https://ntrs.nasa.gov/search.jsp?R=19720021186 2020-03-11T18:19:45+00:00Z

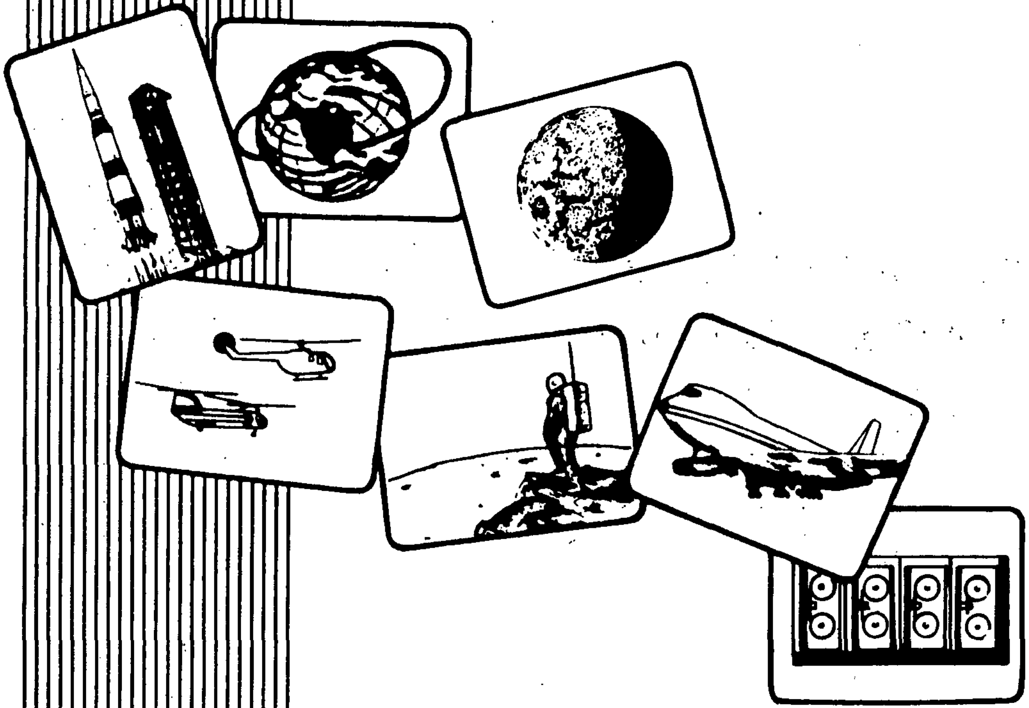
N72- 28836

CASE FILE COPY

D2-118387-1A

USERS GUIDE FOR GUIDANCE AND CONTROL LAUNCH AND ABORT SIMULATION FOR SPACECRAFT (LASS)

VOLUME I



THE BOEING COMPANY
HOUSTON, TEXAS
JUNE 30, 1972

D2-118387-1A

USERS GUIDE FOR GUIDANCE AND CONTROL LAUNCH AND ABORT SIMULATION FOR SPACECRAFT (LASS)

VOLUME I

June 30, 1972

D2-118387-1A

USERS GUIDE FOR GUIDANCE AND CONTROL LAUNCH AND ABORT SIMULATION FOR SPACECRAFT (LASS)

VOLUME I

NAS9-12183

June 30, 1972

Prepared by Howard D. Backman
for T. F. Havig

Prepared by Howard D. Backman
H. D. Backman

Approved by W G Ryals
W. G. Ryals
Program Manager

REVISIONS

REV. SYM	DESCRIPTION	DATE	APPROVED
A	Added Orbiter Aerodynamics, Post Burnout Control, Conventional Autopilot, and Program Output. Revised Abort Guidance Programs.	6-30-72	

ABSTRACT

This report constitutes a description of the math models and computer program which are used to implement LASS (Launch and Abort Simulation for Spacecraft). The computer program provides for a simulation of boost to orbit and abort capability from boost trajectories to a prescribed target. The abort target provides a decision point for engine shutdown from which the vehicle coasts to the vicinity of the selected abort recovery site.

The simulation is a six degree of freedom simulation describing a rigid body. The vehicle is influenced by forces and moments from nondistributed aerodynamics. An adaptive autopilot is provided to control vehicle attitudes during powered and unpowered flight. A conventional autopilot is provided for study of vehicle during powered flight.

At launch the vehicle follows a prescribed pitch profile until a preselected point in launch time is reached. At this time, vehicle guidance uses the MIT version of E guidance, as modified for thrust limiting, to orbit insertion. An abort may be commanded as pre-programmed, at any time during launch. A method of abort guidance is provided which describes an abort target as a function of desired recovery site latitude and longitude and desired burnout velocity and altitude. Burnout parameters are functions of reentry flight path.

The document is released in two volumes, D2-118387-1A and D2-118387-2A. Volume I contains discussion of the math models. Volume II contains supporting flow charts, symbol dictionaries, and computer program listing.

This work was performed for the MSC Guidance and Control Division under Exhibit B of Contract NAS9-12183.

LIST OF KEY WORDS

Simulation
Boost
Abort
Targeting
Guidance
Autopilot
Inertial attitudes
Body coordinates
Equations of Motion
Thrust limiting
Coordinate Transformation

TABLE OF CONTENTS
VOLUME I

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

VOLUME II

LIST OF FIGURES
VOLUME I

LIST OF FIGURES (Continued)
VOLUME II

LIST OF TABLES
VOLUME I

LIST OF TABLES (Continued)
VOLUME II

REFERENCES

1. NASA/MSC Memorandum EG2-71-30, "Publication of Aerodynamic and Mass Properties Data for the Delta/Delta Booster/Orbiter, EG/Frank M. Elam, AST, January 29, 1971.

2. "Airplane Performance Stability and Control", John Wiley & Sons, 1949, Courtland D. Perkins and Robert E. Hage.

3. "Classical Mechanics", Addison-Wesley Publishing Company, 1950, Herbert Goldstein.

4. NASA/MSC memo EG6-71-7, "Orbiter Abort Targeting and Guidance for a Powered Downrange Abort during Launch", David W. Gilbert, Feb. 22, 1971.

5. DB 2.1.5-13000-10, "Preliminary Aerodynamic Data for Space Shuttle Delta Wing Orbiter", February 19, 1971.

6. NASA/MSC Memorandum 71-FM36-42, "Delta Wing Shuttle Characteristics Being Used by the Contingency Analysis Section", FM3/Flight Analysis Branch, March 2, 1971.

7. Boeing Memorandum 5-2950-1-HOU-38, "Preliminary Description of SSV Dynamics Analysis Computer Program", February 9, 1970.

8. Digital Program BHA0030-D, "Saturn V/S-IC Flight Dynamics", July 27, 1967.

9. MIT 23A-STS Memo #5-70, "IMU Performance Analysis of STS Launch Trajectories", April 11, 1970.

10. G. W. Cherry, "E Guidance - A General Explicit Optimizing Guidance Law for Rocket-Propelled Spacecraft", August 21, 1964.

11. NASA Memorandum EX23/7004-119C, "Reference Launch Trajectory April Baseline Space Shuttle Vehicle", April 2, 1970.

12. NASA TMX-53414, "General Formulation of the Iterative Guidance Mode", March 22, 1966.

13. Fogarty, L. E. and R. M. Howe, "Analog Computer Solution of the Orbital Flight Equations".

REFERENCES (Continued)

14. Boeing Memorandum 5-2950-1-HOU-73, "Baseline Control Report - SSV Dynamics Studies", April 2, 1970.

15. NASA/MSC Memo EG6-70-93, "Preliminary Shuttle Abort Technique", David W. Gilbert, August 28, 1970.

16. "U. S. Standard Atmosphere", prepared under sponsorship of: ESSA, NASA, and USAF, 1962.

17. "U. S. Standard Atmosphere Supplements", prepared under sponsorship of: ESSA, NASA, and USAF, 1966

18. Boeing Report 5-2581-HOU-068, "Boost and Abort G&C System Requirements", April 27, 1972.

19. Boeing Report 5-2950-1-HOU-278, "Derivation of a Drift Minimum/ Loads Minimum Control Law for Booster, Atmospheric Flight", May 12, 1971.

1.0 INTRODUCTION

Associated with each nominal launch is the possibility that the launch trajectory must be aborted. In the aborted case, the spacecraft must be given the capability of leaving the nominal launch trajectory and returning to earth. This requires that abort guidance and control procedures must be studied to assure safe return of the spacecraft and its crew. The Launch and Abort Simulation for Spacecraft (LASS) as discussed in this report was developed to conduct these required abort guidance and control studies.

The basic abort simulation must be initialized from either the simulated launch trajectory or a table of launch trajectory data. A very important part of LASS is a basic simulation of boost to orbit insertion; therefore, aborts are initialized from the simulated launch trajectory generated by LASS. LASS is structured to make efficient utilization of the Aerodynamic (AERO), Equations of Motion (EOM) and Autopilot (AUTOPI or AUTOP2) subroutines regardless of whether the simulation is in launch or abort mode. The AERO, EOM, and AUTOPI or AUTOP2 subroutines are called five times faster than either launch guidance (GUIDAN) or abort targeting and guidance (ABGUID).

The initialization portion is used to compute targeting conditions used in the launch guidance and initial conditions used in the equations of motion, and to initialize variables to a start configuration. The guidance subroutine is used to provide the desired vehicle attitude information to

1.0 INTRODUCTION (Continued)

the autopilot. The EOM subroutine is used to compute vehicle translational and angular velocities and positions, and to provide present vehicle attitudes to the autopilot. The ideal or conventional autopilot is used to control vehicle attitudes when the main rocket engines are burning. When engines are not burning the ideal autopilot serves as a pseudo RCS.

In the simulation, the nominal launch is initialized, prior to launch, with the inertial platform aligned with the launch plane. At launch, the vehicle follows a prescribed pitch profile until a mission time of approximately 212 seconds is reached. After this time, the MIT version of E guidance as modified for thrust limiting is used for vehicle guidance to orbit insertion.

An abort may be commanded at any time during launch in the LASS program. When a preprogrammed abort is desired prior to nominal stage separation, the booster engines are shut off and a normal separation sequence is followed. The TARGET routine computes an abort target starting from a desired recovery site latitude and longitude in an iterative manner. The abort guidance (ABGUID) provides the desired inertial attitudes required by the autopilot and engine burn time remaining for targeting. When the engines shut down the orbiter coasts to a terminal energy state.

1.0 INTRODUCTION (Continued)

This document is divided into two volumes, D2-118387-1A and D2-118387-2A. Volume 1 contains a discussion of the engineering models used in development of LASS. Volume 2 contains flow charts, symbol dictionaries, and computer program listings that supplement the discussions contained in Volume 1. Any reference contained in Volume 1 to appendices A and B is understood to be a reference to Volume 2. Appendix A contains both flow charts of the simulation and symbol definition tables. Appendix B contains a program listing of the current LASS simulation.

2.0 OBJECTIVES

2.1 OBJECTIVE OF THE SIMULATION

The current simulation as discussed in this document was developed as a rigid body six degree of freedom simulation. The total simulation is to be constructed with a modular concept such that additional "real world" effects can be added with a minimum of effort at some later date. The ultimate objective of this simulation will be to provide a basic engineering framework that lends itself to a flexible guidance and control simulation for vehicle boost to orbit or abort during boost.

2.2 STUDY OBJECTIVES

In general, the simulation as described herein will be used to study guidance and control requirements during an abort from a nominal launch trajectory at any point (time) during launch. The objectives of the simulation studies will be to develop and test abort guidance techniques that will:

a. Steer the orbiter to an abort target coincident with fuel depletion after the abort has taken place.

b. Place the orbiter at a terminal energy state from which a landing could be made.

c. Result in an abort trajectory which can be controlled and does not violate selected guidelines.

d. Determine what launch trajectories or portions of trajectories would result in loss of the spacecraft if an abort were to take place.

3.0 SCOPE

The simulation is written in FORTRAN V for application on the NASA/MSC UNIVAC 1108 computers, The LASS program description presented in this report is essentially for use on the 1108 computer.

The simulation logic is designed such that after separation the spacecraft alone is simulated. Both the launch configuration and spacecraft alone simulations are six degree-of-freedom simulations of a rigid non-symmetrical vehicles. The simulation is arranged such that launch and abort guidance procedures can be readily changed to reflect different concepts. Basic vehicle configuration, limitations and assumptions are described in the Summary in Section 4.0.

4.0 SUMMARY

The math models were developed for a six degree-of-freedom nonsymmetrical rigid vehicle. The vehicles simulated are in two configurations. The first is the launch configuration and the second is the soacecraft only configuration.

The present LASS simulations were developed with certain limitations. It is intended that these limitations be corrected when adequate and consistent data are available. These limitations are:

1. Spacecraft center of gravity travel is limited to displacements alonq the body X direction. Displacements along body Y and Z axis are constrained to zero.
2. Provisions have not been made for wind effects on the booster at low altitude aborts.
3. Provisions have not been made for fuel slosh and vehicle bendinq.

LASS was developed with certain assumptions, which in some cases, could be classified as limitations. However, the following assumptions are not considered to materially affect the study of abort guidance and control procedures:

1. Both booster and orbiter are simulated as a rigid body with mass symmetry in the body X_B - Z_B plane.
2. The simulation assumes both vehicles contain ideal engine actuators and engines can be gimbaled without inertial effects.
3. Aborts result from a booster malfunction. Hence, orbiter systems are assumed to function in their nominal sense.

4.0 SUMMARY (Continued)

4. Assumed simple inertial attitude hold during vehicle separation is adequate to assure booster-spacecraft separation. The simulation is not oriented for detail studies of separation dynamics.

5. Assumed canned roll maneuver during launch is not required (vehicle oriented in boost plane at liftoff).

6. The simulation is started with a normal launch. Additional means of initiating an abort are not provided.

7. Assumed attitude control augmentation using auxiliary controls will not be required.

8. Assumed guidance method starting with a fixed pitch profile and followed by the MIT version of E guidance as modified for thrust limiting is adequate to provide nominal launch trajectories. The simulation was developed such that an abort can only take place at a preselected launch time during simulation of a nominal launch. This allows launch azimuth, orbit plane inclination angle and launch site to be variables.

9. Assumed provisions for thrust build-up and tail-off are not required.

10. Assumed thrust forces along body X and Z axis are mainly dependent on engine pitch gimbal actuation. Hence, coupling can be ignored when computing forces in the X and Z directionsfor determining moments about the Y axis due to thrust.

11. No RCS jet dynamics or jet select logic will be simulated. During periods of time when the vehicle engines are burning, attitude control is accomplished by engine gimbaling. When engines are off and the vehicle is coasting, the ideal autopilot will provide the required RCS control torques to the appropriate moment equation. Thus, the

4.0 SUMMARY (Continued)

pseudo RCS system will provide sufficient torques to keep attitude errors nearly zero.

12. Tables for aerodyanmic data for the spacecraft are compatible for angles of attack up to 60°. If angles of attack (α) between 60° and 90° are desired, the aerodynamic coefficient for an α of 60° is used. If α is greater than 90°, aerodynamic coefficients are set to zero. If α is less than zero the Tables are interrogated with the magnitude of α. Maneuvers requiring large angles of attack (α greater than 60°) will be restricted to times when the dynamic pressure (q) is small. Thus, errors of the above process are assumed small with respect to thrusting effects.

13. Interrogation of aerodynamic tables for large angle-of-sideslip (β) is assumed to be a linear extrapolation of the data tables.

14. The aerodynamic force and moment equations are written in body axis. The aerodynamic data is transformed from stability axis to body axis where necessary. Data updates may require transformation prior to use in the simulation.

15. Aerodynamic force and moment coefficients are assumed to be a linear function of α about the trim value for the longitudinal derivatives and a linear function of β about the trim value for the lateral derivatives. For the simulation mechanization, all of the linear coefficients will be programmed and looked-up as a function of Mach number and α or β.

5.0 SIMULATION DESCRIPTION

Basic flow charts and symbol definitions are shown in Appendix A. The charts and symbols used are symbols as coded in LASS. Provisions are made to denote corresponding engineering symbols where they are applicable. Thus, a correspondence is provided between the coded and engineering symbol and a common definition.

In general, an attempt was made to code the symbols based on their origin. As an example, symbols of variables that were generated in the INITIA subroutine started with the letter S. This philosophy was established after the AERO, ATMOS3, and COEF Subroutines were coded. The variables generated by these subroutines and used by other Subroutines should have been coded with symbols starting with the letter D. The letter D was given to forces and moments generated in AERO and used in EOM.
There are several other isolated exceptions, but the symbol format shown in Table I was generally selected.

SUBROUTINE	SYMBOL PREFIX LETTER
INITIA	S
GUIDAN	G
EOM & Functions	E
AERO, ATMOS3, COEF	D
TARGET & ABGUID	T
AUTOP2 & AUTOPI & ACC	A
PBC	P
READO	R

TABLE I - Symbol code assignment philosophy

The program listings of LASS are provided in Appendix B. These listings represent the status of LASS at the time of this document preparation.

5.0 SIMULATION DESCRIPTION (Continued)

Constants, variables requiring an initial value, input data tables, and variables which control the simulation are placed primarily in the data block of the INITIA subroutine. The function of the INITIA subroutine is to initialize, provide an interface with the program user, and provide launch targeting. In some instances the master control routine is used to initialize some constants peculiar to the orbiter. Additional data is read from input data cards as described in section 1.1 in Volume 2. In general the user can perform production simulation runs with changes made to the master simulation control input data cards or the INITIA subroutine. The user should be aware of the repetition rates of the simulation. GUIDAN, ABGUID or PBC subroutines are executed once while the EOM, AERO, and autopilot subroutines are executed five times. One pass through the simulation is considered to be one second. This fact is embedded in the EOM and autopilot subroutines. If the users would want to change this relationship, both EOM and autopilot subroutines would require modification.

5.1 MASTER SIMULATION CONTROL

Flow charts and symbol definitions of the master simulation control are shown in Appendix A. The master simulation control is used to sequence LASS, update time and read input data cards, reference section 1.1, Volume 2. First, as shown in Figure 1, the INITIA subroutine is called to initialize LASS. Then, GUIDAN is called to provide desired launch guidance commands. When time (T) reaches a predetermined abort desired time (ABTT), the abort is commanded and ABGUID subroutine is called to provide abort guidance commands. Regardless of whether launch or abort guidance is used, AERO, EOM, and AUTOPI or AUTOP2 subroutines are called in sequence to provide a thrust vector that follows the guidance commands.

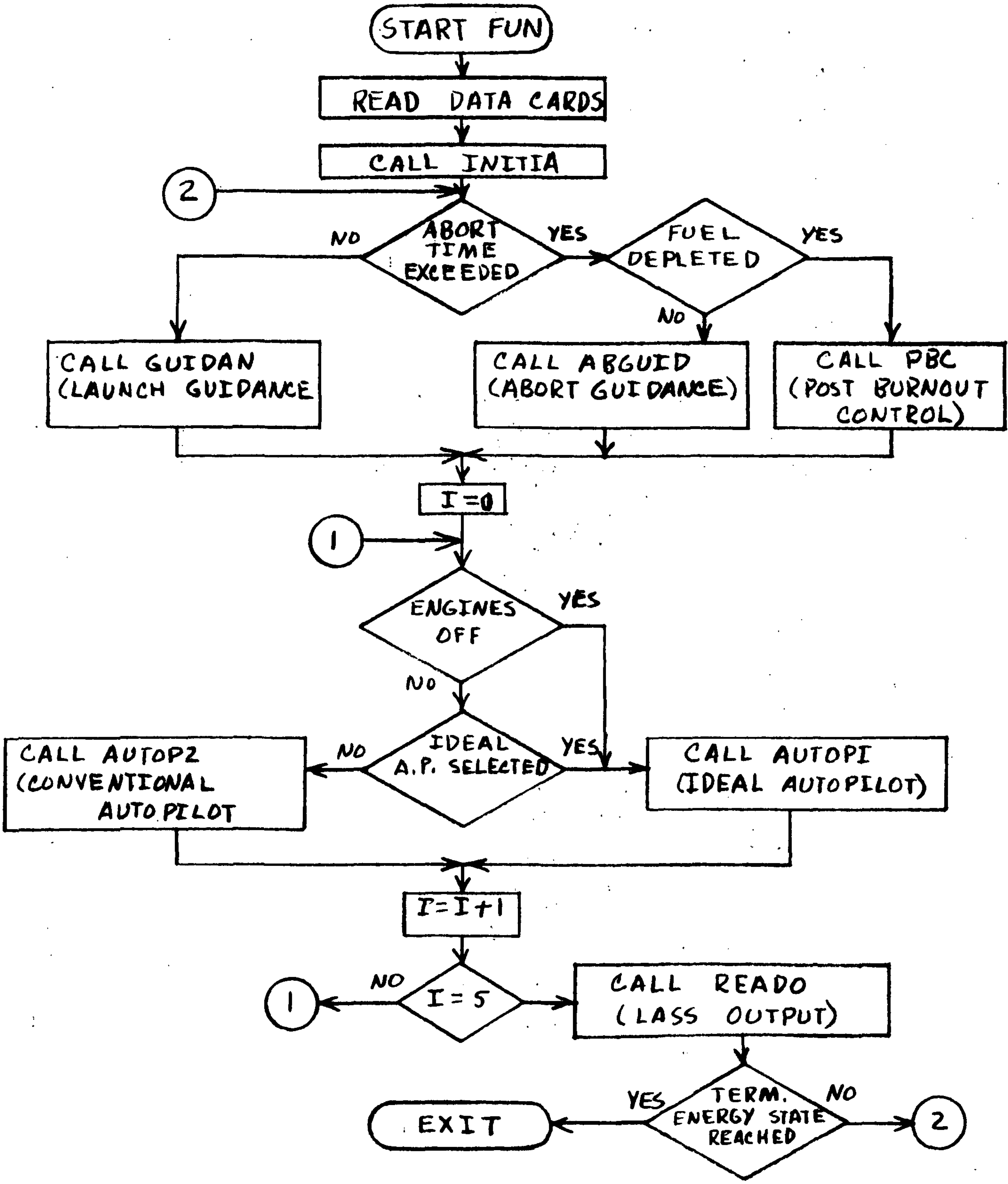

START FUN
READ DATA CARDS
CALL INITIA
2
ABORT TIME EXCEEDED
NO
YES
FUEL DEPLETED
YES
NO
CALL GUIDAN (LAUNCH GUIDANCE
CALL ABGUID (ABORT GUIDANCE)
CALL PBC (POST BURNOUT CONTROL)
I =0
1
ENGINES OFF
YES
NO
IDEAL A.P. SELECTED
NO
YES
CALL AUTOP2 (CONVENTIONAL AUTO PILOT
CALL AUTOP1 (IDEAL AUTOPILOT)
I=I+1
I = 5
NO
1
CALL READO (LASS OUTPUT)
TERM. ENERGY STATE REACHED
YES
NO
EXIT
2

FIGURE 1 BASIC FORMAT OF LASS

5.1 MASTER SIMULATION CONTROL (Continued)

Booster engines off (IES) may be commanded at any point in the simulation. As soon as engines off is commanded the master control continues cycling until KC11 is counted down. When this occurs, separation (ISEP) is commanded. As soon as ISEP is commanded, several orbiter constants are initialized. When KC12 is counted down, the master control resets IES and commands an engine start. Thus, KC11 controls time after Booster engine shutdown until separation and KC12 controls time after separation until orbiter engine start. If an abort is commanded, after separation, logic is provided to prevent an attempt to shut down the engines.

Following a normal launch, orbit insertion is presently defined by a test on the difference between the magnitude of the existing inertial velocity and the down range velocity, $\dot{Z}$, desired. If the absolute value of this difference is less than 5 ft/sec, an engine shutdown is commanded. In the case of an abort situation, a test is made on time to go to abort target (TGØT) and fuel remaining. If TGØT is less than one second or fuel is depleted, an engine shutdown is commanded. In both cases, after consumation of powered flight, the vehicle is allowed to coast until a terminal energy state is reached.

5.2 INITIALIZATION MATH MODEL

The initialization math model is denoted as the INITIA subroutine. The main purpose of INITIA is to initialize the remaining math models comprising the simulation and provide launch targeting. The flow chart and list of symbol definitions are shown in Appendix A. A functional descriotion of the INITIA subroutine is shown in Figure 2. As shown in Figure 1, INITIA is executed only once during a simulation run. The math model must be initialized with the information shown in Table II.

The INITIA subroutine is basically divided into data block initialization and special computation. Special situations depend on launch pad to earth geometry, desired orbit, and desired abort landing site.

5.2.1 Data Block

The data block is used to initialize LASS with constant values. The actual form of the data block together with definitions are shown in Appendix A. Many large data tables are housed in INITIA data block but used in other LASS subroutines. These tables are discussed in specific sections of the document in which they are used. Specific references are shown in Table III.

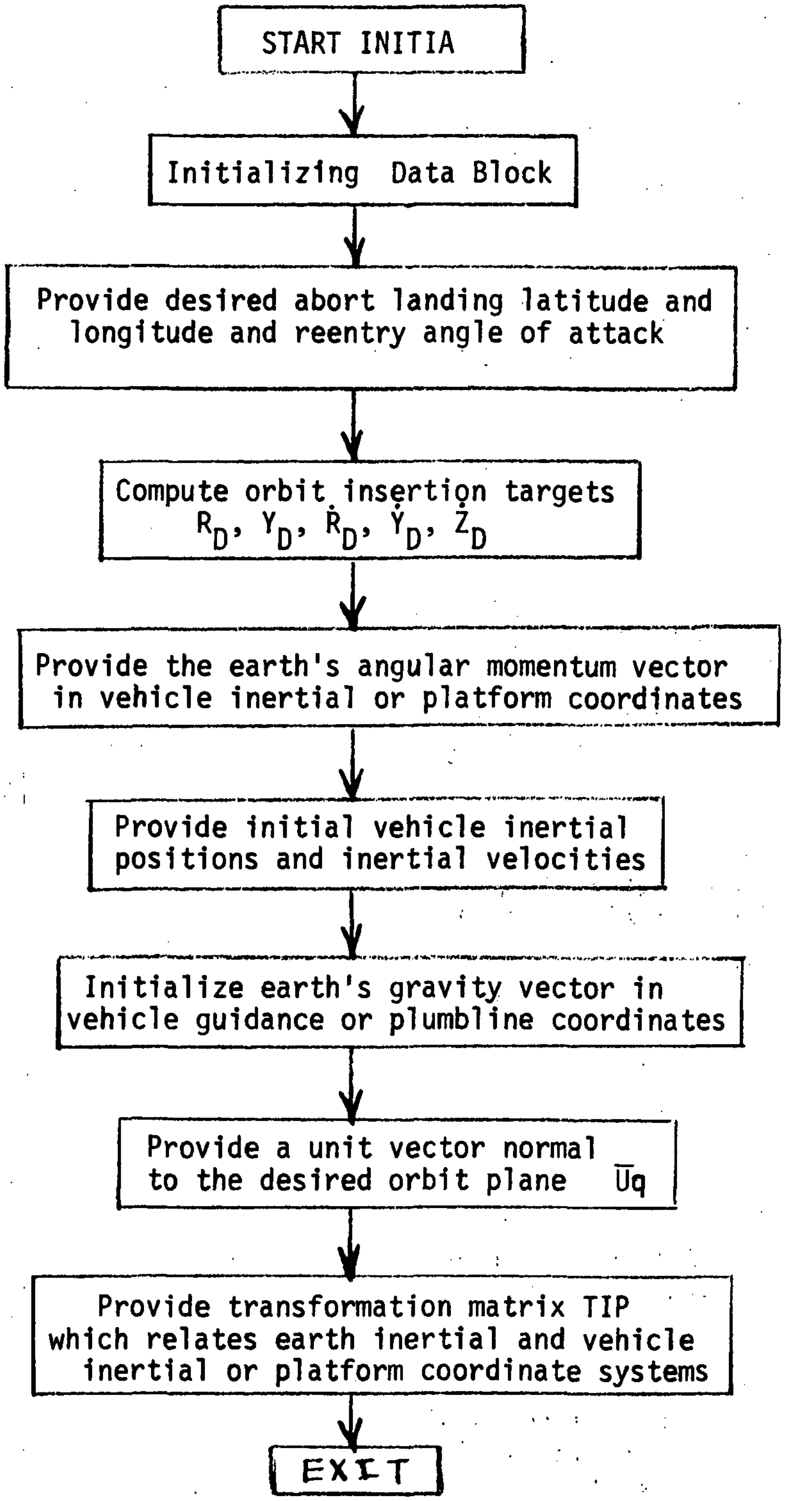

START INITIA
Initializing Data Block
Provide desired abort landing latitude and longitude and reentry angle of attack
Compute orbit insertion targets
R_D, Y_D, $\dot{R}_D$, $\dot{Y}_D$, $\dot{Z}_D$
Provide the earth's angular momentum vector in vehicle inertial or platform coordinates
Provide initial vehicle inertial positions and inertial velocities
Initialize earth's gravity vector in vehicle guidance or plumbline coordinates
Provide a unit vector normal to the desired orbit plane $\overline{U}q$
Provide transformation matrix TIP which relates earth inertial and vehicle inertial or platform coordinate systems
EXIT

FIGURE 2 FUNCTIONAL DESCRIPTION OF THE INITIA SUBROUTINE

TABLE II INPUTS REQUIRED TO INITIALIZE INITIA SUBROUTINE

CODED SYMBOL	ENGR. SYMBOL	DEFINITION
TLAT	---	Latitude of desired landing site after an abort - (deg.)
TLONG	---	Longitude of desired landing site after an abort - (deg.)
SPERH	h_P	Desired perigee altitude (ft.)
SAPOH	h_a	Desired apogee altitude (ft.)
STHETA	θ_{PA}	Desired perigee phase angle from insertion (rad.)
SMU	μ	Gravitational constant (ft^3/sec^2)
SEROT	ω_e	Earth rotational constant (rad./sec.)
SRMEAN	r_a	Earth mean radius (ft)
SVEII	β	Central angle between a vector through the launch pad and the intersection of orbit and launch planes (deg.)
SLAZI	A_Z	Launch azimuth (deg.)
SLATI	λ_L*	Latitude of launch pad (deg.)
SINCI	A_L	Desired orbit plane inclination angle (deg.)
SLONGI	ϕ_L	Longitude of launch pad (deg.)

TABLE III CROSS REFERENCE OF DATA TABLES IN THE DATA BOOK

DATA TABLE	USING SUBROUTINE OR FUNCTION	NOMENCLATURE
W(14)	FUNCTION	Mass properties
XCG(14)	FUNCTION	Mass properties
YCG(14)	FUNCTION	Mass properties
ZCG(14)	FUNCTION	Mass properties
MXX(14)	FUNCTION	Mass properties
MYY(14)	FUNCTION	Mass properties
MZZ(14)	FUNCTION	Mass properties
MXZ(14)	FUNCTION	Mass properties
GMT(12)	GUIDAN	Launch Guidance
GTA(12)	GUIDAN	Launch Guidance
TT(6)	TARGET	Burnout velocity
HB(10)	ATMOS3	Atmospheric properties
ZB(14)	ATMOS3	Atmospheric properties
TMB(24)	ATMOS3	Atmospheric properties
ALP(24)	ATMOS3	Atmospheric properties
XGB(14)	ATMOS3	Atmospheric properties
EL(12, 3)	AUTOPI & EOM	Engine locations (booster)
ELO(2, 3)	AUTOPI & EOM	Engine locations (orbiter)

5.2.2 Launch Targeting

Launch targeting is divided into two parts, the first part constitutes development of orbit insertion targets. The second constitutes the definition of a unit vector normal to the desired orbit plane.

The targets are developed in a conventional manner. The desired orbit is defined by an apogee altitude (h_a), perigee altitude (h_p) , and a phase angle θ_{PA}. The phase angle is set to a value other than zero if insertion is to be made at some orbit point other than perigee.

In LASS, θ_{PA} is set equal to zero and insertion is made at perigee. The apogee and perigee radius are given respectively by:

$$R_A = r_a + h_a \quad \text{and}$$

$$R_P = r_a + h_P$$

Where r_a is the mean earth radius.
The semi-major axis is expressed by:

$$A = \frac{R_A + R_P}{2}$$

The eccentricity is computed by:

$$\epsilon = \frac{h_a - h_p}{R_A + R_P}$$

Then, the desired radial displacement is:

$$R_D = \frac{A\ (1-\epsilon^2)}{1+\ \cos\theta_{PA}}$$

5.2.2 Launch Targeting (Continued)

Since we want to end up on the desired orbit plane, Y_D, the out of plane distance, is set equal to zero. The downrange distance Z_D is not constrained and allowed to be whatever is attained. Thus, R_D and Y_D are the desired displacement targets.

The desired velocity targets are defined by computing total orbit velocity using:

$$V_P{}^2 = \mu\left(\frac{2}{R_P} - \frac{1}{A}\right)$$

The specific angular momentum at perigee is:

$$H = R_P(V_P)$$

and is a constant. The desired downrange velocity is a function of H and R_D. Thus, this desired velocity is:

$$\dot{Z}_D = \frac{H}{R_D}$$

Since, we do not want to allow an out of orbit plane velocity component, $\dot{Y}_D$ is set to zero. This results in a radial velocity of

$$(\dot{R}_D)^2 = (V_P)^2 - (Z_D)^2 - (\dot{Y}_D)^2$$

which reduces to:

$$\dot{R}_D = \left[\mu\left(\frac{2}{P_D} - \frac{1}{A}\right) - (\dot{Z}_D)^2\right]^{\frac{1}{2}}$$

Thus, the velocity targets $\dot{R}_D$, $\dot{V}_D$ and $\dot{Z}_D$ are defined.

The next requirement for launch targeting is to describe the desired orbit plane with respect to the earth and launch pad. This is accomplished by describing a unit vector, $\hat{U}_q$, normal to the desired orbit plane and expressed in vehicle inertial or platform coordinates.

Referring to Figure 3, this is accomplished by assuming we have a unit vector $\hat{W}$ normal to the launch plane. If $\hat{R}_L$ is a unit vector along the radial direction through the launch pad, a unit vector

$$\hat{E} = \hat{R}_L \times \hat{W}$$

is defined in the launch plane. Now let us define an angle β as the angle between $\hat{R}_L$ and a unit vector $\hat{S}$ along the intersection of the orbit and launch planes. Therefore, $\hat{S}$ is defined by:

$$\hat{S} = \hat{R}_L \cos\beta + \hat{E} \sin\beta$$

and in both orbit and launch planes. Another unit vector:

$$\hat{E}_P = \hat{S} \times \hat{W}$$

can be obtained such that it is in the launch plane and also normal to $\hat{W}$. Thus, $\hat{E}_P$ can now be rotated into the desired unit vector $\hat{U}_q$ by:

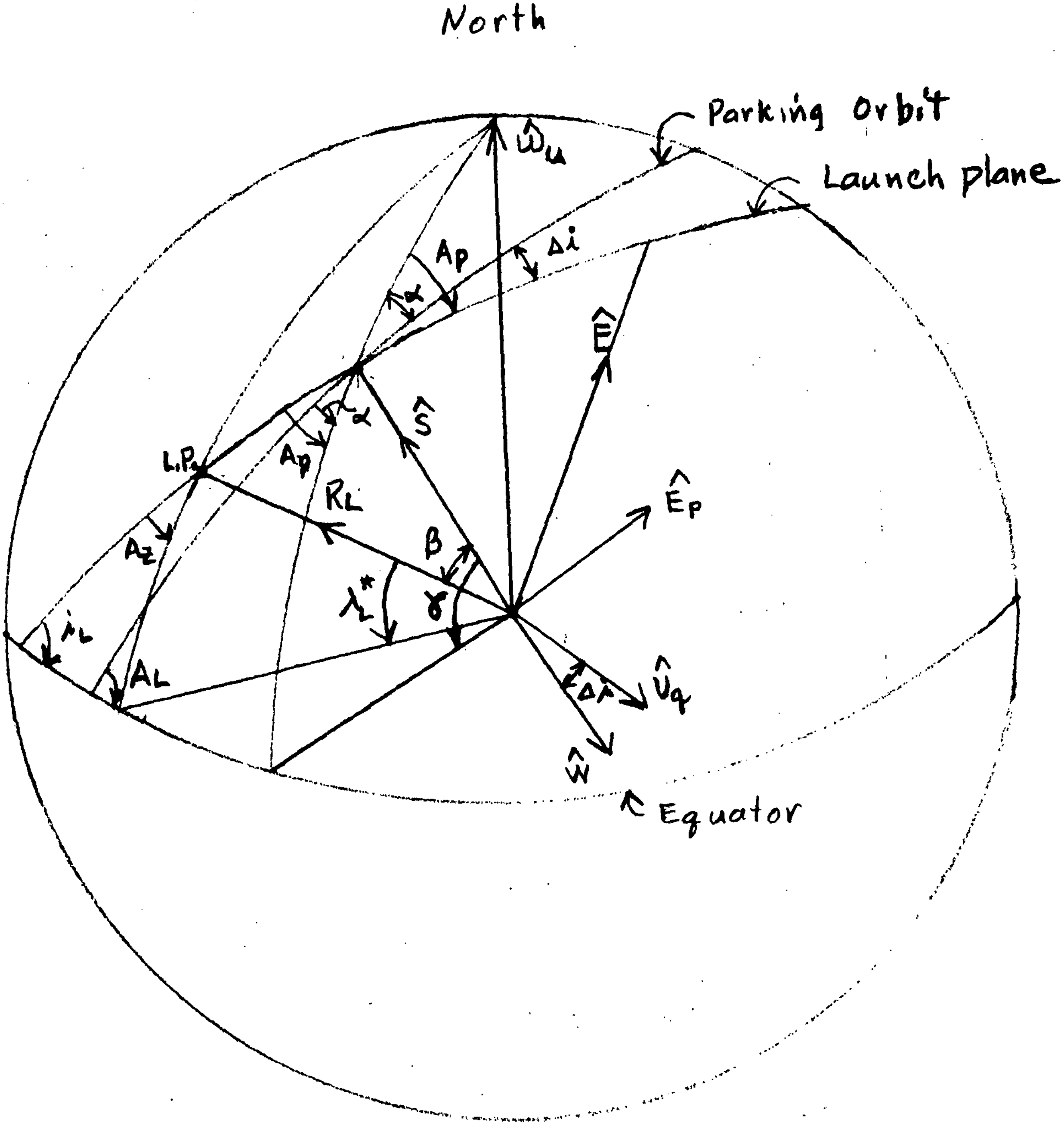


FIGURE 3 DEVELOPMENT OF UNIT VECTOR NORMAL TO ORBIT PLANE

5.2.2 Launch Targeting (Continued)

$$\hat{U}_q = \hat{W} \cos \Delta i + \hat{E}_P \sin \Delta i$$

where Δi is the angle between the orbit and launch planes.

