AC ELECTRONICS DIVISION GENERAL MOTORS CORPORATION

INTERPLANETARY GUIDANCE SYSTEM REQUIREMENTS STUDY

VOLUME II

COMPUTER PROGRAM DESCRIPTION

PART 3

NOMINAL POWERED FLIGHT TRAJECTORIES

Prepared for
ERC Systems Laboratory
under Contract NAS 12-7

AC ELECTRONICS DIVISION
General Motors Corporation
El Segundo, California

TABLE OF CONTENTS

SECTION	TITLE	PAGE
1.0	INTRODUCTION AND SUMMARY	1-1
2.0	MATHEMATICAL MODEL	2-1
	2.1 Coordinate Systems	2-2
	2.1.1 Inertial Coordinate Systems	2-2
	2.1.2 Noninertial Coordinate System	2-2
	2.2 Environmental Models	2-3
	2.2.1 Figure of Planets	2-3
	2.2.2 Gravitational Potential	2-3
	2.2.3 Planetary Atmosphere	2-4
	2.3 Missile Dynamics	2-4
	2.3.1 Propulsion Model	2-4
	2.3.2 Aerodynamic Model	2-5
	2.3.3 Equations of Motion	2-5
	2.4 Autopilot Model	2-6
	2.5 Guidance Schemes	2-6
	2.5.1 General Guidance and Steering Concepts	2-6
	2.5.2 β^* Guidance	2-7
	2.5.3 Velocity Profile Guidance	2-7
	2.5.4 Explicit Guidance with Time of Flight Constraint	2-8
	2.5.5 Explicit Guidance with Velocity Direction Constraints	2-9
	2.5.6 Trim Steering Guidance	2-10
3.0	ORGANIZATION OF THE PROGRAM	3-1
	3.1 Schema for Flow Chart Presentation	3-2
	3.2 Definition of Flow Chart Symbols	3-2
	3.3 Definition of Mathematical Symbols	3-4
	3.4 Coordinate Systems	3-23
	3.5 Generalized Booster Guidance Simulation	3-28
	3.5.1 Level I Flow Chart	3-28
4.0	COMPUTER PROGRAM DESCRIPTION	4-1
	4.1 Input and Output	4-2
	4.1.1 Definition of Flags	4-2
	4.1.2 Definition of Input Quantities	4-4

TABLE OF CONTENTS

SECTION TITLE PAGE

4.1.3 Input - Block A 4-14
4.1.4 Block B 4-25

4.2 Basic Computation Blocks 4-32

4.2.1 Initialization - Block I 4-33
4.2.2 Dynamics - Block II 4-41
4.2.3 Integration - Block III 4-77
4.2.4 Guidance Logic - Block IV 4-81
4.2.5 Control and Printout - Block V 4-107
4.2.6 Phase Setup - Block VI 4-111
4.2.7 Atmospheric Trigger - Block VII 4-113

5.0 USER'S GUIDE 5-1

5.1 General Capabilities of Program 5-1
5.2 Discussion of Input Quantities 5-6

5.2.1 Dynamics 5-6
5.2.2 Guidance 5-10
5.2.3 Operational 5-16

5.3 Specific Examples 5-18

5.3.1 Simulation of Boost from a Circular Orbit about Earth into an Interplanetary Trajectory 5-18
5.3.2 Deboost into a Circular Orbit 5-21
5.3.3 Simulation of Launch from Mars into a Circular Orbit 5-24
5.3.4 Flyby Trajectory Transfer 5-26
5.3.5 Midccurse Correction 5-28
5.3.6 Deboost Prior to Earth Re-entry 5-34

6.0 REFERENCES 6-1

APPENDIX A A-1

APPENDIX B B-1

APPENDIX C C-1

APPENDIX D - COMPUTER PROGRAM 118.0 D-1

D.1 Introduction D-1
D.2 Programmer's Operational Information D-2

TABLE OF CONTENTS

SECTION TITLE PAGE

D.2.1 System Configuration D-2
D.2.2 Programmer's Guide for 118.0 D-3
D.2.3 Operator's Guide for 118.0 D-3
D.2.4 Tape Format for 118.0 D-5
D.2.5 Utility Subroutines D-7

D.3 Program Listing, Program 118.0 D-62

APPENDIX E E-1

1.0 INTRODUCTION AND SUMMARY

The purpose of this document is to describe the booster program 118.0 and provide a guide for an engineer to use in running the program.

Program 118.0 was initially written for simulation of various vehicles from launch at the earth's surface into various orbits. It is a complex but flexible program and is adaptable for simulating space boost missions. Seven specific missions that have been generated on this program are provided as examples in Section 5, the User's Guide.

Program 118.0 is used as a tool to shape the booster nominal trajectories. This can be done in various ways within the program, depending on the familiarity with the program and ingenuity of the user. Program 117.0 can also perform the same function as program 118.0 although less efficiently, since program 117.0 is essentially a combination of program 118.0 and an error analysis program. The same input will work in either program for booster simulation.

After a trajectory is shaped in the booster program the data is changed only to obtain print of the nominal trajectory and the state transition matrix on tape for performance assessment studies. The nominal may be run on either program 117.0 or 118.0

Much of this discussion is constructed about the basic framework of flow charts and equations. These diagrams serve to indicate logical flow connecting different functional blocks. The flow charts are arranged according to a hierarchical structure. The highest level, Level I, depicts the overall structure of the program. The blocks in Level I are each further described by the next level flow chart, Level II. This breakdown continues on to the lowest level where no further logic remains to be described and the basic equations are presented.

Section 2.0 presents the mathematical model for the programmed equations.

Section 3.0 presents the organization of the program with definitions of flow chart symbols, mathematical symbols, and coordinate systems.

Section 4.0 presents the computer program description of the input, output, initialization and computation blocks.

Section 5.0 presents the user's guide to the program with detailed explanation of input quantities and specific examples.

2.0 MATHEMATICAL MODEL

2.1 COORDINATE SYSTEMS

Most coordinate systems are right-handed cartesian coordinate systems with the right-hand convention for positive rotations. Exceptions will be noted.

2.1.1 Inertial Coordinate Systems

2.1.1.1 PCI Coordinates (X, Y, Z). The reference coordinate system is a planet-centered, inertial (PCI), cartesian coordinate system (X, Y, Z), oriented with the X-Y plane coincident with the planets equatorial plane, the Z-axis along the planet's positive axis of rotation. A reference launch point is automatically defined in the X-Z plane by inputs of launch point geographic coordinates. All forces, acceleration, velocities, positions, and reference trajectory planes are ultimately defined as vector componenents in this coordinate system.

2.1.1.2 Launch Level Coordinates (P_{I0}, Y_{A0}, R_{00}). The initial orientation of the reference body axes is defined by the inertial cartesian coordinate system (P_{I0}, Y_{A0}, R_{00}). The orientation of this coordinate system, with respect to the PCI system, is defined by input of the geographic coordinates of the launch point and the figure of the reference planet. The R_{00} axis is directed outward from the reference planet, along the direction of launch local vertical, the P_{I0}-Y_{A0} plane is parallel to the local level plane at the launch site. $-Y_{A0}$ is oriented in the direction defined by the input azimuth, ψ_p, measured positively in an eastward direction from true North. This inertial coordinate system is used as a reference for the gimbal angles α_1, α_2, and α_3 which describe the time-varying orientation of the missile body fixed axes (P_I, Y_A, R_0).

2.1.1.3 Velocity Profile Steering Coordinates (P, Q, R). The desired velocity profile of velocity profile guidance is defined in the (P, Q, R) inertial coordinate system. The (P, Q, R) coordinate system is geometrically related to the (P_{I0}, Y_{A0}, R_{00}) coordinate system by a positive rotation, θ_{cc}, about the P_{I0} axis. θ_{cc} is normally selected to align the R axis along an average thrust direction.

2.1.2 Noninertial Coordinate System

2.1.2.1 Geographic Coordinates. A reference launch point is defined by the input coordinates of geodetic or geographic latitude (λ_L), the geographic longitude, (λ_L), the equatorial radius, (a_e), and ellipticity of the reference spheroid, (ϵ), geoidal altitude (H_L), and launch point geoidal separation, (N_L). The local vertical at the launch site and the launch level plane are defined by additional inputs of the astronomic latitude and longitude (λ_{LA}, μ_{LA}), which account for local deflections of the vertical.

The sign convention for latitude and longitude is that North latitudes and East longitudes are positive. The geographic coordinate system is clearly not a cartesian coordinate system, but rather one that conforms to conventional planet-fixed geographic coordinates currently employed in Earth-mapping procedures.

2.1.2.2 (P_I, Y_A, R_0) Coordinates. The time-varying missile body attitude is defined by a reference missile body-fixed coordinate system (P_I, Y_A, R_0), and is employed to conveniently resolve missile aerodynamic and thrust forces. The initial orientation is coincident with the (P_{I0}, Y_{A0}, R_{00}) coordinate system, and its orientation at other times is specified by the time history of the missile body angular rates ω_{PI}, ω_{YA}, ω_{R0}.

2.1.2.3 Gimbal Angle Coordinates $(\alpha_1, \alpha_2, \alpha_3)$. An alternate representation of the missile time-varying body axis orientation in inertial space is given by the three Euler angles, α_1, α_2 and α_3 which are gimbal angles in a roll, yaw, and pitch order. These Euler angles are initially referenced to the (P_{I0}, Y_{A0}, R_{00}) coordinate system and are calculated by a trigonometric resolution of angles between the unit coordinate vectors of the (P_I, Y_A, R_0) and (P_{I0}, Y_{A0}, R_{00}) coordinate systems.

2.1.2.4 Trajectory Plane Coordinates $(\overline{U}_r, \overline{U}_v, \overline{U}_w)$. An instantaneous trajectory plane coordinate system is defined by an input unit vector $\overline{U}_{wj}$, which is normal to a desired trajectory plane, oriented so that $\overline{U}_r$, the unit instantaneous position vector, and $\overline{U}_v = \overline{U}_{wj} \times \overline{U}_r$ form the right-handed triad $(\overline{U}_r, \overline{U}_v, \overline{U}_{wj})$. The required velocity, $(\overline{V}_{REQ})$, is defined in this coordinate system for both modes of explicit guidance and for trim steering guidance. The index j refers to a particular guidance mode.

2.1.2.5 Planet-fixed Cylindrical Coordinates (r, θ, z). The program has the option of inputting aim point coordinates in either the reference PCI coordinate system or in planet-fixed cylindrical coordinates (r, θ, z). The z-axis is parallel to the Z-axis of the PCI system, r is normal to Z and θ is the angle measured in a positive sense about z from the meridian plane containing the missile at $t = 0$. Components of the aim point in this coordinate system at $t = 0$ are j-dependent and are designated by (C_{1j}, C_{2j}, C_{3j}).

2.2 ENVIRONMENTAL MODELS

2.2.1 Figure of Planets

A reference surface, corresponding to the physical surface of the planet, is defined as an oblate spheroid by inputs of equatorial radius, ellipticity and stellar angular rate. The figure is oriented with the Z-axis of the PCI coordinate along the reference axis of rotation.

2.2.2 Gravitational Potential

The gravitational potential of the reference planet is defined by a series representation in Legendre polynomials including the zero, second, third, and fourth harmonics. The gradient of this potential function defines the gravitational forces in the PCI coordinate system. Input parameters are GM, the universal gravitational constant times the mass of the planet; the second, third, and fourth harmonics in the conventional J, H, and D form; and the position vector PCI components.

2.2.3 Planetary Atmosphere

The atmosphere of a planet is assumed to be affixed to the planet, rotating with the planet's inertial angular rate. The atmosphere is defined to exist between the planet's surface and an input altitude above which atmospheric properties are considered to have insignificant affects upon the missile trajectory. The atmospheric models a properties thereof are discussed in greater detail in the following paragraphs.

2.2.3.1 Earth Atmosphere. The mathematical model defining the earth's atmospheric properties such as pressure, density, and speed of sound is based on the fundamental data and computational procedures used to generate "U.S. Standard Atmosphere 1962"[1]. Briefly, the atmosphere is assumed to be in hydrostatic equilibrium with respect to the Earth. A geopotential function due to mass attraction, including the second harmonic, and centrificial forces is calculated. This potential function, divided by the reference sea-level value of gravity, defines a "geopotential altitude." The variation of molecular-scale temperature with geopotential altitude is the fundamental defining property of the atmosphere, and all other properties are derived from this data. The data, as employed by COSEA in generating the "U.S. Standard Atmosphere 1962," defined the molecular-scale temperature as a series of connected line segments, linear in geopotential altitude up to 90 km and linear in geometric altitude above 90 km. The AC Electronics molecular-scale temperature model employs segments linear in geopotential altitude for all altitudes. The geopotential altitude range encompassed by the input data is -5 km to 630 km (-5 km to 700 km geometric altitude). The speed of sound has been held constant at geopotential altitudes greater than 88.743 km for purposes of mathematical continuity, although it is not a well-defined physical property of the atmosphere at higher altitudes.

2.2.3.2 Atmosphere Model of Planets other than Earth. Simple exponential models for atmospheric pressure and density are employed when near a planet other than Earth for powered flight trajectories. The speed of sound is defined as a constant.

2.3 MISSILE DYNAMICS

2.3.1 Propulsion Model

The total thrust force acting on the missile is calculated as the product of the specific impulse, (I_{sp}), the mass rate, ($\dot{M}$), and the sea-level value of gravity (g'_o). $g'_o I_{sp}$ is a phase-dependent tabulated function of atmospheric pressure. $\dot{M}$ is a phase-dependent

[1] See reference in Section 6.

tabulated function of the expended mass in each phase. A very arbitrary time-varying thrust function can be simulated with this model. One of the principal requirements of boost guidance is to compensate for propulsion variations, therefore, provision for perturbing either $g'_0 I_{sp}$ or $\dot{M}$ or both is included. Thus, total impulse and/or time duration of the impulse can be perturbed. For powered flight out of the atmosphere, the thrust force is directed along the missile roll axis, R_0.

During flight in the atmoshpere, a thrust force normal to the missile roll axis, (F_{MN}), is defined to be that which exactly nulls the moment due to the normal aerodynamic force. The remaining component of thrust force, (F_{MA}), is defined to be directed along the missile roll axis, R_0.

2.3.2 Aerodynamic Model

Aerodynamic forces on the missile are calculated in the body-fixed coordinates (P_I, Y_A, R_0) and utilize standard approximations appropriate to slender cylindrical bodies and small angles of attack. The axial drag force, (F_{DA}), is defined as the product of dynamic pressure, (q), the reference area, (S), and the drag coefficient, (C_D), a tabulated function of mach number. This drag force is directed along the negative R_0 axis.

The aerodynamic force normal to the roll axis, (F_{DN}), is in the plane defined by the missile roll axis, (R_0), and the missile velocity vector with respect to the planet's air mass. The normal force magnitude is defined by the product of the dynamic pressure, (q), reference area, (S), the normal drag coefficient, (C_N), which is a tabulated function of mach and angle of attack, and the angle of attack, (α) which is approximated by sin α in the simulation program. This normal aerodynamic force acts at a point along the missile longitudinal axis, termed the center of pressure, (C_p), which is defined by a tabular function of mach and angle of attack.

2.3.3 Equations of Motion

The inertial velocity and position of the center of mass of the missile in the PCI coordinate system are obtained by integrating the inertial accelerations defined by the sum of gravity and the specific force-to-mass ratio.

$$\ddot{\vec{X}} = \vec{g} + \frac{[\vec{F}_{MA} + \vec{F}_{MN} + \vec{F}_{DA} + \vec{F}_{DN}]}{M}$$

The initial conditions $\dot{\vec{X}}_0$ and $\vec{X}_0$ are computed by the initialization block for planet surface conditions, or are directly input in PCI components for initialization for other start conditions.

The time-varying missile mass, per phase, is expressed as

$$M(t) = M_o + \int_{t_o}^{t} \dot{M}\,(M_o - M(\tau))d\tau$$

where M_o, t_o, and $\dot{M}\,(M_o - M(\tau))$, a tabulated function of the expended mass, are phase-dependent values.

The orientation of the vehicle reference body axes is specified by three mutually orthogonal unit vectors in the PCI coordinates along the (P_I, Y_A, R_O) coordinate axes; namely, $\bar{P}_I$, $\bar{Y}_A$, $\bar{R}_0$. The vehicle angular rates in the P_I, Y_A, R_O coordinate system, denoted by ω_{PI}, ω_{YA}, and ω_{RO}, respectively, are employed to calculate $\bar{Y}_A$ and $\bar{R}_O$. These six first-order differential equations are integrated to define $\bar{Y}_A$ and R_O, $\bar{P}_I$ is defined by a vector crossproduct

$$\vec{Y}_A = \omega_{PI}\vec{R}_O - \omega_{RO}\vec{P}_I$$

$$\vec{R}_O = \omega_{YA}\vec{P}_I - \omega_{PI}\vec{Y}_A$$

$$\vec{P}_I = \vec{Y}_A \times \vec{R}_O$$

2.4 AUTOPILOT MODEL

The definition of body angular rates as a function of a guidance error signal is accomplished with a "perfect" autopilot model in which the angular rates about the (P_I, Y_A, R_O) coordinate axes are defined to be directly proportional to appropriately resolved guidance error signals. The guidance calculations of this simulation define gimbal angle commands. These gimbal angle commands are differenced with the computer gimbal angles to define a gimbal angle error which is resolved through the gimbal angle coordinate system to obtain an error signal in body axis coordinates. This error signal is limited in magnitude and the body angular rates are defined to be directly proportional to it, e.g., $\omega_{PI} = K\theta'_{EC}$, etc.

2.5 GUIDANCE SCHEMES

2.5.1 General Guidance and Steering Concepts

Each guidance scheme defines a particular trajectory objective, a mathematical statement of the objective, and a measure of the missile's error in attaining that objective. If, in the mathematical statement of the objective, the criteria is defined in terms of quantities measured by the missile guidance system, a closed-loop guidance system is said to exist. If the criteria is defined otherwise, it is referenced to as an open-loop guidance scheme; e.g., attitude vs time program. In most closed-loop guidance systems the guidance objective is stated in terms of a vector required velocity. A control is effected by "cross-product" steering so as to steer to reduce to zero the angular misalignment of missiles' actual velocity with respect to the required velocity and thus the achievement of the guidance requirement is reduced to determining the velocity magnitude error, as the angular errors are theoretically eliminated by the steering function.

2.5.2 β^* Guidance

The β^* guidance scheme is an open-loop trajectory design guidance scheme used to generate data required for the definition of one of the closed-loop guidance schemes, especially for that portion of the trajectory which is in the sensible atmosphere.

The objective of the β^* steering scheme is to obtain a constant $q\alpha$ during the initial portion of the atmospheric flight, followed by a zero angle of attack during the latter portion of the trajectory. This is accomplished by computing a pitch gimbal angle command of the form

$$\alpha_{3c} = \beta^* + \frac{K_\alpha}{q} = \tan^{-1}\left[\frac{\vec{V}_A \cdot \vec{Y}_{Ao}}{\vec{V}_A \cdot \vec{R}_{Oo}}\right] + \frac{K_\alpha}{q}$$

where V_A is the computed inertial velocity with respect to the planet-fixed atmosphere, K_α is the desired value of $q\alpha$ (usually set to zero in the latter portion of the trajectory).

A vertical rise phase preceeds this form of guidance so that $\vec{V}_A$ and q are physically (and numerically) well defined, nonzero values.

2.5.3 Velocity Profile Guidance

The velocity profile guidance scheme is a closed-loop guidance scheme. Its objective is to control the angle of attack of the missile while in the significant atmosphere by steering the missile along a nominal pitch plane velocity profile of accelerometer measured velocities. Data required to define this profile is obtained from a β^* guided reference trajectory. The pitch plane profile is defined in the (P, Q, R) coordinate system; the Q-R plane being the pitch plane. The nominal value of the "normal" velocity, $\dot{Q}_a$ is curve fit by a cubic polynomial in the nominal "tangential" velocity, $\dot{R}_a$. The nominal value of the pitch gimbal command, α_{3c}, is also curve fit by a cubic polynomial in $\dot{R}_a$. The closed-loop pitch gimbal angle command is then defined to be of the form

$$\alpha_{3c} = \alpha_{3c\,(\text{nominal})} + K\,[\dot{Q}_a - \sum_{j=1}^{3} K_j \dot{R}_a^j]$$

where

$$\alpha_{3c\,(\text{nominal})} = \sum_{j=1}^{3} C_i \dot{R}_a^j$$

and $\dot{R}_a$, $\dot{Q}_a$ are the measured values of the thrust velocity components. The thrust velocity component normal to the pitch plane is nulled. Velocity profile steering is initiated following a vertical rise phase. The thrust velocities, viz., the integrated accelerometer outputs, are utilized for steering to permit the steering loop to operate independently of the navigation equations during that portion of the trajectory in the effective atmosphere where this mode of guidance is generally employed.

2.5.4 Explicit Guidance with Time of Flight Constraint

The explicit guidance scheme defined in this simulation is employed above the effective atmosphere where no atmospheric steering constraints need be considered. The explicit guidance equations define a unique ellipse between the present position of the vehicle and a defined target point. The ellipse is unique because a time of flight between the launch and the interception of the target point is specified. The time constraint is introduced into the Keplarian explicit equations by the following relationship, defining the semi-latus rectum, p, of the ellipse.

$$p_{i+1} = p_i + [K_{13} + K_{14}\, p_i][T_A - (t_i + T_i)]$$

where K_{13}, K_{14} are curve fit constraints derived from a reference trajectory. T_A is the specified time of flight, T_i is the current estimate of the time of flight from Lambert's equation, and t_i is the time since launch.

After a few iterations the orbital parameters are in a reasonably close neighborhood of the desired orbital parameters and p_i is estimated by a nonlinear logic which selects the value determined by either a linear extrapolation, the previous value, or redetermined by the above relationship. This logic is based upon the preceding changes in (p_i) and in $(t_i + T_i)$.

For each major guidance cycle the position of the missile is determined and a velocity, V_{REQ}, is determined as the instantaneous elliptical velocity which satisfies the fixed total time of flight constraint

$$\vec{V}_{REQ} = K\, [(\frac{e \sin v}{\sqrt{p}})\, \vec{U}_r + \frac{\sqrt{p}}{r}\, \vec{U}_v]$$

where K is gravitational constant (GM); e, the eccentricity; v, the true anomaly of vehicle; r, the radial position; $\vec{U}_r$ and $\vec{U}_v$ are orbit plane unit vectors.

The velocity to be gained is defined as

$$\vec{V}_g = \vec{V}_{REQ} - \vec{V}$$

$\bar{F}_{MN}$	$[F_{MNX}\bar{i} + F_{MNY}\bar{j} + F_{MNZ}\bar{k}]$ main engine normal thrust vector.
g	sea level value of gravitational acceleration
g'_o	sea level value of earth's gravitational acceleration
g_X, g_Y, g_Z	components of gravity measured in a planet centered coordinate system
GM	universal gravitational constant times mass of planet
GM'	universal gravitational constant times mass of earth
$\bar{g}$	$[g_X\bar{i} + g_Y\bar{j} + g_Z\bar{k}]$ gravity vector
h_c	height of the sensible atmosphere
H	altitude above sea level at present position
H_g	coefficient of the third harmonic of the earth's gravitational potential
H^*	geopotential altitude
$H^{*'}_{(n)}$	geopotential altitude at the bottom of the n^{th} molecular temperature segment
H_L	height of launchsite above sea level
i	subscript denoting present value
$i - 1$	subscript denoting past value
$\bar{i}, \bar{j}, \bar{k}$	unit vectors along the X, Y, Z PCI axes
I_{SP}	specific impulse
j	subscript denoting "Jacker" trigger used to determine guidance logic

J_g	coefficient of the second harmonic of the earth's gravitational potential
k	subscript used to denote the different phases in HS1 and HS2
$k'_{(n)}$	slope of the n^{th} segment of the molecular weight vs. geopotential altitude approximation
k_2, k_3	steering gains used in $\overline{V}_g$ steering to compute commanded gimbal angles (j dependent)
k_4, k_5	steering gains used in $\overline{V}_g$ steering to compute commanded gimbal angles (j dependent)
K	the universal gravitational constant, $\sqrt{GM}$ (j dependent)
K_A	steering gains used during trim guidance (j dependent)
K_H	steering gain used during trim guidance (j dependent)
K_r	1/K (j dependent)
K_{REV}	thrust coefficient
K_{VG}	V_g coefficient
K_{13}	semilatus rectum correction coefficient (j dependent)
K_{14}	semilatus rectum correction coefficient (j dependent)
K_{50}, K_{51}	$\dot{P}, \dot{Q}, \dot{R}$ steering gains for α_{1c}
K_{52}	$\dot{P}, \dot{Q}, \dot{R}$ steering gains for α_{3c}
$K_{53} - K_{61}$	curve fit constants for $\dot{P}, \dot{Q}, \dot{R}$ steering
K_α	constant for criteria used in pitch over with β^* guidance between T_1 and T_3

C_g	location of the center of mass
C_n	$(n = 0, 1, 2, 3, 4)$ coefficients of the zonal harmonics of the gravitational potential function
C_p	location of the center of pressure
C_A	axial drag coefficient
C_N	normal drag coefficient
$\bar{C}_S'$	speed of sound constant
C_1, C_2, C_3	normalized cylindrical offset coordinates of the aiming point used where aiming point is rotating with the planet. (BLK HA4B) (j dependent)
C_2'	coefficient of second harmonic of earth's gravitational potential
d_1	distance between the center of gravity and the center of pressure
D_g	coefficient of the fourth harmonic of the earth's gravitational potential
e	eccentricity of free flight ellipse
E	eccentric anomaly at the vehicle's present position
E_A	eccentric anomaly at aiming point
E_G	location of the engine gimbal
F_{AIM}	trigger for determining aiming point computation, HA4A or HA4B. (j dependent) $(F_{AIM_j} = 0 \rightarrow \text{HA4A}, \; F_{AIM_j} = 1 \rightarrow \text{HA4B})$

F_{CON} trigger for trajectory constraint at the target of constant time of arrival, HA6A or constant flight path angle, HA6B. (j dependent) ($F_{CON_j} = 0 \rightarrow$ HA6A, $F_{CON_j} = 1 \rightarrow$ HA6B)

F_{DA} magnitude of the axial drag force

F_{DAX}, F_{DAY}, F_{DAZ} components of the axial drag force vector

F_{DN} magnitude of the normal drag force

F_{DNZ}, F_{DNY}, F_{DNZ} components of the normal drag force vector

F_{DNPI} projection of $\bar{F}_{DN}$ on pitch axis

F_{DNYA} projection of $\bar{F}_{DN}$ on yaw axis

F_{MAX}, F_{MAY}, F_{MAZ} components of the main engine axial thrust vector, $\bar{F}_{MA}$

F_{MNX}, F_{MNY}, F_{MNZ} components of the main engine normal thrust vector, $\bar{F}_{MN}$

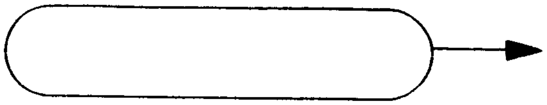
F_N thrust force on the vehicle

F_{TRIM} trigger for determining whether cutoff of phase based on velocity to be gained is to be used. (j dependent) ($F_{TRIMj} = 0$ cutoff on V_g, $F_{TRIMj} = 1$ no cutoff on V_g)

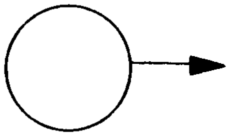
$\bar{F}_{DA}$ $[F_{DAX}\bar{i} + F_{DAY}\bar{j} + F_{DAZ}\bar{k}]$, axial drag force vector

$\bar{F}_{DN}$ $[F_{DNX}\bar{i} + F_{DNY}\bar{j} + F_{DNZ}\bar{k}]$, normal drag force vector.

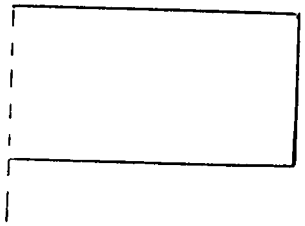
$\bar{F}_{MA}$ $[F_{MAX}\bar{i} + F_{MAY}\bar{j} + F_{MAZ}\bar{k}]$ main engine axial thrust vector



Connector used on Level II flow charts to indicate entry source or exit destination.



Connector used on Level II and below charts.



Summary of all quantities required in computations of flow chart on which this symbol appears or, alternatively, summary of all quantities computed in this flow chart which are required in other operations.

3.3 DEFINITION OF MATHEMATICAL SYMBOLS

a_e	equatorial radius of the planet
a'	equatorial radius of the reference geoid
a_T	thrust acceleration
a_X, a_Y, a_Z	components of thrust and drag acceleration measured in a planet centered coordinate system
$\bar{a}$	$[a_X \bar{i} + a_Y \bar{j} + a_Z \bar{k}]$, acceleration vector of thrust and drag measured in a planet centered coordinate system
A_{RC}, B_{RC}, C_{RC}	curve fit constants for radial distance as a function of time used in trim guidance. (j dependent)
A_{RD}, B_{RD}, C_{RD}	curve fit constants for radial rate as a function of time used in trim guidance. (j dependent)
A_{RV}, B_{RV}, C_{RV}	curve fit constants for tangential velocity as a function of time used in trim guidance. (j dependent)
$A_\theta, B_\theta, C_\theta$	curve fit constants for reducing steering errors (j dependent)
c	cosine of the angle between the present vehicle position and the aiming point measured at the center of the planet
C	speed of sound
C'_{a1}	change of units coefficient
C'_{a2}	change of units coefficient
C'_{a3}	change of units coefficient
C'_{a4}	change of units coefficient
C'_{a5}	change of units coefficient

3.0 ORGANIZATION OF THE PROGRAM

3.1 SCHEMA FOR FLOW CHART PRESENTATION

The flow charts are arranged according to "levels". The Level I flow chart provides the most general description since it depicts the overall program. Each functional block is further described by lower level flow charts down to the lowest level where sets of equations and/or logic decisions are used.

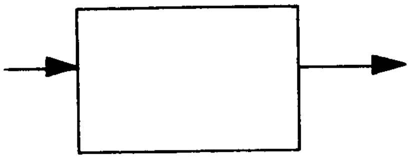
In the Level I flow charts, the titles in the blocks are intended to be suggestive of the nature of the role to be performed within the block. Those functions that are to be performed in the basic computational cycle are designated by Roman numerals and those functions that play a passive role are designated by Arabic symbols.

In Level II flow charts and below, the quantities that are required for all logical and computational operations within this block are stated in the input and output on the chart. The input quantities are differentiated into stored or computed values. The stored values are those initially put in by the user and the computed values are those determined in other portions of the program. The output quantities indicated are those generated in this block for use in other blocks or for printout.

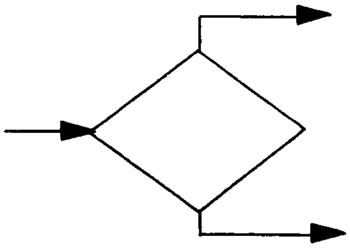
The lowest level low charts are those that do not require further explanation or are a set of equations with the input and output quantities specified.

3.2 DEFINITION OF FLOW CHART SYMBOLS

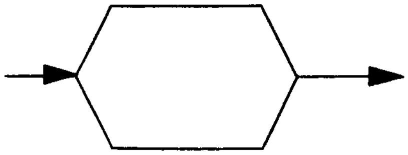
The following symbols represent the only ones that are used in the flow charts presented below.



Set of equations that is to be described further by additional flow charts or by equations.



Logical decision.



Operations that are predefined.

At each guidance index change, (j), during the explicit guidance mode, a reference set of gimbal angles (α_{1vg}, α_{3vg}), corresponding to the alignment of the roll vector along V_g are computed. A reference unit vector along $\bar{V}_g$, $\bar{U}_{vgj}$, is also computed at this time. A cross-product steering error function is defined by

$$\overrightarrow{\Delta R} = \vec{V}_g \times \vec{U}_{vgj}$$

and is resolved into appropriate gimbal angle components.

$$\Delta\alpha_{1c} = -K_{2j} (\overrightarrow{\Delta R} \cdot \vec{R}_{Oo}) - K_{3j}\, \Sigma(\overrightarrow{\Delta R} \cdot \vec{R}_{Oo})\, \Delta t$$

$$\Delta\alpha_{2c} = 0$$

$$\Delta\alpha_{3c} = -K_{4j} (\overrightarrow{\Delta R} \cdot \vec{P}'_I) - K_{5j}\, \Sigma(\overrightarrow{\Delta R} \cdot \vec{P}'_I)\, \Delta t_i$$

where P'_I is along the current pitch axis of the missile.

The gimbal angle commands are

$$\alpha_{1c} = \Delta\alpha_{1c} + \alpha_{1vgj} + \vec{\theta}_K \cdot \vec{R}_{Oo}$$

$$\alpha_{2c} = 0$$

$$\alpha_{3c} = -\Delta\alpha_{3c} + \alpha_{3vgj} + \vec{\theta}_K \cdot \vec{P}'_I$$

where
$$\vec{\theta}_K = F(V_g)\, [\vec{U}_r \times \vec{U}_{vgj}]$$

$F(V_g)$ is a curve fit function of V_g which offsets the gimbal angle commands to compensate for gravitational and centrificial acceleration. Powered flight is terminated when V_g is zero. This is an iterated condition which is calculated to a precision specified by an input ϵ, and is not restricted to a time specified by a multiple of either major or minor computational cycles.

2.5.5 Explicit Guidance with Velocity Direction Constraints

This formulation of explicit guidance is perhaps the most useful form of explicit guidance, as the terminal direction of the velocity vector at the target point is a very convenient parameter for direct-injection-into-orbit trajectories. The guidance model is exactly the same as the explicit time of flight constraint guidance except that the semilatus rectum, p, is defined using the desired velocity angle, $K_{\gamma j}$, and the calculated angle, γ_A.

$$\gamma_A = \sin^{-1} \left[\frac{e \sin V_A}{\sqrt{1 + 2 (e \cos V_A) + e^2}} \right]$$

$$p_{i+1} = p_i + [K_{13} + K_{14}\, s][K_{\gamma j} - \gamma_A]$$

where s is the sine of the range angle between the present position and the target point; K_{13}, K_{14} are curve fit constants, taken from a reference trajectory. A nonlinear logic is again employed to select p_{i+1}; either p_i, a linear extrapolated value, or the value defined by the equation above is chosen, depending upon the prior values p_i and γ_A. The same steering logic as described in 2.5.4 is employed with this guidance mode.

2.5.6 Trim Steering Guidance

Trim steering guidance is designed to guide a vehicle into a specified plane with specifications on the following end conditions:

a. desired radial position r_c

b. desired radial speed $\dot{r}_c$

c. desired "tangential" speed $(r\dot{v}_c)$ where $\dot{v}$ is the time derivative of the true anomaly.

The quantities r_c, $\dot{r}_c$, and $(r\dot{v}_c)$ are curve fit second-order polynomials in time, measured from the initiation of trim steering.

The orbit plane is specified by an input vector normal to the desired orbital plane, $\vec{U}_{wj}$, and an inplane unit vector is defined by the cross-product relationship $\vec{U}_v = \vec{U}_{wj} \times \vec{U}_r$.

$\vec{V}_{REQ}$ is defined as

$$\vec{V}_{REQ} = \dot{r}_c \vec{U}_r + (rv)_c \vec{U}_v + K_{Aj} [\dot{r}_c - \dot{r} + K_{Hj} (r_c - r)] \vec{U}_r$$

The difference $\vec{V}_{REQ} - \vec{V}$ is defined as $\vec{V}_g$ and the same V_g cross-product steering scheme that was employed in the explicit guidance schemes is used for this guidance scheme.

The validity of the time polynomial expressions for r_c, $\dot{r}_c$, and $(r\dot{v}_c)$ is restricted to a reasonably small neighborhood of the desired injection conditions, insofar as the simultaneous realization of all three of the conditions will result in the desired orbit.

For most applications of this steering, it is initiated when the vehicle radial position is close to the desired value, and V_{REQ} is essentially along U_v.

$K_{\alpha L}$	lower bound for α_{1vg} (k dependent)
$K_{\alpha u}$	upper bound for α_{1vg} (k dependent)
K_{γ}	desired flight path angle at the aim point (j dependent)
$K_{\gamma L}$	lower bound for α_{3vg} (k dependent)
$K_{\gamma u}$	upper bound for α_{3vg} (k dependent)
K_{θ}	pitch position gain
$K_{\dot{\theta}}$	pitch rate gain
K_{φ}	roll position gain
$K_{\dot{\varphi}}$	roll rate gain
K_{ψ}	yaw position gain
$K_{\dot{\psi}}$	yaw rate gain
M	present vehicle mass
M_o	initial vehicle mass
M'_o	sea level value of mean molecular weight of air
$\dot{M}$	propulsion mass flow rate
$\dot{M}_{T6}$	perturbation in mass flow rate which is used from T6 or T7 to the end of the phase
$\dot{M}_{T8}$	perturbation in mass flow rate which is used from T8 or T9 to the end of the phase
N_L	difference between reference geoid and actual sea level
p	semilatus rectum for the free flight ellipse (j dependent)

P	atmospheric pressure
$P'_{(m)}$	atmospheric pressure
P_n	(n = 0,1,2,3,4,5) legendre polynomials associated with the gravity expansion
$P'_{(n)}$	atmospheric pressure at base of n^{th} approximation segment
PCI	planet-centered-inertial, an acronym referring to the X, Y, Z coordinate system which is a right-handed, orthogonal, irrotational coordinate system with origin at the positive Z-axis along the North polar axis, X and Y in the equatorial plane
P_{FLO}	perturbation factor in mass flow rate
P_{ISP}	perturbation factor in I_{SP}
P_{IX}, P_{IY}, P_{IZ}	components of the unit vector $\overline{P}_I$
$P_{IX_0}, P_{IY_0}, P_{IZ_0}$	components of the unit vector $\overline{P}_{I_0}$
P_E	perturbation factor for east wind
P_N	perturbation factor for north wind
P'_2	second order legendre polynomial of gravitational potential
$\dot{P}_a, \dot{Q}_a, \dot{R}_a$	components of velocity due to thrust in a computational coordinate system which is the $\overline{R}_0$, $\overline{P}_{I_0}$, $\overline{Y}_{A_0}$ coordinate system rotated about the $\overline{P}_{I_0}$ by $\overline{\theta}_{cc}$
$\overline{P}_I$	unit vector along the pitch axis
P_{I_0}	unit vector along the initial pitch axis, $[P_{IX_0} \overline{i} + P_{IY_0} \overline{j} + P_{IZ_0} \overline{k}]$

q	incompressible dynamic pressure
r	distance from center of the planet to the present position
r'	distance of the point (X, Y, Z) from earth's center
r_c	desired commanded radial distance for the vehicle
r_A	radial distance from the planet center to the target point (j dependent)
r_{EX}	radial distance criterion for beginning of explicit guidance (j dependent)
r_{TR}	radial distance criterion for beginning of trim guidance (j dependent)
$\dot{r}$	radial velocity
$\dot{r}_c$	desired commanded radial velocity for the vehicle
$\ddot{r}$	radial acceleration
$\bar{r}$	$[X\bar{i} + Y\bar{j} + Z\bar{k}]$ radius vector to present position
R_{SL}	sea level value of r
R_{SLL}	sea level value of r at launch site
R_{OX}, R_{OY}, R_{OZ}	components of the unit vector $\bar{R}_0$
R_{OX_0}, R_{OY_0}, R_{OZ_0}	components of the unit vector $\bar{R}_{0_0}$
R_{STOP1}	$\dot{R}_a$ criterion for computing T_1 with $\dot{P}$, $\dot{Q}$, $\dot{R}$ guidance
R_{STOP4}	$\dot{R}_a$ criterion for computing T_4 with $\dot{P}$, $\dot{Q}$, $\dot{R}$ guidance

R_V mach number

$R^{*\prime}$ universal gas constant

$\bar{R}_A$ target coordinates in inertial coordinates for explicit guidance (Block HA4A) (j dependent)

$\bar{R}_0$ $[R_{0X}\bar{i} + R_{0Y}\bar{j} + R_{0Z}\bar{k}]$ unit vector along the roll axis

$\bar{R}_{0_0}$ $[R_{0X_0}\bar{i} + R_{0Y_0}\bar{j} + R_{0Z_0}\bar{k}]$ unit vector along the initial roll axis

s sine of the angle between the present vehicle position and the aiming point measured at the center of the planet

S effective drag area during boost

t current value of time

t_m time at the beginning of trim guidance

T time of free flight

T_A input constant specifying the time of arrival when constant time of flight constraint is used (j dependent)

T_0 initial estimate for time of flight for use with planet fixed target aiming point (Block HA4B) (j dependent)

T_1 end of zero attitude steering and beginning of pitchover phase

T_3 end of pitchover and beginning of zero angle of attack guidance

T_4 end of zero angle of attack guidance and beginning of explicit guidance

T_5	beginning of trim guidance in phases 4 or 5
T_6	beginning of explicit guidance in phase 7
T_7	beginning of trim guidance in phase 7
T_8	beginning of explicit guidance in phase 9
T_9	beginning of trim guidance in phase 9
$T'_{(n)}$	molecular scale temperature at the base of the n^{th} approximation segment
U'_f	total gravitational and centrifugal potential
U'_o	gravitational and centrifugal potential with respect to the potential at surface of geoid
$\overline{U}_r$	unit vector along radius vector at present position
$\overline{U}_v$	unit vector perpendicular to $\overline{U}_r$ in trajectory plane
$\overline{U}_w$	unit vector perpendicular to the nominal trajectory plane (j dependent)
$\overline{U}_A$	unit vector pointing to the aim point (j dependent)
$\overline{U}_{EX}$	unit vector criterion for beginning of explicit guidance (j dependent)
$\overline{U}_{TR}$	unit vector criterion for beginning of trim guidance (j dependent)
v	true anomaly of vehicle
v_A	true anomaly at aim point
$\dot{v}$	rate of change of the true anomaly

V	magnitude of present velocity
V_{gX}, V_{gY}, V_{gZ}	components of the velocity-to-be-gained vector
V_A	magnitude of present vehicle velocity with respect to air
V_{REQ}	magnitude of the present required velocity
V_{REQX}, V_{REQY}, V_{REQZ}	components of the required velocity in PCI coordinates
$\bar{V}$	$[\dot{X}\bar{i} + \dot{Y}\bar{j} + \dot{Z}\bar{k}]$ vector of present vehicle velocity
$\bar{V}_a$	$[\int a_X \bar{i} + \int a_Y \bar{j} + \int a_Z \bar{k}]$ vector of integrated accelerometer output
$\bar{V}_g$	velocity-to-be-gained vector
$\bar{V}_{gj}$	initial value of $\bar{V}_g$ when guidance logic (j) is changed
$\bar{V}_A$	$[\dot{X}_A \bar{i} + \dot{Y}_A \bar{j} + \dot{Z}_A \bar{k}]$ vector of present vehicle velocity with respect to air
$\bar{V}_{REQ}$	vector of the present required velocity
W_E	velocity of East wind (tabular value)
W_N	velocity of North wind (tabular value)
V'_o	$M'_o/R^{*\prime}$
X, Y, Z	planet centered inertial (PCI) components of the vehicle
X_A, Y_A, Z_A	components of the aim point in PCI coordinates (j dependent)
X_0, Y_0, Z_0	initial position components of vehicle in PCI coordinates

$\dot{X}, \dot{Y}, \dot{Z}$	planet centered inertial (PCI) components of the vehicle velocity
$\dot{X}_A, \dot{Y}_A, \dot{Z}_A$	components of the vehicle velocity with respect to air
$\dot{X}_o, \dot{Y}_o, \dot{Z}_o$	initial velocity components of vehicle in PCI coordinates
$\ddot{X}, \ddot{Y}, \ddot{Z}$	planet centered inertial (PCI) components of vehicle acceleration corrected for gravity
Y_{AX}, Y_{AY}, Y_{AZ}	components of the unit vector $\bar{Y}_A$ in PCI coordinates
$Y_{AX_0}, Y_{AY_0}, Y_{AZ_0}$	components of the unit vector $\bar{Y}_{A_0}$ in PCI coordinates
$\bar{Y}_A$	unit vector along the yaw axis
$\bar{Y}_{A_0}$	$[Y_{AX_0}\bar{i} + Y_{AY_0}\bar{j} + Y_{AZ_0}\bar{k}]$ unit vector along the initial yaw axis

α	angle of attack
α_{PI}	angle of attack in pitch plane
α_{YA}	angle of attack in yaw plane
$\alpha_1, \alpha_2, \alpha_3$	inner, middle, and outer gimbal angles
$\alpha_{1c}, \alpha_{2c}, \alpha_{3c}$	commanded inner, middle, and outer gimbal angles
$\alpha_{1vg}, \alpha_{2vg}, \alpha_{3vg}$	inner, middle, and outer gimbal angles to align the roll axis of the vehicle along the $\vec{U}_{vg_j}$
$\alpha_{1_\epsilon}, \alpha_{2_\epsilon}, \alpha_{3_\epsilon}$	inner, middle, and outer gimbal angle errors
β^*	pitch angle command for zero angle of attack trajectory profile
β	angle between the radius vector, $\bar{r}$, to the vehicle and the vehicle velocity vector with respect to air, $\bar{V}_A$
β_i	atmospheric density decay parameter
β_j	phase dependent parameter defining the maximum angular rate of the vehicle
γ	angle between the radius vector $\bar{r}$, and the vehicle velocity vector, $\bar{V}$
γ'	ratio of specific heat of air at constant pressure to that at constant volume
γ_A	flight path angle at aim point
δ	effective angle thrust acts through due to normal forces on the vehicle
δt	minor cycle length phase dependent dynamics block integration step size
Δt	major computational cycle length, phase dependent guidance block integration step size

Δt_{go}	time to go
ΔP	phase dependent print interval
$\Delta P'$	tape print interval
Δv	time from beginning of trim guidance
ΔR_X, ΔR_Y, ΔR_Z	components of $\Delta\vec{R}$ in PCI coordinates
$\Delta\overline{R}$	vector used in $\vec{V}_g$ so that changes in $\vec{V}_g$ are in magnitude only
$\Delta\alpha_{1c}, \Delta\alpha_{2c}, \Delta\alpha_{3c}$	incremental gimbal angle commands to align $\vec{V}_g$ along $\vec{U}_{vg_j}$
ϵ	ellipticity or flattening of the reference geoid
ϵ_p	criterion used to determine computation of p_{i+1} in blocks HA6A and HA6B
ϵ_s	criterion used to determine computation of (e sin ν) in block HA5
ϵ_u	criterion used to determine computation of $\vec{U}_v$ in block HA5
ϵ_{vg}	criterion for velocity to be gained cutoff
ϵ_T	criterion used to determine computation of p_{i+1} in block HA6A
ϵ_{TR}	criterion for reaching radial distance condition for a critical time point
ϵ_{TRA}	criterion for reaching a position in determination of a critical time point
ϵ_γ	criterion used for determining computation of p_{i+1} in block HA6B
θ_{cc}	angle between computational coordinates for $\dot{P}, \dot{Q}, \dot{R}$ steering and launch level coordinates measured in the X-Z plane
θ'_{EC}	error in pitch attitude

$\bar{\theta}_k$	a vector used to align the vehicle so that the thrust acceleration along the roll axis causes the vehicle to accelerate along $\vec{V}_g$
λ	geodetic latitude of the vehicle
λ_L	geodetic latitude of launch site
λ_{LA}	astronomical latitude at launch site
λ_{LC}	geocentric latitude at launch site
μ	current geodetic or geocentric longitude of the vehicle
μ_L	geodetic or geocentric longitude at launch site
μ_{LA}	astronomical longitude at launch site
μ_{LD}	$\mu_{LA} - \mu_L = \mu_{LD}$
π	$= 3.1415926536$
ρ	density of the atmosphere
ρ_o	mean sea level atmospheric density
ρ'_m	density of the air
τ_5	duration of phase 6
τ_6	criterion for the determination of T6
τ'_6	criterion for the determination of T7
τ_7	duration of phase 8
τ_8	criterion for the determination of T8
τ'_8	criterion for the determination of T9
Φ'_{EC}	error in roll attitude
ψ'_{EC}	error in yaw attitude

ψ_p	azimuth angle of the pitch plane with respect to North
ω_{PI}	angular rate of vehicle about the pitch axis
ω_{RO}	angular rate of vehicle about the roll axis
ω_{YA}	angular rate of vehicle about the yaw axis
Ω	planet angular stellar rate
Ω'	rotational velocity of the earth
$\vec{\omega}$	angular rate vector for the vehicle

ATMOPT = EARFLG

BETLOG guidance blocks for zero angle of attack guidance used from T_o to T_4

BETSTS block to compute β^* and gimbal angle commands

BETTRG trigger specifying whether the phase is before T3 or after T3 for BETLOG

COMPQR routine to compute $\dot{P}_a, \dot{Q}_a, \dot{R}_a$ and Δt_{go}

EARFLG = ATMOPT trigger specifying desired atmospheric routine
(ATMOPT = 0 → earth atmospheric routine
ATMOPT = 1 → planetary atmospheric routine)

EATM earth atmospheric routine based on "U. S. Standard Atmosphere, 1962" as defined by NASA, USAF, and U. S. W. B.

