

# LM AGS

## PROGRAMMED EQUATIONS DOCUMENT

### FLIGHT PROGRAM 6

Prepared for:  
National Aeronautics and Space Administration  
Manned Spacecraft Center, Houston, Texas  
Under Contract No. NAS 9-8166

**TRW**  
SYSTEMS GROUP



Don Stever

LM AGS PROGRAMMED EQUATIONS  
DOCUMENT

FLIGHT PROGRAM 6  
(Formerly Flight Program X)

1969 April

Prepared for  
National Aeronautics and Space Administration  
Manned Spacecraft Center, Houston, Texas  
Under Contract No. NAS 9-8166

Approved by: *H. B. Grossman*

H. B. Grossman  
Manager, Data Processing  
Systems Department

Approved by: *T. S. Bettwy*

T. S. Bettwy  
Assistant Manager,  
System Design and  
Analysis Department

Approved by: *T. W. Layton*

T. W. Layton  
Chairman, LM/AGS  
Software Design Review Board



## CONTENTS

1.	INTRODUCTION	1
2.	MISSION DESCRIPTION	3
3.	COMPUTER INPUTS, OUTPUTS, AND OVERALL FUNCTIONAL DIAGRAM	5
4.	INERTIAL REFERENCE FRAMES	11
4.1	PREFERRED ALIGNMENT	11
4.2	NOMINAL ALIGNMENT	11
4.3	LUNAR LANDING AND LAUNCH ALIGNMENT	11
5.	COMPUTATIONAL CYCLES	13
5.1	TIMING OF THE FLIGHT EQUATION COMPUTATIONS	14
5.2	BASIC BLOCK COMPUTATION	15
5.2.1	20-Msec Functions	15
5.2.2	40-Msec Functions	15
5.2.3	2-Sec Functions	16
6.	DESCRIPTION OF THE PROGRAMMED EQUATIONS	17
6.1	20-MSEC COMPUTATIONS	19
6.1.1	Sensor Input Processing	19
6.1.2	Align or Calibration Mode	20
6.1.3	Inertial Reference Mode	23
6.2	40-MSEC COMPUTATIONS	25
6.2.1	Updating Accumulated Velocity and Calculation of Velocity-to-be-Gained	25
6.2.2	Establishing Vehicle Status	26
6.2.3	Steering Modes	27
6.2.4	Steering Mode Decision Logic	30
6.2.5	Engine ON/OFF Logic	31
6.3	2-SEC COMPUTATIONS	32
6.3.1	CSM Navigation	33
6.3.2	LM Navigation	33
6.3.3	Navigation Initialization	34
6.3.4	Navigation Equations Description— Branch 1	35
6.3.5	CSM Orbit Parameters and LM Total Velocity — Branches 2, 3, and 4	37
6.3.6	Altitude Update and Radar Filter — Branches 5, 6, and 7	38
6.3.7	Guidance Equations	42

## CONTENTS (Continued)

7.	FLOW DIAGRAMS OF FLIGHT PROGRAM	61
7.1	DEFINITIONS OF DISCRETES, FLAGS, AND COUNTERS	132
7.1.1	Discretetes	132
7.1.2	Internal Decision Logic Flags	133
7.1.3	Counters	136
7.2	PROGRAM SUBROUTINES	137
7.2.1	Normalize	137
7.2.2	Square Root	138
7.2.3	Sine-Cosine	139
7.2.4	Arctangent	141
7.2.5	Vector Cross Product	143
7.2.6	Normalize Vector	144
7.2.7	Double Precision Vector Magnitude	145
8.	DESCRIPTION OF AUXILIARY ROUTINES	147
8.1	GROUND SUPPORT EQUIPMENT (GSE) SERVICE ROUTINE FOR THE AEA FLIGHT PROGRAM	147
8.1.1	Use of GSE Discretetes	147
8.1.2	Flow Diagram of GSE Routine	148
8.1.3	Tape Load Startup Checksum	148
8.2	LM AGS TELEMETRY	152
8.2.1	Description of Telemetered Quantities	152
8.2.2	Program Logic for Telemetry	161
8.2.3	Telemetry List for Flight Program 5	161
8.3	INFLIGHT SELF-TEST ROUTINE	166
8.3.1	Test Operation	167
8.3.2	Logic Test	167
8.3.3	Memory Test	167
8.4	DATA ENTRY AND DISPLAY ASSEMBLY (DEDA)	170
8.4.1	DEDA Functions	170
8.4.2	PGNCS Downlink Input Routine	173
8.4.3	Relation of DEDA to AGS System	176
8.5	POWER TURN ON SEQUENCE	185
9.	DEDA ACCESSIBLE PARAMETERS	189
10.	DEFINITIONS OF CONSTANTS	201
	REFERENCES	208
	APPENDIX A. FLIGHT PROGRAM LISTING—LM AGS FP6 S03	209

## ILLUSTRATIONS

2.1	Coelliptic Rendezvous Flight Profile	3
3.1	AGS Functional Block Diagram	6
4.1	Vehicle Reference Axes	12
5.1	2-Sec Computational Cycle	13
6.1	IMU Euler Angles	21
6.2	Accumulated velocity and Velocity-to-be-Gained Diagram	25
6.3	Evaluation of Gravity Vector	36
6.4	Altitude Rate Versus Final Radial Distance	49
6.5	CSI to CDH Geometry	51
7.1	AGS Total Software System Block Diagram	63
7.2	Start of 20-msec Computations, Input Data	65
7.3	Gyro Compensation	67
7.4	Direction Cosine Change	69
7.5	Direction Cosine Update and Velocity Resolution	71
7.6	Processing of PGNCS Angles, Body Axis Align Computations	73
7.7	40-msec Attitude Error Computations	75
7.8	Engine Discrete Logic; Direction Cosine Normality and Orthogonality Computations	77
7.9	Lunar Align Computations	79
7.10	FDAI Computations	81
7.11	LM Initialization, Branch IC1	83
7.12	LM State Vectors, Branch IC2, Gravity and Display Computations, IC3	85
7.13	Navigation Update Equations, Branch 1	87
7.14	Computation of CSM Orbit Parameters from Ephemeris Data, Branch 2	89
7.15	Prediction of Present CSM Position and Velocity, Branch 3	91
7.16	Total Velocity, Altitude Update, Range and Range Rate	93
7.17	Covariance Matrix Update and Processing of Range and Range Rate Inputs	95
7.18	LM State Vector and Covariance Matrix Update Following a Radar Input	97
7.19	LM Orbit Parameters, Branches 8, 9, and 10	99
7.20	Guidance Selection Logic, Branch 11	101
7.21	LM Prediction	103
7.22	CSM Prediction	105

## ILLUSTRATIONS (Continued)

7.23	Central Angle/Line of Sight Angle at $t_{ig}$ and CSM at CDH/Rendezvous	107
7.24	Prediction of Transfer Orbit Vectors and Central Angle Sine and Cosine	109
7.25	p-Iteration Calculations	111
7.26	Transfer and Braking Impulse Computations	113
7.27	Total Velocity for TPI	115
7.28	Computation of $V_G$ Components for CSI, CDH, and OI	117
7.29	Guidance Law	119
7.30	Pitch and Yaw Steering Equations Velocity-to-be-Gained Equations	121
7.31	The Executive Branch, Branch 50	123
7.32	Branch Control	125
7.33	Orbit Parameters and Ellipse Prediction Subroutines	127
7.34	Gravity Subroutine	129
7.35	Display Routine: Altitude Rate and Out of CSM Plane Velocity	131
7.36	Normalize Subroutine	137
7.37	Square Root Subroutine	138
7.38	Sine - Cosine Subroutine	140
7.39	Arctangent Subroutine	142
7.40	Cross Product Subroutine	143
7.41	Normalize Vector Subroutine	144
7.42	Vector Magnitude Subroutine	145
8.1	GSE (Ground Support Equipment) Service Routine	151
8.2	AGS Telemetry Subroutines	163
8.3	Inflight Self-Test Routine	169
8.4	PGNCS Downlink Input Subroutine	175
8.5	DEDA Processing Routine	181
8.6	40-msec Sampling of Discrete Word Two	183
8.7	Power Turn-On Startup Sequence	187

## TABLES

8.1	Bit Pattern, Guidance Selector $S_{10}$	155
8.2	Bit Pattern, Self-Test Indicator $S_{12}$	157
8.3	Bit Pattern, Mode Word $S_{00}$	159
8.4	Bit Pattern, Discrete Input Word Number One	160
8.5	AEA Telemetry Word List	164
8.6	Time Required to Complete Self-Test Following the Entry of $S_{12} = 0$ via DEDA	166
9.1	DEDA Accessible Region	192
9.2	DEDA Inputs	193
9.3	DEDA Outputs	196
9.4	"S" Selectors	199
10.1	Equation Constants	202

## 1. INTRODUCTION

This document describes the Flight Program 6 (FP6) equations that are programmed for the Abort Electronics Assembly (AEA) of the Abort Guidance Section (AGS). The functional requirements for this program are given in Reference 1, and the operating procedures are given in Reference 2.

The document is basically composed of the following parts:

- 1) A brief nontechnical description of the equations and a summary of the logic flow. (For derivations of the equations, see References 3, 4, and 5.)
- 2) The equations and logic flow charts programmed in FP6.
- 3) The description of the auxiliary routines: Ground Support Equipment (GSE) Service Routine, Telemetry Routine, Inflight Self-Test Routine, Data Entry and Display Assembly (DEDA) Routine, and the Power Turn-On Routine.
- 4) A list of program constants and a list of selected DEDA accessible parameters.
- 5) Program listing.

The LM Abort Guidance Section (AGS) is a strapdown inertial navigation and guidance system used primarily to effect rendezvous of the LM with the CSM in the event of a Primary Guidance and Navigation Control System (PGNCS) failure. The AGS consists of the following three assemblies:

- a) Abort Electronics Assembly (AEA)
- b) Data Entry and Display Assembly (DEDA)
- c) Abort Sensor Assembly (ASA)

The AEA is a general-purpose digital computer located in the LM aft equipment compartment used for the data processing function of the AGS. It is a fixed-point, 18-bit machine with 17 value bits and a sign bit using 2's complement arithmetic. The AEA memory is of conventional ferrite core construction with a 4096-word capacity. Half the memory is hardwired at manufacture and the other half is programmable.

The DEDA is a general-purpose astronaut input-output device for the AGS located on the right-hand side of the LM cockpit. It consists of a keyboard, electroluminescent address, and digital data displays.

The ASA consists of three strapped-down pendulous accelerometers, three strapped-down gyros, and associated electronic circuitry. It is located on the LM navigation base above the flight crew. The ASA provides the AEA with incremental angular rotation information about the vehicle X, Y, and Z axes and incremental velocity changes along the vehicle X, Y, and Z axes. These incremental data are in the form of pulses and are utilized in the AEA to maintain knowledge of the LM vehicle attitude, position, and velocity.

## 2. MISSION DESCRIPTION

The flight equations are mechanized to compute any of the maneuvers comprising the coelliptic flight plan. Figure 2.1 illustrates all the various maneuvers required in a nominal coelliptic rendezvous flight profile from the lunar surface.

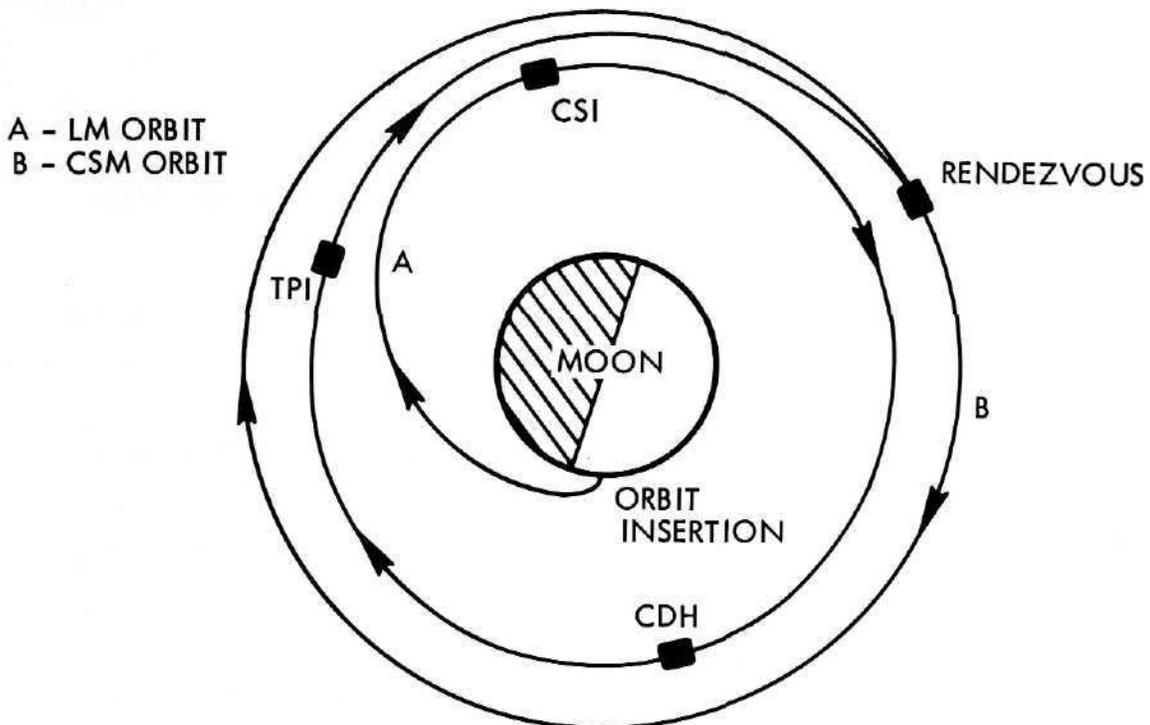


Figure 2.1. Coelliptic Rendezvous Flight Profile

The LM initiates the rendezvous sequence with an orbit insertion maneuver at a time chosen to produce the desired phasing with the CSM, which is nominally in a circular orbit. The orbit insertion maneuver is targeted to drive the LM to a desired altitude, altitude rate and horizontal velocity with zero crossplane velocity. The LM then coasts until apocynthion where the Coelliptic Sequence Initiate (CSI) burn is performed.

The CSI calculation determines the magnitude of the horizontal burn to be performed parallel to the CSM orbit plane at CSI time in such a manner that, following the coelliptic maneuver (CDH), the desired line-of-sight angle between LM and CSM will be achieved at nominal TPI time.

The coelliptic (CDH) burn is performed either 1/2 or 3/2 orbital periods after CSI and makes the LM orbit coelliptic with the CSM orbit.

The Terminal Phase Initiate (TPI) burn is performed at the time the desired line of sight ( $\theta_{LOS}$ ) between the LM and the CSM is achieved. The transfer time is specified and the rendezvous point is the predicted position of the CSM (or offset from it). By means of an iteration on the semi-latus rectum ( $p$ ) of the transfer orbit, the orbit which passes from the present position to the rendezvous position in the specified time is determined. Once the orbit is determined, the velocity impulse needed to achieve the desired trajectory is calculated. Mid-course maneuvers are calculated in the same manner.

For a lunar mission, the aborts may occur anytime after separation of the LM from the CSM. The sequence of events followed after the abort depends upon the abort situation. That is, if the abort occurs when the LM is near the lunar surface, the above sequence of events would be desired. If the abort occurs when the LM is high above the lunar surface, it may be desirable to begin immediately with the coelliptic sequence. Finally, there is always the possibility of doing a direct transfer to rendezvous. This can be accomplished by utilizing the TPI modes available. Built-in mission planning capabilities are available when these modes are employed. These capabilities and some restrictions are discussed in Paragraph 6.3.7.5.

### 3. COMPUTER INPUTS, OUTPUTS AND OVERALL FUNCTIONAL DIAGRAM

An overall functional block diagram of the LM/AGS is shown in Figure 3.1.

The functions of the computer program are attitude reference, navigation, abort guidance including attitude commands, Engine ON/OFF command, and data display. The flow of the program and, in particular, the above functions are controlled by the following inputs. A brief description of each input is given below; a more detailed description is given where each input is referenced in Section 6.

#### AGS Sensors

$\Delta\alpha_{Xi}, \Delta\alpha_{Yi}, \Delta\alpha_{Zi}$	Sensed gyro inputs about LM X, Y, and Z body axes, respectively. These inputs are in the form of pulses, and 1 pulse nominally represents an angular increment of $2^{-16}$ rad.
$\Delta V_{Xi}, \Delta V_{Yi}, \Delta V_{Zi}$	Accelerometer inputs along LM X, Y, and Z body axes, respectively. These inputs are in the form of pulses, and 1 pulse nominally represents 1/320 ft/sec.

#### Primary Guidance and Navigation System

$\theta_{pi}, \psi_{pi}, \phi_{pi}$	PGNCS Euler angles. These inputs are in the form of pulses, and 1 pulse represents $2\pi \times 2^{-15}$ rad.
$\underline{r}_L, \underline{V}_L, t_L, \underline{r}_E, \underline{V}_E, t_E$	PGNCS supplied ephemeris data: L signifies LM, E signifies CSM.

#### Radar Data

There is no hardware connection between the radar and the AEA; thus, the radar data is input to the AEA via DEDA into J<sup>18</sup> and J<sup>17</sup>.

J <sup>18</sup>	Relative range from CSM to LM at the time of taking the radar measurement.
J <sup>17</sup>	Range rate between CSM and LM.

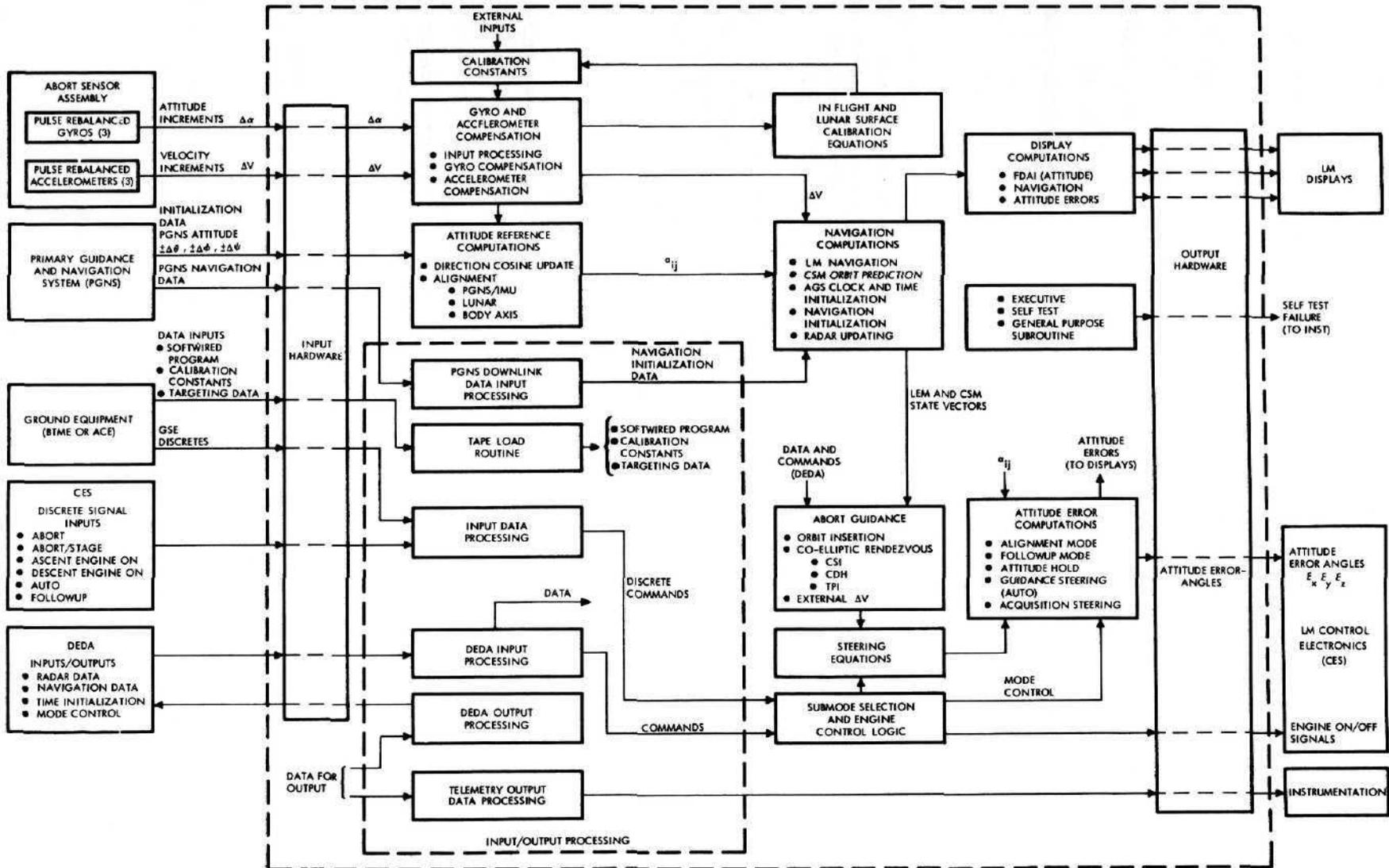


Figure 3.1. AGS Functional Block Diagram

Discrete Signals

Six discrettes are received from the Control Electronic Section (CES). These discrettes, designated with the symbol  $\beta$  and explained below, are automatically received by the AGS when the appropriate circumstances exist.