Now, if we define the vehicle inertial or platform coordinate system with the XZ plane coincident with the launch plane and X-axis along the radial through the launch pad, $\hat{R}_L$ and $\hat{W}$ become

$$\hat{R}_L = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{W} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Then $\hat{E}$ becomes:

$$\hat{E} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and $\hat{S}$ becomes:

$$\hat{S} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cos\beta + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \sin\beta = \begin{bmatrix} \cos\beta \\ 0 \\ \sin\beta \end{bmatrix}$$

The unit vector $\hat{E}_p$ is:

$$\hat{E}_p = \begin{bmatrix} \cos\beta \\ 0 \\ \sin\beta \end{bmatrix} X \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin\beta \\ 0 \\ \cos\beta \end{bmatrix}$$

Therefore, the desired unit vector $\hat{U}_q$ is

$$\hat{U}_q = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cos \Delta i + \begin{bmatrix} -\sin\beta \\ 0 \\ \cos\beta \end{bmatrix} \sin \Delta i = \begin{bmatrix} -\sin\beta \sin \Delta i \\ \cos \Delta i \\ \cos\beta \sin \Delta i \end{bmatrix}$$

5.2.2 Launch Targeting (Continued)

Now we have to define the angle between the launch plane and desired orbit plane. This is accomplished by describing a unit north vector, $\hat{W}_u$. The vector $\hat{W}_u$ is obtained by defining the earth's angular momentum in vehicle inertial or platform coordinates and then reducing that vector to unit form. The development of the earth's angular momentum vector is shown later. By observation of Figure 3:

$$\sin \gamma = \hat{W}_u \cdot \hat{S} = SSG$$

From spherical trigometry:

$$\cos i_L = \sin A_Z \cos \lambda_L^* = SCIL$$

and

$$\sin A_P = \frac{\cos i_L}{\cos \gamma} = SAP$$

and

$$\sin \alpha = \frac{\cos A_L}{\cos \gamma} = SSALF$$

Then the desired angle is:

$$\Delta i = A_P - \alpha$$

The symbols SSG, SCIL, SAP, and SSALF are the coded symbols used in INITIA for the respective defined values.

5.2.2 Launch Targeting (Continued)

The angle β is selected as a function for optimizing the launch trajectory. Different values of β were tried and a value of -17^{o} was selected as near optimum for an orbit plane inclination of 55^{o}.

5.2.3 Coordinate Systems

The INITIA subroutine recognizes two basic coordinate systems. These are the earth inertial system (X_F, Y_F, and Z_F) and vehicle inertial or platform system (X_P, Y_P, and Z_P). Both systems are earth centered and maintain a fixed relationship between each other as shown in Figure 4. The Z_F axis points north along the earth's rotation axis. The X_F axis points toward the intersection of the equator and the ecliptic with the positive sense in the descending node of the ecliptic. The Y_F axis completes the right handed orthogonal triad.

The platform system is taken with the X_P axis along an earth's radial directed away from the earth's center. The Y_P axis is taken as normal to the launch plane and directed to the right. Z_P completes the right handed orthogonal triad pointing downrange. The TIP transformation provides for the relationship between these two coordinate systems. The rotational references are shown in Figure 5. The actual transformation about the X_P axis is programmed as a rotation about $(\pi/2 - A_Z)$. If these equations , shown in the flow chart of Appendix A for INITIA, are expanded, they would reduce to the form shown in Figure 5.

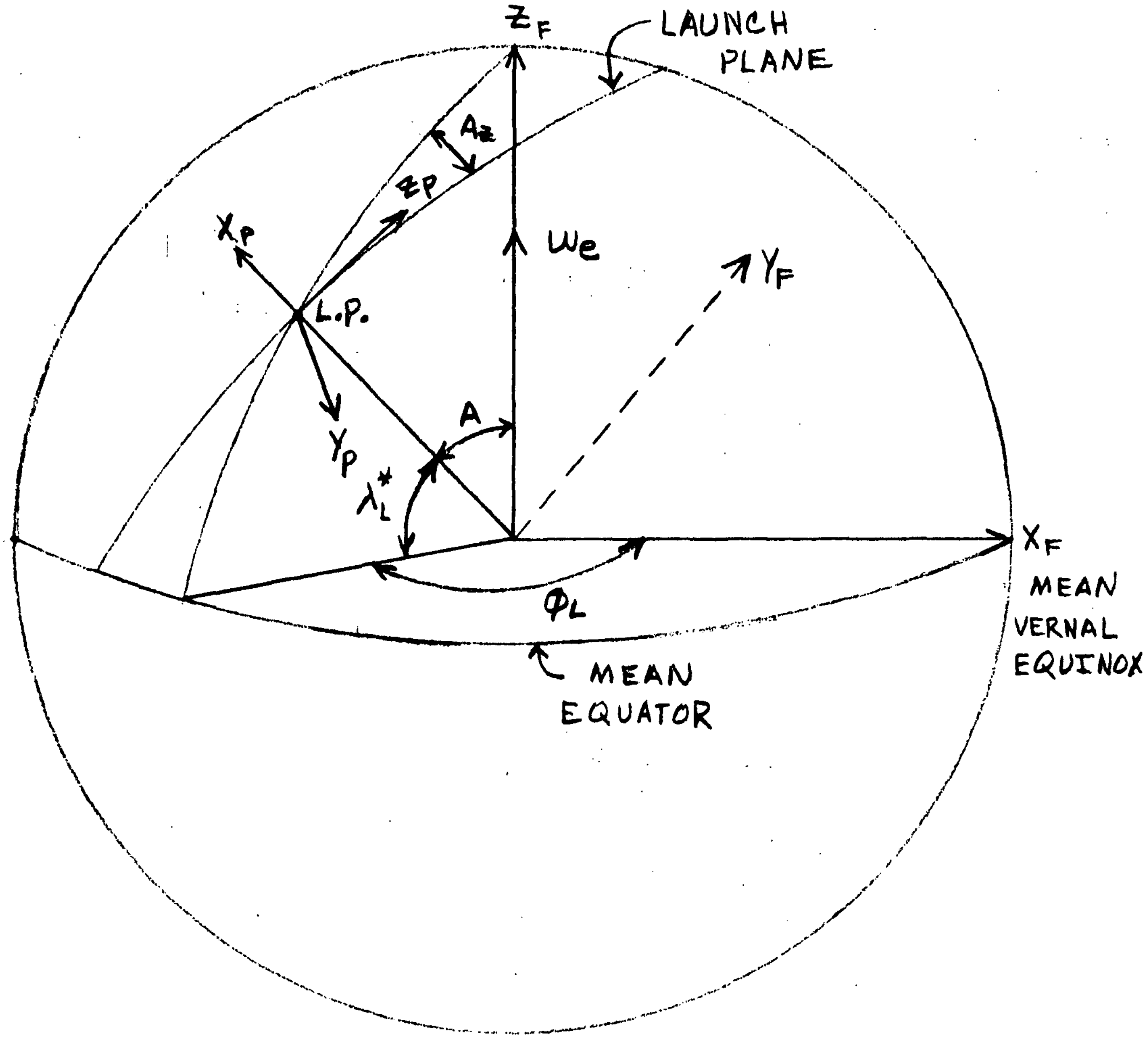
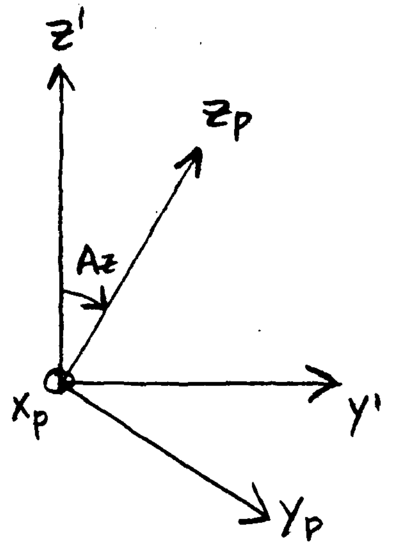


FIGURE 4 COORDINATE SYSTEMS USED IN INITIA SUBROUTINE

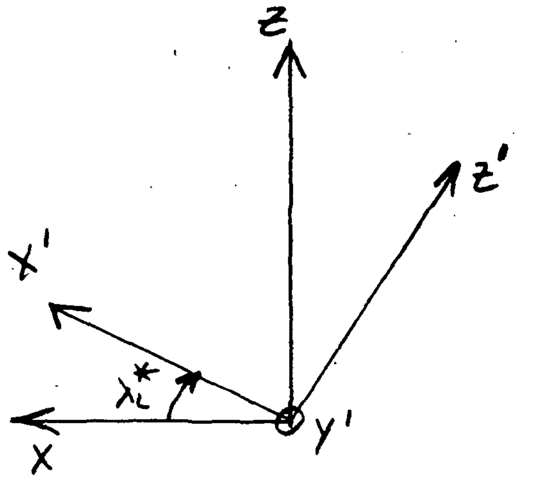


$X_P = X'$

$Y_P = Y'\cos A_Z + Z'\sin A_Z$

$Z_P = -Y'\sin A_Z + Z'\cos A_Z$

ROTATION IN
AZIMUTH

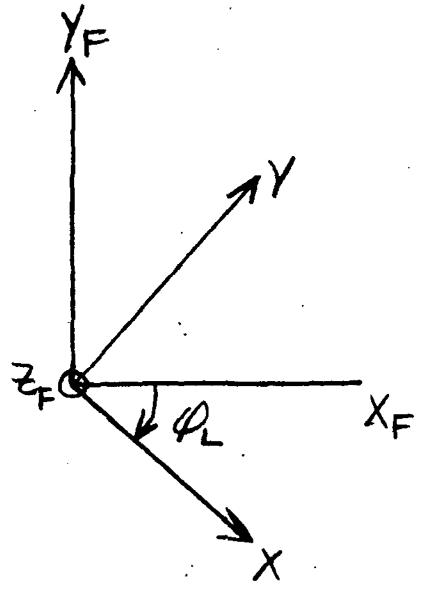


$X' = X\cos\lambda_L^* - Z\sin\lambda_L^*$

$Y' = Y$

$Z' = +X\sin\lambda_L^* + Z\cos\lambda_L^*$

ROTATION IN
LATITUDE



$X = X_F\cos\Phi_L - Y_F\sin\Phi_L$

$Y = X_F\sin\Phi_L + Y_F\cos\Phi_L$

$Z = Z_F$

ROTATION IN
LONGITUDE

Note: Φ_L is inputed as a negative angle

FIGURE 5 ROTATION INVOLVED IN THE TIP COORDINATE TRANSFORMATION

5.2.4 Miscellaneous Functions of INITIA Subroutine

The INITIA subroutine is used to provide the earth's angular momentum vector in platform coordinates. Referring to Figure 4 geometry we can readily see that:

$$A = 90^{o} - \lambda_{L}{}^{*}$$

$$\omega_{ep}(1) = \omega_{e} \cos A$$

$$\omega_{ep}(2) = \omega_{e} \sin A_{Z} \sin A$$

$$\omega_{ep}(3) = \omega_{e} \cos A_{Z} \sin A$$

$$\omega_{ep}(4) = |\overline{\omega_{ep}}|$$

The earth's angular momentum vector expressed in platform coordinates is used to develop the $\hat{\Omega}_{u}$ unit vector used in computing $\hat{U}_{q}$ shown in Figure 3. The $\overline{\omega_{ep}}$ vector is also used in EOM, TARGET, AERO, and READO subroutines.

The INITIA also provides for initial values of inertial position, velocity, and acceleration in platform coordinates. The values are used in the EOM subroutine to initialize the respective integrators.

5.3 LAUNCH GUIDANCE MATH MODEL (GUIDAN)

Flow charts and symbol definitions of GUIDAN are shown in Appendix A. The GUIDAN subroutine is used to provide desired inertial attitudes to the AUTOPI or AUTOP2 subroutine during a nominal launch. The autopilot uses these desired angles with actual vehicle angles to control the thrust vector. This keeps the vehicle going in the desired direction during the launch portion of vehicle flight. Whenever an abort is commanded, GUIDAN relinquishes control of the vehicle to the ABGUID subroutine.

The GUIDAN subroutine requires input data listed in Table IV. GUIDAN outputs are inertial angles to the AUTOPI as given below:

Coded Sym	Engr Sym	Nomenclature
GTHED	θ_D	inertial pitch angle
GPSID	ψ_D	inertial yaw angle
GPHID	ϕ_D	inertial roll angle

These angles represent the only outputs from GUIDAN required by other subroutines.

A functional flow chart of the GUIDAN subroutine is shown in Figure 6. The vehicle follows a prescribed pitch angle for a designated time. Guidance selection is made by assigning a value to GTPPCØ. After time designated by GTPPCØ, the simulation uses E-guidance to guide the vehicle. Acceleration limits of 2.5 g's for the launch configuration (booster/orbiter) and 3 g's for the orbiter alone configuration are utilized by the guidance procedures. Regardless of the situation,

TABLE IV INPUT DATA REQUIRED BY GUIDAN SUBROUTINE

CODED SYM.	ENGR. SYM.	NOMENCLATURE
SRD	R_D	Orbit insertion target
SYD	Y_D	Orbit insertion target
SRDOTD	$\dot{R}_D$	Orbit insertion target
SYDOTD	$\dot{Y}_D$	Orbit insertion target
SZDOTD	$\dot{Z}_D$	Orbit insertion target
SUQ	U_q	Unit vector normal to orbit plane
ERP	$\bar{R}_P$	Inertial position
EVP	$\bar{V}_P$	Inertial velocity
AZCF	Z_{CF}	Reference engine cluster location
EXCG(AMTØT)	X_{CG}	Vehicle center of gravity location
EYCG(AMTØT)	Y_{CG}	Vehicle center of gravity location
EZCG(AMTØT)	Z_{CG}	Vehicle center of gravity location
EATT	A_T	Vehicle thrust acceleration
EPT	---	Fractional throttle setting
GCØ(1)	---	Time to go limit
GCØ(2)	A_{LB}	Acceleration limit
GCØ(3)	A_{LO}	Acceleration limit
GDT	---	GUIDAN cycle time
EGEFF	---	Gravity acceleration
GEGVEL	V_e	Engine exhaust gas velocity
GTPPCØ	---	Time to terminate pitch angle lookup
IES	---	Engines on
ISEP	---	Separation
GTA(12)	---	Pitch angle
PI	π	Constant
T	---	Mission time

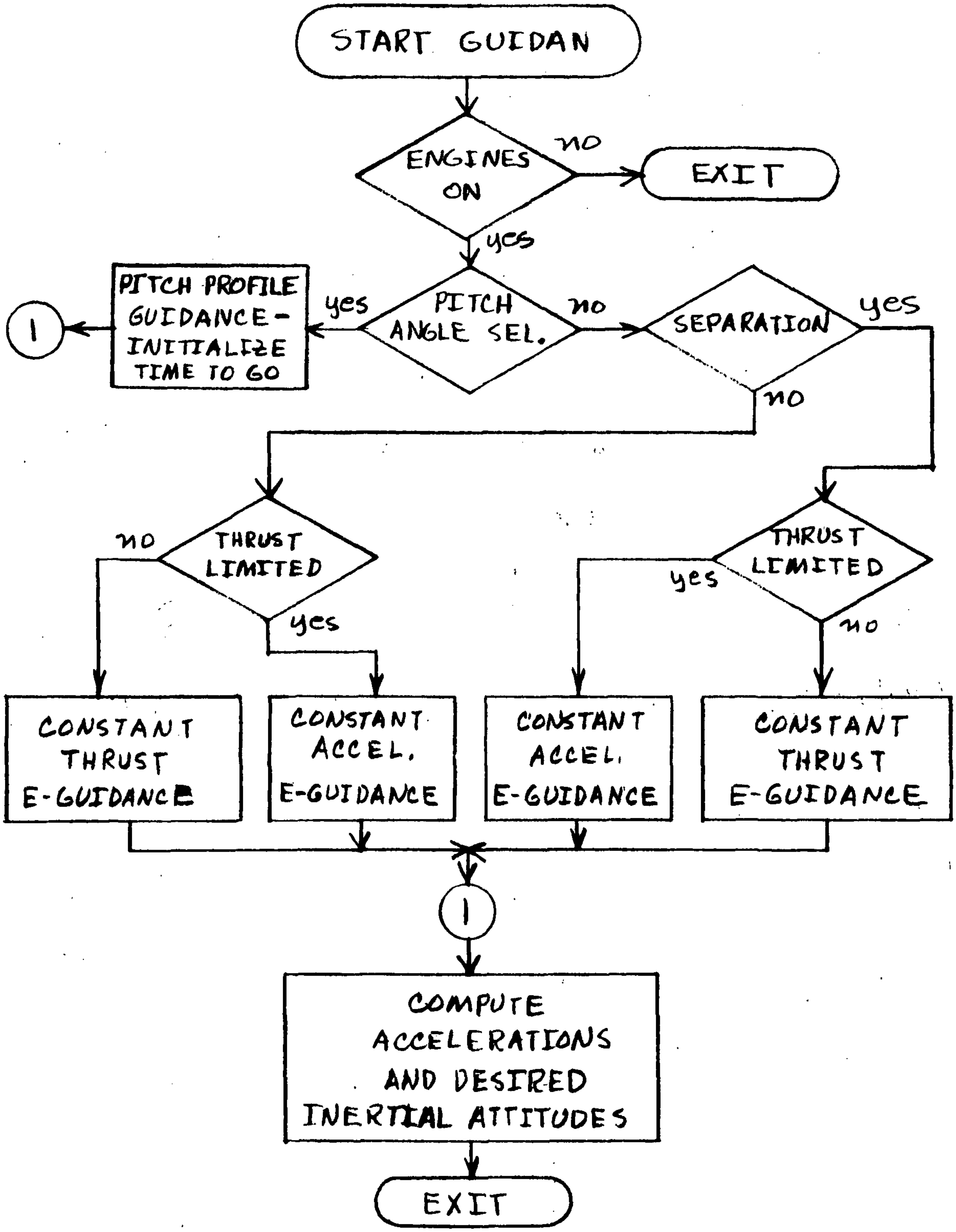

START GUIDAN
ENGINES ON
no
EXIT
yes
PITCH ANGLE SEL.
yes
PITCH PROFILE GUIDANCE - INITIALIZE TIME TO GO
1
no
SEPARATION
yes
no
THRUST LIMITED
no
yes
THRUST LIMITED
yes
no
CONSTANT THRUST E-GUIDANCE
CONSTANT ACCEL. E-GUIDANCE
CONSTANT ACCEL. E-GUIDANCE
CONSTANT THRUST E-GUIDANCE
1
COMPUTE ACCELERATIONS AND DESIRED INERTIAL ATTITUDES
EXIT

FIGURE 6 FUNCTIONAL FLOW CHART OF GUIDAN SUBROUTINE

5.3 LAUNCH GUIDANCE MATH MODEL (GUIDAN) (Continued)

desired accelerations are computed. These are used to provide the inertial angles that locate the direction of the available thrust in platform coordinate space.

5.3.1 COORDINATE SYSTEMS

The basic coordinate system used in GUIDAN is the platform system described in section 5.2.3. GUIDAN erects a guidance coordinate system which is referenced to the platform system. The launch guidance plumbline coordinate system is shown in Figure 7. The X_G-Axis is taken along a radial pointing away from the earth's center and out of the paper in Figure 7. The Z_G-Axis is in a local horizontal plane normal to X_G and $\hat{U}_q$. Thus, Z_G is directed downrange and parallel to the launch plane. The Y_G-Axis completes the righthanded orthogonal triad.

The desired inertial attitude angles are shown in Figure 8. The inertial pitch angle θ_{DI} locates the desired acceleration vector with respect to platform X_P axis, where the sub P denotes platform coordinates. The actual computation is:

$$\theta_{DI} = TAN^{-1}\left[\frac{A_{D3}}{A_{D1}}\right]$$

θ_D locates the projection of body X-axis in the $X_P Z_P$ plane with respect to X_P-Axis. Then:

$$\theta_D = -\theta_{DI} + TAN^{-1}\left[\frac{Z_{CF} - Z_{CG}}{Z_{CG}}\right]$$

where Z_{CF} is the engine thrust centerline offset.

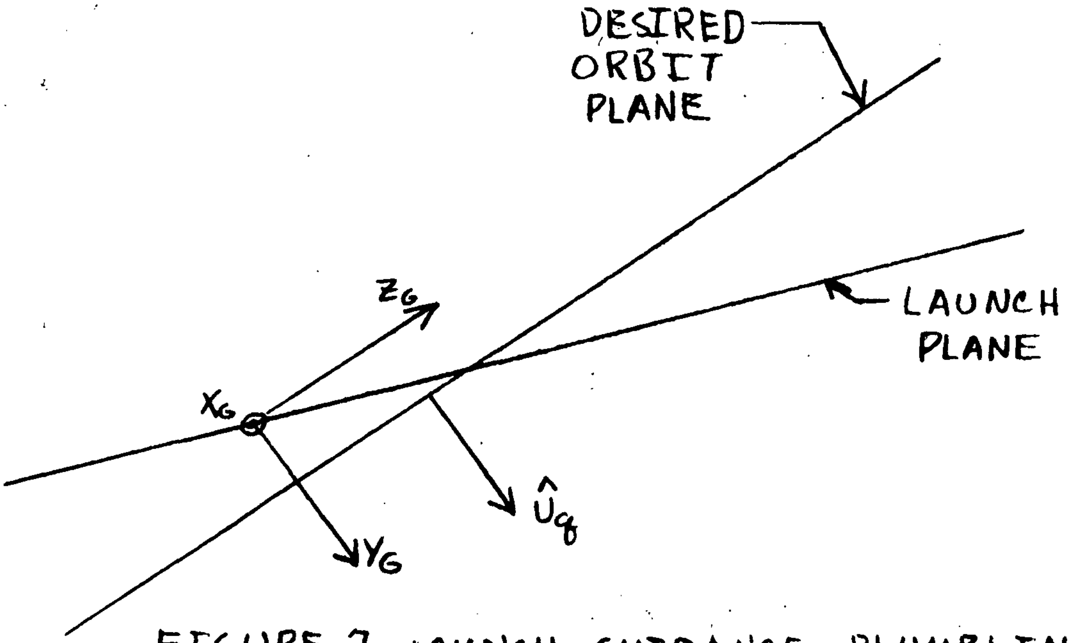


FIGURE 7 LAUNCH GUIDANCE PLUMBLINE COORDINATE SYSTEM

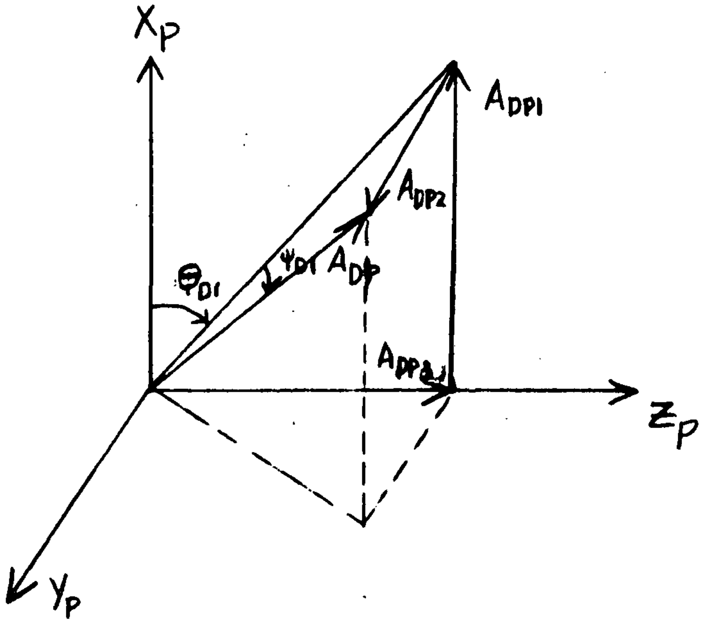


FIGURE 8 SENSE OF DESIRED INERTIAL ATTITUDE ANGLES.

5.3.1 COORDINATE SYSTEMS (Continued)

The negative sense is taken as nose down. The body axis coordinate system is explained in section 5.4.3.

The inertial yaw angle ψ_{D1} is computed by:

$$\psi_{D1} = \text{SIN}^{-1} \left[\frac{A_{D2}}{|\bar{A}_D|} \right]$$

This locates the desired acceleration with respect to the X_pZ_p plane. ψ_D locates body X-axis with respect to the X_pY_p plane and is:

$$\psi_D = \psi_{D1} - \text{TAN}^{-1} \left[\frac{Y_{CG}}{X_{CG}} \right]$$

The inertial roll , ϕ_D, is the remaining rotation about the X_p axis. The LASS ϕ_D is maintained at zero. Since, travel over the earth's surface is not too extensive, the remaining ϕ_D necessary to keep wings level is ignored. The order of rotation is nose up, nose right, and then roll in the counterclockwise sense about X_p.

5.3.2 PITCH PROFILE GUIDANCE

During initial portions of the launch, the vehicle is forced to fly a canned pitch inertial angle. The actual pitch angles used are shown in Table V. Time is used to enter the table and linear interpolation is used to obtain intermediate values. Pitch angle is inserted as negative angles, since the angles are defined as nose up being in the positive sense.

5.3.2 PITCH PROFILE GUIDANCE (Continued)

Pitch angles are selected from the pitch profile table until a preset time, GTPPCØ, is reached. GTPPCØ is termed pitch profile cutoff time. Thus, data is taken from the profile table for any time duration from launch to the maximum value in the table. It should be noted, launch guidance used is not arranged in a form to be used as two stage guidance. The booster operates in a constant thrust mode until a 2.5 g limit is reached. At that time thrust is limited so the booster flies in a constant acceleration mode. In order for pitch profile guidance to be compatible with the E-guidance, used in LASS, GTPPCØ should not be set less than the time to booster thrust transition. A GTPPCØ time of 212 seconds is presently being used in LASS.

TABLE V INERTIAL PITCH PROFILE TABLE

TIME (Sec)	PITCH ANGLE (Radians)
14	-.02189
20	-.06383
30	-.10598
40	-.15467
50	-.22454
60	-.31066
100	-.81915
120	-1.04268
140	-1.21357
180	-1.41588
220	-1.56391
240	-1.59823

Present LASS program is arranged to compute desired acceleration, A_{DP}, in platform coordinates. When in the pitch profile mode of guidance, A_{DP} is computed but not used. However, in E-guidance mode of operation A_{DP} is developed and used to compute the desired inertial angles.

5.3.2 PITCH PROFILE GUIDANCE (Continued)

Six seconds prior to time equal to GTPPCØ,branching is accomplished to compute an initial estimate of time-to-go, T_{GO}, This time is an estimate of time remaining to achieve orbit insertion. The estimate serves to initialize E-guidance with an initial value of T_{GO} based on present velocity and effective gravity. A discussion of the T_{GO} equations can be found in the next section.

5.3.3 E-GUIDANCE

The use of E-guidance depends on a good estimate of time, T_{GO}, remaining to reach targeted orbit insertion. Effective gravity, q_{eff}, and T_{GO} has a large effect when determining velocity to be gained, V_G. But V_G is used to provide T_{GO}. Thus, an iteration loop is formed. For this reason, T_{GO} is initialized when in the pitch profile mode of guidance, as noted previously.

T_{GO} is obtained by first erecting the plumbline coordinate system. This is accomplished by computing unit vectors:

$$\hat{U}_{RP} = UNIT\ (\overline{R}_P)$$

$$\hat{U}_{ZP} = \hat{U}_{RP} \times \hat{U}_q$$

$$\hat{U}_{YP} = \hat{U}_{ZP} \times \hat{U}_{RP}$$

where $\hat{U}_q$ is the unit normal to the orbit plane and $\hat{U}_{RP}$, $\hat{U}_{YP}$, and $\hat{U}_{ZP}$ correspond to X_G, Y_G, and Z_G in Figure 7 respectively. $\hat{U}_{RP}$, $\hat{U}_{YP}$, and $\hat{U}_{ZP}$ are expressed in the platform coordinate system and form a direction cosine transformation matrix relating the platform and plumbline coordinate systems. Present vehicle inertial velocity is converted to velocities in the plumbline system by:

5.3.3 E-GUIDANCE (Continued)

$$\begin{bmatrix} \dot{R} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \hat{U}_{RP} \\ \hline \hat{U}_{YP} \\ \hline \hat{U}_{ZP} \end{bmatrix} \begin{bmatrix} \bar{V}_P \end{bmatrix}$$

Since INITIA has already provided the desired target velocities, the velocity to be gained in the plumbline system is

$$\begin{bmatrix} \dot{R}_G \\ \dot{Y}_G \\ \dot{Z}_G \end{bmatrix} = \begin{bmatrix} \dot{R}_D \\ \dot{Y}_D \\ \dot{Z}_D \end{bmatrix} - \begin{bmatrix} \dot{R} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}$$

Where $\dot{R}_G$, $\dot{Y}_G$, and $\dot{Z}_G$ correspond to X_G, Y_G, and Z_G plumbline coordinate axes respectively.

By taking into account gravity effects, the velocity to be gained in the platform system is

$$\begin{bmatrix} \bar{V}_G \end{bmatrix} = \begin{bmatrix} \hat{U}_{RP} & \vdots & \hat{U}_{YP} & \vdots & \hat{U}_{ZP} \end{bmatrix} \begin{bmatrix} \dot{R}_G \\ \dot{Y}_G \\ \dot{Z}_G \end{bmatrix} - \frac{1}{2} g_{eff} T_{GO} \begin{bmatrix} \hat{U}_{RP} \end{bmatrix}$$

Now, if we define

$$\tau = \frac{Ve}{A_T}$$

where A_T is thrust acceleration, T_{GO} is computed in its first approximation by

$$T_{GO} = \tau \left[1 - e^{-\frac{|\bar{V}_G|}{Ve}} \right]$$

The actual derivation of T_{GO} equation is found by solving the characteristic velocity equation found in the references. These equations form the iterative loop which provides T_{GO} to be used in the guidance equations.

5.3.3 E-GUIDANCE (Continued)

The guidance equations used in LASS recognize the separation of the vehicles. For this reason a pitch profile is provided up to separation. In actual practice, the selection of a pitch angle from the profile table can be terminated when the booster/orbiter configuration is in the constant acceleration mode of operation without much effect on results attained. If termination occurs while in the constant thrust mode of operation, T_{GO} computation and guidance equations must reflect two stage method of launch. The two stage effects have not been placed in LASS but are available and could be added.

The actual guidance equations are divided into two parts. These are constant thrust, where the vehicle is allowed to use all the available thrust the engines can provide, and constant acceleration, where the engines are throttleddown to maintain a constant vehicle acceleration.

The starting point for constant thrust E-guidance is to define a time (T_{FL}) until limiting acceleration (A_L) is reached. An estimate of T_{FL} is accomplished by

$$T_{FL} = \frac{Ve}{A_T} - \frac{Ve}{A_L}$$

where, V_e is the exhaust gas velocity of the engines and A_T is the total acceleration. The characteristic velocity equation for constant thrust is

$$\Delta V_i = Ve \ln \left[\frac{\tau}{\tau - T} \right]$$

5.3.3 E-GUIDANCE (Continued)

where, T is the time duration of constant thrust and τ is the time to consume the mass of the vehicle and computed previously. This equation can be written as

$$-\Delta V_i = V_e \ln \left[1 - \frac{T}{\tau}\right]$$

Then

$$-\Delta V_i = V_e \ln \left[1 - \frac{T_{FL}}{\tau}\right]$$

represent velocity gained by a constant thrust for T_{FL} seconds. During constant acceleration time to go is simply

$$T_{GO} = \frac{\text{Velocity to be gained}}{A_L}$$

Then the T_{GO} equation for constant thrust followed by constant acceleration is

$$T_{GO} = T_{FL} + \frac{\left[|\bar{V}_G| + \Delta V_i\right]}{A_L}$$

where ΔV_i is taken in its negative sense and represents the velocity gained during T_{FL} seconds of constant thrust, The bracketed term represents velocity to be gained at the instant that throttling starts to occur.

The Guidance problem now becomes one of dividing the available thrust acceleration into its components in a manner which allows the eventual attainment of targeted conditions. From Figure 9 we see that

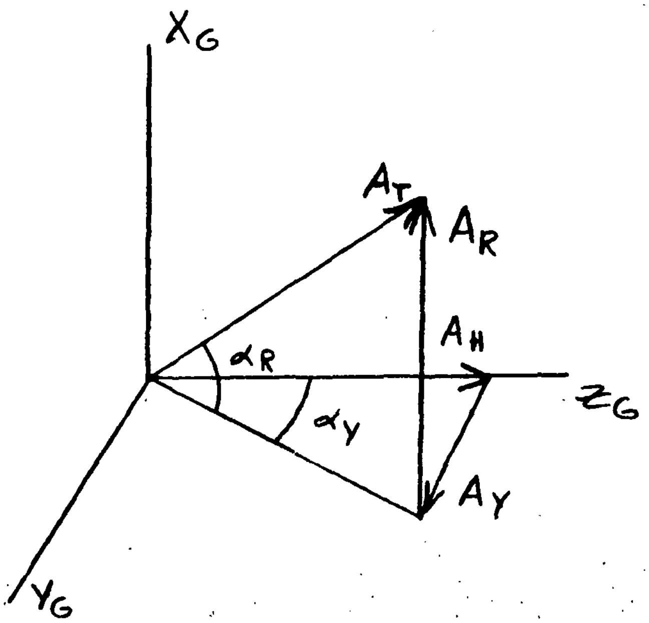


FIGURE 9 DESIRED ACCELERATION COMPONENTS IN PLUMBLINE LAUNCH COORDINATES

5.3.3 E-GUIDANCE (Continued)

$$A_R = A_T \sin \alpha_R$$
$$A_Y = A_T \sin \alpha_Y$$

where A_R and A_Y are in launch guidance plumbline coordinates and A_H is the remaining thrust which is allowed to build up downrange velocity. Then we assume that we can relate the desired acceleration components by

$$A_R = A_T(A + Bt) - g_{eff}$$
$$A_Y = A_T(C + Dt)$$

where A, B, C, and D are linear coefficients and t is interval of time over which the equations are solved. Now by neglecting the gravity term and considering only the A_R equation:

$$\frac{d^2R}{dt^2} = A_T(A + Bt)$$

The first integration yields:

$$\dot{R}(T_{GO}) - \dot{R}(0) = A\int_0^{T_{GO}} A_T(t)\,dt + B\int_0^{T_{GO}} A_T(t)\,t\,dt$$

The second integration yields

$$R(T_{GO}) - R(0) - \dot{R}(0)\,T_{GO} = A\int_0^{T_{GO}}\left[\int_0^t A_T(s)\,ds\right]dt + B\int_0^{T_{GO}}\left[\int_0^t A_T(s)\,s\,ds\right]dt$$

5.3.3. E-GUIDANCE (Continued)

Let use define

$$A_{11} = \int_0^{T_{GO}} A_T(t)\,dt$$

$$A_{12} = \int_0^{T_{GO}} A_T(t)\,t\,dt$$

$$A_{21} = \int_0^{T_{GO}} \left[\int_0^{t} A_T(s)\,ds \right] dt$$

$$A_{22} = \int_0^{T_{GO}} \left[\int_0^{t} A_T(s)\,s\,ds \right] dt$$

We are considering the case where the vehicle is presently in the constant thrust mode and will be eventually in constant acceleration mode. Since A_T is a discontinuous function over time T_{GO}, the integrals must be redefined to be

$$A_{11} = \int_0^{T_{FL}} A_T(t)\,dt + \int_{T_{FL}}^{T_{GO}} A_L\,dt$$

$$A_{12} = \int_0^{T_{FL}} A_T(t)\,t\,dt + \int_{T_{FL}}^{T_{GO}} A_L\,t\,dt$$

$$A_{21} = \int_0^{T_{FL}} \left[\int_0^{t} A_T(s)\,ds \right] dt$$

$$+ \int_{T_{FL}}^{T_{GO}} \left[\int_0^{T_{FL}} A_T(s)\,ds \right] dt + \int_{T_{FL}}^{T_{GO}} \left[\int_{T_{FL}}^{t} A_L\,ds \right] dt$$

$$A_{22} = \int_0^{T_{FL}} \left[\int_0^{t} A_T(s)\,S\,ds \right] dt$$

$$+ \int_{T_{FL}}^{T_{GO}} \left[\int_0^{T_{FL}} A_T(s)\,S\,ds \right] dt + \int_{T_{FL}}^{T_{GO}} \left[\int_{T_{FL}}^{t} A_L\,S\,ds \right] dt$$

where $A_T(S) = \frac{V_e}{\tau - S}$ and A_L is a constant.

