A_a + C_b + B_g + J_{m\beta} - (A_b + A_g)\cos^2\beta_2 - (B_b + B_g)\sin^2\beta_2] \quad \text{(A-28)}$$

$$\sin\alpha_2 \cos\alpha_2$$

$$a_{23} = -I_{yz} - (A_b + A_g - B_b - B_g)(\sin\alpha_1 \sin\beta_1 \cos\beta_1 - \cos\alpha_2 \sin\beta_2 \cos\beta_2)$$

$$+ [C_a - A_a + C_b + B_g + J_{m\beta} - (A_b + A_g)\cos^2\beta_3 - (B_b + B_g)\sin^2\beta_3] \quad \text{(A-29)}$$

$$\sin\alpha_3 \cos\alpha_3$$

$$a_{31} = -I_{zx} - (A_b + A_g - B_b - B_g)(\sin\alpha_2 \sin\beta_2 \cos\beta_2 - \cos\alpha_3 \sin\beta_3 \cos\beta_3)$$

$$+ [C_a - A_a + C_b + B_g + J_{m\beta} - (A_b + A_g)\cos^2\beta_1 - (B_b + B_g)\sin^2\beta_1] \quad \text{(A-30)}$$

$$\sin\alpha_1 \cos\alpha_1$$

(A-31) → (A-32) → (A-33)

$$a_{21} = a_{12} \quad \text{(A-31)}$$

$$a_{32} = a_{23} \quad \text{(A-32)}$$

$$a_{13} = a_{31} \quad \text{(A-33)}$$

$$a_{14} = (C_b + B_g + J_{m\beta} G_\beta) \sin\alpha_1 \quad \text{(A-34)}$$

$$a_{15} = (A_b + A_g - B_b - B_g) \cos\alpha_1 \sin\beta_1 \cos\beta_1 \quad \text{(A-35)}$$

$$a_{16} = (C_b + B_g + J_{m\beta} G_\beta) \cos\alpha_2 \quad \text{(A-36)}$$

$$a_{17} = -(A_b + A_g - B_b - B_g) \sin\alpha_2 \sin\beta_2 \cos\beta_2 \quad \text{(A-37)}$$

$$a_{19} = B_a + J_{m\alpha} G_\alpha + (A_b + A_g)\sin^2\beta_3 + (B_b + B_g)\cos^2\beta_3 \quad \text{(A-38)}$$

$$a_{25} = B_a + J_{m\alpha}G_\alpha + (A_b+A_g)\sin^2\beta_1 + (B_b+B_g)\cos^2\beta_1 \tag{A-39}$$

$$a_{26} = (C_b + B_g + J_{m\beta}G_\beta)\sin\alpha_2 \tag{A-40}$$

$$a_{27} = (A_b + A_g - B_b - B_g)\cos\alpha_2 \sin\beta_2 \cos\beta_2 \tag{A-41}$$

$$a_{28} = (C_b + B_g + J_{m\beta}G_\beta)\cos\alpha_3 \tag{A-42}$$

$$a_{29} = -(A_b + A_g - B_b - B_g)\sin\alpha_3 \sin\beta_3 \cos\beta_3 \tag{A-43}$$

$$a_{34} = (C_b + B_g + J_{m\beta}G_\beta)\cos\alpha_1 \tag{A-44}$$

$$a_{35} = -(A_b + A_g - B_b - B_g)\sin\alpha_1 \sin\beta_1 \cos\beta_1 \tag{A-45}$$

$$a_{37} = B_a + J_{m\alpha}G_\alpha + (A_b+A_g)\sin^2\beta_2 + (B_b+B_g)\cos^2\beta_2 \tag{A-46}$$

$$a_{38} = (C_b + B_g + J_{m\beta}G_\beta)\sin\alpha_3 \tag{A-47}$$

$$a_{39} = (A_b + A_g - B_b - B_g)\cos\alpha_3 \sin\beta_3 \cos\beta_3 \tag{A-48}$$

$$a_{55} = B_a + J_{m\alpha}G_\alpha^2 + (A_b+A_g)\sin^2\beta_1 + (B_b+B_g)\cos^2\beta_1 \tag{A-49}$$

$$a_{77} = B_a + J_{m\alpha}G_\alpha^2 + (A_b+A_g)\sin^2\beta_2 + (B_b+B_g)\cos^2\beta_2 \tag{A-50}$$

$$a_{99} = B_a + J_{m\alpha}G_\alpha^2 + (A_b+A_g)\sin^2\beta_3 + (B_b+B_g)\cos^2\beta_3 \tag{A-51}$$

(A-52) → (A-53) → (A-54)

$$\begin{aligned}\Delta b_1 = -\Big\{ & -(\omega_y + G_\alpha\dot{\alpha}_1)\omega'_{z1}J_{m\alpha}\cos\alpha_1 + (\omega'_{z1} + G_\beta\dot{\beta}_1)\omega'_{y1}J_{m\beta}\cos\alpha_1 \\ & + [\dot{\alpha}_1 J_{m\beta} + (\omega_y + G_\alpha\dot{\alpha}_1)J_{m\alpha}]\omega'_{x1}\sin\alpha_1 + (\omega_z + G_\alpha\dot{\alpha}_2)\omega'_{x2}J_{m\alpha}\sin\alpha_2 \\ & + [\dot{\alpha}_2 J_{m\beta} + (\omega_z + G_\alpha\dot{\alpha}_2)J_{m\alpha}]\omega'_{y2}\cos\alpha_2 - (\omega'_{x2} + G_\beta\dot{\beta}_2)\omega'_{z2}J_{m\beta}\sin\alpha_2 \\ & - (\omega'_{y3} + G_\beta\dot{\beta}_3)\omega'_{z3}J_{m\beta}\Big\}\end{aligned} \tag{A-52}$$

$$
\begin{aligned}
\Delta b_2 = - \Big\{ & - (\omega'_{z1} + G_\beta \dot{\beta}_1)\, \omega'_{x1} J_{m\beta} + [\dot{\alpha}_2 J_{m\beta} + (\omega_z + G_\alpha \dot{\alpha}_2)\, J_{m\alpha}]\, \omega'_{y2} \sin\alpha_2 \\
& + (\omega'_{x2} + G_\beta \dot{\beta}_2)\, \omega'_{z2} J_{m\beta} \cos\alpha_2 - (\omega_z + G_\alpha \dot{\alpha}_2)\, \omega'_{x2} J_{m\alpha} \cos\alpha_2 \\
& + (\omega_x + G_\alpha \dot{\alpha}_3)\, \omega'_{y3} J_{m\alpha} \sin\alpha_3 - (\omega'_{y3} + G_\beta \dot{\beta}_3)\, \omega'_{x3} J_{m\beta} \sin\alpha_3 \\
& + [\dot{\alpha}_3 J_{m\beta} + (\omega_x + G_\alpha \dot{\alpha}_3)\, J_{m\alpha}]\, \omega'_{z3} \cos\alpha_3 \Big\}
\end{aligned}
\tag{A-53}
$$

$$
\begin{aligned}
\Delta b_3 = - \Big\{ & (\omega_y + G_\alpha \dot{\alpha}_1)\, \omega'_{z1} J_{m\alpha} \sin\alpha_1 - (\omega'_{z1} + G_\beta \dot{\beta}_1)\, \omega'_{y1} J_{m\beta} \sin\alpha_1 \\
& + [\dot{\alpha}_1 J_{m\beta} + (\omega_y + G_\alpha \dot{\alpha}_1)\, J_{m\alpha}]\, \omega'_{x1} \cos\alpha_1 - (\omega'_{x2} + G_\beta \dot{\beta}_2)\, \omega'_{y2} J_{m\beta} \\
& - (\omega_x + G_\alpha \dot{\alpha}_3)\, \omega'_{y3} J_{m\alpha} \cos\alpha_3 + (\omega'_{y3} + G_\beta \dot{\beta}_3)\, \omega'_{x3} J_{m\beta} \cos\alpha_3 \\
& + [\dot{\alpha}_3 J_{m\beta} + (\omega_x + G_\alpha \dot{\alpha}_3)\, J_{m\alpha}]\, \omega'_{z3} \sin\alpha_3 \Big\}
\end{aligned}
\tag{A-54}
$$

Equation (A-55)

$$
\begin{aligned}
b_1 = \Bigg\{ & M_x + \Delta b_1 + \dot{I}_{xy}\,\omega_y - \dot{I}_{xx}\,\omega_x + \dot{I}_{xz}\,\omega_z - Q\,[\omega_y (x\dot{y} - y\dot{x}) + \omega_z (x\dot{z} - z\dot{x}) + (y\dot{z} - z\dot{y})] \\
& - [-A_a \cos\alpha_1\, \omega'_{z1}\, \dot{\alpha}_1 + C_a \sin\alpha_1\, \omega'_{x1} \dot{\alpha}_1 - (A_b + A_g) \cos\alpha_1 \cos^2\beta_1\, \omega'_{z1} \dot{\alpha}_1 + (A_b + A_g) \cos\alpha_1 \cos\beta_1\, \omega_{y1} \dot{\beta}_1 \\
& - (B_b + B_g) \cos\alpha_1 \sin^2\beta_1\, \omega'_{z1} \dot{\alpha}_1 + \cos\alpha_1 \sin\beta_1 (B_b + B_g)\, \omega_{x1} \dot{\beta}_1 + (C_b + B_g) \sin\alpha_1\, \omega'_{x1} \dot{\alpha}_1 \\
& + C_a \cos\alpha_2 \omega'_{y2} \dot{\alpha}_2 + A_a \sin\alpha_2 \omega'_{x2} \dot{\alpha}_2 + (C_b + B_g) \cos\alpha_2 \omega'_{y2} \dot{\alpha}_2 + (A_b + A_g) \sin\alpha_2 \cos^2\beta_2\, \omega'_{x2} \dot{\alpha}_2 \\
& - (A_b + A_g) \sin\alpha_2 \cos\beta_2\, \omega_{z2} \dot{\beta}_2 + (B_b + B_g) \sin\alpha_2 \sin^2\beta_2\, \omega'_{x2} \dot{\alpha}_2 - (B_b + B_g) \sin\alpha_2 \sin\beta_2\, \omega_{y2} \dot{\beta}_2 \\
& + (B_b + B_g) \sin\beta_3 \cos\beta_3\, \omega'_{y3} \dot{\alpha}_3 - (B_b + B_g) \cos\beta_3\, \omega_{z3} \dot{\beta}_3 - (A_b + A_g) \sin\beta_3 \cos\beta_3\, \omega'_{y3} \dot{\alpha}_3 \\
& + (A_b + A_g) \sin\beta_3\, \omega_{x3} \dot{\beta}_3] - \dot{H}_1 \cos\alpha_1 \cos\beta_1 + \dot{H}_2 \sin\alpha_2 \cos\beta_2 - \dot{H}_3 \sin\beta_3 \\
& - \Big\{ (\omega_z^2 - \omega_y^2)\, I_{yz} + \omega_y \omega_z (I_{zz} - I_{yy}) + \omega_x (\omega_z I_{xy} - \omega_y I_{zx}) + \omega'_{y1} \omega'_{z1} (C_a - B_a) \cos\alpha_1 + \omega'_{x1} \omega'_{y1} (B_a - A_a) \sin\alpha_1 \\
& + \omega_{y1} \omega_{z1} (C_b - B_b) \cos\alpha_1 \cos\beta_1 - [\omega_{x1} \omega_{z1} (A_b + A_g - C_b - B_g) + \Omega_1 \omega_{z1} A_g] \cos\alpha_1 \sin\beta_1 \\
& + [\omega_{x1} \omega_{y1} (B_b + B_g - A_b - A_g) - \Omega_1 \omega_{y1} A_g] \sin\alpha_1 + \omega'_{y2} \omega'_{z2} (B_a - A_a) \cos\alpha_2 \\
& - \omega'_{x2} \omega'_{z2} (C_a - B_a) \sin\alpha_2 + [\omega_{y2} \omega_{z2} (B_b + B_g - A_b - A_g) - \Omega_2 \omega_{z2} A_g] \cos\alpha_2 \\
& - \omega_{x2} \omega_{z2} (C_b - B_b) \sin\alpha_2 \cos\beta_2 + [\omega_{x2} \omega_{y2} (A_b + A_g - C_b - B_g) + \Omega_2 \omega_{x2} A_g] \sin\alpha_2 \sin\beta_2 + \omega'_{y3} \omega'_{z3} (A_a - C_a) \\
& + [\omega_{y3} \omega_{z3} (A_b + A_g - C_b - B_g) + \Omega_3 \omega_{y3} A_g] \cos\beta_3 + \omega_{x3} \omega_{y3} (C_b - B_b) \sin\beta_3 \Big\}
\end{aligned}
$$