$\beta_1$             Descent Engine Discrete. This discrete has the logic value 0 if the descent engine is off and the logic value 1 if the descent engine is on.

$\beta_2$             Ascent Engine Discrete. This discrete has the logic value 0 if the ascent engine is off and the logic value 1 if the ascent engine is on.

Follow-Up Discrete. The follow-up signal,  $\beta_3 = 1$ , is transmitted to the AGS under either of the following conditions:

- a. Guidance control switch in PGNCS position
- b. Mode control switch in "ATT HOLD" and attitude controller out of the detent position.

If neither a nor b is true, then  $\beta_3 = 0$ .

When the follow-up signal is present ( $\beta_3 = 1$ ), all engine commands and attitude error signals issued by the AGS to the CES are inhibited from controlling the main engines or RCS thrusters.

$\beta_4$             Automatic Discrete. The presence of the automatic discrete in the AEA is completely controlled by the mode control switch.

<u>Mode Control Switch Positions</u>	<u>Automatic Discrete</u>
OFF	Absent ( $\beta_4 = 0$ )
ATT HOLD	Absent ( $\beta_4 = 0$ )
AUTO	Present ( $\beta_4 = 1$ )

The presence of the automatic discrete is one of the necessary conditions for the AGS to exert guidance steering (not merely attitude hold) and engine control.

- $\beta_5$       Abort-Stage Discrete. Depression of the abort-stage button on the instrument panel applies the abort-stage signal ( $\beta_5 = 1$ ) to the AEA. A second depression of the abort-stage button removes the abort-stage signal ( $\beta_5 = 0$ ).
- $\beta_6$       Abort Discrete. Depression of the abort button on the instrument panel applies the abort signal ( $\beta_6 = 1$ ) to the AEA. A second depression of the abort button removes the abort signal ( $\beta_6 = 0$ ).

### DEDA Control Switches

The DEDA control switches are entries by the astronaut that select the various options available in the program.

#### AGS Function Selector ( $S_{00}$ )

$S_{00} = 0$	Attitude Hold	}	Inertial Reference Mode
$S_{00} = 1$	Guidance Steering		
$S_{00} = 2$	Z Axis Steering		
$S_{00} = 3$	PGNCS/AGS Align	}	Alignment and Calibration Mode
$S_{00} = 4$	Lunar Align		
$S_{00} = 5$	Body Axis Align		
$S_{00} = 6$	Calibration		
$S_{00} = 7$	Inflight Accelerometer Calibration		

#### External $\Delta V$ Reference ( $S_{07}$ )

$S_{07} = 0$	Do not freeze velocity-to-be-gained vector or accumulate sensed velocity increments.
$S_{07} = 1$	Freeze velocity-to-be-gained vector in AGS inertial frame and accumulate sensed velocity increments.

Guidance Mode Selector ( $S_{10}$ )

$S_{10} = 0$	Orbit Insertion
$S_{10} = 1$	CSI
$S_{10} = 2$	CDH
$S_{10} = 3$	TPI Search
$S_{10} = 4$	TPI Execute
$S_{10} = 5$	External $\Delta V$

Engine Select ( $S_{11}$ )

$S_{11} = 1$	Select cant angle corrections to guidance steering attitude errors
$S_{11} = 0$	Do not select cant angle corrections

Inflight Self-Test Control ( $S_{12}$ )

$S_{12} = 0$	Entry-initialize test, Readout-test not complete
$S_{12} = 1$	Test successfully completed
$S_{12} = 3$	Logic test failure
$S_{12} = 4$	Memory test failure
$S_{12} = 7$	Logic and memory test failure

Store Landing Azimuth and Set Lunar Surface Flag ( $S_{13}$ )

$S_{13}$	Any entry into $S_{13}$ will perform function
----------	---

Navigation Initialization ( $S_{14}$ )

$S_{14} = 0$	Initialization complete
$S_{14} = 1$	Initialize LM and CSM using PGNCS data
$S_{14} = 2$	Initialize LM using DEDA data
$S_{14} = 3$	Initialize CSM using DEDA data

Radar Data (S<sub>15</sub>)

S<sub>15</sub> Any entry into S<sub>15</sub> will store LM Z axis direction cosines, computed range vector, range rate, and the time since the last radar range update

CDH Selection (S<sub>16</sub>)

S<sub>16</sub> = 1 Compute CDH solution at CSI time plus 1/2 LM orbit period

S<sub>16</sub> = 3 Compute CDH solution at CSI time plus 3/2 LM orbit period

Radar Filter Initialization (S<sub>17</sub>)

S<sub>17</sub> = 0 Normal readout

S<sub>17</sub> = 1 Entered to initialize radar filter

Submodes of Z Axis Steering (S<sub>507</sub>)

S<sub>507</sub> = 0 Direct the LM Z body axis toward the AGS indicated position of the CSM

S<sub>507</sub> = 1 Orient the LM Z body axis to the desired thrust direction

Submodes of Guidance Steering (S<sub>623</sub>)

S<sub>623</sub> = 0 Orient the Z body axis parallel to the CSM orbit plane

S<sub>623</sub> = 1 Orient the Z body axis parallel to the plane defined by the unit vector  $\underline{W}_b$

The primary outputs of the LM AGS are the attitude error signals  $E_X$ ,  $E_Y$ ,  $E_Z$  and the engine ON/OFF signals. In addition, many quantities are available for information via the DEDA and telemetry and displays. See Sections 8 and 9 for a more detailed explanation of DEDA and telemetry and their quantities.

#### 4. INERTIAL REFERENCE FRAMES

There are many possible inertial reference frames which may be established during a mission. It may be assumed that, for any of the three attitude reference alignment capabilities, the inertial reference frame established will be such that the x-z plane will lie within 10 deg of the CSM orbit plane. Since the preferred method of alignment is the PGNCS to AGS Alignment, the inertial reference frames established for given periods may be expected to correspond to the orientations established for the IMU stable member. These orientations are discussed in Paragraphs 5.1.4.2 and 5.6.3.4 of Reference 6. The descriptions given below correspond to those given in the reference. For all of these frames, the origin is at the center of the attracting body.

##### 4.1 PREFERRED ALIGNMENT

For certain thrusting maneuvers, the x inertial axis will be established to be parallel to the desired thrusting direction at the instant of ignition as determined by the PGNCS. The y inertial axis will be determined by forming the vector product of the desired thrusting direction and the position vector provided these two vectors are not essentially parallel. In the event the desired thrusting direction is parallel to the position vector, the y inertial axis will be determined by forming the vector product of the desired thrusting direction and the velocity vector. The z inertial axis will complete the right-hand orthogonal triad.

##### 4.2 NOMINAL ALIGNMENT

The nominal alignment for orbital operations will establish the x inertial axis to be along the position vector at a given instant, positive outward from the center of the attracting body. The inertial y axis will lie along the LM orbit angular momentum vector, positive in the direction opposite to the momentum. The inertial z axis will complete the right-hand orthogonal triad.

##### 4.3 LUNAR LANDING AND LAUNCH ALIGNMENT

For the lunar landing mission, the x inertial axis will be established to be along the designated landing site position vector at the nominal landing time, positive outward from the lunar center. The vector product of

the CSM orbit angular momentum vector and the x inertial axis will define the z inertial axis. The y inertial axis will complete the righthand orthogonal triad.

After landing and re-alignment, the inertial reference frame will be similar to that given above except that the x inertial axis will be defined to pass through the actual landing site at the nominal time of launch.

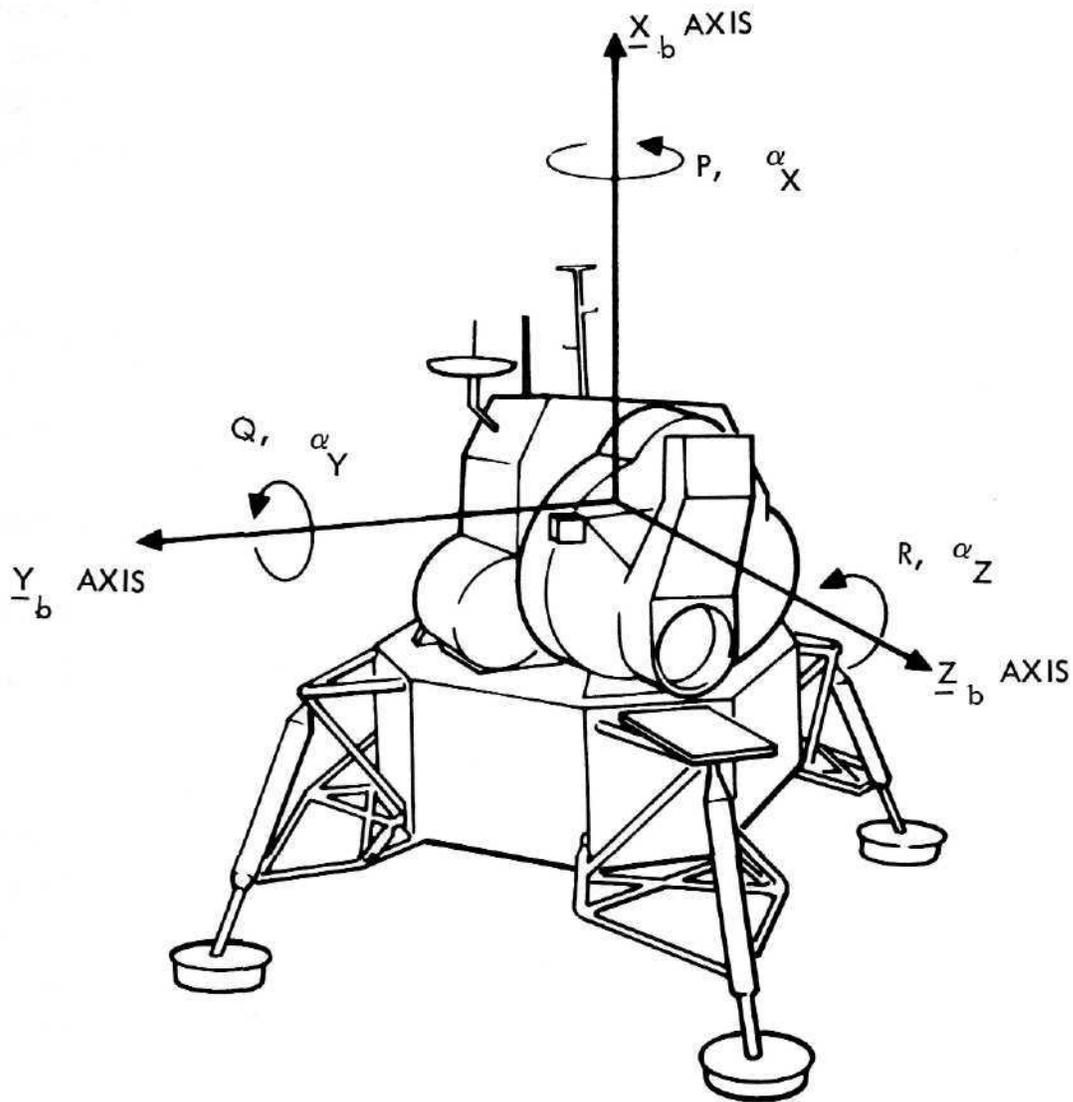


Figure 4.1. Vehicle Reference Axes

## 5. COMPUTATIONAL CYCLES

The LM AGS major computational cycle is 2 sec in duration. In this period of time, the equations update navigation, solve the rendezvous guidance problem, perform inflight self test, and if requested, establish the inertial frame of reference. This major 2-sec cycle is broken into 100 segments. Each segment takes 20 msec of time to complete.

Every 20-msec segment is divided into two parts. The first portion of the 20-msec segment contains equations called "20-msec computations" which must be calculated anew every 20 msec. The second portion of the 20-msec time segment is reserved for the calculation of either of two different sets of equations.

One set of equations must be completely recalculated every 40 msec. This set is called the "40-msec computations." This name is not to be construed to mean that the calculations take 40 msec to complete. This set of equations is recalculated in the second part of alternate 20-msec time segments, thus assuring that this set of equations is recalculated every 40 msec.

The other set of equations must be newly calculated every 2 sec. These equations are called the "2-sec computations," and are too large to fit into the second portion of the 20-msec time segment. Because of this fact, the 2-sec computations are split into smaller groups of equations. Each group of equations is small enough to be calculated in one of the remaining 50 parts of the 20-msec time segment.

Figure 5.1 illustrates the segmentation of the major 2-sec computational cycle.

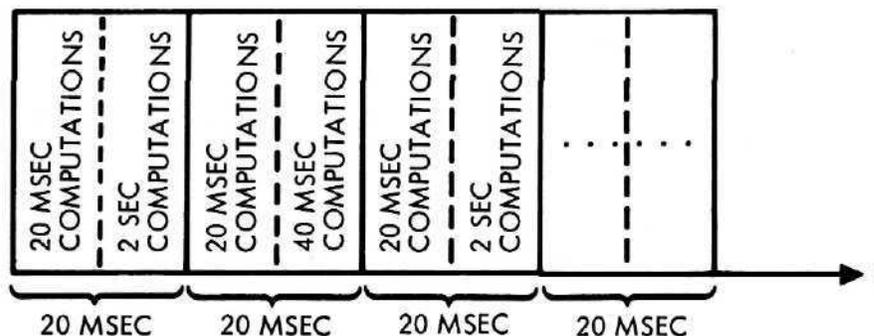


Figure 5.1. 2-Sec Computational Cycle

### 5.1 TIMING OF THE FLIGHT EQUATION COMPUTATIONS

The programmed equations consist of a 20- and a 40-msec section, a 2-sec section, and basic and auxiliary hardwired subroutines. Figure 7.1 shows the relative program flow among these sections but does not show the basic subroutines.

When the computer is first turned on, an initial startup routine is executed. This routine selects Branch 50 as the first 2-sec branch to be executed and zeroes  $\mu_{10}$ . Branch 50 will select Branch 1 as the next 2-sec cycle branch to be executed. The counter  $\mu_{10}$  routes the program flow to Branch 50 every 50<sup>th</sup> 2-sec cycle and controls the initialization of the output telemetry routine, which is done every 25<sup>th</sup> 2-sec cycle.

After executing start up, the program proceeds to DELAY where it waits until an internal clock pulse signals that 20 msec have passed since the last pulse.

At this time, the 20-msec computations are performed. Then the program tests FLAG1. If FLAG1 is zero, it is set to 1 and the 40-msec computations are performed, after which the program returns to DELAY. If FLAG1 is one, it is zeroed and one branch of the 2-sec computations is executed. In this manner, the 40-msec and 2-sec computations alternate every 20 msec.

The 2-sec computations are divided into 50 2-sec branches. Branch 50 is executed every 50<sup>th</sup> branch and normally routes the 2-sec logic to the first guidance equation branch, Branch 1. Branch 1 in turn selects Branch 2 as the next 2-sec cycle, and this process continues until all the branches of the selected guidance mode (determined by the guidance mode selector switch  $S_{10}$ ) have been executed. The remaining branches are used to execute the self-test routine until  $\mu_{10}$  has counted 49 branches, after which Branch 50 is again selected.

If an initialization has been commanded (see Subsection 6.3), the three initialization branches are sequentially selected followed by 42 self-test branches, 4 DEDA branches, and Branch 50.

Four times per two seconds (during branches 12, 24, 36, and 48), the DEDA routine is selected and DEDA input/output functions are performed if requested. The DEDA routine is selected by the counter  $\mu_{20}$ . When  $\mu_{20} = 11$ , the selected branch control is saved; the DEDA routine restores the branch control and zeros  $\mu_{20}$ .

After the current branch has been executed and the routing for the next branch selected, the program flow proceeds to EXEC, where  $\mu_{20}$  is tested and incremented. Then, the display routine is executed and  $\mu_{10}$  routes the program as discussed above. The program then proceeds to DELAY.

## 5.2 BASIC BLOCK COMPUTATION

The major functions performed in each of the three different computational blocks are listed below.

### 5.2.1 20-Msec Functions

Those functions performed every 20 msec consist of:

- 1) Gyro data processing and compensation (corrections)
- 2) Accelerometer data processing and compensation
- 3) Attitude direction cosine updating
- 4) PGNCS downlink data input routine
- 5) Telemetry output
- 6) PGNCS/AGS or body axis align computations.

### 5.2.2 40-Msec Functions

Those functions performed every 40 msec consist of:

- 1) Thrust and velocity vector incrementing
- 2) Attitude control and engine ON/OFF selection logic
- 3) Output AGS attitude error signals
- 4) Output AGS main engine commands
- 5) Computation and output to the instrument panel of FDAI angles
- 6) Lunar align computations when commanded

- 7) DEDA and external discrete sampling (CES, GSE)
- 8) Normality and orthogonality corrections for direction cosines
- 9) Decrementing of velocity-to-be-gained to engine shutdown
- 10) GSE service routine entrance

### 5.2.3 2-Sec Functions

Those functions performed every 2 sec consist of:

- 1) Decision logic for AGS guidance
- 2) LM navigation
- 3) Initialization of LM and/or CSM ephemeris if DEDA or downlink initialization requested
- 4) Calibration using a 2-sec sample data servo loop for nulling gyro non-g-sensitive drift, when DEDA requested. During inflight calibration or accelerometer only calibration, the accelerometer bias is also nulled.
- 5) Radar data processing; reestimation of the LM ephemeris
- 6) Computation of landing azimuth on command via DEDA
- 7) Computation of orbit insertion steering
- 8) Computation of coelliptic sequence initiate (CSI) burn
- 9) Computation of coelliptic maneuver, CDH burn
- 10) Calculation of direct intercept transfer orbit (TPI) using p-iteration routine
- 11) Computation of an external  $\Delta V$  maneuver
- 12) Computation of velocity-to-be-gained,  $V_G$ , and desired vehicle thrust direction
- 13) Computation of altitude, altitude rate, and LM out-of-CSM plane velocity every 2 sec and interpolation for output every 200 msec

## 6. DESCRIPTION OF THE PROGRAMMED EQUATIONS

Flow charts of the program are contained in Section 7. This section contains a narrative description of the program.

The flow charts show the equations of Flight Program 6 (FP6). FP5 was used as a baseline for FP6 and these flow charts reflect the software changes which have been approved and implemented into FP6. The flow charts are a functional representation of the program rather than an exact one. The exact implementation can be obtained from the program listing contained in Appendix A. Hardwired equations are enclosed by dashed lines. The location of escape points is noted on each block of hardwired equations by the  $\bigcirc$  symbol.

The lines with logical branches and arrowheads (or other instructions) specify the box of equations to be computed next. Thus, there are never two simultaneous paths indicated. The lines do not indicate from which earlier box each input to an equation is obtained.

Equation constants are denoted by  $K_j^i$ , decimal DEDA inputs by  $J_j^i$ , external hardware discrete signals from the CES by  $\beta_i$ , astronaut DEDA commands by  $S_i$ , internal decision logic flags by  $\delta_i$ , counters by  $\mu_i$ , and external input discrete signals from downlink, telemetry, GSE, and DEDA by  $\xi_i$ .

In some areas of the equations, it is necessary to distinguish between the value of a quantity in different computing cycles or subcycles. The subscript n-i is used to denote the value determined in the  $i^{\text{th}}$  2-sec cycle preceding the current computing cycle; the subscript m-i is used to denote the value determined in the  $i^{\text{th}}$  20-msec subcycle preceding the current 20-msec subcycle. The absence of the subscript from a quantity on the righthand side of the equation is to be interpreted as the last value determined, either earlier in this cycle or in the preceding cycle. Thus the equation  $X = X + Y$  is to be interpreted "set X equal to the last value of X (independent of when X was last changed in value) plus the last value of Y."