Edit equations which compute parameters used for descriptive purposes

ENDSTG routine used to set up the next phase

HSTART routine used to start the program at the beginning of a phase other than 1

INTG integration routine which integrates the derivatives computed in the dynamics blocks from t to t+1 in steps of δt

JACKER guidance blocks used after T4

LLLGIC guidance logic, phase change and commanded attitudes are computed

MBS mass at the beginning of phase

MES mass at the end of phase

NOR6PT option for state transition matrix generation
NOR6PT = 1. → nominal only; NOR6PT = 0 → compute STM

OUTIN trigger to define whether the vehicle is in or out of the atmosphere
(OUTIN = 0 → vehicle in atmosphere
OUTIN = 1 → vehicle out of atmosphere)

PATM	planetary atmospheric routine
PHASE	interval of the simulation during which phase dependent parameters are constant
PRTOUT	routine used to partially control the program and to determine if printout is required
PRTTRG	trigger used in PRTOUT which has the following values 0 - normal test for print and time adjustment 1 - special print point 2 - end of phase
PSITH	routine to compute gimbal angle commands from $\dot{P}_a, \dot{Q}_a, \dot{R}_a$
Q-BIAS	a time-varying integer used as a bias for t. Its purpose is to allow t to be expressed in more significant figures when a critical time point is reached.
QSANDE	time for next print based on ΔP using QBIAS for a bias
SEPXXX	criterion for achieving end of phase
SETSTG	routine used to set up the phase just being entered
STBEGS	beginning phase number
SIENTB	dynamics blocks, position, velocity, and attitude are computed
time	defined by: $t = \text{time} + \text{QBIAS}$ ($\text{time} < 2\Delta t$)
TES	time at the end of a phase
TIMEST	time to go values based on mass cutoff, velocity-to-be-gained cutoff, and critical time points
t+1	the sum of t+1 and QBIAS is the t at the next integration point
WFLAG	trigger to denote whether winds are desired in the computation of $\overline{V}_A$ (WFLAG = 0 → no winds desired)

XPRTXX	data print
XSTAGE	phase number

3.4 COORDINATE SYSTEMS

The basic coordinate system in program 118.0 is the Planet Centered Inertial (PCI) coordinate system (X, Y, Z). The X and Y axes are in the earth's equatorial plane and Z is along the direction of the earth's positive rotation. When starting in phase 1, the vehicle's initial position is in the X-Z plane, specified by inputs of launch altitude, launch latitude, and planet radius. When starting in phases other than phase 1, direct input of the PCI coordinates is employed and the vehicle is initialized at an arbitrary point. The unit vectors $\vec{i}$, $\vec{j}$, and $\vec{k}$ lie along the X, Y and Z axes, respectively.

Another basic coordinate system in the program is the reference body axis coordinate system $(P_{I_O}, Y_{A_O}, R_{O_O})$. This cartesian coordinate system is oriented by inputs of latitude, λ_{LA}, longitude (μLD), and azimuth (ψ_p) as computed below and illustrated in Figure 3.4.1.

$$\begin{bmatrix} \vec{R}_{O_o} \\ \vec{P}_{I_o} \\ \vec{Y}_{A_o} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_p & -\sin\psi_p \\ 0 & \sin\psi_p & \cos\psi_p \end{bmatrix} \begin{bmatrix} \cos\lambda_{LA} & 0 & -\sin\lambda_{LA} \\ 0 & 1 & 0 \\ \sin\lambda_{LA} & 0 & \cos\lambda_{LA} \end{bmatrix} \begin{bmatrix} \cos\mu_{LD} & \sin\mu_{LD} & 0 \\ \sin\mu_{LD} & -\cos\mu_{LD} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}$$

There is also a body axis coordinate system (P_I, Y_A, R_O) which rotate with the vehicle and lie along the vehicle axes. When beginning in phase 1, the (P_I, Y_A, R_O) coordinate system is initially aligned along the $(P_{I_O}, Y_{A_O}, R_{O_O})$ coordinate system. When beginning in phases 2-9, the (P_I, Y_A, R_O) coordinate system is initially aligned with respect to the $(P_{I_O}, Y_{A_O}, R_{O_O})$ coordinate system through the set of gimbal angles $(\alpha_1, \alpha_2, \alpha_3)$ which may be input into the program. The relationship between the two coordinate systems can be computed from the following equation and is illustrated in Figure 3.4.2.

$$\begin{bmatrix} \vec{P}_I \\ \vec{Y}_A \\ \vec{R}_O \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_3 & \sin\alpha_3 \\ 0 & -\sin\alpha_3 & \cos\alpha_3 \end{bmatrix} \begin{bmatrix} \cos\alpha_2 & 0 & -\sin\alpha_2 \\ 0 & 1 & 0 \\ \sin\alpha_2 & 0 & \cos\alpha_2 \end{bmatrix} \begin{bmatrix} \cos\alpha_1 & \sin\alpha_1 & 0 \\ -\sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{P}_{I_o} \\ \vec{Y}_{A_o} \\ \vec{R}_{O_o} \end{bmatrix}$$

In the velocity polynomial guidance (0P0123 = 1.) there is a computational coordinate system (P, Q, R) into which the velocity is resolved. The (P, Q, R) coordinate system is related to the $(P_{I_O}, Y_{A_O}, R_{O_O})$ coordinate system a θ_{cc} rotation about the P_{I_O} axis as presented in the following equation and illustrated in Figure 3.4.3.

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{cc} & \sin \theta_{cc} \\ 0 & -\sin \theta_{cc} & \cos \theta_{cc} \end{bmatrix} \begin{bmatrix} P_{I_o} \\ Y_{A_o} \\ R_{O_o} \end{bmatrix}$$

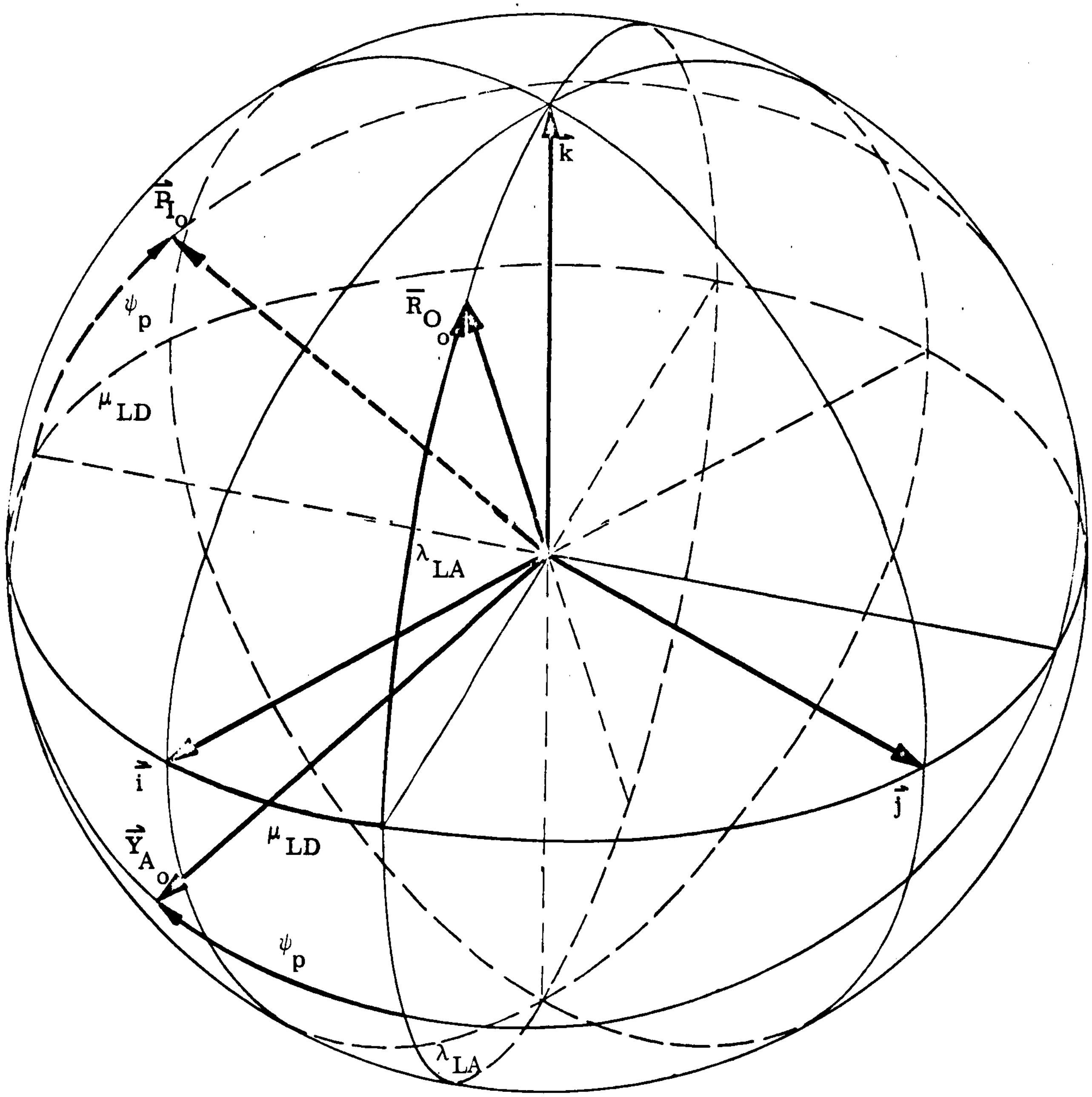


Figure 3.4.1. Initial Reference Body Axes

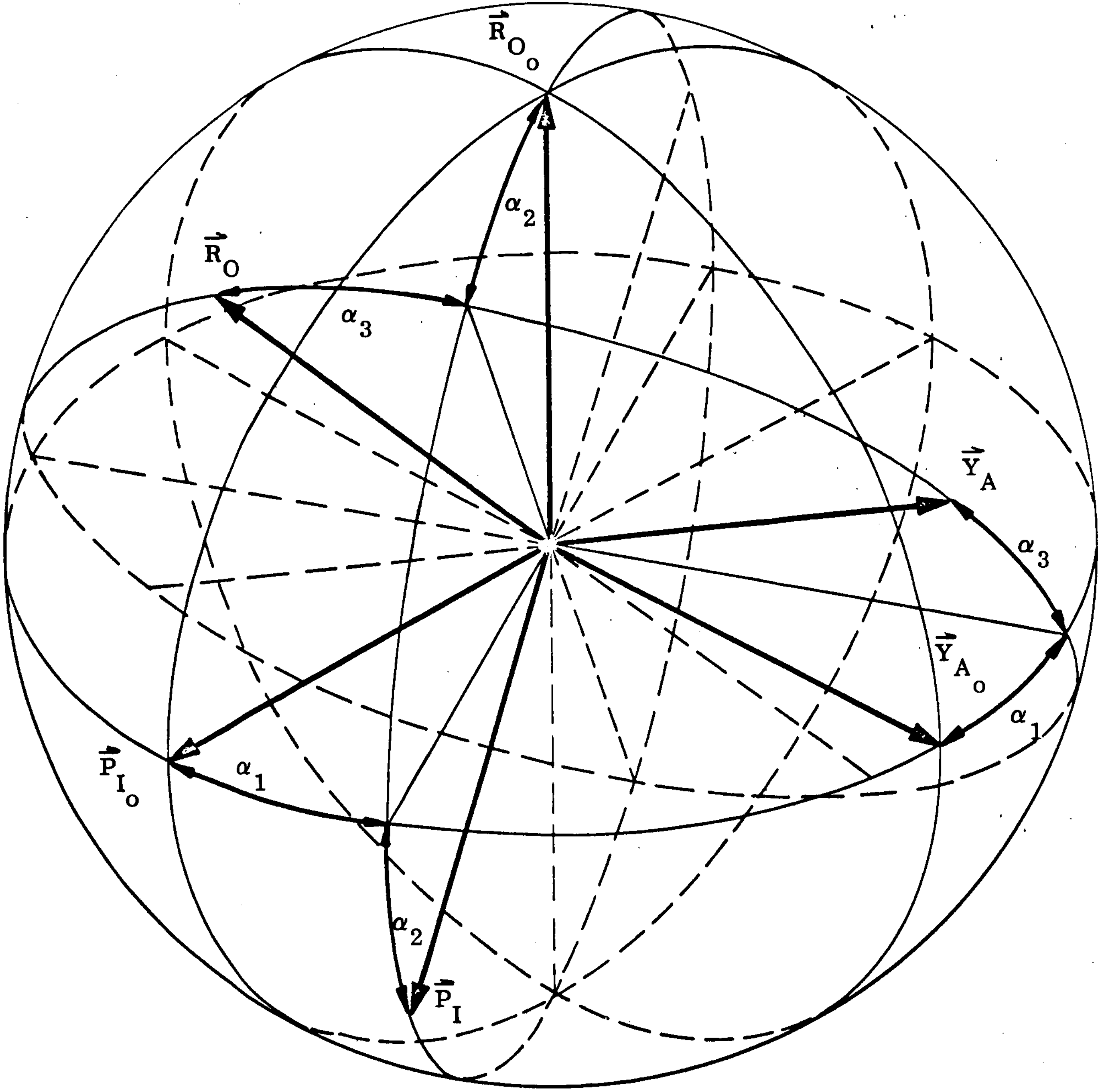


Figure 3. 4. 2. Gimbal Angles

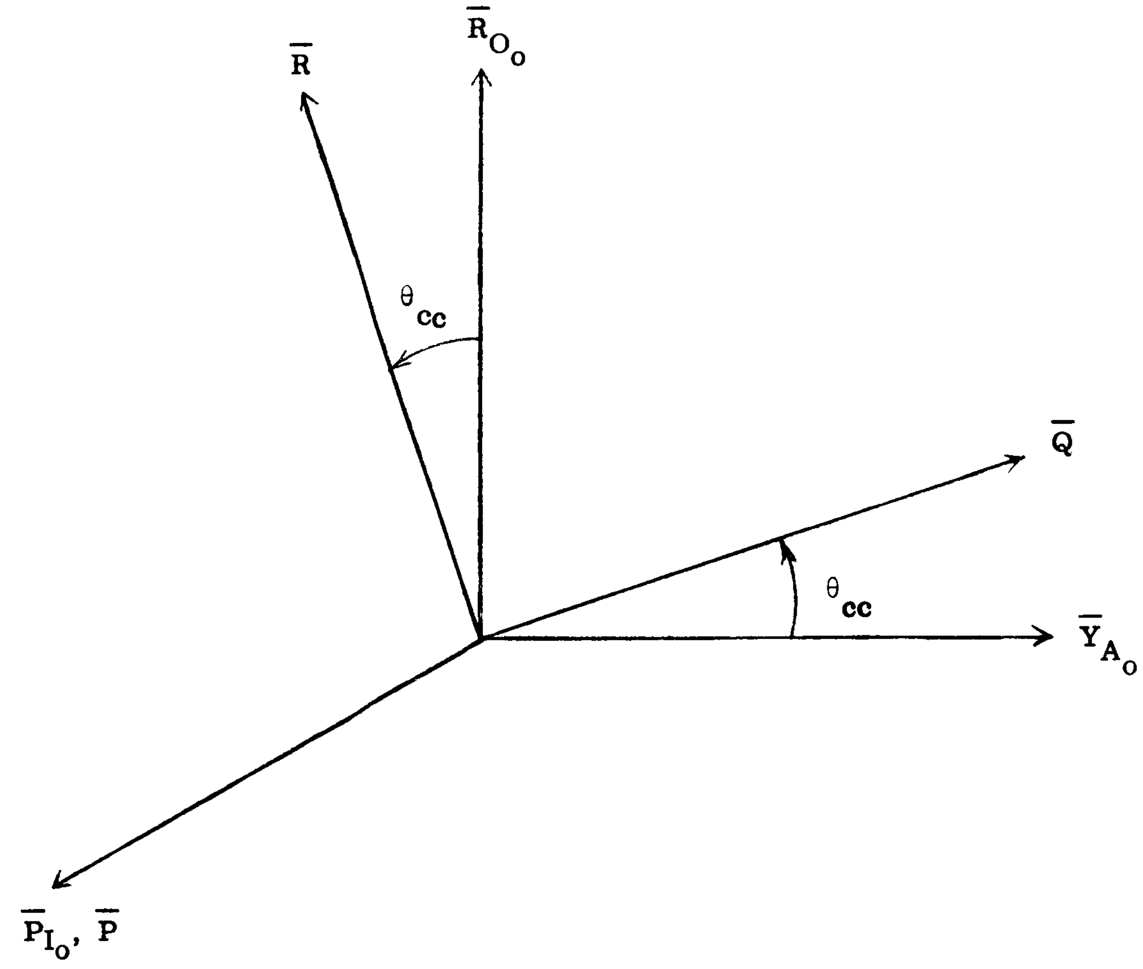
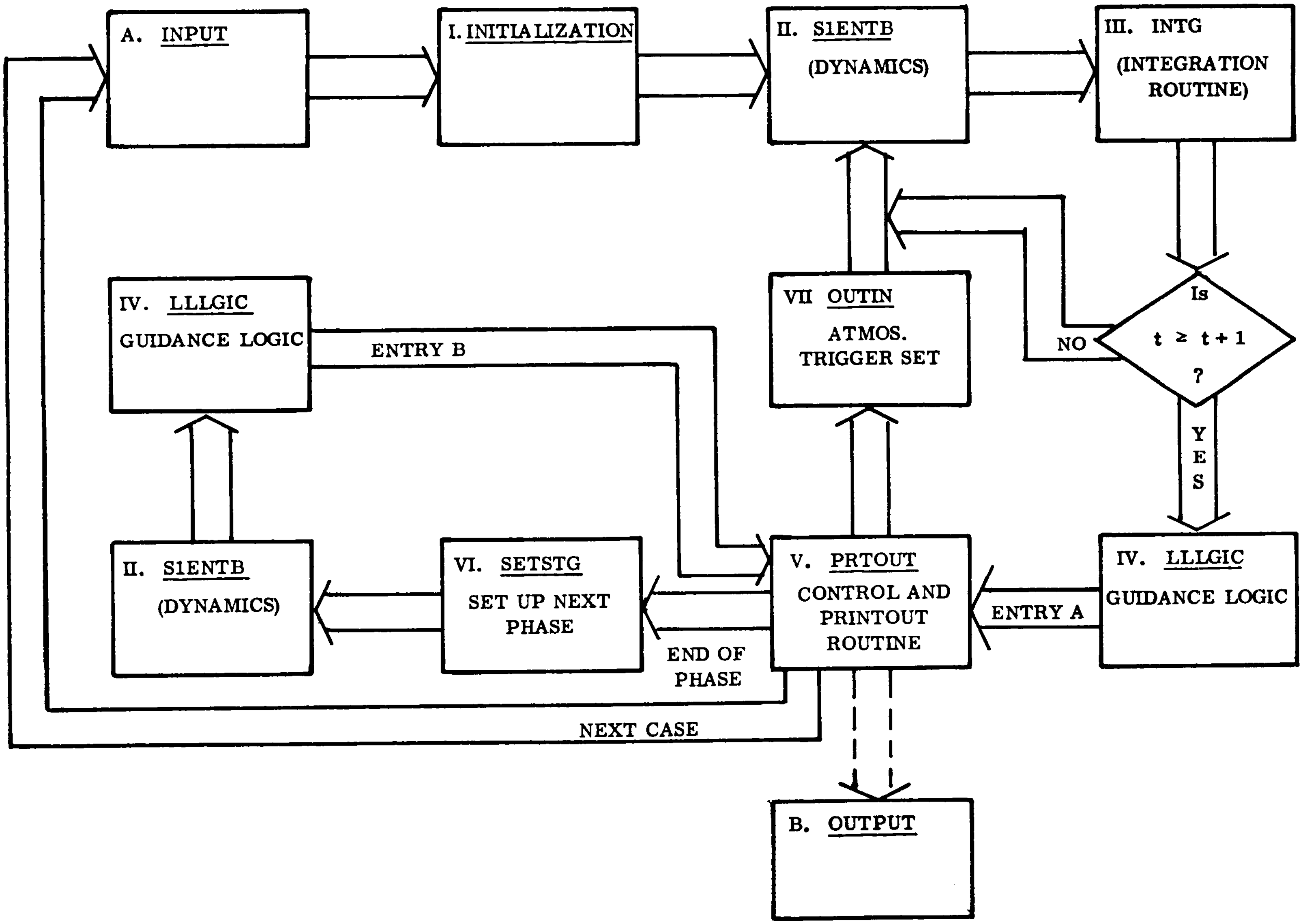

$\overline{R}$
$\overline{R}_{O_o}$
θ_{cc}
$\overline{Q}$
θ_{cc}
$\overline{Y}_{A_o}$
$\overline{P}_{I_o}$, $\overline{P}$

Figure 3.4.3

3.5 GENERALIZED BOOSTER GUIDANCE SIMULATION

3.5.1 LEVEL I FLOW CHART



Level I Flow Chart - Generalized Booster Guidance System

4.0 COMPUTER PROGRAM DESCRIPTION

The basic structure of the program is summarized in the Level I flow chart. It constitutes, according to the preceeding definitions. the Level I flow chart and consists of two different classes of blocks. Those which define the basic computational cycles of the program (Roman numerals), and those necessary to start the program in a prescribed way or define the required output (Arabic letters A and B).

Blocks A and B are described in Section 4.1; Blocks I through VII in Section 4.2.

The Input block represents a summary of the quantities that an engineer must input. No computations are contained within this block. The Output block defines the quantities that are to be available for printout purposes (including storing or magnetic tape) and contains edit computations that are not required in the basic computational cycle. The General Initialization block contains computations that must be performed once during a specific simulation run and/or logical decisions that must be made for proper operation within the basic computational cycle.

The program is basically broken up into two parts: that of integrating the equations of motion (Blocks II and III), and that of guiding the path of the vehicle (Block IV). The other blocks are used in control and logic of the program. The major output of this program is a time history of the vehicle motion throughout the trajectory.

4.1 INPUT AND OUTPUT

4.1.1 Definition of Flags

4.1.1.1 Nominal Trajectory

1. **ATMOPT** — **flag for specifying desired atmospheric routine**
ATMOPT = 0 → U. S. Standard Atmosphere, 1962
ATMOPT = 1 → planetary atmosphere determined by ρ_o, β_1, C for planet

2. **OPO123** — **flag for specifying guidance scheme through sensible atmosphere**
OPO123 = 0 → beta star guidance
OPO123 = 1 → $\dot{P}$, $\dot{Q}$, $\dot{R}$ curve fit guidance

3. **ST2BET** — **flag used in H-start to set up BETTRG for phase 2 starts**
ST2BET = 0 → guidance before T_3
ST2BET = 1 → guidance after T_3

4. **ST2T4T** — **flag used in H-start to set up T_4 trigger for phase 2 starts**
ST2T4T = 0 → guidance before T_1
ST2T4T = 1 → guidance before T_4

5. **ST3BET** — **same as ST2BET for phase 3 starts**

6. **ST3T4T** — **same as ST2T4T for phase 3 starts**

7. **STINOU** — **flag used in H-start to set up OUTIN trigger**
STINOU = 0 → vehicle is in the atmosphere
STINOU = 1 → vehicle is out of the atmosphere

8. **ST5R12** — **flag used in H-start to set up trigger for T_5 with phase 5 starts**
ST5R12 = 0 → look for T_5
ST5R12 = 1 → do not look for T_5

9. **ST7134** — **flag used in H-start to set up trigger for T_6 with phase 7 starts**
ST7134 = 0 → look for T_6
ST7134 = 1 → do not look for T_6

10. ST9138 flag used in H-start to set up trigger for T_8 with phase 9 starts
ST9138 = 0 → look for T_8
ST9138 = 1 → do not look for T_8

11. WFLAG flag used to denote whether winds are desired in the computation of $\overline{V}_A$
WFLAG = 0 → no winds desired
WFLAG = 1 → winds desired

4.1.1.2 State Transition Matrix

1. NOR6PT flag for computing or skipping state transition matrix
NOR6PT = 0 → compute state transition matrix
NOR6PT = 1 → do not compute state transition matrix

2. PROPT flag for specifying the printout desired and for saving data on tape
PROPT = 0 → print state transition matrix and necessary data for all inertial guidance system and save on tape
PROPT = 1 → print standard nominal data

4.1.2 Definition of Input Quantities

4.1.2.1 Nominal Trajectory

GUIDANCE INPUTS FROM T_0 to T_4:

OP0123	Guidance option for β^* or $\dot{P}$, $\dot{Q}$, $\dot{R}$ guidance
KK_{50}, KK_{51}	$\dot{P}$, $\dot{Q}$, $\dot{R}$ steering gains for α_{1C}
KK_{52}	$\dot{P}$, $\dot{Q}$, $\dot{R}$ steering gains for α_{3C}
KK_{53}- KK_{61}	curve fit constants for $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering
CCTHET (θ_{cc})	angle between computational coordinates for $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering and launch level coordinates measured in the X-Z plane (DMS)
RSTOP1	$\dot{R}_a$ criteria for computing T_1 with $\dot{P}$, $\dot{Q}$, $\dot{R}$ guidance (ft/sec)
RSTOP4	$\dot{R}_a$ criteria for computing T_4 with $\dot{P}$, $\dot{Q}$, $\dot{R}$ guidance (ft/sec)
KXXKXX (K_α)	constant for criteria used in pitch over with β^* guidance between T_1 and T_3
BESTT1 (T_1)	end of zero attitude steering and beginning of pitchover phase (secs)
BESTT3 (T_3)	end of pitchover and beginning of zero angle of attack guidance (secs)
BESTT4 (T_4)	end of zero angle of attack guidance and beginning of explicit guidance (secs)

AIMING POINT INPUTS FOR GUIDANCE AFTER T_4:

FAIMJ	trigger for determining aiming point computation, HA4A or HA4B. (j dependent) (F_{AIM_j} = 0 → HA4A, F_{AIM_j} = 1 → HA4B)
HRAJVX, HRAJVY, HRAJVZ (C_1, C_2, C_3) when FAIMJ = 1	normalized cylindrical offset coordinates of the aiming point used where aiming point is rotating with the planet. (BLK HA4B) (j dependent)
HRAJVX, HRAJVY, HRAJVZ ($\overline{R}_A$) when FAIMJ = 0	target coordinates in inertial coordinates for explicit guidance (Block HA4A) (j dependent)

HTOJ (T_o) initial estimate for time of flight for use with planet fixed target aiming point (Block HA4B) (j dependent) (secs)

CONSTANTS FOR V_g COMPUTATION:

HKJ (K) the universal gravitational constant, $\sqrt{GM}$ (j dependent)

HKRJ (K_r) 1/K (j dependent)

HPJ (p) semilatus rectum for the free flight ellipse (j dependent)

HUWJVX, HUWJUY, HUWJVZ ($\overline{U}_w$) unit vector perpendicular to the nominal trajectory plane (j dependent)

HES (ϵ_s) criterion used to determine computation of (e sin v) in block HA5

HEU (ϵ_u) criterion used to determine computation of $\overline{U}_v$ in block HA5

CONSTANTS FOR TRAJECTORY CONSTRAINTS:

FCONJ trigger for trajectory constraint at the target of constant time of arrival, HA6A or constant flight path angle, HA6B. (j dependent) ($F_{CON_j} = 0 \rightarrow$ HA6A, $F_{CON_j} = 1 \rightarrow$ HA6B)

HK13J (K_{13}) semilatus rectum correction coefficient (j dependent)

HK14J (K_{14}) semilatus rectum correction coefficient (j dependent)

HEP (ϵ_p) criterion used to determine computation of p_{i+1} in blocks HA6A and HA6B

HET (ϵ_T) criterion used to determine computation of p_{i+1} in block HA6A

HTAJ (T_A) input constant specifying the time of arrival when constant time of flight constraint is used (j dependent) (secs)

HEGAM (ϵ_{γ}) criterion used to determine computation of p_{i+1} in block HA6B

HKGJ (K_{γ}) desired flight path angle at the aim point (j dependent) (radians)

CONSTANTS FOR CURVE FIT GUIDANCE:

HARCJ, HBRCJ, HCRCJ (A_{RC}, B_{RC}, C_{RC}) curve fit constants for radial distance as a function of time used in trim guidance. (j dependent)

HARDJ, HBRDJ, HCRDJ (A_{RD}, B_{RD}, C_{RD}) curve fit constants for radial rate as a function of time used in trim guidance. (j dependent)

HARVJ, HBRVJ, HCRVJ (A_{RV}, B_{RV}, C_{RV}) curve fit constants for tangential velocity as a function of time used in trim guidance (j dependent)

HKAJ (K_A) steering gains used during trim guidance (j dependent)

HKHJ (K_H) steering gain used during trim guidance (j dependent)

FTRIMJ trigger for determining whether cutoff of phase based on velocity to be gained is to be used. (j dependent) ($F_{TRIMj} = 0$ cutoff on V_g, $F_{TRIMj} = 1$ no cutoff on V_g)

CONSTANTS FOR ATTITUDE COMMANDS:

HK2J, HK3J, (k_2, k_3) steering gains used in $\overline{V}_g$ steering to compute commanded gimbal angles (j dependent)

HK4J, HK5J, (k_4, k_5) steering gains used in $\overline{V}_g$ steering to compute commanded gimbal angles (j dependent)

HAOJ, HBOJ, HCOJ (A_{θ}, B_{θ}, C_{θ}) curve fit constants for reducing steering errors (j dependent)

CRITICAL TIME POINTS:

HTAU5 (τ_5) duration of phase 6 (secs)

HTAU6 (τ_6)	criterion for determination of T6 (secs)
HTAU6P (τ'_6)	criterion for the determination of T7 (secs)
HTAU7 (τ_7)	duration of phase 8 (secs)
HTAU8 (τ_8)	criterion for the determination of T8 (secs)
HTAU8P (τ'_8)	criterion for the determination of T9 (secs)

CRITERIA FOR LOGICAL DECISIONS:

HRTRJ (r_{TR})	radial distance criterion for beginning of trim guidance (j dependent) (ft)
HETR (ϵ_{TR})	criterion for reaching radial distance condition for a critical time point (secs)
HUTRJX, HUTRJY, HUTRJZ ($\overline{U}_{TR}$)	unit vector criterion for beginning of trim guidance (j dependent)
HETRA (ϵ_{TRA})	criterion for reaching a position in determination of a critical time point (secs)
HREXJ (r_{EX})	radial distance criterion for beginning of explicit guidance (j dependent) (ft)
HUEXJX, HUEXJY, HUEXJZ ($\overline{U}_{EX}$)	unit vector criterion for beginning of explicit guidance (j dependent)
HEVG (ϵ_{vg})	criterion for velocity to be gained cutoff (ft/sec)
KAL1, KAL2, KAL3 ($K_{\alpha L}$)	lower bound for α_{1vg} (k dependent) (radians)
KAU1, KAU2, KAU3 ($K_{\alpha U}$)	upper bound for α_{1vg} (k dependent) (radians)
KGL1, KGL2, KGL3 ($K_{\gamma L}$)	lower bound for α_{3vg} (k dependent) (radians)

KGU1, KGU2, KGU3 ($K_{\gamma U}$)	upper bound for α_{3vg} (k dependent) (radians)
MDOTT6	perturbation in mass flow rate which is used from T6 or T7 to the end of the phase
MDOTT8	perturbation in mass flow rate which is used from T8 or T9 to the end of the phase

DYNAMICS INPUT OPTIONS:

ATMOPT	trigger specifying desired atmospheric routine (ATMOPT = 0 → earth atmospheric routine ATMOPT = 1 → planetary atmospheric routine)
WFLAG	trigger to denote whether winds are desired in the computation of V_A (WFLAG = 0 → no winds desired)
EFLAG (P_E)	perturbation factor for east wind
NFLAG (P_N)	perturbation factor for north wind
CKREV (K_{REV})	thrust coefficient
IIIIII	number of the last phase to be used up to phase 9

PLANET PHYSICAL CONSTANTS:

A(a_e)	equatorial radius of the planet
ELLIP (ϵ)	ellipticity or flattening of the reference geoid
EE (GM)	universal gravitational constant times mass of planet
OMEGA (Ω)	planet angular stellar rate
HC (h_c)	height of the sensible atmosphere
LGSEP (N_L)	difference between reference geoid and actual sea level
LALT (H_L)	height of launchsite above sea level

GS (g)	sea level value of gravitational acceleration
DGLAT (λ_L)	geodetic latitude of launch site (DMS)
DALAT (λ_{LA})	astronomical latitude at launch site (DMS)
DGLON (μ_L)	geodetic or geocentric longitude at launch site (DMS)
DALON (μ_{LA})	astronomical longitude at launch site (DMS)
DPAZ (Ψ_p)	azimuth angle of the pitch plane with respect to North (DMS)
JXJXJ (J_g)	coefficient of the second harmonic of the earth's gravitational potential
HXHXH (H_g)	coefficient of the third harmonic of the earth's gravitational potential
DXDXD (D_g)	coefficient of the fourth harmonic of the earth's gravitational potential
RSUBC1 (R_{C1})	range coefficient
CSUBI	speed of sound constant
BETAI (β_i)	atmospheric density decay parameter
RHOO (ρ_o)	mean sea level atmospheric density
PNOT (P_o)	mean sea level atmospheric pressure

PHASE-DEPENDENT INPUTS:

M	present vehicle mass (slugs)
E_G	location of the engine gimbal (ft)
S	effective drag area during boost (ft^2)
P_{FLO}	perturbation factor in mass flow rate
P_{ISP}	perturbation factor in I_{SP}

δt	minor cycle length phase dependent dynamics block integration step size (secs)
Δt	major computational cycle length, phase dependent guidance block integration step size (secs)
MES	mass at the end of phase (slugs)
β_j	phase dependent parameter defining the maximum angular rate of the vehicle (DMS/sec)
K_θ	pitch position gain
$K_{\dot{\theta}}$	pitch rate gain
K_φ	roll position gain
$K_{\dot{\varphi}}$	roll rate gain
K_ψ	yaw position gain
$K_{\dot{\psi}}$	yaw rate gain
ΔP	phase dependent print interval (secs)

DYNAMICS TABLES:

Header for tables	title for printout to identify the vehicle being simulated
"A" TABLES (gI_{SP} vs. P)	table of 32.174x specific impulse vs. pressure for each phase of the vehicle simulation (ft/sec vs. lbs/in^2)
"B" TABLES ($\dot{M}$ vs. $\int \dot{M}$)	table of mass flow rate vs. expended mass for each phase of the vehicle simulation (slugs/sec vs. slugs)
"C" TABLES (C_A vs. R_v)	table of axial drag coefficient vs. mach number for each phase of the vehicle simulation
"G" TABLES (C_g vs. M)	table of center of gravity vs. mass for each phase of the vehicle simulation (ft vs. slugs)

TBLJ (North Wind vs. H)	table of north wind velocities vs. altitude (ft/sec vs. ft)
TBLK (East Wind vs. H)	table of east wind velocities vs. altitude (ft/sec vs. ft)
TBLD (C_N vs. R_v, α)	bivariate table of normal drag coefficient vs. mach number and angle of attack for each phase of the vehicle simulation (ND vs. ND and radians)
TBLH (C_p vs. R_v, α)	bivariate table of center of pressure vs. mach number and angle of attack for each phase of the vehicle simulation (ft vs. ND and radians)

GENERAL INPUT FOR STARTING SIMULATION AT BEGINNING OF ANY PHASE GREATER THAN 1:

STTIME	starting time for the phase (secs)
STMASS	starting mass for the phase (slugs)
STPOS	starting position vector (ft)
STVEL	starting velocity vector (ft/sec)
STGIMB	starting gimbal angles (radians)
STINOU	starting trigger for in or out of the atmosphere 0 = in atmosphere; 1 = out of atmosphere
VAXYZ	starting $\int a_x$, $\int a_y$, $\int a_z$ (ft/sec)
STBETS	beginning phase
STJ	starting "j" number
STK	starting "k" number
QBIAS	whole number part of time - not necessary if included in STTIME (secs)

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 2:

ST2BET	flag for BETTRG 0. = after T_3; 1. = before T_3
ST2L	L number
ST2P11	flag for print of line 13 in normal print 0. = no line 13 print; 1. = print line 13
ST2T4T	flag for T_4 trigger 0 = before T_1; 1. = after T_1 and before T_4

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 3:

ST3BET	same as ST2BET
ST3L	same as ST2L
ST3P11	same as ST2P11
ST3T4T	same as ST2T4T

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 5:

ST4R12	flag for T_5 trigger 0 = look for T_5; 1 = do not look for T_5

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 6:

ST6ES6	TES6 (secs)

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 7:

ST7R6	T_6 (secs)
ST7R7	T_7 (secs)
ST7134	flag for T_6 trigger 0. = look for T_6; 1. = do not look for T_6

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 8:

ST8135	TESS (secs)

INPUT QUANTITIES FOR STARTING AT BEGINNING OF PHASE 9:

ST9136	T_8 (secs)
ST9137	T_9 (secs)
ST9138	flag for T_8 trigger 0 = look for T_8; 1 = don't look for T_8

MISCELLANEOUS INPUT:

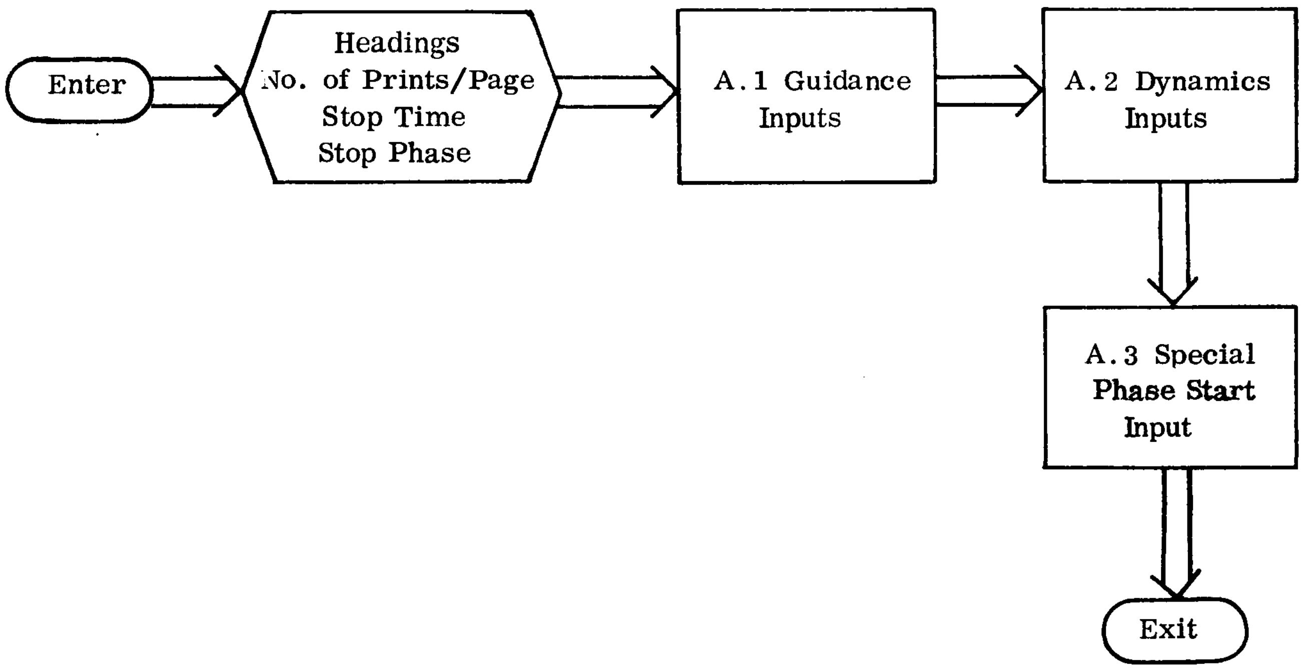
KNPPP	number of time prints per page
RUNNO	run number
SFINAL	final phase for cutoff
TFINAL	final time for cutoff (secs)

4.1.2.2 State Transition Matrix Generation Inputs

NOR6PT	flag for state transition matrix 0 = compute state transition matrix 1 = do not compute state transition matrix
PROPT	flag for output 0 = printout state transition matrix and necessary data for all-inertial guidance system and save on tape 1 = print nominal print
$\Delta P'$	interval for saving data on tape (secs)

4.1.3 INPUT - BLOCK A

4.1.3.1 LEVEL II FLOW CHART



BLOCK A. INPUT

4.1.3.2 BLOCK A.1

GUIDANCE INPUTS

OP 0123 (Guidance Option for β* or P, Q, R Guidance)

Constants for Zero Angle of Attack Guidance (BLOCK A3)

$$\left.\begin{array}{ll} 1. & K_{50} - K_{61} \\ 2. & \theta_{cc} \\ 3. & R_{STOP1} \\ 4. & R_{STOP4} \end{array}\right\} \text{For P, Q, R}$$

$$\left.\begin{array}{ll} 5. & K_{\alpha} \\ 6. & T_{1} \\ 7. & T_{3} \\ 8. & T_{4} \end{array}\right\} \text{For } \beta^{*}$$

Constants for Determining Aiming Point (BLOCKS 4A and 4B)

1. F_{AIMj}
2. C_{1j}
3. C_{2j}
4. C_{3j}
5. $\overline{R}_{Aj}$
6. T_{Oj}

Constants for $\overline{V}_g$ Computation (BLOCK 5)

1. K_j
2. K_{rj}
3. p_j

Constants for $\overline{V}_g$ Computation (BLOCK 5) - con't

4. $\overline{U}_{wj}$
5. ϵ_u
6. ϵ_s
7. K_{VG}

Constants for Trajectory Constraints (BLOCKS 6A and 6B)

1. F_{CONj}
2. K_{13j}
3. K_{14j}
4. ϵ_p
5. ϵ_T
6. T_{Aj}
7. ϵ_Y
8. K_{Yj}

Constants for Curve Fit Guidance (BLOCK A7)

1. $\overline{U}_{wj}$
2. A_{RCj}
3. B_{RCj}
4. C_{RCj}
5. A_{RDj}
6. B_{RDj}
7. C_{RDj}
8. A_{RVj}

Constants for Curve Fit Guidance (BLOCK A7) - con't

9. B_{RVj}

10. C_{RVj}

11. K_{Aj}

12. K_{Hj}

13. F_{TRIMj}

Constants for Attitude Commands (BLOCK S1)

1. k_{2j}

2. k_{3j}

3. k_{4j}

4. k_{5j}

5. $A_{\theta j}$

6. $B_{\theta j}$

7. $C_{\theta j}$

Critical Time Points

1. τ_5

2. τ_6

3. τ_6'

4. τ_7

5. τ_8

6. τ_8'

Criteria for Logic Decisions

1. r_{TR}
2. ϵ_{TR}
3. $\overline{U}_{TRj}$
4. ϵ_{TRAj}
5. r_{EXj}
6. $\overline{U}_{EXj}$
7. ϵ_{vg}
8. $K_{\alpha Lk}$
9. $K_{\alpha uk}$
10. $K_{\gamma Lk}$
11. $K_{\gamma uk}$
12. $\dot{M}_{T6}$
13. $\dot{M}_{T8}$

4.1.3.3 BLOCK A.2

DYNAMICS INPUT

Dynamics Options

1. EARFLG (atmospheric option)
2. WFLAG (wind option)
   - 2.1 North Wind Perturbation
   - 2.2 East Wind Perturbation
3. Number of Phases
4. Run number
5. K_{REV} (thrust coefficient)

Planet Physical Constants

1. a_e
2. ϵ
3. GM
4. Ω
5. h_c
6. N_L
7. H_L
8. g
9. λ_L
10. μ_L
11. λ_{LA}
12. μ_{LA}
13. ψ_p
14. J_g
15. H_g
16. D_g
17. GM'

Planet Physical Constants - con't

18. a'

19. u'_o

20. g'_o

21. C'_2

22. Ω'

23. C'_{a1}

24. C'_{a2}

25. C'_{a3}

26. C'_{a4}

27. C'_{a5}

28. γ'

29. $\bar{C}'_s$

30. Table used in atmospheric routine

31. β_i

32. ρ_o

33. P_o

Phase Dependent Parameters

1. M_0

2. E_G

3. S

4. P_{ISP}

5. P_{FLO}

6. Δt

7. δt

Phase Dependent Parameters - con't

8. MES
9. β_j
10. K_θ
11. $K_{\dot{\theta}}$
12. K_ψ
13. $K_{\dot{\psi}}$
14. K_φ
15. $K_{\dot{\varphi}}$
16. ΔP

Dynamics Tables

1. Header for tables
2. gI_{SP} vs. P (Phase dependent)
3. $\dot{M}$ vs $\int \dot{M}$ (Phase dependent)
4. C_A vs R_v (Phase dependent)
5. C_g vs M (Phase dependent)
6. North Wind vs H
7. East Wind vs H
8. C_N vs R_v, α (Phase dependent)
9. C_p vs R_v, α (Phase dependent)

State Transition Matrix Inputs

1. NOR6PT
2. PROPT
3. $\Delta P'$

4.1.3.4 BLOCK A.3

START INPUT

General Start Parameters

1. Time
2. M
3. X, Y, Z
4. $\dot{X}$, $\dot{Y}$, $\dot{Z}$
5. α_1, α_2, α_3
6. In-Out atmospheric trigger
7. $\int a_X, \int a_Y, \int a_Z$
8. STBEGS
9. Q-BIAS
10. j
11. k