5.3.3 E-GUIDANCE (Continued)

The integrals are evaluated to

$$A_{11} = -V_e \ln\left(1 - \frac{T_{FL}}{\tau}\right) + A_L\left(T_{GO} - T_{FL}\right)$$

$$A_{12} = -V_e \tau \ln\left(1 - \frac{T_{FL}}{\tau}\right) - V_e T_{FL} + \frac{1}{2} A_L\left(T_{GO}^2 - T_{FL}^2\right)$$

$$A_{21} = V_e \tau \ln\left(1 - \frac{T_{FL}}{\tau}\right) + V_e T_{FL} - V_e T_{GO} \ln\left(1 - \frac{T_{FL}}{\tau}\right) + \frac{1}{2} A_L\left(T_{GO} - T_{FL}\right)^2$$

$$A_{22} = \tau\left[V_e \tau \ln\left(1 - \frac{T_{FL}}{\tau}\right) + V_e T_{FL} - T_{GO} V_e \ln\left(1 - \frac{T_{FL}}{\tau}\right)\right] + V_e T_{FL}\left(\frac{T_{FL}}{2} - T_{GO}\right) + \frac{A_L}{6}\left(T_{GO}^3 - 3 T_{GO} T_{FL}^2 + 2 T_{FL}^3\right)$$

These equations of the integrals are found in the references and are the forms programmed in LASS.

For the case where the vehicle is in the constant acceleration mode the integral equations are readily simplified to

$$A_{11} = A_L T_{GO}$$

$$A_{12} = \frac{A_L}{2} T_{GO}^2$$

$$A_{21} = A_{12}$$

$$A_{22} = \frac{A_L}{6} T_{GO}^3$$

Where T_{GO} is

$$T_{GO} = \frac{|\bar{V}_G|}{A_L}$$

5.3.3 E-GUIDANCE (Continued)

Consider once again the equations

$$\dot{R}(T_{GO}) - \dot{R}(0) = A_{11} A + A_{12} B$$

$$R(T_{GO}) - R(0) - \dot{R}(0) T_{GO} = A_{21} A + A_{22} B$$

These equations are solved for A and B to obtain

$$A = -A_{12} \frac{\left[R(T_{GO}) - R(0) - \dot{R}(0) T_{GO}\right]}{\Delta} + A_{22} \frac{\left[\dot{R}(T_{GO}) - \dot{R}(0)\right]}{\Delta}$$

$$B = A_{11} \frac{\left[R(T_{GO}) - R(0) - \dot{R}(0) T_{GO}\right]}{\Delta} - A_{21} \frac{\left[\dot{R}(T_{GO}) - \dot{R}(0)\right]}{\Delta}$$

where Δ is the determinent of the coefficients. In a similar manner, for the out of plane component of acceleration,

$$C = -A_{12} \frac{\left[Y(T_{GO}) - Y(0) - \dot{Y}(0) T_{GO}\right]}{\Delta} + A_{22} \frac{\left[\dot{Y}(T_{GO}) - \dot{Y}(0)\right]}{\Delta}$$

$$D = A_{11} \frac{\left[Y(T_{GO}) - Y(0) - \dot{Y}(0) T_{GO}\right]}{\Delta} - A_{21} \frac{\left[\dot{Y}(T_{GO}) - \dot{Y}(0)\right]}{\Delta}$$

The quantities $R(T_{GO})$, $\dot{R}$ (T_{GO}), $Y(T_{GO})$ and $\dot{Y}(T_{GO})$ are the targeted R_D, $\dot{R}_D$, Y_D, and $\dot{Y}_D$ quantities respectively. The terms R(0), $\dot{R}(0)$, Y(0), and $\dot{Y}(0)$ are present condition of the vehicle $|\overline{R}_P|$, $\dot{R}$, $(U_q.R_P)$, and $\dot{Y}$ respectively. The quantity$(\hat{U}_q$, $\overline{R}_P)$is the present vehicle position normal to the orbit plane.

5.3.3 E-GUIDANCE (Continued)

The term $\overline{R}_p$ represents the present vehicle inertial position.

All of the above equations are used to provide the desired acceleration components which are

$$A_R = A_T\,(A + Bt) - g_{eff}$$

$$A_Y = A_T\,(C + Dt)$$

where t is the time interval between solutions of these equations. The remaining thrust acceleration, if any, is used to gain downrange velocity $\dot{Z}$. The remaining component $\ddot{Z}$, is

$$A_H = \left[A_T^{\,2} - A_R^{\,2} - A_Y^{\,2}\right]^{\frac{1}{2}}$$

These accelerations are converted from plumbline coordinates to platform coordinates using the conversion matrix obtained from the unit vectors. The matrix has been described previously and the desired accelerations are

$$\left[\,\overline{A_{DP}}\,\right] = \left[\,\hat{U}_{RP} \;\middle|\; \hat{U}_{YP} \;\middle|\; \hat{U}_{ZP}\,\right] \begin{bmatrix} A_R \\ A_Y \\ A_H \end{bmatrix}$$

Referring to Figure 8 the inertial attitudes are computed in a conventional manner as noted previously. Again, the inertial roll angle is constrained to zero.

5.4 EQUATIONS OF MOTION

Flow chart and symbol definitions of the EOM subroutine are shown in Appendix A. The EOM subroutine is used to simulate the vehicle propulsion system and vehicle relationships in space and executed five times a second. Thus, the EOM provides vehicle attitude rates with respect to body coordinates and inertial attitudes, inertial velocities, and inertial positions with respect to platform coordinates. The EOM subroutine is shown in functional form in Figure 10 and requires the input information shown in Table V.

As shown in Figure 10, the EOM subroutine provides thrust acceleration in two forms. The first is with thrust limiting and the second is without thrust acceleration. The ratio of desired thrust acceleration limit to unlimited thrust acceleration is termed fractional throttle setting (EPT).

The EOM subroutine outputs are shown in Table VI. These outputs are required by AUTOPI or AUTOP2, GUIDAN, AERO, and ABGUID subroutines.

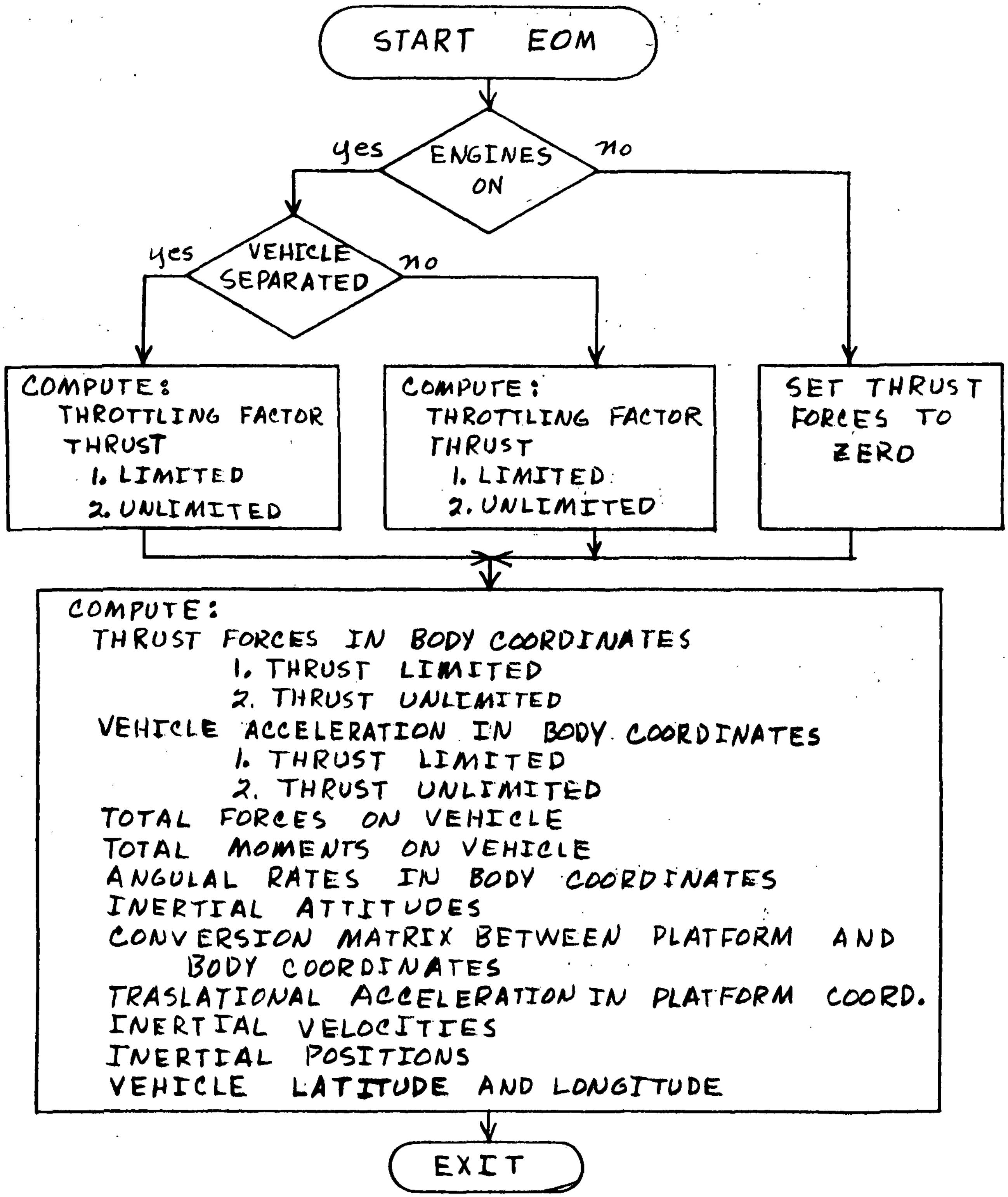


FIGURE 10 FUNCTIONAL FLOW CHART OF EOM SUBROUTINE

TABLE VI INPUT INFORMATION REQUIRED BY THE EOM SUBROUTINES

CODED SYM.	ENGR. SYM.	NOMENCLATURE
AMT	M_T	vehicle mass remaining
AUGRAV	A_g	Acceleration due to gravity
DFAB	F_{AB}	Aerodynamic forces
DMAXB	M_{AXB}	Aerodynamic moment
DMAYB	M_{AYB}	Aerodynamic moment
DMAZB	M_{AZB}	Aerodynamic moment
EAEB	A_{EB}	Booster engine nozzle area
EAEØ	A_{EO}	Orbiter engine nozzle area
EBP(12)	β_P	Engine pitch gimbal angles
EBY(12)	β_Y	Engine yaw gimbal angles
EBPØ	β_{PO}	Engine pitch gimbal angle offset
EL(12, 3)	---	Booster engine locations
ELØ(2,3)	---	Orbiter engine locations
EMXX(AMT)	M_{XX}	Moment of inertia about body X-axis
EMXZ(AMT)	M_{XZ}	Moment of inertia about body XZ plane
EMYY(AMT)	M_{YY}	Moment of inertia about body Y-axis
EMZZ(AMT)	M_{ZZ}	Moment of inertia about body Z-axis
EPA1	P_{SL}	Atmospheric pressure at sea level
ETSLB(12)	T_{SLB}	Booster engine thrust at sea level
ETSLO	T_{SLO}	Orbiter engine thrust at sea level
EXCG(AMT)	X_{CG}	Center of gravity location
EYCG(AMT)	Y_{CG}	Center of gravity location
EZCG(AMT)	Z_{CG}	Center of gravity location

TABLE VI INPUT INFORMATION REQUIRED BY THE EOM SUBROUTINES (Continued)

CODED SYM.	ENGR. SYM.	NOMENCLATURE
IES	---	Engines on
ISEP	---	Vehicle separation
P	P_a	Ambient atmospheric pressure
TIP(3, 3)	---	Relationship between platform and earth inertial coordinates

TABLE VII OUTPUT OF EOM SUBROUTINE

CODED SYM.	ENGR. SYM.	NOMENCLATURE	
ATENG(12)	T_T	Total engine thrust	
ETHETA	θ	Y platform rotation	Euler Angles
EPSI	ψ	Z line of nodes rotation	
EPHI	ϕ	X body rotation	
EPB	P_b	Angular body rate, Roll	
EQB	Q_b	Angular body rate, Pitch	
ERB	R_b	Angular body rate, Yaw	
EPT	---	Fractional throttle setting	
EVP(4)	V_P	Velocity vector in platform coordinates	
ERP(4)	R_P	Position vector in platform coordinates	
EATT	A_T	Thrust acceleration	
EGEFF	---	Magnitude of gravity	
EHV	H_V	Altitude from oblate earth	
ELATC	λ_V	Geocentric latitude	
ELONGC	ϕ_V	Geodetic longitude	
ERF(4)	R_F	Position vector in earth inertial coordinates	

5.4.1 Coordinate Transformations

The EOM subroutine recognizes two basic coordinate systems. These are the platform system and the body system. At the instant of launch, both of these coordinate systems are coincident. The platform coordinate system is described in section 5.2.3. The body coordinate system is vehicle centered. Body X axis points out the nose of the vehicle. Body Y axis points to vehicle right. The Body Z-axis completes the righthanded orthogonal triad and points toward the earth when the vehicle is in level flight. Roll rotation is positive in the counterclockwise sense with right wing down and is a rotation about the body X-axis. Yaw is a rotation about body Z with positive in the sense of nose right. Pitch is a rotation about body Y axis with positive in the sense of nose up.

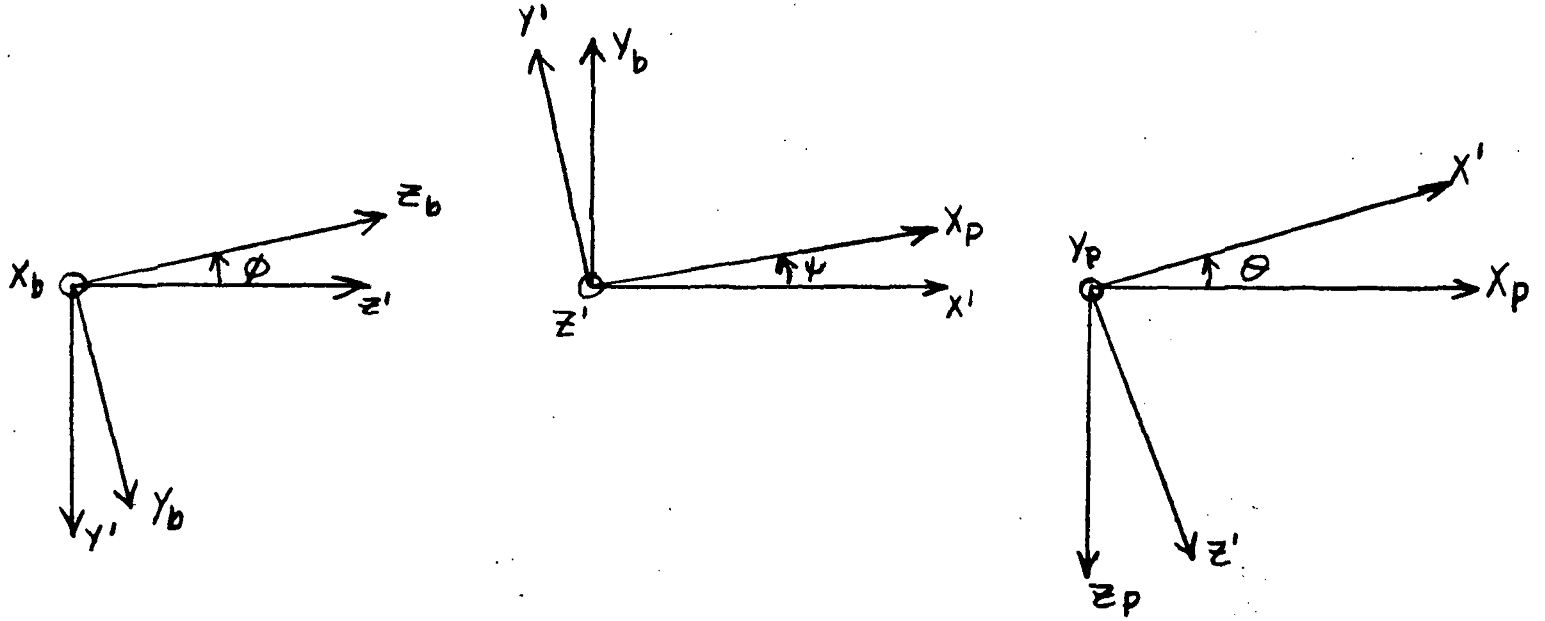
The rotations involved in the transformation from body to platform coordinates is shown in Figure 11. The matrix is constructed in a rotational order of nose up (pitch), nose right (yaw), and rotation about platform X-axis (roll). The resulting matrix is the EB (3, 3) shown in flow chart in Figure A-4. Detailed information concerning the transformation is shown in the references.

5.4.2 Thrust Limiting

In order to obtain a fractional throttle setting, actual vehicle accelerations are computed without thrust limiting. Engine thrust is obtained by summing sea level thrust and altitude effects. This gives:

$$T_{TA} = T_{SL} + A\ (P_{A1} - P)$$

b SUBSCRIPT- BODY AXIS
p SUBSCRIPT- PLATFORM AXIS
θ, ψ, ϕ - EULER ROTATIONS



ROTATION ABOUT THE X_b AXIS

ROTATION ABOUT THE Z' AXIS

ROTATION ABOUT THE Y_p AXIS

FIGURE 11.- RELATION BETWEEN BODY AND PLATFORM COORDINATES

5.4.2 Thrust Limiting (Continued)

where A is engine nozzle exit area, P_{A1} is atmospheric pressure at sea level and P is ambient atmospheric pressure. These are converted to body coordinates using:

$$T_{BXA} = T_{TA} \cos\beta_P \cos\beta_Y$$

$$T_{BYA} = T_{TA} \sin\beta_P$$

$$T_{BZA} = \sin\beta_P \cos\beta_Y$$

where β_P and β_Y are engine pitch and yaw angle respectively. It is recognized that these equations must be summed over all active engines on the vehicle. The resulting forces are summed with the aerodynamic forces and using present vehicle mass the translational accelerations are found.

$$\overline{A_{BA}} = \frac{\overline{T_{BA}} + \overline{F_{AB}}}{M_T}$$

where F_{AB} are aerodynamic forces in body coordinates, M_T is vehicle mass, T_{BA} is unlimited thrust forces in body coordinates, and A_{BA} is unlimited acceleration in body coordinates. This acceleration is converted to units of g's and

$$EPT = \frac{A_L}{|\overline{A_{BA}}|}$$

is the fractional throttle setting, where A_L is the limiting acceleration constraint placed on the vehicle.

5.4.3 Mechanics of Motion

The actual vehicle thrust is provided by:

$$T_T = T_{TA}(EPT)$$

where T_{TA} is unlimited thrust and EPT is the fractional throttle setting. Vehicle acceleration, A_B, in body coordinates, is computed in the same manner as Section 5.4.2, but using the limited thrust terms.

Moments due to thrust are computed by the following equation set:

$$M_{TXB} = \sum_{I=1}^{NE} T_{BZ}(I)\ [L(I,2)]\ -T_{BY}(I)\ [L(I,3)-Z_{CG}]$$

$$M_{TYB} = \sum_{I=1}^{NE} T_{BX}(I)\ [L(I, 3)\ -Z_{CG}]+T_{BZ}(I)[X_{CG} - L(I, 1)]$$

$$M_{TZB} = \sum_{I=1}^{NE} -T_{BX}(I)[\ L(I, 2)] - T_{BY}(I)\ [X_{CG} - L(I, 1)]$$

where:

M_{TXB}, M_{TYB} T_{TZB} = moments about X, Y, and Z axes respectively

T_{BX}, T_{BY}, T_{BZ} = thrust components along X, Y, and Z axes respectively

L(I, J) = engine locations where I is engine number and J = 1, 2, 3 corresponds to distance from X, Y, or Z axis respectively

X_{CG}, Y_{CG}, Z_{CG} = Components of center of gravity position

NE = number of engines

These equations are found in enclosure C of Reference 1.

The angular rates and angular positions are found in conventional manner. Equations (10-11) on page 380 of Reference 2 are linear in $\dot{P}_B$, $\dot{Q}_B$, and $\dot{R}_B$. They are solved simultaneously to yield angular body rates.

$$M = M_{TXB} + M_{AXB} + Q_B R_B \, (I_{YY} - I_{ZZ}) + P_B Q_B I_{XZ} + M_{XRCS}$$

$$N = M_{TZB} + M_{AZB} + P_B Q_B (I_{XX} - I_{YY}) - Q_B R_B I_{XZ} + M_{ZRCS}$$

$$\dot{P}_B = \frac{M I_{ZZ} + N I_{XZ}}{I_{XX} \, I_{ZZ} - I_{XZ}^{\;2}}$$

$$\dot{Q}_B = \frac{M_{TYB} + M_{AYB} + P_B R_B \, (I_{ZZ} - I_{XX}) + (R_B^2 - P_B^2) \, I_{XZ} + M_{YRCS}}{I_{YY}}$$

$$\dot{R}_B = \frac{M I_{XZ} + N I_{XX}}{I_{XX} \, I_{ZZ} - I_{XZ}^{\;2}}$$

where:

$\dot{P}_B$, $\dot{Q}_B$, $\dot{R}_B$ = angular rates about X, Y, and Z axes respectively

I_{AB} = moment of inertia about the A axis due to moments about the B axis;

M_{AXB}, M_{AYB}, M_{AZB} = aerodynamic moments about the X, Y, and Z axes respectively.

M_{XRCS}, M_{YRCS}, M_{ZRCS} = Moments reflecting pseudo RCS function during coasting flight.

5.4.3 Mechanics of Motion (Continued)

Equations 4-103 found on page 134 of Reference 3 are linear in θ, ψ, and ϕ. These are solved simultaneously to yield the euler angle rates expressed by:

$$\dot{\theta} = \frac{Q_B \cos\phi}{\cos\psi} - \frac{R_B \sin\phi}{\cos\psi}$$

$$\dot{\psi} = Q_B \sin\phi + R_B \cos\phi$$

$$\dot{\phi} = P_B - Q_B \tan\psi \ \cos\phi + R_B \tan\psi \ \sin\phi$$

where the righthanded coordinate rotations are:

θ = Y platform rotation

ψ = Z line of nodes rotation

ϕ = X body rotation

P_B, Q_B, and R_B are found by simple integration using:

$$P_B = P_B + .2\ (\dot{P}_B)$$

$$Q_B = Q_B + .2\ (\dot{Q}_B)$$

$$R_B = R_B + .2\ (\dot{R}_B)$$

where .2 represents the rate at which the EOM is executed in simulated time.

5.4.3 Mechanics of Motion (Continued)

The Euler angles are determined by integration of θ, ψ, and ϕ respectively. The angle θ is found by:

$$\theta = \frac{1}{2}\Delta t\left(\dot{\theta}_{LAST} + \dot{\theta}_{PRESENT}\right) + \theta_{LAST}$$

This equation represents a typical angular integration using a trapezoidal technique which assumes a linear change in values. The angle θ can be considered to be represented by some function which is a Taylor series expansion, such that:

$$\theta = \theta_{LAST} + \dot{\theta}\Delta t + \frac{1}{2}\dot{\theta}(\Delta t)^2 + \cdots$$

Only the first two terms are used. The integration technique is demonstrated in Figure 12. The total angle accumulated in Δt seconds corresponds to the shaded area. This area is Δt ($\dot{\theta}_{PAST}$) + ½Δt ($\dot{\theta}_{PRESENT}$ - θ_{PAST}) = ½Δt ($\dot{\theta}_{PAST}$ +$\dot{\theta}_{PRESENT}$). Therefore, $\dot{\theta}$ used in the equation is the average of the present and past angular rates and ΔT is .2 seconds, which is the repetition rate of the EOM subroutine.. The remaining angles Ψ and Ø are found in a similar manner.

If the above procedure is used to predict a future data point, the equation becomes:

$$\dot{X} = \dot{X}_0 + \Delta t(\ddot{X}_{PRESENT}) + \frac{1}{2}\Delta t(\ddot{X}_{PRESENT} - \ddot{X}_{PAST})$$

$$\dot{X} = \dot{X}_0 + \frac{1}{2}\Delta t(3\ddot{X}_{PRESENT} - X_{PAST})$$

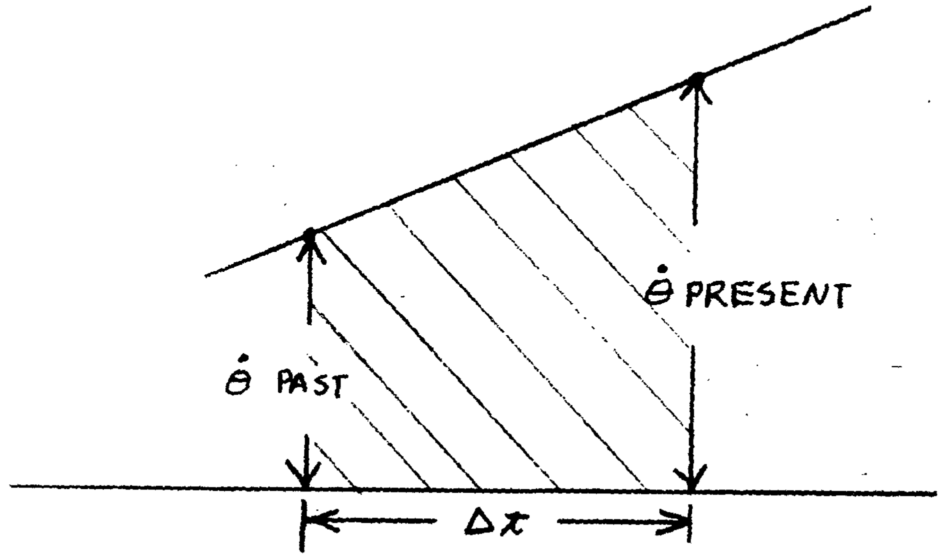


FIGURE 12 TRAPAZOIDAL INTEGRATION

5.4.3 Mechanics of Motion (Continued)

This procedure is accomplished to provide inertial velocity. Since inertial velocity is a function of gravitational acceleration and accelerations due to external forces on the vehicle, the desired Vector is:

$$\overline{V}_P = \overline{V}_{PO} + \frac{1}{2}\Delta t\,(3A_{VP\ PRESENT} - A_{VP\ PAST} + 3G_{PRESNT} - G_{PAST})$$

where Δt is .2 seconds, the cycle time of the EOM subroutine, and G represents gravitational acceleration. Then, $\overline{V}_p$ is a predicted value that determines what the velocity will be at the time position is calculated. Position vector is determined by:

$$\overline{R}_P = \overline{R}_{PO} + \frac{1}{2}\Delta t\,(\overline{V}_{P\ PRESENT} + \overline{V}_{P\ PAST})$$

Vehicle position vector is transformed to earth inertial coordinates by using the TIP transformation provided by the INITIA subroutine. This vector is formed by:

$$\left[\overline{R}_F\right] = \left[TIP\right]^T \left[\overline{R}_P\right]$$

Present vehicle latitude and longitude is:

$$\lambda_V = SIN^{-1}\left[\frac{R_{FZ}}{|\overline{R}_F|}\right]$$

$$\Phi_V = TAN^{-1}\frac{R_{FY}}{R_{FX}} - \omega_e T$$

5.4.3 Mechanics of Motion (Continued)

where ω_e is the earth's angular rotation and T is ground elapsed time since launch. Altitude above an oblate earth is provided in a conventional manner by:

$$R_{EV} = \frac{R_{meq}\,(1 - TF)}{\left[1 - TF(2 - TF)\cos^2\lambda_v\right]^{1/2}}$$

where TF is the earth flattening factor and R_{meq} is the mean earth radius at the equator.

5.5 AERODYNAMIC SUBROUTINE (AERO)

The main functions of the aerodynamic subroutine, AERO, are shown in Figure 13, which also shows three subroutines called by AERO. These subroutines, COEF, COEFO and ATMOS3, are integral parts of AERO and are described in Paragraphs 5.5.3, thru 5.5.5, respectively. Inputs to and outputs from AERO are shown in Tables VIII and IX. Detailed flow charts of AERO are shown in Appendix A, Figure A-5 and a program listing is given in Appendix B.

5.5.1 General Description

The first series of operations performed in the Aerodynamics Subroutine (AERO) erects a thrust plane coordinate system. From this coordinate system, the inertial flight path angle, gamma, is computed. Next, a relative flight path angle, gammar, is computed by accounting for the earth's rotational rate. A transformation matrix relating the body to the platform by the direction cosines is then used to obtain the body velocities. The body velocities are then corrected for earth's rotation to obtain body velocities relative to the air. The body velocities relative to the air are used to compute the angle of attack, Alpha, and the sideslip angle, Beta.

After converting the altitude in feet to altitude in meters, the Atmospheric Subroutine (ATMOS3) is called. The ATMOS3 subroutine provides, as inputs, unitless scale factors as a function of altitude for: the atmospheric pressure, p; the molecular temperature, T_N; the density of the air, ρ; and the speed of sound V_S.

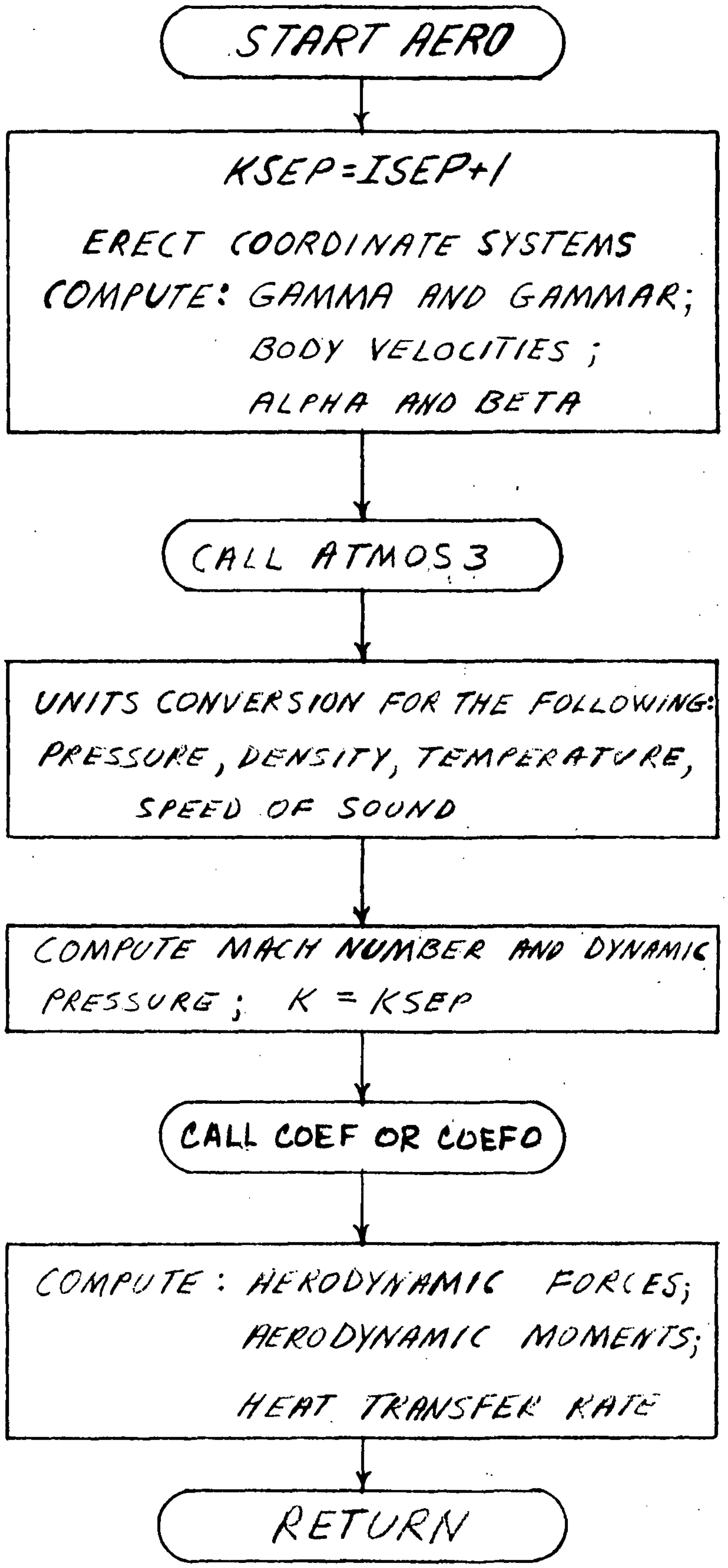


FIGURE 13 FUNCTIONAL DESCRIPTION OF AERO SUBROUTINE

TABLE VIII - INPUTS TO AERO SUBROUTINE

CODED SYM.	ENGR. SYM.	SYMBOL DEFINITION OR USE	UNITS
AMTOT	M_{TOT}	Total mass remaining	slugs
ANSW (8)	$AN_{(I)}$	Ratio inputs from ATMOS3	unitless
EXCG	X_{CG}	X-body c.g. location	ft
EYCG	Y_{CG}	Y-body c.g. location	ft
EZCG	Z_{CG}	Z-body c.g. location	ft
H	h	Vehicle altitude	ft
ISEP	I_{SEP}	Separation discrete	discrete
PITCH	θ	Inertial pitch angle from EOM	rad
Q2 (3)	C_{xo}, C_{yo}, C_{zo}	Aerodynamic force coefficients from COEF	unitless
Q3 (3)	C_{mo}, C_{no}	Aerodynamic moment coefficients from COEF	unitless
ROLL	ϕ	Inertial roll angle from EOM	rad
RP (4)	$\overline{R}_p$	Platform radius vector	ft
VP (4)	$\overline{V}_p$	Platform velocity vector	ft/sec
WEP	$\overline{W}_{EP}$	Earth's rotation vector	ft/sec
YAW	ψ	Inertial yaw angle from EOM	rad

TABLE IX - AERO SUBROUTINE OUTPUTS

CODED SYM.	ENGR. SYM.	SYMBOL DEFINITION OR USE	UNITS
ALPHA	α	Angle of attack	degrees
BETA	β	Sideslip angle	degrees
FAB (4)	$\overline{F}_{AB}$	Aerodynamic body forces	lbs
HM	h_m	Altitude in meters	meters
HMAX	h_{max}	Maximum altitude for ATMOS3 subroutine	meters
K	K	Vehicle configuration	unitless
MAB (4)	M_{AB}	Aerodynamic body moments	ft-lbs
MACH	M	Mach number	unitless
Q1 (3)	M,α,β	Mach number, Alpha, and Beta output to COEF	

5.5.1 General Description (Continued)

After computing the Mach number, M, and the dynamic pressure, q, the Coefficient Subroutines (COEF or COEFO) are called. These subroutines interpolates the values of the aerodynamic coefficients (C_{xo}, C_{yo}, C_{zo}, C_{mo}, and C_{no}) which are functions of Mach number and Alpha or Mach number and Beta. These quantities, along with the dynamic pressure, q; the chord, $\bar{c}$; the span, b; the surface area, S; and the c.g. location (X_{CG}, Y_{CG}, and Z_{CG}), are used to compute the aerodynamic moments acting on the vehicle which are used by EOM. COEF is used prior to separation and COEFO is used after separation.