A list of variables with their program symbols is given with the flow diagrams to briefly describe each quantity and its range of values. All variables marked with an asterisk (\*) are not saved by the program and,

therefore, do not have a program symbol. The range of values is essentially specified as a maximum magnitude and a least significance or quantization. If a variable's magnitude exceeds the scaling values given and causes overflow in the computer, the resulting value is used unless specified otherwise in the flow charts.

The external CES discretizes, internal logic flags, and counters are described separately after the flow charts. The astronaut DEDA selections are described in Paragraph 8.4.1.

These flow charts show the sequence of computations as well as the distribution of computations within a 2-sec cycle. The various guidance modes require different numbers of branches to complete their computations. From guidance computation completion to Branch 50, all branches not used for DEDA (Branches 12, 24, 36, 48) are used for computer self test.

The start of each branch is labeled with the branch number ( $\mu_{10}$  count) at which it is normally performed. For TPI solutions, overflow tests resulting in a transfer to point (FB) can cause branches to be bypassed, changing the nominal branch numbers. Since the different guidance modes require different numbers of branches, the steering computations come up at different branch numbers.

The letters appearing in the upper left above blocks or in the circles are usually program symbols used for linkage within the program. In some cases, the letters in the circles are solely for flow chart linkage. Unless otherwise indicated, the circle is a connection that appears on the following page (except EXEC, which is on Figure 7.32).

In the flow charts and definitions of symbols, lower case and capital x, y, and z are used to specify inertial and body vector components, respectively. Upon entry to the Orbit Parameters subroutine, the most recently processed velocity and position vectors are used in the computations unless otherwise indicated. The orbit parameters computed are then used along with the velocity and position in the following ellipse prediction branch. The ellipse predictor blocks in most cases show the quantities input to and output from the prediction routine. The actual inputs and outputs are in special temporary storage memory locations associated with the routine.

The program includes various general-purpose subroutines which are described in Section 7.2. These subroutines perform a specific function and then return control to the point in the program from which they were selected. The program flow charts do not explicitly show the use of the subroutines. For example, whenever a vector cross product is shown on the flow charts, it is performed by use of the Vector-Cross Product subroutine.

## 6.1 20-MSEC COMPUTATIONS

### 6.1.1 Sensor Input Processing

The 20-msec computations begin with Figure 7.2. At the top of the figure, the DELAY block ensures that the execution of the next block of equations is delayed until 20 msec have elapsed since the last timing pulse. PGNCS Euler angles, gyro data, and accelerometer data inputs are appropriately stored. The first calculations compute the velocity increments over a 20-msec cycle. The accelerometer data is in the form of pulses and an accumulation of 640 pulses per 20 msec indicates no acceleration. The equations subtract 640 pulses ( $K_7^1$ ) from the number of pulses accumulated and the difference is the net pulses representing the sensed acceleration. The net number of pulses is then multiplied by the appropriate scale factor ( $K_{18}^1, K_{20}^1, K_{22}^1$ ) to convert pulses to feet per second. These scale factors can be varied according to ASA calibration measurements. In a zero g environment, the accelerometer may not output exactly 640 pulses. This bias is compensated by the constants ( $K_{19}^1, K_{21}^1, K_{23}^1$ ) for each accelerometer, respectively. These numbers are also obtained and varied as a result of calibration. Then  $\Delta a_{X \text{ rem}}$ , the rotation about the X body axis, is increased by  $K_{14}^1 (\Delta V_X)$ . This term is the correction for X spin axis mass unbalance. This correction term is required because an imperfect gyro, when accelerated, will give indications that the attitude is changing even when it is not. Terms such as this one do not appear in the Y or Z gyro equations because error analyses have indicated that they are not necessary.

Following calculation of the mass unbalance compensation, a check is made to determine if the equations are in the inertial reference mode or an align or calibrate mode. The latter mode is entered if  $S_{00} \geq 3$ .

### 6.1.2 Align or Calibration Mode

If in an align or calibrate mode, the equations determine which specific submode and act accordingly. The three alignment modes establish the orientation of the LM body axes in inertial space by computing a direction cosine matrix,  $[A]$ .

#### 6.1.2.1 PGNCS/AGS Align

If a PGNCS/AGS align has been commanded ( $S_{00} = 3$ ), the logic flow proceeds to the top of Figure 7.6 (point ZNOC) where the direction cosine orthonormality corrections ( $E_1, E_3, E_{13}$ ) are zeroed. The PGNCS Euler angle inputs (pulses) are then converted to radians by the conversion constant  $K_{25}^1$ . Operation on these angles yields the desired direction cosines ( $a_{11D}, a_{12D}, a_{13D}, a_{31D}, a_{32D}, a_{33D}$ ). The quantities ( $\Delta a_{1j}, \Delta a_{3j}, j = 1, 2, 3$ ) are then set equal to zero. These quantities will be discussed later. For discussion purposes in this section, it is sufficient to state that when any alignment is performed these quantities are zeroed. The logic flow of the equations is to the first DL on Figure 7.5.

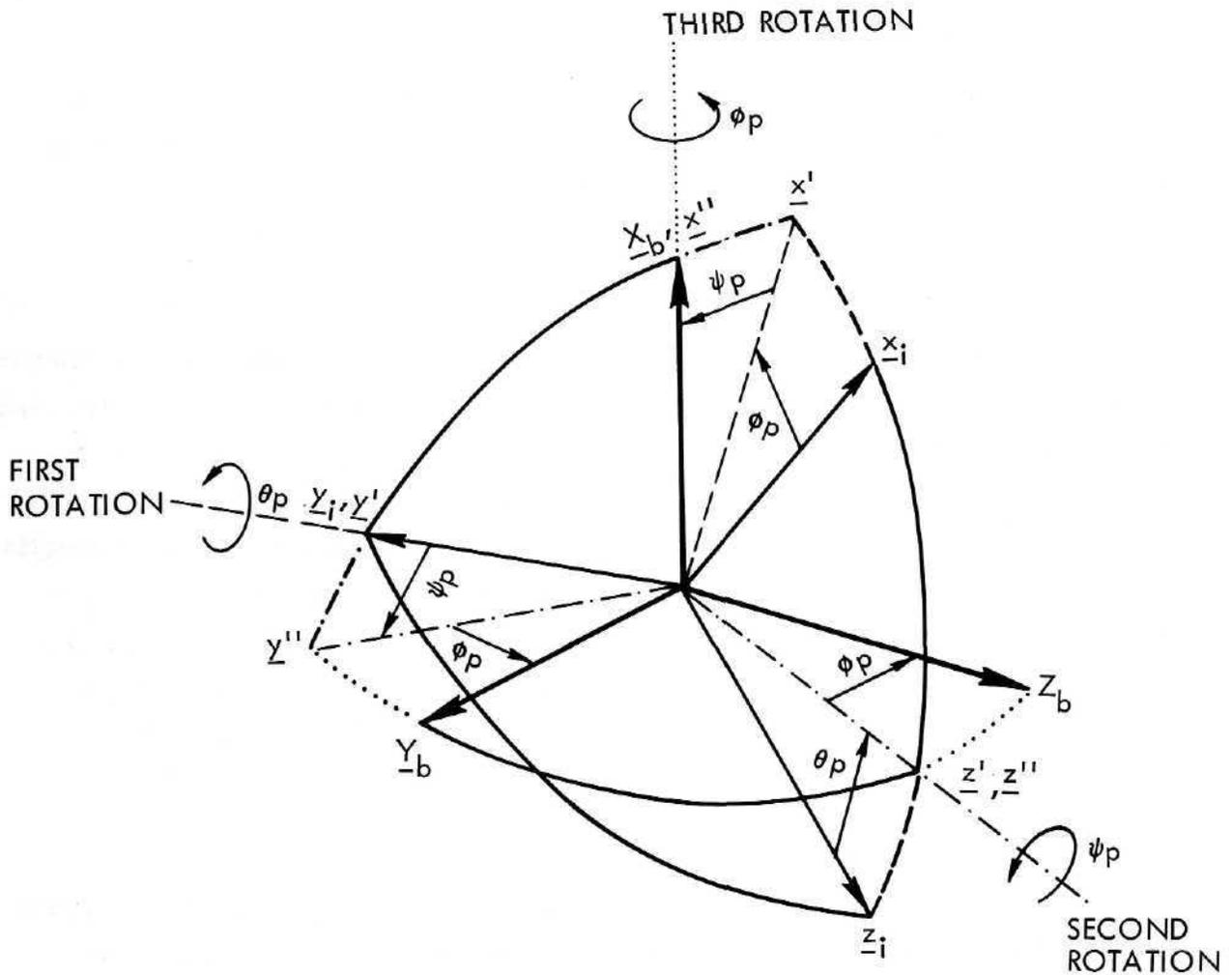
PGNCS Euler angles ( $\theta_p, \psi_p, \phi_p$ ) which are the IMU gimbal angles of the PGNCS are used by the equations in PGNCS/AGS align. The Euler angle sequence is shown in Figure 6.1.

#### 6.1.2.2 Lunar Align

If the lunar align mode has been selected ( $S_{00} = 4$ ), the logic flow is to the gyro compensation block AD, Figure 7.3.

The lunar align mode mechanized in the LM AGS is the backup alignment system in case the PGNCS is inoperable after the LM has successfully landed on the lunar surface.

Basically, the lunar align equations compute the  $[A]$  transformation matrix so that the X, Y, Z computational inertial frame coincides with a selenocentric coordinate frame which is defined with the X axis along the lunar local vertical positive outward from the lunar center and with the Z axis direction obtained by crossing the unit angular momentum vector of the CSM orbit with a unit vector along the X axis. The lunar align equations mechanize a low gain filter to compute the desired  $[A]$  matrix from the



1ST ROTATION  $\theta_p$  ABOUT  $\underline{y}_i$  ----, INNER GIMBAL  
 2ND ROTATION  $\psi_p$  ABOUT  $\underline{z}'$  -.-.-., MIDDLE GIMBAL  
 3RD ROTATION  $\phi_p$  ABOUT  $\underline{x}''$  ....., OUTER GIMBAL

EULER ANGLES ( $\theta_p, \psi_p, \phi_p$ ) ARE USED TO SPECIFY  
 THE VEHICLE REFERENCE FRAME ( $\underline{x}_b, \underline{y}_b, \underline{z}_b$ ) WITH  
 RESPECT TO THE PGNCs INERTIAL REFERENCE FRAME  
 ( $\underline{x}_i, \underline{y}_i, \underline{z}_i$ ).

Figure 6.1. IMU Euler Angles

compensated accelerometer output for leveling, an azimuth which is established after lunar landing, and an azimuth correction constant ( $\delta \Delta$ ). These equations are computed in the 40-msec computational cycle and are shown in Figure 7.9.

For lunar align to function properly, an azimuth reference must have been stored after LM touchdown on the lunar surface by the setting of  $S_{13} = 1$  or the quantities  $\sin \delta_L$  and  $\cos \delta_L$  must have been entered via DEDA.

#### 6.1.2.3 Body Axis Align

If the body axis align mode ( $S_{00} = 5$ ) has been selected, the logic flow is to point ORBLIN (Figure 7.6) where the direction cosines are initialized as  $a_{11} = a_{33} = 1$  and  $a_{12} = a_{13} = a_{31} = a_{32} = 0$ . The remaining direction cosines are computed from the preceding cosines. This alignment procedure forces the X inertial axis to lie along the X body axis, the Y inertial axis to lie along the Y body axis, and the Z inertial axis to lie along the Z body axis. This alignment must be constrained so that the XZ inertial plane lies within 80 deg of the LM orbit plane for the TPI equations to function properly. The logic flow again arrives at the first DL in Figure 7.5.

#### 6.1.2.4 Calibration Submodes

When the equations are in the calibrate mode ( $S_{00} = 6$ ), the program updates the accelerometer and gyro bias compensations. When in the accelerometer calibration mode ( $S_{00} = 7$ ), only the accelerometer bias compensations are updated. If the equations are in a calibrate mode ( $S_{00} = 6, 7$ ), the lunar surface flag ( $\delta_{21}$ ) is checked to determine if the LM is on the lunar surface ( $\delta_{21} = 1$ ) or in flight ( $\delta_{21} = 0$ ). If  $\delta_{21} = 0$ , sensed velocity increments are accumulated for use in the accelerometer calibration modes ( $S_{00} = 6, 7$ ). If  $\delta_{21} = 1$ ,  $\Delta \underline{a}_{rem}$  is compensated for lunar rotation rate by adding to it the components of the y inertial vector along the body directions multiplied by the lunar rotation constant ( $K_{56}^1$ ). In either event, the logic flow arrives at the gyro compensation block of equations and, as in the inertial reference mode, at point AD in Figure 7.3.  $S_{00}$  should NOT be set to 7 when the LM vehicle is on the lunar surface.

### 6.1.3 Inertial Reference Mode

If the equations are in the inertial reference mode or the lunar align mode, the logic flow is to the gyro compensation block in Figure 7.3. The purpose of these equations is to obtain compensated values of the gyro inputs. In order to achieve greater accuracy in the equations, a form of double precision is used in these calculations. For convenience, the following discussion is concerned with the X axis gyro. The equations are similar for the other axes except as noted below.

The double precision operates as follows. The quantity  $\Delta a_{X \text{ rem}}$  is calculated at a quantization level of  $2^{-30}$ . In calculating  $\Delta a_X$  (the compensated incremented component of rotation about the X body axis), only that part of  $\Delta a_{X \text{ rem}}$  that is greater than or equal to  $2^{-16}$  is used. This part is denoted by  $\Delta a_{X \text{ rem}} \big|_{\geq 2^{-16}}$ . Then  $\Delta a_{X \text{ rem}}$  is recomputed for use in the next computing cycle as  $\Delta a_{X \text{ rem}} - \Delta a_{X \text{ rem}} \big|_{\geq 2^{-16}}$ . Thus, the part of  $\Delta a_{X \text{ rem}}$  not used in the present cycle is retained for use in the next cycle. The three equations under discussion are

$$\Delta a_{X \text{ rem}} = \Delta a_{X \text{ rem}} + K_1^1 + K_2^1 K_3^1 (\Delta a_{X_i} - K_7^1)$$

$$\Delta a_X = K_2^1 (\Delta a_{X_i} - K_7^1) + \Delta a_{XA} + \Delta a_{X \text{ rem}} \big|_{\geq 2^{-16}}$$

$$\Delta a_{X \text{ rem}} = \Delta a_{X \text{ rem}} - \Delta a_{X \text{ rem}} \big|_{\geq 2^{-16}}$$

These equations do the following: The raw X axis gyro output ( $\Delta a_{X_i}$ ) is received in the form of pulses. In any given 20-msec computing increment, a zero angular increment would be indicated by reception of 640 pulses ( $K_7^1$ ). Thus, a negative angular increment would be detected if less than 640 pulses were received per 20 msec and a positive angular increment if more than 640 pulses per 20 msec were received. Multiplication of  $(\Delta a_{X_i} - K_7^1)$  by the constant  $K_2^1$  converts the pulse count to radians. Multiplication of the angular increment  $K_2^1 (\Delta a_{X_i} - K_7^1)$  by the constant  $K_3^1$  forms a correction for the attitude rate scale factor error of the X gyro. Added to this quantity is the gyro drift compensation constant ( $K_1^1$ ).  $\Delta a_{X \text{ rem}}$  is then formed as the summation of all the terms just discussed and that part of  $\Delta a_{X \text{ rem}}$  of the previous computation cycle that

did not exceed  $2^{-16}$  rad. Note that  $\Delta a_{X \text{ rem}}$  appears as a summation of correction terms to the raw gyro input. To obtain the compensated gyro input ( $\Delta a_X$ ) used in the attitude reference equations, the part of  $\Delta a_{X \text{ rem}}$  that exceeds  $2^{-16}$  rad is added to the raw gyro input converted to radians  $[K_2^1 (\Delta a_{Xi} - K_7^1)]$ . In addition, the alignment correction  $\Delta a_{XA}$  is added. This last term is used when a lunar align is being performed and is zeroed when any other mode is entered. Following this calculation, the  $\Delta a_{X \text{ rem}}$  term is set up for the next computing cycle by subtracting from  $\Delta a_{X \text{ rem}}$  that part of  $\Delta a_{X \text{ rem}}$  exceeding  $2^{-16}$ . The flow logic then proceeds to **BB** in Figure 7.4.

In Figure 7.4, a check is made on the magnitude of the angular increments per 20 msec to determine the desired scaling on various quantities. Scaling at  $2^{-9}$  is necessary to gain precision in updating the direction cosines at low turning rates. Scaling at  $2^{-6}$  is necessary to obtain the required dynamic range of rotation about each axis ( $\pm 25$  deg/sec). Switchover of the scaling occurs at approximately 5 deg/sec. The LM usually rotates at a rate less than 5 deg/sec so that the desired precision is obtained. The logic flow then proceeds to the direction cosine update.

Figure 7.4 contains the equations to update six of the nine direction cosines based upon the compensated gyro inputs as just discussed. Error terms ( $E_1, E_3, E_{13}$ ) are used appropriately to keep the direction cosines orthonormal. In addition, direction cosine remainder terms are added to maintain sufficient computational accuracy.  $E_1, E_3,$  and  $E_{13}$  are then set to zero.

Regardless of the mode of the equations (Figure 7.2), all logic paths arrive at the first **DL** in Figure 7.5. (See Sections 8.2 and 8.4.2 for an explanation of the telemetry and downlink routines, respectively.) At this point the remaining direction cosines ( $a_{21}, a_{22}, a_{23}$ ) are calculated from  $a_{11}, a_{12}, a_{13}, a_{31}, a_{32}, a_{33}$ . With these direction cosines the sensed velocity increments along the LM body axes are transformed to x, y, z inertial coordinates. These velocity increments are denoted by  $\Delta V_{xs}, \Delta V_{ys},$  and  $\Delta V_{zs}$ , respectively. This group of equations is computed last in the 20-msec subcycle. At the bottom of Figure 7.5, the decision is made to proceed to the 40-msec computations or to the 2-sec computations. These paths are alternately selected.

## 6.2 40-MSEC COMPUTATIONS

### 6.2.1 Updating Accumulated Velocity and Calculation of Velocity-to-be-Gained

At the beginning of the 40-msec subcycle the components of the vectors  $\underline{V}_d$  and  $\underline{\Delta V}_g$  are computed. A discussion of these quantities and several associated quantities is now presented. Appropriate definitions are:

$\underline{V}_d$  is the accumulated sensed velocity vector updated every 40 msec.

$\underline{V}_D$  is the accumulated velocity vector which is updated (set equal to  $\underline{V}_d$ ) every 2 sec.

$\underline{V}_G$  is the remaining velocity-to-be-gained vector (computed every 2 sec).

$\underline{V}_g$  is the total velocity that must be gained and equals the sum of the velocity already gained ( $\underline{V}_D$ ) and the remaining velocity-to-be-gained ( $\underline{V}_G$ ). This quantity is calculated every 2 sec.

$\underline{\Delta V}_g$  is the velocity-to-be-gained vector that is updated every 40 msec and is computed as  $\underline{V}_g - \underline{V}_d$ .

Figure 6.2 shows the relationship between these quantities. For simplicity, the quantities that are updated every 40 msec are depicted as being updated continuously. Also for simplicity, the vector notation on the quantities is dropped.