Equation (A-56)
Cycles from (A-55)

$$
\begin{aligned}
b_2 = \Bigg\{ & M_y + \Delta b_2 - \dot{I}_{yy}\omega_y + \dot{I}_{yz}\omega_z + \dot{I}_{xy}\omega_x - Q\,[\omega_x(y\dot{x} - x\dot{y}) + \omega_z(y\dot{z} - z\dot{y}) + (z\dot{x} - x\dot{z})] \\
& - [-(A_b + A_g)\sin\beta_1\cos\beta_1\omega'_{z1}\dot{\alpha}_1 + (A_b + A_g)\sin\beta_1\omega_{y1}\dot{\beta}_1 + (B_b + B_g)\sin\beta_1\cos\beta_1\dot{\omega}'_{z1}\dot{\alpha}_1 \\
& \quad - (B_b + B_g)\cos\beta_1\omega_{x1}\dot{\beta}_1 + C_a\sin\alpha_2\omega'_{y2}\dot{\alpha}_2 - A_a\cos\alpha_2\omega'_{x2}\dot{\alpha}_2 + (C_b + B_g)\sin\alpha_2\omega'_{y2}\dot{\alpha}_2 \\
& \quad - (A_b + A_g)\cos\alpha_2\cos^2\beta_2\omega'_{x2}\dot{\alpha}_2 + (A_b + A_g)\cos\alpha_2\cos\beta_2\omega_{z2}\dot{\beta}_2 - (B_b + B_g)\cos\alpha_2\sin^2\beta_2\omega'_{x2}\dot{\alpha}_2 \\
& \quad + (B_b + B_g)\cos\alpha_2\sin\beta_2\omega_{y2}\dot{\beta}_2 + C_a\cos\alpha_3\omega'_{z3}\dot{\alpha}_3 + A_a\sin\alpha_3\omega'_{y3}\dot{\alpha}_3 \\
& \quad + (B_b + B_g)\sin\alpha_3\sin^2\beta_3\omega'_{y3}\dot{\alpha}_3 - (B_b + B_g)\sin\alpha_3\sin\beta_3\omega_{z3}\dot{\beta}_3 + (C_b + B_g)\cos\alpha_3\omega'_{z3}\dot{\alpha}_3 \\
& \quad + (A_b + A_g)\sin\alpha_3\cos^2\beta_3\omega'_{y3}\dot{\alpha}_3 - (A_b + A_g)\sin\alpha_3\cos\beta_3\omega_{x3}\dot{\beta}_3] - \dot{H}_1\sin\beta_1 - \dot{H}_2\cos\alpha_2\cos\beta_2 + \dot{H}_3\sin\alpha_3\cos\beta_3 \\
& - \Big\{ \omega_x\omega_z(I_{xx} - I_{zz}) + (\omega_x^2 - \omega_z^2)I_{zx} + \omega_y(\omega_x I_{yz} - \omega_z I_{xy}) + \omega'_{x1}\omega'_{z1}(A_a - C_a) \\
& \quad + \omega_{y1}\omega_{z1}(C_b - B_b)\sin\beta_1 + [\omega_{x1}\omega_{z1}(A_b + A_g - C_b - B_g) + \Omega_1\omega_{z1}A_g]\cos\beta_1 \\
& \quad + \omega'_{y2}\omega'_{z2}(B_a - A_a)\sin\alpha_2 + \omega'_{x2}\omega'_{z2}(C_a - B_a)\cos\alpha_2 + [\omega_{y2}\omega_{z2}(B_b + B_g - A_b - A_g) - \Omega_2\omega_{z2}A_g]\sin\alpha_2 \\
& \quad + \omega_{x2}\omega_{z2}(C_b - B_b)\cos\alpha_2\cos\beta_2 - [\omega_{x2}\omega_{y2}(A_b + A_g - C_b - B_g) + \Omega_2\omega_{x2}A_g]\cos\alpha_2\sin\beta_2 \\
& \quad + \omega'_{x3}\omega'_{z3}(B_a - A_a)\cos\alpha_3 - \omega'_{x3}\omega'_{y3}(C_a - B_a)\sin\alpha_3 + [\omega_{y3}\omega_{z3}(A_b + A_g - C_b - B_g) + \Omega_3\omega_{y3}A_g](\sin\alpha_3\sin\beta_3) \\
& \quad + [\omega_{x3}\omega_{z3}(B_b + B_g - A_b - A_g) - \Omega_3\omega_{x3}A_g]\cos\alpha_3 - \omega_{x3}\omega_{y3}(C_b - B_b)\sin\alpha_3\cos\beta_3 \Big\}
\end{aligned}
$$

Equation (A-57)
Cycles from (A-56)

$$
\begin{aligned}
b_3 = \Bigg\{ & M_z + \Delta b_3 + \dot{I}_{xz}\,\omega_x + \dot{I}_{yz}\,\omega_y - \dot{I}_{zz}\,\omega_z - Q\,[\,\omega_x\,(z\dot{x} - x\dot{z}) + \omega_y\,(z\dot{y} - y\dot{z}) + (x\dot{y} - y\dot{x})\,] \\
& - [\,A_a \sin\alpha_1\,\omega'_{z1}\,\dot{\alpha}_1 + C_a \cos\alpha\,\omega'_{x1}\,\dot{\alpha}_1 + (A_b + A_g)\sin\alpha\cos^2\beta_1\,\omega'_{z1}\,\dot{\alpha}_1 - (A_b + A_g)\sin\alpha_1\cos\beta_1\,\omega_{y1}\,\dot{\beta}_1 \\
& \quad + (B_b + B_g)\sin\alpha_1\sin^2\beta_1\,\omega'_{z1}\,\dot{\alpha}_1 - (B_b + B_g)\sin\alpha_1\sin\beta_1\,\omega_{x1}\,\dot{\beta}_1 + (C_b + B_g)\cos\alpha_1\,\omega'_{x1}\,\dot{\alpha}_1 - (A_b + A_g)\sin\beta_2\cos\beta_2\,\omega'_{x2}\,\dot{\alpha}_2 \\
& \quad + (A_b + A_g)\sin\beta_2\,\omega_{z2}\,\dot{\beta}_2 + (B_b + B_g)\sin\beta_2\cos\beta_2\,\omega'_{x2}\,\dot{\alpha}_2 - (B_b + B_g)\cos\beta_2\,\omega_{y2}\,\dot{\beta}_2 + C_a \sin\alpha_3\,\omega'_{z3}\,\dot{\alpha}_3 \\
& \quad - A_a \cos\alpha_3\,\omega'_{y3}\,\dot{\alpha}_3 - (B_b + B_g)\cos\alpha_3\sin^2\beta_3\,\omega'_{y3}\,\dot{\alpha}_3 + (B_b + B_g)\cos\alpha_3\sin\beta_3\,\omega_{z3}\,\dot{\beta}_3 \\
& \quad + (C_b + B_g)\sin\alpha_3\,\omega'_{z3}\,\dot{\alpha}_3 - (A_b + A_g)\cos\alpha_3\cos^2\beta_3\,\omega'_{y3}\,\dot{\alpha}_3 + (A_b + A_g)\cos\alpha_3\cos\beta_3\,\omega_{x3}\,\dot{\beta}_3\,] + \dot{H}_1\sin\alpha_1\cos\beta_1 - \dot{H}_2\sin\beta_2 - \dot{H}_3\cos\alpha_3\cos\beta_3 \\
& - \Big\{ \omega_x\,\omega_y\,(I_{yy} - I_{xx}) + (\omega_y^2 - \omega_x^2)\,I_{xy} + \omega_z\,(\omega_y\,I_{zx} - \omega_x\,I_{yz}) - \omega'_{y1}\,\omega'_{z1}\,(C_a - B_a)\sin\alpha_1 \\
& \quad + \omega'_{x1}\,\omega'_{y1}\,(B_a - A_a)\cos\alpha_1 - \omega_{y1}\,\omega_{z1}\,(C_b - B_b)\sin\alpha_1\cos\beta_1 + [\omega_{x1}\,\omega_{z1}\,(A_b + A_g - C_b - B_g) + \Omega_1\,\omega_{z1}\,A_g\,]\sin\alpha_1\sin\beta_1 \\
& \quad + [\,\omega_{x1}\,\omega_{y1}\,(B_b + B_g - A_b - A_g) - \Omega_1\,\omega_{y1}\,A_g]\cos\alpha_1 + \omega'_{x2}\,\omega'_{y2}\,(A_a - C_a) + \omega_{x2}\,\omega_{z2}\,(C_b - B_b)\sin\beta_2 \\
& \quad + [\omega_{x2}\,\omega_{y2}\,(A_b + A_g - C_b - B_g) + \Omega_2\,\omega_{x2}\,A_g]\cos\beta_2 + \omega'_{x3}\,\omega'_{z3}\,(B_a - A_a)\sin\alpha_3 \\
& \quad + \omega'_{x3}\,\omega'_{y3}\,(C_a - B_a)\cos\alpha_3 - [\omega_{y3}\,\omega_{z3}\,(A_b + A_g - C_b - B_g) + \Omega_3\,\omega_{y3}\,A_g]\cos\alpha_3\sin\beta_3 \\
& \quad + [\omega_{x3}\,\omega_{z3}\,(B_b + B_g - A_b - A_g) - \Omega_3\,\omega_{x3}\,A_g]\sin\alpha_3 + \omega_{x3}\,\omega_{y3}\,(C_b - B_b)\cos\alpha_3\cos\beta_3 \Big\}
\end{aligned}
$$

(A-58) → (A-59) → (A-60)

$$M_{\alpha 1} = T_{\alpha 1} - T_{c\alpha} \operatorname{sgn}(\dot{\alpha}_1) \tag{A-58}$$

$$M_{\alpha 2} = T_{\alpha 2} - T_{c\alpha} \operatorname{sgn}(\dot{\alpha}_2) \tag{A-59}$$

$$M_{\alpha 3} = T_{\alpha 3} - T_{c\alpha} \operatorname{sgn}(\dot{\alpha}_3) \tag{A-60}$$

(A-61) → (A-62) → (A-63)

$$M_{\beta 1} = T_{\beta 1} - T_{c\beta} \operatorname{sgn}(\dot{\beta}_1) \tag{A-61}$$

$$M_{\beta 2} = T_{\beta 2} - T_{c\beta} \operatorname{sgn}(\dot{\beta}_2) \tag{A-62}$$

$$M_{\beta 3} = T_{\beta 3} - T_{c\beta} \operatorname{sgn}(\dot{\beta}_3) \tag{A-63}$$

(A-64) → (A-65) → (A-66)

Right hand side cycles
LHS increments by 2

$$b_4 = M_{\beta 1} - (C_b + B_g + J_{m\beta} G_\beta)\, \omega'_{x1} \dot{\alpha}_1 + \omega_{y1} \Omega_1 A_g + (A_b + A_g - B_b - B_g)\, \omega_{x1} \omega_{y1} \tag{A-64}$$

$$b_6 = M_{\beta 2} - (C_b + B_g + J_{m\beta} G_\beta)\, \omega'_{y2} \dot{\alpha}_2 + \omega_{z2} \Omega_2 A_g + (A_b + A_g - B_b - B_g)\, \omega_{y2} \omega_{z2} \tag{A-65}$$

$$b_8 = M_{\beta 3} - (C_b + B_g + J_{m\beta} G_\beta)\, \omega'_{z3} \dot{\alpha}_3 + \omega_{x3} \Omega_3 A_g + (A_b + A_g - B_b - B_g)\, \omega_{x3} \omega_{z3} \tag{A-66}$$