5.5.2 Detailed Description

The X, Y, and Z c.g. locations (as a function of the mass remaining) are input for use in the moment equations described later. Next, a subscript to identify vehicle configuration for data table use is defined as:

$$K_{SEP} = I_{SEP} + 1$$

In addition, two conversion factors for converting degrees to radians and meters to feet are defined respectively as:

$$F_1 = 57.29578$$

$$F_2 = 0.3048$$

5.5.2 Detailed Description (Continued)

A thrust plane coordinate system is then erected as follows:

$$\hat{U}_{XTP} = \text{UNIT}(\bar{R}_p)$$

$$\hat{U}_{YTP} = \text{UNIT}(\bar{V}_p \text{ X } \bar{R}_p)$$

$$\hat{U}_{ZTP} = \hat{U}_{XTP} \text{ X } \hat{U}_{YTP}$$

where:

$\bar{R}_p$ = platform radius vector

$\bar{V}_p$ = platform velocity vector

The altitude rate, $\dot{h}$, is then obtained from

$$\dot{h} = \bar{V}_p \cdot \hat{U}_{XTP}$$

The magnitude of the down range rate, $\dot{Z}$, is:

$$\dot{Z} = \bar{V}_p \cdot \hat{U}_{ZTP}$$

The inertial flight path angle, γ , is then obtained from:

$$\gamma = \text{TAN}^{-1} \left[\frac{\dot{h}}{\dot{Z}} \right]$$

The relative flight path angle, γ_R, is obtained by subtracting the earth's rotational rate components in the following manner:

$$\dot{h}_{LPP} = \overline{SV}_{LPP} \cdot \hat{U}_{XTP}$$

$$\dot{Z}_{LPP} = \overline{SV}_{LPP} \cdot \hat{U}_{ZTP}$$

$$\dot{h}_e = \dot{h} - \dot{h}_{LPP}$$

$$\dot{Z}_e = \dot{Z} - \dot{Z}_{LPP}$$

$$\gamma_R = \text{TAN}^{-1} \left[\frac{\dot{h}_e}{\dot{Z}_e} \right]$$

where

$\overline{SV}_{LPP}$ = launch pad velocity vector

5.5.2 Detailed Description (Continued)

The vehicle inertial velocity corrected for earth's rotation is:

$$\bar{V}_A = \bar{V}_p - (\bar{W}_{EP} \times \bar{R}_p)$$

where:

$\bar{W}_{EP}$ = earth's rotation vector

The velocity of the vehicle relative to the air, $\bar{V}_{AB}$, is then determined as:

$$\begin{bmatrix} V_{AB1} \\ V_{AB2} \\ V_{AB3} \end{bmatrix} = \begin{bmatrix} C1C2 & S2 & -S1C2 \\ -C1S2C3+S1S3 & C2C3 & S1S2C3+C1S3 \\ C1S2S3+S1C3 & -C2S3 & -S1S2S3+C1C3 \end{bmatrix} \begin{bmatrix} V_{A1} \\ V_{A2} \\ V_{A3} \end{bmatrix}$$

where the transformation matrix terminology has 1, 2, 3 corresponding to θ, ψ, ϕ and C and S stand for cos and sin respectively. The respective values composing the matrixare taken from the EOM subroutine and represents the transpose of the EB(3,3) matrix described in Section 5.4.1. The magnitude of $\bar{V}_{AB}$ is:

$$V_{AB4} = \left[(V_{AB1})^2 + (V_{AB2})^2 + (V_{AB3})^2\right]^{1/2}$$

The angle of attack, α , is then

$$\alpha = \tan^{-1}\left[\frac{V_{AB3}}{V_{AB1}}\right]$$

If the absolute values of $[V_{AB2} \cos(\alpha)]$ and V_{AB1} are greater than zero, the sideslip angle, β , in degrees is determined from:

$$\beta = F_1 \left[\tan^{-1}\left(\frac{V_{AB2} \cos(\alpha)}{V_{AB1}}\right)\right]$$

5.5.2 Detailed Description (Continued)

where

F_1 = conversion factor for radians to degrees

Alpha (α) is then converted to degrees also using the conversion factor F_1.

The altitude in feet is determined from

$$h = R_{p4} - K_5$$

where:

R_{p4} = magnitude of the inertial radius vector

K_5 = earth's radius constant

The altitude is then converted to meters by the conversion factor $F_2 = 0.3048$ for use in the atmospheric subroutine (ATMOS3) which is called next. For continuity, the detailed description of ATMOS3 will be given later and we will proceed with the description of AERO.

The inputs, $AN_{(I)}$, to AERO from ATMOS3 are unitless scale factors as a function of altitude which are used to determine the values for the pressure, temperature, density, and speed of sound in the units being used.

5.5.2 Detailed Description (Continued)

The pressure, p, is determined as:

$$p = AN_1 \left[2116.22\right] \text{ lbs/ft}^2$$

where:

AN_1 = the ratio of the pressure at the present altitude to the pressure at sea level

The temperature, T_N, is determined as:

$$T_N = AN_2 \left[288.16\right] \text{ degrees Kelvin}$$

where:

AN_2 = the ratio of the temperature at the present altitude to the sea level value

The density, ρ, likewise is:

$$\rho = AN_3 \left[2.376885E-3\right] \text{ slugs/ft}^3$$

where:

AN_3 = the density ratio

The speed of sound:

$$V_S = AN_4 \left[1116.45\right] \text{ ft/sec}$$

where:

AN_4 = speed of sound ratio

5.5.2 Detailed Description (Continued)

From which the MACH number is determined as:

$$MACH = V_{AB4}/V_S$$

and the dynamic pressure is found as:

$$q = \frac{\rho\left[V_{AB4}\right]^2}{2} \quad \text{lbs/ft}^2$$

If the magnitude of the vehicle velocity relative to the air (V_{AB4}) is less than 3 ft/sec and the component in the Y-body direction (V_{AB2}) is less than 0.01 ft/sec, the sideslip angle, Beta, is set to zero.

The separation subscript, K, is then set equal to K_{SEP} for use in the coefficient subroutine and subsequent equations. The MACH number, M, the angle of attack, α , and the sideslip angle, β , are then provided as inputs to the coefficient subroutine COEF or COEFO (which is called next). The inputs are commanded through the call routine by:

$$Q1_1 = MACH = A_1$$

$$Q1_2 = \alpha = A_2$$

$$Q1_3 = \beta = A_3$$

Again, for continuity, the COEF subroutine will be described later.

The inputs to AERO from COEF and COEFO are the interpolated values for the aerodynamic coefficients C_{xo}, C_{yo}, C_{zo}, C_{mo}, and C_{no} which are functions of Alpha or Beta and/or Mach number.

5.5.2 Detailed Description (Continued)

A calculation is next made for the total dynamic force, F_D, by:

$$F_D = q \left[S(K)\right]$$

where:

q = dynamic pressure in lbs/ft^2

S = effective aerodynamic surface area of the vehicle

K = subscript denoting vehicle configuration (1 = combined Booster/Orbiter; 2 = Orbiter only)

The aerodynamic forces are then determined as:

$$F_{AB1} = C_{xo} \left[F_D\right]$$

$$F_{AB2} = C_{yo} \left[F_D\right]$$

$$F_{AB3} = C_{zo} \left[F_D\right]$$

The magnitude of $\overline{F}_{AB}$ is then determined as

$$F_{AB4} = \left[F_{AB1}^{\;2} + F_{AB2}^{\;2} + F_{AB3}^{\;2}\right]^{\frac{1}{2}}$$

Next, the aerodynamic moment arms are determined by subtracting the aerodynamic reference coordinates from the c.g. location coordinates as:

$$D_1 = X_{CG} - X_{REF(K)}$$

$$D_2 = Y_{CG} - Y_{REF(K)}$$

$$D_3 = Z_{CG} - Z_{REF(K)}$$

5.5.2 Detailed Description (Continued)

The aerodynamic moments are then computed as:

$$M_{AB1} = qS \left[C_{yo}D_3 - C_{zo}D_2\right]$$

$$M_{AB2} = qS \left[C_{mo}\bar{c}(K) + C_{zo}D_1 - C_{xo}D_3\right]$$

$$M_{AB3} = qS \left[C_{no}b(K) - C_{yo}D_1 + C_{xo}D_2\right]$$

where:

$\bar{c}$ = aerodynamic chord of the vehicle configuration, K

b = aerodynamic span of the vehicle configuration, K

If the magnitude of the body velocity, V_{AB4}, is less than 10 ft/sec, the heat transfer rate, $\dot{Q}$, is set to zero. Otherwise, $\dot{Q}$ is computed in two steps by:

$$QQ1 = 3.15 \left[\log \left(\frac{V_{AB4}}{26000}\right)\right]$$

$$\dot{Q} = 17600 \left[\left[AN_3\right] \; QQ1\right]^{\frac{1}{2}}$$

where:

AN_3 = air density ratio

The other numbers are units conversion values.

5.5.3 Coefficient Subroutine (COEF)

The COEF subroutine uses a linear interpolation of the data tables for the aerodynamic coefficients (C_{xo}, C_{yo}, C_{zo}, C_{mo}, and C_{no}) which are functions of Alpha or Beta and/or Mach number. A flow chart of the subroutine is shown in Appendix A, Figure A-6, and computer listing is shown in Appendix B. A table of booster/orbiter coefficients are shown in Table X.

First, the routine determines a Mach number table value greater than the input Mach number from AERO. A scale factor, SF, is then computed as:

$$SF = \frac{MACH - MCH(I-1)}{MCH(I) - MCH(I-1)}$$

where:

MCH(I) = table value of Mach number greater than present Mach number

MCH(I-1) = table value of Mach number less than present Mach number

The scale factor is, therefore, the ratio of the present excess Mach number over the next lowest Mach number table value to the difference between the next highest Mach number table value and the next lowest Mach number table value.

TABLE X - AERODYNAMICS FOR BOOSTER/ORBITER VEHICLE

REFERENCE 1

		C_y		C_N		C_7		C_M		C_ℓ
MACH	C_X	β=0°	β=1°	β=0°	β=1°	α=0°	α=1°	α=0°	α=1°	β=1°
0	-.320	0	-.023	0	.017	-.020	-.062	-.075	-.117	-.005
.50	-.180	0	-.023	0	.018	-.040	-.067	-.119	-.128	-.005
.60	-.170	0	-.026	0	.018	-.065	-.068	-.120	-.131	-.005
.70	-.160	0	-.030	0	.019	-.090	-.068	-.119	-.132	-.006
.80	-.155	0	.034	0	.020	-.140	-.068	-.117	-.133	-.006
.90	-.160	0	.038	0	.022	-.175	-.067	-.113	-.133	-.007
1.00	-.235	0	-.041	0	.025	-.180	-.068	-.107	-.137	-.008
1.10	-.240	0	-.043	0	.030	-.170	-.071	-.100	-.141	-.008
1.20	-.200	0	-.041	0	.029	-.138	-.069	-.690	-.138	-.007
1.30	-.175	0	-.040	0	.027	-.112	-.062	-.080	-.123	-.007
1.40	-.150	0	-.038	0	.026	.091	-.058	-.070	-.114	-.006
1.50	-.135	0	-.037	0	.025	-.075	-.055	-.062	-.108	-.006
1.75	-.090	0	-.034	0	.022	-.035	-.049	-.045	-.095	-.005
2.00	-.062	0	-.032	0	.021	-.005	-.045	-.032	-.087	-.005
3.00	-.020	0	-.030	0	.018	+.010	-.037	+.005	-.069	-.004
4.00	+.020	0	-.028	0	.016	-.015	-.035	+.010	-.065	-.004
5.00	+.035	0	-.026	0	.014	-.005	-.033	+.010	-.060	-.003
20.00	+.045	0	-.023	0	.012	.000	-.020	+.010	-.036	-.003

5.5.3 Coefficient Subroutine (COEF) (Continued)

The aerodynamic drag force coefficient(C_{xo}) is only a function of Mach number and is computed as:

$$B_1 = C_x(I-1,K) + \left[C_x(I,K) - C_x(I-1,K)\right] SF$$

where:

$B_1 = C_{xo}$

I = i^{th} element of the k^{th} row determined by MCH(I)

K = number of the row as determined by the vehicle configuration (1 = booster, 2 = orbiter)

$C_x(I,K)$ = table value of C_x determined by MCH(I) and vehicle configuration, K

$C_x(I-1,K)$ = table value of the next lower element in the array

The expression for C_{xo} is, therefore, the table value of C_x corresponding to the next lowest table Mach number, MCH(I-1), plus the difference between the C_x values corresponding to MCH(I) and MCH(I-1) multiplied by the scale factor.

The Z-body aerodynamic force coefficient, C_{zo}, is a function of both Mach number and Alpha and is computed in three steps. The first eighteen elements (I=1 to 18) in the 36 table array correspond to the C_z value as a function of Mach number and Alpha equal to 0°. The next eighteen elements (I=19 to 36) in the same row correspond to the C_z values as a function of Mach number with Alpha equal to 1°. First, a linear interpolation is made between the data points for Alpha equal to zero by incorporating the scale factor as follows:

5.5.3 Coefficient Subroutine (COEF) (Continued)

$$QQ1 = C_z\ (I-1,K) + \left[C_z(I,K) - C_z(I-1,K)\right]\ SF$$

where I, K and SF were described earlier. The term QQ1 is just a scratch-pad erasable location.

Next, a linear interpolation is made between the data points as a function of Mach number for Alpha equal to one degree, again by use of the scale factor in the following manner:

$$QQ2 = C_z(I+19,K) + \left[C_z(I+20,K) - C_z(I+19,K)\right]\ SF$$

The third step computes the value of C_{zo} by taking the sum of the value at Alpha equal to zero plus the slope of the curve (QQ2-QQ1) multiplied by Alpha as follows

$$B_3 = QQ1 + A_2\ \left[QQ2\text{-}QQ1\right]$$

where

$$B_3 = C_{zo}$$

$$A_2 = \text{Alpha}$$

The aerodynamic pitching moment coefficient, C_{mo}, is computed similarly. The Y-body aerodynamic force coefficient, C_{yo}, and the aerodynamic rolling moment coefficient, C_{no}, are also computed in a similar manner except that they are functions of Beta and the slope is multiplied by A_3(Beta) instead.

5.5.3 Coefficient Subroutine (COEF) (Continued)

The output quantities of the COEF routine are then:

$B_1 = C_{xo}$ $B_4 = C_M$

$B_2 = C_{yo}$ $B_5 = C_N$

$B_3 = C_{zo}$ $C_1 = C_\ell$

5.5.4 Coefficient Subroutine (COEFO)

The COEFO Subroutine uses a linear interpolation of data tables for the aerodynamic coefficients (C_x, C_y, C_z, C_M, C_N, C_ℓ, $C_{\ell p}$, $C_{\ell r}$, C_{Np}, and C_{Nr}). These coefficients are functions of Alpha or Beta and/or Mach numbers. A flow chart of the subroutine is shown in Appendix A, Figure A-7, and a computer listing is shown in Appendix B. A table of orbiter coefficients is shown in Table XI.

Inputs to the subroutine are through the first argument of the subroutine argument list. These are:

Mach number = A(1)

Angle of attack = A(2)

Angle of
  sideslip = A(3)

TABLE XI
161-C ORBITER AERO COEFFICIENTS

$C_A = -C_X$

REFERENCE 5

$M \setminus \alpha$	0°	5°	10°	15°	20°	30°	45°	60°	90°
0.0	.020	.010	-.014	-.054	-.090	.058	.044	.015	.015
.5	.020	.013	-.012	-.052	-.091	.044	.020	.007	.007
.6	.021	.015	-.011	-.053	-.092	.040	.017	.006	.006
.7	.022	.016	-.009	-.054	-.090	.028	.013	.006	.006
.8	.025	.018	-.007	-.055	-.088	.017	.010	.006	.006
.9	.030	.023	-.005	-.053	-.083	.008	.006	.006	.006
1.0	.040	.034	.010	-.032	-.065	.014	.011	.010	.010
1.1	.052	.047	.025	-.009	-.046	.022	.015	.012	.012
1.2	.060	.056	.041	.010	-.030	.027	.023	.013	.013
1.3	.058	.053	.040	.013	-.023	.019	.022	.012	.012
1.4	.057	.052	.039	.018	-.014	.013	.021	.011	.011
1.5	.055	.049	.038	.021	-.008	.003	.020	.010	.010
1.75	.052	.046	.037	.025	.005	.012	.021	.012	.012
2.0	.050	.044	.036	.029	.019	.021	.021	.014	.014
3.0	.041	.038	.034	.033	.031	.025	.019	.011	.011
4.0	.037	.036	.032	.033	.030	.027	.019	.009	.009
5.0	.035	.033	.031	.029	.028	.024	.017	.008	.008
20.0	.027	.026	.025	.024	.022	.018	.012	.005	.005

TABLE XI (Continued)

C_{Y_β}

α / M	0°	5°	10°	15°	20°	30°	45°	60° & 90°
0.0	- 0.100	- 095	-.097	- 0.100	- 0.120	- 0.203	- 0.125	-.125
.5	- 0.103	- 097	-.101	- 0.105	- 0.122	- 0.175	- 0.120	-.120
.6	- 0.105	- 0.100	-.105	- 0.110	- 0.125	- 0.135	- 0.105	-.105
.7	- 0.110	- 0.107	-.111	- 0.116	- 0.132	- 0.145	- 0.075	-.075
.8	- 0.115	- 0.115	-.119	- 0.123	- 0.139	- 0.162	- 0.068	-.068
.9	- 0.120	- 0.121	-.125	- 0.129	- 0.145	- 0.171	- 0.068	-.068
1.0	- 0.125	- 0.127	-.127	- 0.130	- 0.135	- 0.150	- 0.075	-.074
1.1	- 0.128	- 0.128	-.128	- 0.129	- 0.125	- 0.080	- 0.087	-.087
1.2	- 0.130	- 0.130	-.127	- 0.125	- 0.114	- 0.068	- 0.100	-.100
1.3	- 0.128	- 0.125	-.123	- 0.121	- 0.111	- 0.080	- 0.097	-.097
1.4	- 0.123	- 0.120	-.118	- 0.116	- 0.108	- 0.088	- 0.090	-.090
1.5	- 0.115	- 0.115	-.113	- 0.110	- 0.105	- 0.086	- 0.083	-.083
1.75	- 0.110	- 0.110	-.108	- 0.105	- 0.103	- 0.095	- 0.083	-.083
2.0	- 0.105	- 0.105	-.107	- 0.105	- 0.102	- 0.092	- 0.084	-.084
3.0	- 0.110	- 0.110	-.105	- 0.105	- 0.102	- 0.089	- 0.083	-.083
4.0	- 0.107	- 0.102	-.098	- 0.094	- 0.090	- 0.084	- 0.080	-.094
5.0	- 0.100	- 0.095	-.088	- 0.082	- 0.075	- 0.079	- 0.079	-.093
20.0	- 0.090	- 0.082	-.078	- 0.075	- 0.070	- 0.069	- 0.070	-.085

TABLE XI (Continued)

$C_N = -C_Z$

α / M	0°	5°	10°	15°	20°	30°	45°	60° & 90°
0.0	.02	-.15	-.36	-.55	-.75	-1.21	-1.51	-1.63
0.5	.03	-.14	-.34	-.53	-.73	-1.13	-1.35	-1.43
0.6	.04	-.13	-.34	-.52	-.72	-1.10	-1.31	-1.38
0.7	.05	-.13	-.34	-.52	-.72	-1.11	-1.35	-1.43
0.8	.06	-.12	-.34	-.53	-.73	-1.13	-1.39	-1.49
0.9	.07	-.12	-.34	-.53	-.74	-1.14	-1.43	-1.54
1.0	.07	-.13	-.35	-.56	-.77	-1.17	-1.51	-1.68
1.1	.06	-.14	-.37	-.58	-.79	-1.21	-1.60	-1.83
1.2	.05	-.15	-.38	-.60	-.81	-1.23	-1.68	-1.98
1.3	.05	-.15	-.37	-.57	-.77	-1.18	-1.67	-1.98
1.4	.04	-.15	-.35	-.55	-.75	-1.16	-1.67	-1.99
1.5	.04	-.15	-.34	-.53	-.72	-1.12	-1.67	-2.00
1.75	.04	-.13	-.30	-.47	-.65	-1.03	-1.60	-2.10
2.0	.04	-.11	-.26	-.41	-.57	-.92	-1.52	-2.20
3.0	.03	-.08	-.22	-.34	-.48	-.80	-1.36	-2.00
4.0	.03	-.06	-.18	-.28	-.42	-.73	-1.26	-1.91
5.0	.03	-.04	-.15	-.25	-.37	-.67	-1.23	-1.83
20.0	.02	-.02	-.08	-.15	-.26	-.57	-1.17	-1.78

TABLE XI (Continued)

C_m

M \ α	0°	5°	10°	15°	20°	30°	45°	60° & 90°
0.0	.055	-.032	-.122	-.210	-.308	-.560	-.260	-.260
0.5	.090	-.010	-.110	-.211	-.330	-.575	-.325	-.325
0.6	.115	.002	-.102	-.212	-.342	-.580	-.355	-.355
0.7	.150	.035	-.085	-.209	-.350	-.587	-.395	-.395
0.8	.195	.070	-.068	-.206	-.358	-.594	-.460	-.460
0.9	.257	.102	-.052	-.203	-.375	-.600	-.562	-.562
1.0	.280	.070	-.220	-.320	-.490	-.690	-.750	-.750
1.1	.290	-.030	-.388	-.760	-.880	-.840	-.930	-.930
1.2	.280	-.135	-.555	-.850	-.975	-.935	-1.040	-1.040
1.3	.260	-.140	-.507	-.780	-.920	-.910	-1.030	-1.030
1.4	.230	-.140	-.459	-.670	-.820	-.810	-.940	-.940
1.5	.180	-.130	-.410	-.600	-.700	-.700	-.835	-.835
1.75	.120	-.100	-.310	-.420	-.500	-.520	-.720	-.720
2.0	.085	-.070	-.210	-.295	-.340	-.380	-.640	-1.100
3.0	-.020	-.025	-.033	-.040	-.060	-.165	-.462	-.890
4.0	-.040	-.020	.005	.040	.030	-.070	-.370	-.810
5.0	-.040	-.020	.030	.065	.060	-.010	-.320	-.770
20.0	-.065	-.020	.040	.070	.110	.008	-.240	-.690

TABLE XI (Continued)

$\underline{C_{l_\beta}}$

α / M	0°	5°	10°	15°	20°	30°	(45°, 60° & 90°)
0.0	-.047	-.099	-.151	-.205	-.136	.250	-.165
0.5	-.055	-.115	-.175	-.235	-.198	-.075	-.295
0.6	-.060	-.120	-.180	-.240	-.253	-.150	-.340
0.7	-.070	-.125	-.180	-.235	-.234	-.095	-.310
0.8	-.095	-.138	-.181	-.223	-.217	0.0	-.240
0.9	-.120	-.151	-.182	-.212	-.199	.050	-.210
1.0	-.137	-.159	-.181	-.203	-.208	0.0	-.222
1.1	-.140	-.163	-.186	-.208	-.213	-.100	-.248
1.2	-.138	-.164	-.190	-.215	-.199	-.210	-.270
1.3	-.130	-.154	-.178	-.203	-.183	-.215	-.282
1.4	-.120	-.144	-.168	-.192	-.166	-.200	-.285
1.5	-.103	-.129	-.155	-.182	-.165	-.190	-.280
1.75	-.075	-.103	-.131	-.158	-.165	-.210	-.236
2.0	-.053	-.083	-.113	-.142	-.152	-.220	-.220
3.0	.008	-.032	-.072	-.112	-.135	-.200	-.240
4.0	.014	-.022	-.058	-.095	-.118	-.193	-.252
5.0	0.0	-.030	-.060	-.090	-.119	-.190	-.250
20.0	-.010	-.033	-.046	-.080	-.119	-.180	-.220

TABLE XI (Continued)

C_{l_p}

δ / M	0°	5°	10°	15°	20°	30°	(45°, 60° & 90°)
0.0	-.187	-.190	-.194	-.205	-.216	-.162	-.096
0.5	-.200	-.204	-.210	-.218	-.227	-.162	-.106
0.6	-.205	-.210	-.214	-.224	-.230	-.163	-.110
0.7	-.211	-.216	-.219	-.228	-.235	-.164	-.114
0.8	-.220	-.222	-.225	-.233	-.240	-.166	-.120
0.9	-.230	-.230	-.231	-.238	-.245	-.168	-.130
1.0	-.242	-.240	-.232	-.240	-.240	-.170	-.140
1.1	-.249	-.243	-.237	-.236	-.230	-.173	-.153
1.2	-.250	-.243	-.236	-.230	-.216	-.170	-.166
1.3	-.248	-.236	-.230	-.217	-.205	-.163	-.175
1.4	-.237	-.226	-.216	-.203	-.194	-.152	-.183
1.5	-.220	-.210	-.201	-.192	-.185	-.145	-.188
1.75	-.190	-.180	-.172	-.168	-.165	-.138	-.185
2.0	-.167	-.160	-.157	-.153	-.152	-.134	-.177
3.0	-.125	-.122	-.185	-.118	-.115	-.125	-.162
4.0	-.100	-.094	-.104	-.096	-.092	-.118	-.156
5.0	-.065	-.073	-.090	-.078	-.080	-.116	-.153
20.0	-.040	-.048	[illegible]	-.063	-.070	-.118	-.156

TABLE XI (Continued)

C_{l_r}

α / M	0^0	5^0	10^0	15^0	20^0	30^0	45^0	60^0 & 90^0
0.0	.055	.070	.085	.097	.110	.120	.180	.180
0.5	.054	.071	.087	.097	.105	.118	.170	.170
0.6	.053	.070	.088	.096	.105	.115	.165	.165
0.7	.053	.069	.087	.095	.105	.110	.162	.162
0.8	.054	.070	.086	.095	.106	.100	.160	.160
0.9	.055	.070	.085	.096	.107	.098	.158	.158
1.0	.060	.074	.090	.100	.113	.097	.145	.145
1.1	.070	.082	.095	.111	.127	.100	.135	.135
1.2	.085	.095	.107	.119	.133	.103	.128	.128
1.3	.097	.105	.112	.124	.135	.095	.120	.120
1.4	.104	.109	.114	.124	.133	.078	.110	.110
1.5	.107	.111	.115	.123	.130	.065	.098	.098
1.75	.105	.108	.110	.114	.118	.053	.060	.060
2.0	.095	.095	.095	.095	.095	.043	.045	.045
3.0	.060	.059	.057	.041	.025	.028	.030	.030
4.0	.045	.045	.045	.030	.015	.018	.020	.023
5.0	.045	.044	.040	.026	.010	.013	.016	.018
20.0	.045	.043	.040	.025	.007	.009	.012	.015

TABLE XI (Continued)

C_{n_p}

$M \backslash \alpha$	0^0	5^0	10^0	15^0	20^0	30^0	45^0	60^0 & 90^0
0.0	.078	.059	.041	.058	.075	.087	.044	.044
0.5	.085	.068	.051	.063	.076	.091	.049	.049
0.6	.088	.071	.055	.066	.078	.090	.050	.050
0.7	.092	.075	.059	.070	.082	.078	.051	.051
0.8	.096	.080	.064	.075	.088	.060	.052	.052
0.9	.103	.087	.070	.082	.096	.046	.053	.053
1.0	.113	.094	.078	.089	.102	.045	.049	.049
1.1	.125	.105	.087	.097	.107	.052	.043	.043
1.2	.140	.120	.100	.105	.110	.054	.040	.040
1.3	.150	.129	.109	.111	.112	.046	.041	.041
1.4	.156	.135	.114	.114	.114	.032	.042	.042
1.5	.158	.136	.115	.115	.115	.025	.043	.043
1.75	.150	.132	.113	.112	.110	.020	.023	.023
2.0	.138	.122	.105	.099	.093	.015	.010	.010
3.0	.094	.081	.090	.038	.008	.008	.010	.010
4.0	.070	.062	.052	.029	.005	.008	.012	.015
5.0	.060	.053	.045	.025	.005	.008	.012	.015
20.0	.055	.050	.044	.025	.005	.008	.012	.012

TABLE XI (Continued)

C_{n_r}

α / M	0°	5°	10°	15°	20°	30°	45°	60° & 90°
0.0	-.190	-.200	-.210	-.210	-.210	-.120	-.200	-.200
0.5	-.200	-.210	-.220	-.215	-.210	-.121	-.200	-.200
0.6	-.210	-.220	-.230	-.221	-.213	-.124	-.210	-.210
0.7	-.220	-.232	-.245	-.232	-.220	-.130	-.220	-.220
0.8	-.235	-.250	-.265	-.257	-.250	-.154	-.250	-.250
0.9	-.250	-.270	-.290	-.290	-.290	-.184	-.275	-.275
1.0	-.275	-.297	-.320	-.320	-.320	-.213	-.290	-.290
1.1	-.330	-.342	-.355	-.352	-.350	-.240	-.298	-.298
1.2	-.363	-.364	-.365	-.362	-.360	-.245	-.302	-.302
1.3	-.378	-.373	-.366	-.364	-.363	-.239	-.305	-.305
1.4	-.388	-.378	-.368	-.366	-.364	-.200	-.302	-.302
1.5	-.393	-.379	-.365	-.365	-.365	-.185	-.298	-.298
1.75	-.390	-.376	-.363	-.363	-.363	-.197	-.280	-.280
2.0	-.380	-.367	-.355	-.352	-.350	-.212	-.265	-.265
3.0	-.315	-.309	-.300	-.258	-.213	-.235	-.270	-.270
4.0	-.280	-.276	-.270	-.240	-.210	-.233	-.260	-.290
5.0	-.260	-.255	-.250	-.230	-.210	-.232	-.260	-.290
20.0	-.250	-.245	-.240	-.225	-.210	-.230	-.260	-.290

5.5.4 Coefficient Subroutine (COEFO) (Continued)

Outputs of the subroutine use the second and third arguments of the subroutine argument list. The outputs are:

$B(1) = C_x$

$B(2) = C_y$

$B(3) = C_z$

$B(4) = C_M$

$B(5) = C_N$

$C(1) = C_\ell$

$C(2) = C_{\ell p}$

$C(3) = C_{\ell r}$

$C(4) = C_{Np}$

$C(5) = C_{Nr}$

Interpolation is accomplished in a manner similar to that used in the COEF subroutine. However, the COEFO subroutine uses the Z1 function subroutine to perform the scaling computations used in the interpolations.

5.5.5 Atmosphere Subroutine (ATMOS3)

Figure 14 is a functional description of the atmospheric subroutine, ATMOS3. The model for ATMOS3 was obtained from "U.S. Standard Atmosphere, 1962" (Reference 16).

The outputs, $AN_{(I)}$, of this subroutine are shown in Table XII. These quantities are unitless scale factors of the ratios of the pressure, density, temperature, molecular temperature, speed of sound, acceleration due to gravity, molecular weight, and coefficient of viscosity (all as a function of altitude) to their respective sea level values. As inputs

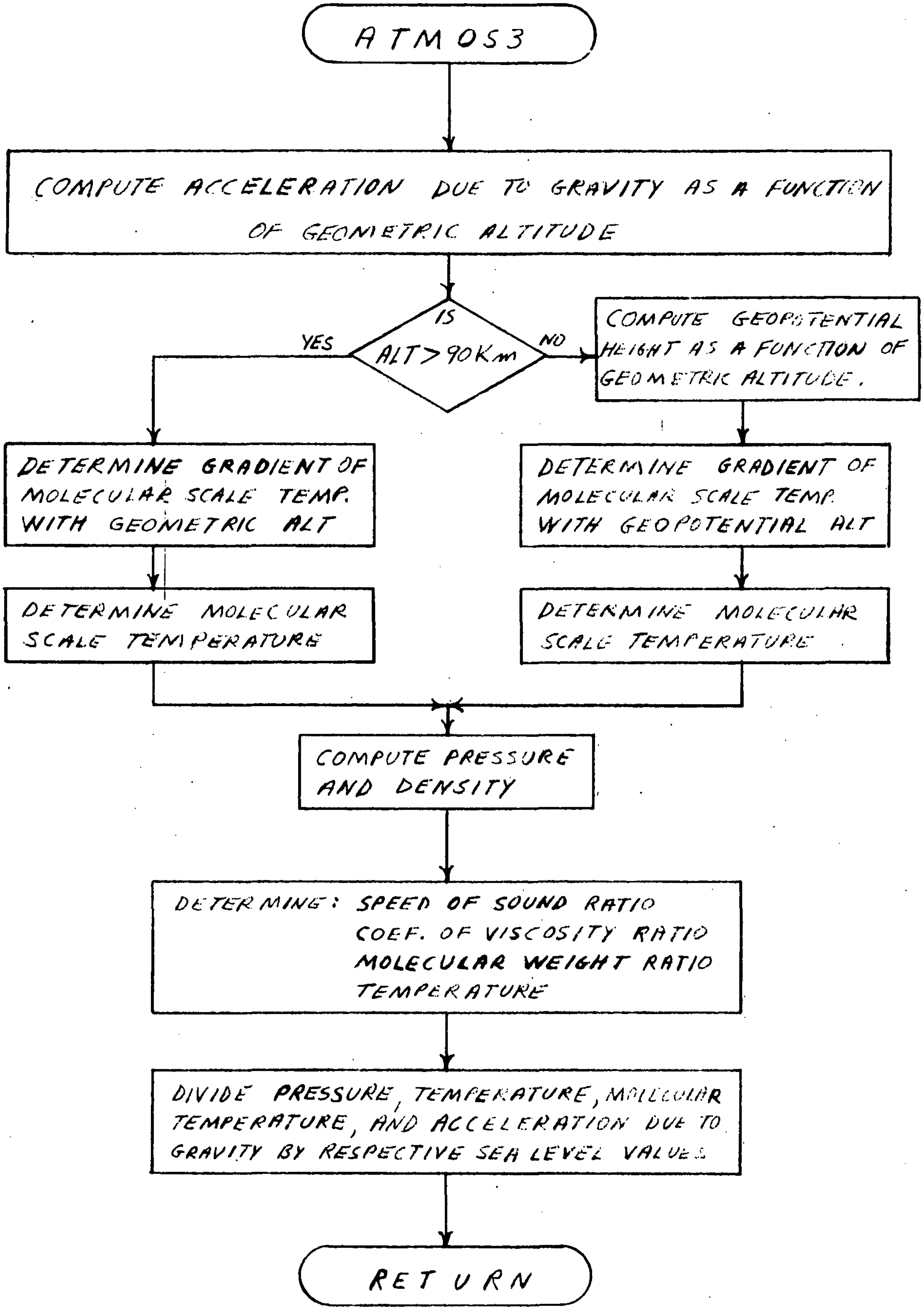

ATMOS3
COMPUTE ACCELERATION DUE TO GRAVITY AS A FUNCTION OF GEOMETRIC ALTITUDE
IS ALT > 90 Km
YES
NO
COMPUTE GEOPOTENTIAL HEIGHT AS A FUNCTION OF GEOMETRIC ALTITUDE.
DETERMINE GRADIENT OF MOLECULAR SCALE TEMP. WITH GEOMETRIC ALT
DETERMINE GRADIENT OF MOLECULAR SCALE TEMP. WITH GEOPOTENTIAL ALT
DETERMINE MOLECULAR SCALE TEMPERATURE
DETERMINE MOLECULAR SCALE TEMPERATURE
COMPUTE PRESSURE AND DENSITY
DETERMINE: SPEED OF SOUND RATIO
COEF. OF VISCOSITY RATIO
MOLECULAR WEIGHT RATIO
TEMPERATURE
DIVIDE PRESSURE, TEMPERATURE, MOLECULAR TEMPERATURE, AND ACCELERATION DUE TO GRAVITY BY RESPECTIVE SEA LEVEL VALUES
RETURN

FIGURE 14 FUNCTIONAL DESCRIPTION OF ATMOS3 SUBROUTINE

TABLE XII- ATMOS3 SUBROUTINE OUTPUTS

CODED SYM.	ENGR. SYM.	SYMBOL DEFINITION OR USE	UNITS
ANSW(1)	AN_1	Ratio of atmospheric pressure to sea level value	unitless
ANSW(2)	AN_2	Ratio of atmospheric temperature to sea level value	unitless
ANSW(3)	AN_3	Ratio of atmospheric density to sea level value	unitless
ANSW(4)	AN_4	Ratio of atmospheric speed of sound to sea level value	unitless
ANSW(5)	AN_5	Ratio of acceleration due to gravity to sea level value	unitless
ANSW(6)	AN_6	Ratio of atmospheric molecular scale temperature to sea level value	unitless
ANSW(7)	AN_7	Ratio of atmospheric molecular weight to sea level value	unitless
ANSW(8)	AN_8	Ratio of atmospheric coefficient of viscosity to sea level value	unitless

5.5.5 Atmosphere Subroutine (ATMOS3) (Continued)

to AERO, these quantities are multiplied by the sea level values in the respective units being used. The inputs to ATMOS3 from AERO are the present altitude in meters and the maximum altitude for which the model may be used (500,000 meters in LASS). All outputs are set to zero above 500,000 meters, except for the speed of sound ratio which is set to 0.791785926 above 90,000 meters.