Start Parameters (Phases 2 to 4)

1. BETTRG
2. STEXT + 11
3. T_4 trigger
4. ℓ

Start Parameter (Phase 5)

1. T_5 trigger

Start Parameter (Phase 6)

1. TES 6

Start Parameters (Phase 7)

1. T_6
2. T_7
3. T_6 trigger

Start Parameter (Phase 8)

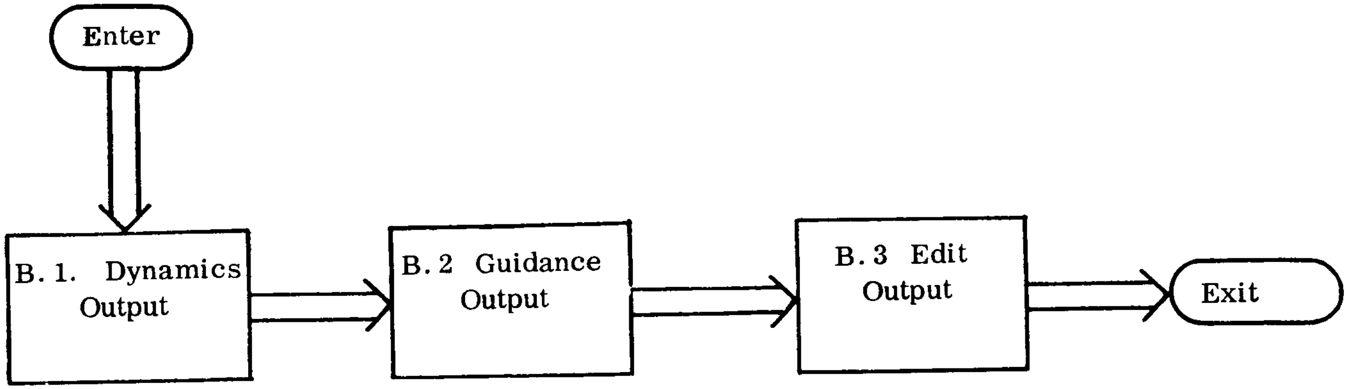
1. TES 8

Start Parameters (Phase 9)

1. T_8
2. T_9
3. T_8 trigger

4.1.4 BLOCK B

4.1.4.1 LEVEL II FLOW CHART



BLOCK B. OUTPUT

4.1.4.2 BLOCK B.1

DYNAMICS OUTPUT

1. time
2. height
3. dynamic pressure
4. $\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$
5. $\dot{X}$, $\dot{Y}$, $\dot{Z}$
6. X, Y, Z
7. g_X, g_Y, g_Z
8. $\bar{\omega}_{PI}$, $\bar{\omega}_{YA}$, $\bar{\omega}_{RO}$
9. r
10. $\dot{r}$
11. R_V
12. α_1, α_2, α_3
13. F_N
14. M
15. $\dot{M}$
16. V
17. V_A
18. a_T

4.1.4.3 BLOCK B.2

GUIDANCE OUTPUT

1. V_{REQX}, V_{REQY}, V_{REQZ}
2. V_{gX}, V_{gY}, V_{gZ}
3. p_i, p_{i-1}
4. T_i, T_{i-1}
5. a
6. e
7. V_{REQ}
8. α_{1c}, α_{2c}, α_{3c}
9. β^*
10. r_c
11. $\dot{r}_c$
12. $(r\dot{v})_c$
13. θ'_{EC} ψ'_{EC}
14. Δt_{go}
15. α_{1vg_j}, α_{2vg_j}, α_{3vg_j}
16. $\Delta\alpha_{1c}$, $\Delta\alpha_{2c}$, $\Delta\alpha_{3c}$
17. ΔR_X, ΔR_Y, ΔR_Z
18. $\overrightarrow{\Delta R} \cdot \vec{R}_{o_o}$, $\overrightarrow{\Delta R} \cdot \vec{P}'_I$
19. s
20. θ_k
21. γ_{A_i}, $\gamma_{A_{i-1}}$
22. Δv
23. $\dot{R}_a$, $\dot{Q}_a$

4.1.4.4 Block B.3 - Edit Output

INPUT: $X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \dot{X}_A, \dot{Y}_A, \dot{Z}_A, \Omega, t, \epsilon, r, R_{c_1}, \theta'_{EC}, \psi'_{EC}, \varphi'_{EC}$

OUTPUT: $\lambda, \mu, \beta, \gamma$, Range, $\overline{\omega}_{PI}, \overline{\omega}_{YA}, \overline{\omega}_{R0}$

1. $\lambda = \tan^{-1}\left[\frac{Z}{(1-\epsilon)^2\sqrt{(X^2+Y^2)}}\right]$ $\quad -\frac{\pi}{2} < \lambda \leq \frac{\pi}{2}$

2. $\mu = \mu_L - \Omega t + \tan^{-1}\left[\frac{Y}{X}\right]$ $\quad -\pi < \mu \leq \pi$ $\quad -\pi < \tan^{-1}\frac{Y}{X} \leq \pi$

3. $\beta = \tan^{-1}\left\{\frac{|\overline{r} \times \overline{V}_A|}{|\overline{r} \cdot \overline{V}_A|}\right\}$ $\quad 0 \leq \beta < \pi$

4. $\gamma = \tan^{-1}\left\{\frac{|\overline{r} \times \overline{V}|}{|\overline{r} \cdot \overline{V}|}\right\}$ $\quad 0 \leq \gamma < \pi$

5. Funct. $= [\cos(\mu - \mu_L)]\,[(X^2 + Y^2)^{1/2}/r]\,[\cos\lambda_{LC}] + [Z/r]\sin\lambda_{LC}$

6. Range $= (R_{c_1})(\cos^{-1}$ Funct) $\quad 0 \leq \cos^{-1}$ Funct. $\leq \pi$

7. $\overline{\omega}_{PI} = \frac{\theta'_{EC_{i-1}} - \theta'_{EC_i}}{\Delta t_i}$

8. $\overline{\omega}_{YA} = \frac{\psi'_{EC_{i-1}} - \psi'_{EC_i}}{\Delta t_i}$

9. $\overline{\omega}_{R0} = \frac{\varphi'_{EC_{i-1}} - \varphi'_{EC_i}}{\Delta t_i}$

1	t	R	h	F_N	M	$\dot{M}$
2	q	$q\alpha$	θ_k	$\Delta\overline{R} \cdot \overline{R}_{00}$	$\Delta\overline{R} \cdot P'_I$	s
3	$\ddot{X}$	$\ddot{Y}$	$\ddot{Z}$	$V_{REQ\ X}$	$V_{REQ\ Y}$	$V_{REQ\ Z}$
4	$\dot{X}$	$\dot{Y}$	$\dot{Z}$	V_{gX}	V_{gY}	V_{gZ}
5	X	Y	Z	$\Delta\alpha_{1c}$(DMS)	$\Delta\alpha_{2c}$(DMS)	$\Delta\alpha_{3c}$(DMS)
6	g_X	g_Y	g_Z	a	Δv	e
7	$\overline{\omega}_{PI}$	$\overline{\omega}_{YA}$	$\overline{\omega}_{RO}$	ΔR_X	ΔR_Y	ΔR_Z
8	λ (DMS)	μ(DMS)	β(DMS)	a_T	θ'_{EC}(DMS)	ψ'_{EC}(DMS)
9	r	$\dot{r}$	γ (DMS)	V_A	V_{REQ}	V
10	r_c	$\dot{r}_c$	$(r\ \dot{v})_c$	α_1(DMS)	α_2(DMS)	α_3(DMS)
11	p_i	γ_{Ai}	T_i	α_{1c}(DMS)	α_{2c}(DMS)	α_{3c}(DMS)
12	p_{i-1}	γ_{Ai-1}	T_{i-1}	α_{1vgj}(DMS)	α_{2vgj}(DMS)	α_{3vgj}(DMS)
13*	α_3(rad)	$\dot{Q}_a$	$\dot{R}_a$	β*	Δt_{go}	R_v

* Line 13 is printed out up to T_4 only. (DMS) = (Degrees, Minutes, Seconds)

PRINT KEY (PROPT = 1.)

Print Key for Program 118.0

1	TIME	RUN NO.	PHASE	$\bar{\omega}_{PI}$	$\bar{\omega}_{YA}$	$\bar{\omega}_{RO}$
2	X	Y	Z	$\dot{X}$	$\dot{Y}$	$\dot{Z}$
3	a_X	a_Y	a_Z	α_1	α_2	α_3
4	$\Phi_{1,1}$	$\Phi_{1,2}$	$\Phi_{1,3}$	$\Phi_{1,4}$	$\Phi_{1,5}$	$\Phi_{1,6}$
5	$\Phi_{2,1}$	$\Phi_{2,2}$	$\Phi_{2,3}$	$\Phi_{2,4}$	$\Phi_{2,5}$	$\Phi_{2,6}$
6	$\Phi_{3,1}$	$\Phi_{3,2}$	$\Phi_{3,3}$	$\Phi_{3,4}$	$\Phi_{3,5}$	$\Phi_{3,6}$
7	$\Phi_{4,1}$	$\Phi_{4,2}$	$\Phi_{4,3}$	$\Phi_{4,4}$	$\Phi_{4,5}$	$\Phi_{4,6}$
8	$\Phi_{5,1}$	$\Phi_{5,2}$	$\Phi_{5,3}$	$\Phi_{5,4}$	$\Phi_{5,5}$	$\Phi_{5,6}$
9	$\Phi_{6,1}$	$\Phi_{6,2}$	$\Phi_{6,3}$	$\Phi_{6,4}$	$\Phi_{6,5}$	$\Phi_{6,6}$

PRINT KEY (PROPT = 0, NOR6PT = 0)

(Decimal Data)

$\int a_X$	$\int a_Y$	$\int a_Z$

Beginning of Phase (No.)

t	QBIAS	MASS	α_1	α_2	α_3
$\int a_X$	$\int a_Y$	$\int a_Z$	—	—	—
X	Y	Z	$\dot{X}$	$\dot{Y}$	$\dot{Z}$

OCTAL CODED DATA PRINT AT BEGINNING OF EACH PHASE

Print Keys for Program 118.0

4.2 BASIC COMPUTATION BLOCKS

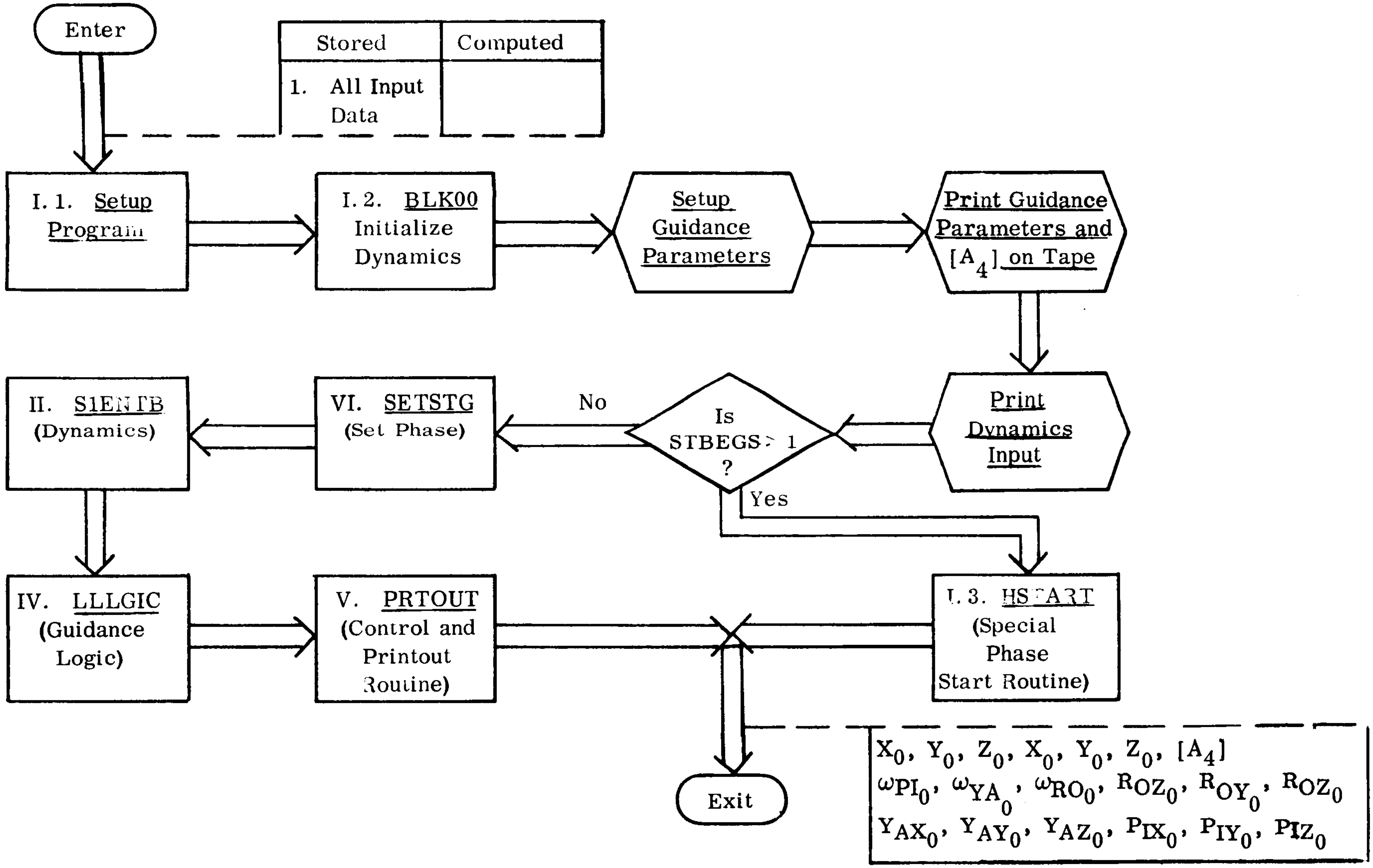
The blocks covered in this section are those in which computations affecting the run are performed. The initialization block (I) is only computed once at the beginning of the run and is not in the computational cycle.

4.2.1 Initialization - Block I

The initial position and velocity of the vehicle in the reference PCI coordinates and the initial orientation of the reference body axes are calculated in this block. The vehicle is placed in the X-Z plane (for $\mu_{LD} = 0$) of the PCI coordinate system and has an initial velocity in the Y direction due to the planet's stellar rate of rotation. The geoidal separation at launch, N_L (earth only), and the altitude above the reference figure for the planet, H_L, are used for calculation of the initial position. The body axes are oriented with respect to the local vertical, as defined by λ_{LA} and μ_{LA}, and the launch azimuth ψ_p. The roll axis, $\overline{R}_o$, is vertical at the launch site and for a zero (North) launch azimuth, ψ_p; the pitch axis, $\vec{P}_I$, is pointed West and the yaw axis, $\vec{Y}_A$, is pointed South in the launch level plane.

Detailed flow charts and equations follows.

This is a blank page


Enter
Stored
Computed
1. All Input Data
I. 1. Setup Program
I. 2. BLK00 Initialize Dynamics
Setup Guidance Parameters
Print Guidance Parameters and [A4] on Tape
Print Dynamics Input
Is STBEGS > 1 ?
No
Yes
VI. SETSTG (Set Phase)
II. S1ENTB (Dynamics)
IV. LLLGIC (Guidance Logic)
V. PRTOUT (Control and Printout Routine)
I. 3. HSTART (Special Phase Start Routine)
Exit
X0, Y0, Z0, X0, Y0, Z0, [A4]
ωPI0, ωYA0, ωRO0, ROZ0, ROY0, ROZ0
YAX0, YAY0, YAZ0, PIX0, PIY0, PIZ0

4.2.1.1 BLOCK I. INITIALIZATION

I. 1 SETUP PROGRAM

1. Setup outside subroutines
2. Clear storage
3. Compute C_0, C_1, C_2, C_3, C_4 in BLK 01
4. Setup prints per page
5. Setup edit
6. Setup atmospheric trigger
7. Setup number of equations integrated

I.2 BLOCK 00

Output: Initial Conditions

Input: ϵ, λ_L, a_e, N_L, H_L, Ω, μ_{LA}, μ_L, λ_{LA}, ψ_P

1) $\lambda_{LC} = \tan^{-1} [(1 - \epsilon)^2 \tan \lambda_L]$

2) $\mathrm{Cos}\ \lambda_{LC}$

3) $R_{SLL} = [a_e(1 - \epsilon)]/[1 - (2\epsilon - \epsilon)^2 \cos^2 \lambda_{LC}]^{1/2}$

4) $R_L = R_{SLL} + N_L + H_L$

5) $X_0 = R_L \cos \lambda_{LC}$

6) $Y_0 = 0$

7) $Z_0 = R_L \sin \lambda_{LC}$

8) $\dot{X}_0 = 0$

9) $\dot{Y}_0 = \Omega X_o$

10) $\dot{Z}_0 = 0$

11) $\omega_{PI} = 0$

12) $\omega_{YA} = 0$

13) $\mu_{LD} = \mu_{LA} - \mu_L$

14) $\mathrm{Cos}\ \lambda_{LA}$

15) $\mathrm{Sin}\ \lambda_{LA}$

16) $\mathrm{Cos}\ \mu_{LD}$

17) $\mathrm{Sin}\ \mu_{LD}$

18) $\mathrm{Cos}\ \psi_P$

(continued)

19) $\text{Sin}\ \psi_P$

20) $R_{0X_0} = \cos\lambda_{LA}\cos\mu_{LD}$

21) $R_{0Y_0} = \cos\lambda_{LA}\sin\mu_{LD}$

22) $R_{0Z_0} = \sin\lambda_{LA}$

23) $Y_{AX_0} = +\cos\psi_P\sin\lambda_{LA}\cos\mu_{LD} + \sin\psi_P\sin\mu_{LD}$

24) $Y_{AY_0} = +\cos\psi_P\sin\lambda_{LA}\sin\mu_{LD} - \sin\psi_P\cos\mu_{LD}$

25) $Y_{AZ_0} = -\cos\psi_P\cos\lambda_{LA}$

26) $P_{IX_0} = -\sin\psi_P\sin\lambda_{LA}\cos\mu_{LD} + \cos\psi_P\sin\mu_{LD}$

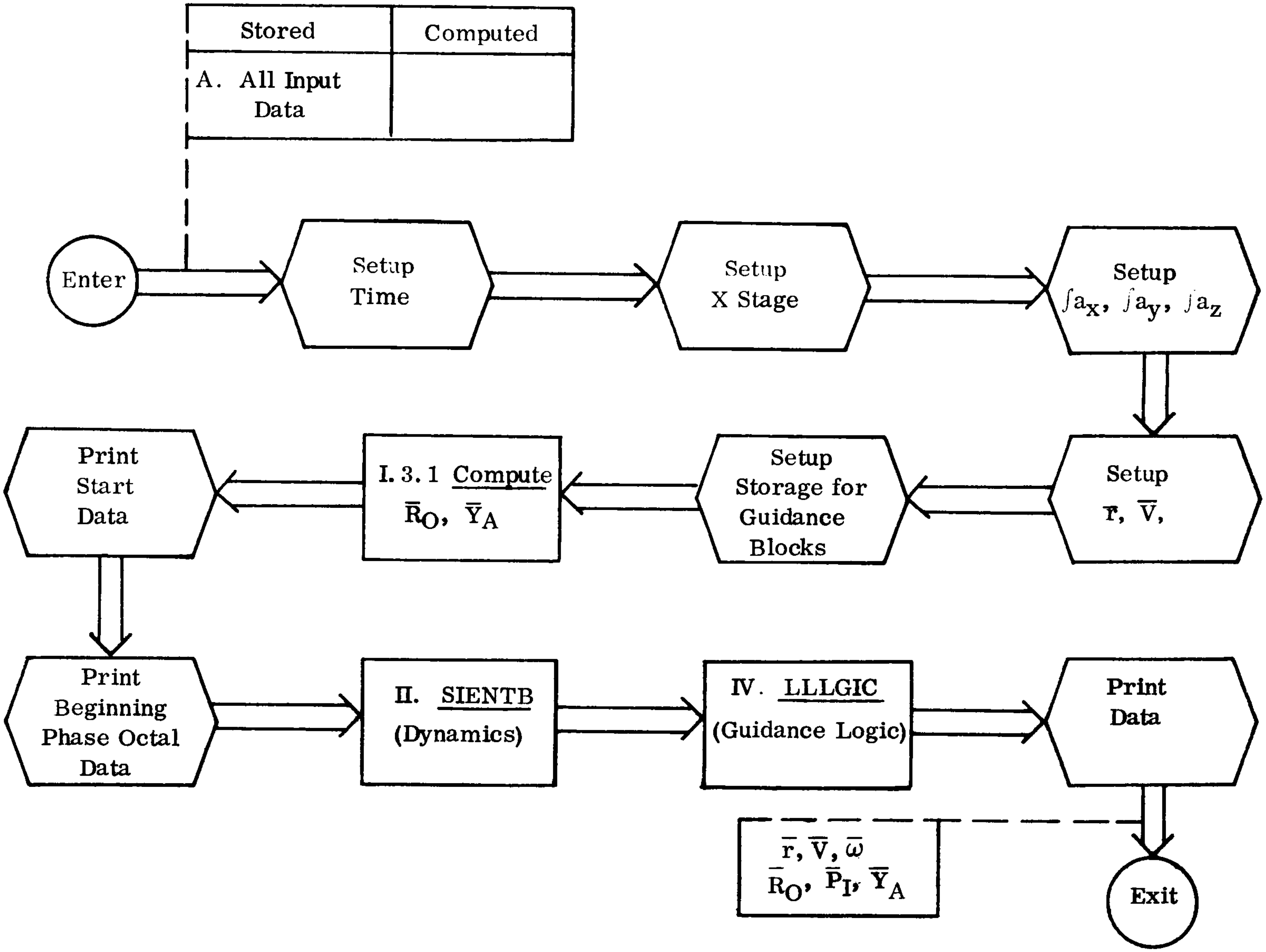
27) $P_{IY_0} = -\sin\psi_P\sin\lambda_{LA}\sin\mu_{LD} - \cos\psi_P\cos\mu_{LD}$

28) $P_{IZ_0} = \sin\psi_P\cos\lambda_{LA}$

$\vec{P}_{I_0} = (\vec{Y}_{A_0} \times \vec{R}_{0_0})$

29) $\omega_{R0} = 0$

30) $$[A_4] = \begin{bmatrix} P_{IX_0} & P_{IY_0} & P_{IZ_0} \\ Y_{AX_0} & Y_{AY_0} & Y_{AZ_0} \\ R_{OX_0} & R_{OY_0} & R_{OZ_0} \end{bmatrix}$$


Stored
Computed
A. All Input Data
Enter
Setup Time
Setup X Stage
Setup $\int a_x, \int a_y, \int a_z$
Setup $\overline{r}, \overline{V},$
Setup Storage for Guidance Blocks
I. 3. 1 Compute $\overline{R}_O, \overline{Y}_A$
Print Start Data
Print Beginning Phase Octal Data
II. SIENTB (Dynamics)
IV. LLLGIC (Guidance Logic)
Print Data
$\overline{r}, \overline{V}, \overline{\omega}$
$\overline{R}_O, \overline{P}_I, \overline{Y}_A$
Exit

I. 3 H-START (SPECIAL START ROUTINE)

I.3.1 Compute $\overline{R}_0$ and $\overline{Y}_A$

Output: $\overline{R}_0$, $\overline{Y}_A$

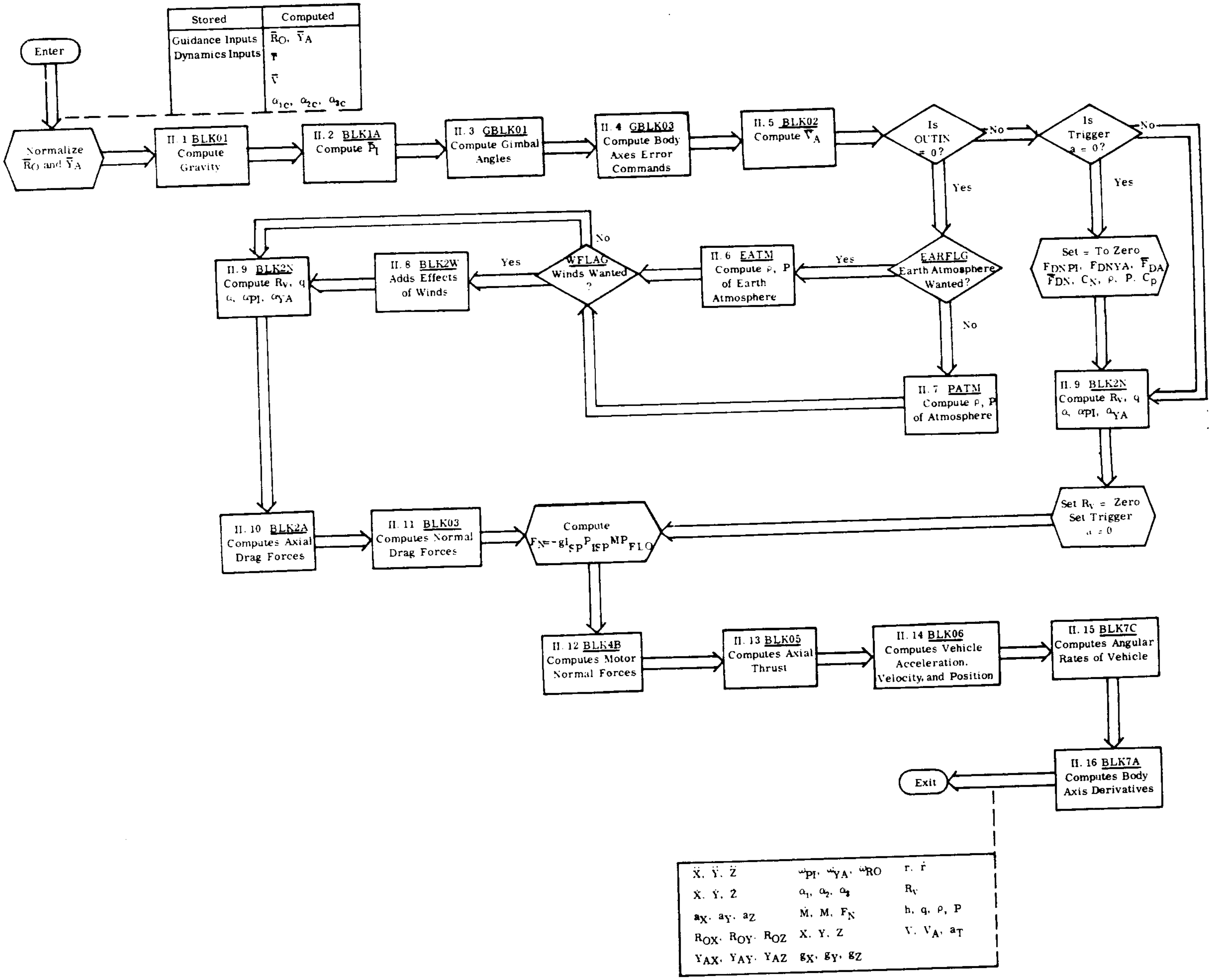
Input: α_1, α_2, α_3, $\overline{P}_{I_0}$, $\overline{Y}_{A_0}$, $\overline{R}_{0_0}$

1. $$\overline{R}_0 = (\sin\alpha_1 \sin\alpha_3 + \cos\alpha_1 \sin\alpha_2 \cos\alpha_3)\,\overline{P}_{I_0} + (-\cos\alpha_1 \sin\alpha_3 + \sin\alpha_1 \sin\alpha_2 \cos\alpha_3)\,\overline{Y}_{A_0} + (\cos\alpha_2 \cos\alpha_3)\,\overline{R}_{0_0}$$

2. $$\overline{Y}_A = (-\sin\alpha_1 \cos\alpha_3 + \cos\alpha_1 \sin\alpha_2 \sin\alpha_3)\,\overline{P}_{I_0} + (\cos\alpha_1 \cos\alpha_3 + \sin\alpha_1 \sin\alpha_2 \sin\alpha_3)\,\overline{Y}_{A_0} + (\cos\alpha_2 \sin\alpha_3)\,\overline{R}_{0_0}$$

4.2.2 DYNAMICS - BLOCK II

4.2.2.1 LEVEL II FLOW CHART


Stored
Computed
Guidance Inputs
Dynamics Inputs
$\bar{R}_O$, $\bar{Y}_A$
$\bar{r}$
$\bar{V}$
α_{1c}, α_{2c}, α_{3c}
Enter
Normalize $\bar{R}_O$ and $\bar{Y}_A$
II. 1 BLK01 Compute Gravity
II. 2 BLK1A Compute $\bar{P}_I$
II. 3 GBLK01 Compute Gimbal Angles
II. 4 GBLK03 Compute Body Axes Error Commands
II. 5 BLK02 Compute $\bar{V}_A$
Is OUTIN = 0?
No
Yes
Is Trigger a = 0?
No
Yes
Set = To Zero F_{DNPI}, F_{DNYA}, $\bar{F}_{DA}$ $\bar{F}_{DN}$, C_N, ρ, P, C_p
EARFLG Earth Atmosphere Wanted?
Yes
No
II. 6 EATM Compute ρ, P of Earth Atmosphere
II. 7 PATM Compute ρ, P of Atmosphere
WFLAG Winds Wanted ?
Yes
No
II. 8 BLK2W Adds Effects of Winds
II. 9 BLK2N Compute R_V, q α, α_{PI}, α_{YA}
II. 9 BLK2N Compute R_V, q α, α_{PI}, α_{YA}
Set R_V = Zero Set Trigger a = 0
II. 10 BLK2A Computes Axial Drag Forces
II. 11 BLK03 Computes Normal Drag Forces
Compute $F_N = -g I_{SP} P_{ISP} MP_{FLO}$
II. 12 BLK4B Computes Motor Normal Forces
II. 13 BLK05 Computes Axial Thrust
II. 14 BLK06 Computes Vehicle Acceleration, Velocity, and Position
II. 15 BLK7C Computes Angular Rates of Vehicle
II. 16 BLK7A Computes Body Axis Derivatives
Exit
$\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$
$\dot{X}$, $\dot{Y}$, $\dot{Z}$
a_X, a_Y, a_Z
R_{OX}, R_{OY}, R_{OZ}
Y_{AX}, Y_{AY}, Y_{AZ}
ω_{PI}, ω_{YA}, ω_{RO}
α_1, α_2, α_3
$\dot{M}$, M, F_N
X, Y, Z
g_X, g_Y, g_Z
r, $\dot{r}$
R_V
h, q, ρ, P
V, V_A, a_T

BLOCK II. S1ENTB (DYNAMICS LOGIC AND EVALUATION OF INTEGRANDS)

4.2.2.1.1 Compute Gravity (II. 1)

The calculation of the gravitational accelerations which act on the vehicle is done by computing the gradient of the gravitational potential function which includes the spherical zonal harmonics of the zero, second, third, and fourth order. The functional representation employed is in terms of the Legendre function of the first kind and the Legendre associated function of the first kind. The model is extenable, should it prove worthwhile to do so, to higher order zonal harmonics and even tesseral harmonics.

This block is also used to calculate the altitude, H, which is measured from the surface of the reference ellipsoid along a line from the center of the planet to the vehicle.

Detailed equations follow.

II. 1 BLOCK 01

Output: $g_X, g_Y, g_Z, R_{SL}, H, \lambda, \mathrm{Cos}\,\lambda, \dot{r}, \ddot{r}$

Input: $X, Y, Z, \epsilon, GM, a_e, C_0, C_2, C_3, C_4$

1) $r = [X^2 + Y^2 + Z^2]^{1/2}$

2) $\mathrm{Cos}\,\lambda = [(X^2 + Y^2)^{1/2}]/r$

3) $R_{SL} = [a_e(1 - \epsilon)]/[1 - (2\epsilon - \epsilon^2) \cos^2 \lambda]^{1/2}$

4) $H = r - R_{SL}$

5) $P_n(\xi) = \left(\frac{1}{2^n\ n!}\right) \left[\frac{d^n (\xi^2 - 1)^n}{d^{\xi\, n}}\right]$

$n = 0, 1, 2, 3, 4, 5$

$P_0(\xi) = 1$

$P_n(1) = 1$

$\xi = \sin \lambda$

$P_0(\xi) = 1$

$P_1(\xi) = \xi$

$P_2(\xi) = 3/2\xi^2 - 1/2$

$P_3(\xi) = 5/2\xi^3 - 3/2\,\xi$

$P_4(\xi) = 35/8\xi^4 - 15/4\,\xi^2 + 3/8$

$P_5(\xi) = 63/8\,\xi^5 - 35/4\,\xi^3 + 15/8\,\xi$

$C_0 = 1$

$C_1 = 0$

$C_2 = -2/3\ Jg$

$C_3 = 2/5\ Hg$

$C_4 = 8/35\ Dg$

(continued)

6) $P'_n(\xi) = (1 - \xi)^{1/2} \dfrac{d(P_n)}{d\xi}$

$n = 1, 2, 3, 4, 5$

$\xi = \sin \lambda$

$C_1 = 0$

$P'_1(\xi) = (1 - \xi^2)^{1/2}$

$P'_2(\xi) = 3\,\xi\,(1 - \xi^2)^{1/2}$

$P'_3(\xi) = 3/2\,(5\,\xi^2 - 1)\,(1 - \xi^2)^{1/2}$

$P'_4(\xi) = 5/2\,(7\,\xi^3 - 3\xi)\,(1 - \xi^2)^{1/2}$

$P'_5(\xi) = 1/8\,(315\,\xi^4 - 210\,\xi^2 + 15)\,(1 - \xi^2)^{1/2}$

7) $g_X = -(GM/r^2)(X/r) \sum_{n=0}^{4} C_n \left(\frac{a_e}{r}\right)^n P'_{n+1}(\xi)/P'_1(\xi)$

8) $g_Y = -(GM/r^2)(Y/r) \sum_{n=0}^{4} C_n \left(\frac{a_e}{r}\right)^n P'_{n+1}(\xi)/P'_1(\xi)$

9) $g_Z = -(GM/r^2) \sum_{n=0}^{4} (n+1)\, C_n \left(\frac{a_e}{r}\right)^n P_{n+1}(\xi)$

10) $\dot{r} = [(X)(\dot{X}) + (Y)(\dot{Y}) + (Z)(\dot{Z})]/r$

11) $\ddot{r} = [(X)(\ddot{X}) + (Y)(\ddot{Y}) + (Z)(\ddot{Z}) + (\dot{X})^2 + (\dot{Y})^2 + (\dot{Z})^2 - (\dot{r})^2]/r$

4.2.2.1.2 Compute $\overline{P}_I$ - (II.2)

The pitch vector of the vehicle is computed from the roll and yaw vectors to preserve orthogonality.

II. 2 BLK 1A

Output: $\bar{P}_I$

Input: $\bar{R}_0$, $\bar{Y}_A$

1) $P_{IX} = - R_{0Y} Y_{AZ} + R_{0Z} Y_{AY}$

2) $P_{IY} = - R_{0Z} Y_{AX} + R_{0X} Y_{AZ}$

3) $P_{IZ} = - R_{0X} Y_{AY} + R_{0Y} Y_{AX}$

4.2.2.1.3 GBLK01 - (II. 3)

This block computes the platform gimbal angles where α_1 is the inner gimbal angle, α_2 is the middle gimbal angle, and α_3 is the outer gimbal angle.

II. 3 GBLKO1

Input: $\overline{P}_I$, $\overline{P}_{I_0}$, $\overline{Y}_A$, $\overline{Y}_{A_0}$, $\overline{R}_0$, $\overline{R}_{0_0}$

Output: α_1, α_2, α_3

$$\alpha_1 = \tan^{-1}\left[\frac{(\overline{P}_I \cdot \overline{Y}_{A_0})}{\overline{P}_I \cdot \overline{P}_{I_0}}\right] \qquad K\alpha u_k - 2\pi \le \alpha_1 \le K\alpha u_k$$

$$\alpha_2 = \sin^{-1}[-(\overline{P}_I \cdot \overline{R}_{0_0}] \qquad -\frac{\pi}{2} \le \alpha_2 \le \frac{\pi}{2}$$

$$\alpha_3 = \tan^{-1}\left[\frac{(\overline{Y}_A \cdot \overline{R}_{0_0})}{(\overline{R}_0 \cdot \overline{R}_{0_0}}\right] \qquad K\gamma u_k - 2\pi \le \alpha_3 \le K\gamma u_k$$

$K\alpha u_k$ and $K\gamma u_k$ given in rad.

4.2.2.1.4 GBLK03 - (II.4)

This block computes the error commands about the roll, pitch, and yaw axis.

II.4 GBLKO3

Input: α_{1c}, α_{2c}, α_{3c}, α_1, α_2, α_3, β_j

Output: φ'_{EC}, ψ'_{EC}, θ'_{EC}

$$\alpha'_{1\epsilon} = \alpha_{1c} - \alpha_1$$

$$\alpha'_{2\epsilon} = \alpha_{2c} - \alpha_2$$

$$\alpha'_{3\epsilon} = \alpha_{3c} - \alpha_3$$

$$\varphi'_{EC} = (\cos \alpha_2)(\cos \alpha_3)\alpha_{1\epsilon} - (\sin \alpha_3)\alpha_{2\epsilon} \qquad |\varphi'_{EC}| \leq \beta_j$$

$$\psi'_{EC} = (\cos \alpha_2)(\sin \alpha_3)\alpha_{1\epsilon} + (\cos \alpha_3)\alpha_{2\epsilon} \qquad |\psi'_{EC}| \leq \beta_j$$

$$\theta'_{EC} = -(\sin \alpha_2)\alpha_{1\epsilon} + \alpha_{3\epsilon} \qquad |\theta'_{EC}| \leq \beta_j$$

where

$$\alpha_{1\epsilon} = \alpha'_{1\epsilon} - 2\pi \qquad \alpha_{1\epsilon} > \pi$$

$$\alpha_{1\epsilon} = \alpha'_{1\epsilon} \qquad -\pi < \alpha_{1\epsilon} \leq +\pi$$

$$\alpha_{1\epsilon} = \alpha_{1\epsilon} + 2\pi \qquad \alpha_{1\epsilon} \leq -\pi$$

NOTE: φ'_{EC}, ψ'_{EC}, and θ'_{EC} are limited by β_j.

4.2.2.1.5 BLK02 - (II.5)

This block computes the velocity of the vehicle relative to planet's air mass. It is assumed that the air mass is fixed to the rotating planet.

II. 5 BLKO2 (Without Winds)

Output: $\vec{V}_A$, $|\vec{V}_A|$

Input: X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$, Ω

1. $\dot{X}_A = \dot{X} + \Omega Y$

2. $\dot{Y}_A = \dot{Y} - \Omega X$

3. $\dot{Z}_A = \dot{Z}$

4. $|\vec{V}_A| = [\dot{X}_A^2 + \dot{Y}_A^2 + \dot{Z}_A^2]^{1/2}$

4.2.2.1.6 EATM - (II.6)

This block computes the density, speed of sound, temperature, and pressure for the earth's atmosphere as a function of position. This atmospheric subroutine is based on the U.S. Standard Atmosphere, 1962.

II. 6 EATM

ATMOSPHERIC SUBROUTINE

Input: X Y Z, constants, Table

A-1 $r^2 = X^2 + Y^2 + Z^2$ (ft^2)

A-2 $r = +\sqrt{r}$ (ft)

A-3 $\cos^2 \lambda_c = 1 - \frac{Z^2}{r^2}$ (nd)

A-4 $P'_2 = \frac{1}{2}(3\frac{Z^2}{r^2} - 1)$ (nd)

A-5 $U'_f = \frac{GM'}{r}(1 + C'_2 P'_2 \frac{a'^2}{r^2}) + \frac{\Omega'^2 r^2 \cos^2 \lambda_c}{2}$ (ft^2/sec^2)

A-6 $U = U'_o - C'_{a1} U'_f$ (m^2/sec^2)

A-7 $H^{*\prime} = \frac{U}{g'_o}$ (m)

From Table, determine $H^{*\prime}_{(n)}$, $H^{*\prime}_{(n+1)}$, $T'_{(n)}$, $M'_{(n)}$, $k_{(n)}$, $L_{(n)}$, $P'_{(n)}$ by the conditions:

A-8 (a) $H^{*\prime}_{(n)} \leq H^{*\prime} \leq H^{*\prime}_{(n+1)}$

(b) $H^{*\prime}_{(n)} \longrightarrow T'_{(n)}, \quad L_{(n)}, \quad P'_{(n)}, \quad M'_{(n)}, \quad k_{(n)}$

Then

A-9 $T = T'_{(n)} + L_{(n)} [H^{*\prime} - H^{*\prime}_{(n)}]$ ($^\circ$K)

If $L_{(n)} = 0$

A-10 $$P'_{(m)} = P'_{(n)} \exp \left[- \frac{W'_o g'_o}{T_{(n)}} (H^{*\prime} - H^{*\prime}_{(n)}) \right]$$ (mb)

If $L_{(n)} \neq 0$

A-11 $$P'_{(m)} = P'_{(n)} \left[\frac{T_{(n)}}{T} \right]^{\left[\frac{W'_o g'_o}{L_{(n)}} \right]}$$ (mb)

A-12 $$\rho'_{(m)} = W'_o \frac{P'_{(m)}}{T} C'_{a5} \quad (k_g/m^3)$$

A-13 $$M' = M'_{(n)} - k_{(n)} (H^{*\prime} - H^{*\prime}_{(n)}) \quad (nd)$$

For $H^{*\prime} \leq 88.743$ (Km)

A-14 $$C_s = C'_{a2} (\gamma' T)^{1/2} \quad (ft/sec)$$

For $H^{*\prime} > 88.743$ (Km)

A-15 $$C_s = \bar{C}'_s \quad (ft/sec)$$

A-16 $$\rho = C'_{a3} \rho'_{(m)} \quad (slugs/ft^3)$$

A-17 $$P = C'_{a4} P'_{(m)} \quad (lbs/in^2)$$

4.2.2.1.7 PATM - (II.7)

This block computes density and speed of sound for the atmosphere of any planet.

II.7 PATM

Output: ρ, C, P

Input: H, β_i, ρ_o, C, P_o

1) $\rho = \rho_o e^{-\beta_i H}$

2) C = constant

3) $P = P_o e^{-\beta_i H}$

4.2.2.1.8 BLK2W - (II.8)

This block computes the effect of atmospheric winds. The wind velocities are tabular functions of altitude with linear interpolation in altitude to determine current wind velocity. The perturbation factors, P_N and P_E, are used to change magnitude and direction of the wind velocities.

II. 8 BLK 2W

(With Winds)
(Uses BLK 02)

*Denotes value computed in BLK 02

Output: $\vec{V}_A$, $|\vec{V}_A|$

Input: X, Y, Z, $\dot{X}_A^*$, $\dot{Y}_A^*$, $\dot{Z}_A^*$, r, H, P_N, P_E

Tables:

North Wind (W_N) vs. Altitude (H)

East Wind (W_E) vs. Altitude (H)

Note: W_N and W_E are perturbed by P_N, P_E

1) $\dot{X}_A = \dot{X}_A^* + W_N XZ/[(X^2 + Y^2)^{1/2} r] + W_E Y/[X^2 + Y^2]^{1/2}$

2) $\dot{Y}_A = \dot{Y}_A^* + W_N YZ/[(X^2 + Y^2)^{1/2} r] + W_E X/[X^2 + Y^2]^{1/2}$

3) $\dot{Z}_A = \dot{Z}_A^* - W_N [X^2 + Y^2]^{1/2}/r$

4) $|\vec{V}_A| = [\dot{X}_A^2 + \dot{Y}_A^2 + \dot{Z}_A^2]^{1/2}$

4.2.2.1.9 BLK2N - (II. 9)

The determination of aerodynamic effects is accomplished in this simulation by the evaluation of functions of Mach number, angle of attack, and dynamic pressure. This block computes these quantities, utilizing the atmospheric subroutines. The total angle of attack, α, is computed as an absolute magnitude. The sign of α_{PI} is positive when $\vec{R}_0$ is above the vector V_A, and α_{YA} is positive when the vector $\vec{V}_A$ is to the left of $\vec{R}_0$ (when viewed from above in a forward-looking sense).

II.9 BLK2N

Output: R_V, q, α, α_{PI}, α_{YA}

Input: $\overline{V}_A$, $\overline{R}_0$, $\overline{Y}_A$, ρ, C_s

1) $R_V = |\overline{V}_A| / C_s$

2) $q = 1/2 \rho \, |\overline{V}_A|^2$

3) $V_A \cos \alpha = \overline{R}_0 \cdot \overline{V}_A = R_{0X}\dot{X}_A + R_{0Y}\dot{Y}_A + R_{0Z}\dot{Z}_A$

4) $$\alpha = \tan^{-1} [(\overline{R}_0 \mathrm{~x~} \overline{V}_A) / (\overline{R}_0 \cdot \overline{V}_A)]$$
$$= \tan^{-1} \{ [(R_{0Y}\dot{Z}_A - R_{0Z}\dot{Y}_A)^2 + (R_{0Z}\dot{X}_A - R_{0X}\dot{Z}_A)^2 + (R_{0X}\dot{Y}_A - R_{0Y}\dot{X}_A)^2]^{1/2} / (V_A \cos \alpha) \}$$

5) $$\alpha_{PI} = \tan^{-1} \{ [(\overline{R}_0 \mathrm{~x~} \overline{V}_A) \cdot \overline{P}_I] / (\overline{R}_0 \cdot \overline{V}_A) \}$$
$$= \tan^{-1} \{ [(R_{0Y}\dot{Z}_A - R_{0Z}\dot{Y}_A) P_{IX} + (R_{0Z}\dot{X}_A - R_{0X}\dot{Z}_A) P_{IY} + (R_{0X}\dot{Y}_A - R_{0Y}\dot{X}_A) P_{IZ}] / V_A \cos \alpha \}$$

6) $$\alpha_{YA} = \tan^{-1} \{ [\overline{R}_0 \mathrm{~x~} \overline{V}_A) \cdot \overline{Y}_A] / (\overline{R}_0 \cdot \overline{V}_A) \}$$
$$= \tan^{-1} \{ [(R_{0Y}\dot{Z}_A - R_{0Z}\dot{Y}_A) Y_{AX} + (R_{0Z}\dot{X}_A - R_{0X}\dot{Z}_A) Y_{AY} + (R_{0X}\dot{Y}_A - R_{0Y}\dot{X}_A) Y_{AZ}] / V_A \cos \alpha \}$$

4.2.2.1.10 BLK2A - (II. 10)

For powered flight phases, the axial drag force is resolved along the negative roll axis. The axial drag coefficient is a phase-dependent tabulated function of Mach number. The reference area, S, is a phase-dependent parameter.

II. 10 BLK2A

Output: $\overline{F}_{DA}$

Input: q, S, $\overline{R}_0$, R_V

Tables:

Axial Drag Coefficient (C_A) vs. Mach No. (R_V)

1) $F_{DAX} = -q\, S\, C_A\, R_{0X}$

2) $F_{DAY} = -q\, S\, C_A\, R_{0Y}$

3) $F_{DAZ} = -q\, S\, C_A\, R_{0Z}$

4.2.2.1.11 BLK03 - (II. 11)

The normal drag coefficient, C_N, is tabulated as a function of two variables, Mach number and angle of attack, and is calculated by a bivariate interpolation process from these tables. The C_N table and the reference area, S, are phase-dependent parameters. The normal force magnitude is resolved along a vector normal to $\vec{R}_O$ and is scaled by cos α. This normal force vector is resolved into components along the reference body pitch and yaw axis to define aerodynamic moments for use in the pitch and yaw moment equations. This force acts at the center of pressure, which is also a function of Mach number and angle of attack.