(A-67) → (A-68) → (A-69)

R.H.S. cycles, L.H.S. increments by 2

$$b_5 = \left\{ \begin{array}{l} M_{\alpha 1} + (\omega'_{z1} + G_\beta \dot{\beta}_1)\, \omega'_{x1} J_{m\beta} - \dot{H}_1 \sin\beta_1 \\ + (A_b + A_g)\, [\sin\beta_1 \cos\beta_1 \omega'_{z1} \dot{\alpha}_1 + \sin^2\beta_1\, \omega'_{x1} \dot{\beta}_1 - \sin\beta_1 \cos\beta_1 (\omega_y + \dot{\alpha}_1)\, \dot{\beta}_1] \\ + (B_b + B_g) \cos\beta_1 [- \sin\beta_1 \omega'_{z1} \dot{\alpha}_1 + \cos\beta_1\, \omega'_{x1} \dot{\beta}_1 + \sin\beta_1 (\omega_y + \dot{\alpha}_1)\, \dot{\beta}_1] \\ - \omega_{z1} \Omega_1 A_g \cos\beta_1 + (C_a - A_a)\, \omega'_{x1} \omega'_{z1} \\ + (B_b - C_b)\, \omega_{y1} \omega_{z1} \sin\beta_1 - (A_b + A_g - C_b - B_g) \cos\beta_1 \omega_{x1} \omega_{z1} \end{array} \right. \tag{A-67}$$

$$
b_7 = \left\{ \begin{aligned}
& M_{\alpha 2} + (\omega'_{x2} + G_\beta \dot{\beta}_2)\, \omega'_{y2} J_{m\beta} - \dot{H}_2 \sin\beta_2 \\
& - (A_b + A_g) \sin\beta_2 [-\cos\beta_2 \omega'_{x2} \dot{\alpha}_2 - \sin\beta_2 \omega'_{y2} \dot{\beta}_2 + (\omega_z + \dot{\alpha}_2)\, \dot{\beta}_2 \cos\beta_2] \\
& - (B_b + B_g) \cos\beta_2 [\sin\beta_2 \omega'_{x2} \dot{\alpha}_2 - \cos\beta_2 \omega'_{y2} \dot{\beta}_2 - (\omega_z + \dot{\alpha}_2)\, \dot{\beta}_2 \sin\beta_2] \\
& - \omega_{x2} \Omega_2 A_g \cos\beta_2 - (A_a - C_a)\, \omega'_{x2} \omega'_{y2} \\
& + (B_b - C_b)\, \omega_{x2} \omega_{z2} \sin\beta_2 - (A_b + A_g - C_b - B_g)\, \omega_{x2} \omega_{y2} \cos\beta_2
\end{aligned} \right. \tag{A-68}
$$

$$
b_9 = \left\{ \begin{aligned}
& M_{\alpha 3} + (\omega'_{y3} + G_\beta \dot{\beta}_2)\, \omega'_{z3} J_{m\beta} - \dot{H}_3 \sin\beta_3 \\
& - (B_b + B_g) \cos\beta_3 [\sin\beta_3 \omega'_{y3} \dot{\alpha}_3 - \cos\beta_3 \omega'_{z3} \dot{\beta}_3 - (\omega_x + \dot{\alpha}_3)\, \dot{\beta}_3 \sin\beta_3] \\
& + (A_b + A_g) \sin\beta_3 [\cos\beta_3 \omega'_{y3} \dot{\alpha}_3 + \sin\beta_3 \omega'_{z3} \dot{\beta}_3 - (\omega_x + \dot{\alpha}_3)\, \dot{\beta}_3 \cos\beta_3] \\
& - \omega_{y3} \Omega_3 A_g \cos\beta_3 - (A_a - C_a)\, \omega'_{y3} \omega'_{z3} \\
& + (B_b - C_b)\, \omega_{x3} \omega_{y3} \sin\beta_3 - (A_b + A_g - C_b - B_g)\, \omega_{y3} \omega_{z3} \cos\beta_3
\end{aligned} \right. \tag{A-69}
$$

$$
a'_{11} = \left(a_{11} - \frac{a_{14}^2}{a_{44}} - \frac{a_{15}^2}{a_{55}} - \frac{a_{16}^2}{a_{66}} - \frac{a_{17}^2}{a_{77}} - \frac{a_{19}^2}{a_{99}} \right) \tag{A-70}
$$

$$
a'_{12} = \left(a_{12} - \frac{a_{15} a_{25}}{a_{55}} - \frac{a_{16} a_{26}}{a_{66}} - \frac{a_{17} a_{27}}{a_{77}} - \frac{a_{19} a_{29}}{a_{99}} \right) \tag{A-71}
$$

$$
a'_{13} = \left(a_{13} - \frac{a_{14} a_{34}}{a_{44}} - \frac{a_{15} a_{35}}{a_{55}} - \frac{a_{17} a_{37}}{a_{77}} - \frac{a_{19} a_{39}}{a_{99}} \right) \tag{A-72}
$$

$$a'_{22} = \left(a_{22} - \frac{a_{25}^2}{a_{55}} - \frac{a_{26}^2}{a_{66}} - \frac{a_{27}^2}{a_{77}} - \frac{a_{28}^2}{a_{88}} - \frac{a_{29}^2}{a_{99}}\right) \tag{A-73}$$

$$a'_{23} = \left(a_{23} - \frac{a_{25}a_{35}}{a_{55}} - \frac{a_{27}a_{37}}{a_{77}} - \frac{a_{28}a_{38}}{a_{88}} - \frac{a_{29}a_{39}}{a_{99}}\right) \tag{A-74}$$

$$a'_{33} = \left(a_{33} - \frac{a_{34}^2}{a_{44}} - \frac{a_{35}^2}{a_{55}} - \frac{a_{37}^2}{a_{77}} - \frac{a_{38}^2}{a_{88}} - \frac{a_{39}^2}{a_{99}}\right) \tag{A-75}$$

$$b'_1 = b_1 - \frac{a_{14}}{a_{44}} b_4 - \frac{a_{15}}{a_{55}} b_5 - \frac{a_{16}}{a_{66}} b_6 - \frac{a_{17}}{a_{77}} b_7 - \frac{a_{19}}{a_{99}} b_9 \tag{A-76}$$

$$b'_2 = b_2 - \frac{a_{25}}{a_{55}} b_5 - \frac{a_{26}}{a_{66}} b_6 - \frac{a_{27}}{a_{77}} b_7 - \frac{a_{28}}{a_{88}} b_8 - \frac{a_{29}}{a_{99}} b_9 \tag{A-77}$$

$$b'_3 = b_3 - \frac{a_{34}}{a_{44}} b_4 - \frac{a_{35}}{a_{55}} b_5 - \frac{a_{37}}{a_{77}} b_7 - \frac{a_{38}}{a_{88}} b_8 - \frac{a_{39}}{a_{99}} b_9 \tag{A-78}$$

Perform the indicated matrix inversion and multiplication to compute the body accelerations. acclerations.

$$\begin{bmatrix} \dot{\omega}_X \\ \dot{\omega}_Y \\ \dot{\omega}_Z \end{bmatrix} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{12} & a'_{22} & a'_{23} \\ a'_{13} & a'_{23} & a'_{33} \end{bmatrix}^{-1} \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \end{bmatrix} \tag{A-79}$$

$$\ddot{\beta}_1 = (b_4 - a_{14}\dot{\omega}_x - a_{34}\dot{\omega}_z)/a_{44} \tag{A-80}$$

$$\ddot{\alpha}_1 = (b_5 - a_{15}\dot{\omega}_x - a_{25}\dot{\omega}_y - a_{35}\dot{\omega}_z)/a_{55} \tag{A-81}$$

$$\ddot{\beta}_2 = (b_6 - a_{16}\dot{\omega}_x - a_{26}\dot{\omega}_y)/a_{66} \tag{A-82}$$

$$\ddot{\alpha}_2 = (b_7 - a_{17}\dot{\omega}_x - a_{27}\dot{\omega}_y - a_{37}\dot{\omega}_z)/a_{77} \tag{A-83}$$

$$\ddot{\beta}_3 = (b_8 - a_{28}\dot{\omega}_y - a_{38}\dot{\omega}_z)/a_{88} \tag{A-84}$$

$$\ddot{\alpha}_3 = (b_9 - a_{19}\dot{\omega}_x - a_{29}\dot{\omega}_y - a_{39}\dot{\omega}_z)/a_{99} \tag{A-85}$$

Appendix B

VEHICLE ATTITUDE EQUATIONS

The Euler angles that transform the inertial frame $\begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix}$ to the vehicle frame $\begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix}$ are:

1. ψ about z_I
2. θ about y_I' (new y_I)
3. ϕ about x_B.

The Euler rates can be expressed as

$$\dot{\phi} = \omega_x + [\omega_z \cos\phi + \omega_y \sin\phi] \tan\theta \tag{B-1}$$

$$\dot{\theta} = \omega_y \cos\phi - \omega_z \sin\phi \tag{B-2}$$

$$\dot{\psi} = [\omega_z \cos\phi + \omega_y \sin\phi]/\cos\theta \tag{B-3}$$

The elements of the direction cosine matrix,

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$, (body to inertial) are:

$$\left.\begin{aligned}
m_{11} &= \cos\psi \cos\theta \\
m_{12} &= \cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi \\
m_{13} &= \cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi \\
m_{21} &= \sin\psi \cos\theta \\
m_{22} &= \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi \\
m_{23} &= \sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi \\
m_{31} &= -\sin\theta \\
m_{32} &= \cos\theta \sin\phi \\
m_{33} &= \cos\theta \cos\phi
\end{aligned}\right\} \tag{B-4}$$

Appendix C

CMG GIMBAL EQUATIONS

Figure C-1 is a diagram of the number one CMG's outer gimbal servo loop. Inspection of this diagram leads to equations (C-1) through (C-5):

$$\dot{e}'_{f\alpha i} = \frac{1}{\tau_{D\alpha}} \left(G_\alpha K_{2\alpha} \dot{\alpha}_i - e'_{f\alpha i}\right) \tag{C-1}$$

$$\dot{e}'_{M\alpha i} = \frac{1}{\tau_\alpha} \left[K_\alpha \left(K_{SF\alpha} \dot{\alpha}_{ic} - \tau_{N\alpha} \dot{e}'_{f\alpha i} - e'_{f\alpha i}\right) - e'_{M\alpha i}\right] \tag{C-2}$$

$$e_{M\alpha i} = \begin{cases} e'_{M\alpha i}, & |e'_{M\alpha i}| \le e_{L\alpha} \\ e_{L\alpha}, & e_{M\alpha i} > e_{L\alpha} \\ -e_{L\alpha}, & e_{M\alpha i} < -e_{L\alpha} \end{cases} \tag{C-3}$$

$$\dot{T}_{\alpha i} = \frac{1}{\tau_{M\alpha}} \left[\frac{G_\alpha K_{T\alpha}}{R_{M\alpha}} \left(e_{M\alpha i} - G_\alpha K_{B\alpha} \dot{\alpha}_i\right) - T_{\alpha i}\right] \tag{C-4}$$

$$M_{\alpha i} = T_{\alpha i} - T_{f\alpha i} \tag{C-5}$$

For inner loop equations, replace $\dot{\alpha}_{ic}$, $\dot{\alpha}$ with $\dot{\beta}_{ic}$, $\dot{\beta}$ and replace the subscripts (α) with subscripts (β).