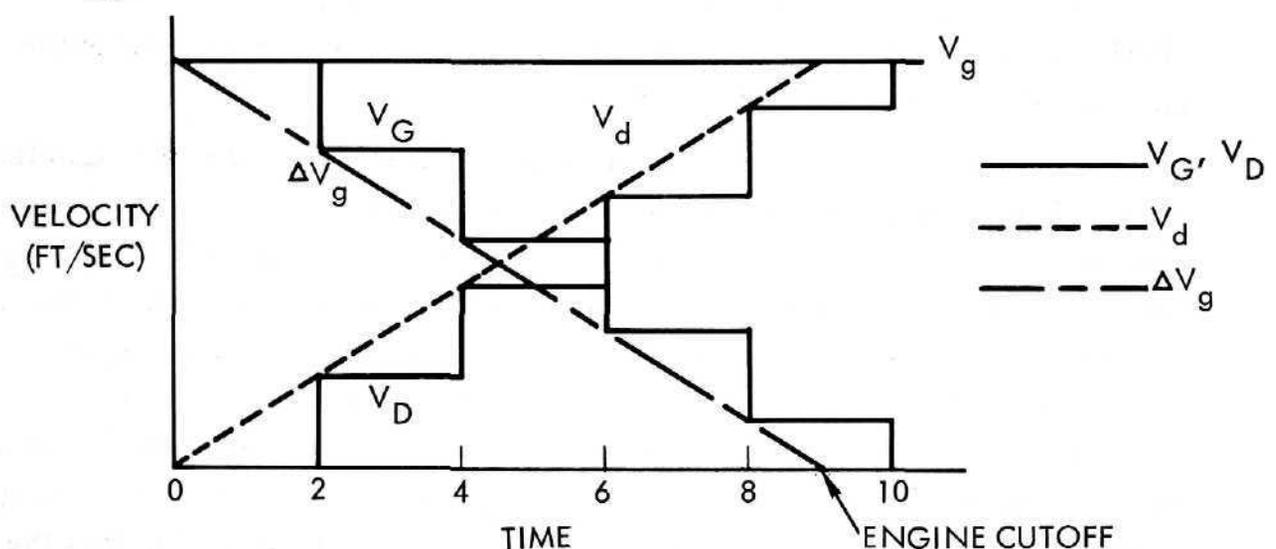


Figure 6.2. Accumulated Velocity and Velocity-to-be-Gained Diagram

The quantities are calculated this way so that the engine may be shut off based upon current information ( $\Delta V_g$ ) rather than information that may be up to 2 sec old ( $V_G$ ). Note that since  $V_D$  should increase by the same amount that  $V_G$  decreases in each 2-sec interval, the quantity  $V_g$  remains constant. Thus, the velocity remaining to be gained (calculated each 40 msec) is readily computed as

$$\Delta V_g = V_g - V_d$$

velocity to-be-gained
constant
velocity already gained updated every 40 msec

In the first block in Figure 7.7, the quantity ( $V_{dX}$ ) is computed as

$$V_{dX} = V_{dX} + \Delta V_X + \Delta V_{X, m-1}$$

which in words means: set the x component of velocity already gained equal to that gained up to 40 msec ago plus the velocity gained in the previous 20 msec computing cycle plus the velocity gained in the second previous 20 msec cycle. A remainder similar to  $\Delta \alpha_{X \text{ rem}}$  is also added to  $V_{dX}$ . The remainder holds that part of  $\Delta V_X + \Delta V_{X, m-1}$  which is less than 0.0625 ft/sec (lunar mission), the quantization of  $V_{dX}$ .

### 6.2.2 Establishing Vehicle Status

Following these calculations, the status of the  $\beta$  discrettes is checked. Every 40 msec the  $\beta$  discrettes are examined to determine the appropriate mode of operation for the computer.

Several checks are made to determine the status of the LM vehicle. These checks ascertain whether the descent section has been staged or whether the vehicle is on the lunar surface. The first check is on the  $\delta_2$  flag which is equal to one if the descent section has been staged. If the  $\delta_2$  flag has not been set to 1, then the  $\beta_2$  discrete is checked to see if the ascent engine is on ( $\beta_2 = 1$ ). If the ascent engine is not on, then the status of the vehicle is assumed to be the same as in the previous 40-msec cycle. However, if the ascent engine is on, then  $\delta_2$  is set to 1 indicating that the descent section has staged; and  $\delta_{21}$  is set to 0 indicating that the LM is not on the surface of the moon.

The remainder of the flow in the 40-msec subcycle is displayed in Figures 7.7, 7.8, 7.9, and 7.10. These diagrams are very useful for considering the results of various switch settings.

Regardless of the logic flow path just taken,  $S_{00}$  is checked to determine if the system is in an inertial reference mode ( $S_{00} < 3$ ) or in an align or calibrate mode ( $S_{00} \geq 3$ ). If in an align or calibrate mode, zero attitude errors are output. If the specific mode is lunar align ( $S_{00} = 4$ ), the logic flow proceeds to Figure 7.9 where the lunar align calculations are done. The logic flow then proceeds to point **SOFT3** in Figure 7.8 where the engine discrete is set to OFF. The logic flow also proceeds to **SOFT3** if any of the other align or calibrate modes is operating.

### 6.2.3 Steering Modes

Prior to discussing the logic in the center of Figure 7.7, the various steering modes are considered. The result of all steering mode calculations is attitude errors ( $E_X$ ,  $E_Y$ , and  $E_Z$ ) about the LM body axes. A general description of the equations follows.

#### 6.2.3.1 Attitude Hold

The "attitude hold" equations generate the steering commands to maintain the vehicle in the inertial attitude existing when the attitude hold mode is first entered. This is accomplished by utilizing the  $\delta_5$  check at the top center of Figure 7.7. When the attitude hold mode is first entered,  $\delta_5$  equals zero. This causes the desired pointing directions ( $\underline{X}_D$  and  $\underline{Z}_D$ ) to be established as the present pointing direction of the vehicle ( $\underline{X}_b$  and  $\underline{Z}_b$ ), respectively. In addition, flag  $\delta_5$  is set equal to 1 to bypass setting  $\underline{X}_D = \underline{X}_b$  and  $\underline{Z}_D = \underline{Z}_b$  in the future. The attitude error signals are then computed as

$$E_X = - \underline{Y}_b \cdot \underline{Z}_D$$

$$E_Y = - \underline{Z}_b \cdot \underline{X}_D$$

$$E_Z = \underline{Y}_b \cdot \underline{X}_D$$

which are all zero whenever  $\delta_5 = 0$ .

6.2.3.2 Guidance Steering

The "guidance steering" mode of attitude control orients the LM thrust vector along the desired thrust vector. The LM thrust vector orientation depends upon which engines are being used. The RCS engines are mounted along the body axes. If thrusting is done with the X axis thruster, the thrust vector will be along the X body axis. If the DPS engine is being used, the thrust vector is again along the positive X body axis. However, if the APS engine is used, the thrust vector is displaced from the LM positive X body axis by the pitch and roll cant angles of the APS engine. Thus, steering commands must be generated depending upon which engine is used. These equations are

$$E_X = - \frac{W}{c} \cdot \underline{Z}_b \text{ or } - \underline{W}_b \cdot \underline{Z}_b$$

$$E_Y = - \underline{Z}_b \cdot \underline{X}_{bD}$$

$$E_Z = \underline{Y}_b \cdot \underline{X}_{bD}$$

when the DPS or RCS engines are used. These equations appear at the bottom of Figure 7.7 following the logic path  $S_{11} = 0$  and  $\beta_2 = 0$ . They appear slightly different there but are equivalent to those above. For example,  $E_Y$  is calculated as

$$E_Y = E_Y - (\underline{Z}_b \cdot \underline{X}_D)$$

which in words means put into cell  $E_Y$  the previous value of  $E_Y$  minus the value  $\underline{Z}_b \cdot \underline{X}_D$ . The previous value of  $E_Y$  is zero (set to zero at the top left of Figure 7.7) and  $\underline{X}_D = \underline{X}_{bD}$  so that the resultant value of  $E_Y$  is

$$E_Y = - \underline{Z}_b \cdot \underline{X}_{bD}$$

These equations drive the positive X body axis to the desired thrust direction and the positive Z body axis perpendicular to either the angular momentum vector of the CSM orbit pointing toward the lunar surface

(defined by  $\underline{W}_c$ ) or perpendicular to the vector  $\underline{W}_b$  (which is an input quantity). The desired mode of yaw steering is selected by setting  $S_{623}$  to 0 or 1, respectively.

If the APS engine is used, the attitude error equations are (following the path of  $S_{11} = 1$  or  $\beta_2 = 1$ )

$$\begin{aligned} E_X &= -\underline{W}_c \cdot \underline{Z}_b \text{ or } -\underline{W}_b \cdot \underline{Z}_b \\ E_Y &= K_7^4 (\underline{X}_b \cdot \underline{X}_{bD}) - (\underline{Z}_b \cdot \underline{X}_{bD}) \\ E_Z &= -K_8^4 (\underline{X}_b \cdot \underline{X}_{bD}) + (\underline{Y}_b \cdot \underline{X}_{bD}) \end{aligned}$$

The logic check on the  $S_{11}$  switch followed by the check on  $\beta_2$  at the bottom of Figure 7.7 appears at first glance to be redundant. This logic has been inserted because  $\beta_2$  is 0 until the APS engine is turned on, then it is set to 1. Thus, up until this time, the desired steering would not consider the canted engine. In order to avoid a steering transient, the  $S_{11}$  switch has been inserted so that the correct steering will be done prior to the initiation of the APS burn.

#### 6.2.3.3 Z Axis Steering

The "Z Axis Steering" mode ( $S_{00} = 2$ ) of attitude control points the positive Z body axis of the LM in either the computed direction of the CSM or along the desired thrust vector. The error equations are computed as

$$\begin{aligned} E_Z &= \underline{W}_c \cdot \underline{X}_b \\ E_Y &= \underline{X}_b \cdot \underline{Z}_{bD} \text{ or } \underline{X}_b \cdot \underline{X}_{bD} \\ E_X &= -\underline{Y}_b \cdot \underline{Z}_{bD} \text{ or } -\underline{Y}_b \cdot \underline{X}_{bD} \end{aligned}$$

where  $\underline{Z}_{bD}$  is the computed direction of the CSM and  $\underline{X}_{bD}$  is the desired thrust direction of the LM. The desired Z body axis steering submode is

chosen by setting  $S_{507}$  to 0 or 1, respectively. In this mode, the X body axis is oriented perpendicular to the angular momentum vector of the CSM. If the CSM is ahead of the LM, the X body axis will be above the local LM horizontal plane; and if the CSM is behind the LM, the X body axis will be below the local horizontal plane.

#### 6.2.4 Steering Mode Decision Logic

The remainder of the logic in Figure 7.7 is now considered. At the upper left hand side of the figure, the followup flag ( $\beta_3$ ) is checked. Consider the situation where  $\beta_3 = 0$ . A check is then made on the "auto" discrete ( $\beta_4$ ). If  $\beta_4 = 0$  (attitude hold), flag  $\delta_{20}$  is set equal to 1 and the attitude hold equations are then computed.  $\delta_{20}$  is used to control the logic flow in Figure 7.8. The result of  $\delta_{20}$  being set equal to 1 maintains the AGS engine follow-up mode. That is, if either  $\beta_1$  or  $\beta_2$  equals 1, the engine discrete is set ON; if not, then the engine discrete is set OFF.

In Figure 7.7 if the  $\beta_4$  flag equals 1 (rather than 0 as just discussed), then the "abort stage" discrete ( $\beta_5$ ) is checked. Consider first the case where no abort stage is commanded ( $\beta_5 = 0$ ). Immediately following this case, the lunar surface flag  $\delta_{21}$  is checked. If on the lunar surface ( $\delta_{21} = 1$ ), zero attitude errors are sent to the autopilot and the logic flow proceeds to point ENCOM1 in Figure 7.8 where again the engine discrete is determined by the status of  $\beta_1$  and  $\beta_2$ . If not on the lunar surface ( $\delta_{21} = 0$ ), the abort discrete ( $\beta_6$ ) is checked. If  $\beta_6 = 0$ , then again  $\delta_{20}$  is set to 1. Then either attitude hold, guidance steering, or Z axis steering is done, depending upon the setting of  $S_{00}$ .

The only other alternative remaining is an abort stage command,  $\beta_5 = 1$  (center of Figure 7.7). If the LM is on the lunar surface, the logic flow is to the attitude hold equations. This same logic flow repeats every 40 msec until the "staging recognition and remove lunar surface signal" logic on the left-hand side of Figure 7.7 is exercised as discussed above. That is, when the ascent engine comes on ( $\beta_2 = 1$ ),  $\delta_{21}$  is set to zero so that, following the  $\beta_5 = 1$  check at the center of the figure, the check on  $\delta_{21}$  now sends the logic flow downward toward the  $\beta_2$  check.

Since  $\beta_2$  was set to 1, the logic flow proceeds to the  $\mu_6$  counter check.  $\mu_6$  is incremented by 1 every 40 msec. The vehicle stays in attitude hold until  $\mu_6$  exceeds  $K_{23}^4$ , then the logic flow proceeds to the steering equations selected by  $S_{00}$ . If the abort stage had been set in flight with the ascent engine off ( $\beta_2 = 0$ ), then the  $\mu_6$  logic would be bypassed (until  $\beta_2 = 1$ ) and the nominal guidance steering continued. The  $\mu_6$  logic allows the vehicle to be in attitude hold automatically as the vehicle lifts off the lunar surface and during staging. After a prescribed amount of time (controlled by  $K_{23}^4$ ), the vehicle will orient to the attitude defined by the chosen steering equations.

Consider now again the check on  $\beta_3$ . In this instance, let the AGS be in followup ( $\beta_3 = 1$ ). If  $\beta_4 = 0$  (altitude hold), then the attitude errors are zeroed and the logic flow proceeds to ENCOM1 in Figure 7.8. If  $\beta_4 = 1$  (auto), then  $\delta_{20}$  is set to 1 and the attitude errors are computed for display purposes. These error signals are inhibited by the autopilot so that the vehicle will not act on them but they may be used for display purposes. Also,  $\delta_5$  is set to 0 to inhibit attitude hold error signals (if  $S_{00} = 0$ , attitude errors equal 0 in followup).

#### 6.2.5 Engine ON/OFF Logic

As already indicated, there are four exits from Figure 7.7. These exits have been discussed above with the exception of the exit to SOFT2 when  $\delta_{20}$  is zero. SOFT2 appears in Figure 7.8. If  $\delta_{20}$  is zero and the mode of the computer ( $S_{00}$ ) is not in guidance steering, then the engine is commanded OFF. However, if  $S_{00} = 1$  (guidance steering), a series of checks are made to determine the proper status of the engine discrettes. First, the ullage counter is checked. If this counter has not accumulated to the value  $K_9^1$ , the engine discrete is set OFF. If  $\mu_8 \geq K_9^1$ , then a check is made on the velocity-to-be-gained in the X body direction. If this (as computed every 40 msec) is greater than the value  $K_{25}^4$  (engine shutdown impulse), the engine discrete is set ON. If less than  $K_{25}^4$ , a check on total velocity-to-be-gained is made against  $K_{26}^4$ . The total velocity-to-be-gained is denoted by  $\Delta V_G$ . This is a dummy variable set equal to  $V_G$  near the end of each 2-sec computing increment. If  $\Delta V_G \geq K_{26}^4$ , the engine discrete is set ON. If less than  $K_{26}^4$ , the attitude hold

mode is entered ( $S_{00} = 0$ ),  $S_{11}$  is set to zero and the engine discrete is set to OFF. The attitude hold mode of operation is entered prior to engine OFF because the residual  $V_G$  vector could point in any direction and it is not desirable for the vehicle X body axis to try to point in this arbitrary direction.

Checking  $\Delta V_G$  against  $K_{26}^4$  guards against the situation (that may occur in an abort) when the vehicle attitude is poorly oriented and the velocity-to-be-gained is large. In this case,  $\Delta V_{gX}$  (the quantity that is used to actually shut off the engine) could be negative yet the desired condition is for the engine to be ON. The check of  $\Delta V_G$  against  $K_{26}^4$  ensures that the desired conditions are obtained.

The remainder of Figure 7.8 indicates the sampling of DEDA and GSE discrettes and the computing of the quantities  $E_1, E_3, E_{13}$ . These three terms are used for the orthonormal corrections to the direction cosine matrix. The remainder of the 40-msec computations are shown in Figure 7.10 where  $\sin \alpha, \cos \alpha, \sin \beta, \cos \beta, \sin \gamma, \cos \gamma$  are computed for the FDAI and sent to the D/A converter.

### 6.3 2-SEC COMPUTATIONS

The usual mode of operations is to execute the 2-sec computations beginning with LM navigation, the first branch in Figure 7.13. The first ten 2-sec cycles are common to every guidance mode. Thereafter, the program flow is determined by the selected guidance mode ( $S_{10}$  guidance mode switch setting). For all guidance modes, the fiftieth (last) 2-sec cycle is the executive branch, Branch 50 (Figure 7.31), which routes the program flow to the first branch.

The only time the 2-sec computations do not start with the LM navigation is if an initialization is to be performed.

If  $S_{14}$  is set to 1 and all of the downlink data has been transmitted from PGNCS, logic flag  $\delta_{31}$  will be set to 1. If this occurs or if  $S_{14}$  is equal to 2, then Branch 50 routes the program flow to the initialization branches (Figures 7.11 and 7.12) and no guidance equations (Figures 7.13 to 7.30) are solved during this 2-sec period.

Prior to discussing the initialization equations it is appropriate to explain the navigation techniques for the CSM and LM.

### 6.3.1 CSM Navigation

The CSM position and velocity at any time is determined by utilization of a subroutine called the ellipse predictor. This subroutine has the capability of accepting position and velocity of a vehicle at any time, i. e.,  $t_E$ , and determining the position and velocity of the vehicle at any other time, i. e.,  $t_E + T_i$ . Here,  $T_i$  can be either a positive or negative number. The prediction of position and velocity is based upon the assumption that the acceleration due to gravity is of the form  $k/r^2$  where  $k$  is a constant and  $r$  is the distance from the center of the attracting body. This form of the gravity model will hereafter be called spherical. For the CSM, ephemeris information in the form of  $\underline{r}_E$ ,  $\underline{V}_E$ , and  $t_E$  is stored in the computer. Here,  $\underline{r}_E$  is the CSM position vector at the epoch ( $t_E$ ) and  $\underline{V}_E$  is the CSM velocity vector at the epoch ( $t_E$ ). Thus, if it is desired to know the position and velocity of the CSM at the present time ( $t$ ), then the CSM position ( $\underline{r}_E$ ) and velocity ( $\underline{V}_E$ ) are input to the ellipse predictor subroutine along with the time interval ( $t_b = t - t_E$ ). The output of the ellipse predictor is then called  $\underline{r}_c$  and  $\underline{V}_c$ , the position and velocity of the CSM at time  $t$ . Note that the epoch point is not updated to the present time but rather that the original values of  $\underline{r}_E$ ,  $\underline{V}_E$ , and  $t_E$  are maintained. The only time the epoch point for the CSM is changed is when new ephemeris data is obtained via downlink or DEDA, or when the epoch is older than one orbital period ( $t_b > T_{CSM}$ ), it is updated (bootstrapped, Figure 7.14) one full orbital period. This updating prevents the quantity  $t_b$  from exceeding the computer scaling.

### 6.3.2 LM Navigation

The ellipse predictor subroutine as discussed above does not take into account any acceleration of the vehicle other than that due to spherical gravity. The technique used for CSM navigation is not valid for LM navigation since the LM vehicle goes through various thrusting phases. LM navigation is done by integrating the equations of motion of the vehicle in a central force field. In this way external accelerations (other than gravity) can be entered when appropriate. The only time the ellipse predictor subroutine is used for LM navigation is when new LM ephemeris

data is obtained and this data must be updated to the present time. Of course for the updating to be valid, no LM thrust acceleration can occur between the time of the ephemeris point and the present time.

### 6.3.3 Navigation Initialization

If  $S_{14}$  is 1 and downlink has been completed, then a CSM and LM ephemeris update is to be performed utilizing information obtained from the PGNCS downlink (Figures 7.11 and 7.14). This information is converted to the correct format and stored in the DEDA cells. The quantity  $t_b$  discussed above is set to zero. If  $S_{14} = 2$ , then the LM ephemeris is to be updated via DEDA inputs in the form of constants  $J_1^1$  thru  $J_7^1$ . No new CSM data is obtained at this time.

$S_{14}$  is automatically set to zero so that, in the next 2-sec computing increment, the normal guidance equations will be utilized. Also, the present time  $t$  is incremented by 2 sec. Several quantities are obtained for the LM orbit from the orbit parameters subroutine that appears in Figure 7.33. This subroutine computes the following orbital elements:

- $r$  - distance of LM from the center of moon
- $a$  - semimajor axis of LM orbit
- $n$  - LM mean orbital rate
- $C$  - product of LM eccentricity and cosine of eccentric anomaly
- $S$  - product of LM eccentricity and sine of eccentric anomaly.

Figure 7.12 indicates how the LM ephemeris data is updated to the present time. First, the quantity  $T_i$ , the time increment from when the data were valid to the present time, is calculated. If the vehicle is on the lunar surface ( $\delta_{21} = 1$ ), then the position and velocity vectors are updated simply as

$$\underline{r} = \underline{r}_0 + \underline{V}_0 T_i$$

$$\underline{V} = \underline{V}_0$$

However, if  $\delta_{21}$  is 0, then the present position and velocity of the LM are obtained via the ellipse predictor as indicated previously. Following these updating calculations, the gravity subroutine (Figure 7.34) is entered to compute the following quantities:

$\underline{U}_1$  - unit vector from lunar center to LM

$\dot{h}$  - radial rate of LM

h - altitude of LM

$\underline{\Delta IG}$  - integrated gravity vector per  
2-sec cycle

These quantities are computed for use in the next 2-sec computing increment and for display. Then  $\underline{\Delta IG}$  is initialized and the accumulated thrust along the inertial axes ( $\underline{\Delta V}_s$ ) is zeroed.