II. 11 BLK03

Output: $\overline{F}_{DN}$, F_{DNPI}, F_{DNYA}

Input: α, q, S, V_A, $\overline{R}_0$, $\overline{P}_I$, $\overline{Y}_A$, R_V, $\overline{V}_A$

Tables:

Normal Drag Coefficient (C_N) vs. Mach (R_V) and Angle of Attack (α)

1) $F_{DNX} = q\,S\,[\cos\alpha\, R_{0X} - \dot{X}_A/V_A]\; C_N$

2) $F_{DNY} = q\,S\,[\cos\alpha\, R_{0Y} - \dot{Y}_A/V_A]\; C_N$

3) $F_{DNZ} = q\,S\,[\cos\alpha\, R_{0Z} - \dot{Z}_A/V_A]\; C_N$

4) $F_{DNPI} = \overline{P}_I \cdot \overline{F}_{DN} = P_{IX}F_{DNX} + P_{IY}F_{DNY} + P_{IZ}F_{DNZ}$

5) $F_{DNYA} = \overline{Y}_A \cdot \overline{F}_{DN} = Y_{AX}F_{DNX} + Y_{AY}F_{DNY} + Y_{AZ}F_{DNZ}$

4.2.2.1.12 BLK 4B - (II. 12)

The motor normal force is defined to be that needed to exactly cancel the moment due to the aerodynamic normal force. The center of pressure, C_p, is a phase-dependent tabulated function of Mach number and angle of attack, and is computed by a bivariate interpolation process from the tabulated values. The vehicle's center of gravity, C_g, is a phase-dependent tabulated function of mass. The engine gimbal distance, E_G, is a phase-dependent value. These three distances are commonly defined with respect to a position reference in a "station number" fashion.

II.12 BLK4B

Output: $\overline{F}_{MN}$

Input: $\overline{F}_{DN}$, R_V, α, Mass, E_G

Tables:

Center of Gravity (C_g) vs. Mass (M)

Center of Pressure (C_p) vs. Mach (R_V) and Angle of Attack (α)

1) $d_1 = C_g - C_p$

2) $d_2 = E_G - C_g$

3) $F_{MNX} = [d_1/d_2]\ [F_{DNX}]$

4) $F_{MNY} = [d_1/d_2]\ [F_{DNY}]$

5) $F_{MNZ} = [d_1/d_2]\ [F_{DNZ}]$

4.2.2.1.13 BLK05 - (II.13)

The thrust force in the axial ($\vec{R}_0$) direction is defined to be the magnitude of the difference between F_N and the thrust forces normal to $\vec{R}_0$. This calculation also accounts for the magnitude of the motor normal force due to the roll couple force which does not appear in $\vec{F}_{MN}$ explicitly.

II. 13 BLK05

Output: $\overline{F}_{MA}, F_N$

Input: $\overline{R}_0, \overline{F}_{MN}, P, P_{ISP}, P_{FLO}, K_{REV}$

Tables:

gI_{SP} vs. Pressure (P)

MASS FLOW ($\dot{M}$) vs. EXPENDED MASS ($\int \dot{M}\,dt$)

1) $F_N = (g)\,(I_{SP})\,(P_{ISP})\,(\dot{M})\,(P_{FLO})$

2) $F_N \cos\delta = [F_N^2 - F_{MNX}^2 - F_{MNY}^2 - F_{MNZ}^2]^{1/2}\,[K_{REV}]$

3) $F_{MAX} = R_{0X} F_N \cos\delta$

4) $F_{MAY} = R_{0Y} F_N \cos\delta$

5) $F_{MAZ} = R_{0Z} F_N \cos\delta$

4.2.2.1.14 BLK06 - (II. 14)

The thrust and aerodynamic forces acting on the vehicle are divided by the vehicle mass to define the accelerations which are sensed by the accelerometers. These accelerations, together with the gravity acceleration, define the net acceleration of the vehicle which is integrated to calculate the velocity and position of the vehicle in reference PCI coordinates.

II.14 BLK06

Output: $\ddot{X}, \ddot{Y}, \ddot{Z}, \overline{a}, \dot{X}, \dot{Y}, \dot{Z}, X, Y, Z, \dot{M}, M, \overline{V}_a$

Input: $\overline{F}_{DN}, \overline{F}_{DA}, \overline{F}_{MA}, \overline{F}_{MN}, \overline{g}, M, \dot{X}_0, \dot{Y}_0, \dot{Z}_0, X_0, Y_0, Z_0$

Tables:

Mass Flow ($\dot{M}$) vs. Expended Mass ($\int \dot{M}\,dt$)

1) $a_X = [F_{DAX} + F_{DNX} + F_{MAX} + F_{MNX}]/M$

2) $a_Y = [F_{DAY} + F_{DNY} + F_{MAY} + F_{MNY}]/M$

3) $a_Z = [F_{DAZ} + F_{DNZ} + F_{MAZ} + F_{MNZ}]/M$

4) $\ddot{X} = a_X + g_X$

5) $\ddot{Y} = a_Y + g_Y$

6) $\ddot{Z} = a_Z + g_Z$

7) $\dot{X} = \dot{X}_0 + \int_0^t \ddot{X}\,dt$

8) $\dot{Y} = \dot{Y}_0 + \int_0^t \ddot{Y}\,dt$

9) $\dot{Z} = \dot{Z}_0 + \int_0^t \ddot{Z}\,dt$

10) $M = M_0 + \int_0^t \dot{M}\,dt$

11) $X = X_o + \int_o^t \dot{X}\,dt$

12) $Y = Y_o + \int_o^t \dot{Y}\,dt$

13) $Z = Z_o + \int_o^t \dot{Z}\,dt$

14) $V_{aX} = \int_o^t a_X\,dt$

15) $V_{aY} = \int_o^t a_Y\,dt$

16) $V_{aZ} = \int_o^t a_Z\,dt$

4.2.2.1.15 BLK7C - (II.15)

This block computes the vehicle rates assuming a perfect autopilot.

II. 15 BLK7C

Output: $\overline{\omega}$

Input: θ'_{EC}, ψ'_{EC}, K_θ, $K_{\dot\theta}$, K_ψ, $K_{\dot\psi}$, φ'_{EC}, K_φ, $K_{\dot\varphi}$

1) $\omega_{PI} = [K_\theta/K_{\dot\theta}]\ [\theta'_{EC}]$

2) $\omega_{YA} = [K_\psi/K_{\dot\psi}]\ [\psi'_{EC}]$

3) $\omega_{R0} = [K_\varphi/K_{\dot\varphi}]\ [\varphi'_{EC}]$

4.2.2.1.16 BLK7A - (II.16)

The instantaneous orientation of the body axes are calculated by integrating the rates of the six components of yaw and roll vectors in PCI coordinates. These vectors are normalized and the pitch vector is computed using a cross product relationship in Block II.2.

II.16 BLK7A

Output: $\overline{R}_0, \overline{Y}_A$

Input: $\overline{R}_{0_0}, \overline{Y}_{A_0}, \omega_{PI}, \omega_{YA}, \omega_{R0}$

1) $Y_{AX} = Y_{AX_0} + \int_0^t (\omega_{PI} R_{0X} - \omega_{R0} P_{IX})\, dt$

2) $Y_{AY} = Y_{AY_0} + \int_0^t (\omega_{PI} R_{0Y} - \omega_{R0} P_{IY})\, dt$

3) $Y_{AZ} = Y_{AZ_0} + \int_0^t (\omega_{PI} R_{0Z} - \omega_{R0} P_{IZ})\, dt$

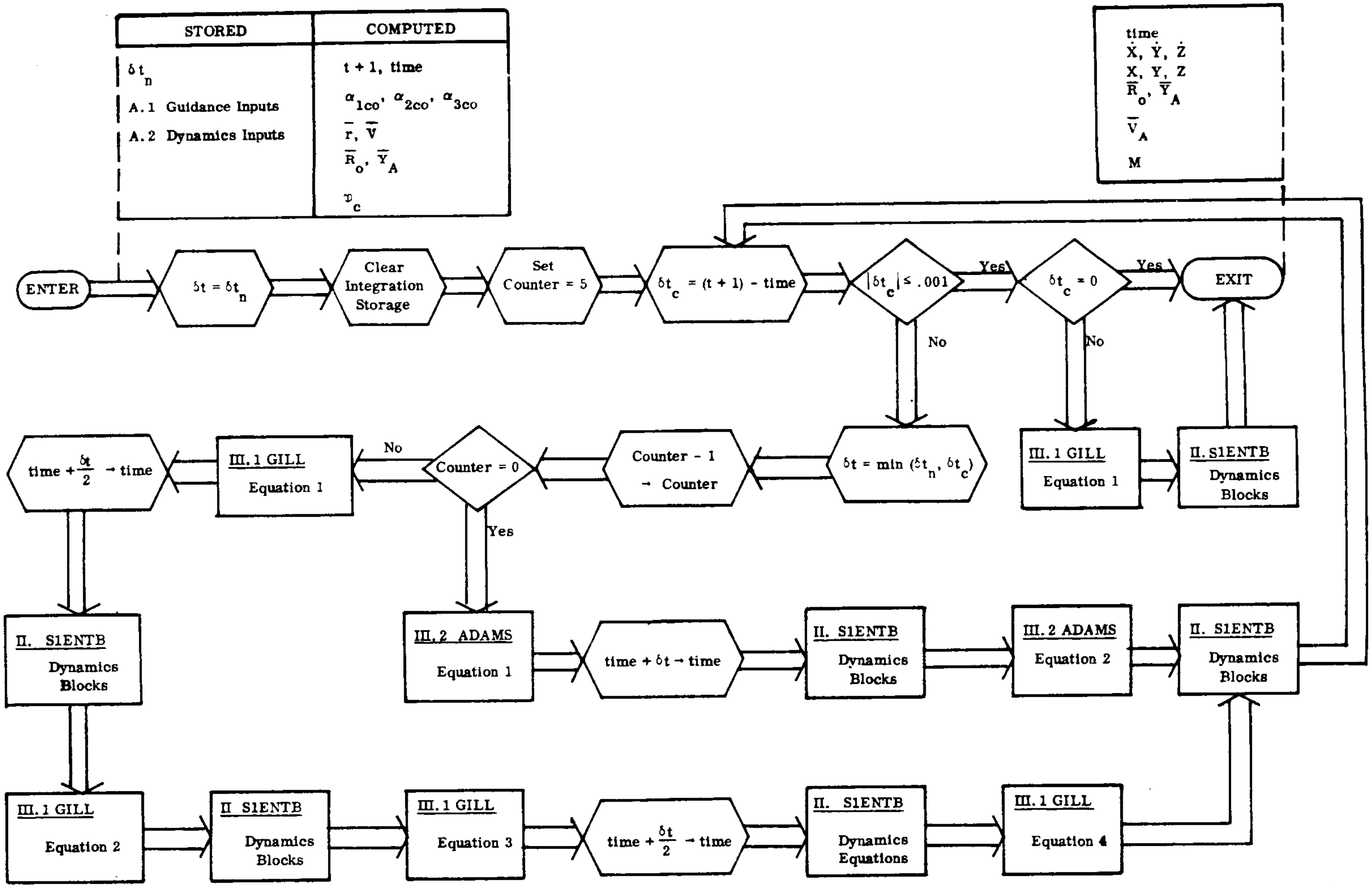
4) $R_{0X} = R_{0X_0} + \int_0^t (-\omega_{PI} Y_{AX} + \omega_{YA} P_{IX})\, dt$

5) $R_{0Y} = R_{0Y_0} + \int_0^t (-\omega_{PI} Y_{AY} + \omega_{YA} P_{IY})\, dt$

6) $R_{0Z} = R_{0Z_0} + \int_0^t (-\omega_{PI} Y_{AZ} + \omega_{YA} P_{IZ})\, dt$

4.2.3 INTEGRATION - BLOCK III

4.2.3.1 LEVEL II FLOW CHART


STORED
δt_n
A.1 Guidance Inputs
A.2 Dynamics Inputs
COMPUTED
t + 1, time
$\alpha_{1co}, \alpha_{2co}, \alpha_{3co}$
$\bar{r}, \bar{V}$
$\bar{R}_o, \bar{Y}_A$
ν_c
time
$\dot{X}, \dot{Y}, \dot{Z}$
X, Y, Z
$\bar{R}_o, \bar{Y}_A$
$\bar{V}_A$
M
ENTER
$\delta t = \delta t_n$
Clear Integration Storage
Set Counter = 5
$\delta t_c = (t + 1) - \text{time}$
$|\delta t_c| \leq .001$
Yes
No
$\delta t_c = 0$
Yes
No
EXIT
III. 1 GILL Equation 1
II. S1ENTB Dynamics Blocks
$\delta t = \min(\delta t_n, \delta t_c)$
Counter - 1 → Counter
Counter = 0
No
Yes
III. 1 GILL Equation 1
$\text{time} + \frac{\delta t}{2} \rightarrow \text{time}$
III. 2 ADAMS Equation 1
$\text{time} + \delta t \rightarrow \text{time}$
II. S1ENTB Dynamics Blocks
III. 2 ADAMS Equation 2
II. S1ENTB Dynamics Blocks
II. S1ENTB Dynamics Blocks
III. 1 GILL Equation 2
II S1ENTB Dynamics Blocks
III. 1 GILL Equation 3
$\text{time} + \frac{\delta t}{2} \rightarrow \text{time}$
II. S1ENTB Dynamics Equations
III. 1 GILL Equation 4

BLOCK III. INTG (INTEGRATION ROUTINE)

III.1 GILL

Output: Y_{n+1}

Input: δt, Y_0 derivatives

1) $$Y^{(1)}_{n+1} = Y_n + \frac{1}{2}\ \delta t\ [Y'(t, Y_n)]$$

2) $$Y^{(2)}_{n+1} = Y^{(1)}_{n+1} + \left(\frac{2 - \sqrt{2}}{2}\right) \delta t\ [Y'\ (t + \frac{\delta t}{2},\ Y^{(1)}_{n+1}) - Y'(t,\ Y_n)]$$

3) $$Y^{(3)}_{n+1} = Y^{(2)}_{n+1} + \left(\frac{2 + \sqrt{2}}{2}\right) \delta t\ [Y'\ (t + \frac{\delta t}{2},\ Y^{(2)}_{n+1}] - \delta t\ [Y'(t + \frac{\delta t}{2},\ Y^{(1)}_{n+1})] + \left(\frac{1 - \sqrt{2}}{2}\right) \delta t\ [Y'(t, Y_n)]$$

4) $$Y^{(4)}_{n+1} = Y^{(3)}_{n+1} + \frac{1}{6}\ \delta t\ [Y'(t, Y_n) + Y'(t + \delta t\ Y^{(3)}_{n+1})\ ] - \left(\frac{2 + \sqrt{2}}{2}\right) \delta t\ [Y'(t + \frac{\delta t}{2},\ Y^{(2)}_{n+1})\ ] + \left(\frac{1 + \sqrt{2}}{2}\right)\ \delta t\ [Y'(t + \frac{\delta t}{2},\ Y^{(1)}_{n+1})]$$

$$Y_{n+1} = Y^{(4)}_{n+1}$$

III. 2 ADAMS

Output: Y_{n+1}

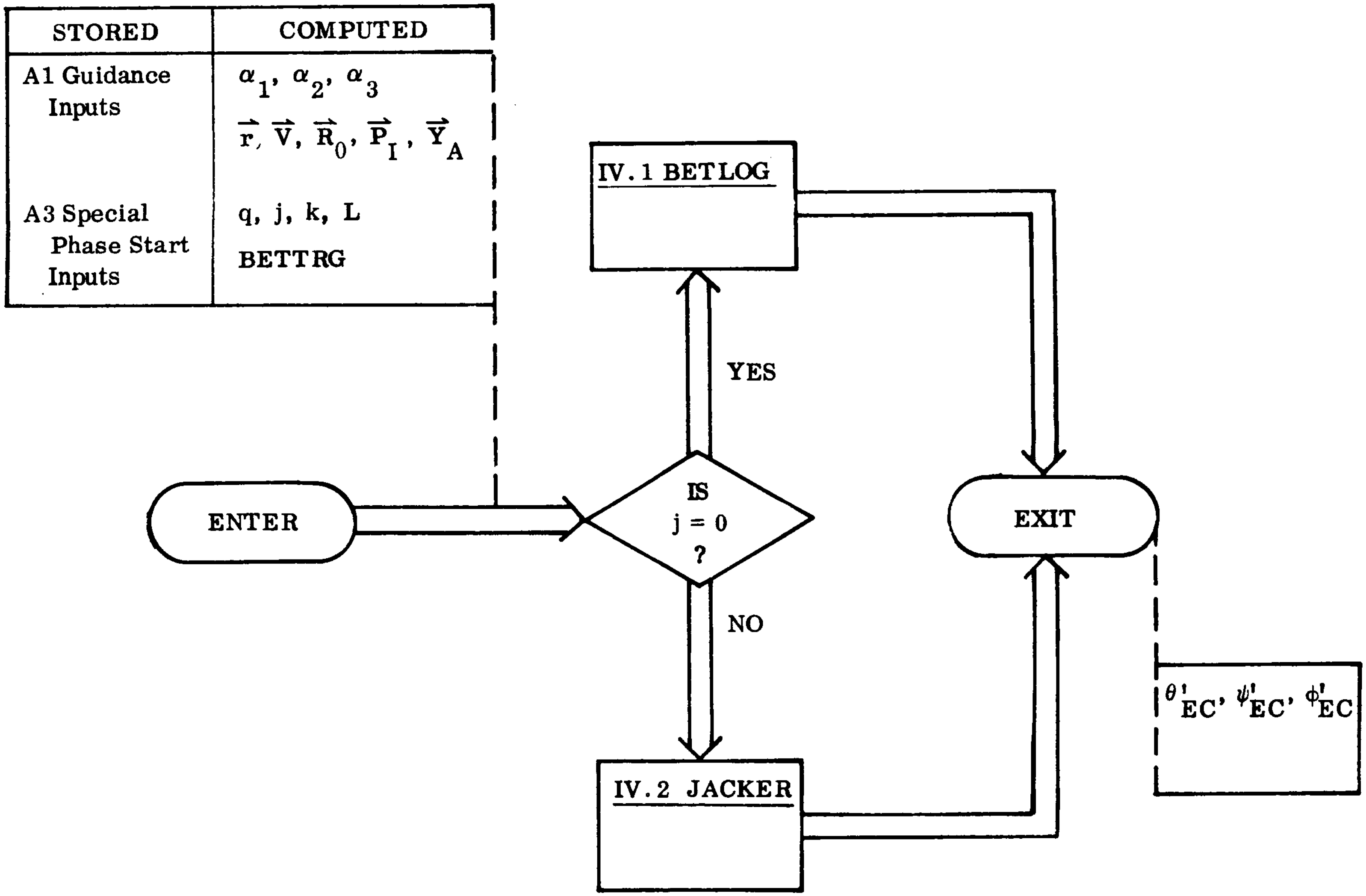
Input: δt, Y_0 derivatives

1) $Y^{(p)}_{n+1} = Y_n + \frac{\delta t}{24} [55 Y'_n - 59 Y'_{n-1} + 37 Y'_{n-2} - 9 Y'_{n-3}]$

2) $Y^{(c)}_{n+1} = Y_n + \frac{\delta t}{24} [9 Y'^{(p)}_{n+1} + 19 Y'_n - 5 Y'_{n-1} + Y'_{n-2}]$

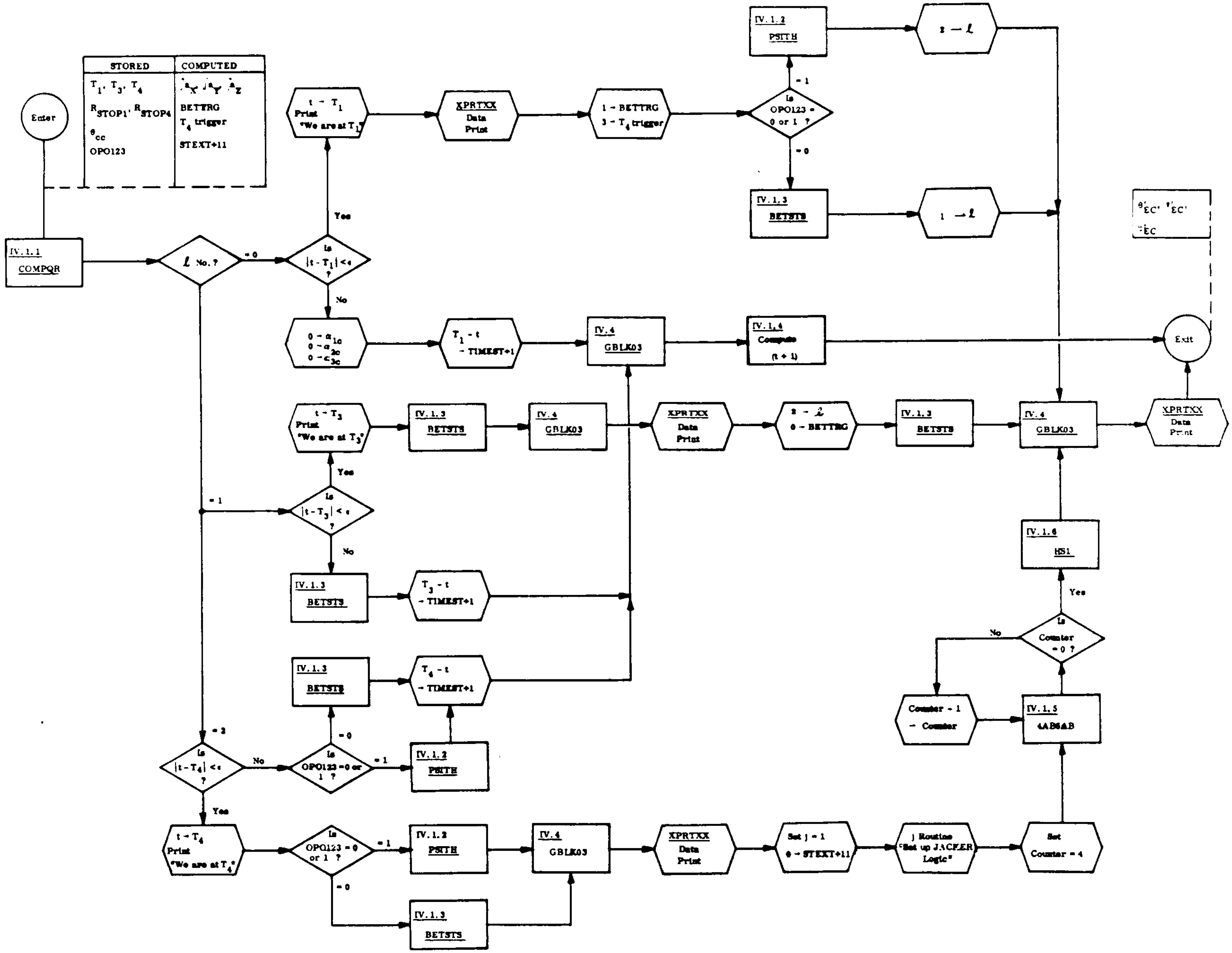
4.2.4 GUIDANCE LOGIC - BLOCK IV

4.2.4.1 LEVEL II FLOW CHART

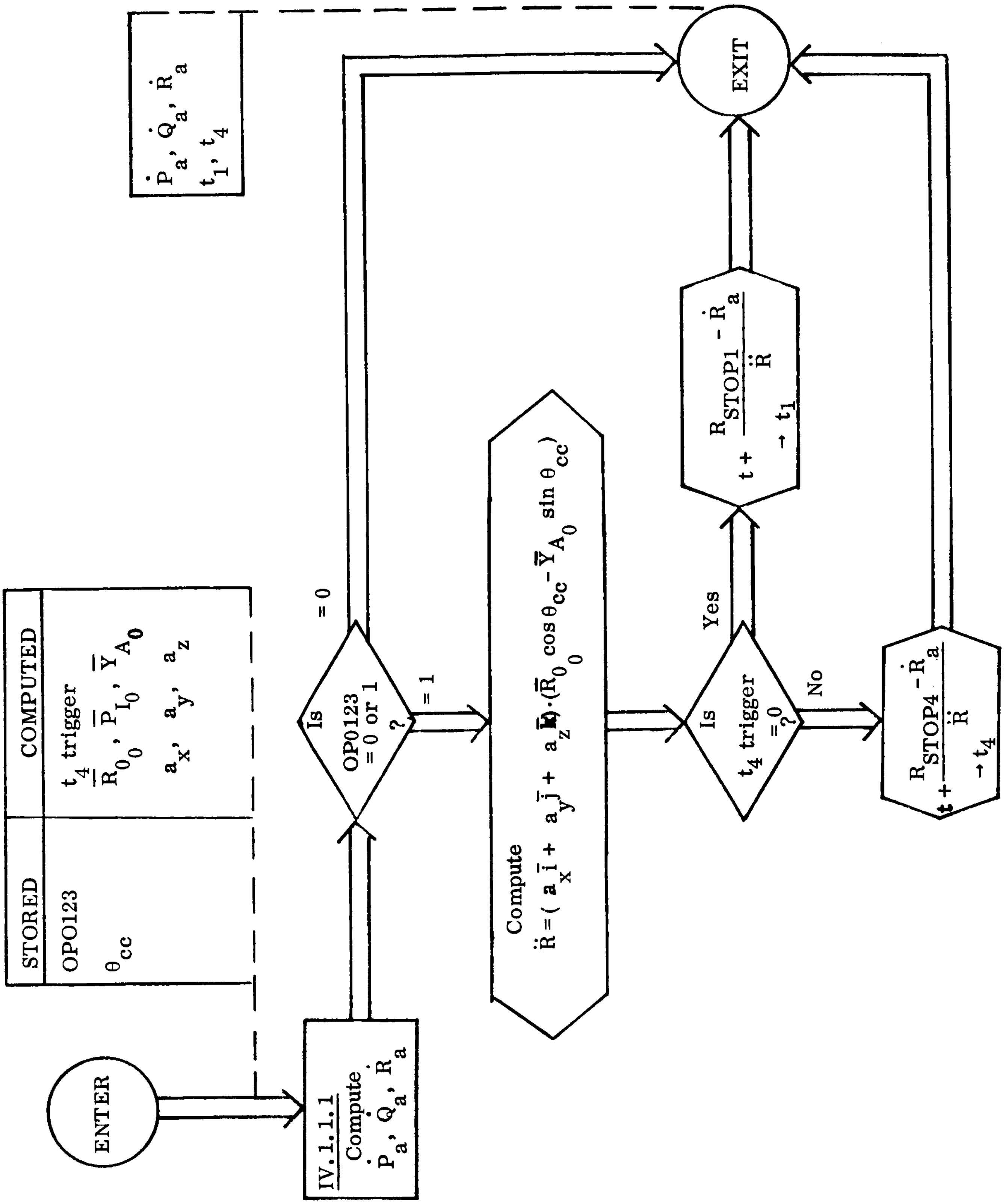


STORED	COMPUTED
A1 Guidance Inputs	$\alpha_1, \alpha_2, \alpha_3$ $\vec{r}, \vec{V}, \vec{R}_0, \vec{P}_I, \vec{Y}_A$
A3 Special Phase Start Inputs	q, j, k, L BETTRG

BLOCK IV. LLLGIC (Guidance Logic)



4.2.4.1.1 IV.1 BETLOG (GUIDANCE LOGIC FROM T_0 TO T_4)


ENTER
IV.1.1.1
Compute
$\dot{P}_a, \dot{Q}_a, \dot{R}_a$
STORED
OPO123
θ_{cc}
COMPUTED
t_4 trigger
$\overline{R}_{0_0}, \overline{P}_{I_0}, \overline{Y}_{A_0}$
a_x, a_y, a_z
Is OP0123 = 0 or 1 ?
= 0
= 1
Compute
$\ddot{R} = (a_x\bar{i} + a_y\bar{j} + a_z\bar{k}) \cdot (\overline{R}_{0_0} \cos\theta_{cc} - \overline{Y}_{A_0} \sin\theta_{cc})$
Is t_4 trigger $= 0$?
Yes
No
$t + \frac{R_{STOP1} - \dot{R}_a}{\ddot{R}}$
$\rightarrow t_1$
$t + \frac{R_{STOP4} - \dot{R}_a}{\ddot{R}}$
$\rightarrow t_4$
EXIT
$\dot{P}_a, \dot{Q}_a, \dot{R}_a$
t_1, t_4

IV.1.1 COMPQR

IV.1.1.1 Compute $\dot{P}_a$, $\dot{Q}_a$, $\dot{R}_a$

Output: $\dot{P}_a$, $\dot{Q}_a$, $\dot{R}_a$

Input: $\overline{V}_a$, $\overline{R}_{0_0}$, $\overline{P}_{I_0}$, $\overline{Y}_{A_0}$, θ_{cc}

1. $\dot{P}_a = (\overline{V}_a \cdot \overline{P}_{I_0})$

2. $\dot{Q}_a = (\overline{V}_a \cdot \overline{Y}_{A_0}) \cos \theta_{cc} + (\overline{V}_a \cdot \overline{R}_{0_0}) \sin \theta_{cc}$

3. $\dot{R}_a = (\overline{V}_a \cdot \overline{R}_{0_0}) \cos \theta_{cc} - (\overline{V}_a \cdot \overline{Y}_{A_0}) \sin \theta_{cc}$

IV.1.2 PSITH

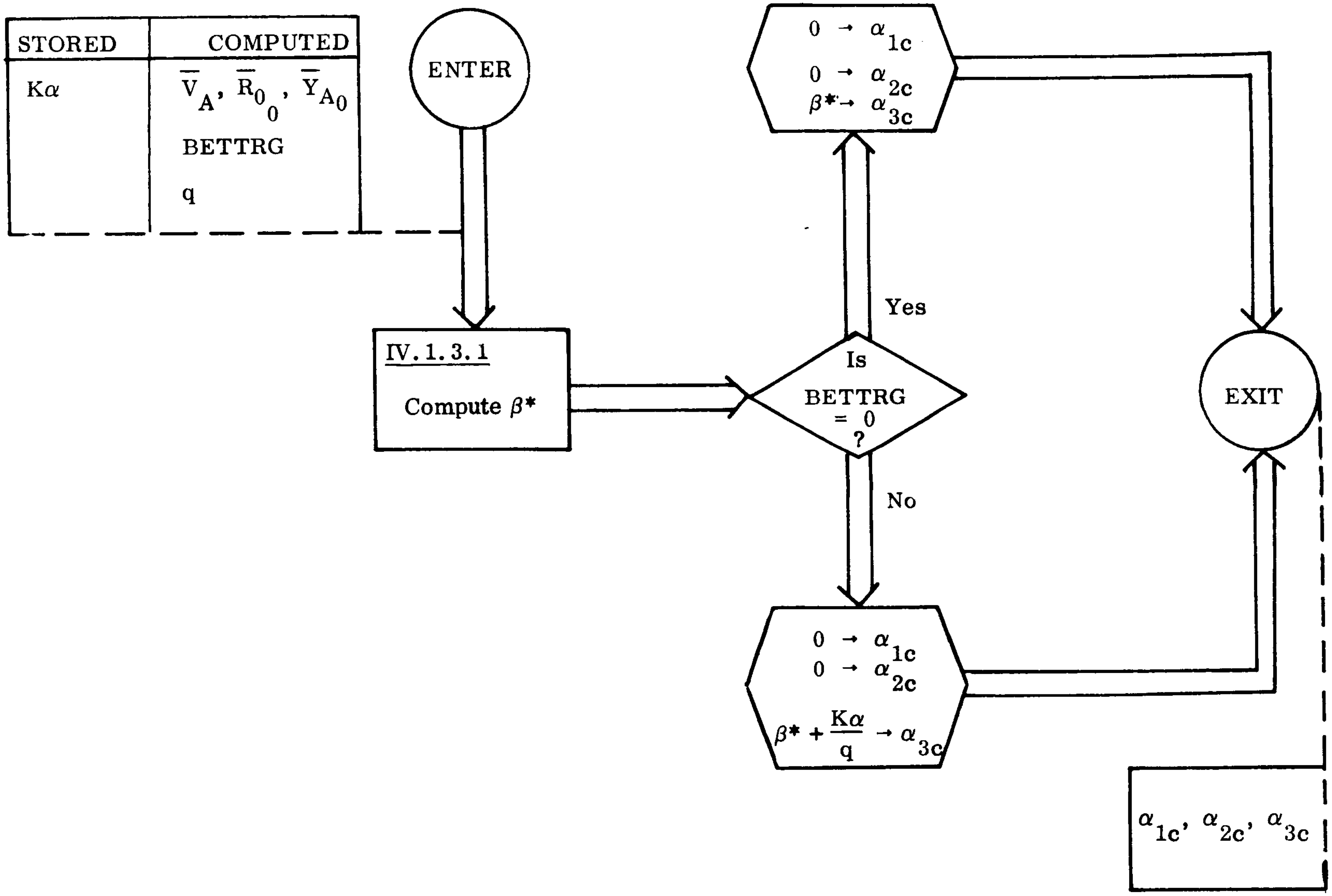
Output: α_{1c}, α_{2c}, α_{3c}

Input: $\dot{P}_a$, $\dot{Q}_a$, $\dot{R}_a$, K_{52} - K_{61}

1. $\alpha_{1c} = K_{50} \dot{P}_a + K_{51} \Sigma \dot{P}_a \Delta t$

2. $\alpha_{2c} = 0$

3. $$\alpha_{3c} = K_{52} [\dot{Q}_a - K_{53} - K_{54} (\dot{R}_a) - K_{55} (\dot{R}_a)^2 - K_{56} (\dot{R}_a)^3] + K_{58} + K_{59} (\dot{R}_a) + K_{60} (\dot{R}_a)^2 + K_{61}(\dot{R}_a)^3$$


STORED
COMPUTED
$K\alpha$
$\overline{V}_A$, $\overline{R}_{0_0}$, $\overline{Y}_{A0}$
BETTRG
q
ENTER
IV.1.3.1
Compute β^*
Is BETTRG = 0 ?
Yes
No
$0 \rightarrow \alpha_{1c}$
$0 \rightarrow \alpha_{2c}$
$\beta^* \rightarrow \alpha_{3c}$
$0 \rightarrow \alpha_{1c}$
$0 \rightarrow \alpha_{2c}$
$\beta^* + \frac{K\alpha}{q} \rightarrow \alpha_{3c}$
EXIT
α_{1c}, α_{2c}, α_{3c}

IV.1.3 BETSTS

IV.1.3.1 Compute β^*

Output: β^*

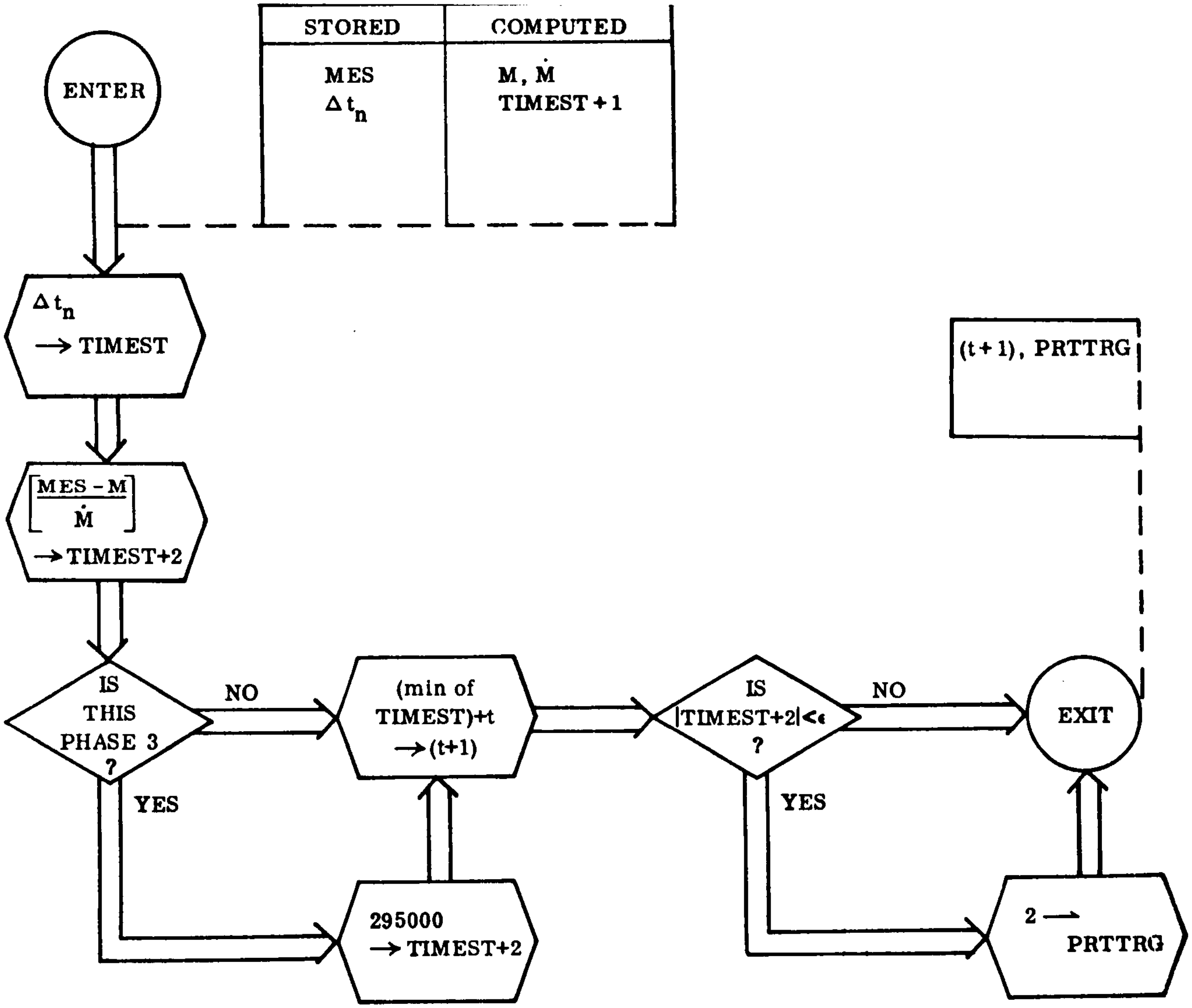
Input: $\overline{V}_A, \overline{R}_{0_0}, \overline{Y}_{A_0}$

1. $\beta^* = 0$ for $(\overline{V}_A \cdot \overline{R}_{0_0}) = 0$

2. $\beta^* = \tan^{-1} \left[\frac{(\overline{V}_A \cdot \overline{Y}_{A_0}}{(\overline{V}_A \cdot \overline{R}_{0_0})} \right]$ for $(\overline{V}_A \cdot \overline{R}_{0_0}) \neq 0$

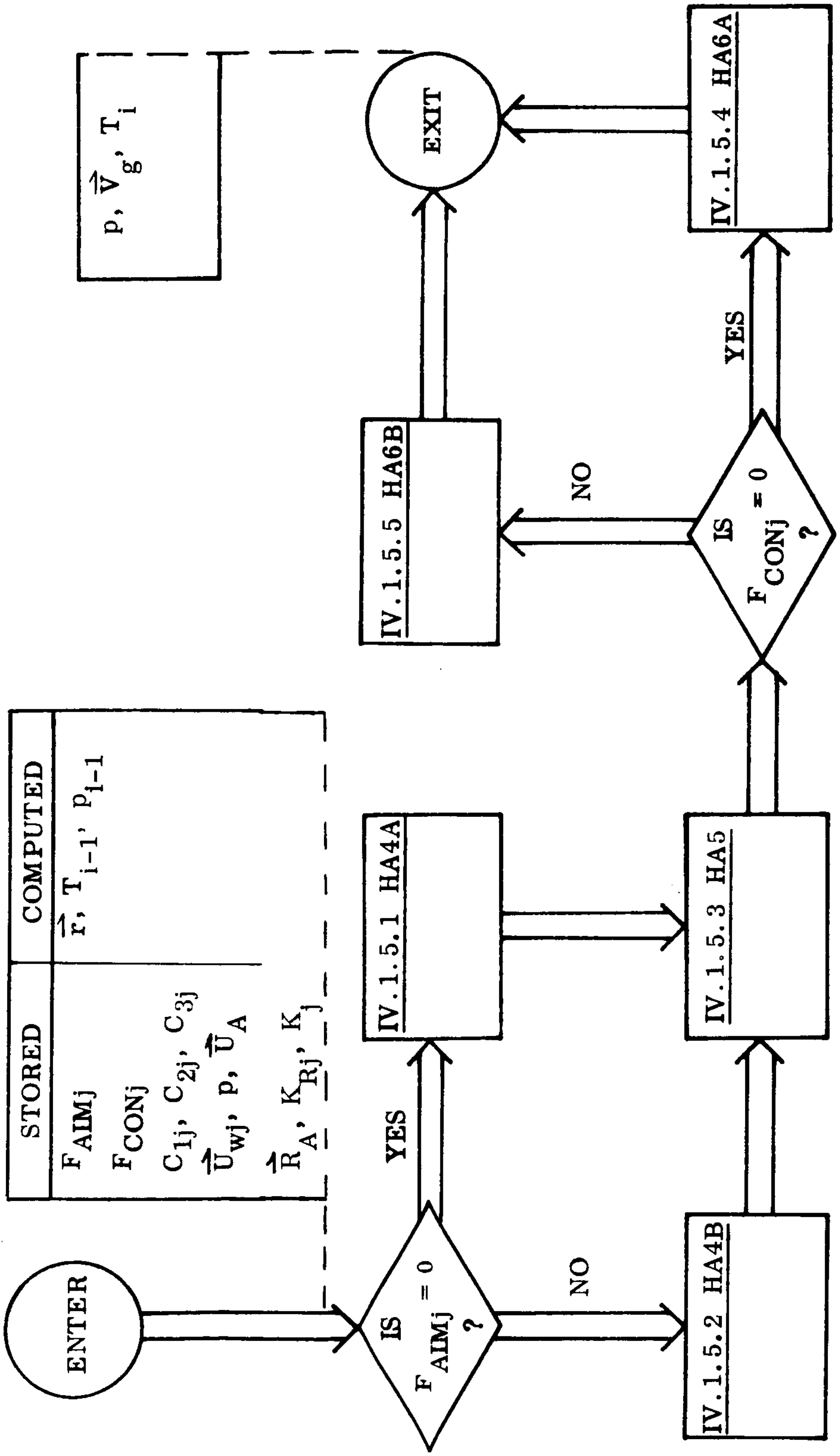
for $(\overline{V}_A \cdot \overline{Y}_{A_0}) \leq 0$ $0 < \beta^* \leq \pi$

for $(\overline{V}_A \cdot \overline{Y}_{A_0}) > 0$ $\pi < \beta^* < 2\pi$



IV.1.4 Compute (t+1)

This is a blank page



IV.1.5 4AB6AB (Explicit Guidance)

IV. 1. 5. 1 HA4A - Target or Aiming Point

Input: X_{Aj}, Y_{Aj}, Z_{Aj}, $[\text{Flag } j = F_{AIM\,j} = 0]$

Output: $\bar{U}_{Aj}$, r_{Aj}

This block is computed only once at staging or at T_4

$$\bar{U}_{Aj} = \frac{X_{Aj}\,\bar{i} + Y_{Aj}\,\bar{j} + Z_{Aj}\,\bar{k}}{\sqrt{X_{Aj}^2 + Y_{Aj}^2 + Z_{Aj}^2}}$$

$$r_{Aj} = \sqrt{X_{Aj}^2 + Y_{Aj}^2 + Z_{Aj}^2}$$

IV.1.5.2 HA4B - Target or Aiming Point

Input: t_{i-1}, T_{i-1}, C_{1j}, C_{2j}, C_{3j}, $F_{AIMj} = 1$

Output: $\overline{U}_{Aj}$, r_{Aj}

This block is computed every major computational cycle

$$\overline{U}_{Aj} = \frac{C_{1j} \cos [C_{2j} + \Omega (t_{i-1} + T_{i-1})] \overline{i} + C_{1j} \sin [C_{2j} + \Omega (t_{i-1} + T_{i-1})] \overline{j} + C_{3j} \overline{k}}{r_{Aj}}$$

$$r_{Aj} = + \sqrt{C_{1j}^2 + C_{3j}^2}$$

IV.1.5.3 HA5 - $\overline{V}_g$ Computation

Input: $\overline{U}_{Aj}, \overline{r}, r, r_{Aj}, p_j, K_{rj}, K_j, \mu, \overline{U}_{wj}, K_{VG}, \epsilon_s, \epsilon_u$

Output: $\overline{V}_g, T_i$

Note: An initial value of p_j is given every time this block has a new aiming point.

$$\overline{U}_r = \overline{r}/r$$

$$c = \overline{U}_r \cdot \overline{U}_{Aj}$$

$$s = \pm \sqrt{1 - c^2} \quad \text{sign } s = \text{sign of } \overline{U}_r \times \overline{U}_{Aj} \cdot \overline{U}_{wj}$$

$$e \cos v = p\,\mu - 1$$

$$\mu_A = 1/r_A$$

$$e \cos v_A = p\,\mu_A - 1$$

Is $s < \epsilon_s$

Yes: $e \sin v_i = e \sin v_{i-1}$

$e \sin v_{Ai} = e \sin v_{Ai-1}$

No:

$$e \sin v = \frac{(c)(e \cos v) - (e \cos v_A)}{s}$$

$$e \sin v_A = \frac{(e \cos v) - (c)\,(e \cos v_A)}{s}$$

$$e^2 = (e \sin v_A)^2 + (e \cos v_A)^2$$

IV.1.5.3 HA5 - V_g Computation - continued

$$a = \frac{p}{1 - e^2}$$

$$\dot{r} = (K_j)(e \sin v)/\sqrt{p}$$

Is $s < \epsilon_u$?

Yes: $\bar{U}_v = \bar{U}_{wj} \times \bar{U}_r$

No: $\bar{U}_v = [\bar{U}_{Aj} - (c)(\bar{U}_r)]/s$

$$\bar{V}_{REQ} = (\dot{r})\ \bar{U}_r + \frac{(K_j)(\sqrt{p})}{r}\ \bar{U}_v$$

$$\bar{V}_g = (\bar{V}_{REQ} - \bar{V})\ (K_{VG})$$

$$e \sin E = (K_{rj})(r)(\dot{r})/\sqrt{a}$$

$$e \sin E_A = (r_A)(e \sin v_A)/\sqrt{a}\sqrt{p}$$

$$E_A - E = \tan^{-1}\left[\frac{\sqrt{a}\sqrt{p}\,[r_A(e \sin v_A)(1 - r/a) - r(e \sin v)\ (1 - r_A/a)]}{e^2[\,a\,p + (c - 1)\ r\ r_A\,]}\right]$$

$$0 \leq E_A - E < 2\pi$$

$$T_i = a\sqrt{a}\ K_{rj}\ [E_A - E - e \sin E_A + e \sin E]$$

IV.1.5.4 HA6A Constraint (Constant Time of Arrival)

Input: T_{Aj}, T_i, t_i, ϵ_T, p_i, p_{i-1}, $F_{conj} = 0$, K_{13j}, K_{14j}, ϵ_p

Output: p_{i+1}

Is $|T_{Aj} - (t_i + T_i)| \leq \epsilon_T$?

a) Yes, $p_{i+1} = p_i$

b) No, Is $|p_i - p_{i-1}| \leq \epsilon_p$?