To compute $T_{f\alpha i}$ for equation (C-5) use the logic of figure 2-4 where:

- J is replaced with $a_{2i+3,\ 2i+3}$
- T_c is replaced with $T_{c\alpha}$
- ω is replaced with $\dot{\alpha}_i$
- T_a is replaced with $T_{\alpha i} - H_i \dot{\beta}_i \cos \beta_i$

For the corresponding $T_{f\beta i}$

J is replaced with $a_{2i+2,\ 2i+2}$

T_c is replaced with $T_{c\beta}$

ω is replaced with $\dot{\beta}_i$

T_a is replaced with $T_{\beta i} + H_i \dot{\alpha}_i \cos \beta_i$

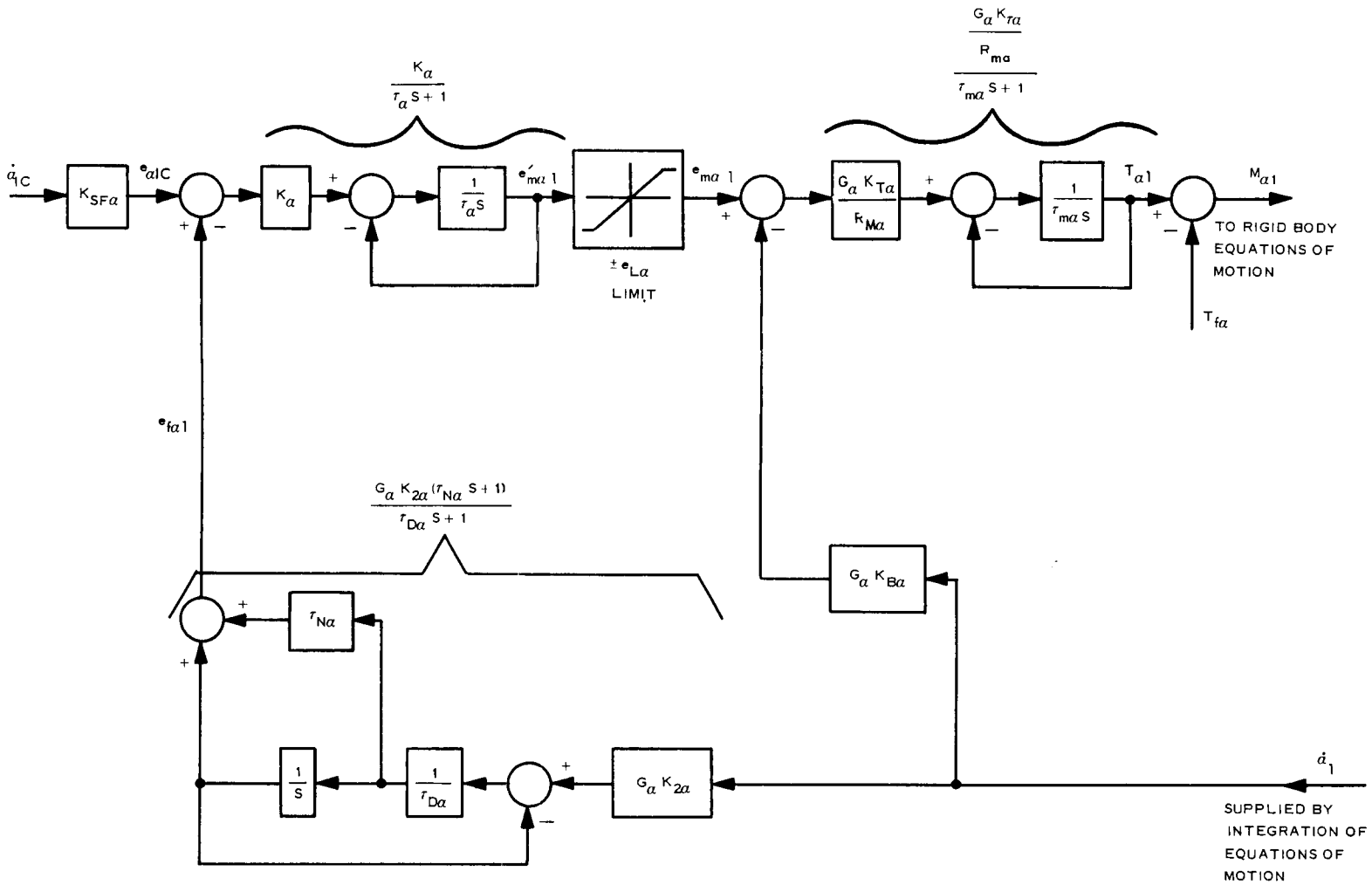


Figure C-1. Gimbal Servo Loop

Appendix D

SENSOR EQUATIONS

Star Trackers

This section lists the equations describing a star tracker servo loop conforming to the model supplied in the CMG RFP L-7035.

Equations (D-1) through (D-4) are dependent on sensor mounting and are presented in pairs. The suffix (a) denotes experiment two configuration (Inertial Mode) and suffix (b) represents the experiment three mounting scheme (Horizon Spectrometry).

$$\underline{u}_a = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{D-1a}$$

$$\underline{u}_a = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{D-1b}$$

$$\underline{u}_e = \begin{pmatrix} 0 \\ \cos a' \\ \sin a' \end{pmatrix} \tag{D-2a}$$

$$\underline{u}_e = \begin{pmatrix} -\cos a' \\ -\sin a' \\ 0 \end{pmatrix} \tag{D-2b}$$

Note that in the case of $\underline{u}_{e2}$ of Equation (D-2a) the gimbal angle a_2' is approximated with the boresight angle a_2, since the azimuth channel of the #2 star tracker is not simulated, as specified in the RFP.

The boresight angles are:

$$a_i = \tan^{-1}\left(\frac{-s_{iy}}{s_{iz}}\right) \tag{D-3a}$$

$$a_i = \tan^{-1}\left(\frac{-s_{ix}}{s_{iy}}\right) \tag{D-3b}$$

$$e_i = \sin^{-1}(s_{ix}) \tag{D-4a}$$

$$e_i = -\sin^{-1}(s_{iz}) \tag{D-4b}$$

Identification of:

$$\epsilon_a = (a - a') \cos e$$

$$\epsilon_e = e - e'$$

$$G_n(s) = \frac{1}{\tau_{WNG} s + 1}$$

$$G_1(s) = \frac{1}{\tau_1 s + 1}$$

$$G_2(s) = K_2 \frac{(\tau_{2N} s + 1)}{(\tau_{2D} s + 1)}$$

$$G_3(s) = K_3 \frac{(\tau_{3N} s + 1)(\tau'_{3N} s + 1)}{(\tau_{3D} s + 1)(\tau'_{3D} s + 1)}$$

$$G_4(s) = 1$$

$$G_5(s) = \frac{K_5}{\tau_5 s + 1}$$

in conjunction with the block diagram from the RFP leads to Figure D-1. The following equations can be written from inspection of this diagram

$$\dot{n}_{Va} = \frac{1}{\tau_{WNG}} (N_{Va} - n_{Va}) \tag{D-5}$$

$$\dot{\lambda}_{1a} = \frac{1}{\tau_1} (K_T \epsilon_a - \lambda_{1a}) \tag{D-6}$$

$$\dot{\lambda}_{2a} = \frac{1}{\tau_{2D}} [K_2 (\lambda_{1a} - n_{Va}) - \lambda_{2a}] \tag{D-7}$$

$$\dot{\lambda}_{3a} = \frac{1}{\tau_{3D}} [K_3 (\lambda_{2a} + \tau_{2N} \dot{\lambda}_{2a} - \lambda_{5a}) - \lambda_{3a}] \tag{D-8}$$

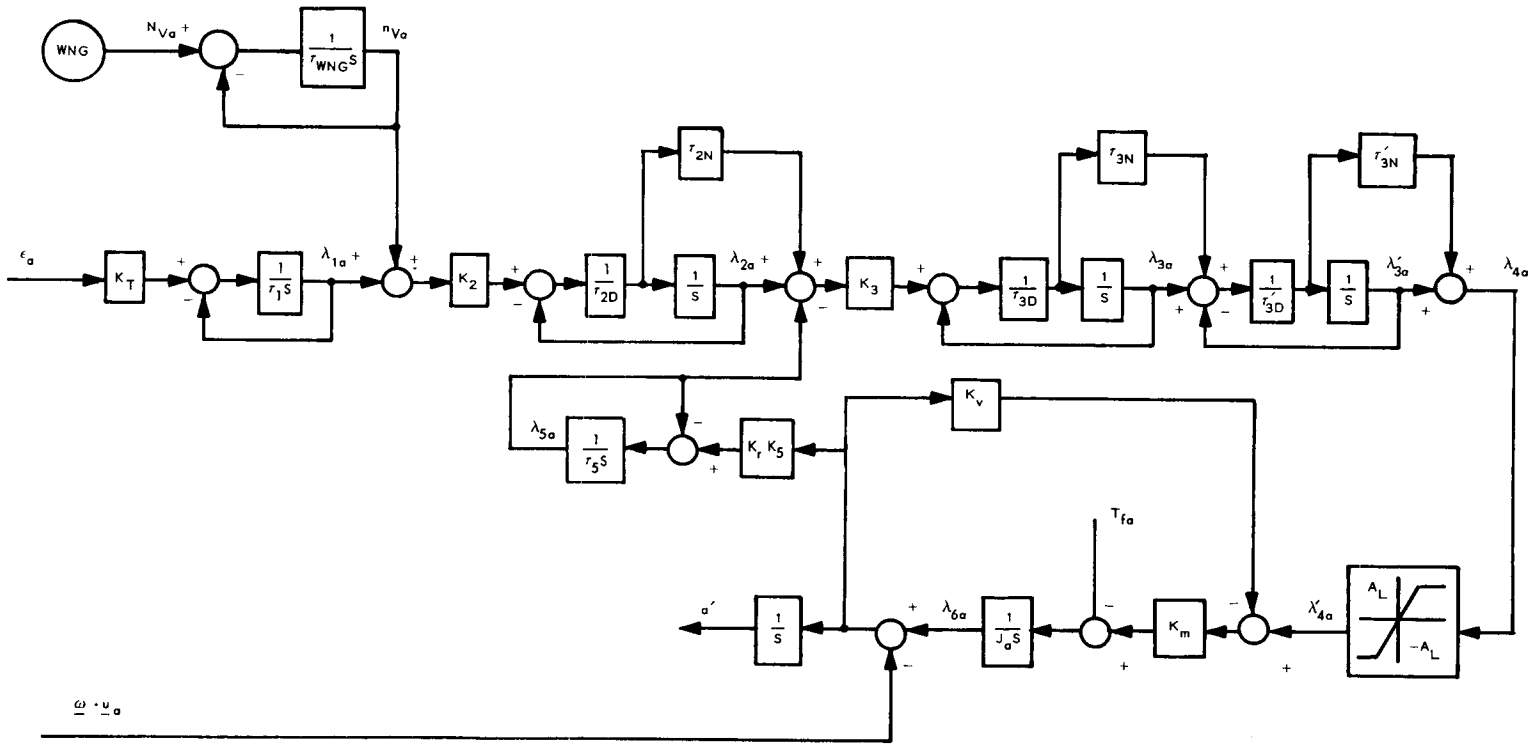


Figure D-1. Star Tracker Servo Loop

$$\dot{\lambda}'_{3a} = \frac{1}{\tau'_{3D}} (\lambda_{3a} + \tau_{3N}\dot{\lambda}_{3a} - \lambda'_{3a}) \tag{D-9}$$

$$\lambda_{4a} = \lambda'_{3a} + \tau'_{3N}\dot{\lambda}'_{3a} \tag{D-10}$$

$$\lambda'_{4a} = \begin{cases} \lambda_{4a}, & |\lambda_{4a}| \leq A_L \\ A_L, & \lambda_{4a} > A_L \\ -A_L, & \lambda_{4a} < -A_L \end{cases} \tag{D-11}$$

$$\dot{a}' = \lambda_{6a} - \underline{\omega} \cdot \underline{u}_a \tag{D-12}$$

$$\dot{\lambda}_{5a} = \frac{1}{\tau_5} (K_r K_5 \dot{a}' - \lambda_{5a}) \tag{D-13}$$

$$\dot{\lambda}_{6a} = \frac{1}{J_a} [K_m (\lambda'_{4a} - K_V \dot{a}') - T_{fa}] \tag{D-14}$$

where T_{fa} is computed using the logic of Figure 2-4 with:

$$\left.\begin{aligned} J &= J_a \\ T_c &= T_{fst} \\ \omega &= \dot{a}' \\ T_a &= K_m (\lambda'_{4a} - K_v \dot{a}') \end{aligned}\right\} \tag{D-15}$$

For the elevation axis equations replace $\dot{a}'$ with $\dot{e}'$ and replace the subscripts (a) with subscripts (e) in equations (D-5) through (D-15).