This completes the initialization routine and no additional guidance or navigation computations are done in the present 2-sec computing increment.

#### 6.3.4 Navigation Equations Description - Branch 1

If  $S_{14}$  is neither 1 nor 2, then the 2-sec computations begin in Figure 7.13. The equations in this figure obtain the LM position and velocity based upon attitude and the sensed accelerometer  $\Delta V$ 's, accumulated during the 20-msec computation cycles.

If the LM is on the lunar surface ( $\delta_{21} = 1$ ), the accumulated velocities ( $\underline{V}_d$ ), the accumulated inertial sensed velocities ( $\underline{\Delta V}_s$ ), and the gravity integral  $\underline{\Delta IG}$  are set equal to zero, the ullage counter is set to  $K_9^1$  (its threshold value), and the navigation update equations are entered. If not on the lunar surface, thrust acceleration is obtained by differencing accumulated velocity along the X body axis this cycle with that 2 sec ago and dividing by 2. As explained above,  $V_{dX}$  is the accumulated thrust velocity along the X- body axis valid at the present time and  $V_{DX}$  is that valid 2 sec in the past. The thrust acceleration ( $a_T$ ) is checked against constant  $K_{35}^4$  ( $0.1 \text{ ft/sec}^2$ ). If  $a_T \geq K_{35}^4$ , the ullage counter ( $\mu_8$ ) is increased by 1. If  $a_T < K_{35}^4$ , the ullage counter ( $\mu_8$ ) is set to zero.  $\Delta V_s^*$  is the sum of the absolute values of the components of sensed velocity during the 2-sec computation interval and is checked against the constant



The average value of  $\underline{G}$  is computed to be  $[\underline{G}_n + \frac{1}{2}(\underline{G}_n - \underline{G}_{n-1})]$ . When this is multiplied by 2 sec to obtain velocity, the expression

$$\Delta \underline{IG} = 3\underline{G}_n - \underline{G}_{n-1}$$

is obtained. The vector value of  $\underline{G}$  is obtained by the expression

$$\underline{G} = -\frac{\mu r}{r^3} = -\frac{\mu}{r^2} \underline{U}_1$$

where  $\underline{U}_1$  is the radial unit vector  $\frac{\underline{r}}{r}$  of the LM.

The value of the LM position vector is updated by taking the old position vector and adding to it 2 times the average value of the velocity vector during the previous 2 sec. Thus,

$$\underline{r} = \underline{r} + 2 \left[ \frac{\underline{V}_n + \underline{V}_{n-1}}{2} \right] = \underline{r} + \underline{V}_n + \underline{V}_{n-1}$$

Following these computations, the gravity subroutine computes the LM altitude (h) for later guidance purposes, the LM radial rate (h) for display purposes, and updates  $\Delta \underline{IG}$  for use in the next 2-sec computations. In addition, time is incremented by 2 sec. The logic flow then proceeds to Figure 7.14 where the CSM orbit parameters are computed from the ephemeris data.

#### 6.3.5 CSM Orbit Parameters and LM Total Velocity – Branches 2, 3, and 4

If  $S_{14} = 3$ , a new CSM ephemeris point has been input via the DEDA. At the top of Figure 7.14, a check of  $S_{14}$  is made against 3. If it is 3, the new information is entered and  $S_{14}$  and  $t_b$  are zeroed.  $\underline{W}_c$ , the unit vector parallel to the CSM angular momentum, is then computed. The same orbit parameters as determined for the LM in Figure 7.11 are obtained for the CSM from the orbit parameter subroutine in Figure 7.33. Next, the CSM epoch bootstrap is accomplished if required. As indicated previously, if the time since the CSM epoch, i. e.,

$$t_b = t - t_E$$

is greater than one CSM orbital period, then the epoch is increased by one orbit. This increase is accomplished at the bottom of Figure 7.14 where  $T_{\text{CSM}}$  is the CSM orbital period.

The logic flow then proceeds to Branch 3, Figure 7.15, where the calculations are done to compute  $t_b$  and call the ellipse prediction subroutine and obtain the CSM position and velocity ( $\underline{r}_c$  and  $\underline{V}_c$ ) valid at the present time. These quantities are then used for the computation of the LM to CSM range, pointing direction, and range rate (Figure 7.16).

In Branch 4, Figure 7.16, the LM total velocity is computed for use in computing LM orbit parameters in Branch 8.

#### 6.3.6 Altitude Update and Radar Filter - Branches 5, 6, and 7

In branches 5, 6, and 7 (Figures 7.16, 7.17, and 7.18) the altitude update and radar filter equations are performed. The altitude update feature enables the astronaut to update the LM position vector by making a DEDA entry of the present altitude into  $J^{25}$ . The altitude is obtained from the landing radar or from visual estimation. If  $J^{25}$  is greater than zero, the magnitude of the LM position vector,  $r$ , is computed by adding  $J^{25}$  and  $J^5$ , the nominal lunar landing site radius. The LM inertial position components are then computed by multiplying  $r$  by the unit vector from the lunar center to the LM.

If  $J^{25}$  does not exceed zero, no altitude update is performed. In any event,  $J^{25}$  is zeroed at the end of branch 5.

The rest of branch 5 and all of branches 6 and 7 constitute the radar filter. Prior to discussing the equation flow logic of the radar filter, a general description of the radar operation procedure is given.

There is no hardware connection between the radar and the AEA. Angle information is obtained in the computer by "saving" the Z body axis direction cosines ( $a_{31}$ ,  $a_{32}$ ,  $a_{33}$ ) at the time that the radar gimbal angles are zero. This time is indicated to the computer when an  $S_{15}$  DEDA entry is performed. At this same time the astronaut must read the range or range rate tape meter and within 30 sec enter the radar data via the DEDA. Radar range is entered as manual constant  $J^{18}$ ; range rate is entered as manual constant  $J^{17}$ . The procedure for a range update is as follows:

1. The astronaut sets  $S_{00} = 2$  and  $S_{507} = 0$ . This commands the Z body axis of the LM to be pointed in the estimated direction of the CSM.
2. The astronaut sets the mode control switch to attitude hold ( $\beta_4 = 0$ ) and, by means of the hand control, the astronaut orients the LM until the radar gimbal angles on his attitude display are all zero. At this time the Z body axis is pointing toward the CSM. When the hand control is taken out of its detent and  $\beta_4 = 0$ , the AGS computer automatically enters the "followup" mode ( $\beta_3 = 1$ ).
3. When the gimbal angles have been zeroed, the astronaut makes a DEDA entry into  $S_{15}$ . Within 0.5 sec of this entry, the Z body axis direction cosines are saved in  $\underline{Z}_b^*$  and the computed range vector and range rate are saved in  $\underline{R}^*$  and  $\dot{R}^*$ , respectively. Also,  $\Delta t$ , the time since the last radar range entry was processed, is computed by subtracting  $t_1$  from the present absolute time (see Figure 8.5).
4. Range, read from the tape meter at the time of the  $S_{15}$  entry, is entered into  $J^{18}$ .

The procedure for a range rate update is to make an entry into  $S_{15}$  at the time when range rate is read from the tape meter. Then, the range rate is entered into  $J^{17}$  via DEDA.

The radar filter is a suboptimal filter simplified from a 16-variable Kalman filter (Reference 4). The filter operates by assuming that knowledge of the CSM position and velocity is perfect. LM position and velocity are then updated based upon the navigation estimate of LM present position and velocity, the radar measurement, and the relative credibility of these two quantities. The out-of-plane state vector components,  $r_y$  and  $V_y$ , are updated essentially by a constant gain method. For the in-plane components, the relative credibility is expressed as the 4 x 4 matrix P, the LM error covariance matrix.

P may be initialized to a diagonal matrix by setting  $S_{17}$  to one via DEDA. The matrix is updated every 16 sec to reflect the updating of the LM state vector in coasting flight. During any 2-sec cycle in which P is updated, radar data will be processed if it has been entered. If radar data is processed, the LM state vector and the covariance matrix are updated, the latter reflecting the improved state vector.

During the processing of radar data, the in-plane state vector is updated by computing a weighting matrix  $B$ . The difference between the radar data and computed data is then weighted by  $B$  and used to improve the state vector.

Refer now to the equations on Figure 7.16.

After the altitude update processing, the range and relative velocity vectors ( $\underline{R}$  and  $\dot{\underline{R}}$ ) from the LM to the CSM are computed. The unit vector along the range,  $\underline{Z}_{bD}$ , is computed and dotted with the range rate vector to compute the range rate magnitude  $\dot{R}$ . Then, three gradients of the gravitational acceleration on the LM,  $a$ ,  $c$ , and  $j$ , are computed for use in the next branch. At the end of the branch,  $J^{25}$  is zeroed.

In branch 6, the radar filter 2-sec cycle counter is incremented. If it is less than eight, the branch control word is set to point to the end of branch 7, and branch 6 is ended. If the counter equals eight, 16 sec have passed since the last updating of the LM error covariance matrix,  $P$ . The counter is then zeroed, and  $P$  is updated to reflect the change of LM state during the last 16 sec.

If any computer overflows occurred during this computation or if  $S_{17}$  has been set to one via DEDA, the covariance matrix is initialized to a diagonal matrix,  $S_{17}$  is zeroed, and the time of the last radar range update,  $t_1$ , is set equal to  $J^{29}$ .

Then, the radar range ( $J^{18}$ ) and range rate ( $J^{17}$ ) are tested. If no DEDA entry has been made to either quantity, the branch control word is set to point to the end of branch 7, and branch 6 is ended.

In the range and range rate update equations, three intermediate computation matrices,  $Q$ ,  $M$ , and  $L$ , are used. The derivations and physical significance of these matrices are contained in Reference 4.

Consider a range rate update. If  $J^{17}$  is non-zero,  $\Delta r_1$ , the difference between the radar range rate and the computed range rate, is calculated. Then,  $J^{17}$  is zeroed. Elements  $q_{12}$  and  $q_{22}$  of matrix  $Q$  are set equal to zero and one, respectively. Elements  $m_{11}$ ,  $m_{12}$ ,  $m_{13}$ , and  $m_{14}$  of vector  $M$  are computed from the present values of LM to CSM pointing direction, range, and relative velocity (computed in branch 5). The first column of matrix  $L$  is computed from  $M$  and  $P$ , and the second column of  $L$  is zeroed.

Matrix element  $q_{11}$  is then computed using  $M$  and  $L$ ;  $\Delta$ , the determinant of  $Q$ , is set equal to  $q_{11}$ . This concludes the range rate update processing in branch 6.

Next, consider a range update. If  $J^{17}$  equals zero,  $J^{18}$  is tested. If  $J^{18}$  is greater than zero, it is multiplied by  $\underline{Z}_b^*$  to compute the inertial components of the range. Then,  $t_1$  is set equal to the present absolute time.

Differences between input range  $\underline{R}^{**}$  and computed range  $\underline{R}^*$  are stored in  $\Delta r_1$ ,  $\Delta r_3$ , and  $\Delta r_2$ .

The out-of-plane difference  $\Delta r_3$  is multiplied by constant  $K_5^6$  and added to the LM y inertial position component. The LM y inertial velocity component correction is computed by multiplying  $\Delta r_3$  by the constant  $K_6^6$  and dividing by the time since the last range update processing.

The matrix elements of  $Q$  are computed from the present values of LM to CSM pointing direction and range (computed in branch 5) and the covariance matrix. The elements are then used to determine  $\Delta$ , the determinant of  $Q$ .

At the conclusion of branch 6, the first column of matrix  $L$  is set equal to the negative of the first row of  $P$  and the second column of  $L$  is set equal to the second row of  $P$ .

In branch 7, the weighting matrix  $B$  is computed using the  $L$  and  $Q$  matrices. For a range rate update,  $q_{12}$  and the second column of  $L$  equal zero and  $q_{22}$  equals one. Therefore, the first column of vector  $B$  equals the first column of  $L$  divided by  $\Delta$  and the second column of  $B$  equals zero for range rate updates. The LM in-plane state vector is updated using the weighting matrix and the differences between computed and input range. For a range rate update,  $\Delta r_2$  is not used since the second column of  $B$  equals zero.

The covariance matrix is updated using matrices  $L$  and  $B$ . Finally,  $J^{18}$  is zeroed and the branch is ended.

Regardless of whether an error covariance matrix update or a radar update has occurred, the radar filter uses branches 5, 6, and 7.

### 6.3.7 Guidance Equations

After leaving the radar filter, the logic flow proceeds to Figure 7.19 where the actual guidance equation calculations begin.

The top of Figure 7.19 shows that several calculations are performed to obtain parameters of the LM orbit. Specifically, LM horizontal velocity ( $V_h$ ), semi-latus rectum ( $p$ ), and eccentricity squared ( $e_1^2$ ) are computed. From the quantities ( $e_1$  and  $p$ ), the quantity ( $q$ ) is computed, which is the pericyynthion of the LM orbit. This quantity later has the value of the landing site radius subtracted from it with the result stored in  $q_{LT}$ , which is used for display purposes. If  $e_1^2$  is greater than  $2^{-6}$ , the calculation of  $q$  can exceed its scaling in the computer so that  $q$  is merely set to the constant  $K_3^2$ . Proceeding to Branch 9, the LM state vector is stored for the computation of the LM orbit parameters using the orbit parameters subroutine (Figure 7.33). The eccentric anomaly ( $E$ ) is computed and used with the LM orbit parameters to calculate the time to perifocus of the LM ( $T_{perg}$ ), and the apofocus altitude of the LM orbit ( $q_a$ ) is computed using the LM semi-major axis.

In Branch 10, several unit vectors are generated. Based upon the present position of the LM,  $\underline{U}_1$  ( $= \underline{r}/|r|$ ), and the unit vector  $\underline{W}_c$ , parallel to the angular momentum vector of the CSM, the horizontal vector  $\underline{V}_1$ , parallel to the CSM orbit plane, is generated. Then, from  $\underline{U}_1$  and  $\underline{V}_1$  the third unit vector  $\underline{W}_1$  for the LM is calculated. The current LM position and velocity vectors are stored in  $\underline{r}_5$  and  $\underline{V}_5$ , respectively, for later use if the guidance mode is orbit insertion. The present out-of-CSM-plane position and velocity are computed. An angle ( $\xi$ ), which is the inplane angle between the Z body axis and the LM local horizontal, is computed for display purposes.

In Branch 11 (Figure 7.20), a series of checks are made on  $S_{10}$  to determine the guidance mode of operation. The various modes of  $S_{10}$  are:

$S_{10} =$	}	0	orbit insertion
		1	CSI
		2	CDH
		3, 4	TPI search, TPI execute
		5	external $\Delta V$

#### 6.3.7.1 External $\Delta V$

The first check on  $S_{10}$  is against the value 5. When  $S_{10}$  is 5, the external  $\Delta V$  mode is entered, and the first check in this path is against the quantity  $S_{07}$ . To initialize the external  $\Delta V$  computations,  $S_{07}$  must be zero. To ensure that this is so,  $S_{07}$  is set to zero when the CSI, CDH, or TPI guidance mode is selected. Assuming  $S_{07}$  is zero, the next check made is on the ullage counter. The ullage counter must also have a value of zero for the external  $\Delta V$  mode initialization to be performed. This means that the vehicle must not be thrusting in the positive X body axis direction during the cycle the external  $\Delta V$  mode is entered. The next block of equations establishes the external  $\Delta V$  maneuver to be performed. This block is bypassed once thrusting has started. Prior to discussing this initialization, the two possible types of external  $\Delta V$  maneuvers are discussed.

##### 1) RCS Thrusting with Vehicle in Attitude Hold

Assume for some reason it is desired to perform a maneuver with the vehicle attitude fixed in inertial space. This can be accomplished by setting the switch  $S_{00}$  to zero,  $S_{10}$  to 5, and inputting the appropriate maneuver constants (discussed below). This maneuver is to be performed with the RCS engines since the E/OFF command will be generated when the mode selector ( $S_{00}$ ) is not equal to 1 (see Paragraph 6.2.5). The maneuver will be performed separately along each axis.  $S_{07}$  must be set to 1 just prior to the initiation of the maneuver by the astronaut. Then, the external  $\Delta V$  initialization equations are bypassed as desired when the maneuver has begun. The Y and Z body axes components should be nulled prior to nulling the X body axis  $\Delta V$ .

2) Thrusting with the Positive X Body Axis in the Direction of the Desired Velocity-to-be-Gained

In this mode,  $S_{10}$  is set to 5 and  $S_{00}$  to 1. The vehicle will automatically orient itself to point in the correct direction. No extra entry (on the switch  $S_{07}$ ) is required and no constraints are placed on which engines can be used.

3) Thrusting with the Positive Z Body Axis in the Direction of the Desired Velocity-to-be-Gained

In this mode,  $S_{10}$  is set to 5,  $S_{00}$  to 2, and  $S_{507}$  to 1. The vehicle will automatically orient itself to point the positive Z body axis to the desired thrust direction. Just prior to the initiation of the maneuver, the astronaut must set  $S_{07}$  to 1, which causes the external  $\Delta V$  initialization equations to be bypassed. The astronaut then performs the maneuver using the Z axis thrusters. When the velocity-to-be-gained becomes less than 15 fps, the mode control switch should be set to ATT HOLD.

In the external  $\Delta V$  mode the desired maneuver is selected by testing  $J_1^{28}$ . When the program is in the TPI guidance mode,  $J_1^{28}$  is set equal to  $K_{11}^2$ , a large velocity. The CSI and CDH mode equations compute  $J_1^{28}$  (and  $J_2^{28}$ ,  $J_3^{28}$ ); this  $J_1^{28}$  will always be much less than  $K_{11}^2$ . Also, if  $J_1^{28}$ ,  $J_2^{28}$ , and  $J_3^{28}$  are input via DEDA,  $J_1^{28}$  will be much less than  $K_{11}^2$ . Therefore, if  $J_1^{28}$  is less than  $K_{11}^2$ , the desired maneuver is defined by the values of the input constants  $J_1^{28}$ ,  $J_2^{28}$ , and  $J_3^{28}$ . A value of  $J_1^{28}$  characterizes the velocity-to-be-gained in the LM horizontal direction parallel to the CSM orbit plane with a positive value indicating a posigrade maneuver.  $J_2^{28}$  characterizes the velocity-to-be-gained in the horizontal out-of-CSM-plane direction with a positive value indicating a direction opposite to the angular momentum vector.  $J_3^{28}$  characterizes the velocity-to-be-gained in the LM local radial direction with a positive value indicating thrusting toward the moon.

These components of velocity-to-be-gained are resolved into inertial coordinates. The inertial coordinates are fixed once the maneuver starts or when  $S_{07}$  is set to 1, and they yield an initial velocity-to-be-gained vector ( $\Delta \underline{V}$ ).

If  $J_1^{28}$  equals  $K_{11}^2$ , a TPI maneuver solution is desired, and  $\Delta \underline{V}$  is set equal to the velocity-to-be-gained vector  $\underline{V}_G$  (in inertial coordinates) computed in the TPI guidance mode.

After the  $\Delta \underline{V}$  vector is computed, the accumulated velocity vector  $\Delta \underline{q}_s$  (in inertial coordinates) is zeroed until the maneuver starts ( $S_{07}$  set to 1). Then, the initialization block is bypassed and the velocity-to-be-gained vector is computed as

$$\underline{V}_G = \Delta \underline{V} - \Delta \underline{q}_s$$

or, in words, the velocity-to-be-gained equals the original velocity-to-be-gained minus that already gained. The magnitude of  $\underline{V}_G$  is then obtained. The logic flow then proceeds to (Q7) in Figure 7.30. Here, until the velocity-to-be-gained is less than  $K_{26}^5$ , the desired pointing direction is computed to be in the direction of the velocity-to-be-gained

$$\underline{x}_{bD} = \underline{V}_G / V_G$$

When  $V_G$  becomes less than  $K_{26}^5$  and the vehicle is thrusting in the +X body direction ( $\mu_g > 0$ ), the desired pointing direction is not updated because large attitude maneuvers may ensue from the inaccuracies in the calculations of  $\underline{V}_G / V_G$ . Then  $V_G$  is stored in  $\Delta V_G$  for use in the 40-msec engine logic and the velocity-to-be-gained along each body axis ( $V_{GX}$ ,  $V_{GY}$ ,  $V_{GZ}$ ) is computed and used in the calculation of  $\underline{V}_g$  as explained in Paragraph 6.2.1. Note that, if the vehicle is in attitude hold, all the quantities ( $V_{GX}$ ,  $V_{GY}$ ,  $V_{GZ}$ ) could be non-zero. However, if the vehicle is in guidance steering ( $S_{00} = 1$ ), the vehicle will orient itself such that  $V_{GX} = V_G$  and  $V_{GY} = V_{GZ} = 0$ . The discussion just completed concerning the lower portion of Figure 7.30 is common to all guidance modes and will not be discussed in the following subsections. The guidance calculations for the external  $\Delta V$  mode are completed in 11 of the 50 minor cycles in the 2-sec computations.