1) Yes, $p_{i+1} = p_i + [(K_{13})_j + (K_{14})_j\, p_i][T_{Aj} - (t_i + T_i)]$

2) No, Is $|(t_i + T_i) - (t_{i-1} + T_{i-1})| \leq \epsilon_T$

Yes, b1

No, $p_{i+1} = p_i + \dfrac{(p_i - p_{i-1})\,[T_{Aj} - (t_i + T_i)]}{(t_i + T_i) - (t_{i-1} + T_{i-1})}$

Note: The first time this block is entered every Δt, use equation b1.

IV.1.5.5 HA6B Constraint (Constant Flight Path Angle)

Input: $(e \sin v_A)$, $(e \cos v_A)$, (e^2), $K_{\gamma j}$, ϵ_γ, $F_{conj} = 1$, ϵ_p, K_{13j}, K_{14j}

Output: p_{i+1}

$$\gamma_A = \sin^{-1}\left[\frac{(e \sin v_A)}{\sqrt{1 + 2\,(e \cos v_A) + (e^2)}}\right] \qquad -\frac{\pi}{2} \le \gamma_A < \frac{\pi}{2}$$

Is $|K_{\gamma j} - \gamma_A| \le \epsilon_\gamma$?

a) Yes: $p_{(i+1)} = p_i$

b) No: Is $|p_i - p_{(i-1)}| \le \epsilon_p$?

1) Yes, $p_{i+1} = p_i + [K_{13j} + K_{14j}\, s]\,(K_{\gamma j} - \gamma_A)$

2) No, Is $|\gamma_{A\,i} - \gamma_{A\,i-1}| \le \epsilon_\gamma$

Yes, b1

No, $p_{(i+1)} = p_i + \dfrac{p_i - p_{(i-1)}}{\gamma_{Ai} - \gamma_{A(i-1)}}\,(K_{\gamma j} - \gamma_A)$

Note: The first time this block is entered after j is changed, use b1.

IV.1.6 HS1 - $\bar{V}_g$ Steering Commands

Input: $\bar{V}_{gj}, \bar{R}_{0_0}, \bar{P}_{I_0}, \bar{Y}_{A_0}, k_{2j}, k_{3j}, k_{4j}, k_{5j}, \bar{U}_r, \bar{V}_g, A_{\theta j}, B_{\theta j}, C_{\theta j}$

Output: $\alpha_{1c}, \alpha_{2c}, \alpha_{3c}$

$$\alpha_{1vgj} = \tan^{-1}\left[\frac{\bar{V}_g \cdot \bar{P}_{I_0}}{-\bar{V}_g \cdot \bar{Y}_{A_0}}\right] \qquad K\alpha L_k < \alpha_{1vgj} \le K\alpha u_k$$

$$\alpha_{2vgj} = 0$$

$$\bar{Y}'_A = \cos(\alpha_{1vgj})\,\bar{Y}_{A_0} - \sin(\alpha_{1vgj})\,\bar{P}_{I_0}$$

$$\alpha_{3vgj} = \tan^{-1}\left[\frac{-\bar{V}_g \cdot \bar{Y}'_A}{\bar{V}_g \cdot \bar{R}_{0_0}}\right] \qquad K\gamma L_k < \alpha_{3vgj} \le K\gamma u_k$$

$$\bar{U}_{vgj} = \bar{V}_{gj} / \sqrt{\bar{V}_{gj} \cdot \bar{V}_{gj}}$$

The preceding five equations are computed only the first iteration after j is changed and this block is entered.

$$\Delta\bar{R} = \bar{V}_g \times \bar{U}_{vgj}$$

$$\bar{P}'_I = (\cos\alpha_1)\,\bar{P}_{I_0} + (\sin\alpha_1)\,\bar{Y}_{A_0}$$

$$\Delta\alpha_{1c} = -k_{2j}\,(\Delta\bar{R} \cdot \bar{R}_{0_0}) - k_{3j}\,\Sigma\,(\Delta\bar{R} \cdot \bar{R}_{0_0})_i\,\Delta t_i$$

$$\Delta\alpha_{2c} = 0$$

$$\Delta\alpha_{3c} = -k_{4j}\,(\Delta\bar{R} \cdot \bar{P}'_I) - k_{5j}\,\Sigma\,(\Delta\bar{R} \cdot P'_I)_i\,\Delta t_i$$

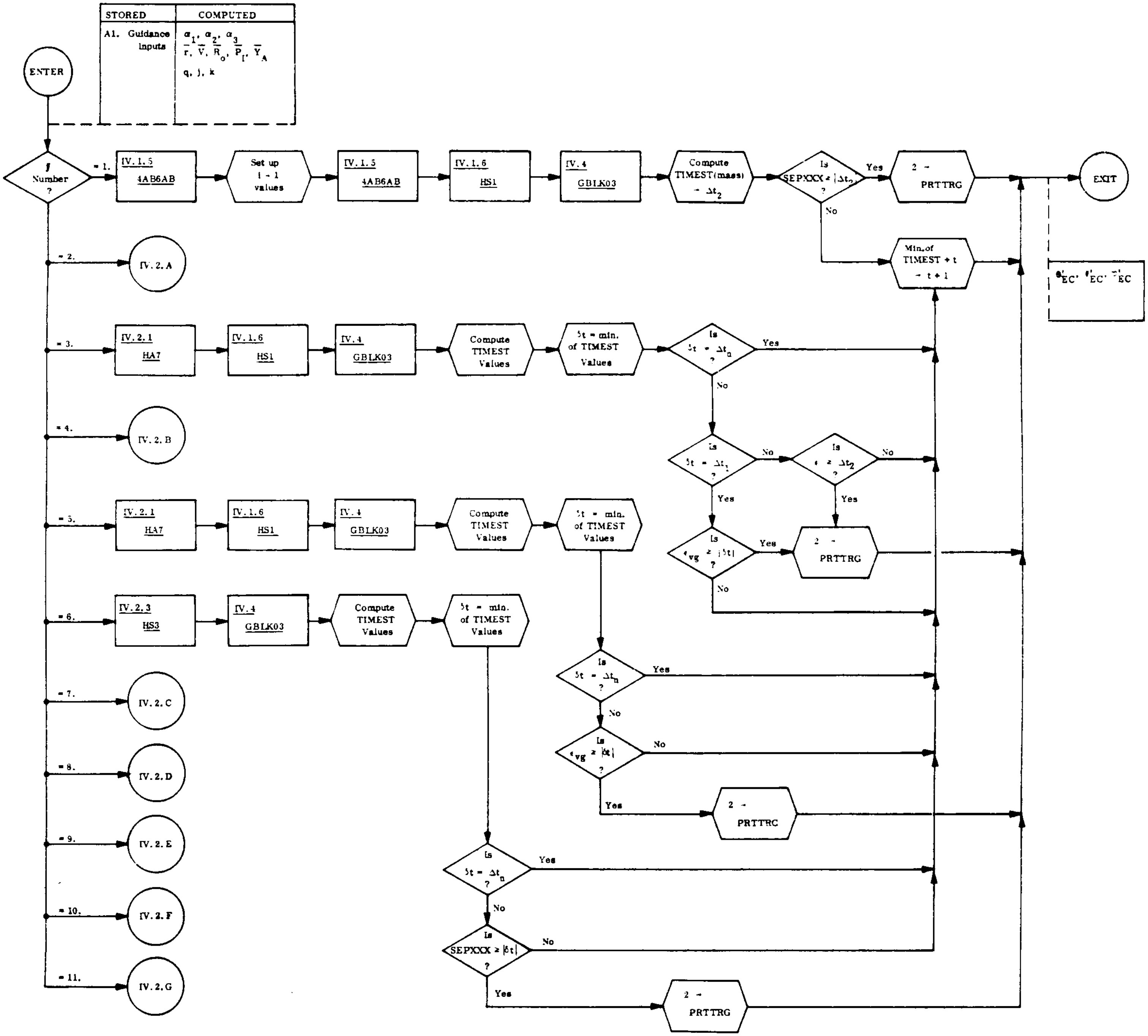
IV.1.6 HS1 - $\bar{V}_g$ Steering Commands - continued

$$\bar{\theta}_k = [A_{\theta j} + B_{\theta j}(V_g) + C_{\theta j}(V_g)^2]\ \bar{U}_r \times \bar{U}_{vgj}$$

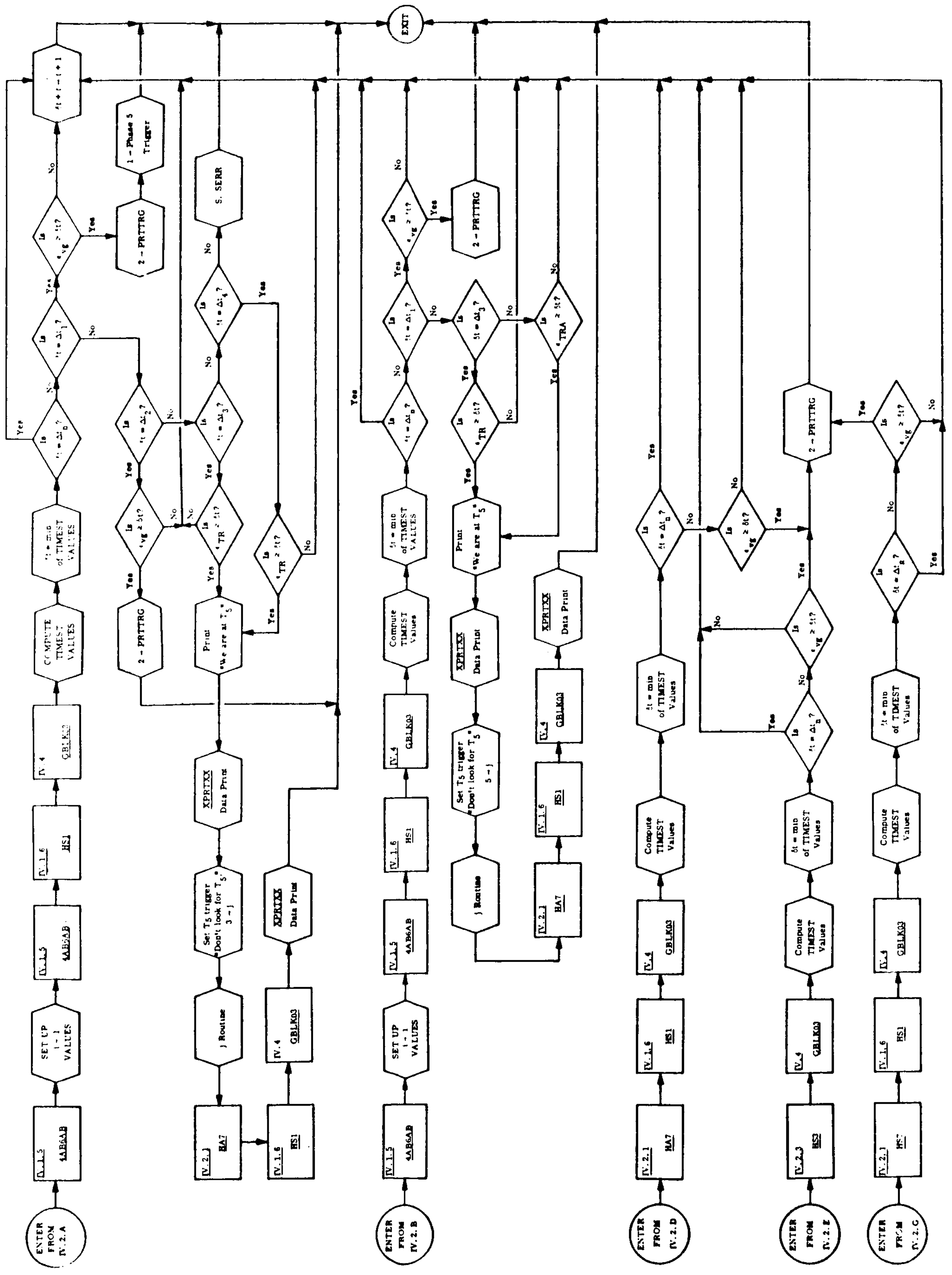
$$\alpha_{1c} = \Delta\alpha_{1c} + (\bar{\theta}_k \cdot \bar{R}_{0_0}) + \alpha_{1vgj}$$

$$\alpha_{2c} = 0$$

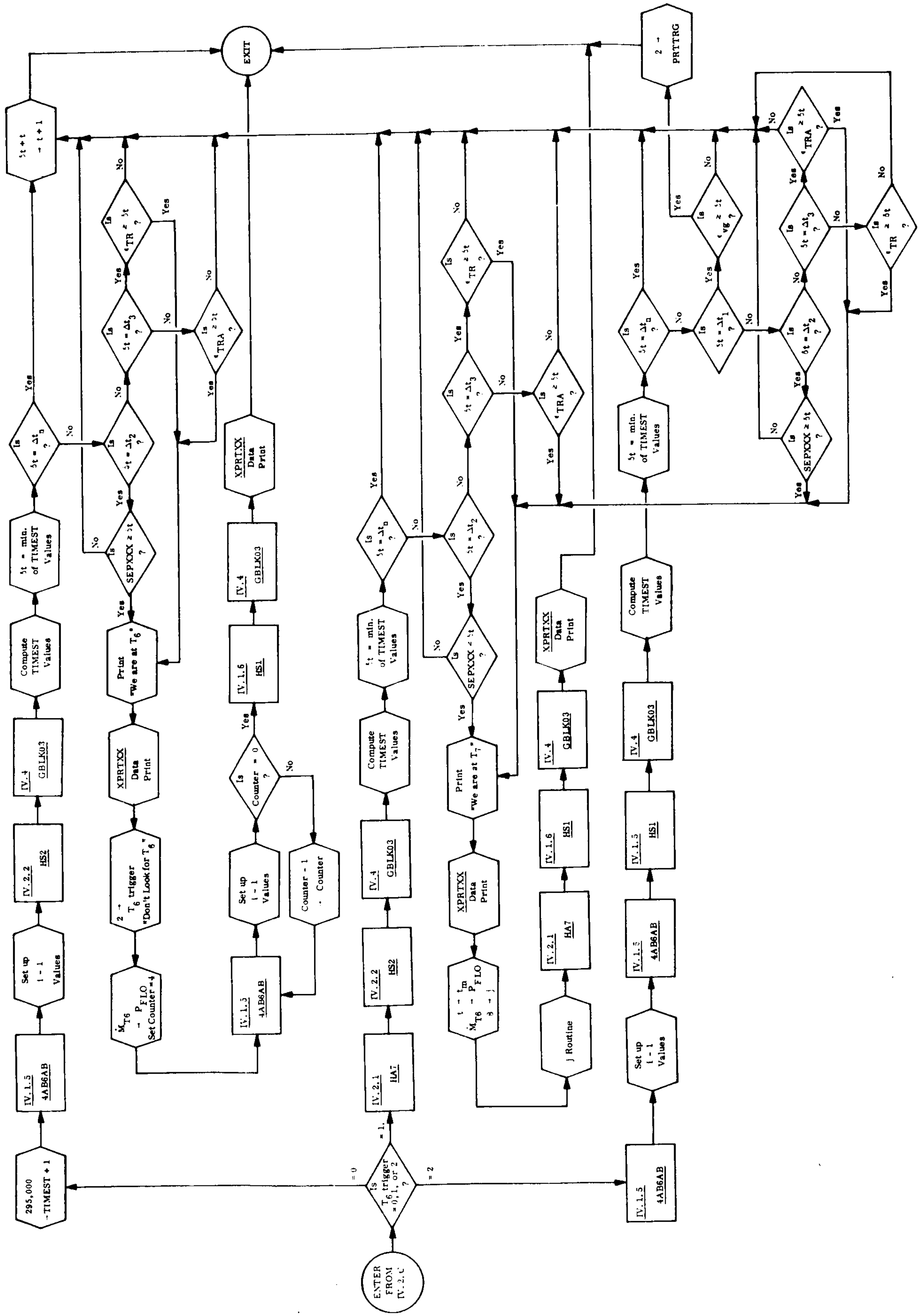
$$\alpha_{3c} = \Delta\alpha_{3c} + (\bar{\theta}_k \cdot \bar{P}'_I) + \alpha_{3vgj}$$



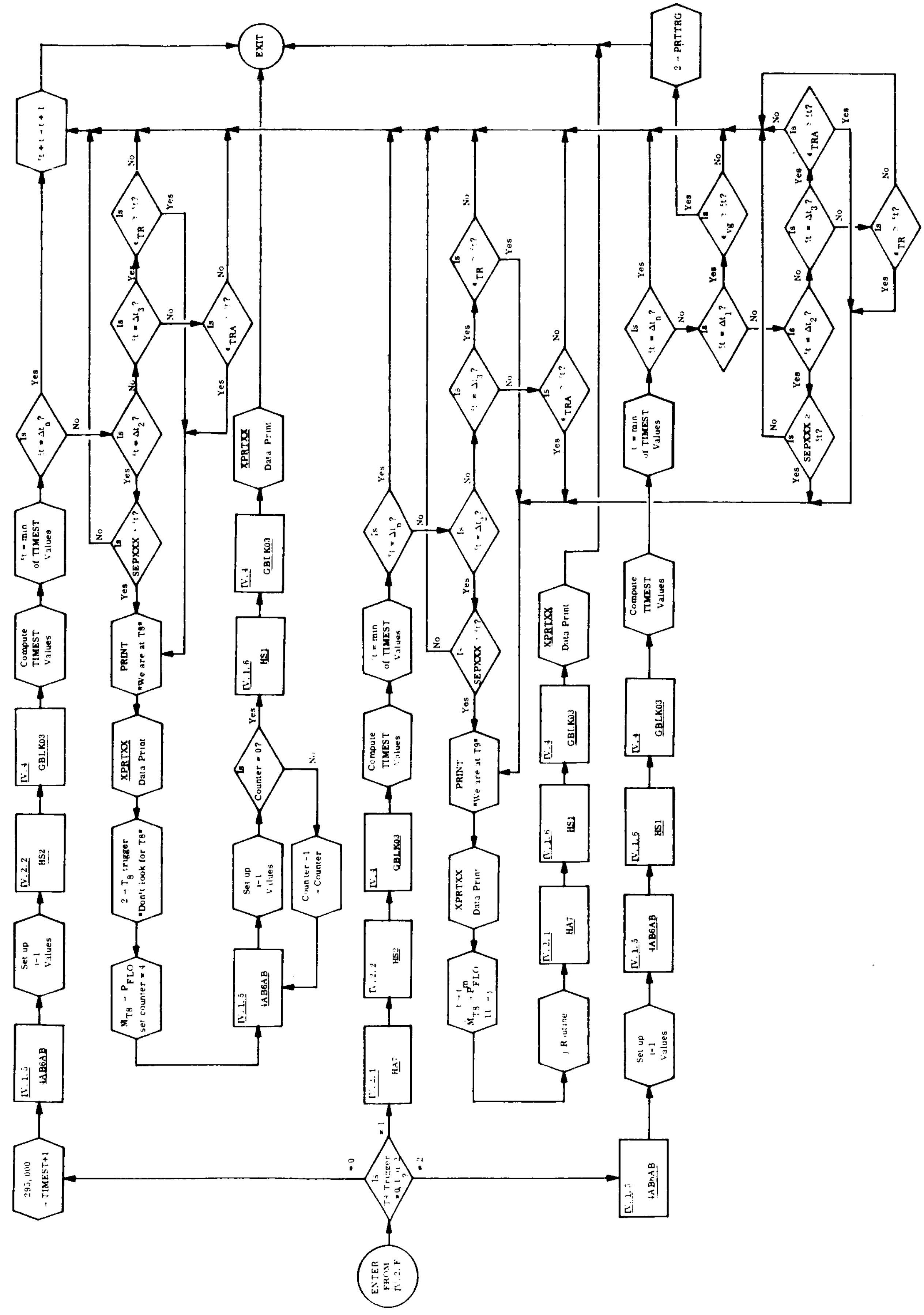
4.2.4.1.2 IV.2 JACKER (GUIDANCE LOGIC AFTER T_4)



IV.2.A, B, D, E, G JACKER (Cont'd)



IV.2.C JACKER (Cont'd)



IV.2.F JACKER (Cont'd)

IV.2.1 HA7 - Trim Steering

Input: $\overline{U}_{wj}$, $\overline{V}$, $\overline{r}$, $\dot{r}$, Curve fit constants, k_{Aj}, k_{Hj}, K_{VG}

Output: $\overline{V}_{g}$

$$\overline{U}_{r} = \frac{r}{|r|}$$

$$\overline{U}_{v} = \overline{U}_{wj} \times \overline{U}_{r}$$

$$\Delta v = (t_{i} - t_{m}) \qquad m = 5,7,9$$

$$r_{c} = A_{RCj} + B_{RCj}(\Delta v) + C_{RCj}(\Delta v)^{2}$$

$$\dot{r}_{c} = A_{RDj} + B_{RDj}(\Delta v) + C_{RDj}(\Delta v)^{2}$$

$$(r\dot{v})_{c} = A_{RVj} + B_{RVj}(\Delta v) + C_{RVj}(\Delta v)^{2}$$

$$\overline{V}_{REQ} = \dot{r}_{c}\,\overline{U}_{r} + (r\,\dot{v})_{c}\,\overline{U}_{v} + k_{Aj}\,(\dot{r}_{c} - \dot{r} + k_{Hj}\,\{r_{c} - r\})\,\overline{U}_{r}$$

$$\overline{V}_{g} = [\overline{V}_{REQ} - \overline{V}]\,[K_{VG}]$$

IV.2.2 HS2 - $\bar{V}_g$ Orientation Steering

Input: $\bar{V}_g, \bar{P}_{I_0}, \bar{R}_{0_0}, \bar{Y}_{A_0}$

Output: $\alpha_{1c}, \alpha_{2c}, \alpha_{3c}$

$$\bar{U}_r = \bar{r}/r$$

$$\alpha_{1vgj} = \tan^{-1}\left[\frac{\bar{V}_g \cdot \bar{P}_{I_0}}{-\bar{V}_g \cdot Y_{A_0}}\right] \qquad K\alpha L_k < \alpha_{1vgj} \leq K\alpha u_k$$

$$\alpha_{2vgj} = 0$$

$$\bar{Y}'_A = \cos(\alpha_{1vgj})\,\bar{Y}_{A_0} - \sin(\alpha_{1vgj})\,\bar{P}_{I_0}$$

$$\alpha_{3vgj} = \tan^{-1}\left[\frac{-\bar{V}_g \cdot \bar{Y}'_A}{\bar{V}_g \cdot \bar{R}_{0_0}}\right] \qquad K\gamma L_k < \alpha_{3vgj} \leq K\gamma u_k$$

$$\bar{U}_{vg} = \bar{V}_g/V_g$$

$$V_g = |\bar{V}_g|$$

$$\bar{\theta}_k = [A_{\theta j} + B_{\theta j}(V_g) + C_{\theta j}\,(V_g)^2]\,\bar{U}_r \times \bar{U}_{vg}$$

$$\alpha_{1c} = \alpha_{1vgj} + (\bar{\theta}_k \cdot \bar{R}_{0_0})$$

$$\alpha_{2c} = 0$$

$$\alpha_{3c} = \alpha_{3vgj} + (\bar{\theta}_k \cdot \bar{P}'_I)$$

where $\bar{P}'_I = (\cos\alpha_{1c})\,\bar{P}_{I_0} + (\sin\alpha_{1c})\,\bar{Y}_{A_0}$

IV.2.3 HS3 - Constant Attitude Steering Commands

Input: $\alpha_{1c}(t = t_j)$, $\alpha_{2c}(t = t_j)$, $\alpha_{3c}(t = t_j)$

Output: α_{1c}, α_{2c}, α_{3c}

$$\alpha_{1c} = \alpha_1 (t = t_j)$$

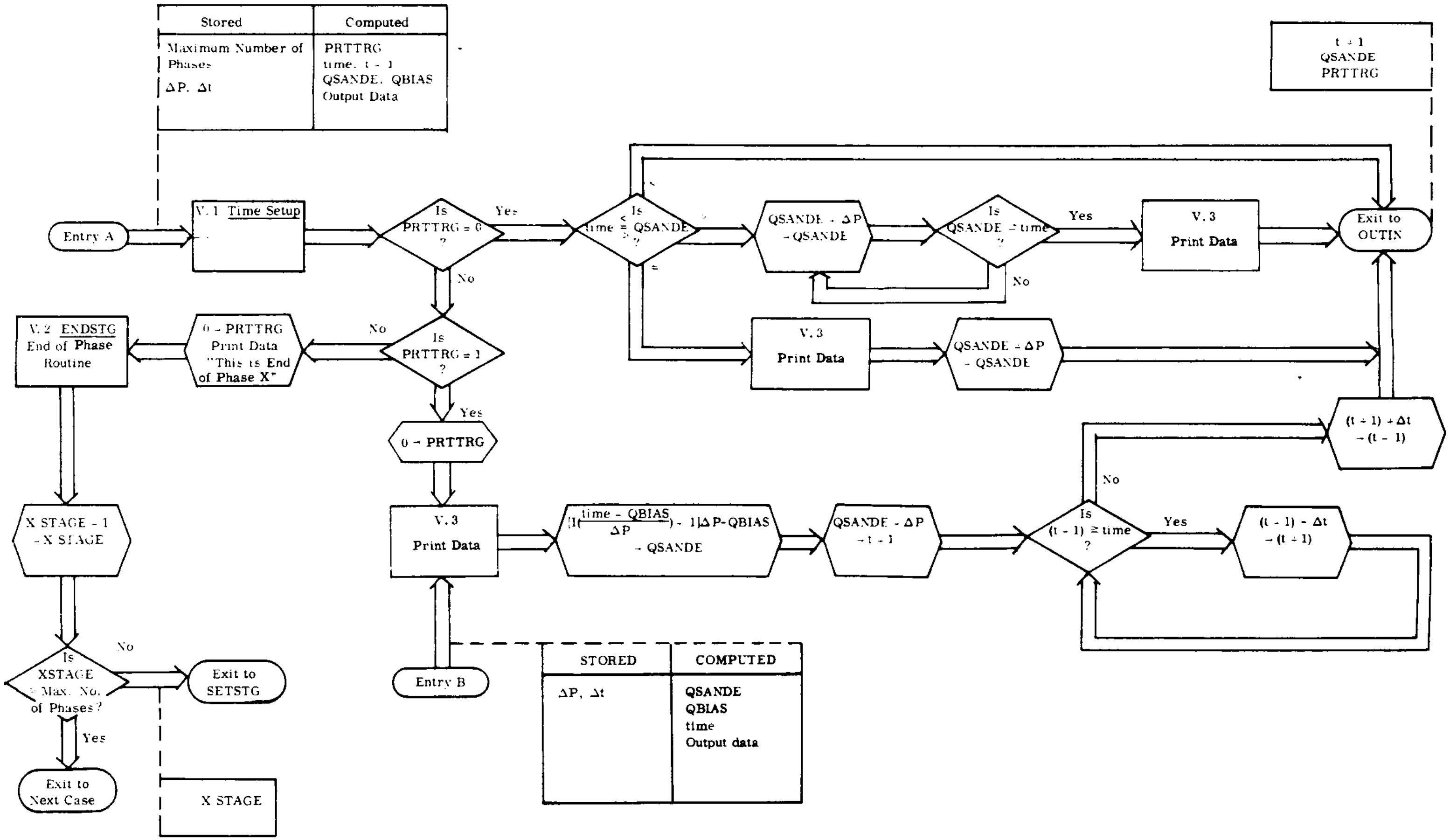
$$\alpha_{2c} = \alpha_2 (t = t_j)$$

$$\alpha_{3c} = \alpha_2 (t = t_j)$$

$$t_j = T_{ES5} \text{ and } T_{ES7}$$

4.2.5 CONTROL AND PRINTOUT - BLOCK V

4.2.5.1 LEVEL II FLOW CHART



BLOCK V. PRTOUT (FRINTOUT AND PROGRAM CONTROL)

4.2.5.1.1 V.1 TIME SETUP

OUTPUT: Time, QBIAS, QSANDE, (t +1)

INPUT: Time, QBIAS, QSANDE, (t +1)

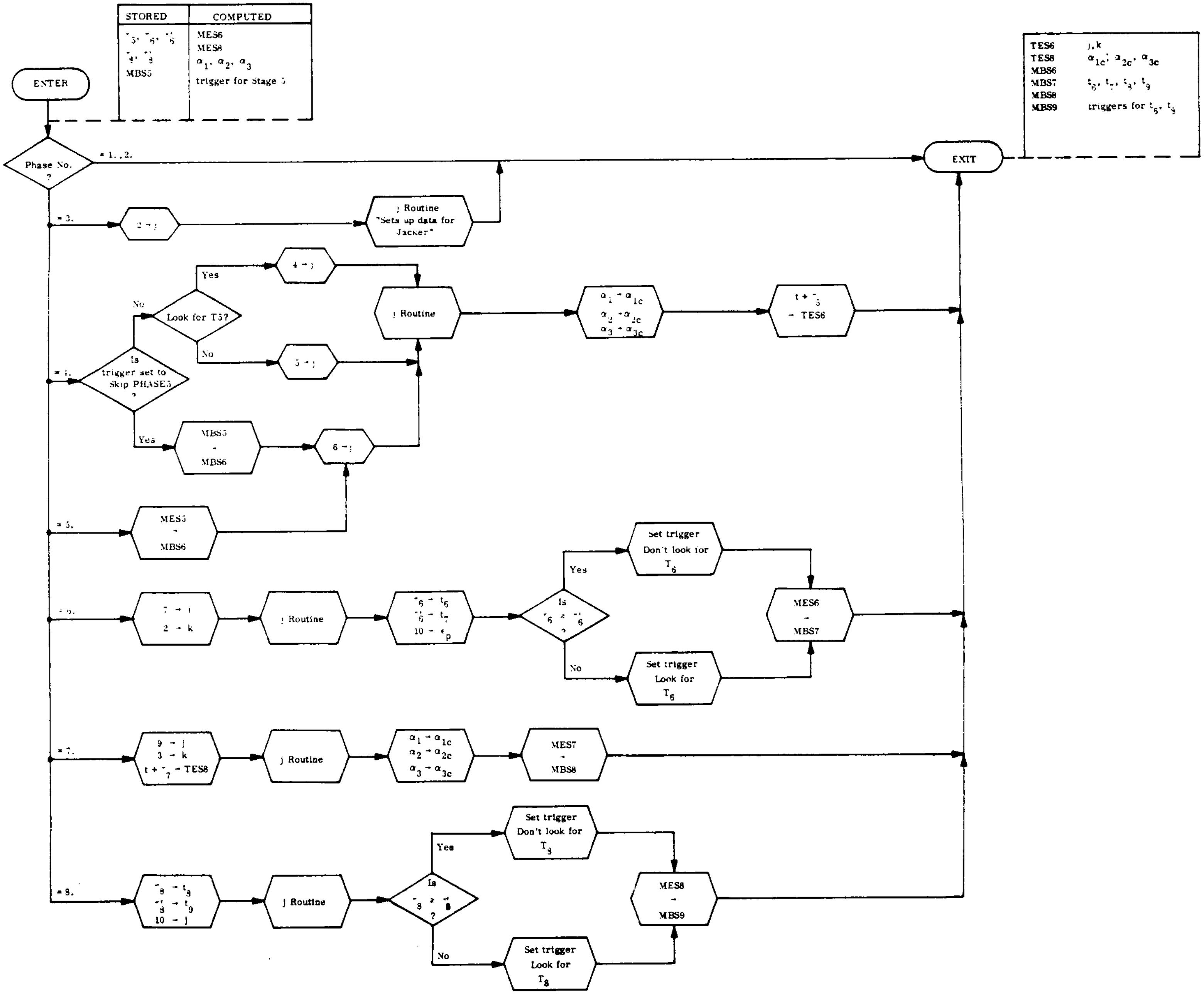
1) I (time) → Common

2) QBIAS + Common → QBIAS

3) QSANDE - Common → QSANDE

4) (t +1) - Common → (t +1)

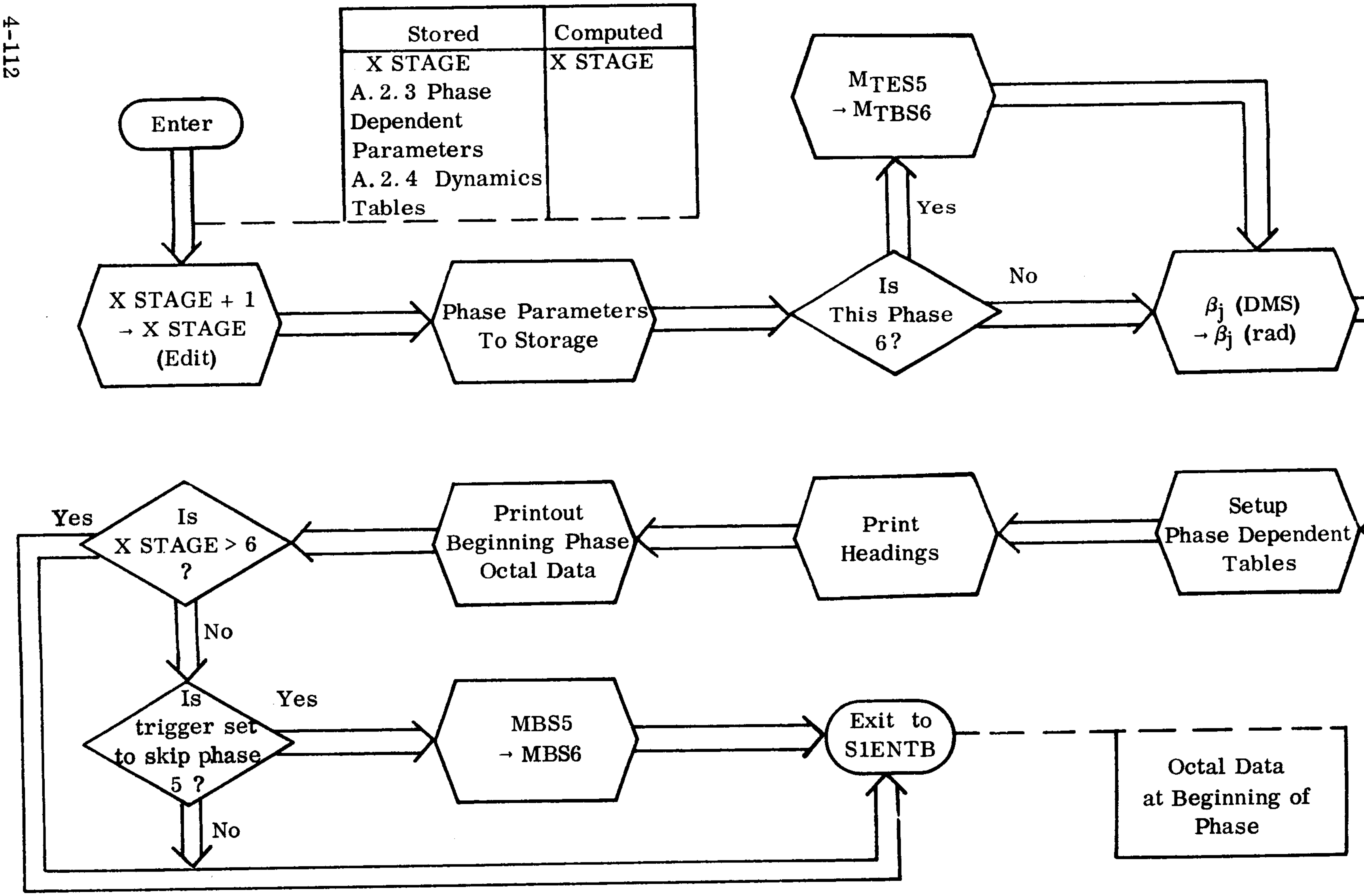
5) Time - Common → Time


STORED
τ_5, τ_6, τ'_6
τ_4, τ'_9
MBS5
COMPUTED
MES6
MES8
α_1, α_2, α_3
trigger for Stage 5
ENTER
Phase No. ?
= 1., 2.
= 3.
= 4.
= 5.
= 6.
= 7.
= 8.
2 → j
j Routine "Sets up data for Jacker"
Is trigger set to Skip PHASE5 ?
No
Yes
Look for T5?
Yes
No
4 → j
5 → j
MBS5 → MBS6
6 → j
MES5 → MBS6
j Routine
$\alpha_1 \to \alpha_{1c}$ $\alpha_2 \to \alpha_{2c}$ $\alpha_3 \to \alpha_{3c}$
$t + \tau_5 \to$ TES6
7 → j
2 → k
j Routine
$\tau_6 \to t_6$ $\tau'_6 \to t_7$ $10 \to \epsilon_p$
Is $\tau_6 \geq \tau'_6$?
Yes
No
Set trigger Don't look for T_6
Set trigger Look for T_6
MES6 → MBS7
9 → j
3 → k
$t + \tau_7 \to$ TES8
j Routine
$\alpha_1 \to \alpha_{1c}$ $\alpha_2 \to \alpha_{2c}$ $\alpha_3 \to \alpha_{3c}$
MES7 → MBS8
$\tau_8 \to t_8$ $\tau'_8 \to t_9$ $10 \to j$
j Routine
Is $\tau_8 \geq \tau'_8$?
Yes
No
Set trigger Don't look for T_8
Set trigger Look for T_8
MES8 → MBS9
EXIT
TES6
TES8
MBS6
MBS7
MBS8
MBS9
j, k
α_{1c}, α_{2c}, α_{3c}
t_6, t_7, t_8, t_9
triggers for t_6, t_8

4.2.5.1.2 V.2 ENDSTG (END OF PHASE ROUTINE)

4.2.6 PHASE SETUP - BLOCK VI

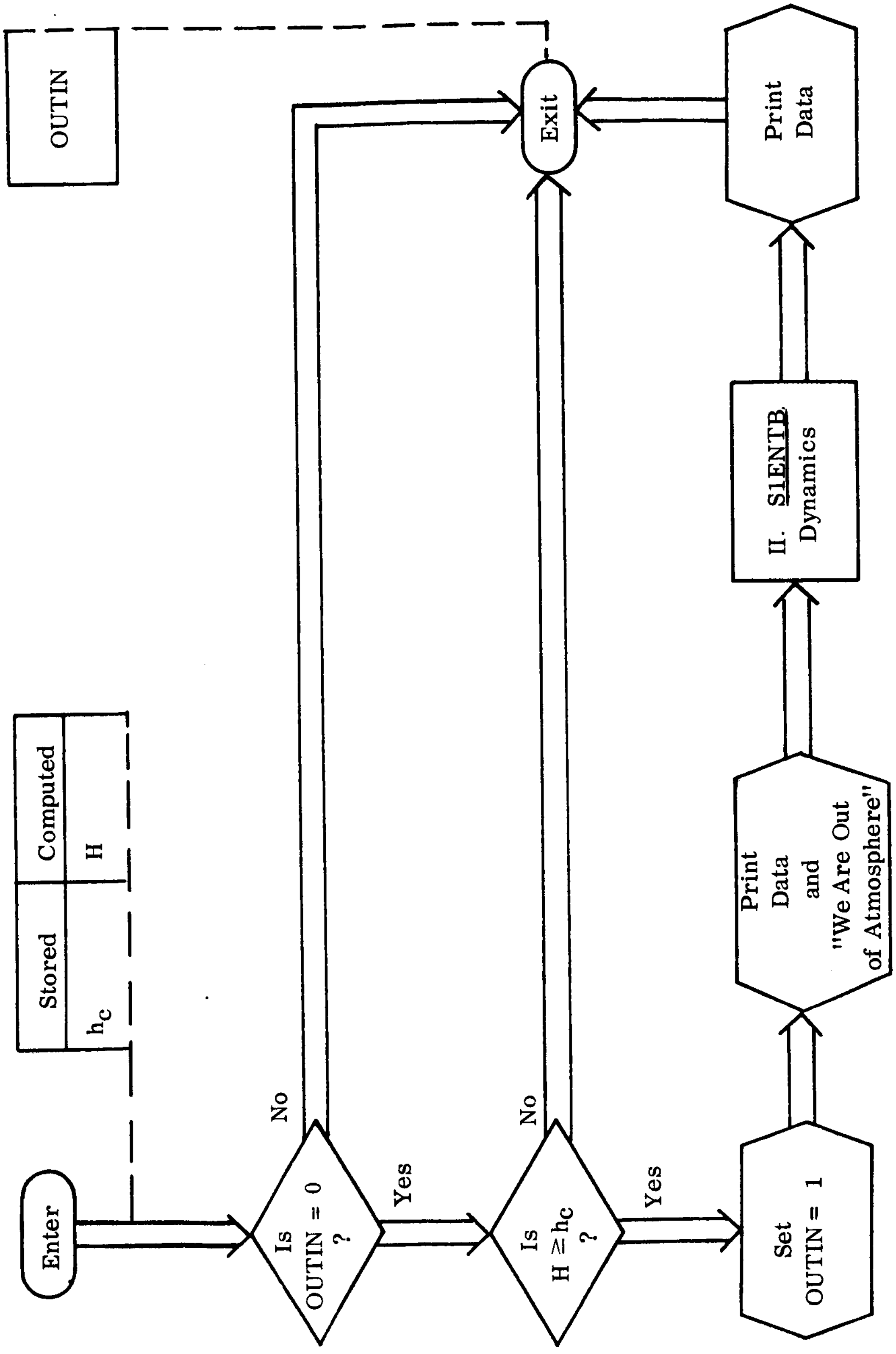
4.2.6.1 LEVEL II FLOW CHART


Enter
Stored
X STAGE
A. 2. 3 Phase Dependent Parameters
A. 2. 4 Dynamics Tables
Computed
X STAGE
X STAGE + 1 → X STAGE (Edit)
Phase Parameters To Storage
Is This Phase 6?
Yes
No
MTES5 → MTBS6
β_j (DMS) → β_j (rad)
Setup Phase Dependent Tables
Print Headings
Printout Beginning Phase Octal Data
Is X STAGE > 6 ?
Yes
No
Is trigger set to skip phase 5 ?
Yes
No
MBS5 → MBS6
Exit to S1ENTB
Octal Data at Beginning of Phase

BLOCK VI. SETSTG (SET PHASE)

4.2.7 ATMOSPHERIC TRIGGER - BLOCK VII

4.2.7.1 LEVEL II FLOW CHART



BLOCK VII. OUTIN (SETUP TRIGGER FOR ATMOSPHERE)

5.0 USER'S GUIDE

5.1 General Capabilities of Program

In order to be able to use program 118.0 it is first necessary to know what is available and in what order the options may be used.

The program can be divided into two main categories, dynamics computations and guidance computations. In the dynamics category, the physical conditions of the vehicle and equations of motion are computed and integrated. In the guidance category, gimbal angle commands are generated based on position and velocity of the vehicle and the desired end conditions.

There are nine phases in this program, each of which can simulate a separate vehicle stage. Phases 1 through 3 are set up to simulate the vehicle from launch at a planet's surface up through the gravity turn. Phases 4 and 5 are used for boosting up to orbital altitude and into circular orbit. Phases 6 and 8 are coast phases and phases 7 and 9 are used for orbit transfer. All phases do not have to be used, however, phases must be used in sequence.

Within the nine phases are four guidance schemes with at least two difference schemes available in every phase except 6 and 8. There is available zero angle of attack steering (β^*), $\dot{P}$, $\dot{Q}$, $\dot{R}$ velocity polynomial steering, explicit guidance and trim curve fit steering. The β^* steering and $\dot{P}$, $\dot{Q}$, $\dot{R}$ steeering is controlled in the "ℓ" logic guidance section and the others are controlled in the "j" logic guidance section. The relationship between the guidance logic and the nine phases can be seen on the chart in Figure 5.1.1. The guidance logic changes at the critical time points T_1 through T_9 or on the end of a phase. Different paths may be selected depending upon the vehicle simulated and the end result desired.

The table in Figure 5.1.2 can be used to relate the intervals between critical time points with the associated "ℓ" or "j" number and also the particular guidance blocks that are used during the interval.

The table in Figure 5.1.3 presents the critical time points and indicates the criteria used in their determination.