A detailed flow chart of the subroutine is shown in Appendix A, Figure A-9 and a program listing is given in Appendix B. The input data tables, HB(10), ZB(14), TMB(24), ALP(24), and XGB(14), are shown in the data block provided in Appendix A, Figure A-III.

5.6 MASS CHARACTERISTICS MATH MODEL

The mass characteristics math model is a table look up, based on present vehicle mass, and an interpolation between given table values. The using subroutines of LASS obtain data from the mass characteristics math model using function statements. Whenever a center of gravity displacement value or a moment of inertia value is desired, the presence of the properly argumented symbol causes the execution of the function statement. For example, whenever GUIDAN subroutine, in its execution, encounters the statement EXCG (AMTOT), the program is interrupted until FUNCTION EXCG (EWT) returns the proper data value. It should be noted that EWT, AMT, and AMTOT all represent the present vehicle mass.

Each of the seven function statements;

```
FUNCTION EXCG (EWT)
   "     EYCG (EWT)
   "     EZCG (EWT)
   "     EMXX (EWT)
   "     EMXZ (EWT)
   "     EMYY (EWT)
   "     EMZZ (EWT)
```

in turn call up the FAT subroutine. At the time the FAT subroutine is called up the vehicle mass table, the proper mass characteristics table, and present vehicle mass is communicated as arguments of the subroutine. The mass characteristics data are shown in Table XIII.

TABLE XIII MASS CHARACTERISTICS DATA USED IN LASS

B9T/161-B LAUNCH CONFIGURATION
REFERENCE 1

	Weight (Slugs)	X_{cg} (ft)	Y_{cg} (ft)	Z_{cg} (ft)	I_{xx} (10^6sl-ft^2)	I_{yy} (10^6sl-ft^2)	I_{zz} (10^6sl-ft^2)	I_{xz} (10^6sl-ft^2)
	156875.3	147.48	0	-3.23	38.310	602.627	588.273	-18.652
	139553.3	140.80	0	-3.70	37.552	542.822	528.759	-21.853
	120932.4	134.23	0	-4.19	36.794	483.000	469.225	-25.054
	103033.1	129.93	0	-5.17	35.858	423.213	409.902	-28.843
	85062.7	118.88	0	-6.14	34.852	363.439	358.663	-32.867
	68417.5	114.55	0	-8.09	32.859	304.817	293.603	-35.769
	51750.4	106.96	0	-11.32	29.170	241.190	233.243	-37.336
				161-C ORBITER REFERENCE 6				
Liftoff	26705.9	+53.75	0	+4.99	5.708	34.072	38.631	0.006
	23824.1	+52.17	0	+4.72	5.212	32.889	35.985	0.018
	20942.3	+50.59	0	+4.46	4.717	29.706	33.339	0.030
	18060.6	+49.01	0	+4.19	4.221	27.522	30.693	0.043
	15178.8	+47.43	0	+3.93	3.726	25.339	28.047	0.055
	12297.1	+45.85	0	+3.66	3.230	23.156	25.401	0.067
Burnout	9415.3	+44.28	0	+3.40	2.734	20.972	22.756	0.080

5.6 MASS CHARACTERISTICS MATH MODEL (Continued)

The first function of FAT is to select the mass break points between which the present vehicle mass, EWT, fall. This establishes a value of K used to select the desired tabular data from the selected mass characteristics table. The next function of FAT is to perform the interpolation

$$EXCG = \frac{X(K-1\ [WT(K) - EWT] - X(K)\ [WT(K-1) - EWT]}{WT(K) - WT(K-1)}$$

where: X(K) - K^{th} break point from desired characteristics table

X(K-1) - $(K-1)^{th}$ break point from desired characteristic table

WT(K) - K^{th} break point from mass table

WT(K-1) - $(K-1)^{th}$ break point from mass table

EWT - present vehicle mass

EXCG - desired data point that is returned to the requesting subroutine with the proper symbol identification

Moment and product of inertia data requires a conversion. The function statement converts the data back to proper form by multiplying the result by 10^6 prior to returning data to the requesting subroutine. This conversion accounts for moment and product of inertia data being tabulated in millions of slug - ft.^{2} form in the data tables.

Flow chart and symbol definitions are found in Appendix A.

A listing of LASS is shown in Appendix B.

5.7 IDEAL AUTOPILOT MATH MODEL

5.7.1 Introduction

After determining the differences between the existing inertial attitude angles (EOM) and the desired inertial attitude angles (Guidance), and transforming the inertial differences to body angle differences, the Autopilot model controls the vehicle by: 1) computing angular accelerations necessary to rotate the vehicle about the appropriate axes, 2) computing the necessary thrust moments to create the desired accelerations, and 3) determining the engine gimbal angles in pitch and yaw to cause the required thrust moments.

The desired moments about the vehicle axes computed by the Autopilot may also be used by an RCS system (not included in LASS at this time) for control of the vehicle when the main engines are off.

The engine gimbal angles computed by the Autopilot model are dependent upon knowledge of engine configuration and engine locations. Changes in vehicle configuration and number or location of vehicle engines are readily accounted for by modifying the Autopilot engine gimbal angle equations accordingly.

The autopilot math model consists of two subroutines, AUTOPI and ACC. The detailed description of AUTOPI is given in paragraph 5.7.3. The detailed description of ACC (acceleration subroutine) is given in paragraph 5.7.3.3 as a subparagraph of AUTOPI for continuity. Flow charts, symbol definitions and initialization data for both subroutines are given in Appendix A and the program listing is given in Appendix B.

5.7.2 General Description

The Autopilot model is entered five times per second while Guidance requests new attitude angles only once per second. This necessitates keeping track of the particular pass through the Autopilot and subscripting certain variables appropriately.

The first computations performed in the Autopilot model shown in Fig. 15 are vehicle mass calculations. The Autopilot model computes the desired angular acceleration, thrust moments, and engine gimbal angles for a vehicle whose mass has been decremented for the next 0.2 sec pass. Since the EOM model is performed before the Autopilot model, the vehicle motions as determined by EOM are based upon up to date vehicle mass, vehicle moments of and products of inertia, and engine gimbal angles which have accounted for these changes accordingly.

After updating the vehicle mass each pass (0.2 sec), the Autopilot determines if it is the first pass (Guidance request has been updated). If it is the first pass, the Autopilot determines the differences ($\Delta\theta_D$, $\Delta\psi_D$, $\Delta\phi_D$) between the requested inertial attitude angles (θ_G, ψ_G, ϕ_G) from Guidance and the existing inertial attitude angles (θ_E, ψ_E, ϕ_E) from EOM. The inertial attitude angular differences are then transformed to desired body angular differences by the following equations:

$$\theta_{ERR} = \Delta\theta_D[\cos(\phi_E)][\cos(\psi_E)] + \Delta\psi_D[\sin(\phi_E)]$$

$$\psi_{ERR} = -\Delta\theta_D[\sin(\phi_E)][\cos(\psi_E)] + \Delta\psi_D[\cos(\phi_E)]$$

$$\phi_{ERR} = \Delta\theta_D[\sin(\psi_E)] + \Delta\phi_D$$

$$\Delta\theta_D = \theta_{ERR}$$

$$\Delta\psi_D = \psi_{ERR}$$

$$\Delta\phi_D = \phi_{ERR}$$

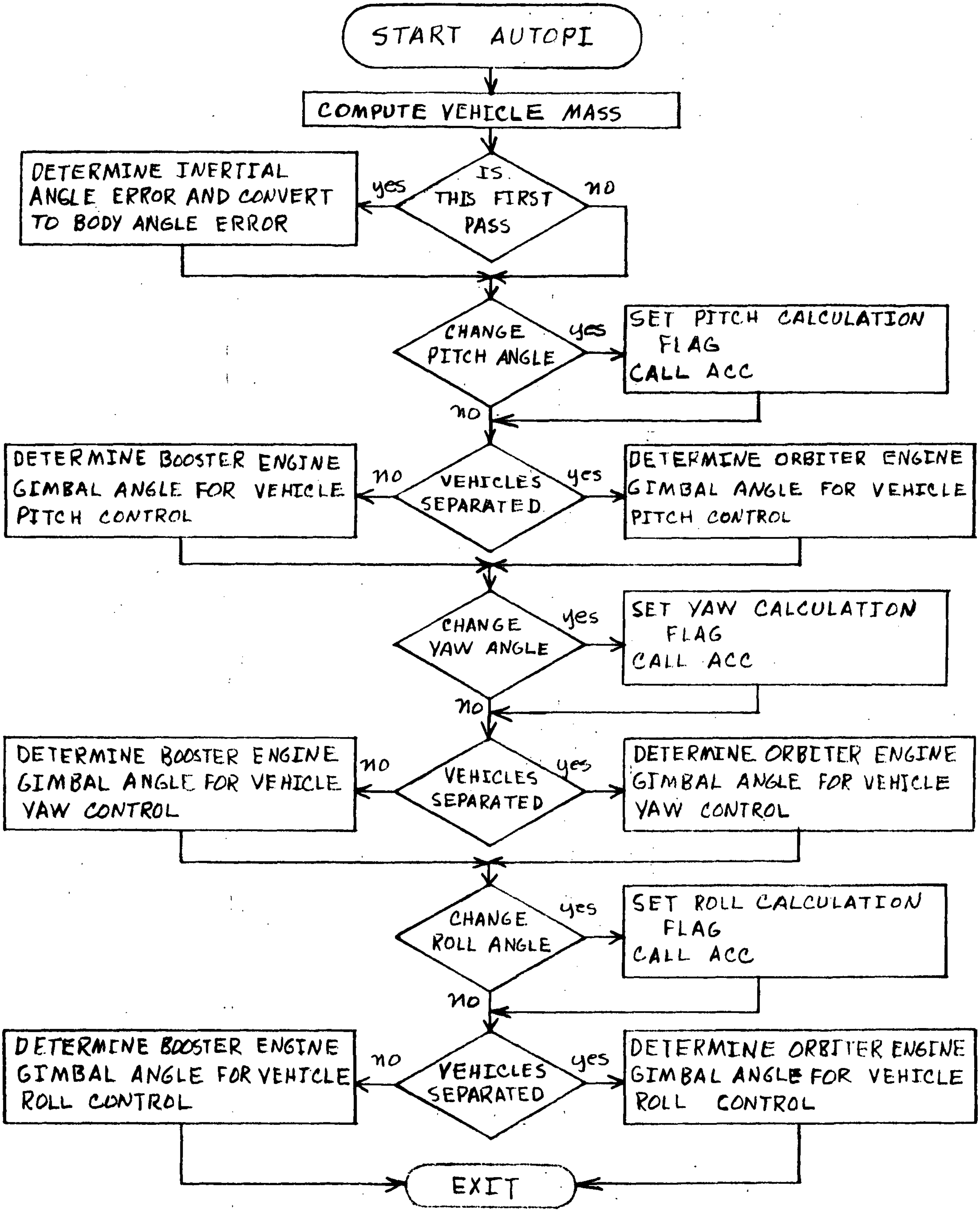

START AUTOPI
COMPUTE VEHICLE MASS
IS THIS FIRST PASS
yes
no
DETERMINE INERTIAL ANGLE ERROR AND CONVERT TO BODY ANGLE ERROR
CHANGE PITCH ANGLE
yes
no
SET PITCH CALCULATION FLAG
CALL ACC
VEHICLES SEPARATED
no
yes
DETERMINE BOOSTER ENGINE GIMBAL ANGLE FOR VEHICLE PITCH CONTROL
DETERMINE ORBITER ENGINE GIMBAL ANGLE FOR VEHICLE PITCH CONTROL
CHANGE YAW ANGLE
yes
no
SET YAW CALCULATION FLAG
CALL ACC
VEHICLES SEPARATED
no
yes
DETERMINE BOOSTER ENGINE GIMBAL ANGLE FOR VEHICLE YAW CONTROL
DETERMINE ORBITER ENGINE GIMBAL ANGLE FOR VEHICLE YAW CONTROL
CHANGE ROLL ANGLE
yes
no
SET ROLL CALCULATION FLAG
CALL ACC
VEHICLES SEPARATED
no
yes
DETERMINE BOOSTER ENGINE GIMBAL ANGLE FOR VEHICLE ROLL CONTROL
DETERMINE ORBITER ENGINE GIMBAL ANGLE FOR VEHICLE ROLL CONTROL
EXIT

FIGURE 15 FUNCTIONAL FLOW CHART FOR AUTOPI SUBROUTINE

5.7.2 General Description (Continued)

The Autopilot then determines the impending increment ($\Delta\theta$, $\Delta\psi$, $\Delta\phi$) between the desired angular changes ($\Delta\theta_D$, $\Delta\psi_D$, $\Delta\phi_D$) and the impending angular changes due to existing body angular rates ($\dot{\theta}_B$, $\dot{\psi}_B$, $\dot{\phi}_B$) at the start of the first pass. On the first pass, the Autopilot also sets to zero the total angular changes accumulated since the start of the first pass ($\Delta\theta_{TOT}$, $\Delta\psi_{TOT}$, $\Delta\phi_{TOT}$) and the angular changes that would result if the vehicle rotation about the particular axis were to be stopped at the end of the pass ($\Delta\theta_{END}$, $\Delta\psi_{END}$, $\Delta\phi_{END}$). If it is not the first pass, all the equations mentioned in this paragraph are by-passed.

The Autopilot then sequentially checks to see if either the body pitch or pitch rate, body yaw or yaw rate, and body roll or roll rate dead-bands has been exceeded. When a dead-band is exceeded, the Acceleration Subroutine ACC (Fig. 16) is called to determine the necessary body angular acceleration to bring about the desired change. If the engines are off, all angles are commanded to zero.

The Acceleration Subroutine (ACC) attempts to attain, in one second, between 95% and 105% of the body angular change desired (up to 3 degrees). Angular acceleration and rotational rate limits are provided. If an angular change greater than three degrees is requested, the vehicle is allowed to coast at or near the rate of 3 degrees per second until the desired angular change is achieved, at which time the vehicle rate is cancelled (or nearly so).

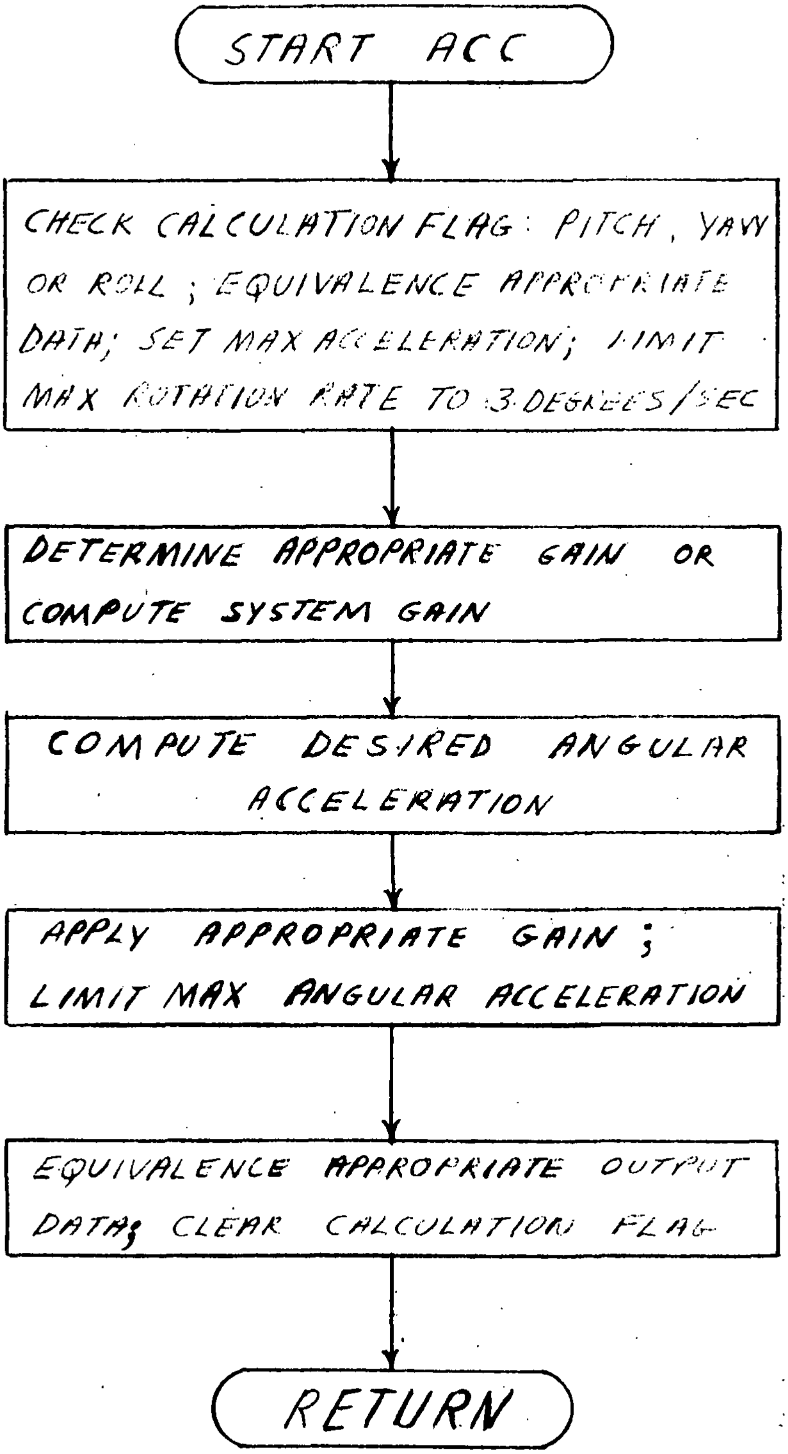

START ACC
CHECK CALCULATION FLAG: PITCH, YAW OR ROLL; EQUIVALENCE APPROPRIATE DATA; SET MAX ACCELERATION; LIMIT MAX ROTATION RATE TO 3 DEGREES/SEC
DETERMINE APPROPRIATE GAIN OR COMPUTE SYSTEM GAIN
COMPUTE DESIRED ANGULAR ACCELERATION
APPLY APPROPRIATE GAIN; LIMIT MAX ANGULAR ACCELERATION
EQUIVALENCE APPROPRIATE OUTPUT DATA; CLEAR CALCULATION FLAG
RETURN

FIGURE 16 FUNCTIONAL FLOW CHART FOR ACC SUBROUTINE

5.7.2 General Description (Continued)

The autopilot then computes the desired moment about the particular axis from:

$$M = I\alpha$$

where

M = Desired moment about the particular axis

I = Moment of inertia about the respective axis

α = Necessary angular acceleration about the axis

The autopilot determines the vehicle configuration (Booster-Orbiter or Orbiter) and, based upon the particular equations used, computes the engine gimbal angles required to create the desired moment. The method of computing the engine gimbal angles is dependent upon the number and location of the engines for the particular vehicle configuration. Roll control is accomplished by gimbaling the engines on one side of the body X-Z plane to a pitch angle equal and opposite to the engines on the other side of the body X-Z plane, forming a couple. The total pitch angle is the sum of the angles required for pitch and roll. Roll control for the Orbiter configuration is accomplished by gimbaling one engine to a yaw angle equal and opposite to that of the other engine, again forming a couple. The total yaw angle is the sum of the angle required for yaw and the angle required for roll.

5.7.3 AUTOPI Detailed Description

5.7.3.1 Mass Calculations

Based upon whether the separation flag (I_{SEP}) has been set or not, the Autopilot sets the number of engines (i) to i_B for the Booster ($i_B = 12$) or to i_0 for the Orbiter ($i_0 = 2$). The mass flow rate ($\dot{W}$) for the type of vehicle engine is determined at the same time, however the same quantity is used for $\dot{W}_B$ and $\dot{W}_0$ in LASS at this time. Tables XIV and XV identify the inputs and outputs,respectively, for the AUTOPI subroutine.

Each time the Autopilot is entered (5 times per sec.), the total change in vehicle mass, ΔM_{TOT}, and the total thrust, T_T are set to zero. The change in vehicle mass per engine is then computed as:

$$\Delta M_i = \dot{W}\left[P_T\right]\left[P_{Ei}\right] t_C$$

where

$\dot{W}$ = mass flow rate per engine

P_T = percent thrust

P_{Ei} = percent thrust due to engine out

t_C = minor cycle time of 0.2 seconds

The total change in mass for the 0.2 second pass is then

$$\Delta M_{TOT} = \Sigma \Delta M_i$$

The total vehicle mass remaining is:

$$M_{TOT} = M_{TOT} - \Delta M_{TOT}$$

If M_{TOT} is less than the mass cutoff value, M_{CO}, the vehicle engines are commanded off ($I_{ES} = 0$).

TABLE XIV INPUTS TO AUTOPI SUBROUTINE

CODED SYM.	ENG. SYMBOL	ORIGIN OF DATA	NOMENCLATURE
ACTPAC	$\alpha_{\theta ACT}$	ACC	Print input of actual pitch acceleration
ACTRAC	$\alpha_{\phi ACT}$	ACC	Print input of actual roll acceleration
ACTYAC	$\alpha_{\psi ACT}$	ACC	Print input of actual yaw acceleration
AGAINP	G_P	ACC	Print input of pitch system gain
AGAINR	G_R	ACC	Print input of roll system gain
AGAINY	G_Y	ACC	Print input of yaw system gain
ALPHPI	$\alpha_{\theta COM}$	ACC	Desired pitch acceleration commanded
ALPHRO	$\alpha_{\phi COM}$	ACC	Desired roll acceleration commanded
ALPHYA	$\alpha_{\psi COM}$	ACC	Desired yaw acceleration commanded
ATENG	T	EOM	Thrust per engine
EMXX	I_{XX}	FUNCTION	Moment of inertia about body X-axis
EMYY	I_{YY}	FUNCTION	Moment of inertia about body Y-axis
EMZZ	I_{ZZ}	FUNCTION	Moment of inertia about body Z-axis
EPB	$\dot{\phi}_B$	EOM	Body roll rate
EPE	P_E	EOM	Percent thrust due to engine out
EPHI	ϕ_E	EOM	Inertial roll angle
EPSI	ψ_E	EOM	Inertial yaw angle
EPT	P_T	EOM	Percent engine thrust
EQB	$\dot{\theta}_B$	EOM	Body pitch rate
ERB	$\dot{\psi}_B$	EOM	Body yaw rate
ETHETA	θ_E	EOM	Inertial pitch angle

TABLE XIV (Continued)

CODED SYM.	ENG. SYMBOL	ORIGIN OF DATA	NOMENCLATURE
EXCG	X_{CG}	FUNCTION	Distance from engine gimbal plane to c.g.
EYCG	Y_{CG}	FUNCTION	Distance from X-Z plane of c.g.
EZCG	Z_{CG}	FUNCTION	Distance from X-Y plane of c.g.
GPHID	ϕ_G	GUIDANCE	Desired inertial roll angle
GPSID	ψ_G	GUIDANCE	Desired inertial yaw angle
GTHED	θ_G	GUIDANCE	Desired inertial pitch angle
ISEP	I_{SEP}	MAIN	Separation flag
J	J	MAIN	Pass counter
KP	KP	MAIN	Data print flag

TABLE XV AUTOPI SUBROUTINE OUTPUTS

CODED SYM.	ENG. SYMBOL	DESTINATION OF DATA	NOMENCLATURE
ADINCP	$\Delta\theta$	ACC	Impending incremental pitch angle
ADINCR	$\Delta\phi$	ACC	Impending incremental roll angle
ADINCY	$\Delta\psi$	ACC	Impending incremental yaw angle
ADMASS	ΔM	PRINT	Mass depleted per eng. per 0.2 sec
ADMTOT	ΔM_{TOT}	PRINT	Total mass depleted per 0.2 sec
ADPDES	$\Delta\theta_D$	ACC	Desired pitch angle change
ADPEND	$\Delta\theta_{END}$	ACC	Pitch angle accumulated if vehicle is stopped
ADPTOT	$\Delta\theta_{TOT}$	ACC	Pitch angle accumulated since start of 1 sec pass
ADRDES	$\Delta\phi_D$	ACC	Desired roll angle change
ADREND	$\Delta\phi_{END}$	ACC	Roll angle accumulated if vehicle is stopped
ADRTOT	$\Delta\psi_{TOT}$	ACC	Roll angle accumulated since start of 1 sec pass
ADYDES	$\Delta\psi_D$	ACC	Desired yaw angle change
ADYEND	$\Delta\psi_{END}$	ACC	Yaw angle accumulated if vehicle is stopped
ADYTOT	$\Delta\psi_{TOT}$	ACC	Yaw angle accumulated since start of 1 sec pass
AMTOT	M_{TOT}	All except ATMOS3, COEF, & UTILITY	Total mass remaining
AOMEGP	ω_P	ACC	Desired pitch rate at start of 1 sec pass
AOMEGR	ω_R	ACC	Desired roll rate at start of 1 sec pass
AOMEGY	ω_Y	ACC	Desired yaw rate at start of 1 sec pass
APRATE	$\dot{\theta}_B$	ACC	Body pitch rate
ARRATE	$\dot{\phi}_B$	ACC	Body roll rate

TABLE XV (Continued)

CODED SYM.	ENG. SYMBOL	DESTINATION OF DATA	NOMENCLATURE
AYRATE	$\dot{\psi}_B$	ACC	Body yaw rate
EBP	θ_{GIM}	EOM	Engine pitch gimbal angle
EBY	ψ_{GIM}	EOM	Engine yaw gimbal angle
IPITCH	I_θ	ACC	Pitch acceleration calculation flag
IRESID	I_R	ACC	Residual acceleration calculation flag
IROLL	I_ϕ	ACC	Roll acceleration calculation flag
IYAW	I_ψ	ACC	Yaw acceleration calculation flag

5.7.3.1 Mass Calculations (Continued)

The total thrust for all engines is summed from:

$$T_T = \Sigma T_i$$

where

T_i = thrust per engine from EOM

If the total thrust is less than some minimum value ($T_{MIN\ B}$ and $T_{MIN\ O}$), the engine gimbal angles are set to zero.

5.7.3.2 Inputs to Angular Acceleration Subroutine (ACC)

Since the angular acceleration computations are identical for all three axes, only a detailed description of the inputs for a desired angular acceleration about the Y-body axis (pitch) will be presented. Table XVI identifies the inputs to the ACC subroutine.

The Autopilot model determines the desired inertial angular change in pitch (θ) during the first pass from:

$$\Delta\theta_D = \theta_G - \theta_E$$

where

$\Delta\theta_D$ = desired angular change

θ_G = desired inertial angular position from Guidance

θ_E = existing inertial angular position from EOM

The desired inertial angular changes in yaw (ψ) and roll (ϕ) are determined similarly and are used in the transformation to desired body angular changes as shown in Section 5.7.2.

TABLE XVI INPUTS TO ACC SUBROUTINE

CODED SYM.	ENG. SYMBOL	ORIGIN OF DATA	NOMENCLATURE
ADINCP	$\Delta\theta$	AUTOPILOT	Impending incremental pitch angle
ADINCR	$\Delta\phi$	AUTOPILOT	Impending incremental roll angle
ADINCY	$\Phi\psi$	AUTOPILOT	Impending incremental yaw angle
ADPDES	$\Delta\theta_D$	AUTOPILOT	Desired pitch angle change
ADPEND	$\Delta\theta_{END}$	AUTOPILOT	Pitch angle accumulated if vehicle is stopped
ADPTOT	$\Delta\theta_{TOT}$	AUTOPILOT	Pitch angle accumulated since start of 1 sec pass
ADRDES	$\Delta\phi_D$	AUTOPILOT	Desired roll angle change
ADPEND	$\Delta\phi_{END}$	AUTOPILOT	Roll angle accumulated if vehicle is stopped
ADPTOT	$\Delta\phi_{TOT}$	AUTOPILOT	Roll angle accumulated since start of 1 sec pass
ADYDES	$\Delta\psi_D$	AUTOPILOT	Desired yaw angle change
ADYEND	$\Delta\psi_{END}$	AUTOPILOT	Yaw angle accumulated if vehicle is stopped
ADYTOT	$\Delta\psi_{TOT}$	AUTOPILOT	Yaw angle accumulated since start of 1 sec pass
AOMEGP	ω_P	AUTOPILOT	Desired pitch rate at start of 1 sec pass
AOMEGR	ω_R	AUTOPILOT	Desired roll rate at start of 1 sec pass
AOMEGY	ω_Y	AUTOPILOT	Desired yaw rate at start of 1 sec pass
APRATE	$\dot{\theta}_B$	AUTOPILOT	Body pitch rate
ARRATE	$\dot{\phi}_B$	AUTOPILOT	Body roll rate
AYRATE	$\dot{\psi}_B$	AUTOPILOT	Body yaw rate
IPITCH	I_θ	AUTOPILOT	Pitch acceleration calculation flag

TABLE XVI (Continued)

CODED SYM.	ENG. SYMBOL	ORIGIN OF DATA	NOMENCLATURE
IRESID	I_R	AUTOPILOT	Residual acceleration calculation flag
IROLL	I_ϕ	AUTOPILOT	Roll acceleration calculation flag
IYAW	I_ψ	AUTOPILOT	Yaw acceleration calculation flag
J	J	MAIN	Pass counter
P	p	AERO	Atmospheric pressure

5.7.3.2 Inputs to Angular Acceleration Subroutines (ACC) (Continued)

The impending angular increment (error) that would result if the vehicle were allowed to coast at its present body rate is also determined on the first pass from :

$$\Delta\theta = \Delta\theta_D - \dot{\theta}_B \left[T_{MAX}\right]$$

where

$\dot{\theta}_B$ = existing pitch body rate from EOM

T_{MAX} = major computation cycle of 1 second

This quantity will be used by the ACC Subroutine to determine if overshoot is impending and to compensate accordingly.

Two quantities for pitch ($\Delta\theta_{TOT}$ and $\Delta\theta_{END}$) are also set to zero at the beginning of the first pass and will be discussed later.

The functions mentioned above are performed only on the first pass through the Autopilot and are by-passed on the next four successive passes of the major cycle time of one second. The following functions are performed during all five passes (again describing only the pitch control).

The present body pitch rate, $\dot{o}(IJ)$, used in the Autopilot and ACC, is set equal to the present body pitch rate, $\dot{\theta}_B$, as determined by EOM. The number of the pass through the Autopilot is identified by (IJ). The pitch calculation flag, I_θ, is set to zero and a check is then made to determine if the absolute value of the desired angular change in pitch, $\Delta\theta_D$, is greater than

5.7.3.2 Inputs to Angular Acceleration Subroutine (ACC) (Continued)

pitch deadband angle, θ_{DB}. If the desired angular change in pitch is greater than the deadband, the pitch calculation flag, I_{θ}, is set. If the angle deadband, θ_{DB}, is not exceeded, a check is made to see if the pitch rate deadband, θ_{DB}, is exceeded. (It is possible for the desired angular change, $\Delta\theta_{D}$, to be less than the angle deadband upon entering the Autopilot on the first pass and yet have a significant residual rate). If the pitch rate deadband, θ_{DB}, is not exceeded, the desired body pitch angular acceleration, $\alpha_{\theta COM}(IJ)$, is set to zero for this pass. The acceleration subroutine is then by-passed and the desired moment about the Y-body axis becomes zero (from $M = I\alpha$) for this pass. The appropriate thrust moment equations are then selected, based upon the state of the separation flag, I_{SEP}, from which the desired engine pitch gimbal angle is calculated as described in detail in the Engine Gimbal Model (Section 5.7.3.4).

If the pitch calculation flag (I_{θ}) is set, AUTOPI calls the Acceleration Subroutine (ACC) and equivalences the desired angular change, Δ_{D}, the impending increment, Δ, the total angle accumulated, Δ_{TOT}, the angle that would be accumulated at the end of this pass, Δ_{END}, the desired angular rate, ω , and the gain, G, to the corresponding pitch (θ) variables. After the acceleration subroutine computes the desired angular acceleration, AUTOPI computes the desired moment and the necessary pitch gimbal angle to create the desired moment as described in Section 5.7.3.4.

5.7.3.3 Acceleration Subroutine (ACC)

Control of the vehicle attitude is accomplished by computing the desired angular accelerations about the respective axes for pitch, yaw, and roll. Table XVII lists the outputs of the ACC Subroutine. The computations are identical for all axes; therefore, only a detailed description for pitch angular acceleration will be presented. Since the Autopilot is entered five times per sec, the acceleration subroutine may be entered 3 times each minor cycle (0.2 sec), once each time either the I_θ, I_ψ, or I_ϕ flag is set. A pass counter is therefore maintained equal to the pass as determined by the Main Program.

The ACC subroutine sets the pass counter IJ = J and then determines which calculation flag has been set by the Autopilot, I_θ, I_ψ, or I_ϕ. When the pitch calculation flag, I_θ, is set, the five quantities for the body angular rates, the angular accelerations commanded, and the actual angular accelerations experienced ($\dot{\Delta}(IJ)$, $\alpha_{COM}(IJ)$, and $\alpha_{ACT}(IJ)$, respectively) used by the acceleration subroutine, are set equal to the last five computed values for each of the respective pitch variables: $\dot{\theta}(IJ)$,$\alpha_{\theta COM}(IJ)$, and $\alpha_{\theta ACT}(IJ)$. (The corresponding yaw or roll values would be equivalenced to the I_ψ or I_ϕ flag had been set.

TABLE XVII ACC SUBROUTINE OUTPUTS

CODED SYM.	ENG. SYMBOL	DESTINATION OF DATA	NOMENCLATURE
ACTPAC	$\alpha_{\theta ACT}$	AUTOPILOT	Print output of actual pitch acceleration
ACTRAC	$\alpha_{\phi ACT}$	AUTOPILOT	Print output of actual roll acceleration
ACTYAC	$\alpha_{\psi ACT}$	AUTOPILOT	Print output of actual yaw acceleration
ALPHPI	$\alpha\theta_{COM}$	AUTOPILOT	Desired pitch acceleration command
ALPHRO	$\alpha_{\phi COM}$	AUTOPILOT	Desired roll acceleration command
ALPHYA	$\alpha_{\psi COM}$	AUTOPILOT	Desired yaw acceleration command

5.7.3.3 Acceleration Subroutine (ACC) (Continued)

The next check made by ACC is to determine if the atmospheric pressure (computed by ATMOS3) is less than 0.75 lbs. ft.^{2}. If it is, the pitch angular acceleration is limited to 7°/sec^2 rather than the initialized value of 10°/sec^2. The desired pitch rotational rate, ω , is determined from:

$$\omega = \frac{\Delta_D}{T_{MAX}}$$

where

ω = the desired pitch angular rate, ω_θ

Δ_D = the desired pitch angular change, $\Delta\theta_D$

T_{MAX} = one second major cycle

If the desired rotational rate is greater than the rate limit, ω_{LIM} (3°/sec), the rotational rate is set equal to the rate limit with the appropriate sign. The acceleration subroutine then tests to see if it is the first pass through the Autopilot. If it is, the actual average angular acceleration experienced is determined by:

$$\alpha_{ACT}(IJ) = \frac{[\dot{\Delta}(IJ) - \dot{\Delta}(IJ + 4)]}{t_c}$$

5.7.3.3 Acceleration Subroutine (ACC) (Continued)

Where:

$\dot{\Delta}(IJ)$ = present pitch body angular rate, $\dot{\theta}(IJ)$

$\dot{\Delta}(IJ + 4)$ = pitch body angular rate during the last 0.2 second cycle, $\dot{\theta}(IJ + 4)$

t_c = time interval of minor cycle, 0.2 seconds

The acceleration subroutine then determines an appropriate gain constant, G, for the overall system. It first checks to see if the last commanded angular acceleration, $\alpha_{COM}(IJ + 4)$ was zero by testing against a constant,α_{CSM},of very small magnitude. If the magnitude of $_{COM}(IJ + 4)$ is less than α_{CSM}, G is set ot $G_1 = (0.9)$. (Although the angular acceleration commanded may have been zero, the vehicle may experience some angular acceleration due to aerodynamics). Selective application of the gain to the new angular acceleration to be commanded will aid in compensating for the aerodynamic moments. The method of applying the gain is described later.