#### 6.3.7.2 Orbit Insertion

This guidance mode has been designed to make the coast orbit prior to CSI a function of the phasing that exists between the LM and CSM prior to orbit insertion. The guidance mode controls the semi-major axis of the coast orbit as a function of the selenocentric angle between the LM

and CSM. The five parameters explicitly controlled are:

Altitude at injection

Altitude rate at injection

Horizontal velocity at injection

Out-of-CSM-plane position at injection (zero desired)

Out-of-CSM-plane velocity at injection (zero desired)

Of course, in some abort situations, all of these conditions cannot be achieved. Comments on these situations are contained in the following discussion.

Return to Figure 7.20 and follow the logic flow down the left-hand side of the figure. Assume that  $S_{10} = 0$  so that the orbit insertion guidance mode is entered.  $T_{\Delta}$  is zeroed so that the CSI, CDH, and TPI calculations which are common to orbit insertion can be utilized, and the logic flow proceeds to **DXFR<sub>10</sub>** on Figure 7.22. Recall that the current LM position and velocity were stored in  $\underline{r}_5$  and  $\underline{V}_5$ , respectively, in Branch 10, Figure 7.19, so that  $\underline{U}_1$  is computed as the unit vector in the radial direction of the LM and thus,  $\dot{r}_A$  is the current LM radial rate. The computations of  $p$ ,  $V_{py}$ , and  $V_{ha}$  are unused in orbit insertion and thus are not discussed here. The CSM ephemeris position and velocity vectors are then stored in  $\underline{r}_0$  and  $\underline{V}_0$ , respectively, for use in the next branch as inputs to the Orbit Parameters Subroutine, where the CSM orbit parameters are computed. After the orbit parameters are computed,  $T_i$ , the elapsed time since the CSM epoch (since  $T_{\Delta} = 0$ ) less one or two CSM orbital periods to avoid later overflows in the ellipse predictor subroutine, is computed. This time is then used in branch **DXFR11** to compute the current CSM state vector,  $(\underline{r}_c, \underline{V}_c)$  and then the logic flow proceeds to DXFR13 Figure 7.23. Here, the current central angle  $\theta_f$  between the CSM and LM position vectors is computed along with the quantities  $T_{\delta}$  and  $T_i$ , which are not used in orbit insertion. The logic flow is then to point **DXFR14** where a check is made on  $S_{10}$ . Since  $S_{10} = 0$  (orbit insertion), control is

transferred to point  $\textcircled{\text{OII}}$ , Figure 7.28, and the following quantities are computed:

- $V_{ha}$  Horizontal component of the LM velocity which is parallel to the CSM orbital plane
- $\dot{r}_f$  Desired final value of the LM radial rate which is limited to be between the constants  $J_3^{23}$  and  $K_6^4$
- $a_L$  Desired semi major axis of the LM orbit at orbit insertion, limited to be between the constants  $J^8$  and  $J^9$
- $r_f$  Predicted LM altitude at orbit insertion which was saved in  $r_b$  during the previous 2 sec Guidance Law computations (Figure 7.29)

Now the required horizontal velocity ( $V_{hf}$ ) is computed based upon  $a_L$  and  $r_f$ , and the horizontal velocity-yet-to-be-gained ( $J_1^{28}$ ) is computed as the difference between the required and current horizontal velocities. The sign of  $J_1^{28}$  is later used in the orbit insertion steering equations to determine whether the thrust maneuver is to be posigrade or retrograde. In the next branch the radial velocity-yet-to-be-gained ( $J_3^{28}$ ), with positive sign indicating the velocity-to-be-gained is toward the moon, is computed. Since  $S_{10} = 0$ , the logic proceeds to the guidance law branch where the magnitude of the velocity-to-be-gained is computed from the following equation:

$$V_G = \left[ (V_{hf} - V_{ha})^2 + (\dot{r}_A - \dot{r}_f)^2 + (V_{y0})^2 \right]^{1/2}$$

Diagram illustrating the components of the velocity-to-be-gained equation:

- $V_{hf}$ : final inplane horizontal velocity
- $V_{ha}$ : present inplane horizontal velocity
- $\dot{r}_A$ : present value of radial rate
- $\dot{r}_f$ : final value of radial rate
- $V_{y0}$ : present value of out-of-plane velocity

Following the calculation of  $V_G$ , the time to burn ( $T_B$ ), the desired values of the derivative of radial rate and the derivative of out-of-plane velocity are computed, with the predicted value of the final radial position being saved in  $r_b$  for use in the next 2 sec computations. Then, the desired values of the second derivative of radial rate and the second derivative of out-of-plane velocity are computed. These calculations are based upon the orbit insertion guidance law which is to maintain  $\ddot{r}_d$  and  $\ddot{y}_d$  constant.

The lower limit of  $\ddot{r}_d$  ( $\ddot{R}_{DTL}$ ) is a function of vehicle configuration. If staging has occurred ( $\delta_2 = 1$ ) and  $a_T$  is less than  $K_{12}^4$ ,  $\ddot{R}_{DTL}$  is set to  $K_{20}^5$ ; otherwise it is set to  $K_{18}$ . Following these computations, the value of  $\ddot{r}_d$  is modified to account for vehicle motion about a spherical potential. The resulting value of  $\ddot{r}_d$  is the desired vertical component of the thrust acceleration. The terms  $K_1^2/r^2$  and  $V_h^2/r$  are the gravitational and centrifugal accelerations. The logic flow then proceeds to the steering branch in Figure 7.30.

The equations in Figure 7.30 generate the desired pointing direction of the vehicle X body axis and obtain the velocity-to-be-gained along each body axis.

$\psi_P$  is defined as the sine of the desired pitch angle and  $\psi_y$  as the sine of the desired out-of-plane angle. In general, these quantities are computed as the desired value of radial acceleration divided by thrust acceleration and the desired value of out-of-plane acceleration divided by thrust acceleration, respectively. However, when the LM lifts from the lunar surface or if an abort occurs near touchdown, it is more desirable to thrust vertically than to follow the computed pitch profile. This is accomplished by setting  $\psi_P$  equal to 1 and  $\psi_y$  equal to 0. Then the desired pointing vector  $\underline{X}_{bD}$  is merely  $\underline{U}_1$ , the unit vector in the radial direction.

The logic at the top of Figure 7.30 is used to determine if the thrust vector should be in the radial direction. The first check is that of LM altitude (h) against the constant  $J^{21}$ . If the altitude is less than  $J^{21}$ , a check is made of altitude rate ( $\dot{r}$ ) against the constant  $J^{22}$ . If  $\dot{r}$  is greater than  $J^{22}$ , the desired pitch profile is flown. If not, then the vehicle is commanded to thrust vertically and  $V_G$  is set to a large value (1024 ft/sec). This last check actually controls the time when the LM starts pitching over after the vertical rise from the surface.

Following calculation of the desired pointing vector  $\underline{X}_{bD}$ , the various components of velocity-to-be-gained are computed as discussed in Paragraphs 6.2.1 and 6.3.7.1. This then completes the orbit insertion calculations for any 2-sec computing cycle. This guidance mode requires 18 of the 50 minor cycles in the 2-sec computations.

Two ideas above are considered here in somewhat more detail. The first is the calculation of  $\dot{r}_f$  and the second is the calculation and limiting of  $\ddot{r}_d$  and  $\ddot{y}_d$ .

The value of  $\dot{r}_f$  is obtained from the equation

$$\dot{r}_f = K_4^4 (K_5^4 - r_b)$$

and then limited between the values of  $J^{23}$  and  $K_6^4$  as follows

$$J^{23} \leq \dot{r}_f \leq K_6^4$$

Then,  $\dot{r}_f$  as a function of final altitude appears as illustrated in Figure 6.4. The reason for this type of function is as follows: When the predicted final value of altitude is near the desired burnout altitude, the altitude rate should be the desired value ( $J^{23}$ ). However, some abort situation may arise where it is impossible for the desired final altitude to be achieved; for example, a low fast abort.

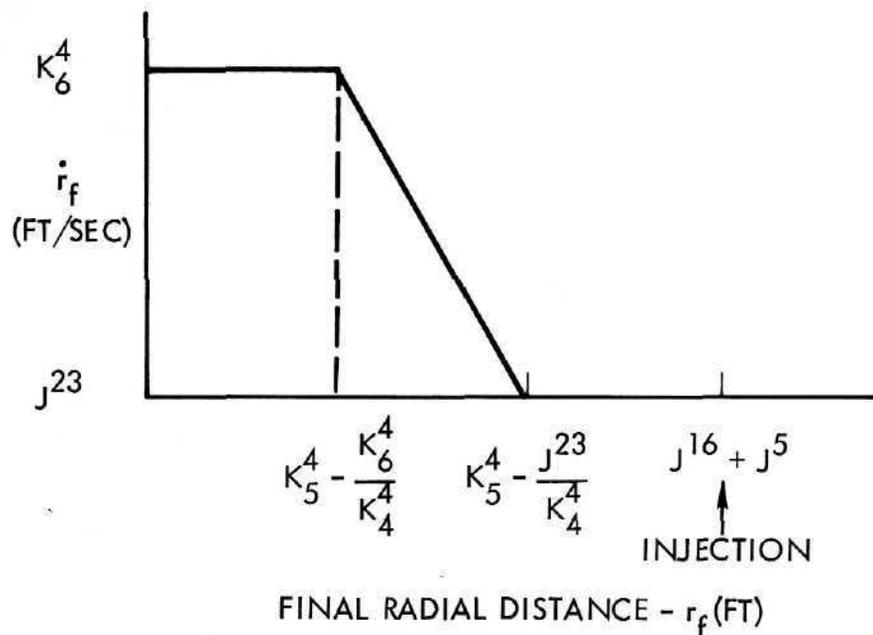


Figure 6.4. Altitude Rate Versus Final Radial Distance

In this situation, the object is to be on a rising ( $\dot{r} > 0$ ) trajectory at the time of engine cutoff so that at a later time appropriate modifications can be made to the orbit to ensure safe pericyynthion. Prescribing  $\dot{r}_f$  as above yields this condition.

The calculations of  $\ddot{r}_d$  and  $\ddot{y}_d$  are contained in Figure 7.29. The equation for  $\ddot{r}_d$  contains the term  $J^{16}$  which is desired final altitude at the time of orbit insertion. The value of  $\ddot{r}_d$  is constrained to lie between the values of  $\ddot{R}_{DTL}$  and  $K_{14}^5$ . These values are chosen to limit the maneuver of the vehicle. When  $\ddot{r}_d$  lies outside either of the limits, the effect is that the equations give up trying to drive the LM to the desired altitude and just make a limited correction. The orbit insertion maneuver will continue, however, until the desired velocity is obtained.

The equation for  $\ddot{y}_d$  is similar to that of  $\ddot{r}_d$ . In this instance, the predicted final value of out-of-plane position error is used in the calculation. This quantity is desired to be driven to zero. Again,  $\ddot{y}_d$  is limited to a value between  $K_{17}^5$  and  $K_{16}^5$ . If the computed value of  $\ddot{y}_d$  is outside the limits, the equations give up trying to drive the LM into the CSM orbit plane at engine cutoff and just make a limited correction. The philosophy used in the selection of these constants has been to steer out approximately 1/2 deg out-of-plane error if the abort occurs at liftoff. If the abort occurs later during orbit insertion, only a smaller out-of-plane position error is removed. Any additional out-of-plane error is removed at the CSI, CDH, or plane change maneuvers.

### 6.3.7.3 CSI Routine

After the orbit insertion maneuver is completed, the astronaut switches  $S_{10}$  to 1; that is, enters the CSI guidance mode and the following targeting parameters:

$t_{ig}$	absolute time at which the CSI maneuver is to be performed
$J^1$	absolute time at which the TPI maneuver is intended to be performed
$J^2$	cotangent of the desired angle between the LM to CSM line-of-sight and the local horizontal at time $J^1$
$S_{16}$	selection switch for CDH at either one- or three-halves LM orbital period following CSI

The CSI calculation determines the magnitude of the horizontal burn to be performed parallel to the CSM orbit plane at CSI time ( $t_{ig}$ ) in such a manner that, following the coelliptic maneuver (CDH maneuver), the cotangent of the desired line-of-sight angle ( $J^2$ ) will be achieved at time  $J^1$ . By definition, coelliptic means: 1) the product of semimajor axis and eccentricity of the LM orbit equals the same product of the CSM orbit and 2) the line of apsides of the two orbits are aligned when one line is projected onto the other vehicle's orbit plane.

Figure 6.5 depicts the CSI geometry discussed in this paragraph.

The discussion of the equation logic flow begins with mission time sometime after orbit insertion and prior to CSI. Each time the logic follows the CSI ( $S_{10} = 1$ ) path (Figure 7.20), the time to the CSI maneuver ( $T_{\Delta}$ ) is computed as the difference between CSI time ( $t_{ig}$ ) and the current absolute time ( $t$ ). Then, CSI time ( $t_{ig}$ ) is recomputed as the sum of

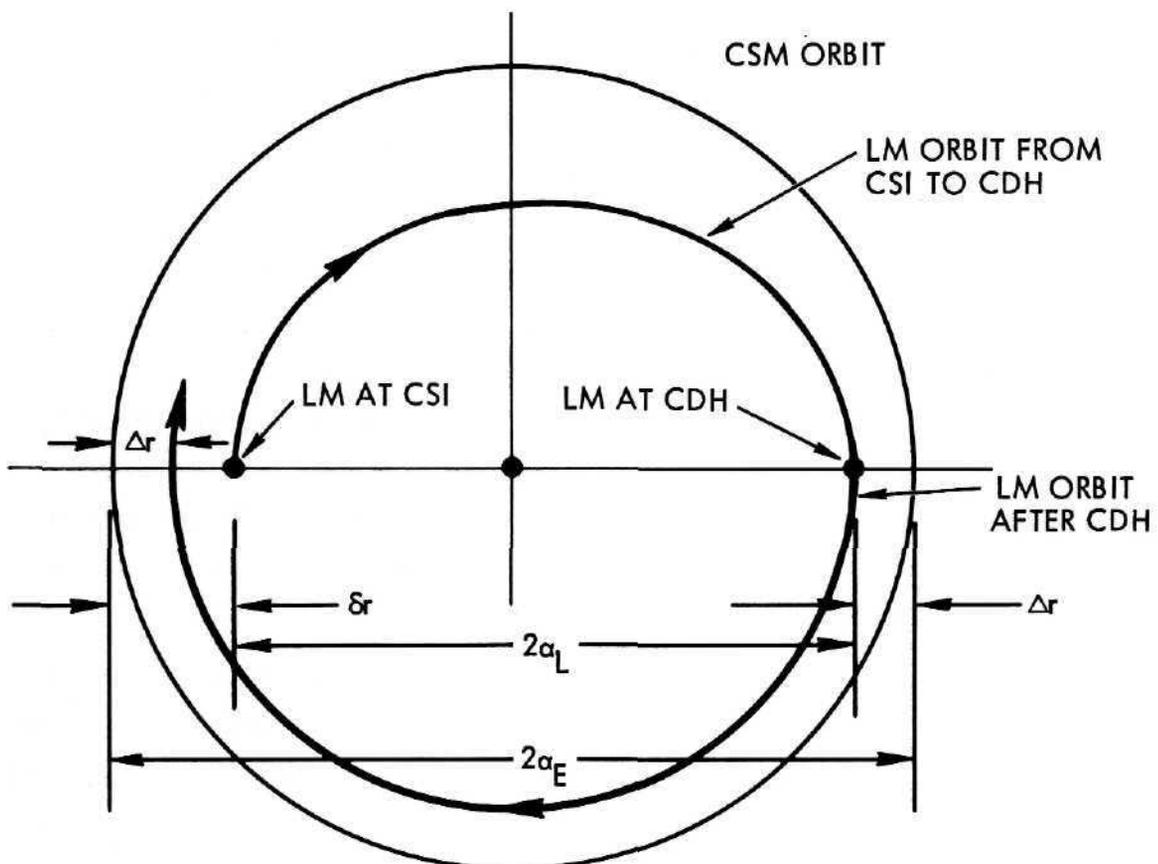


Figure 6.5. CSI to CDH Geometry

absolute time ( $t$ ) and the time to CSI ( $T_{\Delta}$ ). This computation is redundant to the CSI routine. Next, the TPI quantity  $T_r$  is computed.  $T_r$  and many other unneeded CSI quantities are computed in CSI to take advantage of the computations which are common to other guidance modes. Proceeding with the logic,  $T_{\Delta}$  is constrained to the non-negative,  $S_{07}$  is zeroed so that it is zero when external  $\Delta V$  is entered, and  $T_i$ , the time which is input to the ellipse predictor subroutine, is set equal to  $T_{\Delta}$  so that in branch DXFR9 (Figure 7.21) the LM position and velocity are propagated to the time of the CSI maneuver ( $\underline{r}_5$  and  $\underline{V}_5$ ). The constraining of  $T_{\Delta}$  to be non-negative results in the CSI solution being computed based on CSI time ( $t_{ig}$ ) being equal to present absolute time when absolute time exceeds the targeted  $t_{ig}$ . After the LM position and velocity ( $\underline{r}_5$  and  $\underline{V}_5$ ) at CSI are predicted, the following quantities are computed in DXFR10 (Figure 7.22):

$r_f$	magnitude of the predicted LM position at CSI
$p$	TPI quantity which is unused in CSI
$\underline{U}_1$	unit vector in direction of the LM position at CSI
$\dot{r}_A$	predicted value of the LM altitude rate at CSI
$V_{Py}$	predicted LM out-of-CSM-plane velocity at CSI which is computed for DEDA display
$V_{ha}$	magnitude of the predicted LM velocity at CSI

These computations are followed by the storing of the CSM ephemeris position and velocity vectors in  $\underline{r}_0$  and  $\underline{V}_0$ , respectively, for use in the next branch (ORBIT 3) where the CSM orbit parameters are computed along with the time  $T_i$ .  $T_i$  represents the time from the CSM epoch ( $t_E$ ) to CSI time ( $t_{ig}$ ) less one or two CSM orbital periods ( $T_{CSM}$ ) to avoid later overflows when other times are added to  $T_i$  or when  $T_i$  is used in the ellipse predictor subroutine.  $T_i$  is then used in branch DXFR11 to predict the CSM position and velocity ( $\underline{r}_c$  and  $\underline{V}_c$ ) at CSI time.

In the next branch, DXFR13 (Figure 7.23), the central angle ( $\theta_f$ ) from the LM to the CSM at CSI time is computed.  $\theta_f$  is used to compute the differential altitude ( $\delta r$ ) between the CSM and LM orbits as measured on the radial line passing through the LM at CSI time. To obtain  $\delta r$ , the CSM position at CSI time must therefore be propagated through the angle ( $-\theta_f$ ), or by the time it takes to traverse ( $-\theta_f$ ). Thus,  $\theta_f$  is divided by the

mean angular rate of the CSM ( $n_E$ ) to obtain an approximation to this time,  $T_\delta$ . Then, to avoid later overflows, a CSM orbital period ( $2\pi/n_E$ ) is subtracted from  $T_\delta$ .  $T_\delta$  is subtracted from  $T_i$  (recall that at this point  $T_i$  represents the time from the CSM epoch until CSI time, Figure 7.22), and the difference is stored in  $T_i$ . The logic flow then proceeds to branch DXFR14 where  $T_i$  is used to predict the CSM state ( $\underline{r}_T, \underline{V}_T$ ), where  $\underline{r}_T$  lies along the projection of  $\underline{U}_1$  in the CSM orbit plane.