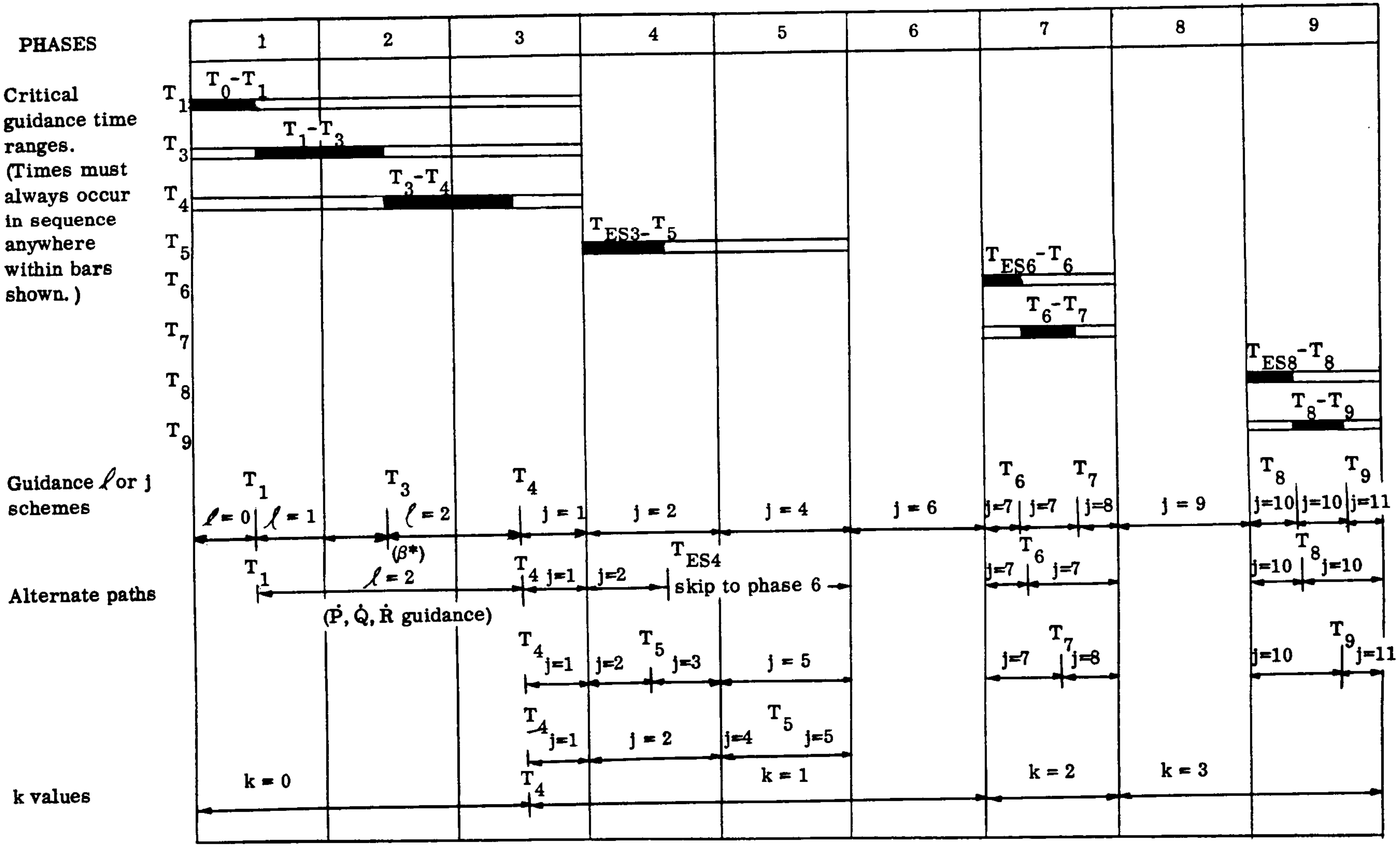


Figure 5.1.1. Relationships Between Guidance Logic and the Nine Phases

Interval	Phase	Phase No.	l	j	k	Blocks Used
T_0-T_1	Vertical rise	1	0	0	↓	See BETLOG
T_1-T_3	Pitchover	1-3	1	0	↓	See BETLOG
T_1-T_4	Curve fit guidance	1-3	2	0	↓	See BETLOG
T_3-T_4	Zero angle of attack	1-3	2	0	↓	See BETLOG
T_4-T_{ES3}	Explicit guidance	1-3	↓	1	1	HA4, HA5, HA6, HS1
$T_{ES3}-T_{ES4}$	Explicit guidance	4	↓	2	1	HA4, HA5, HA6, HS1
$T^*_{ES3}-T^*_5$	Explicit guidance	4	↓	2	1	HA4, HA5, HA6, HS1
$T_{ES4}-T_5$	Explicit guidance	5	↓	4	1	HA4, HA5, HA6, HS1
$T^*_{ES4}-T^*_{ES5}$	Explicit guidance	5	↓	4	1	HA4, HA5, HA6, HS1
$T^*_5-T^*_{ES4}$	Curve fit guidance	4	↓	3	1	HA7, HS1
$T^*_{ES4}-T^*_{ES5}$	Curve fit guidance	5	↓	5	1	HA7, HS1
T_5-T_{ES5}	Curve fit guidance	5	↓	5	1	HA7, HS1
$T_{ES5}-T_{ES6}$	Coast	6	↓	6	1	HS3
$T_{ES6}-T_6$ or T_7	Orientation	7	↓	7	2	HA4, HA5, HA6, HS2 or HA7, HS2
T_6-T_7 or T_{ES7}	Explicit guidance	7	↓	7	2	HA4, HA5, HA6, HS1
T_7-T_{ES7}	Curve fit guidance	7	↓	8	2	HA7, HS1
$T_{ES7}-T_{ES8}$	Coast	8	↓	9	3	HS3
$T_{ES8}-T_8$ or T_9	Orientation	9	↓	10	3	HA4, HA5, HA6, HS2 or HA7, HS2
T_8-T_9 or T_{ES9}	Explicit guidance	9	↓	10	3	HA4, HA5, HA6, HS1
T_9-T_{ES9}	Curve fit guidance	9	↓	11	3	HA7, HS1

*Indicates alternate paths.

Figure 5. 1. 2

CRITICAL TIME PT.	DESCRIPTION	HOW DETERMINED
T_0	Begin vertical liftoff	$T_0 = 0$
T_1	Begin pitchover	Trajectory shaping input
T_{ES1}	End of phase 1	$M = M_{B01}$
T_{ES2}	End of phase 2	$M = M_{B02}$
T_3	Begin zero-angle-of-attack (does not occur when using $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering)	Trajectory shaping input
T_4	Begin explicit guidance (may occur in phase 1, 2, or 3)	Trajectory shaping input
T_{ES3}	End of phase 3	$M = M_{B03}$
T_{ES4}	End of phase 4	$M = M_{B04}$ or $\lvert\vec{V}_g . \vec{U}_{vg}\rvert < \epsilon_{vg}$ whichever comes first
T_5	Begin curve fit guidance (may occur in phase 4 or 5 or not at all, and stage ends with explicit guidance)	$\lvert r - r_{TR}\rvert < \epsilon_{TR}$ or $\lvert\vec{U}_r . \vec{U}_{TR}\rvert < \epsilon_{TRA}$, whichever comes first.
T_{ES5}	End of phase 5	$\lvert\vec{V}_g . \vec{U}_{vg}\rvert < \epsilon_{vg}$
T_{ES6}	End of phase 6 (Coast phase)	$T_{ES6} = T_{ES5} + \tau_5$
T_6	Begin thrust in phase 7 with explicit guidance	$\lvert r - r_{EX}\rvert < \epsilon_{TR}$, $\lvert\vec{U}_r . \vec{U}_{EX}\rvert < \epsilon_{TRA}$, or τ_6 whichever comes first. (If $\tau_6 > \tau'_6$, T_6 does not occur.)
T_7	Begin curve fit guidance (may not occur and stage ends with explicit guidance)	$\lvert r - r_{TR}\rvert < \epsilon_{TR}$, $\lvert\vec{U}_r . \vec{U}_{TR}\rvert < \epsilon_{TRA}$, or τ'_6 whichever comes first.
T_{ES7}	End of phase 7	$\lvert\vec{V}_g . \vec{U}_{vg}\rvert < \epsilon_{vg}$

Figure 5. 1. 3 (page 1 of 2)

CRITICAL TIME PT.	DESCRIPTION	HOW DETERMINED
T_{ES8}	End of phase 8 (coast phase)	$T_{ES8} = T_{ES7} + \tau_7$
T_8	Begin thrust in phase 9 with explicit guidance	$\lvert r - r_{EX} \rvert < \epsilon_{TR}$, $\lvert \vec{U}_r \cdot \vec{U}_{EX} \rvert < \epsilon_{TRA}$, or τ_8 whichever comes first. (If $\tau_8 > \tau'_8$, T_8 does not occur.)
T_9	Begin curve fit guidance (may not occur and stage ends with explicit guidance)	$\lvert r - r_{TR} \rvert < \epsilon_{TR}$, $\lvert \vec{U}_r \cdot \vec{U}_{TR} \rvert < \epsilon_{TRA}$, or τ'_8 whichever comes first.
T_{ES9}	End of phase 9 (end of run)	$\lvert \vec{V}_g \cdot \vec{U}_{vg} \rvert < \epsilon_{vg}$

Figure 5. 1. 3 (page 2 of 2)

5.2 DISCUSSION OF INPUT QUANTITIES

The input to Program 118.0 is very large and the following is an attempt to aid the user in making up a run.

The input sheets supplied with Program 118.0 are set up in a straightforward manner, however, each run does not require completion of the entire set. It is only necessary to fill out the applicable portions. In general, when investigating a particular vehicle, it is useful to have a full input set for all of the dynamics blocks in all phases, since only guidance parameters change. When making a series of runs together, it is sufficient to input all of the data for the first of the stacked runs and then inputting only the data that is to be changed for the succeeding runs. (This applies to Program 118.0 only and does not apply to Program 117.0 where complete data must be inputted for each run.)

The data must be left justified on the input cards with no blank spaces in between. In cases of duplicate inputs, the latest datum is used. The data may be entered either in decimal or "E" format for floating point inputs and in octal format also if the operation "DEC" is changed to "OCT".

The table inputs may be simplified from the input sheet format. For single variable tables, the number of input data points can be reduced to three for constant or straight line functions with the first two being the initial datum point inputted twice, and the third being the end point placed in consecutive computer locations. It is important that the minimum and maximum independent table values include all ranges of the vehicle and that the first and last data points correspond to the minimum and maximum values.

For bivariate tables, see Section 5.2.1.16.

A discussion of all of the inputs is broken up into three sections, 5.2.1 - Dynamics, 5.2.2 - Guidance, and 5.2.3 - Operational. These quantities are discussed by block.

5.2.1 Dynamics

The inputs to the dynamics blocks include all phase-dependent vehicle data and all the physical environment data.

5.2.1.1 The integration step size for dynamics (δt) is input small enough so that the accuracy of the simulation is not impaired. If it is made too small the simulation may deteriorate in accuracy due to roundoff errors as well as consume excess computer time. It is desirable to have the δt one-fourth or less of the guidance step size (Δt) for more realistic simulation. A δt equal to about one-tenth of the period of the system has been found satisfactory.

5.2.1.2 The pitch position gain (K_θ) and the pitch rate gain ($K_{\dot\theta}$) determine the rate for nulling out the pitch error signal. The ratio $K_\theta/K_{\dot\theta}$ should always be equal to $1/\delta t$ or less.

5.2.1.3 The yaw position gain (K_ψ) and the yaw rate gain ($K_{\dot\psi}$) determine the rate for nulling out the yaw error signal. The ratio $K_\psi/K_{\dot\psi}$ should always be equal to $1/\delta t$ or less.

5.2.1.4 The roll position gain (K_ϕ) and the roll rate gain ($K_{\dot\phi}$) determine the rate for nulling out the roll error signal. The ratio $K_\phi/K_{\dot\phi}$ should always be equal to $1/\delta t$ or less.

5.2.1.5 The table for specific impulse vs atmospheric pressure (gISP vs P) is a function of the engine being simulated. The ISP values are multiplied by g before entering into this table for scaling purposes.

5.2.1.6 The table for mass flow rate vs expended mass ($\dot M$ vs $\int \dot M$) is a function of the engine being simulated.

5.2.1.7 The perturbation in specific impulse (P_{ISP}) is used to modify the table values in Section 5.2.1.5 and is normally set equal to 1.0. This input can be used to check the effects of a non-nominal engine.

5.2.1.8 The perturbation in mass flow rate (P_{FLO}) is used to modify the table values in Section 5.2.1.6 and is normally set equal to 1.0. This input can be used to check the effects of a non-nominal engine.

5.2.1.9 The mass at the end of a phase (MES) is usually set equal to the dry weight of the vehicle. For phase 1 through 4 the MES is used to end the phase when the vehicle mass (M_0 - $\int \dot M$) becomes equal to MES. On phases other than 6 and 8, cutoff occurs when the velocity criteria is achieved and care must be taken to avoid using the stage past its expendable mass. Phases 6 and 8 are ended on a time criteria.

5.2.1.10 The initial mass of the vehicle (M) includes the dry mass and the expendable mass and the initial mass (except for cases of special start input) is only used in phases 1 through 5. All other phases assume a restartable engine and the end mass of one phase becomes the initial mass for the next phase.

5.2.1.12 The maximum angular rate limit (β_j) is used to simulate the limits on the engine gimbals by limiting the error angle commands (θ'_{EC}, ψ'_{EC}, ϕ'_{EC}). The rate limit is therefore the product of β_j and the ratios of $K_\theta/K_{\dot\theta}$, $K_\psi/K_{\dot\psi}$, and $K_\phi/K_{\dot\phi}$, respectively.

5.2.1.13 The table of center of gravity vs mass (C_g vs M) is a function of the vehicle simulated. The C_g values are the distances from the reference vehicle station up to the center of gravity.

5.2.1.14 The effective drag area (S) is a function of the vehicle simulated.

5.2.1.15 The table of axial drag vs Mach number (C_A vs R_V) is a function of the vehicle simulated.

5.2.1.16 The table of normal drag vs Mach number and angle of attack (C_N vs R_V, α) is a function of the vehicle simulated. These bivariate tables can be input in a simplified form from the generalized input sheet format. For a constant C_N, one can use the first computer address for the C_N table to input a "zero," "comma,", "constant C_N" as follows:

LOCATION 1	7	OP 8	11	12
R _____ (Initial table location)		DEC		0, C_N (constant)

A regular table of C_N values can take the following form where only the initial table location is used and continuing data put on cards with "DEC" in the Operation columns (OP).

LOCATION 1	7	OP 8	11	DATA 12
R - - - - -		DEC		1, α_1, α_2, ..., α_N, R
		DEC		R_{V1}, $C_N(\alpha_1, R_{V1})$, $C_N(\alpha_2, R_{V1})$, ..., $C_N(\alpha_N, R_{V1})$
		DEC		R_{V2}, $C_N(\alpha_1, R_{V2})$, $C_N(\alpha_2, R_{V2})$, ..., $C_N(\alpha_N, R_{V2})$
		DEC		$\vdots$
		DEC		R_{VM}, $C_N(\alpha_1, R_{VM})$, $C_N(\alpha_2, R_{VM})$, ..., $C_N(\alpha_N, R_{VM})$

There are a maximum of 80 table entries for phases 1 through 5 and a maximum of 60 table entries for phases 6 through 9. For each R_V, there should be as many C_N entries as there are α_N entries. This allows some flexibility not provided for on the input forms provided with the program.

The table of center of pressure vs Mach number and angle of attack (C_p vs R_V, α) is a function of the vehicle simulated. The C_p values are measured from the reference station for the vehicle up to the center of the pressure point. The (C_P vs R_V, α) table can be input in a similar manner to the (C_N vs R_V, α) table above.

5.2.1.17 The tables for North Wind vs Altitude (H) and East Wind vs Altitude (H) are set up on the basis of existing knowledge of the planetary atmosphere. Combinations of North Wind and East Wind through the perturbation constants, (NFLAG) and (EFLAG) can yield any winds desired.

The Wind Flag (WFLAG) is used to select the winds desired and the constants for perturbing North wind (NFLAG) and East wind (EFLAG) are used to multiply table values.

WFLAG = 0	means no wind effects
WFLAG = 1	means East wind only
WFLAG = 2	means North wind only
WFLAG = 3	means both winds together

5.2.1.18 The coefficients for the 2nd, 3rd, and 4th harmonics of a planet's gravitational potential (J_g, H_g, D_g) are based on the knowledge of the planet's gravitational field. The present values for Earth are $J_g = .162345 \times 10^{-2}$, $H_g = .575 \times 10^{-5}$, $D_g = .7875 \times 10^{-5}$. For $J_g = H_g = D_g = 0$, a uniform spherical gravitation model is obtained.

The parameters of equatorial radius (ae), the ellipticity (e), the gravitational constant (GM), the planet's rotational rate (Ω), the altitude at the end of the sensible atmosphere (H_c), gravity (g), and the range coefficient (R_{C1}) are all dependent on the knowledge of the planet's physical characteristics. The R_{C1} is usually input as the radius of the reference sphere in units of nautical miles or kilometers since it is used for edit purposes only in the computation of range from the launch site to the vehicle along the reference sphere.

5.2.1.19 The atmospheric flag (ATMOPT) is used to determine the atmospheric routine to be used in the simulation. ATMOPT = 0 selects a detailed atmospheric model for the Earth with the output in English units only. ATMOPT = 1 selects an exponential atmospheric model requiring the input of the sea level density (ρ_o), the sea level pressure (P_o), the speed of sound (C_I), and the atmospheric constant (β_I).

5.2.1.20 The initial conditions of the vehicle for launch from a planet are determined by the geodetic launch latitude, λ_L, the geodetic launch longitude, μ_L, the astronomical launch latitude, λ_{LA}, the astronomical launch longitude, μ_{LA}, the trajectory azimuth, ψ_p, the vehicle height above the reference sphere, H_L, and the geodal separation, N_L. The above quantities place the roll axis of the vehicle along the local vertical, the yaw axis normal to the roll axis and in the trajectory plane and the pitch axis perpendicular to the trajectory plane as shown in Figure 3.1.3.

5.2.1.21 Coast Periods

The coast periods (Phase 6 and Phase 8) are defined by the τ_5 and τ_7 inputs, respectively. These inputs specify the duration of the phase.

5.2.2 Guidance

The inputs to guidance blocks can be subdivided into four main types of guidance schemes. There is the zero angle of attack steering (β*), the $\dot{P}$, $\dot{Q}$, $\dot{R}$ velocity polynomial steering, explicit guidance, and trim curve fit steering. The explicit guidance inputs and the trim curve fit steering inputs are "j" dependent in which different inputs can be made for each "j". There are many "j" dependent inputs on the input forms which will never get used but were left in for consistency.

5.2.2.1 β* Inputs

The inputs of T_1, T_3, T_4, and K_α are used to shape the trajectory of the vehicle in phases 1 through 3 when option OP0123 is set equal to 0. T_1 controls the time to begin pitchover. K_α controls the pitchover rate by commanding a constant qα. The selection of K_α is influenced by the structural limits of the vehicle where qα is a measure of the stress placed on the vehicle. T_3 is used to end pitchover and begin zero angle of attack. T_4 signifies the end of zero angle of attack steering and beginning explicit guidance.

5.2.2.2 $\dot{P}$, $\dot{Q}$, $\dot{R}$ velocity polynomial steering (Block PSITH)

The $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering is used in phases 1 through 3 where a trajectory is already shaped and the velocity profile is available. To go from β* to an equivalent $\dot{P}$, $\dot{Q}$, $\dot{R}$ trajectory is made fairly easy with this program. While running a β* shaping run with PROPT = 1 for full 13-line print the parameters for curve fitting are all supplied for the input θ_{cc} value. A 3rd order curve fit of $\dot{R}_a$ (independent variable) vs $\dot{Q}_a$ (dependent variable) for obtaining the K_{53} to K_{56} values to the following equation:

$$\dot{Q}_a = K_{53} + K_{54}(\dot{R}_a) + K_{55}(\dot{R}_a)^2 + K_{56}(\dot{R}_a)^3$$

A 3rd order curve fit is also made of $\dot{R}_a$ (independent variable) vs α_3 (dependent variable) for obtaining the K_{58} - K_{61} values to the following equation:

$$\alpha_3 = K_{58} + K_{59}(\dot{R}_a) + K_{60}(\dot{R}_a)^2 + K_{61}(\dot{R}_a)^3$$

The values of R_{STOP1} and R_{STOP4} are set equal to the values of $\dot{R}_a$ at T_1 and T_4 respectively. The values of K_{50}, K_{51}, and K_{52} are guidance gains and are selected to provide a smooth response. Typical values for these gains are $K_{50} = -0.002$, $K_{51} = 0$, $K_{52} = 0.002$. Note that K_{50} has a negative input value.

5.2.2.3 Explicit Guidance

The input for explicit guidance is the most difficult of all the guidance schemes due to the various options. Explicit is available from T_4 to T_5 or j = 1, 2 or 4, from T_6 to T_7

or j = 7, and from T_8 to T_9 or j = 10. T_4 is determined by input with β^* steering or by R_a reaching the R_{STOP4} input value with $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering. T_6 and T_8 are determined by one of three criteria whichever occurs first.

The first of the T_6 and T_8 criteria is based on time such that when integrated time (t) reaches the input values of τ_6 and τ_8 the criteria for T_6 and T_8 are satisfied. It should be noted that τ'_6 and τ'_8 must be made greater than τ_6 and τ_8, respectively or else T_6 and T_8 are not looked for by any of the three criteria.

The second criteria for T_6 and T_8 is that of the vehicle reaching an inputted radial distance (r_{EX}) within an inputted ϵ_{TR} value in time

$$\left(\frac{r_{EX} - r}{\dot{r}} < \epsilon_{TR}\right)$$

The value of r_{EX} for T_6 is inputted in j = 7 and for T_8 is inputted in j = 10.

The last criteria for T_6 and T_8 is that of the vehicle's unit position vector ($\vec{U}_r$) becoming perpendicular to an inputted unit vector ($\vec{U}_{EX}$) within an ϵ_{TRA} value in time

$$\left(\frac{(\vec{U}_r \cdot \vec{U}_{EX})_i\ (\Delta t)}{(\vec{U}_r \cdot \vec{U}_{EX})_{i-1} - (\vec{U}_r \cdot \vec{U}_{EX})_i} < \epsilon_{TRA}\right)$$

The unit vector $\vec{U}_{EX}$ for T_6 is inputted in j = 7 and for T_8 in j = 10.

Typical values of ϵ_{TR} and ϵ_{TRA} have been 10^{-4} seconds for programs that have been run in the past.

Thrust is begun after reaching T_6 or T_7 and the $\dot{M}_{T6}$ input is used to perturb the thrust at this time. Similarly, after reaching T_8 or T_9 a $\dot{M}_{T8}$ input is used.

5.2.2.3.1 Aim Point Conditions (Blocks HA4A and HA4B)

There are two types of aim points available in explicit guidance. There is the inertial target point which remains fixed in inertial space (Block HA4A), and there is the planet fixed target point which revolves with the planet in inertial space (Block HA4B). The inertial target point is selected by making $F_{AIM_j} = 0$ and the planet fixed target point is selected by making $F_{AIM_j} = 1$. The inertial target point is entered as the ($\bar{i}$, $\bar{j}$, $\bar{k}$) components of the aim point vector ($\bar{R}_{A_j}$). The planet fixed target point is entered in cylindrical coordinates (C_1, C_2, C_3) where C_1 is the projection of the aim point on the X-Y plane, C_2 is the angle in the X-Y plane from the X-axis to the projection of the aim point on the X-Y plane measured in the direction of Ω at t = 0, and C_3 is the projection of the aim point on the Z-axis. An estimate in the time of flight (T_{o_j}) should be put in. This input is not critical since it will iterate to the correct value, and the value of T_{0_j} can be updated on successive runs from the iterated values.

5.2.2.3.2 $\vec{V}_g$ Computation (Block HA5)

The form of explicit guidance programmed requires an initial estimate of,P_j, the semilatus rectum be put in. For constant flight path angle restraint runs the value of p_j can be found from

$$p_j = \frac{r\, r_A\ (c - 1)}{r(c + s \tan K_{\gamma_j}) - r_A}$$

For constant time of arrival constraint, it is recommended that runs be made first with the flight path angle restraint until one approaching the desired time of arrival constraint is obtained. The p_j from the constant flight path angle restraint run can be used for the constant time of arrival constraint run.

A unit vector, $\vec{U}_{W_j}$, perpendicular to the desired trajectory plane and positive in the sense of $\vec{r} \times \vec{v}$ is required in this block.

When the range angle from the vehicle to the aim point gets too small ($s < \epsilon_S$) then several of the computations have to be avoided to prevent indeterminate solutions. A typical value of ϵ_S has been 10^{-3}. It should be noted that when the range angle approaches π radians that the value of s becomes small and that, if possible, it is desirable to select aim points that are not near zero or π radians away from the boost vehicle.

The gravitational constant ($K_j = \sqrt{GM}$) and its reciprocal ($K_{r_j} = 1/K_j$) must be inputted for the central body involved.

If the trajectory plane is one of the basic criterion of the mission, then ϵ_u should be made equal to 1. However, if hitting the aim point is one of the basic criterion then ϵ_u should be made equal to some small value like 10^{-3}.

A constant K_{VG} is an input that allows one to modify the $\vec{V}_g$ vector. The value of K_{VG} is nominally equal to 1. However, for deceleration phases it may be set equal to -1 to maintain proper steering commands.

5.2.2.3.3 End Point Restraints (Blocks HA6A and HA6B)

There are two types of restraints at the aim points available in the explicit guidance programmed. There is the restraint on the time of arrival ($F_{CON_j} = 0$; Block HA6A), and restraint on the flight path angle ($F_{CON_j} = 1$; Block HA6B). The flight path angle restraint is the easier to use and is recommended over the constant time of arrival constraint. Even when the time of flight is to be specified it is desirable to use the flight path angle restraint to obtain the approximate constants necessary.

For Block HA6B the desired flight path angle is specified through the K_{γ_j} input where the angle is measured from the horizontal plane (plane perpendicular to the aim point vector), positive in the direction of the aim point vector. The values of K_{13} and K_{14} are curve fit quantities and are evaluated as follows. A trial run is made with

$$K_{13_j} = \frac{p_j^2 \, s}{r_A (1 - c) \cos^2 K_{\gamma_j}}$$

$K_{14_j} = 0$, $\epsilon_\gamma = 10^{-8}$ and $\epsilon_p = 10^{10}$. At each print point thequantities p_i, p_{i-1}, γ_{A_i}, $\gamma_{A_{i-1}}$ and s are obtained with PROPT = 1. Making a plot of

$$\frac{(p_i - p_{i-1})}{(\gamma_{A_i} - \gamma_{A_{i-1}})} \text{ vs s}$$

one can obtain an approximate straight line solution where K_{13_j} is now made equal to the intercept of this line and the s = 0 axis and K_{14_j} is made equal to the slope of this line.

For Block HA6A, the time of arrival at the target is specified by the TA_j input. From an approximated constant flight path angle restraint run an initial value of T_{O_j} can be found as well as the K_{13_j} and K_{14_j} values. A plot of $(p_i - p_{i-1})/(T_i - T_{i-1})$ vs p is made and an approximate straight line solution obtained. Similarly, the K_{13_j} value is made equal to the intercept with the s = 0 axis and the K_{14_j} value is made equal to the slope of this line. By setting the $\epsilon_T = 10^{-8}$ and $\epsilon_p = 10^{10}$ we obtain the smoothest trajectory.

5.2.2.3.4 Explicit Cutoff Option

During the use of explicit guidance, an input (F_{TRIM}) is provided such that it is possible to use explicit guidance past the normal velocity cutoff point. This allows the vehicle to gain additional velocity towards the aim point where the ultimate goal is to achieve circular speed near the aim point. F_{TRIM} is set to 0. for the normal velocity-to-be-gained cutoff and is set to 1. for not using the velocity cutoff.

The velocity cutoff occurs when the time to reduce $\overline{V}_g$ to zero becomes less than or equal to ϵ_{vg}.

$$\text{When } \frac{|\overline{V}_g \cdot \overline{U}_{vg}|_i \, |\Delta t|_i}{|\overline{V}_g \cdot \overline{U}_{vg}|_{i-1} - |\overline{V}_g \cdot \overline{U}_{vg}|_i} \leq \epsilon_{vg} \text{ cutoff occurs.}$$

The vector $\overline{U}_{vg}$ is a unit vector along the initial direction of $\overline{V}_g$ each time a new guidance scheme is used. A typical value for ϵ_{vg} has been 10^{-3} seconds. When a velocity cutoff occurs in Phase 4, Phase 5 is skipped and Phase 6 (the coast phase) is entered directly.

5.2.2.4 Trim Steering (Block HA7)

Trim steering is a form of curve fit guidance and is available from T_5 to TES5 or j = 3 and/or 5, from T_7 to TES7 or j = 8, and from T_9 to TES9 or j = 11.

The critical time points (T_5, T_7, T_9) are determined by the first of the following criteria reached. The first criteria is satisfied when the radial distance of the vehicle, r, is within ϵ_{TR} in time of the inputted value, r_{TR_j}. ($r_{TR_j} - r/\dot{r} \leq \epsilon_{TR}$). The second criteria is satisfied when the vehicle is within ϵ_{TRA} in time of a plane defined by the inputted vector, $\overline{U}_{TR_j}$ $[(\overline{U}_r \cdot \overline{U}_{TR})_i \Delta t_i/(\overline{U}_r \cdot \overline{U}_{TR})_{i-1} - (\overline{U}_r \cdot \overline{U}_{TR}) \leq \epsilon_{TRA}]$ T_7 and T_9 also have an additional criterion which is satisfied when the inputted values of τ_6' and τ_8' are reached respectively.

The trajectory plane is defined by the unit vector, $\overline{U}_{wj}$, which lies along the desired $\bar{r} \times \overline{V}$ vector. The desired inplane quantities of radial distance, radial rate, and tangential rate are each curve fit as a function of time as shown in Block HA7. These curve fits are thus formed as follows:

For radial distance

$$r_c(t) = [A_{RC_j} + B_{RC_j}(t) + C_{RC_j}(t)^2]$$

For radial rate

$$\dot{r}_c(t) = [A_{RD_j} + B_{RD_j}(t) + C_{RD_j}(t)^2]$$

For tangential rate

$$(r\dot{v})_c(t) = V \sin \gamma = [A_{RV_j} + B_{RV_j}(t) + C_{RV_j}(t)^2]$$

The best use for trim steering is for final guidance into a circular orbit. For the circular orbit A_{RC_j} is set equal to the desired orbital radius, A_{RV_j} is set equal to the orbital velocity and all the other quantities are set equal to zero.

The values of K_{A_i} and K_{H_j} are varied until a smooth trajectory is obtained and the desired end conditions are obtained.

5.2.2.5 Steering Block (HS1)

The steering block generates the vehicle attitude commands from the velocity-to-be-gained, V_g. In this block are inputs for the guidance loop gains with k_{2j} and k_{4j} the gains for roll and pitch, respectively and k_{3j} and k_{5j} the rate damping gains for roll and pitch, respectively. In addition to the loop gains, there are input curve fit quantities (A_θ, B_θ, C_θ) that are used to improve the steering errors. The $\overline{\theta}_k$ quantities computed from A_θ, B_θ, C_θ are used to compensate for gravity which tends to turn the vehicle. Except where extremely accurate cutoff must be simulated, these quantities (A_θ, B_θ, C_θ) may be set equal to zero. (Nominals generated for performance assessment do not require the $\overline{\theta}_k$ values.)

The $K_{\alpha Lk}$, $K_{\alpha Uk}$, $K_{\gamma Lk}$, $K_{\gamma Uk}$ values are used to specify limits on the gimbal angles and gimbal angle commands when simulating actual vehicles and IMU's. For purposes of developing nominal trajectories for performance assessment, it is sufficient to set the $K_{\alpha Lk}$ and $K_{\gamma Lk}$ values to $-\pi$ and the $K_{\alpha Uk}$ and $K_{\gamma Uk}$ values to $+\pi$ for $k = 1, 2$, and 3.

For orientation steering (Block HS2), the A_θ, B_θ, C_θ values can always be set equal to zero.

5.2.2.6 Special Start Input (Block H-Start)

There are special start inputs required when beginning a run in a phase other than Phase 1. Some of these (R10016 to R10040) are common to all of the phases except Phase 1 and others (R10041 to R10062) are necessary depending upon the particular phase. The inputs are used to initialize the run at the beginning of the desired phase for rerunning a portion of a previous run or to generate nominals with guidance schemes only available in certain phases. For rerunning a portion of a previous run, part of the data required is printed out in octal form at the beginning of each phase. (See print key in Section 4.1.4.) When inputting octal data, it is necessary to replace the operation "DEC" in columns 8 to 11 by the operation "OCT".

The input quantities that are required for beginning in any phase except Phase 1 are starting fractional part of time (t), starting mass (M), the starting position (X, Y, Z) in $\overline{i}$, $\overline{j}$, $\overline{k}$ coordinates, the starting velocity ($\dot{X}$, $\dot{Y}$, $\dot{Z}$) in $\overline{i}$, $\overline{j}$, $\overline{k}$ coordinates, the starting gimbal angles (α_{PI}, α_{YA}, α_{RO}), the starting atmospheric trigger (0. = in, 1. = out), the starting integrated thrust ($\int a_X$, $\int a_Y$, $\int a_Z$), the starting "j" number, the starting "k" number, the starting phase number and the starting whole part of time (QBIAS). The QBIAS input is normally the whole number of seconds in the total time and the "t" input is the fractional part but the total time value may be input as "t" with QBIAS equal to zero. The input of "j" and "k" are determined by the phase and guidance scheme being used as discussed in Section 5.1.1.

For beginning in Phases 2 or 3, it is also necessary to input the T_3 flag (0 = after T_3, 1. = before T_3), starting "ℓ" number, and T_4 trigger (0 = before T_1, 2. = after T_1), and line 13 printout trigger (0 = no line 13, 1. = line 13). The "ℓ" number input depends upon the guidance at the beginning of the phase and is 0. before T_1, is 1. from T_1 to T_3, and is 2. from T_3 to T_4. For the above "ℓ" numbers, "j" must be set equal to 0. For "ℓ" equal to 0. or 1., the T_3 flag should be equal to 1. and for "ℓ" equal to 2., the T_3 flag should be equal to 0. The T_4 trigger should be set equal to 0. when "ℓ" equals 0. and set equal to 2. when "ℓ" is equal to 1. or 2. It is desirable to have line 13 of the full print (PROPT = 1.) printed out during the interval before T_4 and therefore the trigger should be set equal to 1. for j = 0 cases and set equal to 0. when j = 1.

When beginning the run in Phase 5, it is necessary to set the T_5 trigger which indicates whether or not T_5 has been reached (0 = before T_5, 1. = after T_5).

When beginning in Phase 6, it is necessary to set up to end time of the coast phase by the TES6 input.

When beginning in Phase 7, it is necessary to input values for T_6, T_7, and T_6 trigger. T_6 indicates the time for beginning explicit guidance and is similar to the normal τ_6 input. T_7 indicates the time for beginning of trim curve fit guidance and is similar to the normal τ_6' input. The T_6 trigger is used to indicate whether or not explicit guidance will be used during the phase (0. = explicit guidance, 1. = no explicit guidance). A T_6 input of 1. means that T_6 is not looked for and that trim curve fit guidance is to be used at T_7.

When beginning in Phase 8, it is necessary as in Phase 6 to set up the end time of the coast phase by the TES8 input.

When beginning in Phase 9, it is necessary as in Phase 7 to set up the times for beginning of explicit guidance, T_8, beginning of trim curve fit guidance, T_9, and the trigger for determining the use of explicit guidance, T_8 trigger. (0 = explicit guidance, look for T_8, 1. = no explicit guidance, do not look for T_8.)

5.2.3 Operational

These are inputs for setting up the program in the computer. The inputs that are used to identify each run are the General heading, the heading to identify the inputted dynamics tables, and the run number. The inputs to determine the length of the run are the number of powered phases, the final phase for cutoff and the final time for cutoff. The number of powered phases causes the run to halt at the end of the phase of the inputted number. The final phase for cutoff input causes the run to halt after the initialization of the phase number inputted. This halt condition is used when it is desireable to continue the run at a later time since all of the parameters necessary for special start are made available.

The use of this halt also requires the input of data for the inputted phase number in order that the initialization can be performed. The final time for cutoff condition causes the run to halt when the integrated time reaches the inputted time and is completely independent of phases and critical time points.

There are two inputs that are used jointly (PROPT and NOR6PT), depending upon where trajectory shaping is being done or the nominal is being generated. For trajectory shaping, the full printout of the parameters except the state transition matrix is desired, thus both PROPT and NOR6PT are set equal to 1. The PROPT = 1. also prevents a tape from being written. For the nominal run, the PROPT input is set equal to 0. which provides the data necessary for performance assessment. The NOR6PT input can be set to 0. also and cause the state transition matrix to be computed and printed out. A NOR6PT input of 1. with the PROPT = 0. causes an identity form of state transition matrix to be printed out. Whenever the state transition matrix is printed out, the same data is also put onto tape unit number 2 and a blank tape should be mounted on this tape unit.

There are also inputs to improve the output format such as a conversion constant (CONVCN) which is a scalar multiplier for the velocity ($\dot{X}$, $\dot{Y}$, $\dot{Z}$) and position (X, Y, Z) terms in the printout. The number of prints per page (KNPPP) determines the time points put on a printout page. The print used with trajectory shaping (PROPT = 1.) works best with 4 prints per page and with the nominal (PROPT = 0.) works best with 5 prints per page.

The printout interval (ΔP) should be made a multiple of the guidance step size (Δt) such that $\Delta P = n(\Delta t)$ and the print interval on tape ($\Delta P'$) should be made a multiple of the printout interval such that $\Delta P' = n(\Delta P)$.

A random Euler angle conversion matrix [A] is provided for information purposes only at the beginning of each run. Six rotations are allowed with the axes specified by the "μ" input with 1 representing the X-axis, 2 representing the Y-axis and 3 representing the Z-axis. The angles of rotation are specified in degrees, minutes, and seconds by the "α" input. For example, the matrix [A] defined by the product

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & \cos A & \sin A \\ 0 & -\sin A & \cos A \end{bmatrix} \begin{bmatrix} \cos B & 0 & -\sin B \\ 0 & 1 & 0 \\ \sin B & 0 & \cos B \end{bmatrix} \begin{bmatrix} \cos C & \sin C & 0 \\ -\sin C & \cos C & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

can be found from the following input.

```
R1203  DEC   2., 2., 1.
R1211  DEC   C(DEG), C(MIN), C(SEC), B(DEG), B(MIN), B(SEC), A(DEG), A(MIN), A(SEC)
```

These rotations may be taken in any order desired.

5.3 SPECIFIC EXAMPLES

The nominal trajectories to be discussed were generated for performance assessment purposes. It was desired to generate a representative boost trajectory for the following missions:

1. Injection from Circular Orbit at Earth into a trajectory to Mars (5.3.1)
2. Deboost into a Circular Orbit at Mars (5.3.2)
3. Boost into Circular Orbit at Mars (5.3.3)
4. Trajectory Change at Mars (5.3.4)
5. Midcourse Correction (5.3.5)
6. Deboost Prior to Earth Re-entry (5.3.6).

Included with Program 118.0 are copies of each of the above listed nominals. The section numbers in the parenthesis above indicate where the development of that nominal is discussed.

5.3.1 Simulation of Boost From a Circular Orbit about Earth into an Interplanetary Trajectory

PURPOSE

The purpose of this study is to simulate a boost trajectory from a 100 n. mi. circular orbit about Earth into a given interplanetary trajectory to Mars.

INTRODUCTION

The desired hyperbolic trajectory (A) was specified by the impulsive injection conditions from the circular orbit. In order to boost from the circular orbit into this given trajectory (A), it is necessary to match the conditions of radial distance, velocity, and flight path angle simultaneously at some point along trajectory (A) with the boost trajectory (B) at burnout.

PROCEDURE

First, a time history with r, V, and γ is generated for trajectory (A) by Program 118.0 using the coast phase 6 and beginning with the impulsive trajectory conditions. To use the coast phase for obtaining a time history, an integration step size of 2 seconds was used and a print interval of 2 seconds was used up to 300 seconds. The 300-second duration was based on the expected burning time to go from circular speed to hyperbolic speed.

Near optimal boost is obtained by placing the thrust along the velocity vector. This can be accomplished in Program 118.0 by using β^* steering in Phase 3. Developing a trajectory to match the r, V, and γ of trajectory (A) in the equatorial plane allows the trajectory to be shaped by changing the value of Ω. With $\Omega = 0$, a time history of r, V, and γ for a powered flight trajectory is made beginning at X = orbital radius, $\gamma = Z = 0$, $X = Z = 0$, Y = orbital velocity and $\lambda_L = \lambda_{LA} = 0^o$, $\mu_L = 0$, $\mu_{LA} = 90^o$, $\psi_p = 90^o$. The above is repeated for various values of Ω up to $\Omega = V_o/r_o$.

A plot is made of r vs. V for trajectory (A) and the perturbed powered flight trajectories. This plot is made on expanded scale such that only the intersections of the trajectories are shown. The value of γ_A (flight path angle on trajectory A) at the intersection of trajectory A and the powered flight trajectories can be interpolated as well as γ_B (flight path angle on powered flight trajectories).

A plot of γ_A vs. Ω and γ_B vs. Ω is then made where the intersection of these two curves decides the value of Ω that will provide the desired powered flight trajectory where r, V, and γ will match that of trajectory A. Using the value of Ω found above, a powered flight trajectory B is run and a third-order curve fit of $\dot{Q}$ vs. $\dot{R}$ and α_3 vs. $\dot{R}$ is made.

An interpolation of the velocity and position at the intersection of trajectory A and trajectory B is made where the velocity and position of trajectory A is designated $\overline{V}_A$ and $\overline{r}_A$, and the velocity and position of trajectory B is designated $\overline{V}_B$ and $\overline{r}_B$. $\overline{V}_A$ and $\overline{r}_A$ are in the desired coordinate system and V_B and $\overline{r}_B$ are in the coordinate system used for convenience. The following equations are used to place the initial point of trajectory B for coincidence with trajectory A at burnout and set up the reference body axis.

1. $$\vec{U}_{AF} = \frac{\vec{V}_A \times \vec{r}_A}{|\vec{V}_A \cdot \vec{r}_A|}$$

2. $$\vec{U}_{AR} = \frac{\vec{r}_A}{r_A}$$

3. $$\vec{U}_{AV} = \vec{U}_{AR} \times \vec{U}_{AF}$$

4. $$[M_1] = \begin{bmatrix} U_{AV_X} & U_{AV_Y} & U_{AV_Z} \\ U_{AR_X} & U_{AR_Y} & U_{AR_Z} \\ U_{AF_X} & U_{AF_Y} & U_{AF_Z} \end{bmatrix}$$

5. $$\vec{U}_{BF} = \frac{\vec{V}_B \times \vec{r}_B}{|\vec{V}_B \cdot \vec{r}_B|}$$

6. $$\vec{U}_{BR} = \frac{\vec{r}_B}{r_B}$$

7. $$\vec{U}_{BV} = \vec{U}_{Br} \times \vec{U}_{BF}$$

8. $$[M_2] = \begin{bmatrix} U_{BV_X} & U_{BV_Y} & U_{BV_Z} \\ U_{BR_X} & U_{BR_Y} & U_{BR_Z} \\ U_{BF_X} & U_{BF_Y} & U_{BF_Z} \end{bmatrix}$$

9. $$\begin{bmatrix} X'_o \\ Y'_o \\ Z'_o \end{bmatrix} = [M_1]^{-1} [M_2] \begin{bmatrix} X_o \\ o \\ 0 \end{bmatrix}$$

10. $$\begin{bmatrix} \dot{X}'_o \\ \dot{Y}'_o \\ \dot{Z}'_o \end{bmatrix} = [M_1]^{-1} [M_2] \begin{bmatrix} 0 \\ \dot{Y}_o \\ 0 \end{bmatrix}$$

11. $$\vec{V}_o = X'_o \vec{i} + Y_o \vec{j} + Z'_o \vec{k}$$

12. $$\vec{r}_o = X'_o \vec{i} + Y'_o \vec{j} + Z'_o \vec{k}$$

13. $$\vec{R}_{O_o} = \frac{\vec{V}_o}{V_o}$$

14. $$\vec{P}_{I_o} = \frac{\vec{r}_o \times \vec{V}_o}{|\vec{r}_o \times \vec{V}_o|}$$

15. $$\vec{Y}_{A_o} = \vec{R}_{O_o} \times \vec{P}_{I_o}$$

16. $$\lambda_L = \lambda_{LA} = \sin^{-1} R_{OZ_o} \qquad -\frac{\pi}{2} \leq \lambda_{LA} \leq \frac{\pi}{21}$$

17. $$\mu_{LA} - \mu_L = \tan^{-1} \frac{R_{OY_o}}{R_{OX_o}} \qquad -\pi < \mu_{LA} - \mu_L \leq \pi$$

18. $$\psi_p = \tan^{-1} \frac{P_{IZ_o}}{-Y_{AZ_o}} \qquad -\pi < \psi_p \le \pi$$

A check of the results can be made using again Phase 3 of Program 118.0 with the curve fit steering option of OPO123. The R_{STOP4} must be set so that T_4 occurs at the intersection of trajectory A and trajectory B.

CONCLUSION

A test case provided a match of 0.535 fps in V and 11.08 arcseconds in γ. The components at the cutoff point were as follows:

	f(ft)	X (km)	Y (km)	Z (km)	V (fps)	$\dot{X}$ (km/sec)
Free Fall	22323370.	4584.9128	-3411.8570	-3692.5181	38186.614	9.9645575
Powered	22323370.	4579.4113	-3415.4803	-3695.9911	38187.149	9.9712079
Δ	0	5.5015	3.6233	3.4730	-0.535	-0.006504

	Y (km/sec)	Z (km/sec)
Free Fall	3.30580	5.0252319
Powered	3.2948691	5.0195976
Δ	0.0109329	0.0056343

The accuracy of the match depends on the interpolation and curve fit obtained.

5.3.2 Deboost into a Circular Orbit

PURPOSE

The purpose of this study is to simulate a deboost from a hyperbolic trajectory into a circular orbit.

INTRODUCTION

Given the hyperbolic trajectory conditions for an impulsive maneuver to achieve a circular orbit at closest approach, it is desired to simulate a deboost maneuver that will achieve the same orbit. This maneuver is accomplished by applying the thrust against the velocity vector until the vehicle is near the desired orbital radius where terminal curve fit guidance is used until circular orbit is achieved.

PROCEDURE

It is first necessary to obtain a time history of the incoming trajectory from which an initial deboost point can be selected. The time history of the incoming trajectory is obtained by reversing the velocity vector at closest approach and using the coast phase of the booster simulation. The coast is allowed to run for at least the estimated burning time.

An initial deboost point is chosen at a point along the incoming trajectory slightly less than the estimated burning time away from the closest approach point. Deboost is then accomplished in two phases. In the first phase, thrusting is made against the velocity vector to bring the speed of the vehicle down near circular speed near the desired altitude. Orbit is then achieved using the curve fit guidance mode.

The thrust force is reversed by making the $K_{REV} = -1$, and the vehicle is initially aligned along the velocity vector by the inputs of λ_L, λ_{LA}, μ_L, μ_{LA}, ψ_p, $\overline{\alpha}_o$ computed by the following equations

$$\overline{R}_{O_o} = \frac{\overline{V}}{|\overline{V}|} ; \quad \overline{P}_{I_o} = \frac{\overline{r} \times \overline{V}}{|\overline{r} \times \overline{V}|} ; \quad \overline{Y}_{A_o} = \overline{R}_{O_o} \times \overline{P}_{I_o}$$

where $\overline{V}$ is the initial velocity vector, $\overline{r}$ is the initial position vector.

$$\lambda_L = \lambda_{LA} = \sin^{-1} R_{OZ_o} ; \quad \mu_{LA} = \tan^{-1} \frac{R_{OY_o}}{R_{OX_o}} ; \quad \mu_L = 0, \quad \psi_p = \tan^{-1} \frac{P_{IZ_o}}{-Y_{AZ_o}}$$

$$\overline{\alpha}_o = 0$$

The vehicle is kept aligned with the velocity vector by using the β^* guidance in Phase 3. Several deboost runs should be made with different initial points along the incoming trajectory until a run is obtained where the velocity at the orbital radius is slightly above orbital speed. There must be enough excess velocity at the orbital radius to allow the guidance system sufficient time to achieve the proper conditions when orbital speed is reached.

In order to go from β^* steering to time curve fit guidance, it is necessary to go through the following critical time points:

1. T_4
2. TES3
3. T_5.

T_4 is input at the point where β^* steering is to end and TES3 is made to occur immediately after by the MES3 (mass at the end of phase 3) input. MES3 can be computed knowing the mass flow rate and the time at T_4. T_5 is made to occur by use of the r_{TR} input which is set a foot beneath the radial distance at T_4. It is desired to skip from T_4 to T_5 and by making the proper MES3 and r_{TR} inputs, the vehicle will not have moved appreciably during this interval. One other problem exists in between T_4 and T_5, and that is the explicit guidance computations must be satisfied to prevent a computational fault from allowing the run to continue on the computer. It was found that the following conditions allowed the run to be completed.

1. $\bar{R}_A$ = closest approach point
2. $F_{AIM} = 0$, $F_{CON} = 1$
3. $\bar{U}_{TR} = \bar{r}/r$ where $\bar{r}$ is the position vector at T_4
4. p_2 = r circular
5. $K_{13} = K_{14} = 0$
6. $K_Y = 0$
7. $K_{REV} = -1$

Constants of $r = r_{circular}$, $\dot{r} = 0$, $r\dot{v} = V_{circular}$ are used for the trim guidance phase. The $\bar{U}_W$ input is made equal to $\bar{r} \times \bar{v}/|\bar{r} \times \bar{v}|$ where $\bar{r}$ and $\bar{v}$ are the initial position and velocity vectors on the desired circular orbit.