If the magnitude of the last commanded angular acceleration is greater than α_{CSM}, the acceleration subroutine checks to see if the actual angular acceleration experienced, $\alpha_{ACT}(IJ)$, is in the same direction as that of the commanded acceleration on the last pass. (An opposing aerodynamic moment could be sufficiently large enough to cause the vehicle to accelerate in the opposite direction to that desired.) If the angular acceleration is

5.7.3.3 Acceleration Subroutine (ACC) (Continued)

opposite to that desired, the gain, G, is set equal to G_2 (0.7) for division into the new angular acceleration to be computed on this pass. If the actual angular acceleration is in the desired direction, a test is made to see if the actual angular acceleration is greater than 1.67 times the desired angular acceleration. (The unknown aerodynamic moment could just as well be in the direction to aid the desired angular acceleration resulting in a gain very much larger than 1.) If the actual angular acceleration is greater than 1.67 (arbitrarily selected constant) times the desired, G is set to G_3(0.9), If it is not greater than 1.67 times the last commanded acceleration, the gain, G, is computed from:

$$G = \frac{\alpha_{ACT}(IJ)}{\alpha_{COM}(IJ + 4)}$$

where: $\alpha_{ACT}(IJ)$ = actual pitch angular acceleration experienced, $\alpha_{\theta ACT}(IJ)$

$\alpha_{COM}(IJ+4)$= pitch angular acceleration commanded on the last pass, $\alpha_{\theta COM}(IJ+4)$

The gain, as determined in the above equation, is then checked to see if it is less than the minimum gain allowed (G_{MIN} = 0.35 arbitrarily selected). If it is, the gain (G) is set equal to G_{MIN}. (Application of smaller gains to the new angular acceleration to be computed could result in very large gains on the next pass. Again, the method of applying the gain will be described later).

The next calculation made on the first pass is that of Δ_{END}, the total pitch angular change that would be accumulated if the vehicle were to be

5.7.3.3 Acceleration Subroutine (ACC) (Continued)

stopped at the end of this 0.2 sec cycle, due to the present pitch rotational rate, $\dot{\Delta}(IJ)$. The equation for Δ_{END} is:

$$\Delta_{END} = \Delta_{TOT} + \frac{t_c\,[\dot{\Delta}(IJ)]}{2}$$

where: Δ_{TOT} = total accumulated pitch angular change since the start of the 1 sec major cycle (Δ_{TOT} is zero on first pass)

$\dot{\Delta}(IJ)$ = pitch angular rate this pass, $\dot{\theta}(IJ)$

t_c = 0.2 second minor cycle.

On the 2nd, 3rd, 4th, and 5th pass through the Autopilot, the calculations for $\alpha_{ACT}(IJ)$ and the gain, G, differ only in the substitution of (IJ-1) for(IJ + 4). On these succeeding passes, a value for total angle change accumulated since the first pass, Δ_{TOT}, is computed (prior to the computation for Δ_{END}) from:

$$\Delta_{TOT} = \Delta_{TOT} + \alpha_{ACT}(IJ)\left[\frac{t_c^{\,2}}{2}\right] + \dot{\Delta}(IJ-1)\left[t_c\right]$$

where: $\dot{\Delta}(IJ-1)$ = pitch angular rate, $\dot{\theta}(IJ-1)$, at the start of the last pass.

The other terms are as previously defined.

5.7.3.3.1 Acceleration Computations

The acceleration subroutine then checks to see if the residual flag, I_R, is set. If it is, the desired pitch angular acceleration (unadjusted for system gain) that is needed to cancel the pitch body rate at the end of this minor cycle is determined as:

$$\alpha_{COM}(IJ) = \frac{-\dot{\Delta}(IJ)}{t_c}$$

If the residual flag is not set, a check is made to determine if overshoot is impending. If the signs of the impending incremental pitch angle, Δ, and the present pitch rate, $\dot{\Delta}(IJ)$, differ, overshoot is impending; $\dot{\theta}(IJ)$ is too large or is in the opposite direction . In either case (overshoot or not), a test is made to determine if the total pitch angle to be accumulated after stopping the vehicle, Δ_{END}, is of the same sign as the desired angular change, Δ_D. The reason is that large aerodynamic moments can cause the vehicle to rotate or the vehicle was rotating in the opposite direction to that desired.

If the signs for Δ_D and Δ_{END} are alike (for the case with no impending overshoot), a test is made to determine if the absolute value of Δ_{END} is greater than 95% of the absolute value of the desired angular change, Δ_D. If Δ_{END} is greater than 95% of Δ_D, the desired pitch angular acceleration to stop the vehicle is again computed as:

$$\alpha_{COM}(IJ) = \frac{-\dot{\Delta}(IJ)}{t_c}$$

5.7.3.3.1 Acceleration Computations (Continued)

If Δ_{END} is less than 95% of Δ_D, the flow is to a test on the signs of the present and desired rates. For the cases with or without overshoot and unlike signs for Δ_{END} and Δ_D, the flow comes to the following decision point. If the sign of the present rate, $\dot{\Delta}(IJ)$, differs from the sign of the desired rate, ω, the desired angular acceleration is computed as:

$$\alpha_{COM}(IJ) = \frac{\omega-\dot{\Delta}(IJ)}{t_c}$$

If the signs of $\dot{\Delta}(IJ)$ and ω are alike, a test is made on the magnitude of the present rate, $\dot{\Delta}(IJ)$. If $\dot{\Delta}(IJ)$ is less than 99% of the desired rate ω, the desired angular acceleration is again computed as:

$$\alpha_{COM}(IJ) = \frac{\omega-\dot{\Delta}\ (IJ)}{t_c}$$

If the present rate is equal to or greater than 99% of ω, a test is made to see if the present rate exceeds the angular rate limit, ω_{LIM}(3°/sec). If the rate limit is not exceeded, the desired angular acceleration is set to zero.

$$\alpha_{COM}(IJ) = 0$$

5.7.3.3.1 Acceleration Computations (Continued)

If the present rate exceeds the rate limited, the maximum rate is set equal to the rate limit with appropriate sign by:

$$\dot{\Delta}_{MAX} = \frac{\dot{\Delta}(IJ)\ \left[\omega_{LIM}\right]}{\left|\dot{\Delta}\ (IJ)\right|}$$

And the desired angular acceleration is:

$$\alpha_{COM}(IJ) = \frac{\dot{\Delta}_{MAX} - \dot{\Delta}(IJ)}{t_c}$$

The main discussion, thus far, has been for the condition of no impending overshoot (like signs for $\dot{\Delta}(IJ)$ and Δ). Both sub-conditions of like and unlike signs for Δ_{END} and Δ_D have been described. In addition, one of the sub-conditions(unlike signs of Δ_{END} and Δ_D) has also been described for an impending overshoot case (unlike signs for $\dot{\Delta}$ (IJ) and Δ). The remaining impending overshoot case with sub-condition of like signs for Δ_{END} and Δ_D (indicating the initial pitch rate, $\dot{\theta}$, on the first pass was too large) will now be described.

After determining unlike signs for $\dot{\Delta}(IJ)$ and Δ and then determining like signs for Δ_{END} and Δ_D, the acceleration subroutine checks the magnitude of Δ_{END} vs. Δ_D. If the absolute value of Δ_{END} is less than 95% of the absolute

5.7.3.3.1 Acceleration Computations (Continued)

value of Δ_D, the desired pitch angular acceleration is computed as:

$$\alpha_{COM}(IJ) = \frac{\omega - \dot{\Delta}(IJ)}{t_c}$$

If the magnitude of Δ_{END} is greater than 95% of Δ_D, a test is made to determine if the magnitude of Δ_{END} is greater than 105% of Δ_D. If Δ_{END} is not greater than 105% of Δ_D, the pitch angular acceleration to stop the vehicle is computed as:

$$\alpha_{COM}(IJ) = \frac{- \dot{\Delta}(IJ)}{t_c}$$

If Δ_{END} is greater than 105% of Δ_D, the angular acceleration necessary to reduce the overshoot is computed as:

$$\alpha_{COM}(IJ) = \frac{-1.2\ [\dot{\Delta}(IJ)]}{t_c}$$

5.7.3.3.2 Application of System Gain

All of the branches for the various methods of computing the angular acceleration meet at the same decision point to determine how the system gain, G, is to be applied. First, a test is made to determine if separation has occurred (I_{SEP} = 1). If separation has not occurred, the acceleration subroutine determines if this is the first pass (of each major cycle) through the autopilot. If it is the first pass, the acceleration subroutine determines if the angular acceleration computed on this pass,

5.7.3.3.2 Application of System Gain (Continued)

$\alpha_{COM}(IJ)$, is in the same direction as the angular acceleration computed on the last pass, $\alpha_{COM}(IJ + 4)$. If the signs of $\alpha_{COM}(IJ)$ and $\alpha_{COM}(IJ + 4)$ are alike or if $\alpha_{COM}(IJ + 4)$ is equal to zero, the gain is applied as follows:

$$\alpha_{COM}(IJ) = \frac{\alpha_{COM}(IJ)}{G}$$

If the signs of $\alpha_{COM}(IJ)$ and $\alpha_{COM}(IJ + 4)$ differ, the gain is applied as follows:

$$\alpha_{COM}(IJ) = G\left[\alpha_{COM}(IJ)\right]$$

If $\alpha_{COM}(IJ)$ is equal to zero, the application of G is immaterial. If it is not the first pass, the sign test is made for $\alpha_{COM}(IJ-1)$ instead of α_{COM} (IJ + 4). When separation has occurred ($I_{SEP} = 1$), the gain is applied as if the signs are alike because of the small aerodynamic moments.

The reason for applying the system gain, G, in such a manner is due to the aerodynamic moments acting on the vehicle. The autopilot is unaware of the direction and magnitude of the aerodynamic moments presently acting on the vehicle. However, it can detect the direction of the aerodynamic moments that acted on the vehicle during the last minor cycle. If the acceleration computed for two consecutive minor cycles is in the same direction, the system gain after the first acceleration command more than likely is less than one. Conversely, if the gain is less than one and

5.7.3.3.2 Application of System Gain (Continued)

the desired rotational rate has not been attained, the next acceleration command will be in the same direction as the last. It is to our advantage, in this case, to divide the desired angular acceleration by the system gain.

If the signs of two consecutive acceleration commands differ, the gain resulting from one will be greater than 1.0 and from the other less than 1.0 (assuming no change in the direction of the aerodynamic moments). If the gain after the first acceleration command is greater than 1.0 (aerodynamics aiding), it is to our advantage to multiply the next acceleration command in the opposite direction by the gain to aid in overcoming the aerodynamic moments which will now be opposing.

If the gain after the first acceleration command is less than 1.0 (aerodynamics opposing), it is to our advantage again to multiply the next acceleration command (in the opposite direction) by the gain to reduce the effect of the aerodynamic moments which will now be aiding.

For the particular case of like signs for two consecutive angular acceleration commands with a gain greater than 1.0 after the first, it is again to our advantage to divide the second acceleration command by the gain to lessen the chance of exceeding the desired change in rotational rate. This particular case may arise when a change in rotational rate greater than two degrees per minor cycle (0.2 sec) is requested since the angular acceleration is limited to $10^{o}/sec^{2}$.

5.7.3.3.2 Application of System Gain (Continued)

After application of the system gain, G, a vehicle configuration check is made (Booster or Orbiter) and a test is made to assure that the angular acceleration to be commanded is less than the limit for the vehicle. A check is then made to determine which one of the three calculation flags (I_θ , I_ψ , or I_ϕ) was set. Since I_θ was set in this description, the pitch angular acceleration commands($\alpha_{\theta COM}(I)$) and the appropriate actual angular accelerations experienced ($\alpha_{\theta ACT}(I)$) are set equal to each of the five values for $\alpha_{COM}(I)$ and $\alpha_{ACT}(I)$, respectively. The corresponding calculation flag (I_θ) and the residual flag, I_{RESID}, are set to zero and the acceleration subroutine transfers control back to AUTOPI.

5.7.3.4 Engine Gimbal Model

The engine gimbal model is based on the equation:

$$M = I\alpha$$

where:

M = total moment of the applied forces about the respective axis

I = moment of Inertia about the respective axis

α = desired angular acceleration about the respective axis

The three moment equations are:

$$M_Y = I_{YY}\alpha_Y$$

$$M_Z = I_{ZZ}\alpha_Z$$

$$M_X = I_{XX}\alpha_X$$

In determining the thrust moments about the respective axes, certain assumptions are made regarding the thrust forces. The thrust force in the X body direction from one engine is equal to:

$$F_{TX} = F_T\left[\cos\,(\beta_P)\right]\left[\cos\,(\beta_Y)\right]$$

where:

F_T = engine thrust

β_p = engine gimbal pitch angle

β_Y = engine gimbal yaw angle

5.7.3.4 Engine Gimbal Model (Continued)

Similarly the thrust force in the Z body direction from one engine is equal to:

$$F_{TZ} = -F_T\left[\sin\left(\beta_P\right)\right]\left[\cos\left(\beta_Y\right)\right]$$

where the minus sign is due to a positive pitch angle (β_P) creating a force in the negative Z direction.

The equation for the moment about the Y-body axis from one engine is then:

$$M_Y = F_{TX}\left[\left(Z_i - Z_{cg}\right)\right] + F_{TZ}\left[\left(X_{cg}\right)\right]$$

where:

X_{cg} = distance from the engine mounting plane to the vehicle center of gravity

Z_{cg} = distance from the X-Y plane to the vehicle center of gravity

Z_i = distance from the X-Y plane to the engine mounting centerline of the ith engine.

The sum of the moments from all engines is:

$$\Sigma M_Y = F_{TXi}\left[Z_i - Z_{cg}\right] + F_{TZi}\left[X_{cg}\right]$$

$$= F_{Ti}\left[Z_i - Z_{cg}\right]\left[\cos(\beta_{Pi})\right] \cos\left(\beta_{Yi}\right) - F_{Ti}\left[X_{cg}\right]\left[\sin\left(\beta_{Pi}\right)\right] \cos\left(\beta_{Yi}\right)$$

However, if there were a maximum yaw angle of 7°, each of the two terms above would be multiplied by 0.9925, since cos (7°) = 0.9925. Because the yaw angle does not appreciably affect the moment about the Y axis, the cos (β_Y) term is ignored in the model such that the equation assumed for the thrust moment about the Y body axis is:

5.7.3.4 Engine Gimbal Model (Continued)

$$\Sigma M_Y = F_{Ti}\left[Z_i - Z_{cg}\right] \cos(\beta_{Pi}) - F_{Ti}\left[X_{cg}\right] \sin(\beta_{Pi})$$

5.7.3.4.1 Pitch Gimbal Angle

The above equation is in the form:

$$M_Y = A \cos x - B \sin x$$

where:

$$A = \Sigma F_{Ti}\left[Z_i - Z_{cg}\right]$$

$$B = \Sigma F_{Ti}\left[X_{cg}\right]$$

$$X = \beta_P$$

Multiplying the right side of the euqation by $\frac{\left[A^2 + B^2\right]^{1/2}}{\left[A^2 + B^2\right]^{1/2}}$ we have:

$$M_Y = A \cos X - B \sin X = \left[A^2 + B^2\right]^{1/2}\left[\frac{A}{\left[A^2 + B^2\right]^{1/2}} \cos X - \frac{B}{\left[A^2 + B^2\right]^{1/2}} \sin X\right]$$

There is an angle α such that:

$$\sin \alpha = \frac{A}{\left[A^2 + B^2\right]^{1/2}} \quad \text{and} \quad \cos \alpha = \frac{B}{\left[A^2 + B^2\right]^{1/2}}$$

Through substitution we then have:

$$M_Y = \left[A^2 + B^2\right]^{1/2}\left[\sin \alpha \cos X - \cos \alpha \sin X\right]$$

From the trigonometric identity:

$$\sin(m-n) = \sin m \cos n - \cos m \sin n$$

5.7.3.4.1 Pitch Gimbal Angle (Continued)

we then have:

$$M_Y = \sqrt{A^2 + B^2}\left[\sin(\alpha - x)\right]$$

By letting $\theta = (\alpha - x)$

$$M_Y = \sqrt{A^2 + B^2}\left[\sin\theta\right]$$

Therefore,

$$\theta = \text{Sin}^{-1}\left[\frac{M_Y}{\sqrt{A^2 + B^2}}\right]$$

Since $\theta = \alpha - x$,

$$x = \alpha - \theta$$

and since $\sin\alpha = \dfrac{A}{\sqrt{A^2 + B^2}}$

$$\alpha = \text{Sin}^{-1}\left[\frac{A}{\sqrt{A^2 + B^2}}\right]$$

Substituting the two inverse trignometric terms for α and θ we have:

$$\beta_p = x = \text{Sin}^{-1}\left[\frac{A}{\sqrt{A^2 + B^2}}\right] - \text{Sin}^{-1}\left[\frac{M_Y}{\sqrt{A^2 + B^2}}\right]$$

M_Y = The computed desired moment about the Y-body axis ($M_Y = I_{YY}\alpha_Y$)

A = The moment about the Y-body axis due to forces in the X-body direction

B = The moment about the Y-body axis due to forces in the Z-body direction

5.7.3.4.2 Yaw Gimbal Angle

The equation for the thrust moments about the Z body axis is:

$$\Sigma M_Z = F_{TXi}\left[Y_i - Y_{cg}\right] + F_{TYi}\left[X_{cg}\right]$$

$$= F_{Ti}\left[Y_i - Y_{cg}\right]\left[\cos(\beta_{pi})\right]\cos(\beta_{Yi}) + F_{Ti}\left[X_{cg}\right]\sin(\beta_{Yi})$$

Similar to ignoring the cos β_Y term in the moment about the Y axis, the cos β_P term may be dropped in the moment about the Z axis. However, since the value of β_p has already been determined, we will use it for greater accuracy in the above equation which is in the form:

$$M_Z = -A\cos x + B\sin x$$

Since

$$-\sin(m-n) = \sin(n-m) = -\sin m\cos n + \cos m\sin n$$

$$M_Z = \sqrt{A^2 + B^2}\left[\sin(x-\alpha)\right]$$

Letting $\theta = (x - \alpha)$

$$M_Z = \sqrt{A^2 + B^2}\left[\sin\theta\right]$$

Since $x = \theta + \alpha$

$$\frac{\beta_Y}{.95} = x = \sin^{-1}\left[\frac{M_Z}{\sqrt{A^2 + B^2}}\right] + \sin^{-1}\left[\frac{A}{\sqrt{A^2 + B^2}}\right]$$

Where: M_Z = the computed desired moment about the Z-body axis

$A = F_{Ti}\left[Y_1 - Y_{CG}\right]\cos(B_{pi})$

$B = F_{Ti}\left[X_{CG}\right]$

5.7.3.4.3 Roll Angle (Booster)

In determining rotation about the booster X body axis (Roll), a system is assumed such that the engines can be gimbaled in the pitch plane angle without disturbing the existing yaw angle. The engines to the left of the X-Z body plane (no Y_{cg} travel assumed) are gimbaled to a pitch angle equal and opposite to the direction of the engines to the right of the X-Z body plane, forming a couple. Since the null axis of each engine lies in a plane parallel to the X-Z plane, the yawing moment will not be altered. The total pitch angle (β_{PT}) for each engine is the sum of the pitch angle computed for the moment about the Y body axis (β_P) plus the computed pitch angle for the moment about the X body axis (β_R).

The equation for the moment about the X body axis is:

$$\Sigma M_X = F_{TZi} \left[Y_i - Y_{cg}\right] + F_{TYi} \left[Z_i - Z_{cg}\right]$$

$$= F_{Ti} \left[Y_i - Y_{cg}\right] \left[\text{Sin } (\beta_{Pi})\right] \cos (\beta_{Yi}) + F_{Ti} \left[Z_i - Z_{cg}\right] \text{Sin } (\beta_{Yi})$$

where the first term is the moment about the X axis due to forces in the Z direction, M_{XFZ}, (mainly a function of the pitch angle) and the second term is the moment due to forces in the Y direction, M_{XFY}, (a function of the yaw angle). Since the yaw angle (β_Y) has already been computed, the above equation may be written:

$$M_X - M_{XFY} = M_{XFZ} = F_{Ti} \left[Y_i - Y_{cg}\right] \left[\text{Sin } (\beta_{Pi})\right] \cos (\beta_{Yi})$$

5.7.3.4.3 Roll Angle (Booster) (Continued)

Solving for the magnitude of the roll portion of the pitch angle ($B_R = B_{pi}$) necessitates using absolute values for the Y-body engine locations, since half the pitch angles commanded will be positive and half will be negative, such that:

$$\beta_R = \text{Sin}^{-1}\left[\frac{M_X - M_{XFY}}{F_{Ti}|Y_i - Y_{cg}|\text{Cos}(\beta_{Yi})}\right]$$

The Roll angles commanded are:

$$\beta_{R1} = \beta_R$$
$$\beta_{R2} =-\beta_R$$
$$\beta_{R3} = \beta_R$$
$$\beta_{R4} = \beta_R$$
$$\beta_{R5} =-\beta_R$$
$$\beta_{R6} =-\beta_R$$
$$\beta_{R7} = \beta_R$$
$$\beta_{R8} = \beta_R$$
$$\beta_{R9} =-\beta_R$$
$$\beta_{R10} =-\beta_R$$
$$\beta_{R11} = \beta_R$$
$$\beta_{RR} = -\beta_R$$

The total engine pitch gimbal is the sum of the angles computed for pitch and roll.

$$\beta_{PTi} = \beta_{Pi} + \beta_{Ri}$$

5.7.3.4.4 Roll Angle (Orbiter)

The roll angle for the Orbiter is determined in a manner identical to that of the Booster. The roll moment is created by gimbaling one engine above the Z_{cg} to a pitch angle equal and opposite to that of the engine below the Z_{cg}. The equation for the moment about the X-body axis is identical to the computations for the booster case, except that two engines are being gimbaled rather than twelve.

The total engine pitch angle is the sum of the angles commanded for pitch and roll. These equations are

$$\beta_{PT1} = \beta_{P1} + \beta_{R1}$$

$$\beta_{PT2} = \beta_{P2} - \beta_{R2}$$

5.8 CONVENTIONAL AUTOPILOT MATH MODEL

The AUTOP2 Subroutine is selectable in LASS. If the KAUT discrete in the master or FUN routine is set to zero and the engines are on, the conventional autopilot controls LASS. The AUTOP2 subroutine is used to compute total mass remaining of the vehicle and engine gimbal angles. The autopilot consists of the AUTOP2 subroutine which calls up the Z2 subroutine. The Z2 subroutine is used to interrogate the autopilot gain table.

5.8.1 AUTOP2 Subroutine

A functional flow chart is shown in Figure 17. The subroutine is first used to compute total vehicle mass remaining. Total vehicle mass is used in other routines such as to interrogate mass characteristics tables.

The conventional autopilot uses a conventional engine control law. The form of the control law is

$$\delta = a_0 + a_1 \theta_E + a_2 \dot{\theta}_E$$

where:

δ - engine deflection

a_1, a_2 - weighting coefficients or gains

θ_E - present vehicle attitude error

$\dot{\theta}_E$ - present vehicle attitude rate error

The a_0 term is a bias term that corresponds to the engine gimbal deflection at the end of the present computation cycle. Then

$$\delta_{TRIM} = \delta_{TRIM} + .2 \frac{(\theta_{EL} + \theta_E)}{2}$$

$$a_0 = .2\,\delta_{TRIM}$$

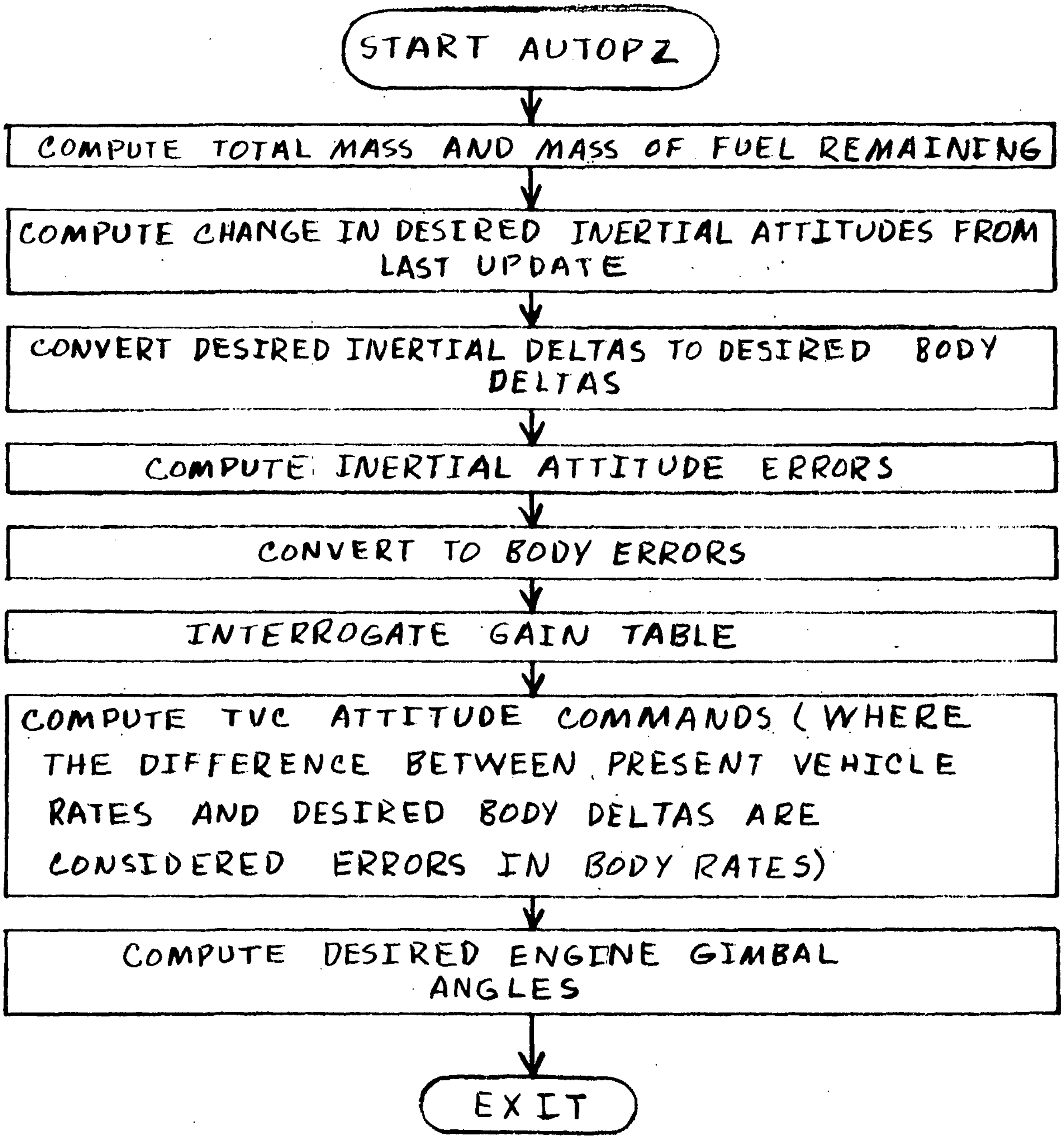

START AUTOPZ
COMPUTE TOTAL MASS AND MASS OF FUEL REMAINING
COMPUTE CHANGE IN DESIRED INERTIAL ATTITUDES FROM LAST UPDATE
CONVERT DESIRED INERTIAL DELTAS TO DESIRED BODY DELTAS
COMPUTE INERTIAL ATTITUDE ERRORS
CONVERT TO BODY ERRORS
INTERROGATE GAIN TABLE
COMPUTE TVC ATTITUDE COMMANDS (WHERE THE DIFFERENCE BETWEEN PRESENT VEHICLE RATES AND DESIRED BODY DELTAS ARE CONSIDERED ERRORS IN BODY RATES)
COMPUTE DESIRED ENGINE GIMBAL ANGLES
EXIT

FIGURE 17 FUNCTIONAL FLOW CHART FOR AUTOPZ SUBROUTINE

5.8.1 AUTOP2 Subroutine (Continued)

where:

θ_{EL} - attitude error from the past computation cycle

provides an estimate of a_o and is used to compute engine pitch gimbal deflections.

The specific form of the equations for the first two engine pitch deflections are:

$$\delta_{P1} = a_o + a_{1P}\,\theta_{EP} + a_{2P}\,\dot{\theta}_{EP} + a_{1R}\,\theta_{ER} + a_{2R}\,\dot{\theta}_{ER}$$

$$\delta_{P2} = a_o + a_{1P}\,\theta_{EP} + a_{2P}\,\dot{\theta}_{EP} + a_{1R}\,\theta_{ER} - a_{2R}\,\dot{\theta}_{ER}$$

where:

P - denotes motion in pitch

R - denotes motion in roll

1 - denotes engine #1

2 - denotes engine #2

The form of the equations for engine 1 and 2 yaw deflections are:

$$\delta_{Y1} = \delta_{Y2} = a_{1Y}\,\theta_{EY} + a_{2Y}\,\dot{\theta}_{EY}$$

where:

y - denotes motion in yaw

These equations are extended to the 12 engine booster configuration by application of equations shown in flow charts of Appendix A and program listing shown in Appendix B.

5.8.1 AUTOP2 Subroutine (Continued)

Attitude and attitude rate feedback from the EOM subroutine is provided as described in the following.

J is the minor loop pass counter in the main program. J goes to 1 at the beginning of each major cycle when a new guidance command (desired Euler attitude) is given. The autopilot logic is set up to use J so that attitude rate feedback is used on major cycles (once per second) and attitude feedback is used on minor cycles (5 times per second). Attitude information is used to derive the desired attitude rate by numerical differentiation, whereas the actual rate is taken from the EOM or, in the real vehicle, from rate gyros which are assumed to have no error. The difference between actual and desired attitude and rates are transformed to body coordinates using an infinitesimal rotation transformation matrix as follows:

$$\underset{\text{Body Coordinates}}{\begin{bmatrix} \Delta\theta\beta \\ \Delta\theta\beta \\ \Delta\theta\beta \end{bmatrix}} = \underset{\text{Infinitesimal Rotation Transformation}}{\begin{bmatrix} c\theta\psi \; c\theta\phi & \sin\phi & 0 \\ -\, c\theta\psi \; \sin\phi & c\theta\phi & 0 \\ \sin\psi & 0 & 1 \end{bmatrix}} \underset{\text{Platform coordinates}}{\begin{bmatrix} \Delta\theta \\ \Delta\psi \\ \Delta\phi \end{bmatrix}}$$

The differences used for attitude calculation are modified to accommodate quadrant boundary transition. For example, the desired attitude could be +179.5° yielding a difference of 1°. The attitude errors are limited so as not to exceed 2 degrees in pitch and 2.5 degrees in yaw and roll. These numbers were selected by trial to allow the engine deflection to peak at about 10 degrees at maximum dynamic pressure during a normal launch.

5.8.1 AUTOP2 Subroutine (Continued)

The pitch command has a bias which allows the thrust vector to be directed through the center of gravity. Other than this, the pitch, yaw, and roll channels are of identical form with rate and attitude feedback to form a conventional gyro autopilot.

The pitch and yaw commands are used to drive the pitch and yaw gimbals, respectively. The roll command is added differentially to the pitch gimbals to provide roll control. Software engine gimbal stops of 10 degrees are used.

Variable inputs to the AUTOP2 subroutine are shown in Table XVIII. These are primarily attitude and rate information provided by the equations of motion and guidance subroutines. Variable output information is shown in Table XIX.

TABLE XVIII AUTOP2 SUBROUTINE INPUTS

CODED SYM.	ENGR. SYM.	SYMBOL DEFINITION OR USE	UNITS
AFLOWB		FUEL FLOW RATE TO BOOSTER ENGINE	
AFLOWO		FUEL FLOW RATE TO ORBITER ENGINE	
EPB	p	BODY ROLL RATE	rad/sec
EPHI	ϕ	INERTIAL ROLL ANGLE	rad
EPSI	ψ	INERTIAL YAW ANGLE	rad
EPT		THROTTLE SETTING FOR ENGINES	PERCENT
EQB	q	BODY PITCH RATE	rad/sec
ERB	r	BODY YAW RATE	rad/sec
ETHETA	θ	INERTIAL PITCH ANGLE	rad
GPHID	ϕ_D	DESIRED INERTIAL ROLL ANGLE	rad
GPSID	ψ_D	DESIRED INERTIAL YAW ANGLE	rad
GTHED	θ_D	DESIRED INERTIAL PITCH ANGLE	rad
IES		IES=1, INDICATES ENGINES ARE ON	DISCRETE
ISEP		ISEP=1 INDICATES SEPARATION HAS TAKEN PLACE	DISCRETE
J		MINOR LOOP COUNTER	DISCRETE
NOENGB		NUMBER OF BOOSTER ENGINES	
NOENGO		NUMBER OF ORBITER ENGINES	
PAO	a_1	PITCH ATTITUDE GAIN	
PA1	a_2	PITCH RATE GAIN	
RAO	a_1	ROLL ATTITUDE GAIN	
RA1	a_2	ROLL RATE GAIN	
SETGGO		ANGLE BETWEEN BODY X AND THRUST VECTOR at LAUNCH	rad
T	t	GROUND ELAPSED TIME	sec
YAO	a_1	YAW ATTITUDE GAIN	
YA1	a_2	YAW RATE GAIN	

TABLE XIX AUTOP2 SUBROUTINE OUTPUTS

CODED SYM.	ENGR. SYM.	SYMBOL DEFINITION OR USE	UNITS
AMTOT		PRESENT MASS OF THE VEHICLE ENGINE GIMBAL	SLUGS
EBP(12)	S_{pi}	PITCH ANGLE	RAD
EBY(12)	δ_{Yi}	ENGINE GIMBAL YAW ANGLE	RAD
IES		IES=1, ENGINES ARE ON	DISCRETE
PBERRA	θ_{EP}	VALUE OF BODY PITCH ANGLE ERROR	RAD
RBERRA	θ_{ER}	VALUE OF BODY ROLL ANGLE ERROR	RAD
YBERRA	θ_{EY}	VALUE OF BODY YAW ANGLE ERROR	RAD

5.8.2 Z2 Subroutine

The Z2 Subroutine is used to interrogate the autopilot attitude and rate gain table. The set of gains are shown in Table XX. When Z2 is called, ground elapsed time is used as a basis to select the gains from the gain table. These gains were generated by running a nominal simulation to orbit insertion assuming a control system characteristic frequency of 1.5 radians per second. Since the abort trajectories to be tested deviate from the nominal, the gains correspond in application to some other frequency. This is not considered to be a serious shortcoming since the simulation is running without bending and slosh considerations and is therefore relatively insensitive to the control frequency used.

The gains were computed for a nominal launch using the ideal autopilot to control the vehicle. The actual gains were computed using

$$a_1 = \frac{\omega_n^2 - C_1}{C_2}$$

$$a_2 = \frac{2\delta \omega_n}{C_2}$$

where

δ - is the damping ratio (set at .7)

ω_n - natural frequency (set at 1.5)

C_1 and C_2 - computed variables depending on angle of attack and sideslip and vehicle velocity, altitude, and geometry.

The actual equations used for computing C_1 and C_2 are described in reference 19.

TABLE XX - Conventional Autopilot Gains (ω= 1.5)

	PITCH		YAW		ROLL	
TIME	PAO ATTITUDE	PA1 RATE	YAO ATTITUDE	YA1 RATE	RAO ATTITUDE	RA1 RATE
0	1.39	1.30	1.35	1.26	1.67	1.55
74	.319	1.05	1.05	1.02	1.49	1.39
140	.922	.855	.968	.874	1.38	1.29
200	.737	.689	.720	.668	1.20	1.12
215	.737	.689	.720	.668	1.20	1.12
216	1.08	1.05	1.28	1.19	1.56	1.46
218	1.08	1.05	1.28	1.19	1.56	1.46
230	1.13	1.05	1.26	1.18	1.51	1.41
300	1.02	.954	1.15	1.07	1.23	1.15
430	.871	.797	.933	.870	.781	.729

Note: Gain table is for 161c delta wing orbiter.

5.9 ABORT GUIDANCE MATH MODEL

The abort guidance math model is described in Reference 4 and is implemented in two phases. The first phase consists of the ABGUID Subroutine which calls the TARGET Subroutine and is used during the abort where engines are providinq thrust. The second phase consists of the PBC Subroutine which steers the orbiter when engines are off and the orbiter is coasting to its terminal energy state.

The TARGET subroutine performs abort targeting and ABGUID provides the desired attitudes which form the abort trajectory. The input data list, shown in Table XXI, and XXII, includes only those items required from other subroutines in LASS. Output from the abort guidance math model (ABGUID and PBC subroutines) are the desired euler angles θ_D, Ψ_D, and ϕ_D) and the simulation termination flag, KGUID, from PBC.