Horizon Sensor Equations

The first order equations for the horizon sensor lag are:

$$\left.\begin{aligned} \dot{P} &= \frac{1}{\tau_H} (G_H P_a - P) \\ \dot{R} &= \frac{1}{\tau_H} (G_H R_a - R) \end{aligned}\right\} \tag{D-16}$$

and the error model

$$\left.\begin{aligned} P' &= P + \eta_p \\ R' &= R + \eta_r \end{aligned}\right\} \qquad \text{(D-17)}$$

Body Rate Sensors

The rate gyro error model is

$$\underline{\omega}' = \underline{\omega} + \underline{\Delta W} \qquad \text{(D-18)}$$

Appendix E

VARIABLE NAME LISTS

All significant program variables are placed in common blocks. Since these variables may appear in various routines with different names, the following breakdown by common block defines the most applicable variable names.

Variables associated with the control computer have their assumed maximum values listed.

Table E-1 is a matrix defining data linkage.

TABLE E-1. COMMON BLOCK—SUBROUTINE MATRIX

	INITIL	INITL2	INITL3	INITL4	COMPFX	COMPFL	RKII	INPUT	OUTPUT	MANUAL	SUMARY	TRIG	RATE	POSVEL	EXPER	DISTRB	FINDMG	FINDA	FINDB	EVDER	ENERGY	STORE	FANDG	UPDAT3	COST	SINCC	CCATAN	CCASIN	SQRTCC	DMULT	QUOTNT	ZERO
/RKYV/	X	.	.	.	.	.	X	.	X	X	X	X	X	.	X	.	X	.	X	X	X	X	.	X	X	.	.	.	.	.	.	.
/RKYVDB/	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/RKYDV/	.	.	.	.	.	.	X	.	X	X	.	.	.	.	X	.	X	.	.	X	X	.	.	.	.	.	.	.	.	.	.	.
/RKC/	X	.	.	.	.	.	X	.	X	X	X	.	.	X	X	X	.	.	X	X	X	.	X	.	.	.	.	.	.	.	.	.
/SINCO/	.	X	.	.	X	X	.	.	X	X	.	X	X	.	X	.	.	X	X	X	.	.	.	X	X	.	.	.	.	.	.	.
/RATES/	.	.	.	.	.	.	.	.	X	.	.	.	X	.	.	.	.	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.
/AGROUP/	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	X
/BGROUP/	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.
/CGYRO/	X	.	.	.	.	.	.	.	X	.	.	.	.	.	.	X	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.
/CGODY/	X	.	.	.	.	.	.	.	X	.	.	.	.	.	.	X	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.
/CSERVO/	X	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	X	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.
/TORQUE/	X	.	.	.	.	.	.	.	X	X	X	.	.	.	.	.	X	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.
/POWER/	.	X	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.
/PVDATA/	X	X	.	.	.	.	.	.	X	X	.	.	.	X	X	.	.	.	.	.	.	.	X	X	.	.	.	.	.	.	.	.
/CONSTS/	X	X	X	X	.	.	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	X	X	.	.	.	.	.	.	.	.	X
/IOCONT/	.	X	.	X	.	.	.	.	X	X	.	.	.	.	.	.	.	.	X	.	.	X	X	.	X	.	.	.	.	.	.	.
/FLOTIN/	.	X	.	.	X	X	.	.	X	X	.	.	.	.	X	X	X	.	X	.	X	X	.	.	X	.	.	.	.	.	.	.
/FLOOUT/	.	X	.	.	X	X	.	.	X	X	.	.	.	.	.	.	X	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.
/SENSOR/	X	.	.	X	.	.	.	.	X	X	.	.	.	.	X	.	.	.	.	.	X	X	.	.	.	.	.	.	.	.	.	.
/DIST/	X	.	.	.	.	.	.	.	X	.	.	.	.	.	.	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.
/MOVE/	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.
/NPSCAL/	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	X	X	X	.
/SCALER/	.	.	X	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/FLOTSC/	.	.	X	X	X	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	.	.	.	.	.	.	.
/MISCEL/	.	.	X	X	X	X	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	X	X	X	X	.
/FIXIN/	.	.	.	.	X	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/QUANT/	.	X	X	X	X	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	.	.	.	.	.	.	.
/FIXOUT/	.	X	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.
/EXP1V/	.	X	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.
/EXP2V/	.	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/EXP3V/	.	.	X	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.
/MOD567/	.	.	.	X	X	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/CONTL1/	.	.	.	X	X	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/CONTL2/	.	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.
/CONTL3/	.	.	.	X	X	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/DESAT/	.	.	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/TVECT/	.	X	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.
/NAV/	.	.	.	.	X	X	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.
/DIRCOS/	.	X	.	X	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/NTRIG/	.	.	.	.	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/SINCOS/	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	X	X	.	.	.	.
/ASINC/	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.
/ATANC/	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.
/BUFFIN/	X	X	X	X	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X

/RKYV/, Environment State Vector

Location	Variable Name	Symbol	Comments
1 - 3	OMEGAB(3)	$\underline{\omega}$	Vehicle body rates
4 - 6	BETAD(3)	$\dot{\beta}_i$	CMG inner gimbal rates
7 - 9	ALPHAD(3)	$\dot{\alpha}_i$	CMG outer gimbal rates
10 - 12	BETA(3)	β_i	CMG inner gimbal angles
13 - 15	ALPHA(3)	α_i	CMG outer gimbal angles
16	PHI	ϕ	Vehicle Euler Angles
17	THETA	θ	
18	PSI	ψ	
19 - 21	TLOADB(3)	$T_{\beta i}$	CMG gimbal servo state variables
22 - 24	TLOADA(3)	$T_{\alpha i}$	
25 - 27	EMB(3)	$e'_{m\beta i}$	
28 - 30	EMA(3)	$e'_{m\alpha i}$	
31 - 33	EFB(3)	$e'_{f\beta i}$	
34 - 36	EFA(3)	$e'_{f\alpha i}$	
37 - 39	HEXT(3)	$\underline{H}_{ex}$	Inertial components of system angular momentum due to external torques
40 - 42	ENRGYB(3)		Mechanical output energy of the CMG gimbal servos
42 - 45	ENERGYA(3)		
46 - 47	HORIZ(2)	P, R	Ideal output of horizon sensor dynamics
48 - 51	NOISE(4)	n_{vi}	Star tracker servo state variables
52 - 55	STV1(4)	λ_{1i}	
56 - 59	STV2(4)	λ_{2i}	
60 - 63	STV3(4)	λ_{3i}	
64 - 67	STV3P(4)	λ'_{3i}	
68 - 71	STV5(4)	λ_{5i}	
72 - 75	SQTI(4)	λ_{6i}	
76 - 79	QTB(4)	e_1, e_2, a_1, a_2	

/RKYVDB/, Double Precision State Vector

Location	Variable Name	Symbol	Comments
1 - 2	TSTOP		End of integration interval (double precision)
3 - 4	DBLT	t	Time (double precision)
5 - 162	DSV(79)		Double precision state vector

/RKYDV/, State Derivative Vector

Location	Variable Name	Symbol	Comments
1 - 3	OMGABD(3)	$\dot{\underline{\omega}}$	Time derivatives of the state vector variables. One to one correspondence with /RKYV/.
4 - 6	BETADD(3)	$\ddot{\beta}_i$	
7 - 9	ALPHDD(3)	$\ddot{\alpha}_i$	
10 - 12	DRBETA(3)	$d/dt\ \beta$	
13 - 15	DRALPH(3)	$d/dt\ \alpha$	
16	PHID	$\dot{\phi}$	
17	THETAD	$\dot{\theta}$	
18	PSID	$\dot{\psi}$	
19 - 21	TLDDB(3)	$\dot{T}_{\beta i}$	
22 - 24	TLDDA(3)	$\dot{T}_{\alpha i}$	
25 - 27	EMBDOT(3)	$\dot{e}'_{m\beta i}$	
28 - 30	EMADOT(3)	$\dot{e}_{m\alpha i}$	
31 - 33	EFBDOT(3)	$\dot{e}'_{f\beta i}$	
34 - 36	EFADOT(3)	$\dot{e}_{f\alpha i}$	
37 - 39	HEXTD(3)	$\underline{T}_{EX}$	
40 - 42	ENRGBD(3)		
43 - 45	ENRGAD(3)		
46 - 47	HORIZD(2)	$\dot{P}, \dot{R}$	
48 - 51	NOISED(4)	$\dot{n}_{vi}$	
52 - 55	STV 1D(4)	$\dot{\lambda}_{1i}$	
56 - 59	STV 2D(4)	$\dot{\lambda}_{2i}$	
60 - 63	STV 3D(4)	$\dot{\lambda}_{3i}$	

Location	Variable Name	Symbol	Comments
64 - 67	STV 3PD(4)	$\dot{\lambda}_{3i}$	
68 - 71	STV 5D(4)	$\dot{\lambda}_{5i}$	
72 - 75	SQTID(4)	$\dot{\lambda}_{6i}$	
76 - 79	QTBD(4)	$\dot{e}'_1$, $\dot{e}'_2$, $\dot{a}'_1$, $\dot{a}'_2$	

/RKC/, Integrator Controls

Location	Variable Name	Symbol	Comments
1 - 79	U(79)		U(I) is the upper bound of the local truncation error for the ith element of the state vector.
80 - 158	UMIN(79)		UMIN(I) is the lower bound of the local truncation error for the ith element of the state vector.
159	DTMIN		Minimum allowable integrator step size.
160	DTEST		Desired integrator step size.
161	DELTAT		Actual integrator step size.
162	NDOUBL		Number of double commands before doubling step size.
163	NINT		Size of state vector.
164	T	t	Time
165	TRETRN		Integration interval
166 - 244	NOINT(79)		Freeze control

/SINCO/, Trigonometric Functions

Location	Variable Name	Symbol	Comments
1 - 3	SINB(3)	$\sin \beta i$	
4 - 6	SINA(3)	$\sin \alpha i$	
7	SINPHI	$\sin \phi$	
8	SINTHE	$\sin \theta$	
9	SINPSI	$\sin \psi$	
10 - 12	COSB(3)	$\cos \beta i$	
13 - 15	COSA(3)	$\cos \alpha i$	

Location	Variable Name	Symbol	Comments
16	COSPHI	$\cos\phi$	
17	COSTHE	$\cos\theta$	
18	COSPSI	$\cos\psi$	
19 - 21	SINBSQ(3)	$\sin^2\beta_i$	
22 - 24	SINASQ(3)	$\sin^2\alpha_i$	
25 - 27	COSBSQ(3)	$\cos^2\beta_i$	
28 - 30	COSASQ(3)	$\cos^2\alpha_i$	
31 - 39	DIRCO(3, 3)	$[m_{ij}]$	

/RATES/

Location	Variable Name	Symbol	Comments
1 - 9	OMEGA(3, 3)	$\omega_{xi}, \omega_{yi}, \omega_{zi}$	
10 - 18	OMEGAP(3, 3)	$\omega'_{xi}, \omega'_{yi}, \omega'_{zi}$	
19 - 21	OMGASQ(3)		auxiliary variables
22 - 24	OMGAXP(3)		

/AGROUP/

Location	Variable Name	Symbol	Comments
1 - 81	A(9, 9)	$[a_{ij}]$	
82 - 90	AP(3, 3)	$[a'_{ij}]$	
91 - 99	APINV(3, 3)	$[a'_{ij}]^{-1}$	
100	DETAP	$\det [a'_{ij}]$	
101 - 103	DIFIXB(3)		auxiliary variables
104 - 106	AAXB(3)		
107 - 109	XBSQ(3)		
110 - 112	BAXB(3)		

/BGROUP/

Location	Variable Name	Symbol	Comments
1 - 57	BAUX(57)		auxiliary variables
58 - 66	BSMALL(9)	b_i	auxiliary variables
67 - 75	BLARGE(9)		

Location	Variable Name	Symbol	Comments
76 - 84	BPRIME(9)	b_i'	