INPUT

The following input must be completed to obtain the incoming time history:

1. Operational data
2. Nominal trajectory data
   a. Δt, δt for Phase 6
   b. number of powered phases
   c. final phase for cutoff
   d. final time for cutoff
   e. special start parameters for Phase 6 ($j = b$, $k = 1$, phase = 6)
3. Vehicle data (phase 7 data only)
4. Physical environment data

This run is ended by the final time for cutoff parameter and the print option (PROPT) is set equal to 1 to obtain the desired history parameters.

The final input must be completed for the deboost phase:

1. Operational data
2. Nominal trajectory data
   - a. ϵ_{TR}, ϵ_{vg}, $\epsilon_s = 10^{-3}$, $\epsilon_u = 1.0$ (other criteria not required)
   - b. j-dependent inputs for j = 1, 2, 3, as required (for j = 1 and 2, see blocks HA4A, HA5, HA6B, and HS1; for j = 3, see blocks HA7 and HS1 from Memo LA-3127)
   - c. Phase 3 and 4 data for phase-dependent inputs
   - d. Special start parameters for Phase 3 (1 1 2, j = 0, k = 1, phase = 3, BETTRG = 0).
   - e. Number of powered phase, final time, phase for cutoff
   - f. Launch conditions.
3. Vehicle data (Phases 3 and 4 data only)
4. Physical environment data
5. Guidance data
   - a. T_4
   - b. Explicit guidance data for j = 1, j = 2; see blocks HA4A, HA5, HA6B, HS1
   - c. Trim curve fit guidance inputs for j = 3 (see blocks HA7, HS1).

CONCLUSION

The method described above provides a nominal deboost trajectory for an incoming vehicle on a hyperbolic trajectory into a given circular orbit. The results of the test case provided a circular orbit with an eccentricity of .0012 which could have been improved further with more trajectory shaping.

5.3.3 Simulation of Launch from Mars into a Circular Orbit

PURPOSE

The purpose of this study is to simulate a boost trajectory of a launch from the surface of Mars into a circular orbit about Mars.

INTRODUCTION

The launch from Mars surface has some similarity to a launch from Earth's surface except for the atmospheric effects. The trajectory consists of a vertical rise to clear obstacles followed by a pitchover period, a zero gravity period, a coast period and a curve fit guided maneuver to achieve final orbit.

PROCEDURE

The launch from Mars into a circular orbit required use of Phases 1, 4, 6, and 7 of Program 118.0. Phases 2, 3, 4, and 5 were unnecessary and were skipped over by the input values.

The run was begun with a 10-second vertical rise phase and then a maximum pitch over phase. The pitchover is controlled by K_{α} and is set up so that $q\alpha$ for the vehicle is not exceeded. For a launch on Mars with the vehicle simulated, it was necessary to have a coast phase since the vehicle would have too much energy for circular orbit if powered continuously. A test run is made to find a point (A) where the vehicle has the energy to reach an apogee slightly above the circular orbital radius and has a flight path angle near 70 degrees. This apogee point is then made the target point for Phases 2, 3, 4, and 5 and T_4 is set to occur about 10 seconds prior to reaching point (A). The end mass for Phase 1 is set to occur just after T_4. Phases 2 and 3 are skipped by making the beginning mass and end mass very close, making the ISP pert = 0 and the flow pert very large. Phase 4 is allowed to continue on until cutoff on required velocity occurs and Phase 5 is automatically skipped. Phase 6 is used until the vehicle is out of the atmosphere and is controlled by the τ_5 input. The data is then set up for trim guidance to begin when r_{TR} is reached when r_{TR} is input as a value slightly less than the desired orbital radius. Trim guidance inputs are made for r = orbital radius, $\dot{r} = 0$, and $r\dot{v}$ = orbital velocity. The final result is the circular orbit desired.

In order to make use of explicit guidance, it is necessary to evaluate p_j, K_{13j}, and K_{14j}. The easiest procedure is to initially make K_{13} a small value like 10^5 and $K14 = 0$, and make "p" a roughly estimated value for the conic through the target poi.it and present position. From a trial run, it is then possible to evaluate K_{13} and K_{14} and obtain a better estimate of p. $(K_{13} + K_{14}S)$ is set equal to $p_i - p_{i-1}/\gamma_i - \gamma_{i-1}$ for several time points and then K_{13} and K_{14} are evaluated. A second trial will give even better results.

CONCLUSION

On a trial case, the burnout errors were 1336 ft-in radius, 27.06 ft/sec in radial rate, 0.068 ft/sec in tangential rate. The eccentricity of the resulting orbit is 0.00244 which gives a difference between apocenter pericenter of 60,000 ft. for a circular orbital radius of 12323102 ft.

5.3.4 Flyby Trajectory Transfer

PURPOSE

The purpose of this study is to simulate by a boost maneuver a given flyby trajectory impulsive transfer.

INTRODUCTION

To match the effect of a given impulsive velocity flyby trajectory transfer with a boost trajectory, the boost must be started prior to the impulsive correction and burnout must occur when the conditions of time, position, and velocity match at some point with the outgoing trajectory after the impulsive maneuver. For the vehicle simulated and the amount of impulsive velocity used for the transfer, an estimate of the boost time can be determined. This estimate is used to determine the lengths of time histories of the impulsive velocity transfer trajectories necessary for boost matching. Curve fit guidance is used for the boost phase which is accomplished in Phase 7 of Program 118.0.

PROCEDURE

An incoming time history of the trajectory before impulsive maneuver is generated by using the coast phase 6 of Program 118.0. The position and velocity just prior to impulsive maneuver are propagated back along the trajectory by reversing the velocity vector. This provides a time history of the incoming trajectory noting that all velocities and times are backwards. The outgoing time history of the trajectory after impulsive maneuver is made propagating the position and velocity just after impulsive maneuver. Both time histories are generated over a time period of the estimated burning period. The parameters, r, $\dot{r}$, $r\dot{v} = V \sin \gamma$ of the outgoing trajectory are curve fit vs. time for the curve fit guidance (Block HA7, see Section IV. 2. 1).
The initial point for the boost phase can be taken to be the point on the incoming trajectory time history at the estimated burning time from impulsive correction. The vehicle is allowed to coast and orient itself at the beginning of Phase 7 until the selected T_7 (initiation of curve fit guidance) is reached. Powered flight starts at T_7 and burnout occurs when velocity-to-be-gained conditions are met. The selection of T_7 will affect the accuracy of the match between the boost burnout and some point along the outgoing trajectory from impulsive correction. A first guess at T_7 could be one-half the estimated burning time from the impulsive correction along the incoming trajectory. By changing T_7 and the adjusting guidance gains, a satisfactory match between the boost burnout and the outgoing trajectory after impulse can be made. The initial conditions can be set up from the initial position ($\bar{v}$) and velocity ($\bar{v}$) of the run from the following equations:

$$R_{O_o} = \frac{\bar{V}}{|\bar{V}|} ; \quad \bar{P}_{I_o} = \frac{\bar{r} \times \bar{v}}{|\bar{r} \times \bar{v}|} ; \quad \bar{Y}_{A_o} = \bar{R}_{O_o} \times \bar{P}_{I_o}$$

$$\lambda_L = \lambda_{LA} = \sin^{-1} R_{OZ_o} ; \quad \mu_{LA} = \tan^{-1} \frac{R_{OY_o}}{R_{OX_o}} ; \quad \mu_L = 0;$$

$$\psi_p = \tan^{-1} \frac{P_{IZ_O}}{-Y_{AZ_O}} ; \quad \bar{U}_w = \frac{\bar{r}_{AI} \times \bar{V}_{AI}}{|\bar{r}_{AI} \times \bar{V}_{AI}|}$$

where $\bar{r}_{AI}$ and $\bar{V}_{AI}$ are the position and velocity vectors just after impulsive correction.

$\bar{\alpha}_o$ is initially set equal to 0.

INPUT

The following input must be completed to obtain the time histories of the incoming and outgoing trajectories for impulsive correction:

1. Operational data
2. Nominal trajectory data
   a. Δt, δt for Phase 6
   b. number of powered phases
   c. final phase for cutoff
   d. final time for cutoff
   e. special start parameters for Phase 6 ($j = 6$, $k = 1$, phase = 6)
3. Vehicle data (Phase 6 data only)
4. Physical environment data.

These runs are to be ended by the final time for cutoff parameter which should be the estimated burning time. The print option (PROPT) should be set equal to 1. to obtain a history of the desired trajectory parameters.

The following input must be completed for simulating the boost phase:

1. Operational data
2. Nominal trajectory data
   a. ϵ_{vg} (other criteria not required)

b. K_A, K_H for j = 7, 8 (all other j-dependent and k-dependent inputs not required)

c. phase 7 and phase 8 data only for phase-dependent inputs

d. special start parameters for phase 7 (j = 7, k = 2, phase = 7, T_6 trigger = 1)

d. number of powered phase, final time, phase for cutoff

f. launch conditions

3. Vehicle data (phase 7 and phase 8 data only)

4. Physical environment data

5. Guidance data (j = 7, 9 data for trim steering only)

This run is ended by setting the final phase for cutoff equal to 8.

CONCLUSION

The method described above provides a satisfactory means for generating a boost maneuver that accomplishes the same mission as the impulsive maneuver. The test case resulted in errors at burnout of 36 ft. in radial distance, 5 fps in radial rate, .05 fps in tangential rate, and 8 arcseconds in flight path angle.

5.3.5 Midcourse Correction

PURPOSE

The purpose of this paper is to describe a method for simulating a given impulsive midcourse maneuver with a booster.

INTRODUCTION

After a space vehicle is launched from a planet, it will gradually drift from the nominal path. A velocity correction can be made at some point to assure arrival at the target point on time. The velocity correction at some point along the actual trajectory can easily be determined; however, it is not possible to make an impulsive maneuver with a vehicle of any significant size. It is, therefore, necessary to generate a boost trajectory which accomplishes the same results as the impulsive maneuver.

PROCEDURE

For a boost maneuver to match an impulsive maneuver, it is necessary for the burnout point of the booster to lie along the trajectory generated by the impulsive correction. In order that time conditions also match, it is necessary to begin boosting somewhere along the incoming trajectory prior to impulsive correction.

The time history of the incoming trajectory is obtained by taking the given conditions just prior to impulsive correction and generate a backward free-fall trajectory. Free-fall trajectories can be generated using Phase 6 of the 118.0 program with the following input:

1. Operational data
   a. general heading
   b. heading for vehicle tables
   c. number of prints per page = 4
   d. run number
   e. NOR6PT = 1
   f. PROPT = 1
   g. CONVCN = 1
   h. ΔP, $\Delta P'$ for Phase 6
2. Nominal trajectory data
   a. number of powered phases = number of last phase used (7)
   b. final phase for cutoff = 7
   c. final time for cutoff (use estimated burning time)
   d. Δt, δt for Phase 6
   e. special start time
   f. special start mass
   g. special start position
   h. special start velocity
   i. special start atmospheric flag = 1
   j. special start parameter j = 6
   k. special start parameter k = 1
   l. special start phase number = 6
   m. special start TES6 (number greater than cutoff time)
3. Vehicle data for Phase 6 (all tables must be fitted)
4. Physical environment data
   a. ATMOPT
   b. P_o

c. R_{C1}

d. a_e

e. ϵ

f. GM

g. Hc

h. g

i. ρ_o

j. β_I

k. C_I

A sample of the input used for a free-fall trajectory is shown in Appendix A. The estimated burning time is found by dividing the impulsive velocity correction by the acceleration of the boost vehicle simulated.

A time history of the outgoing trajectory is obtained as above using the initial condition just after the impulsive correction.

The boost trajectory for this mission was generated by aligning the thrust with the impulsive velocity correction and obtaining a cutoff when conditions along the outgoing trajectory are matched. This maneuver is accomplished by initially aligning the vehicle along the impulsive velocity vector (V_g) with the inputs of λ_L, λ_{LA}, μ_L, μ_{LA}, ψ_p, and $\bar{\alpha}_o$. Using the initial launch position ($\bar{r}_o$), the above quantities are computer as follows:

$$\bar{R}_o = \frac{\bar{V}_g}{|\bar{V}_g|} \; ; \quad P_{I_o} = \frac{\bar{r}_o \times \bar{V}_g}{|\bar{r}_o \times \bar{V}_g|} \; ; \quad Y_{A_o} = \bar{R}_{O_o} \times P_{I_o}$$

$$\lambda_L = \lambda_{LA} = \sin^{-1} R_{OZ_o} \; ; \; \mu_{LA} = \tan^{-1} \frac{R_{OY_o}}{R_{OX_o}} \; ; \; \mu_L = 0$$

$$\psi_p = \tan^{-1} \frac{P_{IZ_o}}{-Y_{AZ_o}}$$

By setting the autopilot gains (K_θ, K_φ, K_ψ) equal to zero, no guidance scheme is actually used to steer the vehicle. In order to sense when boost conditions match with the outgoing trajectory, curve fit guidance which uses second-order curve fits of radial distance (r) vs. time, radial rate ($\dot{r}$) vs. time, and tangential rate ($r\dot{v}$) vs. time is used. This is done in Phase 7, j = 8. The initial launch condition is first chosen to be a point on the inbound trajectory approximately one-half the estimated burning time from the impulsive correction point. The curve fit quantities required for the guidance mode (see Appendix B for equations) are taken from the outbound trajectory time history where $r\dot{v}$ can be computed from $V \sin \gamma$. The values of K_A and K_H, as well as the initial starting point, can be varied until the best match is found. The input required to make this boost run is as follows:

1. Operational data

   a. general heading

   b. heading for vehicle tables

   c. number of time prints per page = 4

   d. run number

   e. NOR6PT = 1

   f. PROPT = 1

   g. CONVCN = 1

   h. ΔP, $\Delta P'$ for Phase 7

2. Nominal trajectory data

   a. ϵ_{vg}

   b. $K_{\alpha Lk} = K_{\gamma Lk} = -\pi$

   c. $K_{\alpha Uk} = K_{\gamma Uk} = \pi$ $\quad k = 2$

   d. K_A, K_H for j = 8

   e. number of powered phases = number of last phase = 8

   f. final phase for cutoff = 8

   g. final time for cutoff = 1.E20

   h. λ_L, λ_{LA}, μ_L, μ_{LA}, ψ_p

   i. Δt, δt for Phase 7

   j. K_θ, K_φ, K_ψ for Phase 7

   k. special start parameter t

   l. special start parameter M

   m. special start position vector

   n. special start velocity vector

   o. special start atmospheric flag = 1

p. $j = 8$, $k = 2$, phase = 7

q. T_6 trigger = 1

3. Vehicle data for Phase 7 (all tables)

4. Physical environment data

a. ATMOPT

b. P_o

c. R_{C1}

d. a_e

e. ϵ

f. GM

g. Ω

h. Hc

i. g

j. ρ_o

k. β_I

l. C_I

5. Guidance data

a. $\overline{U}_w$ for $j = 8$

b. A_{RC} for $j = 8$

c. B_{RC} for $j = 8$

d. C_{RC} for $j = 8$

e. A_{RD} for $j = 8$

f. B_{RD} for $j = 8$

g. C_{RD} for $j = 8$

h. A_{RV} for $j = 8$

i. B_{RV} for $j = 8$

j. C_{RV} for $j = 8$

A sample of the input used here is shown in Appendix C. For ease in running all programs on Program 118.0, a basic data deck has been used consisting of data used to launch a vehicle from a planet's surface into orbit and then into a second orbit.

A run is then made by changing data where necessary, hence, the large amount of input shown in Appendix C. The applicable data for this run is underlined.

INPUT DEFINITION

Operational data consists of appropriate tables, setting up of flags, and data necessary for printing the output. The inputs of Nu and Alpha on the operation data input sheets are not used at present in the program and do not necessarity have to have an input for them.

The nominal trajectory data consists of criteria and special starting parameters. The necessary nominal trajectory data for this run is listed above both for the free-fall and the midcourse correction. The ϵ_{vg} input is used for determining when the velocity-to-be-gained ($\bar{V}_g$) criteria is met from $V_g \cdot V_{gj}/|V_{gj}| \leq \epsilon_{vg}$ (when burnout occurs). ($\bar{V}_{gj}$) is the initial value of $\bar{V}_g$. The $K_{\gamma Lk}$, $K_{\alpha Lk}$, and $K_{\gamma Uk}$, $K_{\alpha Uk}$ are the lower and upper limits used to position and α_1 and α_3 values in the desired quadrants. K_{Aj} and K_{Hj} are guidance gains for the curve fit steering as seen in the equations of Appendix B.

The vehicle data consists of parameters and tables by phase, which describe physical characteristics of the vehicle being simulated. All tables for Phases 6 and 7 must be filled out according to the input sheets supplied for Program 118.0. The input for the bivariate tables, where the output value is constant, can be simplified by taking the first address of the table and, in place of the fixed value of 1, substitute a zero, a comma, and the constant value. For a normal bivariate table input, as shown on the input forms, each space must be filled, even though it means repeating the last points, until the table is filled. The table inputs must cover the proper ranges of the independent variables in order to prevent any faulting for being out of range of a table.

The physical environment data consists of the physical constants for the central body used in the simulation. The values of P_o and ρ_o are sea level values of atmospheric pressure and density in the simple planetary atmospheric model. For the Earth, the more sophisticated atmospheric model may be selected, and the atmospheric constants here are not needed. The R_{CI} constant is the reference radius of the planet being simulated and is used to calculate the surface distance travelled by the vehicle.

The guidance data consists of the curve fit constants shown in Appendix B and the vector ($\bar{U}_w$) indicating the trajectory plane. The value of t_m is the time at the beginning of curve fit steering.

CONCLUSION

The Program 118.0 has many options and is very complex. There are other means available to generate the midcourse guidance nominal with this program. The described method is believed to be one of the better ways of doing the job. The errors at burnout for the test case were 42 ft. in position and .0034 fps in velocity.

5.3.6 Deboost Prior to Earth Re-entry

PURPOSE

The purpose of this study is to simulate a deboost maneuver slowing a vehicle from a hyperbolic trajectory into desireable conditions for a lifting re-entry vehicle.

INTRODUCTION

A given incoming hyperbolic trajectory at Earth is used as initial conditions for the deboost maneuver. Deboost is accomplished by directing the thrust essentially against the velocity vector until the desired terminal conditions are reached.

PROCEDURE

First, we obtain a time history of the incoming hyperbolic trajectory using Phase 6 of Program 118.0. Estimating the time of deboost from the hyperbolic velocity at the top of the atmosphere to the desired re-entry condition and the acceleration of the boost vehicle, we select several initial conditions along the incoming hyperbola.

We now set up Phase 3 of Program 118.0 to use β^* guidance and reverse $\overline{F}_{DA}$. The following conditions are also set up for each initial condition.

$$X_o = r_o; \; Y_o = 0; \; Z_o = 0$$

$$\dot{X}_o = V_o \cos \gamma_o; \; \dot{Y}_o = V_o \sin \gamma_o; \; \dot{Z}_o = 0$$

$$\lambda_L = \lambda_{LA} = 0^o$$

$$\mu_L = 0; \; \mu_{LA} = 0$$

$$\psi_p = 90^o$$

$$GM = g_o r_o^2$$

$$a_e = r_o - h_o$$

$$\epsilon = 0$$

Various runs are made with various values of Ω from zero to v_o/r_o. This provides deboosting along the velocity vector for $\Omega = 0$ and slightly perturbed runs for other values of Ω. By interpolating the value of the flight path angle (γ) and velocity at the desired height for the various values of Ω, we can obtain the velocity resulting at the desired flight path angle at re-entry. With the same information from various initial boost points, it is easy to find the initial point and Ω value that will provide the proper re-entry velocity and flight path angle. By making third-order curve fits of $\dot{Q}$ vs. $\dot{R}$ and α_3 vs. $\dot{R}$ from the desired deboost run, we can now use the $\dot{P}$, $\dot{Q}$, $\dot{R}$ steering mode in Phase 3 and begin at the initial point on the incoming hyperbola. The vehicle must be aligned initially by the λ_L, μ_L, λ_{LA}, μ_{LA}, ψ_p values computed as follows.

$$R_{O_o} = \frac{\overline{V}_o}{V_o} \; ; \quad \lambda_L = \lambda_{LA} = \sin^{-1} R_{OZ_o} \; ; \quad \overline{P}_{I_o} = \frac{\overline{r}_o \times \overline{v}_o}{|r_o \times v_o|} \; ;$$

$$\mu_L = 0^o ; \quad \mu_{LA} = \tan^{-1} \frac{R_{OY_o}}{R_{OX_o}} \; ; \quad \overline{Y}_{A_o} = \overline{R}_{O_o} \times \overline{P}_{I_o} \; ;$$

$$\psi_p = \tan^{-1} \frac{P_{IZ_o}}{-Y_{AZ_o}}$$

CONCLUSION

It is possible to match the conditions for many re-entry vehicle configurations with the above method. The test case slowed the vehicle from 42,436 fps to 30,273 fps at the re-entry altitude of 401,346 ft. with a flight path angle of 6°19'. The desired end conditions were 30,000 fps at 400,000 ft. altitude with a flight path angle of 6°. The conditions obtained were considered to be satisfactory for the vehicle under study.

6.0 REFERENCES

(1) U.S. Standard Atmosphere 1962: United States Committee on Extension to the Standard Atmosphere (COSEA);
K.S.W. Champion, Editor

APPENDIX A

```
R0       BCI      ,CCAST TEST CASE
R12      BCI      ,PHASE 6 TABLES
R1176    DEC      9,5,6,1.E19,C
R1177    DEC      4
R1200    DEC      7.,10.
R1202    DEC      1.
R1233    DEC      .162345E-2,.575E-5,.7875E-5
R1237    DEC      1.,1.
R1244    DEC      1.,0,0,0,0,0,0
R1245    DEC      14.7,344C.5
R1253    DEC      2C92569C.,.33523298E-2,.14076455E17
R1253    DEC      .2092569E8
R1256    DEC      .72921157E-4,.7E6,0,0,32.174
R1256    DEC      .72921158E-4
R1564    DEC      0,10.3,78.5,1.,0,25.,25.,0
R1571    DEC      2.,2.
R1605    DEC      2.,2.
R2366    DEC      C,1CCC.
R2374    DEC      0,10038.,0,1C038.,1000.,10038.
R3247    DEC      0,10000.
R3255    DEC      0,-1.594,0,-1.594,1CCC0.,-1.594
R413C    DEC      2.,50.
R4136    DEC      2.,0.23,2.,0.23,50.,0.23
R5011    DEC      33.36,1144.19
R5017    DEC      33.36,2.29,33.36,2.29,233.36
         DEC      4.22,944.19,11.C8,1144.19,13.01
R6236    DEC      0,0.04
R7436    DEC      0,70.
R10016   DEC      0,847.
R1CC20   DEC      21525690.,C,0
R1C023   DEC      0,25500.,C
R10031   DEC      1.                      OUT OF ATMOS
R1C035   DEC      6,1,6
R10053   DEC      12.
         TRA      5,4
```

APPENDIX B

Guidance Equations for Trim Curve Fit Guidance

Shaping Inputs:

a) $\overline{U}_{wj}$

b) $A_{RC_j}, B_{RC_j}, C_{RC_j}$

c) $A_{RD_j}, B_{RD_j}, C_{RD_j}$

d) $A_{RV_j}, B_{RV_j}, C_{RV_j}$

e) k_{A_j}, k_{H_j}

f) $k_{2_j}, k_{3_j}, k_{4_j}, k_{5_j}$

g) $A_{\theta_j}, B_{\theta_j}, C_{\theta_j}$

h) $K_{\alpha Lk}, K_{\alpha Uk}, K_{\gamma Lk}, K_{\gamma Uk}$

Computed Inputs:

a) $\overline{r}$

b) $\overline{V}$

c) t_m

d) $\overline{R}_{Oo}, \overline{P}_{Io}, \overline{Y}_{Ao}$

Output: $\alpha_{1c}, \alpha_{2c}, \alpha_{3c}$ (gimbal angle commands)

1) $\overline{U}_r = \dfrac{\overline{r}}{|\overline{r}|}$

2) $\overline{U}_v = \overline{U}_{wj} \times \overline{U}_r$

3) $\Delta v = (t - t_m)$

4) $r_c = A_{RC_j} + B_{RC_j}(\Delta v) + C_{RC_j}(\Delta v)^2$

5) $\dot{r}_c = A_{RD_j} + B_{RD_j}(\Delta v) + C_{RD_j}(\Delta v)^2$

6) $(r\dot{v})_c = A_{RV_j} + B_{RV_j}(\Delta v) + C_{RV_j}(\Delta v)^2$

7) $\overline{V}_{REQ} = \dot{r}_c \overline{U}_r + (r\dot{v})_c \overline{U}_v + k_{A_j}[\dot{r}_c - \dot{r} + k_{H_j}(r_c - r)]\overline{U}_r$

8) $\overline{V}_g = \overline{V}_{REQ} - \overline{V}$

9) $\alpha_{1vg_j} = \tan^{-1}\left[\frac{\bar{V}_g \cdot P_{I_o}}{-V_g \cdot \bar{V}_{A_o}}\right] \quad K_{\alpha L_k} < \alpha_{1vgj} \le K_{\alpha Uk}$

10) $\alpha_{2vg_j} = 0$

11) $\bar{Y}'_A = \cos(\alpha_{1vg_j})\,\bar{Y}_{A_o} - \sin(\alpha_{1vg_j})\,\bar{P}_{I_o}$

12) $\alpha_{3vg_j} = \tan^{-1}\left[\frac{-V_g \cdot Y'_A}{V_g \cdot \bar{R}_{O_o}}\right] \quad K_{\gamma Lk} < \alpha_{3vg_j} \le K_{\gamma Uk}$

Computed only at the initial time

13) $\bar{U}_{vg_j} = \frac{\bar{V}_{gj}}{|V_{gj}|}$

14) $\Delta\bar{R} = \bar{V}_g \times \bar{U}_{vg_j}$

15) $P'_I = (\cos\alpha_1)\,P_{I_o} + (\sin\alpha_1)\,Y_{A_o}$

16) $\Delta\alpha_{1c} = -k_{2_j}(\Delta\bar{R} \cdot \bar{R}_{O_o}) - k_{3_j}\,\Sigma\,(\Delta\bar{R} \cdot \bar{R}_{O_o})_i \Delta t_i$

17) $\Delta\alpha_{2c} = 0$

18) $\Delta\alpha_{3c} = -k_{4_j}(\Delta\bar{R} \cdot \bar{P}'_I) - k_{5_j}\,\Sigma(\Delta\bar{R} \cdot P'_I)_i \Delta t_i$

19) $\theta_k = [A_{\theta_j} + B_{\theta_j}|V_g| + C_{\theta_j}|V_g|^2]\,\bar{U}_r \times \bar{U}_{vg_j}$

20) $\alpha_{1c} = \Delta\alpha_{1c} + (\bar{\theta}_k \cdot \bar{R}_{O_o}) + \alpha_{1vg_j}$

21) $\alpha_{2c} = 0$

22) $\alpha_{3c} = \Delta\alpha_{3c} + (\theta_k \cdot \bar{P}'_I) + \alpha_{3vg_j}$

23) Burnout occurs when $\frac{(V_g \cdot \bar{U}_{vgj})_i\,(\Delta t_i)}{(\bar{V}_g \cdot U_{vgj})_{i-1} - (V_g \cdot \bar{U}_{vgj})_i} \le \epsilon_{vg}$

APPENDIX C

```
R0      BCI      ,MIDCOURSE CORRECTION
R12     BCI  ,DYNAMICS TABLES FOR REFERENCE SPACE VEHICLE
R24     DEC      10.,80.,120.
R27     DEC      0,45.,0,0
R33     DEC      -.002,0,.002
R36     DEC      24.144252,.97336,-.16674869E-3,.79875263E-8,0
R43     DEC      -.047209984,.26981997E-3,-.19746393E-7,.42454905E-12
R47     DEC      6.075,383.61026,1.,6890.5,1.,3360.,1.E8,1.E9
R57     DEC      18345.,22884.,1.E9
R62     DEC      1.E-4,1.E-4,1.E-4,100.,1.E-2,1.E-4,1.E-3,1.
R72     DEC      -3.1415927,3.1415927,0,4.53685618
R76     DEC      -3.1415927,3.1415927,-1.7453293,4.5368562
R102    DEC      -3.1415927,3.1415927,0,4.5368562
R106    DEC      0,0,0,0,0,1.,0,1.,1.,0,1.,0
R122    DEC      1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,0
R136    DEC      1.,1.,0,1.,0,0,0,0,0,0,0,0
R152    DEC      21514670.,21514670.,1.E19,21514670.
R156    DEC      1.E19,1.E19,22137677.
R161    DEC      1.E19,1.E19,1.E19,1.E19,1.E19
R166    DEC      0,0,0,0,0,0,1.,0,0,0,0,0
R202    DEC      0,0,0,0,0,0,1.,0,0,0,0,0
R216    DEC      1.,1.,1.,1.,1.,1.,0,1.,1.,1.,1.,1.
R232    DEC      1.E19,1.E19,1.E19,1.E19,1.E19,1.E19
R240    DEC      1.E19,1.E19,1.E19,1.E19,1.E19,1.E19
R246    DEC      0,0,0,0,0,0,0,0,0,1.,0,0
R262    DEC      0,0,0,0,0,0,0,0,0,0,0,0
R276    DEC      1.,1.,1.,1.,1.,1.,1.,1.,1.,0,1.,1.
R312    DEC      18617797.,18617797.,18617797.,18617797.,18617797.,0
R320    DEC      48829226.,0,0,-129411220.,0,0
R326    DEC      6263103.4,6263103.4,6263103.4,6263103.4,6263103.4,0
R334    DEC      129430080.,0,0,48708317.,0,0
R342    DEC      8791097.7,8791097.7,8791097.7,8791097.7,8791097.7,0
R350    DEC      151127.40,0,0,0,0,0
R356    DEC      0,0,0,0,0,0,18963.,0,0,0,0,0
R372    DEC      -.47009006,-.47009006,-.47009006,-.47009006,-.47009006
R377    DEC      -.47009006,-.47009006,-.47009006,-.47009006,0,0,0
R400    DEC      -.062267732,-.062267732,-.062267732
R406    DEC      .17364818,.17364818,.17364818,.17364818,.17364818
R413    DEC      .17364818,.17364818,.17364818,.17364818,0,0,0
R414    DEC      -.77159783,-.77159783,-.77159783
R422    DEC      .86536792,.86536792,.86536792,.86536792,.86536792
R427    DEC      .86536792,.86536792,.86536792,.86536792,1.,1.,1.
R430    DEC      .63305570,.63305570,.63305570
R436    DEC      0,0,0,0,0,0,0,0,0,0,0,0
R452    DEC      0,0,0,0,0,0,0,0,0,0,0,0
R466    DEC      14094540.,15355522.,14374725.,14374725.,14374725.,0
R474    DEC      37145775.,0,0,138.38233E6,0,0
R502    DEC      .2895E9,.2895E9,.2895E9,.2895E9
R506    DEC      0,0,0,0,0,1000.,0,0
R516    DEC      -.7125E9,-.7125E9,-.7125E9,-.7125E9,.95E8,0
R524    DEC      .41204E7,0,0,0,0,0
R532    DEC      21520871.,21520871.,21520871.,21520871.,21520871.,0
R540    DEC      0,22143638.,0,.13838233E9,138274230.,0
R540    DEC      .28847308E9,.28847308E9,.28847308E9
R546    DEC      0,0,0,0,0,0,0,0,0,0,0,0
R554    DEC      0,16876.144,16876.144
R562    DEC      0,0,0,0,0,0,0,0,0,0,0,0
R570    DEC      0,4.5413161,4.5413161
R576    DEC      0,0,0,0,0,0,0,5626.4712,0,0,0,0
R604    DEC      16901.079,16901.079,16901.079
```

```
R612    DEC     0,0,0,0,0,0,0,0,0,0,0,0
R620    DEC     0,-.14031279,-.14031279
R626    DEC     0,0,0,0,0,0,0,0,0,0,0,0
R634    DEC     0,.34239875E-3,.34239875E-3
R642    DEC     25575.06,25575.06,25575.06,25575.06,25575.06,0
R650    DEC     0,32620.318,0,10087.5,10089.6,0
R650    DEC     2942.7327,2942.7327,2942.7327
R656    DEC     0,0,0,0,0,0,0,0,0,0,0,0
R664    DEC     0,-.16633855,-.16633855
R672    DEC     0,0,0,0,0,0,0,0,0,0,0,0
R700    DEC     0,-.11698624E-2,-.11698624E-2
R706    DEC     1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.
R722    DEC     0,0,0,0,0,0,0,0,0,0,0,0
R736    DEC     .00144,.00226,.00226,.01365,.01365,.033
R744    DEC     .033,.033,.02,.02,.02,0
R752    DEC     .9E-4,.142E-3,.142E-3,.00171,.00171,.00296
R760    DEC     .00296,.00296,.0126,.0126,.0126,0
R766    DEC     .00144,.00226,.00226,.01365,.01365,.011
R774    DEC     .011,.011,.02,.02,.02,0
R1002   DEC     .9E-4,.142E-3,.142E-3,.00171,.00171,.98E-3
R1010   DEC     .98E-3,.98E-3,.0126,.0126,.0126,0
R1016   DEC     .12,0,0,0,0,0,0,0,0,0,0,0
R1024   DEC     0,0,0
R1032   DEC     -.3E-4,-.445E-4,0,0,-.55E-3,0,0,0,0,0,0,0
R1040   DEC     0,0,0
R1046   DEC     0,0,0,0,0,0,0,0,0,0,0,0
R1054   DEC     0,0,0
R1062   DEC     .11864424E9,.11864424E9,.11864424E9,.11864424E9
R1066   DEC     .11864424E9,.11867783E9,.11867783E9,.11867783E9
R1072   DEC     .11867783E9,.11867783E9,.11867783E9,0
R1076   DEC     .84285594E-8,.84285594E-8,.84285594E-8,.84285594E-8
R1102   DEC     .84285594E-8,.84341783E-8,.84341783E-8,.84341783E-8
R1106   DEC     .84341783E-8,.84341783E-8,.84341783E-8,0
R1176   DEC     9,4,8,1.E19,1.
R1177   DEC     5
R1203   DEC     3.,2.,3.,3.,2.,0
R1211   DEC     0,0,0,61.,29.,17.,180.,0,0,-99.,24.,0,-45.,0,0
R1233   DEC     .162345E-2,.575E-5,.7875E-5
R1236   DEC     0,0,0,0,0,0
R1244   DEC     1.,0,0,0,0,0,0
R1247   DEC     1.,1.
R1245   DEC     14.7
R1246   DEC     3440.5
R1253   DEC     20925690.,.33523298E-2,.14076455E17
R1253   DEC     .2092569E8
R1256   DEC     .72921157E-4,.7E6,0,0,32.174
R1256   DEC     .72921158E-4
R1263   DEC     -25.,0,25.524
R1266   DEC     0,0,0
R1271   DEC     -25.,0,25.524
R1274   DEC     -159.,5.,35.858
R1277   DEC     137.,52.,39.427
R1302   DEC     0,0,0
R1367   DEC     41885.146,104.17,236.,1.,1.,1.,.25,15678.146
R1377   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.
R1410   DEC     5.,5.
R1420   DEC     15678.146,105.,236.,1.,1.,1.,.25,15053.789
R1430   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.
R1441   DEC     5.,5.
R1451   DEC     11143.8,105.,78.5,1.,1.,1.,.25,3597.37
```

```
R1461   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.
R1502   DEC     3149.15,43.33,78.5,1.,1.,1.,.25,1085.99
R1512   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.
R1523   DEC     2.,2.
R1533   DEC     894.19,10.3,78.5,1.,1.,1.,.25,0
R1543   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.
R1554   DEC     2.,2.
R1564   DEC     0,10.3,78.5,1.,0,25.,25.,0
R1574   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.,1000.,1000.
R1615   DEC     0,10.3,78.5,1.,0,1.,.25,0
R1621   DEC     1.
R1622   DEC     .25,.0625
R1625   DEC     0,0,0,0,0,0,0,0,0,.25,.25
R1636   DEC     .25,.25
R1646   DEC     0,10.3,78.5,1.,0,25.,25.,0
R1656   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.,5000.,5000.
R1677   DEC     0,10.3,78.5,1.,0,1.,.25,0
R1707   DEC     1.,40.,0,3.,1.,3.,1.,3.,1.,100.,100.
R1731   DEC     0,14.73
R1737   DEC     0,7998.,0,7998.,14.70,7155.,14.73,7155.
R2022   DEC     0,14.7
R2030   DEC     0,8719.,0,8719.,14.7,7722.
R2113   DEC     0,14.7
R2121   DEC     0,9098.8,0,9098.8,14.7,8262.3
R2204   DEC     0,1000.
R2212   DEC     0,9942.,0,9942.,1000.,9942.
R2275   DEC     0,1000.
R2303   DEC     0,10038.,0,10038.,1000.,10038.
R2366   DEC     0,1000.
R2374   DEC     0,10038.,0,10038.,1000.,10038.
R2433   DEC     0,1000.
R2441   DEC     0,10038.,0,10038.,1000.,10038.
R2500   DEC     0,1000.
R2506   DEC     0,10038.,0,10038.,1000.,10038.
R2545   DEC     0,1000.
R2553   DEC     0,10038.,0,10038.,1000.,10038.
R2612   DEC     0,26623.05
R2620   DEC     0,-315.674,0,-315.674,7018.40,-269.193
        DEC     12549.67,-233.648,15356.44,-234.146
        DEC     20096.04,-217.244,22682.07,-213.763
        DEC     25634.77,-208.047,26623.05,0
R2703   DEC     0,624.36
R2711   DEC     0,-172.506,0,-172.506,624.36,-52.058
R2774   DEC     0,10000.
R3002   DEC     0,-52.058,0,-52.058,10000.,-52.058
R3065   DEC     0,10000.
R3073   DEC     0,-10.115,0,-10.115,10000.,-10.115
R3156   DEC     0,10000.
R3164   DEC     0,-1.594,0,-1.594,10000.,-1.594
R3247   DEC     0,10000.
R3255   DEC     0,-1.594,0,-1.594,10000.,-1.594
R3314   DEC     0,10000.
R3322   DEC     0,-1.594,0,-1.594,10000.,-1.594
R3361   DEC     0,10000.
R3367   DEC     0,-1.594,0,-1.594,10000.,-1.594
R3426   DEC     0,10000.
R3434   DEC     0,-1.594,0,-1.594,10000.,-1.594
R3473   DEC     0,10.
R3501   DEC     0,.12,0,.12,.178,.123,.263,.124,.551,.133
        DEC     .888,.201,1.1,.363,1.288,.36,1.801,.339,2.313,.329
        DEC    2.9,.319,3.516,.31,4.073,.302,4.657,.295,4.677,.295
        DEC    5.,.294,10.,.292
R3564   DEC    0,10.
R3572   DEC    0,.12,0,.12,.178,.123,.263,.124,.551,.133
        DEC    .888,.201,1.1,.363,1.288,.36,1.801,.339,2.313,.329
        DEC    2.9,.319,3.516,.31,4.073,.302,4.657,.295,4.677,.295
        DEC    5.,.294,10.,.292
R3655   DEC    2.,25.
R3663   DEC    2.,.47,2.,.47,3.,.26,4.,.245,25.,.245
R3746   DEC    2.,50.
R3754   DEC    2.,.23,2.,.23,50.,.23
R4037   DEC    2.,50.
R4045   DEC    2.,.23,2.,.23,50.,.23
R4130   DEC    2.,50.
R4136   DEC    2.,0.23,2.,0.23,50.,0.23
R4175   DEC    2.,50.
R4203   DEC    2.,0.23,2.,0.23,50.,0.23
R4242   DEC    2.,50.
R4250   DEC    2.,0.23,2.,0.23,50.,0.23
R4307   DEC    2.,50.
R4315   DEC    2.,0.23,2.,0.23,50.,0.23
R4354   DEC    15528.15,42135.09
R4362   DEC    15528.15,67.98,15528.15,67.98,15728.15
        DEC    68.02,41935.09,73.42,42135.09,73.46
R4445   DEC    14903.79,15928.15
R4453   DEC    14903.79,67.49,14903.79,64.79,15103.79
        DEC    67.62,15728.15,68.02,15928.15,68.15
R4536   DEC    3447.37,11393.80
R4544   DEC    3447.37,31.44,3447.37,31.44,3647.37
        DEC    32.29,11193.80,64.46,11393.80,65.31
R4627   DEC    935.99,3399.15
R4635   DEC    935.99,13.49,935.99,13.49,1135.99
        DEC    14.56,3199.15,25.61,3399.15,26.68
R4720   DEC    33.36,1144.19
R4726   DEC    33.36,2.29,33.36,2.29,233.36
        DEC    4.22,944.19,11.08,1144.19,13.01
R5011   DEC    33.36,1144.19
R5017   DEC    33.36,2.29,33.36,2.29,233.36
        DEC    4.22,944.19,11.08,1144.19,13.01
R5056   DEC    33.36,1144.19
R5064   DEC    33.36,2.29,33.36,2.29,233.36
        DEC    4.22,944.19,11.08,1144.19,13.01
R5123   DEC    33.36,1144.19
R5131   DEC    33.36,2.29,33.36,2.29,233.36
        DEC    4.22,944.19,11.08,1144.19,13.01
R5170   DEC    33.36,1144.19
R5176   DEC    33.36,2.29,33.36,2.29,233.36
        DEC    4.22,944.19,11.08,1144.19,13.01
R5235   DEC    0,.7E6                  (NORTH WIND TABLE)
R5243   DEC    0,100.,0,100.,.7E6,100.
R5326   DEC    0,.7E6
R5334   DEC    0,100.,0,100.,.7E6,100.
R5416   DEC    1,0,.0349066,.1047198,.1745329,.1
        DEC    0,0,.16,.32,1.12,.5,0,.16,.34,1.19
        DEC    1.,0,.16,.36,1.32,1.12,0,.16,.4,1.48
        DEC    1.25,0,.16,.4,1.5,1.5,0,.16,.39,1.52
        DEC    2.,0,.16,.38,1.52,4.,0,.18,.36,1.5
        DEC    6.,0,.2,.3,1.38,10.,0,.16,.25,1.31
R5536   DEC    1,0,.0349066,.1047198,.1745329,.1
        DEC    0,0,.16,.32,1.12,.5,0,.16,.34,1.19
        DEC     1.,0,.16,.36,1.32,1.12,0,.16,.4,1.48
        DEC     1.25,0,.16,.4,1.5,1.5,0,.16,.39,1.52
        DEC     2.,0,.16,.38,1.52,4.,0,.18,.36,1.5
        DEC     6.,0,.2,.3,1.38,10.,0,.16,.25,1.31
R5656   DEC     1,0,.0349066,.1745329,.5235988,.1
        DEC     1.5,0,.1,.55,2.09,2.,0,.1,.56,2.16
        DEC     4.,0,.12,.61,2.4,8.,0,.11,.63,2.51
        DEC     15.,0,.08,.5,2.46,25.,0,.06,.39,2.03
R5776   DEC     0,.05
R6116   DEC     0,.04
R6236   DEC     0,0.04
R6332   DEC     0,0.04
R6426   DEC     0,0.04
R6522   DEC     0,0.04
R6616   DEC     1,0,.0349066,.1047198,.1745329,.1
        DEC     0,72.3,74.8,78.2,81.,.5,72.8,75.3,79.7,82.7
        DEC     1.,75.4,78.6,83.1,84.2,1.12,61.7,68.9,76.4,79.
        DEC     1.25,61.8,69.2,76.9,79.3,1.5,63.2,71.3,77.3,79.9
        DEC     2.,64.7,72.4,78.,80.4,4.,65.9,73.4,78.6,81.
        DEC     6.,66.7,73.7,78.7,81.1,10.,64.4,73.2,77.8,80.2
R6736   DEC     1,0,.0349066,.1047198,.1745329,.1
        DEC     0,72.3,74.8,78.2,81.,.5,72.8,75.3,79.7,82.7
        DEC     1.,75.4,78.6,83.1,84.2,1.12,61.7,68.9,76.4,79.
        DEC     1.25,61.8,69.2,76.9,79.3,1.5,63.2,71.3,77.3,79.9
        DEC     2.,64.7,72.4,78.,80.4,4.,65.9,73.4,78.6,81.
        DEC     6.,66.7,73.7,78.7,81.1,10.,64.4,73.2,77.8,80.2
R7056   DEC     1,0,.0349066,.1745329,.5235988,.1
        DEC     1.5,36.5,36.9,37.2,38.,2.,37.1,37.8,38.,38.6
        DEC     4.,37.7,38.2,38.4,38.8,8.,38.4,38.9,39.1,39.1
        DEC     15.,38.8,39.,39.1,39.1,25.,37.2,38.5,38.7,38.8
R7176   DEC     0,19.
R7316   DEC     0,7.
R7436   DEC     0,70.
R7532   DEC     0,70.
R7626   DEC     0,70.
R7722   DEC     0,70.
R10016 DEC     0,847.8              T,M
R10020 DEC     137550820.,154107750.,201363310.
R10023 DEC     5558.0115,10075.372,12821.895
R10026 DEC     3.2627713E-5,0,.2382603744
R10030 DEC     0
R10031 DEC     1.
R10032 DEC     0,0,0
R10035 DEC     8,2,7
R10040 DEC     0
R10054 DEC     0,0,1.
R1202   DEC     2.
        TRA     5,4
```

APPENDIX D

COMPUTER PROGRAM 118.0

D.1 INTRODUCTION

The material in this appendix is designed to furnish the data required by a programmer for a detailed understanding of Program 118.0. A complete program listing is given, supplemented by descriptions of utility subroutines, tape format, and operational guide for an IBM 7094 MOD II. This data will facilitate the operational application of Program 118.0 and aid the programmer in possible future modifications or additions.

D.2 PROGRAMMER'S OPERATIONAL INFORMATION

D.2.1 SYSTEM CONFIGURATION (IBSYS VERSION 13)

FUNCTION	SYMBOL	PHYSICAL	FORTRAN IV
Library 1	SYSLB1	A1	
Library 2	SYSLB2	Unassigned	
Library 3	SYSLB3	Unassigned	
Library 4	SYSLB4	Unassigned	
Card Reader	SYSCRD	RDA	
On-line Printer	SYSPRT	PRA	
Card Punch	SYSPCH	A0	
Output	SYSOU1	A3	6
Alternate Output	SYSOU2	A3	
Input	SYSIN1	B3	5
Alternate Input	SYSIN2	B3	
Peripheral Punch	SYSPP1	B4	7
Alternate Peripheral Punch	SYSPP2	B2	
Check Point	SYSCK1	B5	
Alternate Check Point	SYSCK2	B5	
Utility 1	SYSUT1	A4	1
Utility 2	SYSUT2	B1	2
Utility 3	SYSUT3	A2	3
Utility 4	SYSUT4	B2	4
Utility 5	SYSUT5	Unassigned	
Utility 6	SYSUT6	Unassigned	
Utility 7	SYSUT7	Unassigned	
Utility 8	SYSUT8	Unassigned	
Utility 9	SYSUT9	Unassigned	

Attached Units Not Assigned or Reserved

A5	B6
A6	B7
A7	B8
A8	B9
A9	B0

Intersystem Reserve Units

None

D.2.2 PROGRAMMER'S GUIDE FOR 118.0

I. Error Conditions

A. Error conditions result in program dump in floating point from with XR4 being key to location where error occurred.

B. Hints on error conditions

1) Make very sure input data is correct

2) XR4 should tell where error occurred

3) Make sure end cards TRA 5,4, TRA 4,4 are present and used correctly

4) Double check deck makeup

5) Table input data is frequent error.