The TARGET subroutine provides ABGUID with the unit vectors ($\hat{U}_{XGP}$, $\hat{U}_{YGP}$, and $\hat{U}_{ZGP}$), target velocity ($\bar{V}_{DG}$ or $\dot{R}_D$, $\dot{Y}_P$, $\dot{Z}_D$), target position (R_{DG} or R_D, Y_D, Z_D) and effective gravity (TGFIN) at targeted velocities and position. The ABGUID subroutine provides TARGET with an estimate of time to go (T_{GO}).

The purpose of the abort guidance is to direct the orbiter to a landinq site if something happens to the booster during launch. The abort maneuver itself is in two parts. The first part is an orbiter burn to an intermediate target

5.9 ABORT GUIDANCE MATH MODEL (Continued)

TABLE XXI

INPUT TO ABORT GUIDANCE TARGET OR ABGUID SUBROUTINES

CODED SYM.	ENGR. SYM.	TARGET	ABGUID	NOMENCLATURE
AALLM			X	Acceleration Limit
AMTØT	M_T	X	X	Vehicle mass
AZCF	Z_{CF}		X	Reference Z distance
EATT	A_T		X	Thrust acceleration
EPT	-		X	Fractional throttle setting
TRP & ERP(4)	R_p	X	X	Vehicle position
EVP(4)	Y_p	X	X	Vehicle velocity
EXCG(AMTØT)	X_{CG}		X	Center of gravity position
ETCG(AMTØT)	Y_{CG}		X	Center of gravity position
EZCG(AMTØT)	Z_{CG}		X	Center of gravity position
GOTM			X	VC rotation rate limit
GGØ(3)	A_L		X	Limiting acceleration
GEGVEL	Ve		X	Engine exhaust gas velocity
KLPDR		X	X	Abort destination
SMU	u	X	X	Gravitational constant
SRMEAN	R_E		X	Mean earth radius
T	-	X	X	Ground elapsed time
TGØ or TGØT	T_{GO}	X	X	Time to go
THBØ		X		Target attitude
THELIM			X	Pitch attitude rate limit
TIP(3,3)	-	X		Transformation matrix
TLAT	$_{LS}$	X		Latitude of landing site
TLONG	Ø_{LS}	X		Longitude of landing site
TPPCO			X	Pitch profile cutoff time
TRE	R_e	X		Earth radius at launch site
TRLSE(3)		X		Radial vector to landing site
TT(6)	-	X		Transformation matrix
TWE	W_e	X		Earth rotation rate
TWEP	W_{ep}	X		Earth rotation vector in platform coordinates

TABLE XXII INPUT INFORMATION REQUIRED BY THE PBC SUBROUTINE

CODED SYM	ENGR SYM	NOMENCLATURE
H	H	Altitude
T		Time from launch
III		First pass flag
KLPDR		Return to launch pad flag
PVPB	V_{PB}	Re-entry target velocity
SGRAV	G	Acceleration due to gravity
PHPB	H_{PB}	Re-entry target altitude
VA(4)	$\vec{V}_A$	Relative velocity in platform coordinates
UXTP(3)	$\vec{U}_{XP}$	Unit vector from center of earth to vehicle
BETA		Sideslip angle
UYBP(3)	$\vec{U}_{YBP}$	Unit vector expressing body Y axis in platform coordinates
DFAB(4)	$\overrightarrow{FA}$	Aerodynamic force vector
AMT	M	Total mass of vehicle
PTHEC		Command inertial attitude
PGMAX2		Maximum aero deceleration
PALPD3		Desired angle of attack
PALPD4		Desired angle of attack
PALPD5		Desired angle of attack
PHLIM		Constraint altitude
ALPHA		Angle of attack
PVBAC1		Maximum bank angle rate
PBAD		Desired bank angle

5.9 ABORT GUIDANCE MATH MODEL (Continued)

and the second is an unpowered re-entry to between 150,000 to 100,000 feet over the landing site. The intermediate target is chosen such that the orbiter will reach the target coincident with fuel depletion.

5.9.1 ABGUID Subroutine

The powered abort guidance phase is divided into two parts. The first part, termed weighted open loop guidance, is used initially after abort. The orbiter is kept in the launch plane while thrust is used to gain altitude and direct the orbiter in the direction of launch. The process utilizes abort target altitude and altitude rate, provided by the TARGET subroutine, as pseudo terminal conditions. This continues until the orbiter can reach abort targeted terminal conditions coincident with propellant depletion. The second part of powered abort guidance is closed loop, The guidance is now explicit requiring present and terminal state information. In order for clarity, closed loop abort guidance will be discussed first.

5.9.1 ABGUID Subroutine (Continued)

Detailed flow charts and symbol definitions of the ABGUID subroutine are shown in Appendix A. Listings of the actual program are shown in Appendix B. The ABGUID subroutine is used to provide angles, which describe the vehicle's attitude in platform coordinates, and an estimate of time required to reach the intermediate target, provided by the TARGET subroutine. A list of inputs to the ABGUID subroutine is shown in Table XXI. As noted previously, ABGUID outputs are the desired inertial angles (θ_D, ψ_D, and ϕ_D) and estimates of burn time remaining (T_{GO}). The ABGUID subroutine in the present formulation of LASS outputs information to the READØ subroutine for printout. A functional listing of ABGUID is shown in Figure 18. The erection of the guidance coordinate system is provided by the TARGET subroutine and a description can be found in Section 5.9.2. The vehicle position and velocity vectors are established in guidance coordinates by

$$g_{AV} = (g_{eff} + g_{fin})\, T_{GO}$$

$$\left[\bar{R}_G\right] = \begin{bmatrix} R \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} |\bar{R}_P| \\ 0 \\ 0 \end{bmatrix}$$

$$\left[\bar{V}_G\right] = \begin{bmatrix} \dot{R} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \hat{U}_{XGP} \\ \hline \hat{U}_{YGP} \\ \hline \hat{U}_{ZGP} \end{bmatrix} \left[\bar{V}_P\right]$$

$$\left[\bar{V}_{GG}\right] = \begin{bmatrix} \dot{R}_G \\ \dot{Y}_G \\ \dot{Z}_G \end{bmatrix} = \begin{bmatrix} \dot{R}_D \\ \dot{Y}_D \\ \dot{Z}_D \end{bmatrix} - \begin{bmatrix} \dot{R} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} - \begin{bmatrix} g_{AV} \\ 0 \\ 0 \end{bmatrix}$$

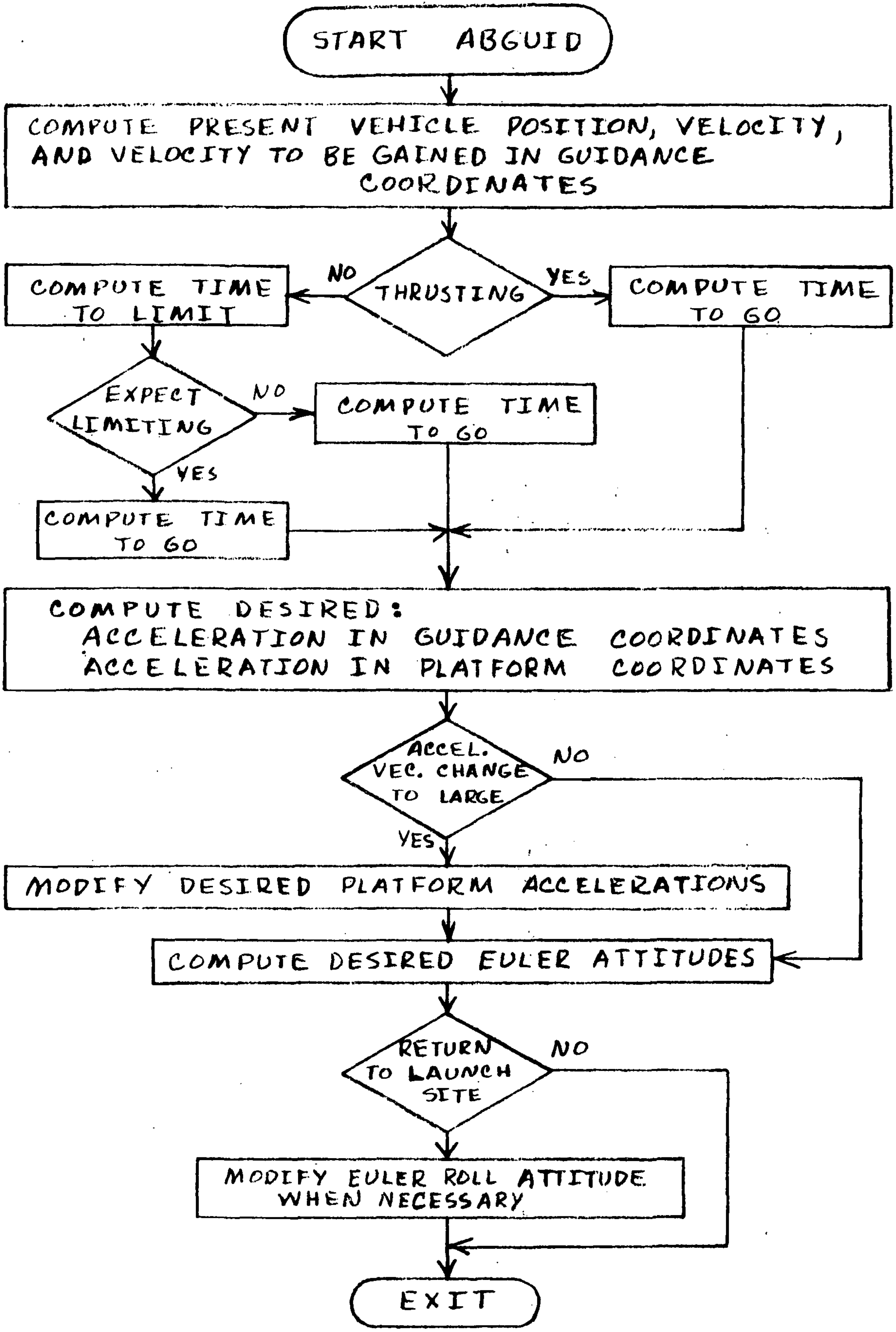

START ABGUID
COMPUTE PRESENT VEHICLE POSITION, VELOCITY, AND VELOCITY TO BE GAINED IN GUIDANCE COORDINATES
THRUSTING
NO
YES
COMPUTE TIME TO LIMIT
COMPUTE TIME TO GO
EXPECT LIMITING
NO
YES
COMPUTE TIME TO GO
COMPUTE TIME TO GO
COMPUTE DESIRED:
ACCELERATION IN GUIDANCE COORDINATES
ACCELERATION IN PLATFORM COORDINATES
ACCEL. VEC. CHANGE TO LARGE
NO
YES
MODIFY DESIRED PLATFORM ACCELERATIONS
COMPUTE DESIRED EULER ATTITUDES
RETURN TO LAUNCH SITE
NO
MODIFY EULER ROLL ATTITUDE WHEN NECESSARY
EXIT

FIGURE 18 FUNCTIONAL FLOW CHART FOR ABGUID SUBROUTINE

5.9.1 ABGUID Subroutine (Continued)

where

$\bar{R}_G$ - present vehicle position

$|\bar{R}_p|$ - magnitude of inertial position

g_{AV} - average velocity loss due to gravity

g_{eff} - effective gravity of vehicle in present situation

g_{FIN} - effective gravity of vehicle at target

T_{GO} - time to reach target

$\bar{V}_p$ - inertial velocity of vehicle in platform coordinates

$\hat{U}_{XGP}$ - unit guidance X in platform coordinates

$\hat{U}_{YGP}$ - unit guidance Y in platform coordinates

$\hat{U}_{ZGP}$ - unit guidance Z in platform coordinates

V_G - present vehicle velocity in guidance coordinates

$\dot{R}_D$, $\dot{Y}_D$, $\dot{Z}_D$- desired velocities at target in guidance coordinates

$\bar{V}_{GG}$ - velocity to be gained in guidance coordinates

Whenever thrust acceleration, A_T, is zero computation of inertial attitudes is bypassed. The last set of inertial attitudes computed are held and maintained in inertial hold. When A_T is not zero an estimate of time to reach the intermediate target, T_{GO}, is provided. In the case where thrust is limited A_T is equal to acceleration limit and

$$T_{GO} = \frac{|\bar{V}_{GG}|}{A_T}$$

5.9.1 ABGUID Subroutine (Continued)

For the case where the thrust is not presently limited, an estimate of time to limit, T_{FL}, is accomplished by

$$T_{FL} = \frac{V_e}{A_T} - \frac{V_e}{A_L}$$

where V_e is the exhaust gas velocity of the orbiter engines. The parameter Υ is given by

$$\Upsilon = \frac{V_e}{A_T}$$

As noted in Section 5.3, for constant thrust, the characteristic velocity equation provides an estimate of velocity gained by the vehicle during the time in constant thrust mode. The velocity gained, in the negative sense, over time T_{FL} is

$$V_{FL} = V_e \ln \left[1 - \frac{T_{FL}}{\Upsilon}\right]$$

The velocity to be gained after thrust limiting is reached is

$$V_{GL} = V_{FL} + |\overline{V}_{GG}|$$

If V_{GL} is less than zero, the vehicle will function in constant thrust mode until intermediate target is reached. T_{GO} is computed by

$$T_{GO} = \Upsilon \left[1 - e^{\frac{-|\overline{V}_{GG}|}{V_e}}\right]$$

5.9.1 ABGUID Subroutine (Continued)

If V_{GL} is greater than zero, the vehicle will function in mixed mode. Constant thrust mode will be effective until time T_{FL} and then the vehicle will function in constant acceleration mode. Under these conditions T_{GO} is computed by

$$T_{GO} = T_{FL} + \frac{V_{GL}}{A_L}$$

For the development of the desired vehicle accelerations, the reader is referred to Section 5.3.3 for similarities in the development. The desired radial acceleration in guidance coordinates is commonly accepted as

$$\ddot{R}_G = A_T\ (A + Bt) - G_{eff} \tag{1}$$

where A and B are linear coefficients.

By taking the first integral over T_{GO} we have

$$A \int_0^{T_{GO}} A_T dt + B \int_0^{T_{GO}} A_T\ t\ dt = \dot{R}_D - \dot{R} \tag{2}$$

or

$$A_{11}\ A + A_{12}\ B = \dot{R}_D - \dot{R} \equiv \Delta \dot{R}$$

where $\dot{R}_D$ is targeted radial velocity and $\dot{R}$ is present vehicle radial velocity. If the double integral of equation (1) is taken over T_{GO}, we obtain

$$A \int_0^{T_{GO}} \left[\int_0^t A_T\ dt \right] dt + B \int_0^{T_{GO}} \left[\int_0^t A_T t dt \right] dt = R_D - R - \dot{R}T_{GO} \equiv \Delta R \tag{3}$$

or

$$A_{21}A + A_{22}\ B = R_D - R - \dot{R}T_{GO}$$

5.9.1 ABGUID Subroutine (Continued)

where R_D is the desired radial distance and R is present vehicle radial distance from the earth's center. Let us assume that, regardless of the modes of vehicle operation, the thrust acceleration can be averaged to a constant A_{AV}. The integrals of equation (2) and (3) reduce to

$$\begin{aligned} A_{11} &\cong A_{AV}\, T_{GO} \\ A_{12} &\cong .5A_{AV}\, T_{GO}{}^2 \\ A_{21} &\cong .5A_{AV}\, T_{GO}{}^2 \\ A_{22} &\cong A_{AV}\, \frac{T_{GO}{}^3}{6} \end{aligned} \qquad (4)$$

The equations (2) and (3) are solved simultaneously.

$$A = \frac{\Delta\dot{R}}{A_{11}} - B\,\frac{A_{12}}{A_{11}} \qquad (5)$$

From (3) and (5)

$$A_{21}\left[\frac{\Delta\dot{R}}{A_{11}} - B\,\frac{A_{12}}{A_{11}}\right] + A_{22}B = \Delta R$$

$$B\,(A_{22} - \frac{A_{12}A_{21}}{A_{11}}) = \Delta R - A_{21}\frac{\Delta\dot{R}}{A_{11}} = A_{21}\,(\frac{\Delta R}{A_{21}} - \frac{\Delta\dot{R}}{A_{11}})$$

$$B = \frac{A_{21}}{A_{22} - \frac{A_{21}A_{12}}{A_{11}}}\,(\frac{\Delta R}{A_{21}} - \frac{\Delta\dot{R}}{A_{11}}) \qquad (6)$$

5.9.1 ABGUID Subroutine (Continued)

Now substituting (6) into (5) gives

$$A = \frac{\Delta\dot{R}}{A_{11}} - \frac{A_{12}}{A_{11}} \frac{A_{21}}{A_{22} - \frac{A_{21}A_{12}}{A_{11}}} \left(\frac{\Delta R}{A_{21}} - \frac{\Delta\dot{R}}{A_{11}} \right) \qquad (7)$$

Substituting the results of (4) into (7)

$$A \cong 3\frac{\Delta R}{A_{21}} - 2\frac{\Delta\dot{R}}{A_{11}} \qquad (8)$$

If we assume $A \gg Bt$, equation (1) becomes

$$\ddot{R}_G \cong A_T \left(\frac{3\Delta R}{A_{21}} - \frac{2\Delta\dot{R}}{A_{11}} \right) - g_{eff} \qquad (9)$$

Now A_{11} is the characteristic velocity and A_{21} is the characteristic distance. Then we assume

$$A_{11} = |\bar{V}_{GG}| \qquad (10)$$

$$A_{21} = \frac{|\bar{V}_{GG}|\ T_{GO}}{2}$$

Equation (9) now reduces to

$$\ddot{R}_G \cong \frac{1}{T_{GO}} \left[\frac{6}{T_{GO}} (R_D - R) - 2(\dot{R}_D + 2\dot{R}) \right] - g_{eff} \qquad (11)$$

by using equations (2), (3), and (10). This is the form of the radial acceleration equation used in Reference 4. When T_{GO} is less than 5, the radial distance constraint is modified and equation (11) reduces to

$$\ddot{R}_G \cong \frac{1}{T_{GO}} (\dot{R}_D - \dot{R}) - g_{eff} \qquad (12)$$

5.9.1 ABGUID Subroutine (Continued)

If we neglect the g_{eff} term, Y_G can be developed in a similar manner where the form of equation (12) is used.

$$\ddot{Y}_G = \frac{1}{T_{GO}} \left(\dot{Y}_D - \dot{Y} \right) \tag{13}$$

is used to provide the desired out of plane acceleration. The downrange acceleration, Z_G, is computed so that

$$A_T{}^2 = \ddot{X}_G{}^2 + \ddot{Y}_G{}^2 + \ddot{Z}_G{}^2 \tag{14}$$

In a manner similar to that described in Section 5.3, the desired accelerations are converted to platform coordinates by using the unit vectors in a direction cosine matrix

$$\left[\bar{A}_{DP}\right] = \left[\hat{U}_{XGP} \,\vdots\, \hat{U}_{YGP} \,\vdots\, \hat{U}_{ZGP}\right] \begin{bmatrix} \ddot{R}_G \\ \ddot{Y}_G \\ \ddot{Z}_G \end{bmatrix} \tag{15}$$

5.9.1 ABGUID Subroutine (Continued)

Equations 11, 13, and 14 provide the means for computing desired vehicle accelerations when ABGUID is steering the orbiter using closed loop procedures. As noted previously, weighted open loop guidance is used initially after abort. The transition between abort guidance processes is identified as a time after abort is initiated. This time is read from input cards as TTDI by the FUN routine.

When using weighted open loop guidance, desired accelerations are computed by modifying equations 11, 13, and 14 to

$$\ddot{R}_G = K_1 (R_D - R) + K_2 (\dot{R}_D - \dot{R}) - g_{eff}$$

$$\ddot{Y}_G = 0$$

$$\ddot{Z}_G = A_T^2 + \ddot{R}_G^2$$

where:

K_1 = .0001 when abort time is greater than 10

K_1 = .0002 when abort time is less than 10

K_2 = .03 when the vehicle is within 50,000 feet of abort altitude and magnitude of altitude rate is less than 1000 ft/sec

K_2 = .01414 for all other conditions

These accelerations are set identical to the desired platform accelerations or

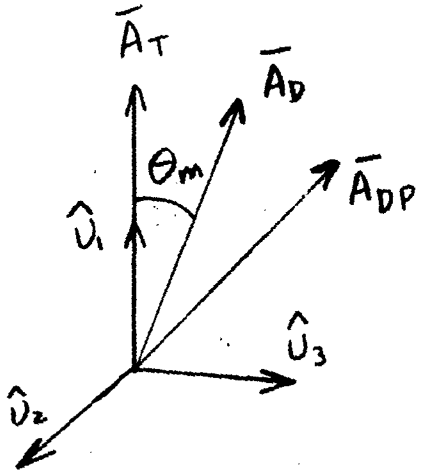
$$\left[\bar{A}_{DP} \right] = \begin{bmatrix} \ddot{R}_G \\ \ddot{Y}_G \\ \ddot{Z}_G \end{bmatrix} \qquad (16)$$

5.9.1 ABGUID Subroutine (Continued)

When an abort is commanded where the orbiter returns to the launch site, large pitching maneuvers will be required. This results in large body attitude erros that are encountered when powered abort guidance is operating closed loop. Large errors are incompatible when using conventional forms of autopilots. As a result the desired inertial accelerations provided by equation 15 are modified to limit large attitude errors to errors more compatible with the autopilot. This is accomplished by erecting a coordinate system shown in Figure 19 where the vectors are referenced to the platform coordinate system. The process shown in Figure 19 is sensitive to the quadrant that $\bar{A}_{DP}$ is in with respect to $\bar{A}_T$. Since the normal abort pitching maneuver causes the body x-axis to pass near vertical, platform x-axis (which is near vertical when vehicle is nearly over the launch pad) is substituted for $\bar{A}_{DP}$ initially. During this period of time, the vehicle rotates following $\bar{A}_D$ until $\bar{A}_D$ is within θ_m of platform x-axis. At this time the vehicle starts to rotate towards $\bar{A}_{DP}$ at a rate equal to θ_m (card input through FUN routine) radians per second.

The vector $\bar{A}_T$ is the thrust vector transformed to platform coordinates. Since $\bar{A}_T$ is the instantaneous thrust vector, it depends on engine gimbaling and is relatively unstable. Thus, after the first computational pass through the VC algorithm, the vector $\bar{A}_D$ used in the previous computational pass is substituted for $\bar{A}_T$. The assumption is that the vehicle will rotate to the new $\bar{A}_D$ vector in one computational pass. Whenever the desired acceleration vector $\bar{A}_{DP}$, provided by equation 15, is equal to or within θ_m degrees of the thrust acceleration vector $\bar{A}_T$, the vector control algorithm is bypassed and $\bar{A}_{DP}$ is not modified. Specific equations are shown in the flow charts of Appendix A and program listings of Appendix B.

The components of $\bar{A}_{DP}$ provided by either one of equations 15, 16, or 17 are used to provide the desired euler attitudes (θ_D and ψ_D) in the same manner as described in Section 5.3. The euler roll angle is either commanded to zero or 180^o degrees as required by the abort case being studied.



$\bar{A}_T$ - THRUST ACCELERATION VECTOR IN PLATFORM COORDINATES FOR INITIAL PASS AND PAST $\bar{A}_D$ FOR SUBSEQUENT PASSES.

$\bar{A}_{DP}$ - DESIRED VEHICLE ACCELERATION VECTOR

$\bar{A}_D$ - MODIFIED $\bar{A}_{DP}$ WHERE $|\bar{A}_D| = |\bar{A}_T|$ AND IN PLANE OF $\bar{A}_T$ AND $\bar{A}_{DP}$

θ_m - ANGLE THROUGH WHICH $\bar{A}_T$ IS ALLOWED TO ROTATE IN ABGUID COMPUTATION TIME.

$$\hat{U}_1 = \text{UNIT}\ (\bar{A}_T)$$

$$\hat{U}_2 = \text{UNIT}\ (\bar{A}_{DP} \times \hat{U}_1)$$

$$\hat{U}_3 = \hat{U}_1 \times \hat{U}_2$$

$$\bar{A}_D = |\bar{A}_T| \left[\hat{U}_1 \cos(\theta_m) + \hat{U}_3 \sin(\theta_m)\right]$$

$\bar{A}_T = \bar{A}_D$ (FROM PREVIOUS PASS) (17)

$\bar{A}_{DP} = \bar{A}_D$ (PRESENT PASS)

FIGURE 19 VC ROTATION ALGORITHM FOR ABORT GUIDANCE

5.9.2 Abort Targeting

The process of abort targeting is accomplished by the TARGET subroutine. Inputs to TARGET required to perform the targeting function is shown in Table XXI. A functional flow chart is shown in Figure 20. Detailed flow charts and symbol definitions are shown in Appendix A.

Using desired landing site latitude and longitude, the TARGET subroutine provides the radial vector to the landing site in platform coordinates. The computations involve the determination of the magnitude of the desired radial vector. This can be accomplished by using the mean radius of an assumed spherical earth. The transformation is:

$$\left[\overline{R_{LSE}}\right] = \left|\overline{R_{LS}}\right| \begin{bmatrix} \cos\lambda_{LS}\cos\phi_{LS} \\ \cos\lambda_{LS}\sin\phi_{LS} \\ \sin\lambda_{LS} \end{bmatrix}$$

$$\left[\overline{R_{LSI}}\right] = \begin{bmatrix} \cos\omega_e t & -\sin\omega_e t & 0 \\ \sin\omega_e t & \cos\omega_e t & 0 \\ 0 & 0 & 1 \end{bmatrix} \left[R_{LSE}\right]$$

$$\left[\overline{R_{LSP}}\right] = \left[IIP\right] \left[R_{LSI}\right]$$

where

$|\overline{R_{LS}}|$ - magnitude of radial vector to landing site

λ_{LS} - latitude of landing site

ϕ_{LS} - longitude of landing site

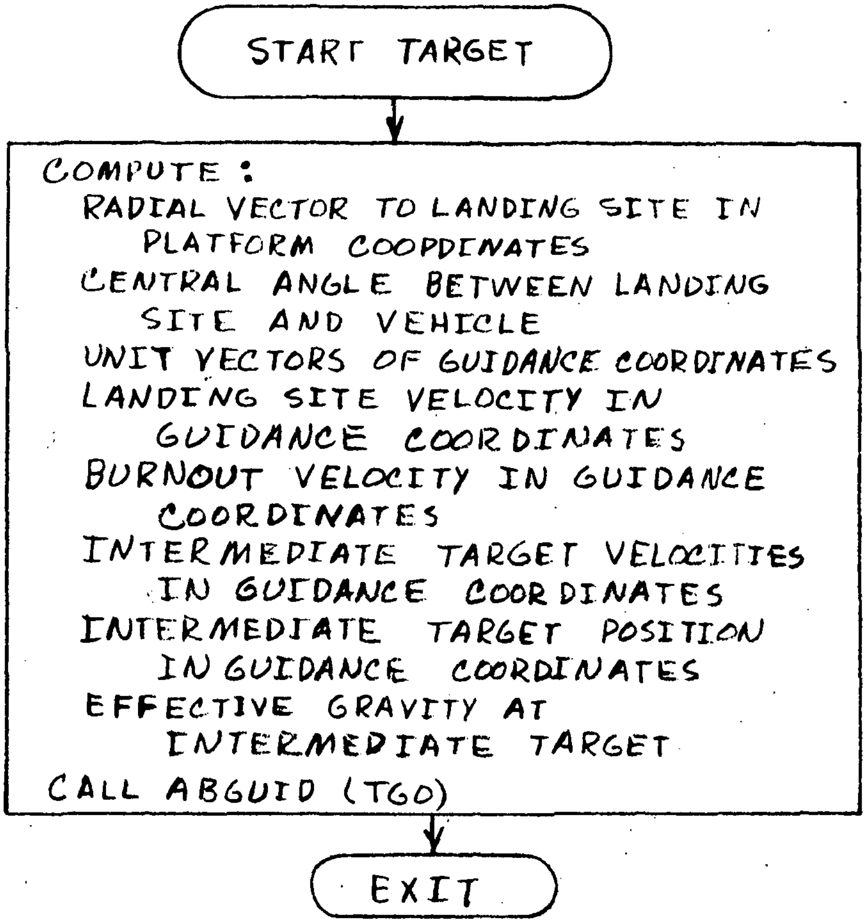

START TARGET
COMPUTE:
RADIAL VECTOR TO LANDING SITE IN PLATFORM COOPDINATES
CENTRAL ANGLE BETWEEN LANDING SITE AND VEHICLE
UNIT VECTORS OF GUIDANCE COORDINATES
LANDING SITE VELOCITY IN GUIDANCE COORDINATES
BURNOUT VELOCITY IN GUIDANCE COORDINATES
INTERMEDIATE TARGET VELOCITIES IN GUIDANCE COORDINATES
INTERMEDIATE TARGET POSITION IN GUIDANCE COORDINATES
EFFECTIVE GRAVITY AT INTERMEDIATE TARGET
CALL ABGUID (TGO)
EXIT

FIGURE 20 FUNCTIONAL FLOW CHART OF TARGET

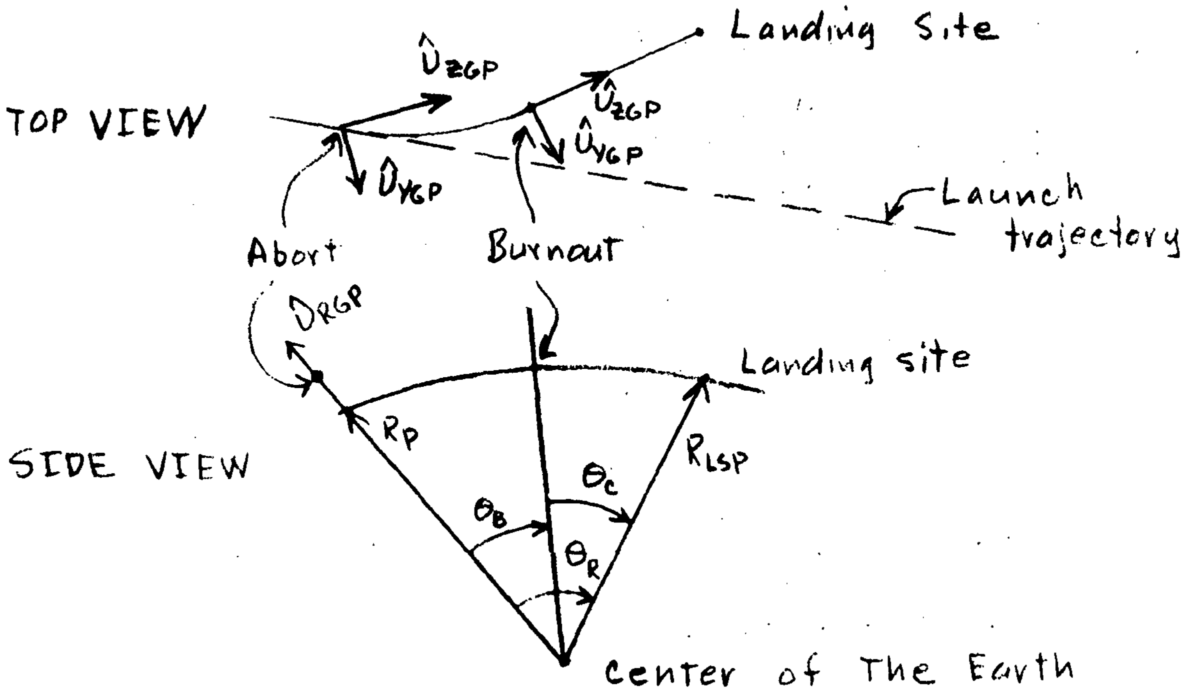


FIGURE 21 UNIT VECTORS DESCRIBING ABORT GUIDANCE COORDINATE SYSTEM

5.9.2 Abort Targeting (Continued)

$\overline{R_{LSE}}$ - Position of landing site in rotating earth coordinates

W_e - Angular velocity of the earth

t - Time since launch

$\overline{R_{LSI}}$ - Position of landing site in inertial earth coordinates

TIP - Transformation matrix provided by INITIA subroutine

$\overline{R_{LSP}}$ - Position of landing site in platform coordinates

The guidance coordinate system is shown in Figure 21 . $\hat{U}_{RGP}$ is a unit vector along the earth radial passing through the vehicle and expressed in platform coordinates. Unit y, $\hat{U}_{RYP}$, is formed by taking the vector cross product of $\overline{R_{LSP}}$ into $\overline{R_p}$ and reducing the result to unit form. The resulting unit vector, $\hat{U}_{ZGP}$, is in a direction which completes the righthanded orthogonal triad and points to the landing site.

The velocity of the landing site due to earth rotation is formed by

$$\overline{V_{LSP}} = \overline{W_{ep}} \text{ X } \overline{R_{LSP}}$$

$$\overline{V_{LSG}} = \begin{bmatrix} \hat{U}_{XGP} \\ \hat{U}_{YGP} \\ \hat{U}_{ZGP} \end{bmatrix} \begin{bmatrix} \overline{V_{LSP}} \end{bmatrix}$$

where

$\overline{W_{ep}}$ - Earth rotation rate in platform coordinates provided by INITIA subroutine

$\overline{V_{LSP}}$ - Landing site velocity in platform coordinates

$\overline{V_{LSG}}$ - Landing site velocity in guidance coordinates

5.9.2 Abort Targeting (Continued)

Figure 22 shows the amount of central angle available for a reentry coast from an intermediate target at an altitude of 200,000 feet. The navigation system can determine what the current central angle between the present position and the landing site is, so the intermediate target velocity and the burn angle must be determined by the target subroutine. It is assumed that the vehicle moves through negligible central angle while changing direction to point toward the intermediate target and that the landing site moves a negligible amount in inertial space during reentry.

The intermediate target positions are computed by setting

$$R_D = |\bar{R}_{LS}| + 200{,}000 \text{ ft.}$$

$$Y_D = 0$$

$$Z_D = 0$$

The intermediate target velocities are provided by

$$\dot{R}_D = 0$$

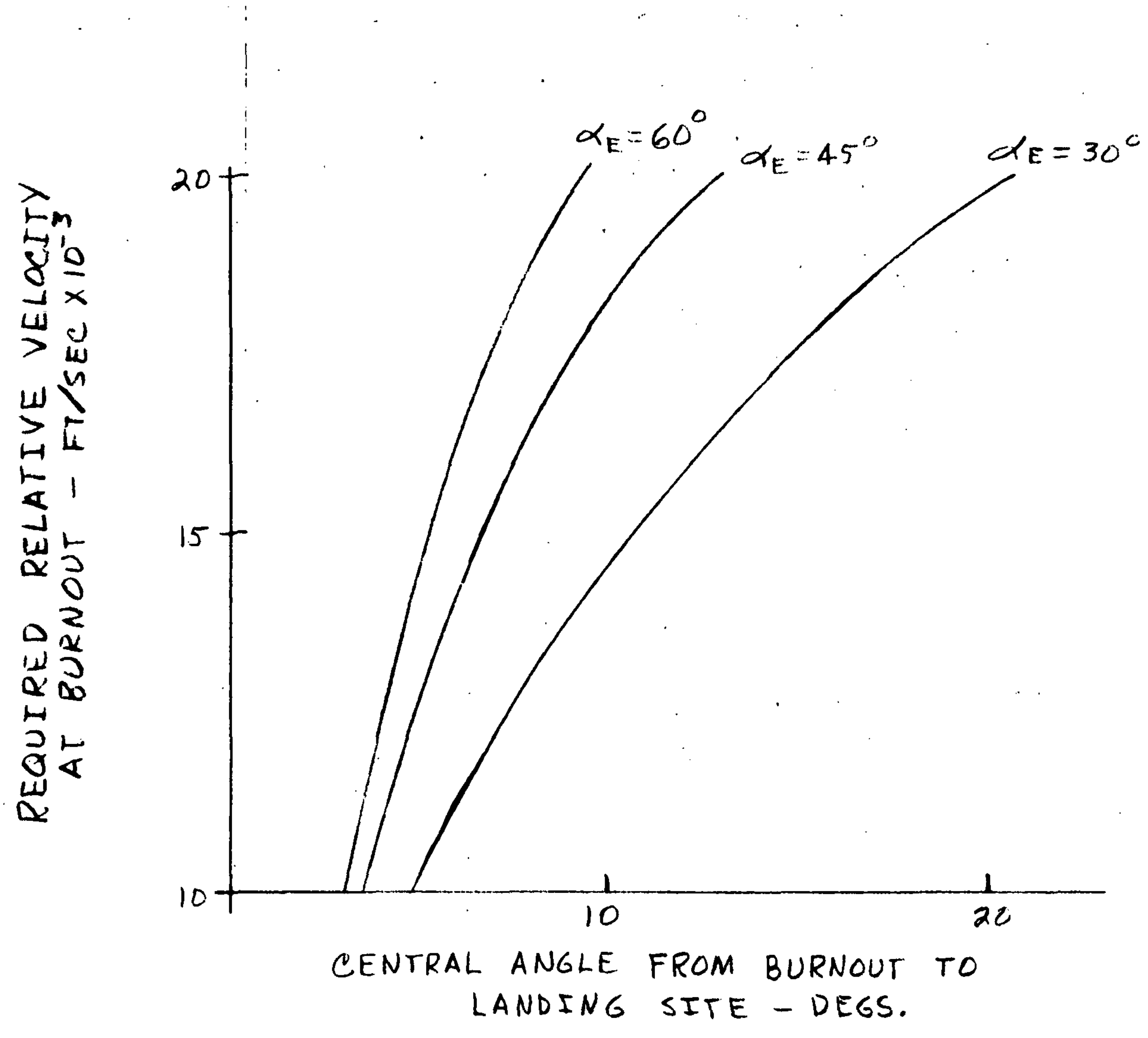
$$\dot{Y}_D = V_{LSG}(2)$$

The other velocity component $\dot{Z}_D$ will require solving the following equations simultaneously

$$V_{BO} = a_0 + a_1 \theta_c + a_2 \theta_c^2 \tag{1}$$

$$\dot{Z}_D = V_{BO} + V_{LSG}(3) \tag{2}$$

$$\theta_c = \theta_R - \frac{T_{GO} \quad (\dot{Z} + \dot{Z}_D)}{2R_E} \tag{3}$$



GRAPH TAKEN FROM REFERENCE 4
ENCLOSURE 2

α_E - ANGLE OF ATTACK HELD DURING REENTRY

FIGURE 22 RELATION BETWEEN COAST ANGLE AND BURNOUT VELOCITY

5.9.2 Abort Targeting (Continued)

where

T_{GO} = Burn time as calculated by guidance

V_{BO} = Velocity between vehicle and landing site at burnout

θ_C = Central angle traveled between burnout and landing site

α_E = Angle of attack during reentry

a_i = Coefficients for curve fitting curves shown in Figure 22.