/CGYRO/

Location	Variable Name	Symbol	Comments
1	AA	A_a	CMG Inertias
2	BA	B_a	
3	CA	C_a	
4	AB	A_b	
5	BB	B_b	
6	CB	C_b	
7	AG	A_g	
8	BG	B_g	
9	JMBETA	$J_{m\beta}$	
10	JMALPH	$J_{m\alpha}$	
11 – 30			Auxiliary constants
31 - 33	AGOMGO(3)	$H_i@t=0$	CMG spin momenta
34 - 36	AGOMGD(3)	$\dot{H}_i$	
37	AGOMGM	H_{max}	
38	HNOM	H_{nom}	

/CBODY/

Location	Variable Name	Symbol	Comments
1	IXX	I_{xx}	Vehicle inertias
2	IYY	I_{yy}	
3	IZZ	I_{zz}	
4	IXY	I_{xy}	
5	IYZ	I_{yz}	
6	IXZ	I_{xz}	

Location	Variable Name	Symbol	Comments
7	IXXO		
8	IYYO		
9	IZZO		Initial vehicle inertias
10	IXYO		
11	IYZO		
12	IXZO		
13	IXXD		
14	IYYD		
15	IZZD		Time derivatives of vehicle inertias
16	IXYD		
17	IYZD		
18	IXZD		
19 - 21	ITERM(3)		Auxiliary variables

/CSERVO/

Location	Variable Name	Symbol	Comments
1 - 3	BINPUT(3)		CMG inner gimbal servo inputs
4 - 6	AINPUT(3)		CMG inner gimbal servo inputs
7	K2BETA	$K_{2\beta}$	
8	K2ALPH	$K_{2\alpha}$	
9	TAUDBE	$\tau_{D\beta}$	
10	TAUDAL	$\tau_{D\alpha}$	
11	KBETA	K_{β}	
12	KALPH	K_{α}	
13	KSFBE	$K_{SF\beta}$	
14	KSFAL	$K_{SF\alpha}$	
15	TAUNBE	$\tau_{N\beta}$	
16	TAUNAL	$\tau_{N\alpha}$	

Location	Variable Name	Symbol	Comments
17	TAUBE	τ_β	
18	TAUAL	τ_α	
19	ELIMBE	$e_{L\beta}$	
20	ELIMAL	$e_{L\alpha}$	
21	KTBE	$K_{T\beta}$	
22	KTAL	$K_{T\alpha}$	
23	KBBETA	$K_{B\beta}$	
24	KBALPH	$K_{B\alpha}$	
25	TAUMBE	$\tau_{M\beta}$	
26	TAUMAL	$\tau_{M\alpha}$	
27	TFBETA	$T_{c\beta}$	
28	TFALPH	$T_{c\alpha}$	
29	BLIMIT	β_{LIM}	
30	ALIMIT	α_{LIM}	
31	GRBETA	G_β	
32	GRALPH	G_α	

/TORQUE/

Location	Variable Name	Symbol	Comments
1 - 3	MJET(3)	M_{Ji}	
4 - 6	MTOT(3)		Total external torques
7 - 9	MB(3)	$M_{\beta i}$	
10 - 12	MA(3)	$M_{\alpha i}$	
13 - 15	MAGJET(3)		Reaction jet torque capability
16 - 18	FUEL(3)		Reaction jet fuel consumed
19 - 21	FLOWRT(3)		Reaction jet fuel rate
22	FUELT		Total jet fuel consumed

/POWER/

Location	Variable Name	Symbol	Comments
1 - 3	PMAXB(3)		Peak mechanical power output of β_i gimbal servo
4 - 6	PMAXA(3)		Peak mechanical power output of α_i gimbal servo
7 - 306	DTLIST(300)		First 300 integrator time steps
307	KDT		Number of integrator steps

/PVDATA/

Location	Variable Name	Symbol	Comments
1 - 3	POS(3)	$\underline{\rho}$	
4 - 6	VEL(3)	$\dot{\underline{\rho}}$	
7 - 9	POSO(3)		Auxiliary variables for Kepler orbit algorithm
10 - 12	VELO(3)		
13	ECCENT		
14	ENOW		
15	MEANA		
16	MEANAO		
17 - 19	FVECT(3)		
20	CETAO		
21	SETAO		
22 - 25	PVCON(4)		
26 - 28	PTARGT(3)	$\underline{R}_o$	

/CONSTS/

Location	Variable Name	Symbol	Comments
1	RTODEG	$180/\pi$	
2	DEGTOR	$\pi/180$	
3	RE	R_e	
4	MU	μ	
5	PIE	π	
6 - 15	WO(10)	ω_o^n	WO(N) = n^{th} power of angular rate of a circular orbit of radius R_e
16	VEO		$= R_e \omega_o$
17 - 26	FIXWO(10)		Scaled version of WO(10)
27	WEARTH	ω_e	Earth's rate

/IOCONT/

Location	Variable Name	Symbol	Comments
1 - 15	ZREAL(15)		A/D delay control variables
16 - 30	NUMBER(15)		
31 - 45	NORDER(15)		
46 - 60	TEVENT(15)		
61	TMATCH		
62	NEVENT		
63 - 77	EVENTT(15)		
78	TCYCLE	t_f	Fast loop cycle time
79	NCOST1		First pass indicator (COST)
80	NMAN1		First pass indicator (MANUAL)
81	NPRINT		Print controls
82	NPRCTL		
83	TEND	t_{END}	Run duration
84	LNECNT		Printer carriage control

/FLOTIN/, Control Computer Input as Supplied by Environment

Location	Variable Name	Symbol	Comments
1 - 3	W(3)	$\underline{\omega}$	These variables are delayed as specified in data cards.
4 - 5	E(2)	e'_1, e'_2	
6 - 7	A(2)	a'_1, a'_2	
8	PH	P'	
9	RH	R'	
10 - 12	BETA(3)	β_i	
13 - 15	ALPHA(3)	α_i	
16 - 18	EDOTC(3)	$\dot{\phi}_c$, $\dot{\theta}_c$, $\dot{\psi}_c$	
19 - 21	EC(3)	ϕ_c, θ_c, ψ_c	
22	TM	t_{man}	
23 - 25	HCL(3)	H_i	
26	MODE	m_x	
27	RATEFB	m_{FB}	
28	UPDATE	m_u	
29	NGAIN	m_G	
30	LAW	m_L	
31	MODCOM	m_ω	
32 - 34	LIMIG(3)	L_{IGi}	
35 - 37	LIMOG(3)	L_{OGi}	

/FLOOUT/, Control Computer Output as Used by Environment

Location	Variable Name	Symbol	Comments
1 - 3	BETADC(3)	$\dot{\beta}_{ci}$	
4 - 6	ALPHDC(3)	$\dot{\alpha}_{ci}$	
7	TELAZC	A_c	
8 - 10	EULER(3)	ϕ', θ', ψ'	For display only
11 - 13	EPSLON(3)	ϵ'_i	
14 - 16	OMGABC(3)	$\underline{\omega}_c$	
17 - 19	JET(3)	$\underline{T}_{Ji}$	

/SENSOR/

Location	Variable Name	Symbol	Comments
1 - 2	ESTAR(2)	E_i	Star tracker variables
3 - 4	ASTAR(2)	A_i	
5	TAUWNG	τ_{WNG}	
6	TAU1	τ_1	
7	TAU2N	τ_{2N}	
8	TAU2D	τ_{2D}	
9	TAU3N	τ_{3N}	
10	TAU3D	τ_{3D}	
11	TAU3NP	τ'_{3N}	
12	TAU3DP	τ'_{3D}	
13	TAU5	τ_5	
14	STKT	K_T	
15	STG2	G_2	
16	STG3	G_3	
17	STAL	A_L	
18	STKM	K_m	
19	STKV	K_v	
20	STTF	$T_{fs\tau}$	
21	STKR	K_r	
22 - 25	JINRT(4)	J_e, J_e, J_a, J_a	
26 - 27	ETA(2)	η_p, η_r	Horizon sensor bias
28	TAUHRZ	τ_H	Horizon sensor time constant
29	GHORIZ	G_H	Horizon sensor gain
30 - 32	DELW(3)	$\Delta\omega_i$	Rate gyro bias
33 - 36	STBIAS(4)	N_{vi}	Mean value, Standard deviation } star tracker noise
37	STSIG	σ_n	

Location	Variable Name	Symbol	Comments
38 - 39	SXI(2)	S_{iXI}	Star tracker Variables
40 - 41	SYI(2)	S_{iYI}	
42 - 43	SZI(2)	S_{iZI}	
44 - 45	SX(2)	S_{ix}	
46 - 47	SY(2)	S_{iy}	
48 - 49	SZ(2)	S_{iz}	
50 - 51	ELBORE(2)	e_i	
52 - 53	AZBORE(2)	a_i	
54 - 57	QST(4)	$\epsilon_{e1}, \epsilon_{e2}, \epsilon_{a1}, \epsilon_{a2}$	Star tracker image plane errors.
58 - 61	WNG(4)	N_{vi}	White noise
62 - 63	HRZACT(2)	P_a, R_a	Actual horizon errors.

/DIST/

Location	Variable Name	Symbol	Comments
1 - 3	MDIST(3)	M_x, M_y, M_z	External disturbance torques
4 - 6	MDNOM(3)	A_{xo}, A_{yo}, A_{zo}	External disturbance torque parameters
7 - 24	SPHASE(3, 6)	$\sin B_{ij}$, i=x, y, z	
25 - 42	CPHASE(3, 6)	$\cos B_{ij}$, i=x, y, z	
43 - 48	SFREQ(6)	$\sin \omega_i t$	
49 - 54	CFREQ(6)	$\cos \omega_i t$	
55 - 72	MDAMP(3, 6)	A_{ij}, i=x, y, z	
73 - 81	FREQ(6)	ω_i	
79 - 81	TQEMM(3)	T_{mm}	Torque due to moving mass.

/MOVE/

Location	Variable Name	Symbol	Comments
1	QMASS	Q	Moving mass parameters
2	QMASS2	$2Q$	
3 - 5	PMASS(3)	x, y, z	
6 - 8	VMASS(3)	$\dot{x}, \dot{y}, \dot{z}$	
9 - 11	AMASS(3)	$\ddot{x}, \ddot{y}, \ddot{z}$	
12 - 14	PMASSO(3)	x_o, y_o, z_o	
15 - 17	VMASSO(3)	$\dot{x}_o, \dot{y}_o, \dot{z}_o$	
18 - 20	AMASSO(3)	$\ddot{x}_o, \ddot{y}_o, \ddot{z}_o$	
21 - 23	R(3)	R_i	
24 - 26	ROMGA(3)	$R_i \Omega_i$	
27 - 29	ROMGA2(3)	$R_i \Omega^2_i$	
30 - 32	OMGAMM(3)	Ω_i	
33 - 35	HALFAM(3)	$\ddot{x}_o/2,\ \ddot{y}_o/2,\ \ddot{z}_o/2$	
36 - 38	COSMM(3)	$\cos \Omega_i t$	
39 - 41	SINMM(3)	$\sin \Omega_i t$	

/NPSCAL/, Integer Scaling Factors

Location	Variable Name	Symbol	Comments
1	NM7	2^{n-7}	
2	NM6	2^{n-6}	
3	NM5	2^{n-5}	
4	NM4	2^{n-4}	
5	NM3	2^{n-3}	
6	NM2	2^{n-2}	
7	NM1	2^{n-1}	
8	NP0	2^{n}	
9	NP1	2^{n+1}	
10	NP2	2^{n+2}	

Location	Variable Name	Symbol	Comments
11	NP3	2^{n+3}	
12	NP4	2^{n+4}	
13	NP5	2^{n+5}	
14	NP6	2^{n+6}	
15	NP7	2^{n+7}	
16	NP8	2^{n+8}	
17	NP9	2^{n+9}	