D.2.3 OPERATOR'S GUIDE FOR PROGRAM 118.0

I. Machine Configuration (System Requirements)

A. Channel A

1) A1 = IBSYS VERSION 13

2) A2 = Utility tape

3) A3 = List tape (output print tape)

4) A4 = Utility tape

B. Channel B

1) B1, B2, and B5 = Utility tape

2) B3 = Card-to-tape (input)

3) B4 = Punch tape (card output)

C. Core Storage

1) 32K

II. Deck Setup

A. Control Cards at Beginning of Deck

1) $ DATE

2) $IBSYS

3) $RESTORE

4) $JOB

5) $ID

6) $EXECUTE

7) $IBJOB

B. Program Decks

1) 118

2) SUBS

3) XTAPEW

4) AREW

C. Control Cards at End of Program Deck

1) $ENTRY

2) 7/8

3) Data Cards (NOT CONTROL CARDS)

4) 7/8

5) $IBSYS

6) $ENDFILE SYSOU1

7) $ENDFILE SYSPP1

8) $STOP

D. Devices Used by Program

1) Fortran logical 2 tape unit = B1

E. Built-in Pauses

1) On-line message "Please place blank tape on S.SUD2" means to be sure blank tape is on B1, then hit start.

2) Pause 66666 with 66666 right justified in AC means to save tape generated on B1-Fortran logical 2, if this option asked for, then hit start.

D.2.4 Tape Format for 118.0 (58 Word Record)

Record No. 1 - Heading Record (All Floating or BCI)

1) -3.0
2) Run No.
3) Phase No.
4-13) Heading
14-54) 0
55-58) 4 BCI Blank Words

Record No. 2

1) -2.0
2) Run No.
3) Phase No.

$$\left.\begin{array}{rl} 4) & P_{IX_o} \\ 5) & P_{IY_o} \\ 6) & P_{IZ_o} \\ 7) & Y_{AX_o} \\ 8) & Y_{AY_o} \\ 9) & Y_{AZ_o} \\ 10) & R_{OX_o} \\ 11) & R_{OY_o} \end{array}\right\} \quad A_4 \text{ matrix in row form}$$

12) R_{OZ_o}
13-54) 0
55-58) 4 BCI Blank Words

Record No. 3 and All Others Except Special End Records

1) TIME

2) Run No.

3) Phase No.

4-6) ω_{PI}, ω_{YA}, ω_{RO}

7-12) X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$

13-18) a_x, a_y, a_z, α_1, α_2, α_3

*19-54) φ_{11}^{-1}, φ_{21}^{-1}, φ_{31}^{-1}, φ_{41}^{-1}, φ_{51}^{-1}, φ_{61}^{-1} In Error Analysis Program, read in as Transpose

φ_{12}^{-1}, ----- -----

* Note: $(t_o - t)$ is in φ_{41}^{-1}, φ_{52}^{-1}, φ_{63}^{-1} This option is built in error analysis portion of program

55-58) 4 BCI words

Record N (Record Separating Cases) Special End Record

1) TIME = 1.E20

2) Run No.

3) Phase No.

4) Remaining words are garbage thru word 58

Record (N+1)

1) If end of all runs, duplicate of Record N

2) If another case, same format as preceding case on tape.

D. 2. 5 Utility Subroutines

Page 1

Utility Subroutines Used in Programs 118.0 (Nominal Boost) and Program 117.0 (Performance Assessment All-Inertial Boost)

	Name	Brief Description of Subroutine
1)	SINCOS	Entry point for sine and cosine subroutines (SIN, COS).
2)	AF2F	Entry point for arctangent and arcsine subroutines (ATAN, ASIN).
3)	BVPI	Bivariate extrapolation subroutine.
4)	INTP	Linear Interpolation subroutine.
5)	INTG	Integration subroutine.
6)	RADEG	Conversion from radians to degrees, minutes, and seconds.
7)	DEGRA	Conversion from degrees, minutes, and seconds to radians.
8)	TRANSF	Transformation of 3 dimensional vector.
9)	DOT	Dot product subroutine.
10)	CROSS	Cross product subroutine.
11)	SIXROT	6 Rotational matrix evaluator
12)	EATM	Extended atmosphere subroutine.
13)	INP9	Variable field input subroutine.
14)	PRT9	Output subroutine.
15)	PFIX	Fixes floating point numbers.
16)	PFLOAT	Floats fix point numbers.
17)	SQRT	Square root subroutine.
18)	COS	Cosine subroutine.
19)	SIN	Sine subroutine.
20)	ATAN	Arctangent subroutine.

Utility Subroutines Used in Programs 118.0 (Nominal Boost) and Program 117.0 (Performance Assessment All-Inertial Boost)

	Name	Brief Description of Subroutine
21)	ASIN	Arcsine subroutine.
22)	LN	Natural logarithm subroutine.
23)	EXP	Exponent subroutine (e^x).
24)	CORE	Dump subroutine (floating or octal).
25)	PANEL	Panel display subroutine.
26)	SETUP	Generalized setup subroutine.
27)	TAPEW	Writes time point history tape in Fortran IV and prints tape after completion of all cases.

Date 3 / 16/ 66
Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Entry points for sine (SIN) and cosine (COS) subroutine

Subroutine Name: SINCØS

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

□ See Attached Listing
☒ See Attached Information
☒ Following Comments

This (SINCØS) routine is nothing but a few instructions that allow the programmer to find the sine and cosine of an angle in radians at one entrance to a subroutine (SINCØS). The sine is in the AC and the cosine is in the MQ upon exit. The calling sequence is as follows:

```
CLA   X
TSX   SINCØS,4
```

where X = location of angle in radians.

See SIN and CØS writeups.

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Entry points for ATAN and ASIN subroutines

Subroutine Name: AF2F

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

- □ See Attached Listing
- ☒ See Attached Information
- ☒ Following Comments

This is nothing but the following two instructions:

```
TRA  ASIN
TRA  ATAN
```

See writeup on ASIN and ATAN

Calling sequence is:

```
for ASIN  { CLA   X              for ATAN  { CLA   X
          { TSX   AF2F,4                   { TSX   AF2F+1,4
          { TSX   ERR,4                    { TSX   ERR,4
```

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Bivariate extrapolation subroutine

Subroutine Name: BVPI

Deck Name: Not applicable

Language: ☐ Fortran IV ☒ MAP

Description:

☐ See Attached Listing

☒ See Attached Information

☒ Following Comments

The table setup of input is described in user's guide. The calling sequence to this subroutine is as follows:

```
CLA  X
LDQ  Y
TSX  BVPI,4
PZE  A
TSX  ERR,4
```

where

X = location of one argument
Y = location of second argument
A = location of table
ERR = programmer's error routine

Date 3 / 16 / 66

Page 1 of 2

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Linear interpolation subroutine

Subroutine Name: INTP

Deck Name : Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

☒ Following Comments

The calling sequence to this subroutine is as follows:

```
CLA   X
TSX   INTP,4
PZE   -TBLB1
TSX   ERR,4 (minimum error exit)
TSX   ERR,4 (maximum error exit)
```

where x = the argument

TBLB1 = location of control section of table

ERR = programmer's error routine

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Linear interpolation subroutine

Subroutine Name: INTP

Deck Name : Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing
☒ See Attached Information
☒ Following Comments

This subroutine is used directly with tables as specified in the input section of 118.0. It assumes a table control section where the minimum and maximum values of the independent variable are input to this control section of the table. The tables themselves must be input as follows. Note the first independent and dependent variables must be repeated.

Control Table	TBLT1	PZE	BITBL
		DEC	0,0, (minimum and maximum)
		PZE	0,0,-BITBL
		DEC	0,0,0 (extra storage for INTP)
Actual Table	BITBL	BSS	30

Example of actual table:

$(BSS\ 30 = I_1, D_1, I_1, D_1, I_2, D_2 \dots I_N, D_N)$

I = independent variable where I_1 = minimum independent variable)
D = dependent variable I_N = maximum independent variable)

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Integration subroutine (ordinary differential eqs.)

Subroutine Name: INTG

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

□ See Attached Listing

☒ See Attached Information

☒ Following Comments

This subroutine is used for all integrations using Gill and Adams technique. There are two types of cal ling sequences possible but I will only discuss the one used in this program. The calling sequence is as follows:

```
PXD  , 0,0
TSX    INTG,4
PZE    A,,-B
PZE    N,,-C
PZE    -B,0,-D
```

Clearing the AC by execution of PXD 0,0 is a must for this call.

A = location of subroutine to evaluate derivatives.
B = location of time (present time)
N = number of equations to be integrated
C = location of integration step size (delta T)
D = location containing value of time to be integrated to.

AC ELECTRONICS DIVISION GENERAL MOTORS CORPORATION

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Conversion from radians to degrees, minutes, and seconds.

Subroutine Name: RADEG

Deck Name : Not applicable

Language: □ Fortran IV ☒ MAP

Description:

□ See Attached Listing

☒ See Attached Information

☒ Following Comments

Converts an angle in radians to degrees, minutes, and seconds. The calling sequence to this subroutine is as follows:

```
TSX   RADEG,4
PZE   N
PZE   -A1,,-B1
 .
 .
 .
PZE   -AN,,-BN
```

where N specifies the number of conversions

A_1 = location of first angle in radians

B_1 = location of first angle in degrees, minutes, and seconds. B_1 is first location of 3 cells.

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Conversion from degrees, minutes, and seconds to radians.

Subroutine Name:DEGRA

Deck Name: Not applicable

Language: ☐ Fortran IV ☒ MAP

Description:

☐ See Attached Listing

☒ See Attached Information

☒ Following Comments

Converts an angle in degrees, minutes, and seconds. to radians. The calling sequence to this subroutine is as follows:

TSX DEGRA,4
PZE N
PZE $-A_1,,-B_1$
⋮
PZE $-A_N,,-B_N$

where N specifies the number of conversions.

A_1 = location of first of 3 cells containing angle in degrees, minutes, and seconds.

B_1 = location of 1 cell containing result in radians.

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Transformation of 3-dimensional vector

Subroutine Name: TRANSF

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

□ See Attached Listing

☒ See Attached Information

☒ Following Comments

Transforms a 3-dimensional vector. The calling sequence to this program (subroutine) is as follows:

TSX TRANSF,4
PZE -M
PZE $-A_1,,-B_1$
PZE $-A_2,,-B_2$
PZE $-A_3,,-B_3$

where

M = first location of a 3 x 3 matrix
A_1 = first component of vector to be transformed (location)
A_2 = second component of vector to be transformed (location)
A_3 = third component of vector to be transformed (location)
B_1 = first location of transformed component
B_2 = second location of transformed component
B_3 = third location of transformed component

Date 3 / 16 / 66
Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Dot product subroutine of 2 N dimensional vectors

Subroutine Name: DØT

Deck Name: Not applicable

Language: ☐ Fortran IV ☒ MAP

Description:

☐ See Attached Listing
☒ See Attached Information
☒ Following Comments

Finds the dot product of 2 N-dimensional vectors. The calling sequence to the subroutine is as follows:

```
TSX   DØT,4
PZE   ANSWER,,N
PZE   -A,,-B
```

where

ANSWER = location where product is to be stored
A = location of first N dimensional vector
B = location of second N dimensional vector
N = dimension of vector

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Cross product of 2 3-dimensional vectors

Subroutine Name: CRØSS

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

□ See Attached Listing

☒ See Attached Information

☒ Following Comments

Finds the cross product to two 3-dimensional vectors. The calling sequence to this subroutine is as follows:

```
TSX   CRØSS,4
PZE   -A,,-B
PZE   -C
```

where

A = location of first 3-dimensional vector

B = location of second 3-dimensional vector

C = location of answer for 3-dimensional product

It is assumed that A, B, and C are the first location component addresses.

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: 6 Rotational matrix evaluator

Subroutine Name: SIXRØT

Deck Name: Not applicable

Language: ☐ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

☒ Following Comments

The calling sequence to this subroutine is as follows:

```
TSX   SIXRØT,4
PZE   A,,B
PZE   C
TSX   SYSERR,4
```

where

A = first of 6 locations containing rotational axis codes (2) whose values may be 0., 1., 2., or 3. Subroutine terminates upon encountering first 2 whose value is zero.

B = first of 18 locations containing rotational angles in degrees, minutes, and seconds corresponding to rotational axis codes.

C = first of 9 locations containing resultant matrix

SYSERR = error return location

GUIDANCE BLOCK 0
(Initial Conditions)

Auxiliary Coordinate Transformation Matrix from Earth Centered Inertial Frame to Computational Inertial Frame

Output: $\hat{Q} = \hat{B}\,\hat{A}$

Input: $\hat{A}$, $T_R = \alpha_R$, $C_R = \gamma_R$ $\quad R = 1, 2, 3, 4, 5, 6$

Using the six rotational matrix evaluate [Q]

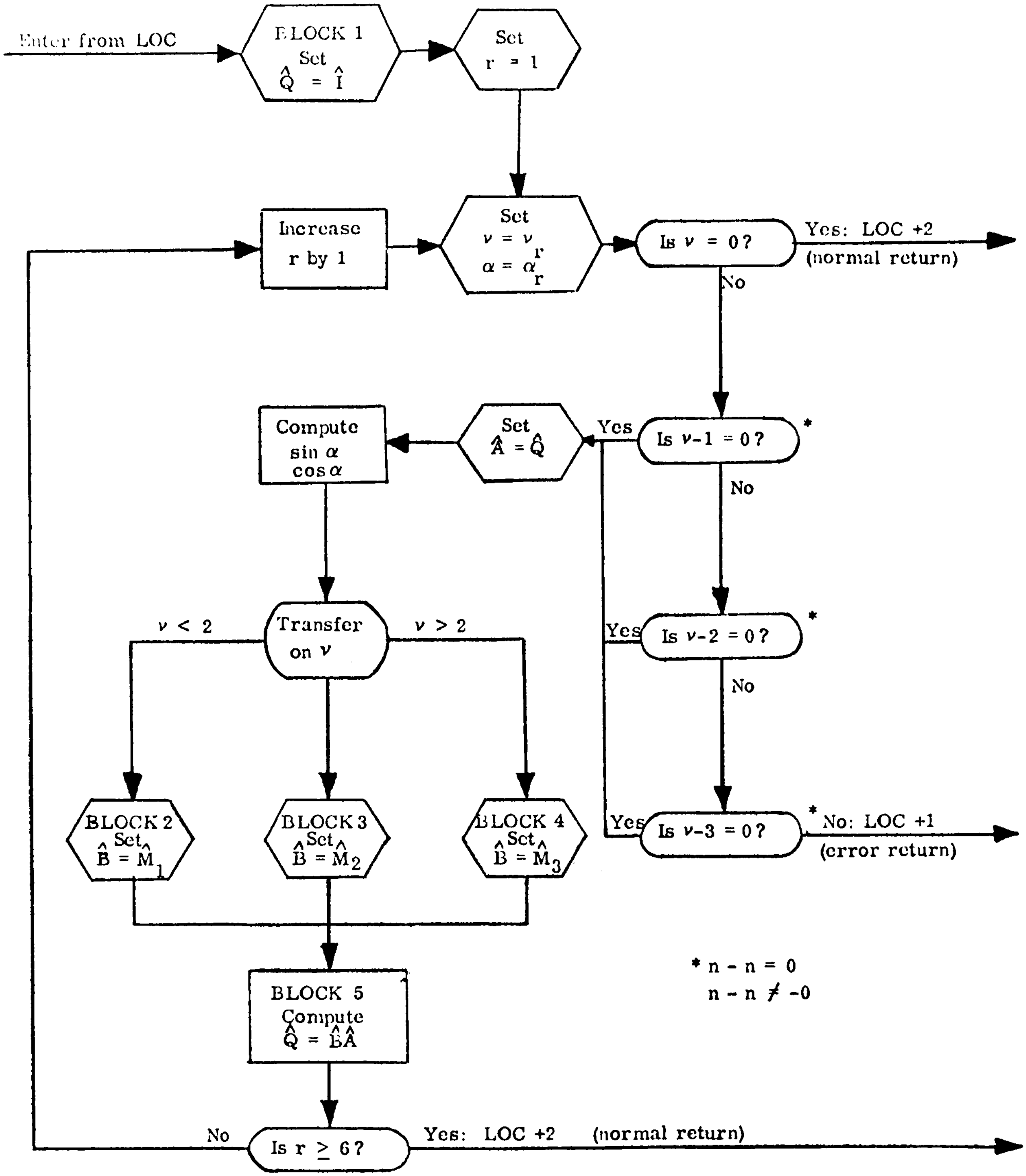

Enter from LOC
BLOCK 1 Set $\hat{Q} = \hat{I}$
Set $r = 1$
Increase r by 1
Set $\nu = \nu_r$ $\alpha = \alpha_r$
Is $\nu = 0$?
Yes: LOC +2 (normal return)
No
Compute sin α cos α
Set $\hat{A} = \hat{Q}$
Yes
Is $\nu-1 = 0$? *
No
Transfer on ν
$\nu < 2$
$\nu > 2$
Yes
Is $\nu-2 = 0$? *
No
BLOCK 2 Set $\hat{B} = M_1$
BLOCK 3 Set $\hat{B} = M_2$
BLOCK 4 Set $\hat{B} = M_3$
Yes
Is $\nu-3 = 0$? *
No: LOC +1 (error return)
* $n - n = 0$
$n - n \neq -0$
BLOCK 5 Compute $\hat{Q} = \hat{B}\hat{A}$
No
Is $r \geq 6$?
Yes: LOC +2 (normal return)

Figure 1. Logic Flow Diagram of 6 Rotation Matrix Evaluator.

SIX ROTATIONAL MATRIX EVALUATOR (3 x 3)

Output: $\hat{Q}$, q_{ij} = elements of matrix $\hat{Q}$

Input: γ_R, α_R $R = 1, 2, 3, 4, 5, 6$

$\gamma = 0, 1, 2, 3$

BLOCK 1. Set $\hat{Q} = I$

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

BLOCK 2. Set $\hat{B} = \hat{M}_1$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

BLOCK 3. Set $\hat{B} = \hat{M}_2$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

BLOCK 4. Set $\hat{B} = \hat{M}_3$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

SIX ROTATIONAL MATRIX EVALUATOR (contd) (3 x 3)

BLOCK 5. Compute $\hat{Q} = \hat{B}\hat{A}$

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

where $q_{ij} = \sum_{K=1}^{K=3} (b_{iK})(a_{Kj})$

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Input subroutine

Subroutine Name:INP9

Deck Name: Not applicable

Language: ☐ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

☐ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

INP9
INP9OP

Author Richard Ameen

Identification

INP9 - Decimal, Octal, BCI Loader using IOOP2 level of IOCS
The RELMOD deck has ENTRY INP9, ENTRY INP9OP, EXTERN COMMON, EXTERN PRT9, and EXTERN PRT9OP

Purpose

1) To read S.SIN1 with redundancy checking.

2) To convert to binary, store in core, and return control to the calling sequence or to a cell specified in the imput record (TRA).

Restrictions

1) This subroutine uses PRT9 to print an error message if error occurred in reading S.SIN1.

2) "OPEN" calling sequence must be performed before routine will read in data.

Errors

1) When using "Continuous Block Loading" or "Discontinuous Block Loading" option if "TO" address (B) is not greater than "FROM" address (A) routine will exit to the 3rd parameter word in calling sequence with 1B35 in AC and 0 in MQ.

2) If program cannot recognize pseudo operations in columns 8 through 10, routine will exit to 3rd parameter word in calling sequence with 3B35 in AC and 0 in MQ.

3) If decimal data is out of range, routine will exit to the 4th parameter word in calling sequence with 4B35 in AC and 0 in MQ.

4) If data is attempted to be loaded into octal location 10005_8 or less, routine will exit to 3rd parameter word in calling sequence with 14_8B35 in AC and 0 in MQ.

5) If error occurred in reading S.SIN1, error message is printed on S.SOU1 and run is terminated.

6) If INP9 has not been OPENED by "OPEN" calling sequence, error message is printed and run is terminated.

Method

1) Decimal numbers are converted to binary integers and then scaled by the indicated power of ten.

2) Octal numbers are converted to binary integers.

3) BCI information are stored directly.

Space Requirements

1) 652 cells + 30 COMMON

Accuracy

1) Decimal to binary floating point; 8 decimal digits (unrounded).

2) Decimal to binary fixed point; 10 decimal digits (unrounded).

3) Decimal integer to binary integer; exact.

4) Octal integer to binary integer; exact.

Range

1) Decimal to binary, floating or fixed; 0 to 10^{-38} and 10^{38}

2) Decimal integer to binary integer; 0 to $2^{35}-1$

3) Octal integer to binary integer; 0 to $2^{35}-1$

Usage

Format:

The following options are available for storage of the input data:

1) Absolute Decimal Origins. The first data word of the input record is stored in the cell specified by the calling sequence or by the absolute decimal location specified in columns 1-6 of the input record. Once loading is initiated, words are stored relative to the origin until columns 1-6 designate a new origin.

2) Absolute Octal Origins. A "B" in column 1 specifies an absolute octal origin. The origin may be assigned anywhere within the location field (columns 2-6) since blanks are ignored. Successive words will be loaded relative to this location until a new origin is designated on the input record.

3) Relocatable Data ("R"card). An "R" in column 1 specifies a relocatable storage location. The location in octal may be assigned anywhere within the location field (columns 2-5) since blanks are ignored. The location given will be adjusted by the address of the first parameter word of the calling sequence. Successive words will be loaded relative to this location until a new origin is designated on the input record.

The general character of the data to be loaded is determined by a three-letter pseudo-operation punched in columns 8-10. These pseudo-operations are: DEC, OCT, and BCI. In addition, there is an operation TRA which provides an exit from the loading program.

Decimal Data: DEC

The decimal data may begin in any of the columns, 12-16. This column is considered the start of the variable field. All columns (11-16) prior to the start of the variable field must contain blanks. The variable field terminates in column 72. Decimal data is converted to binary and assigned to consecutive locations L, L+1,.....

Successive words of data on a card are separated by commas, and the first blank occurring in the variable field indicates that all punching to the right of this blank is non-relevant.

Signs are indicated by + or - (12 or 11 punch) preceding the number, the exponent, or the binary scale factor. However, it is not necessary to use the + sign.

If none of the characters . E or B appear in a decimal data word, the word is converted as a binary integer with the binary point at the right-hand end of the word.

If either of the characters E or . or both appear in a decimal data word and the character B does not appear, the word is converted to a 7040 type floating binary quantity. The decimal exponent used in this conversion is the number which follows immediately after the character E. If the character E does not appear, the exponent is assumed to be zero. If the decimal point does not appear, it is assumed to be ar the right-hand end. For example, 12.345, +12.345, 1.2345E1, 1234.5E-2 and 12345E-3 are all equivalent representations of the same floating point quantity.

If the character B appears in a decimal data word, the word is converted as a fixed-point binary quantity. The binary scale factor used in this conversion is the number which follows immediately after the character B; it being the number of binary places between the left-hand end of the storage cell and the binary point of the fixed-point binary result. If the decimal point does not appear in the decimal word, it is assumed to be at the right-hand end. The decimal exponent used in this conversion is the number which follows immediately after the character E. The order of B and E is not significant. For example, 12.345B4, +1.2345E1B4, and 12345B4E-3 are all equivalent representations of the same fixed-point quantity.

Octal Data: OCT

The octal data may begin in any of the columns, 12 through 16. This column is considered the start of the variable field. All columns (11-16) prior to the start of the variable field must contain blanks. The variable field terminates in column 72.

The binary point is considered to be on the right-hand end of a 7040 word, and assigned to consecutive storage locations L, L+1,...

Successive words are separated by commas and the first blank occurring in the variable field indicates that all punching to the right is to be considered non-relevant.

In the case of 12-digit octal numbers, the following equivalences exist with respect to the high order digit:

-0 ≡ 4 -1 ≡ 5 -2 ≡ 6 -3 ≡ 7

Either form may be used in OCT cards.

Binary Coded Information: BCI

If any of the columns 12 through 16 contain a comma with no preceding word count, 10 six-character words of Hollerith information from the next 60 columns (immediately following the comma) are read and assigned to locations L, L+1, ... , L+9. Blanks will be stored, however, for columns 73 through 76.

If less than 10 BCI words are desired, a word count v $(0 < v \leq 9)$ is punched in any of the columns, 12 through 16. The next column must contain a comma and is followed by the v words of BCI information. All columns from 11 to the column containing the word count must contain blanks.

Transfer: TRA

When a card punched TRA in columns 8-10 is encountered, control will be transferred to the location specified by the address and tag punched, starting with any of the columns 12 through 16. This column is considered the start of the variable field. All columns (11-16) prior to the start of the variable field must contain blanks.

Two options are available for exit from INP9.

1) Absolute decimal with no tag, e.g., TRA 4096 will cause the loading program to transfer control to location $(10000)_8$.

2) Absolute decimal address with tag, e.g., TRA 4,4 will cause the loader to transfer control to the fourth instruction following the TSX INP9,4 entry.

Calling Sequence

1) For opening routine

```
TSX     INP9ØP,4
PZE     K,0,L
```

where K is lowest location in core memory that data can be read into (i.e., built-in memory protection).

Where if L = 0 all input data will be printed via PRT9 as soon as read. If L ≠ 0 no input data will be printed.

2) Continuous block loading from A to B, locations not punched on cards:

```
F        TSX       INP9,4
F+1      PZE       A,Ø,B
F+2                ERROR RETURN
F+3                ERROR RETURN
F+4                NORMAL RETURN
```

Words are stored consecutively as follows: A, A+1, A+2,.... B.

3) Discontinuous block loading, location punched on some or all cards:

```
F        TSX       INP9,4
F+1      PZE       A,Ø,A+N-1
F+2                ERROR RETURN
F+3                ERROR RETURN
F+4                NORMAL RETURN
```

Where N is the number of words to be loaded. If there it not location punched on the first card, the first word will be loaded at A. In the event that all necessary locations are specified on cards, set A = 1.

4) Load until transfer, starting at location A.

```
F        TSX       INP9,4
F+1      PZE       A
F+2                ERROR RETURN
F+3                ERROR RETURN
F+4                NORMAL RETURN
```

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Output subroutine

Subroutine Name: PRT9

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

- ☒ See Attached Listing
- ☒ See Attached Information
- □ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

PRT9
PRT9ØP
PRT9CL
PRT9ØV
PRT9CT

Author Richard Ameen

Identification

PRT9 - General Output Routine for 7040 Using IOOP2 Level of IOCS
The RELMOD deck has entries PRT9ØP, PRT9CL, PRT9ØV, PRT9CT, and PRT9

Purpose

To set up one line of output as specified in the calling sequence and to output this line on the on-line printer or the on-line punch. This routine is also applicable for use with an off-line operation. The following conversions are available: Hollerith to Hollerith, Binary Integer to Decimal Integer, Fixed Binary to Fixed Decimal, Floating Binary to Floating Decimal, Floating Binary to Fixed Decimal, and Binary to Octal.

*Note: The number of lines per page can be controlled by the user.

Restrictions

1) PTW must be followed by PZE.

2) Binary scaling factor, B, must be within the range 0 - 35 inclusive.

3) PW (Print Wheel) position must be 1 < PW < 138.

4) Opening calling sequence must be performed to have printing or punching output.

5) If printing failure occurs, program will print message on typewriter to operator and stop. Operator simply pushes start button to continue.

6) If punching failure occurs, program will print message on typewriter to operator and stop. Program will attempt to re-punch same record two more times if operator pushes start button. If failure still occurs, program will continue.

Method

The line image is written via IOOP2 which must be available for use during the execution of the program.

Space Requirements

628 cells + 0 COMMON.

Usage

Calling sequences:

1) Opening Calling Sequence

```
TSX   PRT9ØP,4
PZE   A,,B
      RETURN
```

where A = the maximum number of lines per page.

If A = 0, the value 60 will be used.

B = exit location for page overflow. Whenever the number of lines per page exceeds A, a TRA B is performed. If B = 0, no exit is performed.

The exit routine, (B ≠ 0), may be used for footings and headings. If footings are desired, the exit routine must place a fixed 1 B35 into PRT9CT and then print desired footings, being careful not to overflow the page. The last calling sequence in the exit routine when using footings must restore page using PØN 3. This last calling sequence most likely will be used as a heading for the next page.

If the exit routine is used only for headings, the user does not have to use a PØN 3 as PRT9 will automatically do this. More than one heading can be printed by the exit routine. In all cases, the very last instruction of the exit routine must be a TRA PRT9ØV.

*Note: If printing or punching is desired, the user must perform the Opening Calling Sequence.

2) Close Calling Sequence

```
TSX   PRT9CL,4
      RETURN
```

This entry will prevent any future printing or punching until an Opening Calling Sequence is performed.

3) Normal Calling Sequence (printing and punching)

```
α           TSX        PRT9,4
α + 1       YYY        A, T, D
α + 2       YYY        A, T, D
  .          .         ., ., .
  .          .         ., ., .
  .          .         ., ., .
α + n - 1   YYY        A, T, D
α + n       ZZZ        CC
α + n + 1   NORMAL     RETURN
```

Explanation of Calling Sequence

Each of the operations denoted by YYY describes the type of conversion to be done for one entry of the line to be output. The address, tag and decrement contain information necessary for the conversion and the spacing of the number in the line.

Any combination of any number of these pseudo-operations may be used to set up a line of output and when followed by one of the ZZZ operations, the line is output on-line or written on tape to be output off-line. For each instruction the tag, T, may be 0, 1 or 2.

YYY Pseudo-Operations

PTH L, T, 1000·N+PW

Hollerith to Hollerith Conversion

PTH prints N BCD words where the first word is located in L-C(T). The last (6th) character of the Nth word is printed by print wheel PW.

Up to 22 BCD words may be printed with one PTH operation, (e.g., PTH L, 0, 22132 will print the contents of L through L+21 using print wheels 1-132).

FØR L, T, PW

Binary Integer to Decimal Integer Conversion

FØR prints the contents of L-C(T) as a decimal integer with the units position printed by print wheel PW. The number in L-C(T) is considered to be at B = 35.

PTW L, T, 1000·D+PW

PZE B

Fixed Binary to Fixed Decimal Conversion

The PTW, PZE combination prints the contents of L-C(T) as a fixed point number rounded to D places after the decimal point. D must be ≤ 8. The last digit of the number is printed by print wheel PW.

B is the number of binary places to the left of the binary point of the number in L-C(T). B must be in the $0 \le B \le 35$.

The sign of the number, if minus, prints in the first position to the left of the most significant digit. If positive, a blank is inserted in this position.

If L-C(T) contains zero and $D = 0$, nothing will print.

PTW and PZE must always be used in the combination shown above.

SIX L, T, 1000·D+PW

Floating Binary to Floating Decimal Conversion

SIX prints the contents of L-C(T) as a floating point decimal number with a mantissa rounded to D places ($D \le 8$) and an exponent of two digits printed to the right of the mantissa. The second position of the exponent is printed by print wheel PW.

Example: $-.123456789 \times 10^{-3}$ printed by SIX L, T, 8060 will print as:

-. 12345679-03 ← (Print Wheel 60)

If L-C(T) contains zero, the number printed is .00000-39 (if $D = 5$).

SVN L, T, 1000·D+PW

Floating Binary to Fixed Decimal Conversion

SVN prints the contents of L-C(T) as a fixed point decimal rounded to D decimal places. ($D \le 8$). If D equals zero, a rounded integer will be printed without a decimal point. If the number is negative, a minus sign will be printed to the left of the left-most characters. No leading zeros to the left of the decimal point will print. If the number is zero, it will print D zeros to the right of the point. If the absolute value of the number exceeds 34,359,738,367, it will be printed in floating decimal form as described under SIX.

Example: -.123456789 x 10^{-3} printed by SVN L,T, 8060 will print as:

-.000123456 ↖———————— (Print Wheel 60)

PZE L,T,0

Transfer

PZE (with decrement equal to zero, and not preceded by PTW) effects a change in the calling sequence location by specifying that the next word in the calling sequence is to be taken from L-C(T). The calling sequence then continues from L-C(T) until a PØN, FVE or another transfer (PZE) is encountered.

PZE L,T,1000·D+PW

Binary to Octal Integer

(PZE not preceded by PTW). D must be ≤ 12. The sign of the number is considered to be part of the number. Three binary bits produce one octal integer.

PZE 0,0,0

Ignore, continue with next parameter.
(PZE not preceded by PTW)

The above pseudo-operations are used in the order given in the calling sequence. Therefore, any overlapping of numbers will erase the overlapped portion of a previously converted number and substitute the numbers converted by the later pseudo-operation.

Spacing of a line should be done in such a way that print position one (1) is not used since it will be lost when writing the output on tape. Print position 1 is used as a carriage control character when writing on S.SOU1.

The operation denoted by ZZZ in the sample calling sequence must be a PØN. It must be the last operation of the calling sequence; it will cause the line set up by the previously described pseudo-operations to be output as described below.

FVE CC (Same as PØN CC).

PØN CC

Print or Write on Tape

PØN causes the line already set up by previous pseudo-operations to be output. CC is a carriage control code which controls spacing on-line and off-line as follows:

CC = 0, Suppress spacing
1, Single space before printing
2, Double space before printing
3, Restore page before printing
-4,5,6, Punch card in 1402 4 pocket

*NOTE: Cards on which punching error occurred will be selected into 1402 NP pocket

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Fixing of floating point numbers

Subroutine Name: PFIX

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn PFIX

Identification

PFIX - General fixed point check and convert routine.
The RELMOD deck has ENTRY PFIX

Method

Each number in a consecutive list is checked against 7 B 12; if it is greater or equal to this number, then it is assumed to be a floating point number. The routine will then replace it with the appropriate fixed point integer scaled at B35; otherwise no change takes place.

Usage

Calling Sequence:

```
δ          TSX PFIX , 4
δ + 1      YYY  A , 0 , X
  .
  .
  .
δ + n      YYY  A , 0 , X
δ + n + 1  ZZZ            (normal return)
```

Explanation of Calling Sequence

Any combination of pseudo-operations PZE and PTH may be used.

YYY Pseudo-operations

PZE A , 0 , X
check X cells beginning at location A
$X \leq 1023_{10}$

PTH A , 0 , X
X is the location of a fixed point number (N) scaled at B35. In this case N consecutive cells are checked beginning at location A.

ZZZ operation
Any operation may terminate the parameter list except TXH with a number in the decrement.

Space Requirements

40 cells

Miscellaneous

This routine is compatible with the 7090/7094.

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Floating of fix point numbers

Subroutine Name: PFLØAT

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn PFLOAT

Identification

PFLØAT - General floating point check and convert routine.
The RELMØD deck has ENTRY PFLØAT.

Method

Each number in a consecutive list is checked against 7 B 12; if it is less than this number, then it is assumed to be a fixed point number scaled at B35. The routine will then replace it with the appropriate floating point number; otherwise no change takes place.

Usage

Calling Sequence:

```
δ          TSX  PFLØAT , 4
δ + 1      YYY  A , 0 , X
 .
 .
 .
δ + n      YYY  A , 0 , X
δ + n + 1  ZZZ                (normal return)
```

Explanation of Calling Sequence

Any combination of pseudo-operations PZE and PTH may be used.

YYY Pseudo-operations

PZE A , 0 , X
checks X cells beginning at location A.
$X \leq 1023_{10}$

PTH A , 0 , X
X is the location of a fixed point number (N) scaled at B35. In this case 2N consecutive cells are checked beginning at location A.

ZZZ operation
Any operation may terminate the parameter list except <u>TXH</u> with a number in the decrement.

Space Requirements

40 cells

Miscellaneous

This routine is compatible with the 7090/7094.

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Square root subroutine

Subroutine Name: SQRT

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn SQRT

Identification

SQRT - Floating Point 7040 Square Root Subroutine
The RELMOD deck has ENTRY SQRT and EXTERN COMMON.

Purpose

To compute the square root of a normalized floating point number.

Restrictions

The argument must be positive or zero.

Method

Let $x = 2^I . f \quad 1/2 \leq f < 1$, then a trial root is computed according to

$$y_o = \begin{cases} 2^{I/2}\left(\frac{9}{16} f + \frac{7}{16}\right) & \text{if I is even} \\ 2^{(I+1)/2}\left(\frac{7}{16} f + \frac{9}{32}\right) & \text{if I is odd} \end{cases}$$

Since this linear approximation yields 7 significant bits, two Newton-Ralphson iterations suffice to produce $\sqrt{x}$.

Accuracy

26 significant bits.

Usage

Calling sequence: With x in the AC

```
TSX SQRT , 4
ERRØR RETURN      with x in the AC, x < 0
NØRMAL RETURN     with √x in the AC
```

Space Requirements

38 + 2 CØMMØN

Miscellaneous

This routine is compatible with the 7090/7094.

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Cosine subroutine

Subroutine Name: CØS

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Sine subroutine

Subroutine Name: SIN

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn

SIN
COS

Identification

SIN-COS Floating-Point 7040 Sine-Cosine Subroutine
The RELMOD deck has ENTRY SIN, ENTRY COS, and EXTERN COMMON

Purpose

To compute floating sin x or cos x when floating x is given.

Restrictions

None

Method

Sin: The argument is reduced (if necessary) to a first quadrant equivalent to which a polynomial approximation is then applied (See RAND sheet 16).

Cos: $\cos x = \sin(\pi/2 + x)$

Accuracy

The statements below apply to the sine function, (ϵ_A = absolute error).

$0 \le \|x\| \le \pi/2$	$\epsilon_A \le 2^{-27}$	The polynomial approximation yields at least 26 significant bits.
$\pi/2 < \|x\| < 2^{27}$	$\epsilon_A < 2/\pi\ \|x\|\ 2^{-27}$	The argument loses significance as x becomes large, and also as x approaches zero of sin x.
$2^{27} \le \|x\|$		The argument, and therefore, the answer has no significance. The answer will always be zero.

Usage

Calling Sequence: With x in the AC

For sin x	For cos x
TSX SIN , 4	TSX CØS , 4
NORMAL RETURN	NORMAL RETURN
sin x in the AC	cos x in the AC

Space Requirements

46 + 2 CØMMØN

Miscellaneous

This routine is compatible with the 7090/7094.

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Arctangent subroutine

Subroutine Name: ATAN

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Arcsine subroutine

Subroutine Name: ASIN

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn

ATAN
ASIN

Identification

ARCFUN - Floating Point Arcfunction Subroutine for the 7040
The RELMOD deck has ENTRY ATAN, ENTRY ASIN, and EXTERN COMMON

Purpose

To compute the arcsin and arccos (or arctan and arccot) of a normalized floating point number

Restrictions

For arcsin and arccos, the argument X must satisfy $|X| \leq 1$

Method

$$\text{Arcsin } X = \arctan\left(\frac{X}{\sqrt{1 - X^2}}\right)$$

$$\text{Arccos } X = \pi/2 - \text{arcsin } x$$

Arctan x obtained from a table loop-up and series evaluation

$$\text{Arccot } x = \pi/2 - \text{arctan } x$$

Accuracy

Arcsin and Arccos: 25 significant bits except for $|X|$ near 1, where $27 - [\log_2 \sqrt{1 - x^2}\,]$ are significant

Arctan and Arccot: at least 26 significant bits

Usage

With x in the AC

for Arcsin and Arccos
TSX ASIN,4
ERROR RETURN $|x|$ in AC
Normal return
$-\pi/2 \leq$ arcsin x in AC $< \pi/2$
$0 \leq$ arccos x in MQ $< \pi$

for Arctan and Arccot
TSX ATAN,4
Unrecognized cell
Normal return
$-\pi/2 \leq$ arctan x in AC $< \pi/2$
$0 \leq$ arccot x in MQ $< \pi$

Space Requirements

125 + 6 COMMON

Miscellaneous

This routine is compatable with the 7090/7094

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Natural logarithm subroutine

Subroutine Name: LN

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

- ☒ See Attached Listing
- ☒ See Attached Information
- □ Following Comments

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author T. C. Littlejohn LN

Identification

LN - Floating Point 7040 Natural Logarithm Subroutine.
The RELMOD deck has ENTRY LN and EXTERN COMMON

Purpose

To compute the natural logarithm of a normalized floating point number.

Restrictions

The argument must be positive. For $x = +0$, the divide check indicator and light are turned on, and the answer $\ell n\, x = -89.069412$ is provided.

Method

Let $x = 2^I \cdot f = 2^I \cdot 2^{\log_2 f}$
Then $\ell n\, x = (I + \log_2 f)\ell n\, 2$
$\log_2 f$ is obtained from the polynomial approximation given on RAND sheet number 42.

Accuracy

$|\ell n\, x| < 1$: $|\text{absolute error}| \leq 1.06 \times 2^{-27} = .790 \times 10^{-8}$
$|\ell n\, x| \geq 1$: $|\text{relative error}| \leq 1.06 \times 2^{-27} = .790 \times 10^{-8}$

Usage

Calling Sequence: With x in the AC

```
TSX     LN , 4
ERRØR RETURN      with x in the AC, x < 0
NØRMAL RETURN     with ln x in the AC
```

Space Requirements

45 + 3 CØMMØN

Miscellaneous

This routine is compatible with the 7090/7094.

Date 3 / 16 / 66
Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Exponential subroutine

Subroutine Name: EXP

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing
☒ See Attached Information
□ Following Comments

EXP

Author T. C. Littlejohn

Identification

EXP - 7040 Floating Point Exponential Subroutine
The RELMOD deck has ENTRY EXP and EXTERN COMMON

Purpose

To compute the exponential (base e) of a normalized floating point number

Restrictions

$|X| \leq 88.028$
$X > 88.028$ Error return with x in the AC
$X < -88.028$ Normal return with $e^x = 0$

Method

Rational approximation based on a paper by E. G. Kogbetliantz (IBM Journal of Research and Development, April 1957)

$$e^x = 2^{x/\ell n2} = 2^I . 2^f \qquad 0 \leq f < 1$$

$$2^f = 1 + 2f\,[a - f + bf^2 - c(f^2 + d^2)^{-1}]^{-1}$$

where

$a = 9.9545957821$	$c = 617.97226953$
$b = 0.03465735903$	$d = 87.417497202$

Accuracy

At least 26 - j significant bits, where j is the number of bits in the integral part of the argument x

Usage

Calling Sequence: With x in the AC

```
TSX    EXP,4
ERROR RETURN
NORMAL RETURN with e^x in the AC
```

Space Requirements

51 + 4 COMMON

Miscellaneous

This subroutine is compatable with the 7090/7094.

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Floating point and octal dump subroutine

Subroutine Name: CØRE

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Display of panel (registers of computers)

Subroutine Name: PANEL

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Generalized setup subroutine

Subroutine Name: SETUP

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

□ Following Comments

Sets up input routine (INP9) and output routine (PRT9) in addition to Error pointers.

AC SPARK PLUG PROGRAMMING MANUAL FOR THE 7040

Author W. Stone

SETUP
CORE
PANEL

Identification (SECOPA)

SETUP - Subroutine for opening PRT9 and INP9 subroutines and to perform other program initializing functions as specified below.

CORE - Floating point decimal and octal dump subroutine.

PANEL - To dump the 7040 panel in floating point decimal and octal as specified below.

The RELMOD deck has ENTRY SETUP, ENTRY CORE, ENTRY PANEL, EXTERN PRT9, EXTERN PRT9OP, and EXTERN INP9OP

Purpose

A) SETUP (To be used at beginning of program)

1) To open INP9 with the print option
2) To open PRT9
3) To initialize the floating point under-over flow routine which set underflows equal to zero and overflow causes a message to be printed which tells where the overflow occurred and TRA S.SERR
4) To setup S.SERR using the parameter word A
5) To inform the operator via the typewriter the location he is to TSX to on index 4 in case of operator intervention. This location is a link to S.SERR
6) To turn off the divide check indicator light
7) To eject a page on S.SOU1.

B) CORE

1) The 8 numbers per line are output on S.SOU1 via PRT9 which uses IOOP2. The "beginning" address (K) and the "ending" address (L) are changed to be equal to the lower and higher address respectively whose units position is a zero. The address of every 8 words is printed at the very left side of the page. When using CORE, a PANEL is automatically given before the CORE dump. If duplicate information exist consecutively, a single space break will indicate this in the CORE dump.

C) PANEL

1) Panel can be used divorced of CORE. The AC and MQ registers are output on S.SOU1 in floating point decimal and octal while the index registers are output in octal only.

In both the CORE and PANEL subroutines, the location counter is output which specified from where the CORE or PANEL was asked. All registers are saved and restored to original status when using CORE and PANEL.

Space Requirements

340 cells

Usage

A) SETUP Calling Sequence

```
TSX    SETUP,4
PZE    A , 0 , B
PZE    C , 0 , D
(NORMAL RETURN)
```

where A = location of error routine of program which is placed in S.SERR

B = lowest location into which data can be read via INP9. If B = 0, the lowest location will be 10005_8

C = number of lines per page as specified in PRT9. If C = 0, 60 lines per page will be used.

D = exit location for page overflow used in PRT9. Whenever the number of lines per page exceeds C, a TRA D is performed. If D = 0, no exit is performed.

B) CORE Calling Sequence

```
STL    CORE
TRA    CORE + 1
PZE    K, T, L
(NORMAL RETURN)
```

where K = beginning address

L = ending address

$$T = \begin{cases} 0 \text{ (octal dump)} \\ 6 \text{ (floating point decimal dump)} \end{cases}$$

*NOTE: When asking for a floating point dump (T = 6) if the number is not in normalized floating point form, that number will be dumped in octal.

C) PANEL Calling Sequence

```
STL  PANEL
TRA  PANEL+1
(NORMAL RETURN)
```

Date 3 /16 /66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Write tape (time point history) in Fortran IV and prints tape on completion of run

Subroutine Name: TAPEW

Deck Name XTAPEW

Language: ☒ Fortran IV ☐ MAP

Description:

☐ See Attached Listing

☐ See Attached Information

☒ Following Comments

Fortran IV program listing is self-explanatory. Even though this program is written in Fortran IV, it is called from the map language Boost program.

Date 3 / 16 / 66

Page 1 of 1

AC ELECTRONICS

IBM 7094 UTILITY SUBROUTINE INFORMATION

Program No.: 117.0 and 118.0

Function: Atmospheric subroutine

Subroutine Name: EATM

Deck Name: Not applicable

Language: □ Fortran IV ☒ MAP

Description:

☒ See Attached Listing

☒ See Attached Information

☒ Following Comments

The calling sequence to this subroutine is as follows:

```
TSX   EATM,4
```

D.3 PROGRAM LISTING, PROGRAM 118.0

The original of the compilation listing has been supplied with the program decks.

APPENDIX E

INPUT FORMS FOR PROGRAM 118.0

The originals of the input forms for Program 118.0 have been supplied with the program decks.