$\dot{Z}$ = Z component of present velocity in guidance coordinate

θ_R = Central angle between present position and landing site

$\dot{Y}_D$ = Desired target velocity along guidance Y direction

$\dot{Z}_D$ = Desired target velocity along guidance Z direction

Equation (1) was obtained by forming a second-order curve-fit to curves in Figure 22. Equation (2) calculates burn-out velocity in guidance coordinates. Equation (3) calculates the coast angle assuming that the velocity changes linearly between time of abort and the intermediate target and that the tangent of the central angle is approximately equal to the central angle. Since arc length is equal to the product of the included angle and the radius of a circle, the third term in Equation (3) is

$$\theta_B = T_{GO} \frac{(\dot{Z} + \dot{Z}_D)}{2R_E}$$

where θ_B is the burn angle of Figure 21.

5.9.2 Abort Targeting (Continued)

The procedure is to provide an estimate of T_{GO}. Then the coast angle θ_c, and burnout velocity, V_{BO}, are recomputed until the difference between the present V_{BO} and its past value is less than 10. TARGET is then ready for the next computational pass.

The present TARGET configuration programmed in LASS uses

$$\alpha_E =$$

$$V_{BO}(\mathrm{ft/sec}) = V_{BO}\left[g(R_E + H_{BO})\right]^{\frac{1}{2}}$$

$$V_{BO} = V_{BO}\ (25984.5)$$

Abort to launch pad	Aborting downrange
$a_0 = .06236$	$a_0 = .06236$
$a_1 = 2.47813$	$a_1 = 2.47813$
$a_2 = -3.4733$	$a_2 = -3.4733$

Where

α_E = angle of attack desired for reentry

V_{BO} = burnout velocity

g = acceleration due to gravity in $\mathrm{ft/sec}^2$

a_i = curve fit transformation for $\alpha_E = 50^0$

H_{BO} = burnout altitude in ft

Changes in α_E will require corresponding changes in a_i.

This is a quadratic curve fit to the curves shown in Reference 4 and reproduced in Figure 22.

5.9.3 Post Burnout Control

Flow chart and symbol definitions of the PBC subroutine are shown in Appendix A. The PBC subroutine is used to control the vehicle attitude from spacecraft main engine shutdown until it reaches an energy state which allows reentry guidance to take over. The PBC subroutine is shown in functional form in Figure 23 and requires the input information shown in Table XXII. The PBC subroutine outputs are euler attitudes and termination flag. These outputs are required by the AUTOPI and ACC subroutines or AUTOP2 subroutine.

The PBC subroutine is used during the unpowered guidance phase of abort guidance. During this time angle of attack is modulated while maintaining a near zero bank angle. By using this process, violation of vehicle constraints and guidelines is avoided.

There are four basic coordinate systems used in the PBC subroutine. They are the body, platform, wind axis, and local vertical systems. The body system X axis points towards the nose along the main propellant tank centerline, the Z axis points down in the engine gimbal plane, and the Y axis points to the pilot's right completing the right-handed triad. The platform system X axis is perpendicular to the fischer ellipsoid at the place and time of launch, the Y axis is perpendicular to the launch plane, and the Z axis is tangent to the fischer ellipsoid pointing in a northerly direction. The platform system remains fixed in inertial space during the simulation. At the time of launch, the body and platform systems are aligned. The wind axis system has its X axis along the relative velocity vector, the Y axis is perpendicular to a vector from the center of the earth and the velocity vector, and the Z axis completes the right handed triad. The origin of the local vertical system is at the intersection of the fischer ellipsoid with a line. This line is perpendicular to the ellipsoid and passes through the center of the body coordinate system. A segment of this line is the Z axis, which points towards the earth's center. The The X axis is in a plane containing the Z axis and the earth rotation axis and points north. The Y axis completes the right handed triad.

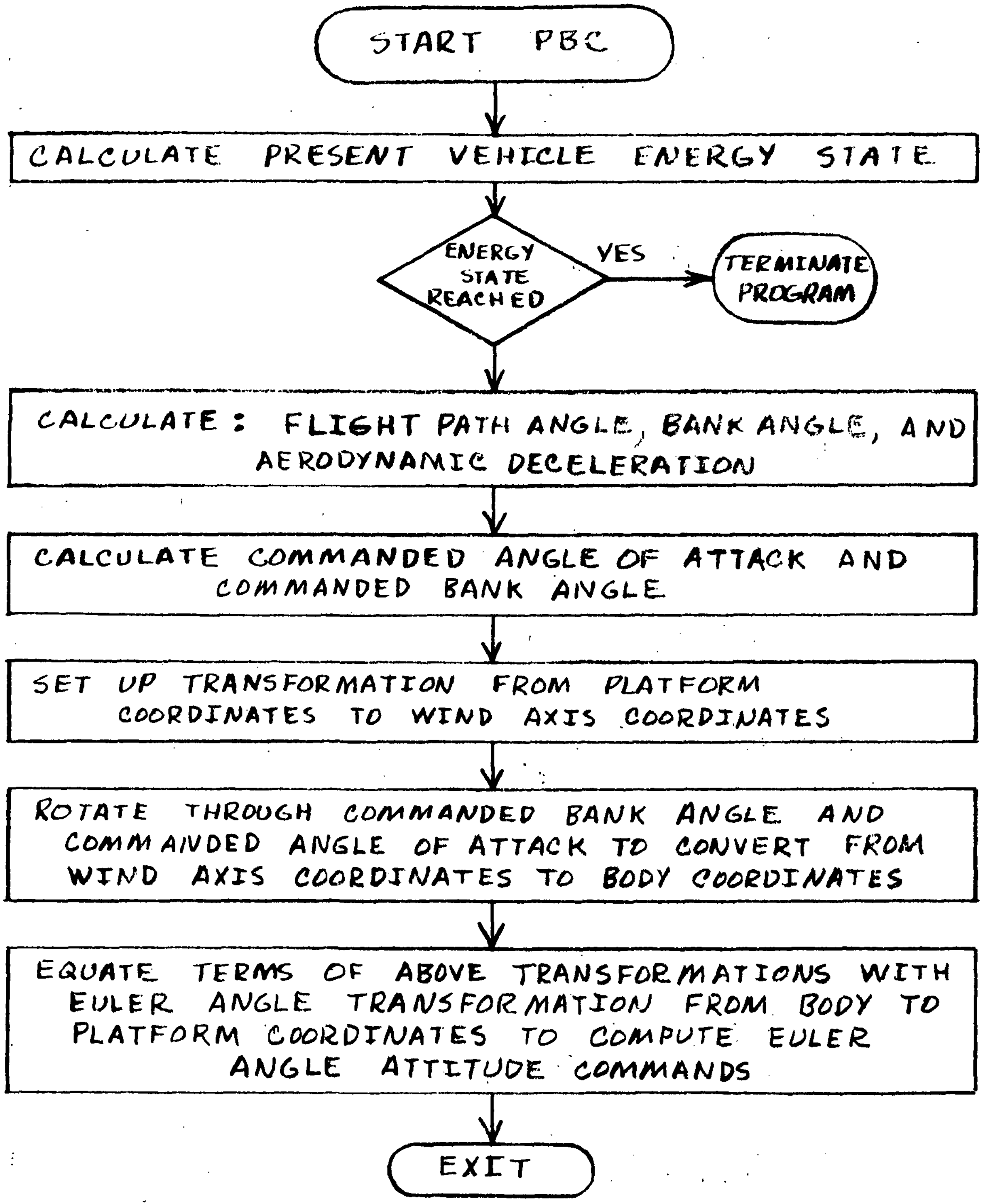

START PBC
CALCULATE PRESENT VEHICLE ENERGY STATE
ENERGY STATE REACHED
YES
TERMINATE PROGRAM
CALCULATE: FLIGHT PATH ANGLE, BANK ANGLE, AND AERODYNAMIC DECELERATION
CALCULATE COMMANDED ANGLE OF ATTACK AND COMMANDED BANK ANGLE
SET UP TRANSFORMATION FROM PLATFORM COORDINATES TO WIND AXIS COORDINATES
ROTATE THROUGH COMMANDED BANK ANGLE AND COMMANDED ANGLE OF ATTACK TO CONVERT FROM WIND AXIS COORDINATES TO BODY COORDINATES
EQUATE TERMS OF ABOVE TRANSFORMATIONS WITH EULER ANGLE TRANSFORMATION FROM BODY TO PLATFORM COORDINATES TO COMPUTE EULER ANGLE ATTITUDE COMMANDS
EXIT

FIGURE 23 FUNCTIONAL FLOW CHART OF PBC SUBROUTINE

5.9.3 Post Burnout Control (Continued)

Post burnout control is initiated by computing a velocity which, coupled with the present orbiter altitude, is equivalent to the desired terminal energy state. Thus,

$$V = [\ V_{PB}^2 - 2G(H-H_{PB})\ ]^{\frac{1}{2}}$$

where: H is present vehicle altitude
V_{PB} and H_{PB} are velocity and altitude at the selected terminal energy state
G is gravitational acceleration

If the vehicle airspeed ($|\overline{V_A}|$) is less than or equal to V_1, the simulation is terminated. If this is not the case, PBC is computed.

The PBC subroutine computes the bank angle of the orbiter. Bank angle is defined as rotation about the orbiter velocity vector. The bank angle, θ_B, is found by equating similar terms of separate transformation from local vertical to body coordinates. The first transformation is

$$\begin{bmatrix} X_{LOCAL\ VERTICAL} \\ Y_{LOCAL\ VERTICAL} \\ Z_{LOCAL\ VERTICAL} \end{bmatrix} = [LV]\ [B] \begin{bmatrix} X_{BODY} \\ Y_{BODY} \\ Z_{BODY} \end{bmatrix} \quad ; \quad [BL1]^T = [LV][B]$$

where [B] is the euler angle transformation between vehicle body and platform coordinates discussed in Section 5.4. The [LV] transformation relates the local vertical coordinate system with the platform coordinate system. The [LV] transformation is found by

$$\overline{LV}(3,j) = -\overline{R}_P / |\overline{R}_P|$$

$$\overline{LV}(2,j) = \overline{LV}(3,j) \times \overline{W}_{EP} / |\overline{LV}(3,j) \times \overline{W}_{EP}|$$

$$\overline{LV}(1,j) = \overline{LV}(2,j) \times \overline{LV}(3,j)$$

5.9.3 Post Burnout Control (Continued)

where:

$\overline{R_P}$ - radial vector to orbiter in platform coordinates

$\overline{W_{EP}}$ - earths' angular momentum vector in platform coordinates

The second transformation is

$$\begin{bmatrix} X_{BODY} \\ Y_{BODY} \\ Z_{BODY} \end{bmatrix} = [\alpha][\beta]^T[\theta_B]\ [\gamma]\ [\theta_H] \begin{bmatrix} X_{LOCAL\ VERTICAL} \\ Y_{LOCAL\ VERTICAL} \\ Z_{LOCAL\ VERTICAL} \end{bmatrix}$$

where:

$[\alpha]$ = rotation thru angle of attack

$[\beta]$ = rotation thru sideslip angle

$[\theta_B]$ = rotation thru bank angle

$[\gamma]$ = rotation thru flight path angle

$[\theta_H]$ = rotation thru heading angle

so, $[BL2] = [\alpha]\ [\beta]^T[\theta_B]\ [\gamma]\ [\theta_H]$

Since corresponding elements of [BL1] and [BL2] are equal,

$$BL1(2,3) = BL2(2,3)$$

$$BL1\ (2,3) = -\sin\gamma\ \sin\beta + \cos\gamma\ \sin\theta_B\ \cos\beta$$

so $$\sin\theta_B = \frac{BL1(2,3)}{\cos\gamma\cos\beta} + \tan\gamma\ \tan\beta$$

5.9.3 Post Burnout Control (Continued)

since

$$BL1\ (2,3) = \vec{U}_{YBP} \cdot \vec{U}_{XP}$$

$$\theta_B = A\ \mathrm{rcsin}\ \frac{\vec{U}_{YPB} \cdot \vec{U}_{XP}}{\cos\gamma \cos\beta} + \tan\gamma \tan\beta$$

This equation for bank angle, θ_B, is used in the PBC subroutine.

During the coasting descent, the vehicle attitudes are controlled to take advantage of aerodynamic generated forces. During this period commanded values of angle of attack and bank angle are used. Thus,

$$\alpha_c = 50^0\ [\ H > 220{,}000 \text{ feet }] \quad , \quad \alpha_c = 25 - \gamma_R[\ A_N < 1.8 \text{ and } H < 220{,}000 \text{ feet}]$$

$$\alpha_c = \alpha - .5(\Delta T)\ [\ A_N > 1.8]$$

$$22^0 < \alpha_c < 50^0$$

$$\theta_B = 0$$

where:

α_C - commanded angle of attack

α - present angle of attack

H - present altitude

γ_R - relative flight path angle

A_N - normal acceleration in gees

ΔT - cycle time of PBC (1 sec)

forms the logic for establishing desired α_C and θ_B . These commands must be converted into euler attitudes for the Ideal autopilot, AUTOP2, to compute restoring torques. This is accomplished by developing the [PU] transformation. The PU transformation is used to convert from platform to wind axis coordinates. The direction cosines are found as follows:

$$\vec{PU1} = \frac{\vec{V}_A}{|V_A|} \quad , \quad \vec{PU2} = \frac{\vec{PU1} \times \vec{LV}(3,j)}{|\vec{PU1} \times \vec{LV}(3,j)|}$$

$$\vec{PU3} = \vec{PU1} \times \vec{PU2} \quad , \quad [PU] = \begin{bmatrix} PU1 \\ --- \\ PU2 \\ --- \\ PU3 \end{bmatrix}$$

5.9.3 Post Burnout Control (Continued)

Then

$$\begin{bmatrix} X_{WIND\ AXIS} \\ Y_{WIND\ AXIS} \\ Z_{WIND\ AXIS} \end{bmatrix} = [PU] \begin{bmatrix} X_{PLATFORM} \\ Y_{PLATFORM} \\ Z_{PLATFORM} \end{bmatrix}$$

and

$$\begin{bmatrix} X_{BODY} \\ Y_{BODY} \\ Z_{BODY} \end{bmatrix} = [\alpha][\theta_B] \begin{bmatrix} X_{WIND\ AXIS} \\ Y_{WIND\ AXIS} \\ Z_{WIND\ AXIS} \end{bmatrix} =$$

$$= [\alpha][\theta_B]\ [PU] \begin{bmatrix} X_{PLATFORM} \\ Y_{PLATFORM} \\ Z_{PLATFORM} \end{bmatrix}$$

Therefore,

$$[\alpha]\ [\theta_B]\ [PU] = [B]^T = [PUC]$$

and

$$-\tan\theta = B(3,1)/B(1,1) = PUC(1,3)/PUC(1,1)$$

$$\sin\psi = B(2,1) = PUC(1,2)$$

$$-\tan\phi = B(2,3)/B(2,2) = PUC(3,2)/PUC(2,2)$$

where θ, ψ, and ϕ are the inertial gimbal angles or euler angles required by AUTOP2 subroutine.

5.10 LASS OUTPUT SUBROUTINE (READO)

A functional flow chart of the READO subroutine is shown in Figure 24. Detailed flow chart and symbol definitions are shown in Appendix A. Program listing of READO, as presently used in LASS, is shown in Appendix B. The READO subroutine performs three functions. The first function provides for the computation of some parameters that may be of interest to an orbiter pilot. The second function collects values of selected parameters, periodically stores these values, and periodically outputs the values in the form of tabulated listings. The third function provides for the periodic collection of selected parameters and, at termination of a LASS run, plots a time history of these parameters.

Many of the selected output parameters are inputted to READO. The subroutine merely changes the engineering units of the parameters for output. A list of inputs to READO are shown in Table XXIII with the subroutine or class of subroutines that computes the parameter. A list of parameters that are tabulated and plotted are shown in Figure A-18 of Appendix A. Many parameters are tabulated as well as plotted.

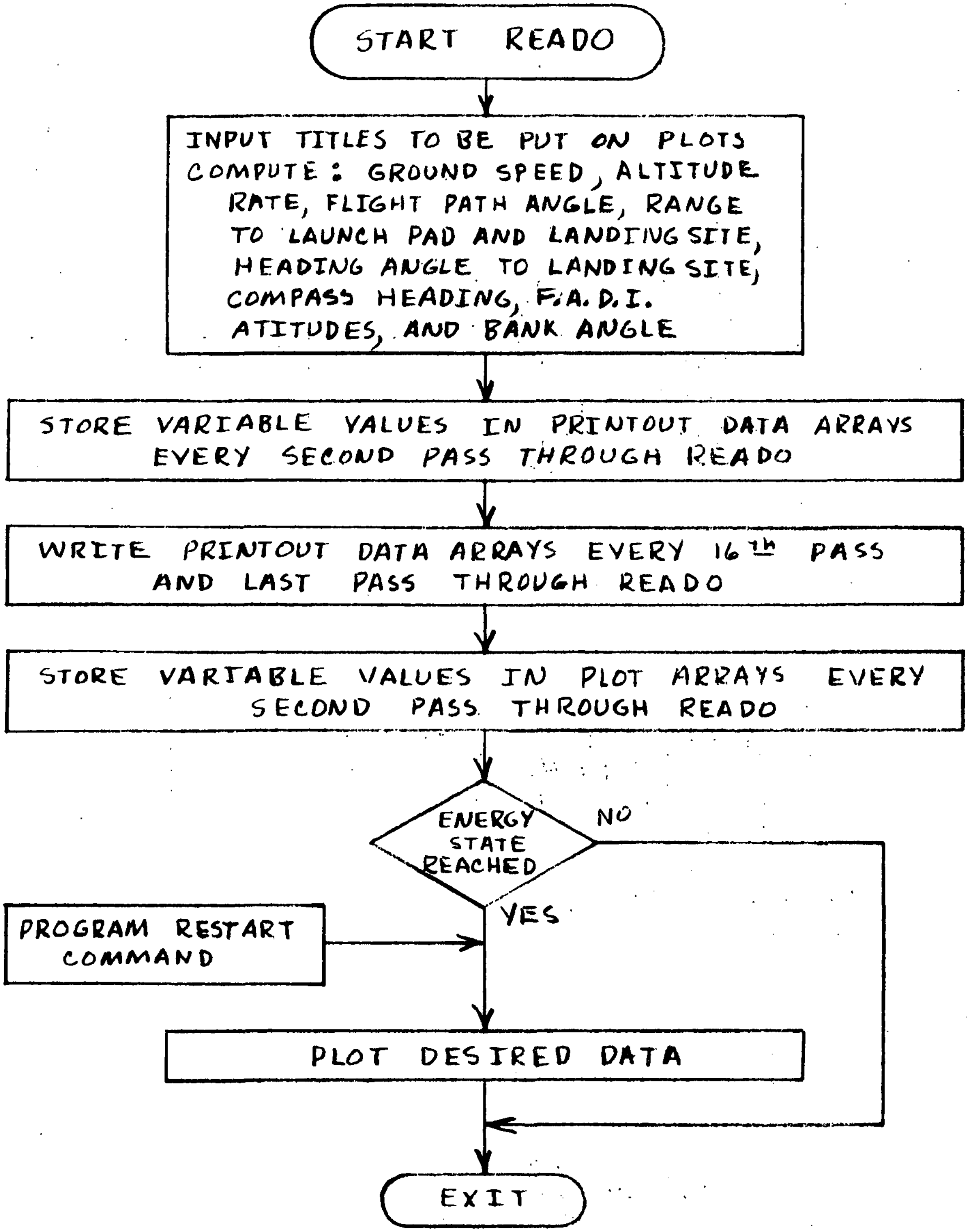

START READO
INPUT TITLES TO BE PUT ON PLOTS COMPUTE: GROUND SPEED, ALTITUDE RATE, FLIGHT PATH ANGLE, RANGE TO LAUNCH PAD AND LANDING SITE, HEADING ANGLE TO LANDING SITE, COMPASS HEADING, F.A.D.I. ATITUDES, AND BANK ANGLE
STORE VARIABLE VALUES IN PRINTOUT DATA ARRAYS EVERY SECOND PASS THROUGH READO
WRITE PRINTOUT DATA ARRAYS EVERY 16th PASS AND LAST PASS THROUGH READO
STORE VARIABLE VALUES IN PLOT ARRAYS EVERY SECOND PASS THROUGH READO
ENERGY STATE REACHED
NO
YES
PROGRAM RESTART COMMAND
PLOT DESIRED DATA
EXIT

FIGURE 24 FUNCTIONAL FLOW CHART OF READO SUBROUTINE

TABLE XXIII INPUT INFORMATION REQUIRED BY READO SUBROUTINE

SYMBOL	SOURCE	NOMENCLATURE
AMT	Autopilot	Vehicle mass remaining
ADPDES	Autopilot	Body pitch attitude error
ADRDES	Autopilot	Body roll attitude error
ADYDES	Autopilot	Body yaw attitude error
ALPHA	AERO	Angle of attack
BETA	AERO	Angle of Sideslip
DFAB(4)	AERO	Forces due to aerodynamics in body frame
DMAXB	AERO	Moment about body x-axis due to aerodynamic forces
DMAYB	AERO	Moment about body y-axis due to aerodynamic forces
DMAZB	AERO	Moment about body z-axis due to aerodynamic forces
EATB(4)	EOM	Acceleration due to thrust in body coordinates
EATT	EOM	Magnitude of acceleration due to thrust
EAVP(4)	EOM	Vehicle acceleration in platform coordinates
EB(3,3)	EOM	Transformation matrix relating body and platform coordinates
EBP(2)	EOM	Engine pitch gimbal angle
EBY(2)	EOM	Engine yaw gimbal angle
ELATC	EOM	Vehicle latitude
ELONGD	EOM	Vehicle longitude
EPB	EOM	Angular rate about body x-axis
EPBD	EOM	Angular acceleration about body x-axis
EPHI	EOM	Actual vehicle euler roll angle
EPSI	EOM	Actual vehicle euler yaw angle
EPT	EOM	Fractional throttle setting
EQB	EOM	Angular rate about body x-axis
EQBD	EOM	Angular acceleration about body y-axis
ERB	EOM	Angular rate about body z-axis

TABLE XXIII INPUT INFORMATION REQUIRED BY READO SUBROUTINE (Continued)

SYMBOL	SOURCE	NOMENCLATURE
ERBD	EOM	Angular acceleration about body z-axis
ERP(4)	EOM	Vehicle position in platform coordinates
ETHETA	EOM	Actual vehicle euler pitch angle
EVP(4)	EOM	Vehicle velocity in platform coordinates
GADP(4)	Guidance	Desired acceleration in platform coordinates
GPHID	Guidance	Desired euler roll angle
GPSID	Guidance	Desired euler yaw angle
GTHED	Guidance	Desired euler pitch angle
H	AERO	Vehicle altitude
IES	FUN	Engine on flag, IES = 1
ISEP	FUN	Vehicle separated flag, ISEP = 1
MACH	AERO	Vehicle Mach number
KGUID	PBC	Simulation terminate flag
QUE	AERO	Dynamic pressure
QDOT	AERO	Aerodynamic heating rate
RAD	INITIA	Conversion degrees to radians
RWVB	READO	Empty weight of booster plus full orbiter
RWVO	READO	Empty weight of orbiter
SEROT	INITIA	Earth's rotation rate
SGRAV	INITIA	Gravitational acceleration at sea level
SLATI	INITIA	Latitude of launch pad
SLONGI	INITIA	Longitude of launch pad
T	FUN	Ground elapsed time
TAG(4)	ABGUID	Desired acceleration in local vertical coordinates
TDVG(4)	ABGUID	Velocity to gained in local vertical coordinates
TGOT	ABGUID	Time left until reaching abort target
THETB	TARGET	Earth's central angle traversed during abort when engines are on

TABLE XXIII INPUT INFORMATION REQUIRED BY READO SUBROUTINE (Continued)

THETC	TARGET	Earth's central angle traversed during abort then orbiter is coasting
THETR	TARGET	Sum of THETB and THETC
TIP (3,3)	INITIA	Transformation matrix relating earth inertial and platform coordinates
TLATI	FUN	Latitude of landing site
TLONGI	FUN	Longitude of landing site
TRLSE(3)	INITIA	Radial to landing site in earth inertial coordinates
TTL	ABGUID	Time until thrust limiting startes
TVBO	TARGET	Relative velocity at burnout
TVDG(4)	TARGET	Target state vector in local vertical coordinates
UXTP(3)	AERO	Unit radial in platform coordinates
UZTP(3)	AERO	Unit vector where plane containing $\overline{UXTP}$ and $\overline{UZTP}$ also contains $\overline{EVP}$
VA(4)	AERO	Relative velocity in platform coordinates
VAB(4)	AERO	Relative velocity in body coordinates
WEP(3)	INITIA	Earth's angular momentum vector in platform coordinates

5.10.1 DISPLAY PARAMETERS PROVIDED BY READO SUBROUTINE

The subroutine is used to compute weight of fuel remaining, inertial flight path angle, relative flight path angle, range from launch pad, range from landing site, and vehicle accelerations. These computations are accomplished in a conventional manner and will not be discussed. However, computations required to obtain compass heading angle, range heading angle, bank angle and flight attitudes are not obvious and will be discussed in some detail.

A coordinate system is errected in READO that is similar to that used in the PBC subroutine. The coordinate system is a local vertical system that is formed by

$$\begin{aligned} \overline{L}_{1,j} &= \text{UNIT } (\overline{R}_P) \\ \overline{L}_{2,j} &= \text{UNIT } (\overline{V}_A \times \overline{L}_{1,j}) \qquad (1) \\ \overline{L}_{3,j} &= \overline{L}_{1,j} \times \overline{L}_{2,j} \end{aligned}$$

where:

$\overline{R}_P$ - Radial vector to vehicle center of gravity in platform coordinates

$\overline{V}_A$ - Vehicle relative velocity in platform coordinates

Compass heading is formed by describing a unit vector pointing east. This is accomplished by

$$\overline{U}_E = \text{UNIT } (\overline{W}_{EP} \times \overline{L}_{1,j})$$

where: $\overline{W}_{EP}$ - earth's angular momentum vector in platform coordinates

The the dot product

$$Q = \overline{V}_A \cdot \overline{U}_E$$

is formed. If Q is less than zero, compass heading is: $\Psi_{CH} = 2\pi - \cos^{-1}(\overline{L}_{2,j} \cdot \overline{U}_E)$

If Q is greater than zero, compass heading is: $\Psi_{CH} = \cos^{-1}(\overline{L}_{2,j} \cdot \overline{U}_E)$

5.10.1 DISPLAY PARAMETERS PROVIDED BY READO SUBROUTINE (Continued)

The range heading angle is the angle between the projections of relative velocity and distance from vehicle to landing site on a local horizontal plane. The coordinate system, shown in equation (1), reflect the vehicle relative velocity orientation with respect to platform coordinates in the unit vectors $\overline{L}_{2,j}$ and $\overline{L}_{3,j}$. The distance between vehicle and landing site is obtained by first describing a radial to the landing site in earth fixed coordinates

$$\overline{R}_{LSE} = (H + R_D) \begin{bmatrix} \cos \lambda_{LS} \cos \Phi_{LS} \\ \cos \lambda_{LS} \sin \lambda_{LS} \\ \sin \Phi_{LS} \end{bmatrix}$$

where:

$H + R_D$ – Constant, sum of vehicle altitude and earth diameter at landing site

λ_{LS} – Latitude of landing site

Φ_{LS} – Longitude of landing site

This is corrected for earth rotation by

$$\overline{R}_{LSI} = \overline{R}_{LSE} \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where: t - ground elapsed time

ω - earth rotation rate

Then $\overline{R}_{LSI}$ is put in platform coordinates using the TIP(3,3) transformation,

$$\overline{R}_{LSP} = [TIP] \; \overline{R}_{LSI}$$

The vector form the vehicle to landing site is

$$\overline{\Delta R} = \overline{R}_{LSP} - \overline{R}_P$$

5.10.1 DISPLAY PARAMETERS PROVIDED BY READO SUBROUTINE (Continued)

This allows us to use equation 1 to compute the range heading angle by

$$\Psi_{RH} = \mathrm{TAN}^{-1}\left[\frac{\overline{\Delta R} \cdot \overline{L}_{2,j}}{\overline{\Delta R} \cdot \overline{L}_{3,j}}\right]$$

The computation for bank angle used the equation developed in section 5.9.3 for post burnout control. The equation is

$$\Phi_{BA} = \mathrm{SIN}^{-1}\left[\frac{-\overline{B}_{j,2} \cdot \overline{L}_{1,j}}{\cos\gamma \ \cos\beta} + \tan\gamma \ \tan\beta\right]$$

where:

γ - relative flight path angle

β - angle of sideslip

$$[\text{ Platform coordinates }] = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} [\text{ Body }]$$

The flight attitudes are found by equating corresponding terms in equivalent matrix transformations. The first transformation is

$$[\text{ Body }] = \begin{bmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \begin{bmatrix} L_{11} & L_{21} & L_{31} \\ L_{12} & L_{22} & L_{32} \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \begin{bmatrix} \text{Local Vertical} \end{bmatrix}$$

The second transformation is

$$[\text{ Body }] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{Local} \\ \text{Vertical} \end{bmatrix}$$

5.10.1 DISPLAY PARAMETERS PROVIDED BY READO SUBROUTINE (Continued)

where:

Ψ - yaw attitude desired

Θ - pitch attitude desired

Φ - roll attitude desired

By making the matrix manipulations and equating like terms, the desired attitudes are:

$$\Theta = \mathrm{Sin}^{-1} \left[\bar{B}_{j1} \cdot \bar{L}_{1j} \right]$$

$$\Phi = \mathrm{Sin}^{-1} \left[-\bar{B}_{j2} \cdot \bar{L}_{1j} / \cos\Theta \right]$$

$$\Psi = \mathrm{Sin}^{-1} \left[\bar{B}_{j1} \cdot \bar{L}_{2j} / \cos\Theta \right]$$

Since $\bar{L}_{2j}$ has a fixed orientation with respect to the projection of the vehicle relative velocity in the local horizontal plane, the yaw angle Ψ requires some modification.

If we let

$$\bar{L}_{2j} = \mathrm{UNIT}\ (\bar{W}_{EP} \times \bar{L}_{1j}) = \bar{U}_E$$

then

$$\Psi = \mathrm{Sin}^{-1} \left[\bar{B}_{j1} \cdot \bar{U}_E \ / \cos\Theta \right]$$

and Ψ is the yaw angle with respect to a north-south line projected in the local horizontal plane.

5.10.2 TABULAR OUTPUTS

A list of tabular outputs are provided in Figure A-18 of Appendix A. Two groups of variables are printed. The first is entitled "Pilot Display Parameters" and contains those parameters thought to be of interest to and orbiter pilot. The second group is entitled "Additional study parameters" and contains those parameters that are helpful when studying the abort problem.

The READO subroutine causes values of parameters to be loaded into data arrays every other pass through READO. This corresponds to a data value every two seconds of ground elapsed time. When the data arrays are full they are printed. This occurs every 16 seconds of ground elapsed time. If LASS should terminate in a normal manner, the remaining unfilled portions of the data arrays are filled with zeros before they are printed. If LASS should terminate, due to some computational error, prior to normal termination, the data arrays being filled at that time are not printed.

All of the printed parameter values are tabulated during the simulated launch and simulated abort phases of the run. The actual value of some parameters may not be of interest during some parts of the simulation. An example is the desired acceleration vector, TAG(4), in local vertical coordinates. Zero values for TAG are tabulated until the abort commences. Values are computed until orbiter engines shutdown. Then the value of TAG computed during the last pass through ABGUID is tabulated during abort coasting phase. Other parameters are printed during periods of the simulation where they are of no interest. An example is orbiter engine gimbal angle EBP and EBY. Gimbal angles of the first two booster

5.10.2 TABULAR OUTPUTS (Continued)

engines are printed until abort commences. Thus, during simulation of launch, these gimbal angle outputs are not used.

5.10.3 OUTPUT PLOTS

A list of parameters to be plotted are provided in Figure A-18 Appendix A. All parameters are plotted as time histories. Time is ground elapsed time starting from launch and continuing until the simulation run is terminated. Actual plotting is accomplished when the LASS run terminates. At this time the parameter values, contained in the plot arrays, are plotted versus time. If LASS should terminate for any reason other than normal attainment of a terminal energy state, values in the plot arrays are plotted.
The master routine, FUN, accomplishes this function. The master routine calls the RECOV subroutine which uses QZKILL subroutine to set a RESTRT flag when a program interrupt is encountered. LASS is restarted to perform the plotting function using the parameter values which were calculated from the beginning of the simulation until the termination occurred.

All plots have the same title. This information is inputted by card into FUN via ITITLE. READO contains the TITA, TITB, and TITC data input statements. These are titles for the ordinate or dependent variable of the plot format. The abscissa or independent variable is time. Data is loaded into the plot arrays every other pass through READO. This is the same as for tabular data and occurs every two seconds of ground elapsed time.

The subroutines performing the plotting functions are adaptations of the standard MSC plot package. The plot grid is set up and labeled using GRID subroutine where

5.10.3 OUTPUT PLOTS (Continued)

the plot grid is scaled according to maximum and minimum variable excursions. Actual plotting of data values is accomplished by using the PLOTIV subroutine. The plot title and identification of plot axes are written on the plots using the RITE2V subroutine, which is part of the $PLOTS package. Actual arguments of these subroutines can be found in program listings of READO found in Appendix B.

At the present time all of the plotting subroutines necessary to perform the plots are on the LASS program tape. Additional fastran files are not required to complete the plotting function provided in LASS.

5.11 UTILITY MATH MODELS

LASS was developed using five utility subroutines. These are:

VECPRD (A, B, C)

DOTPRD (A, B, C)

MAG (A)

UNTVEC (A, B)

MATVEC (K, A, B, C)

The VECPRD (A, B, C) subroutine takes the vector product of two vectors

$$\overline{C} = \overline{A} \text{ X } \overline{B}$$

The DOTPRD (A, B, C) subroutine takes the dot product of two vectors

$$C = \overline{A} \bullet \overline{B}$$

The MAG (A) subroutine returns the magnitude of vector A

$$A(4) = |\overline{A}|$$

The UNTVEC (A, B) subroutine takes vector A, computes its magnitude, and returns vector B as a unit vector. The magnitude of B is not returned.

Subroutine MATVEC (K, A, B, C) performs a matrix product with a vector A. The operation

$$[\overline{C}] = [A][\overline{B}]$$

is formed where A is a 3X3 matrix. If K = 1, the above operation is formed. If K = 2

$$[\overline{C}] = [A]^T [\overline{B}]$$

is formed.

These subroutines are shown in the listings provided in appendix B.