/SCALER/, Integer Scaling Factors

Location	Variable Name	Symbol	Comments
1	CSCALE	2^{5-nh}	Since t_f is adjustable, the program optimizes the scale factors associated with it. The variable nh is computed so that $2^{nh-1} < t_f \leq 2^{nh}$
2	DSCALE	2^{6-nh}	
3	LSCALE	2^{n-11}	
4	LLSCAL	2^{7-nh}	
5	MSCALE	2^{11-n}	
6	NSCALE	2^{12-n}	
7	PSCALE	2^{n-12}	
8	QSCALE	2^{15-n}	The other factors here involve decision scaling (i.e., if n=14, RSCALE= $0 \neq 2^{n-15}$; hence QSCALE= $2^{15-n}=2$ is used instead).
9	RSCALE	2^{n-15}	
10	TSCALE	2^{n-14}	
11	VSCALE	2^{14-n}	

/FLOTSC/, Floating Point Scaling Factors

Location	Variable Name	Symbol	Comments
1	FLNM7	$2.^{n-7}$	
2	FLNM6	$2.^{n-6}$	
3	FLNM5	$2.^{n-5}$	
4	FLNM4	$2.^{n-4}$	
5	FLNM3	$2.^{n-3}$	

Location	Variable Name	Symbol	Comments
6	FLNM2	$2.^{n-2}$	
7	FLNM1	$2.^{n-1}$	
8	FLNP0	$2.^{n}$	
9	FLNP1	$2.^{n+1}$	
10	FLNP2	$2.^{n+2}$	
11	FLNP3	$2.^{n+3}$	
12	FLNP4	$2.^{n+4}$	
13	FLNP5	$2.^{n+5}$	
14	FLNP6	$2.^{n+6}$	
15	FLNP7	$2.^{n+7}$	
16	FLNP8	$2.^{n+8}$	
17	FLNP9	$2.^{n+9}$	
18	FLNM11	$2.^{n-11}$	
19	FLNM10	$2.^{n-10}$	
20	FLNM8	$2.^{n-8}$	
21	FLNP12	$2.^{n+12}$	
22	F2NM25	$2.^{2n-25}$	
23	F2NM15	$2.^{2n-15}$	
24	F2NM10	$2.^{2n-10}$	
25	FL2NM2	$2.^{2n-2}$	
26	FL2NM1	$2.^{2n-1}$	
27	FL2NP0	$2.^{2n}$	
28	FL2NP1	$2.^{2n+1}$	

/MISCEL/

Location	Variable Name	Symbol	Comments
1	FS	$2^{n}-1$	
2	DBLFS	$2^{2n}-1$	
3	N	n	

Location	Variable Name	Symbol	Comments
4	NH	nh	
5	MDLAST	m_{xp}	
6	HALFFS	2^{n-1}	

/FIXIN/, Control Computer Input Vector

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	W(3)	$\underline{\omega}'$	2^{-5}	
4 - 5	E(2)	e'_1, e'_2	2^2	
6 - 7	A(2)	a'_1, a'_2	2^2	
8	PH	P'	2^{-1}	
9	RH	R'	2^{-1}	
10 - 12	BETA(3)	β'_1	2^2	
13 - 15	ALPHA(3)	α'_1	2^2	
16 - 18	EDOTC(3)	$\dot{\phi}_c$, $\dot{\theta}_c$, $\dot{\psi}_c$	2^{-5}	
19 - 21	EC(3)	ϕ_c, θ_c, ψ_c	2^2, 2^1, 2^2	
22	TM	t_{man}	2^8	
23 - 25	HCL(3)	H'_i	2^{11}	
26	MODE	m_x		Discretes
27	RATEFB	m_{FB}		
28	UPDATE	m_u		
29	NGAIN	m_G		
30	LAW	m_L		
31	MODCOM	m_ω		
32 - 34	LIMIG(3)	L_{IGi}		
35 - 37	LIMOG(3)	L_{OGi}		

/QUANT/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 25	NBIN(25)	$2^{n-n_{BIi}}$		n_{BIi} = the number of bits of A/D conversion of the ith variable of /FIXIN/
26 - 41	NBOUT(16)	$2^{n-n_{BOi}}$		n_{BOi} = the number of bits of D/A conversion of the ith variable of /FIXOUT/
42	NFXPNT			Fixed point indicator

/FIXOUT/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	BDOTC(3)	$\dot{\beta}_{ci}$	2^{-2}	
4 - 6	ADOTC(3)	$\dot{\alpha}_{ci}$	2^{-2}	
7	AC	A_c	2^0	
8 - 10	ED(3)	ϕ', θ', ψ'	2^2, 2^1, 2^2	
11 - 13	EP(3)	$\underline{\epsilon}'$	2^{-1}	Double precision
14 - 16	WC(3)	$\underset{*}{\underline{\omega}}_c$	2^{-5}	
17 - 19	NJET(3)	T_{Ji}		Discrete

/EXP1/

Location	Variable Name	Symbol	Max. Value	Comments
1	ACDBL	$\underset{*}{A}_c$	2^0	DP
2	ACDOT	$\dot{A}_c$	2^{-5}	
3	ADSAVE	$\dot{A}_{cp}$	2^{-5}	
4	COSAC	$\cos A_c$	2^0	
5	COSWT	$\underset{*}{\cos}\omega_e t$	2^0	DP
6 - 8	OMEGA(3)	$\underline{\Omega}'_{EI}$	2^{-13}	
9 - 11	OMEGAE(3)	$\underline{\Omega}'_{EII}$	2^{-13}	$= (0, \omega_e, 0)^T$
12 - 14	RO(3)	$\underset{*}{\underline{R}}_o$	2^{25}	DP
15	S	S'	2^{23}	

Location	Variable Name	Symbol	Max. Value	Comments
16	SDOT	$\dot{S}'$	2^{15}	
17	SDUM			Auxiliary variable
18	SINAC	$\sin A_c$	2^0	
19	SINWT	$\sin \underset{*}{\omega}_e t$	2^{-3}	DP
20	SPRIME	$s' \cos A_c$	2^{23}	
21 - 23	SREL(3)	$\underset{*}{\underline{S}}'$	2^{25}	DP
24	SSQ	$\underset{*}{S}'^2$	2^{50}	DP
25	TANAC	$\tan A_c$	2^1	
26 - 28	V(3)	$\underline{V}'_I$	2^{15}	
29 - 31	VC(3)	$\underline{V}'_c$	2^{15}	
32 - 34	VDOUB(3)	$\underset{*}{\underline{V}}'_I$	2^{15}	DP
35	WE	$\underset{*}{\omega}_e$	2^{-13}	DP
36	WE2	$\underset{*}{\omega}^2_e$	2^{-27}	DP
37	WE3	$\underset{*}{\omega}^3_e$	2^{-41}	DP
38	WE4	$\underset{*}{\omega}^4_e$	2^{-54}	DP

/EXP2/

Location	Variable Name	Symbol	Max. Value	Comments
1	DEL	Δ		
2	DELP	Δ_a		
3	DEL1	Δ_1		
4	DEL2	Δ_2		
5	DEL3	Δ_3		
6	S1X	S_{1XI}		
7	S2X	S_{2XI}	2^0	

Location	Variable Name	Symbol	Max. Value	Comments
8	S1Y	S_{1YI}		
9	S2Y	S_{2YI}		
10	S1Z	S_{1ZI}		
11	S2Z	S_{2ZI}		
12	U1X	U'_{1X}		
13	U2X	U'_{2X}		
14	U1Z	U'_{1Z}		
15	U2Z	U'_{2Z}		

/EXP3V/

Location	Variable Name	Symbol	Max. Value	Comments
1	ANGLE	η_*	2^2	DP
2	COSDUM	$\cos \eta$	2^0	
3	COSL	$\cos \lambda'$	2^0	
4	COSLR	$\cos \lambda'_R$	2^0	
5	COSTH	$\cos L'$	2^0	
6	DELANG	$\dot{\eta} \Delta t_s$	2^2	
7	DELX	δ'_{XI}	2^{-8}	
8	DELY	δ'_{YI}	2^{-8}	
9	DELZ	δ'_{ZI}	2^{-8}	
10	DBLPI	π_*	2^2	DP
11	ETADOT	$\dot{\eta}$	2^{-5}	
12	KC		2^{-8}	= .003373
13	PREV	$\dot{\epsilon}'_{zp}$	2^{-5}	
14 - 16	PSP(3)	$\underline{\rho}'$	2^{25}	
17	R	$\lvert \underline{\rho}' \rvert$	2^{25}	
18	SINDUM	$\sin \eta$	2^0	
19	SINL	$\sin \lambda'$	2^0	

Location	Variable Name	Symbol	Max. Value	Comments
20	SINLR	$\sin\lambda'_R$	2^0	
21	SIN2LR	$\sin 2\lambda'_R$	2^0	
22	SINTH	$\sin L'$	2^0	
23 - 24	SPX(2)	S'_{ix}	2^0	
25 - 26	SPY(2)	S'_{iy}	2^0	
27 - 28	SPZ(2)	S'_{iz}	2^0	
29	S1XG	S_{1XG}	2^0	
30	S1ZG	S_{1ZG}	2^0	
31	S2ZG	S_{2ZG}	2^0	
32	WBAR	ω_{AV}	2^{-5}	
33	Z	$\sqrt{\rho'^2_x + \rho'^2_z}$	2^{25}	
34	ZEXI	Z'_{EXI}	2^0	
35	ZEYI	Z'_{EYI}	2^0	
36	ZEZI	Z'_{EZI}	2^0	
37	EPZDOT	$\dot{\epsilon}'_z$	2^{-5}	

/MOD567/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	DELE(3)	$\Delta\phi', \Delta\theta', \Delta\psi'$	$2^2, 2^1, 2^2$	
4 - 6	ECOM(3)	$\phi'_c, \theta'_c, \psi'_c$	$2^2, 2^1, 2^2$	
7 - 9	EDCOM(3)	$\dot{\phi}'_c, \dot{\theta}'_c, \dot{\psi}'_c$	2^{-5}	
10	TENDM	t_{ENDM}	2^{10}	
11 - 13	ECOMDP(3)	$\underset{*}{\phi}'_c, \underset{*}{\theta}'_c, \underset{*}{\psi}'_c$	$2^2, 2^1, 2^2$	DP
14 - 16	DELTAE(3)			Auxiliary DELE(3)
17	MAX			Auxiliary variable
18	MAXRT	ω_{BMAX}	2^{-5}	
19	TP	t'	2^{10}	
20	NHHH		2^{nh}	Auxiliary Δt_f

/CONTL1/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	EPPREV(3)	$\frac{\epsilon'}{*}$ p	2^{-1}	DP
4 - 33	GAIN(6,5)	K_{ei}, m_G	2^{13}	i=1,6 ; m_G = 1,5
34 - 63	GAINP(6,5)	K_{ri}, m_G	2^{10}	
64 - 66	EPP(3)	$\frac{\epsilon''}{*}$	2^{-1}	DP
67	HNOMP	H_{NOM}	2^{11}	
68 - 73	KLCL(6)	k_i	2^{-2}	
74 - 76	MAGA(3)	$\lvert \underline{a}_i \rvert$	2^{11}	
77 - 79	MAGB(3)	$\lvert \underline{b}_i \rvert$	2^{11}	
80 - 82	MCA(3)			Not currently used
83 - 85	MCB(3)	$\underline{T}_b$	2^6	
86 - 88	TRQC(3)	$\underline{T}_d$	2^8	
89 - 91	TRQCP(3)	$\underline{T}_c$	2^8	
92 - 100	UNITVA(3,3)	$\underline{a}_i / \lvert \underline{a}_i \rvert$	2^0	1st subscript=vector number
101 - 109	UNITVB(3,3)	$\underline{b}_i / \lvert \underline{b}_i \rvert$	2^0	2nd subscript=component no.
110 - 114	ZO(5)	$Z_{0\ mG}$		

/CONTL2/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	DELA(3)	$\underline{\Delta \dot{\alpha}}_c$	2^{-2}	
4 - 6	DELB(3)	$\underline{\Delta \dot{\beta}}_c$	2^{-2}	
7 - 9	DOT1(3)	$\underline{\Delta T} \cdot \underline{h}_i$	2^8	
10 - 12	DOT2(3)	$\underline{h}_m \cdot \underline{a}_i$	2^{11}	
13 - 15	DOT3(3)	$\underline{h}_m \cdot \underline{b}_i$	2^{11}	
16	DUM1			Auxiliary variables
17	DUM2			
18	KSAVE	K_s	2^0	
19 - 21	MAGASQ(3)	$\lvert \underline{a}_i \rvert^2$	2^{22}	