In the Transfer Vectors branch (Figure 7.24) the magnitude ( $r_T$ ) of the CSM position ( $\underline{r}_T$ ) along the LM radial line at  $t_{ig}$  and the unit vector ( $\underline{U}_2$ ) in the direction of  $\underline{r}_T$  are computed, along with the TPI quantities  $c_1$  and  $c_2$ . The logic flow is then to branch CSICDH (Figure 7.28) since  $S_{10} = 1$ . Here the difference between the magnitudes of the CSM and LM positions along the radial line through the LM at CSI is computed and stored in  $\delta r$ ; thus,  $\delta r$  is the differential altitude between the CSM and LM at CSI time, positive when the CSM is above the LM. The saving of this difference in  $\Delta r$  is pertinent to the CDH routine and will therefore be discussed there. Next, the radial velocity for the CSM state ( $\underline{r}_T, \underline{V}_T$ ) is computed and stored in  $\dot{r}_f$ . Since  $S_{10} = 1$ , the following quantities are computed:

- |             |   |
|-------------|---|
| A, B, C     | intermediate quantities in the computation of the differential altitude between the CSM and LM at CDH   |
| $\Delta r$  | differential altitude between the CSM and LM at CDH; that is, if a line is drawn from the center of the moon to the LM predicted position at CDH and projected onto the CSM orbit plane, $\Delta r$ is the difference between the radial distance of the CSM at the time it crosses that line and the LM radial distance measured along that line |
| $V_{p0}$    | predicted velocity-to-be-gained in the CDH maneuver   |
| $T_{A0}$    | predicted elapsed time from the CSI maneuver to the CDH maneuver (used by the CDH equations to compute $t_{ig}$ )   |
| $\dot{r}_f$ | set equal to the current LM altitude rate so that the radial component of the CSI velocity-to-be-gained, computed as $J_3^{28} = \dot{r}_A - \dot{r}_f$ , will be zero. This is done since the CSI maneuver is designed to be horizontal.   |

For a derivation of the above quantities, refer to Reference 5.

Proceeding with the logic flow, the predicted semimajor axis of the LM orbit after the CSI maneuver is computed as the CSM semimajor axis minus half the sum of the differential altitudes at CSI and CDH. (The assumption that CSI and CDH occur at apsidal crossings is made. This does not mean that the CSI and CDH maneuvers must occur at apsidal crossings. The effects of this assumption on the CSI solution are negligible. See Figure 6.5.) Next, the desired horizontal velocity ( $V_{hf}$ ) after the CSI maneuver is computed and used to compute the horizontal component ( $J_1^{28}$ ) of the CSI burn. That is, the horizontal component of the burn is the difference between the desired and actual horizontal velocities at CSI. The next branch starts with the computation of the CSI radial velocity-to-be-gained ( $J_3^{28}$ ). Since CSI is designed to be a horizontal burn,  $J_3^{28}$  is computed as zero since  $\dot{r}_f$  had previously been set equal to  $\dot{r}_A$ . The logic then proceeds to the external  $\Delta V$  routine at point XDV (Figure 7.20). The external  $\Delta V$  equations compute the velocity-to-be-gained inertial vector  $\Delta \underline{V}$  valid at  $t_{ig}$ .

Thus, the CSI routine is completed on branch 21 of the 50 2-sec branches.

#### 6.3.7.4 CDH Routine

Following the CSI maneuver, the astronaut enters the CDH guidance mode by setting  $S_{10}$  to 2. At this time, the absolute time of the CDH maneuver is computed by adding  $T_{A0}$  to the CSI maneuver time  $t_{ig}$ . No CDH targeting is required unless another CDH maneuver time is desired, in which case  $t_{ig}$  must be entered.

The CDH routine computes the velocity-to-be-gained parallel to the CSM orbit plane necessary to maintain a constant distance between the LM orbit and the CSM orbit.

The CDH equations consist almost entirely of CSI guidance equations. When  $S_{10}$  equals 2, the logic in Branch 11 (Figure 7.20) computes  $T_{\Delta}$  from  $t_{ig}$ , limits  $T_{\Delta}$  to be non-negative, and zeroes  $S_{07}$  to initialize the external  $\Delta V$  logic. Using the LM state vector predicted ahead to time  $t_{ig}$  in DXFR9 (Figure 7.21) (or the present state vector if  $T_{\Delta}$  is zero),  $\underline{U}_1$ ,  $\dot{r}_A$ ,  $r_f$ ,  $p$ ,  $V_{py}$ , and  $V_{ha}$  are computed in DXFR10 (Figure 7.22). These quantities have the same definitions as in the CSI mode except that they are valid at

CDH time. The next four branches are identical to those of CSI. The CSM state vector is predicted ahead to time  $t_{ig}$ , the LM horizontal unit vector  $\underline{V}_1$  valid at CDH is computed, and the CSM-LM central angle  $\theta_f$  is determined. Then the CSM is retarded by  $\theta_f$ , and in the Transfer Vectors branch (Figure 7.24) the magnitude of the CSM position ( $r_T$ ) and unit vector ( $\underline{U}_2$ ) along the LM radial line at CDH are computed. Since  $S_{10}$  is 2, the equations proceed to CSICDH (Figure 7.28) where the LM-CSM differential altitude at CDH ( $\delta r, \Delta r$ ) is determined. The CSM radial velocity for the state ( $\underline{r}_T, \underline{V}_T$ ) is also computed.

With  $S_{10}$  equal to 2, the equations compute the CDH maneuver time using  $t_{ig}$  and  $T_{A0}$  and set  $T_{A0}$  to zero. Thus, if CDH follows CSI, the CDH maneuver time is computed from the CSI time and  $T_{A0}$  during the first pass through the CDH equations; subsequent passes through CDH do not change  $t_{ig}$ . The LM semimajor axis is determined by subtracting  $\Delta r$  from the CSM semimajor axis. Then the desired CDH horizontal velocity and the horizontal velocity-to-be-gained at CDH ( $J_1^{28}$ ) are computed. In the next branch, the negative of the radial velocity-to-be-gained at CDH is stored in  $J_3^{28}$ , and the logic branches to the external  $\Delta V$  logic (Figure 7.20) where the velocity-to-be-gained vector  $\Delta \underline{V}$ , valid at  $t_{ig}$ , is computed.

The CDH routine is completed in branch 21 of the 50 2-sec branches.

#### 6.3.7.5 TPI

When  $S_{10}$  is either 3 or 4, the TPI guidance routine is performed. The TPI routine is used in the coelliptic rendezvous scheme following the CDH maneuver. The previous CSI and CDH maneuvers were performed such that at the desired TPI time ( $J^1$ ), the proper phasing exists between the LM and CSM to achieve the desired line-of-sight angle (represented by its cotangent  $J^2$ ). For each line-of-sight angle at TPI time, there is a corresponding best time from TPI to rendezvous ( $J^6$ ).

Because there are errors in the system (sensors, navigation, maneuver execution, etc.), the desired line-of-sight will not be achieved at exactly the targeted TPI time, but should occur within some relatively small time period near the nominal TPI time. For this reason, a guidance option has been included for the astronaut to determine when the desired line-of-sight will be achieved and to perform the maneuver at this time if

he desires. In this mode also, the astronaut can determine the total velocity required to rendezvous at various times and perform the TPI maneuver based upon this quantity.

The two different guidance modes operate as follows:

1)  $S_{10} = 3$

The astronaut introduces an increment of time ( $T_{\Delta}$ ) ahead of the present time at which he wishes to look at the rendezvous solution. The time at which the TPI maneuver is to occur is always moving ahead 2 sec per 2 sec because the time of initiation of the maneuver ( $t_{ig}$ ) equals  $t + T_{\Delta}$ .  $T_{\Delta}$  is a fixed number and  $t$  is the present absolute time. When the solution is satisfactory, based upon DEDA monitoring of the line-of-sight angle or total velocity required to rendezvous,  $S_{10}$  can be set equal to 4.

2)  $S_{10} = 4$

Whereas the guidance mode  $S_{10} = 3$  is used for a form of mission planning, the mode  $S_{10} = 4$  is used in general for the maneuver computation itself. If the desired rendezvous solution is found with  $S_{10} = 3$  and then  $S_{10}$  is set to 4, no additional entry of  $t_{ig}$  need be inserted in the computer since this is done automatically. If guidance mode  $S_{10} = 3$  is not used, then when  $S_{10} = 4$ , a value of TPI time ( $t_{ig}$ ) must be inserted. Many different values could be tried if the astronaut desired to do some mission planning in this mode and did not care to use the alternate mode. After a value of  $t_{ig}$  is inserted, a solution is found in one computing<sup>ig</sup> increment, as in the other guidance routines.

Assuming the value of  $t_{ig}$  has been selected (and  $S_{10} = 4$ ), then the quantity  $T_{\Delta}$  is the time to go until the maneuver. This quantity could be used to set the events timer.

#### 6.3.7.5.1 Equations Description

Following the logic in Figure 7.20, it is observed that when  $S_{10} = 3$ , the input quantity is  $T_{\Delta}$  and  $t_{ig}$  is computed, whereas when  $S_{10} = 4$ ,  $t_{ig}$  is the input quantity and  $T_{\Delta}$  is computed. The remainder of the logic is the same for both modes.  $T_r$ , the time to rendezvous, is computed. As in CSI and CDH,  $T_{\Delta}$  is constrained to be non-negative and  $S_{07}$  is zeroed. If  $T_{\Delta}$  is greater than zero, then the position ( $\underline{r}_5$ ) and velocity ( $\underline{V}_5$ ) of the LM at the time of the maneuver are ascertained from the ellipse prediction

in branch DXFR9 (Figure 7.21). If  $T_{\Delta}$  is less than zero, it is set to zero for use in the ellipse predictor and the TPI solution is computed in real time so that  $\underline{r}_5$  and  $\underline{V}_5$  are the LM present position and velocity. In the next branch, DXFR10 (Figure 7.22), the LM position vector magnitude ( $r_f$ ) and the LM position unit vector ( $\underline{U}_1$ ) at TPI time are computed. Also, an initial value of the transfer orbit semi-latus rectum ( $p$ ) is chosen equal to  $r_f + K_{14}^2$ . This quantity is later iterated to select the proper transfer orbit. The predicted radial and horizontal velocities at  $t_{ig}$  ( $\dot{r}_A$  and  $V_{ha}$ ) are computed for use as display quantities. The predicted out-of-plane velocity at  $t_{ig}$  is computed for use in an out-of-plane maneuver.

Then, the CSM position and velocity vectors are stored in  $\underline{r}_0$  and  $\underline{V}_0$  for use in computing the CSM orbit parameters during branch ORBIT3. After computing the orbit parameters, the time  $T_i$  from CSM epoch time to  $t_{ig}$  is computed and made negative to avoid later computational overflows. In branch DXFR11, the position ( $\underline{r}_c$ ) and velocity ( $\underline{V}_c$ ) of the CSM are updated to the LM TPI time. In Branch DXFR13 (Figure 7.23),  $\underline{R}$ , the relative range vector at  $t_{ig}$ , and  $\underline{V}_1$  are computed as in CSI. Since  $S_{10}$  is either 3 or 4, the LM-CSM line-of-sight angle  $\theta_{LOS}$  is computed.  $\theta_{LOS}$  is the angle from the LM horizontal to the CSM. In order to determine the transfer trajectory to rendezvous, the position and velocity of the CSM must be determined at the desired time of rendezvous. This is done by updating the CSM through a time  $T_i$  from the CSM epoch point where (assume  $J^3 = 0$ )

$$T_i = t_b + T_r - T_{CSM}$$

and  $t_b$  is the time from the CSM epoch point to the present time. A CSM orbit period ( $T_{CSM}$ ) is subtracted from  $T_i$  to avoid exceeding the maximum computer value. The time from TPI to rendezvous ( $T = T_r - T_{\Delta}$ ) is computed for use in the  $p$  iterator. During branch DXFR14, the CSM position and velocity are computed using the ellipse predictor and are denoted  $\underline{r}_T$  and  $\underline{V}_T$ , respectively. The logic flow then leads to the transfer vectors branch (Figure 7.24) which is the first branch of calculations of the  $p$  iterator.  $J_1^{28}$  is set to  $K_{11}^2$  to signal the external  $\Delta V$  routine that it is to compute a TPI maneuver solution, as described in Paragraph 6.3.7.1.

The p iterator equations contained in Figures 7.24, 7.25, and 7.26 are used to answer the following questions. What trajectory passes between two specified points in a given time T? When at the first point on a present trajectory, what velocity must be added to achieve the desired trajectory? What velocity is required at the second point to be on the CSM trajectory? Of course, the first point mentioned is either the present position if  $t > t_{ig}$  or the position at the TPI maneuver time if  $t < t_{ig}$ . The second point is the rendezvous point.

The limitations imposed by the p iterator are considered and the desired outputs noted. The first restriction due to the p iterator equations is that the stable member XZ plane should be within 80 deg of the LM orbit plane. This restriction comes about because of the calculation (Figure 7.24)

$$\text{Sgn}(c_2) = \text{Sgn}(y y_c)$$

where  $y$  and  $y_c$  are the components along the  $y$  inertial axis of  $\underline{W}_1$  and  $\underline{W}_c$ , respectively. The term  $\text{Sgn}(y y_c)$  is an approximation to the exact expression  $\text{sgn}(\underline{W}_1 \cdot \underline{W}_c)$ . The constraint of  $\pm 80$  deg imposes no problems on the planned orbital alignment procedures. In the case of lunar alignment, the stable member xz plane essentially coincides with the LM orbit plane.

The next restriction imposed by the p iterator equations is the central angle through which rendezvous is to occur. Theoretically, a p iterator yields no solution to the problem when the central angle between the first and second points discussed above is any multiple of  $\pi$  radians. This is evidenced by several equations in the p iterator which contain divisions by the quantity  $c_2$  which is the sine of the central angle and is zero at these points. Because of the computer word size, a region of increasing inaccuracy occurs near these singularities. For lunar missions, these regions are  $\pm 10$  deg. If  $c_2 < K_4^3$ , the logic proceeds to Figure 7.27 and sets  $V_T$  to  $K_{11}^2$ .

In the p iterator, eight trial trajectories are considered each 2-sec computing increment. If any of these trial trajectories has an eccentricity greater than 0.5, the p iterator will not determine the solution to the problem. A check is made in the p iterator to determine if, after the eight trial trajectories have been tried, the final trajectory is sufficiently close to the desired trajectory. This is done by comparing the time it takes to get from the first point to the second point on the chosen trajectory with the desired time for the transfer to occur. If these numbers differ by more than  $K_{20}^2$  sec, then the p iterator will not yield a solution. One method to check if a solution has been obtained is to examine the quantity  $V_T$  which is essentially the sum of the magnitudes of the two velocities-to-be-gained (initial and final). If the value of  $V_T$  equals  $K_{11}^2$ , then a valid solution has not been obtained.

The outputs of the p iterator needed to compute the maneuver (both for steering and for velocity-to-be-gained) are used in the transfer/braking branch (top of Figure 7.26) to compute  $\dot{r}_f$ , the desired radial rate after the TPI maneuver. The program then retards  $T_i$  by  $J^4$  in order to create a node at  $J^4$  sec before rendezvous. Logic flag  $\delta_{10}$  is set to 1 and the program returns to DT14 (Figure 7.23) where the CSM position and velocity at  $T_i - J^4$  are computed. Then the transfer vector branch (Figure 7.24) is recomputed (if the check on  $c_2$  passes). Since  $\delta_{10} = 1$ , the p iterator is bypassed and the program zeroes  $\delta_{10}$  and proceeds to (T/B) (Figure 7.26). Here,  $\underline{V}_G$  (the velocity-to-be-gained vector) and  $\underline{V}_F$  (the braking velocity vector) are computed. Next, the magnitude of  $\underline{V}_G$  is stored and the LM transfer orbit pericynthion ( $q_1$ ) is computed from the eccentricity ( $e$ ) and the semimajor axis ( $a$ ). Then, if  $c_2$  is less than  $K_4^3$  (see Figure 7.24),  $J^4$  is zeroed, and the logic proceeds to (T/B) (Figure 7.26), retaining the valid solution for which  $J^4$  equals zero. ( $V_T$  will be set to  $K_{11}^2$  for this 2-sec cycle.)

The program proceeds to the total velocity branch (Figure 7.27) and computes the magnitude of  $\underline{V}_F$  and then the total velocity requirement of TPI ( $V_T$ ). If any overflows occurred in the computations for  $\underline{V}_F$ ,  $\underline{V}_G$ , or  $V_T$ , then  $V_T$  is set to  $K_{11}^2$  indicating an invalid TPI solution.

The pericynthion altitude ( $q_{1d}$ ) is computed by subtracting the landing site radius ( $J^5$ ) from the transfer orbit pericynthion ( $q_1$ ),  $q_1$  is zeroed, and the program branches to the external  $\Delta V$  routine (Figure 7.20). In all cases,  $\delta_{10}$  is zeroed before exiting from TPI. The TPI routine computes  $\underline{X}_{bD}$  using  $\underline{V}_G$  as described in the external  $\Delta V$  mode. The external  $\Delta V$  equations compute the velocity-to-be-gained vector valid at time  $t_{ig}$ .

The TPI mode is completed in 20 to 34 of the 50 2-sec minor cycle computations depending on the overflow paths taken.