Location	Variable Name	Symbol	Max. Value	Comments
22 - 24	MAGBSQ(3)	$\lvert \underline{b}_i \rvert^2$	2^{22}	
25 - 27	TREM(3)	$\underline{T}_r$	2^8	
28 - 30	TRQPRD(3)	$\underline{\Delta T}$	2^8	1st sub = vector number
31 - 39	UNITVH(3,3)	$\underline{h}_i$	2^0	2nd sub=component number
40	BHOLD	β_{HOLD}		Floating point
41	BSELFD	β_{SELFD}	2^2	
42	BDOTDS	$\dot{\beta}_{DS}$	2^{-2}	
43	MBMAX	M_β		Discrete

/CONTL3/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	CGYRO(3)	C_{Gi}	2^0	
4 - 6	COSTA(3)	C_{ai}	2^0	
7 - 9	COSTB(3)	C_{bi}	2^0	
10	RTEST		2^{-2}	Auxiliary variables
11	RUSE		2^{-2}	
12	TDESSQ	$\lvert \underline{T}_d \rvert^2$	2^{16}	
13 - 15	TDOTA(3)	$\underline{T}_r \cdot \underline{a}_i$	2^8	
16 - 18	TDOTB(3)	$\underline{T}_r \cdot \underline{b}_i$	2^8	
19	TREMSQ	$\lvert \underline{T}_r \rvert^2$	2^{16}	
20	XKEND	K_x	2^0	
21	ITER	i_{MAX}		Integers
22	NOITER			

/DESAT/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 3	ERRLIM(3)	ϵ_{Li}	2^{-1}	
4 - 6	JETCT(3)			Auxiliary variables
7 - 9	TJCNT(3)			
10 - 12	GIMLIM(3)			

Location	Variable Name	Symbol	Max. Value	Comments
13	BDOTMX	$\dot{\beta}_{LIM}$	2^{-2}	
14	ADOTMX	$\dot{\alpha}_{LIM}$	2^{-2}	

/TVECT/

Location	Variable Name	Symbol	Max. Value	Comments
1	DELT	Δt_{*s}	2^{10}	DP
2	H	Δt_f	2^{nh}	
3 - 12	TIME(10)	$t^n/n!$		DP: $2^{10}, 2^{19}, 2^{28}, 2^{36}, 2^{44}$ $2^{51}, 2^{58}, 2^{65}, 2^{72}, 2^{79}$
13	NPASS	n_{PASS}		Integer
14	NSLOW	n_{SLOW}		Integer
15	XNSLOW	n_{SLOW}		Floating point

/NAV/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 10	F(10)	f_{*i}		DP: $2^{-8}, 2^{-18}, 2^{-27}, 2^{-37}, 2^{-47}, 2^{-56}, 2^{-66}, 2^{-76}, 2^{-85}, 2^{-95}$
11	FDOT	$\dot{f}_{*T}$	2^{-9}	DP
12 - 21	FDUM(10)			Auxiliary variables
22	FTOT	f_{*T}	2^{1}	DP
23 - 32	G(10)	$_{*}g_i$		DP: $2^{1}, 2^{-8}, 2^{-18}, 2^{-27}, 2^{-37}, 2^{-47}, 2^{-56}, 2^{-66}, 2^{-76}, 2^{-85}$
33	GDOT	$_{*}\dot{g}_T$	2^{1}	DP
34 - 43	GDUM(10)			Auxiliary variables
44	GTOT	$_{*}g_T$	2^{11}	DP
45 - 47	P(3)	ρ_*'	2^{25}	DP
48 - 50	PO(3)	ρ_{*0}	2^{25}	DP
51 - 53	PDOT(3)	$\dot{\rho}_*'$	2^{15}	DP
54 - 56	PDOTO(3)	$\dot{\rho}_{*0}$	2^{15}	DP

/DIRCOS/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 9	MS(3,3)	$[m'_{ij}]$	2^0	
10 - 18	MSDOT(3,3)	$[\dot{m}'_{ij}]$	2^{-5}	
19 - 27	ML(3,3)	$[M'_{ij}]$	2^0	
28 - 36	MLDOT(3,3)	$[\dot{M}'_{ij}]$	2^{-5}	
37 - 45	DSAVE(3,3)		2^{-5}	Auxiliary variable
46 - 54	MSDBL(3,3)	$[m'_{*ij}]$	2^0	DP
55 - 63	MLDBL(3,3)	$[M'_{*ij}]$	2^0	DP
64 - 66	XW(3)		2^{-5}	Auxiliary variable
67 - 69	WPREV	$\underline{\omega}'_p$	2^{-5}	
70 - 72	TRATIO(3)	t_{ri}	2^0	
73 - 75	DELW(3)	$\underline{\Delta\omega}$	2^{-5}	

/NTRIG/

Location	Variable Name	Symbol	Max. Value	Comments
1 - 2	COSE(2)	$\cos e'_i$	2^0	
3 - 4	COSA(2)	$\cos a'_i$	2^0	
5 - 7	CBETA(3)	$\cos \beta'_i$	2^0	
8 - 10	CALPHA(3)	$\cos \alpha'_i$	2^0	
11 - 12	SINE(2)	$\sin e'_i$	2^0	
13 - 14	SINA(2)	$\sin a'_i$	2^0	
15 - 17	SBETA(3)	$\sin \beta'_i$	2^0	
18 - 20	SALPHA(3)	$\sin \alpha'_i$	2^0	
21	SINED	$\sin \phi'$	2^0	
22	COSED	$\cos \phi'$	2^0	

/SINCOS/, Parameters of sin-cos Subroutine

Location	Variable Name	Symbol	Max. Value	Comments
1	ASC	a_{sc}	2^{-2}	
2	B	b_{sc}	2^{-6}	
3	C	c_{sc}	2^{-12}	
4	HALFPI	$\pi/2$	2^{1}	
5	PI	π	2^{2}	
6	HAFPI	$\pi/2$	2^{2}	

/ASINC/, Parameters of arc sin Subroutine

Location	Variable Name	Symbol	Max. Value	Comments
1	AAS	a_{as}	2^{1}	
2	BAS	b_{as}	2^{-2}	
3	CAS	c_{as}	2^{-3}	
4	DAS	d_{as}	2^{-5}	

/ATANC/, Parameters of arc tangent Subroutine

Location	Variable Name	Symbol	Max. Value	Comments
1	AAT	a_{at}	2^{0}	
2	BAT	b_{at}	2^{-1}	
3	CAT	c_{at}	2^{-2}	
4	DAT	d_{at}	2^{-4}	
5	QUARPI	$\pi/4$	2^{0}	

/BUFFIN/, Input Data Buffer (Detailed listing in User's Guide)

Location	Variable Name	Symbol	Max. Value	Comments
1 - 339	XZ(339)			System specification (Real)
340 - 463	NZ(124)			System specification (Integer)
464 - 557	XY(94)			Control computer data (Real)
558 - 566	NY(9)			Control computer data (Integer)

Appendix F

ANGULAR MOMENTUM TEST

The proposed test, when satisfied, indicates that nothing basic is being violated. The test is based on the principle of conservation of angular momentum. That is, in the absence of external torques, the angular momentum of the total system is equal to its value at time t = 0. If external torques $\underline{T}_{ex}$ are acting on the system, the angular momentum of the total system, after the momentum increment due to $\underline{T}_{ex}$ has been subtracted from it, should also be constant and equal to the angular momentum of the total system at time t = 0. Specifically,

$$\underset{\text{Vehicle}}{\underline{H}(t)} + \sum_{1}^{3} \underset{\text{Gyro}}{\underline{H}_i(t)} - \int_0^t \underline{T}_{ex}\, dt = \underline{H}(o) = \text{Constant} \qquad \text{(F-1)}$$

where

$\underline{H}(o)$ = Angular momentum of the total system at time t = 0

$\underline{T}_{ex}$ = External torque acting on the vehicle.

Note that, even though the computed LHS of equation (F-1) is constant as a factor quantity, its components in vehicle coordinates change as the vehicle attitude changes. Therefore, to readily verify that the LHS of equation (F-1) is constant, it must be resolved along an inertially fixed frame.

The Angular Momentum Test, therefore, consists of the following steps:

1. At time t = 0 compute the LHS of equation (F-1) with components along vehicle axes.
2. Transform $\underline{H}(o)$ computed in step #1 to obtain its components along the axes of a suitable Inertial frame. For convenience the vehicle axes as they are oriented at time t = 0 can be chosen as the I-frame for this test. In this case, step #2 is not needed.
3. To compute the LHS of equation (F-1) at the end of the nth integration step, that is at $t = n\Delta T$, it is necessary to treat the LHS of equation (F-1) in the following two parts:

a. Compute

$$\underline{H}(t)_{Vehicle} + \sum_{1}^{3} \underline{H}_i(t)_{Gyro}$$

at $t = n\Delta T$ along vehicle axes and convert this to the selected I-frame.

b. Integrate

$$\underline{H}_{ex} = \int_0^{n\Delta T} (\underline{T}_{ex})_I \, dt$$

4. Substitute the vectors computed in steps #3a and #3b into the LHS of equation (F-1). The result, to satisfy the test, should be equal to the angular momentum computed in step #2.

Computation of $\underline{H}_{Vehicle}$. —Along vehicle axes X, Y, Z

$$\underline{H}_{Vehicle} = H_{Vx}\underline{I} + H_{Vy}\underline{J} + H_{Vz}\underline{K}$$

where

$$H_{Vx} = I_{xx}\omega_x - I_{xy}\omega_y - I_{xz}\omega_z$$

$$H_{Vy} = -I_{xy}\omega_x + I_{yy}\omega_y - I_{yz}\omega_z$$

$$H_{Vz} = -I_{xz}\omega_x - I_{yz}\omega_y + I_{zz}\omega_z$$

Computation of $\sum_{1}^{3} \underline{H}_{i_{Gyro}}$. —For #1 CMG

$$\underline{H}_{G1} = H_{G1x}\underline{I} + H_{G1y}\underline{J} + H_{G1z}\underline{K}$$

where

$$H_{G1x} = (\omega'_{z1} + G_\beta \dot{\beta}_1) J_{M\beta} \sin\alpha_1 + A_a \omega'_{x1} \cos\alpha_1 + C_a \omega'_{z1} \sin\alpha_1$$
$$+ [(A_b + A_g)\omega_{x1} + A_g \Omega_1] \cos\alpha_1 \cos\beta_1$$
$$- (B_b + B_g)\omega_{y1} \cos\alpha_1 \sin\beta_1 + (C_b + B_g)\omega_{z1} \sin\alpha_1$$

$$H_{G1y} = (\omega_y + G_\alpha \dot{\alpha}_1) J_{M\alpha} + B_a \omega'_{y1} + (B_b + B_g) \omega_{y1} \cos \beta_1$$
$$+ [(A_b + A_g) \omega_{x1} + A_g \Omega_1] \sin \beta_1$$

$$H_{G1z} = (\omega'_{z1} + G_\beta \dot{\beta}_1) J_{M\beta} \cos \alpha_1 - A_a \omega'_{x1} \sin \alpha_1 + C_a \omega'_{z1} \cos \alpha_1$$
$$- [(A_b + A_g) \omega_{x1} + A_g \Omega_1] \sin \alpha_1 \cos \beta_1$$
$$+ (B_b + B_g) \omega_{y1} \sin \alpha_1 \sin \beta_1 + (C_b + B_g) \omega_{z1} \cos \alpha_1$$

The angular momentum contribution of the three gyros can be computed in a DO LOOP and the results summed. For the DO LOOP cycle subscripts so that

$1 \rightarrow 2 \rightarrow 3 \rightarrow 1$

$x \rightarrow y \rightarrow z \rightarrow x$

These check equations are based on a constant system inertia tensor about a fixed center of mass. The necessary changes to adapt them for the case of a moving mass within the vehicle have not been made.