7. FLOW DIAGRAMS OF FLIGHT PROGRAM

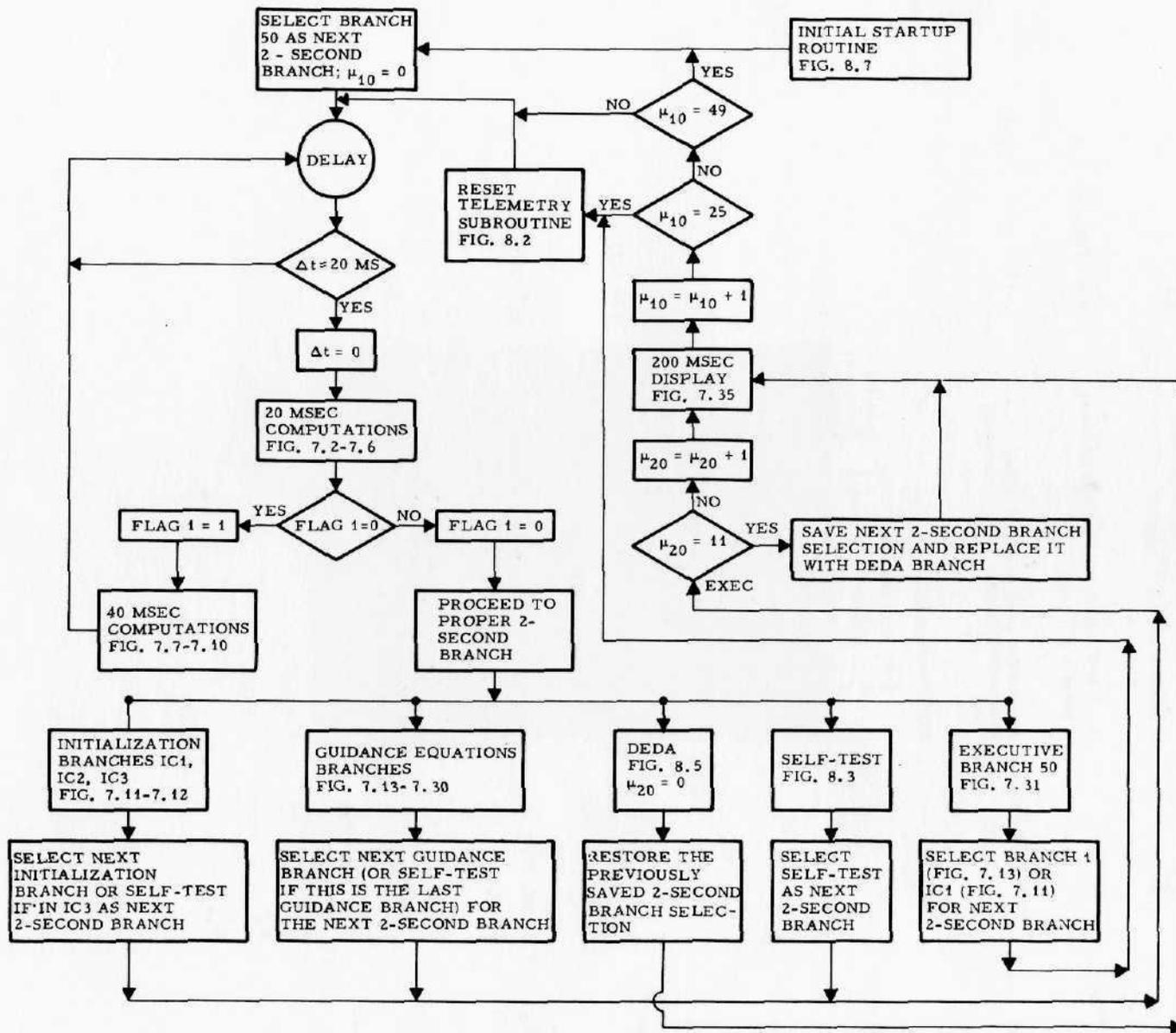


Figure 7.1. AGS Total Software System Block Diagram

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>																		
$\theta_{pi}, \psi_{pi}, \phi_{pi}$	THEP PHIP PSIP	Input PGNCS Euler angles (pulses) 1 pulse = $2\pi \times 2^{-15}$ rad	$\pm 2^{14}$	1																		
$\Delta a_{Xi}, \Delta a_{Yi}, \Delta a_{Zi}$	*	X, Y and Z gyro raw outputs (pulses) 1 pulse = $2^{-16}$ rad	0 to 1220	1																		
$\Delta \sigma_{Xrem}$	DAXREM	Remainder used to obtain extra precision in X gyro input processing (rad)	$2^{-13}$	$2^{-30}$																		
$S_{00}$	S00	AGS Function Selector = <table style="display: inline-table; vertical-align: middle;"> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">0</td> <td style="padding: 0 5px;">Attitude Hold</td> <td rowspan="3" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">} Inertial Reference Mode</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">1</td> <td style="padding: 0 5px;">Guidance Steering</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">2</td> <td style="padding: 0 5px;">Z Axis Steering</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">3</td> <td style="padding: 0 5px;">PGNCS/AGS Align</td> <td rowspan="5" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">} Align Mode</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">4</td> <td style="padding: 0 5px;">Lunar Align</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">5</td> <td style="padding: 0 5px;">Body Axis Align</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">6</td> <td style="padding: 0 5px;">Calibrate</td> </tr> <tr> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">7</td> <td style="padding: 0 5px;">Accelerometer Calibrate</td> </tr> </table>	0	Attitude Hold	} Inertial Reference Mode	1	Guidance Steering	2	Z Axis Steering	3	PGNCS/AGS Align	} Align Mode	4	Lunar Align	5	Body Axis Align	6	Calibrate	7	Accelerometer Calibrate		
0	Attitude Hold	} Inertial Reference Mode																				
1	Guidance Steering																					
2	Z Axis Steering																					
3	PGNCS/AGS Align	} Align Mode																				
4	Lunar Align																					
5	Body Axis Align																					
6	Calibrate																					
7	Accelerometer Calibrate																					
$\Delta V_{Xi}, \Delta V_{Yi}, \Delta V_{Zi}$	*	X, Y and Z accelerometer raw outputs (pulses) 1 pulse = 1/320 fps	0 to 1220	1																		
$\Delta V_X, \Delta V_Y, \Delta V_Z (\Delta \underline{V})$	DVX, DVY, DVZ	Compensated incremental velocity components accumulated per 20 msec along X, Y, Z body axes, respectively, (fps)	$\pm 2^1$	$2^{-16}$																		
$\Delta V_{m-1}$	DVXM1, DVYM1, DVZM1	Compensated incremental velocity components accumulated per 20 msec along X, Y, Z body axes, respectively, (fps) for the previous 20 msec cycle	$\pm 2^1$	$2^{-16}$																		
Constants:	$K_{18}^1, K_{20}^1, K_{22}^1$ $K_{19}^1, K_{21}^1, K_{23}^1$ $K_7^1$ $K_{14}^1$																					

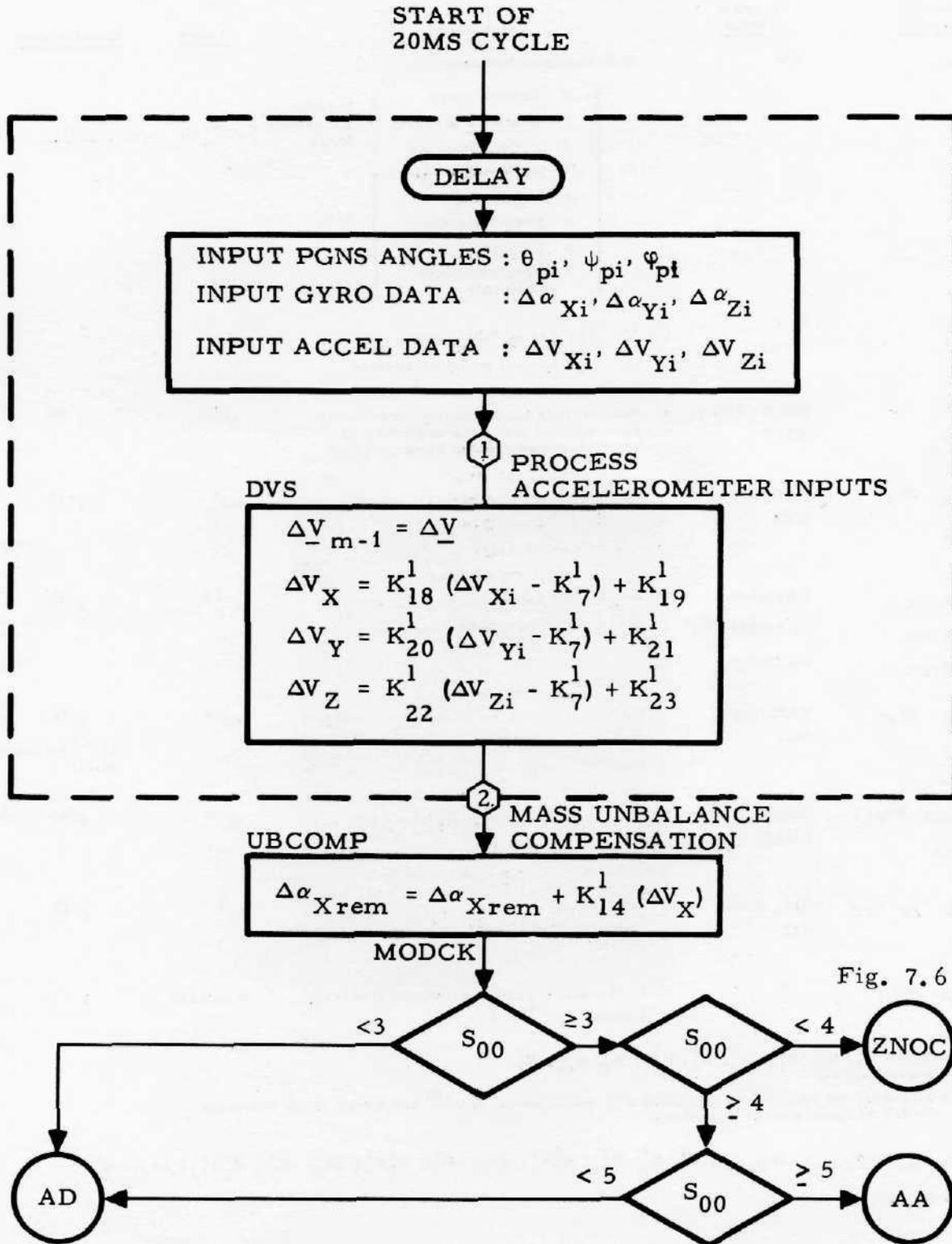


Figure 7.2. Start of 20-msec Computations, Input Data

Equation Variable	Program Symbol	Definition	Range	Quantization
$S_{00}$	S00	AGS Function Selector		
		$= \left\{ \begin{array}{l} 0 \text{ Attitude Hold} \\ 1 \text{ Guidance Steering} \\ 2 \text{ Z Axis Steering} \\ 3 \text{ PGNCS/AGS Align} \\ 4 \text{ Lunar Align} \\ 5 \text{ Body Axis Align} \\ 6 \text{ Calibrate} \\ 7 \text{ Accelerometer Calibrate} \end{array} \right\}$	$\left. \begin{array}{l} \text{Inertial Reference Mode} \\ \text{Align Mode} \end{array} \right\}$	
$\delta_{21}$	DEL21	$= \left\{ \begin{array}{l} 1 \text{ LM on lunar surface} \\ 0 \text{ LM not on lunar surface} \end{array} \right\}$		
$\Delta V_X, \Delta V_Y, \Delta V_Z$	SDVX, SDVY, SDVZ	Sum of body axis velocity increments accumulated since the beginning of accelerometer bias calibration (fps)	$\pm 2^1$	$2^{-16}$
$\Delta V_X, \Delta V_Y, \Delta V_Z$	DVX, DVY, DVZ	Compensated incremental velocity components accumulated per 20 msec along X, Y, Z body axes, respectively (fps)	$\pm 2^1$	$2^{-16}$
$\Delta \alpha_{Xrem}, \Delta \alpha_{Yrem}, \Delta \alpha_{Zrem}$	DAXREM, DAYREM, DAZREM	$\Delta \alpha_X, \Delta \alpha_Y, \Delta \alpha_Z$ remainders used to obtain extra precision (rad)	$2^{-13}$	$2^{-30}$
$\Delta \alpha_X, \Delta \alpha_Y, \Delta \alpha_Z$	DAX, DAY, DAZ	Compensated incremental components of vehicle rotation per 20 msec about the X, Y and Z body axes, respectively (rad)	$\pm 2^{-6}$	$2^{-16}$ ( $2^{-23}$ in lunar align)
$\Delta \alpha_{XA}, \Delta \alpha_{YA}, \Delta \alpha_{ZA}$	DAXA, DAYA, DAZA	Alignment error signals about LM X, Y, Z body axes, respectively (rad) (zero in all modes except lunar align, $S_{00} = 4$ )	$\pm 2^{-6}$	$2^{-23}$
$(a_{12}, a_{22}, a_{32})$	A12, A22, A32	Unit vector coinciding with the y inertial axes expressed in LM body-axes coordinates	$\pm 2^1$	$2^{-15}$
$\Delta \alpha_{Xi}, \Delta \alpha_{Yi}, \Delta \alpha_{Zi}$	*	X, Y, and Z gyro raw outputs (pulses); 1 pulse = $2^{-16}$ rad	0 to 1220	1
Constants: $K_3^1, K_8^1, K_{13}^1, K_1^1, K_6^1, K_{11}^1, K_7^1, K_{56}^1, K_2^1$				
† The following $\Delta \alpha$ terms are computed at a quantization of $2^{-30}$ and added to $\Delta \alpha$ when the remainder accumulates $2^{-16}$ radians.				
$\Delta \alpha_{Xrem}, \Delta \alpha_{Yrem}, \Delta \alpha_{Zrem}, K_1^1, K_6^1, K_{11}^1, K_2^1 K_3^1 (\Delta \alpha_{Xi} - K_7^1), K_2^1 K_8^1 (\Delta \alpha_{Yi} - K_7^1), K_2^1 K_{13}^1 (\Delta \alpha_{Zi} - K_7^1)$				
and $K_{14}^1 \Delta V_X$				

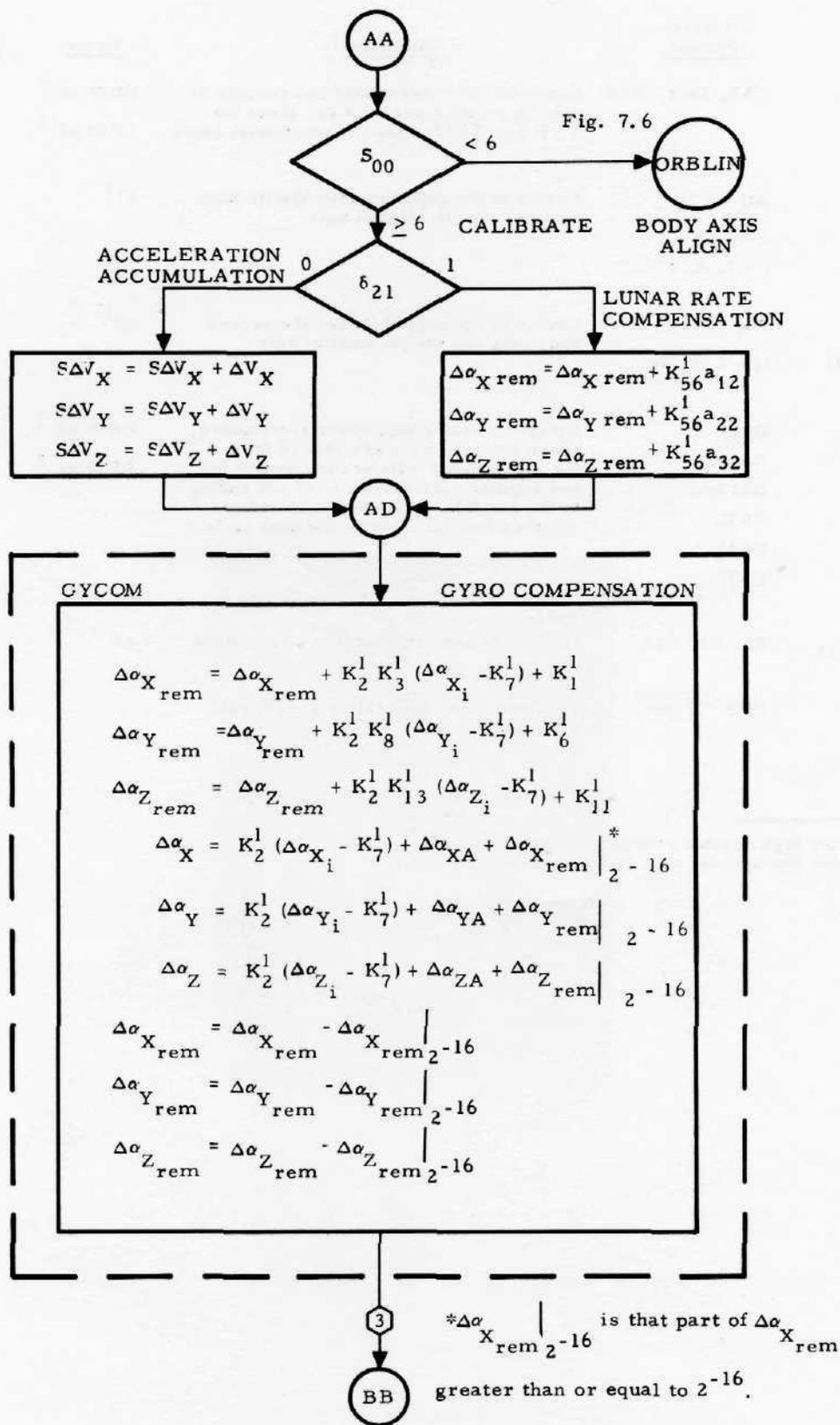


Figure 7.3. Gyro Compensation

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\Delta a_X, \Delta a_Y, \Delta a_Z$	DAX, DAY, DAZ	Compensated incremental components of vehicle rotation per 20 msec about the X, Y and Z body axes, respectively, (rad)	HRS* $\pm 2^{-6}$ LRS* $\pm 2^{-9}$	$2^{-16}$ $2^{-16}$
$a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	Aij $i = 1, 3$ $j = 1, 2, 3$	Cosine of the angle between the ith body axis and the jth inertial axis	$\pm 2^1$	$2^{-16}$
$a_{2j}$ $(j = 1, 2, 3)$	A2j $(j = 1, 2, 3)$	Cosine of the angle between the second body axis and the jth inertial axis	$\pm 2^1$	$2^{-15}$
$\Delta a_{11}, \Delta a_{12}, \Delta a_{13}, \Delta a_{31}, \Delta a_{32}, \Delta a_{33}$	DA11, DA12, DA13, DA31, DA32, DA33	Direction cosine increment remainders. (Each time the $\Delta a$ 's are shifted from $2^{-5}$ for high angular rate scaling or $2^{-8}$ for low angular rate scaling to $2^1$ for adding to the a's, the remainder is saved and summed into the $\Delta a$ 's on the next cycle.)	HRS* $\pm 2^{-5}$ LRS* $\pm 2^{-8}$	$2^{-22}$ $2^{-25}$
$E_1, E_3, E_{13}$	E1, E3, E13	Normality and orthogonality corrections	$\pm 2^{-9}$	$2^{-16}$
HRF	HRF	Flag indicating high (1) or low (0) rate gyro inputs		

\*HRS denotes high angular rate scaling (see Figure 7.4)  
LRS denotes low angular rate scaling (see Figure 7.4)

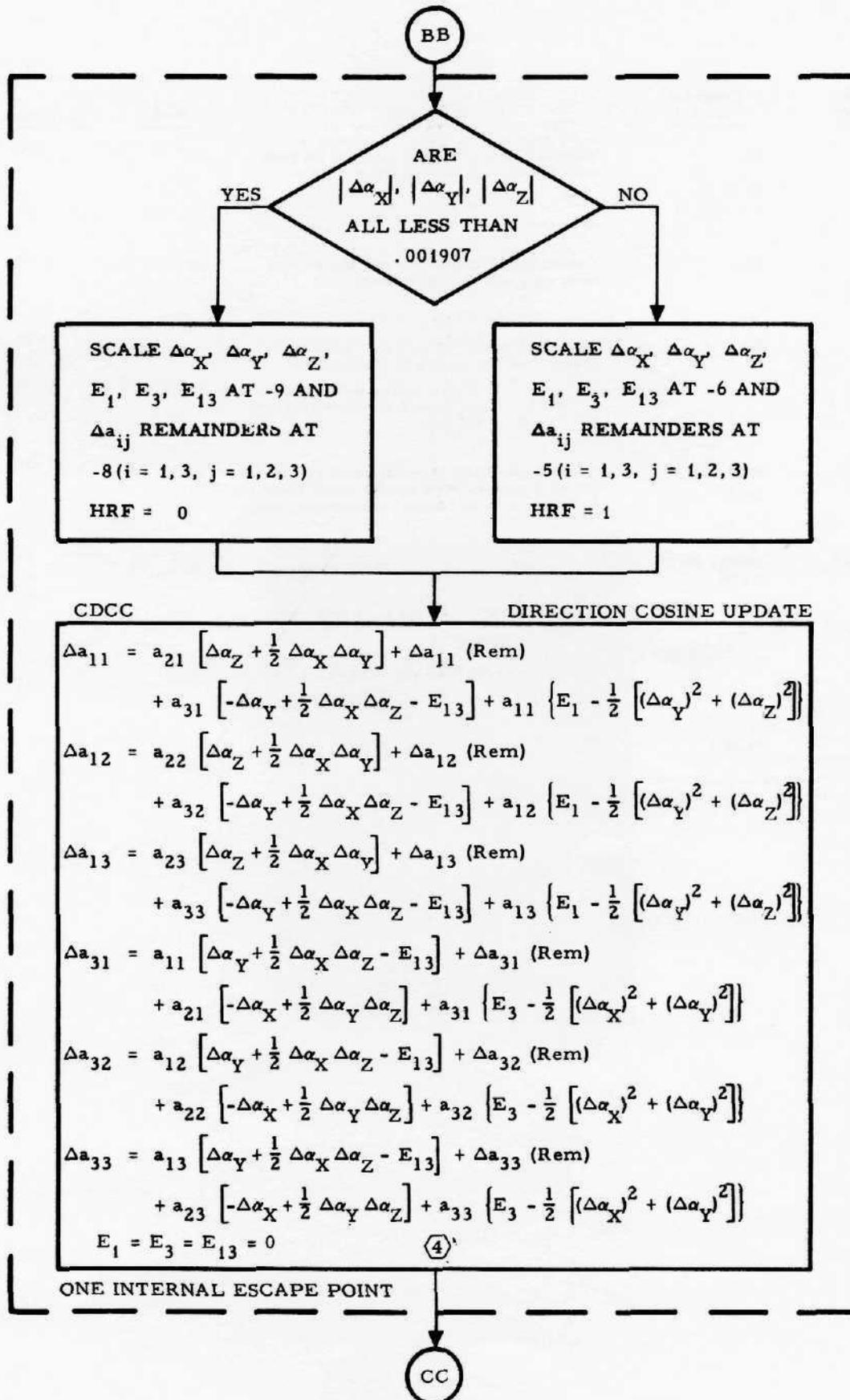


Figure 7.4. Direction Cosine Change

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	Aij $i = 1, 3$ $j = 1, 2, 3$	Cosine of the angle between the $i$ th body axis and the $j$ th inertial axis	$\pm 2^1$	$2^{-16}$
$a_{2j}$ $(j = 1, 2, 3)$	A2j	Cosine of the angle between the second body axis and the $j$ th inertial axis	$\pm 2^1$	$2^{-15}$
$\Delta a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	DAij	Direction cosine increment remainders (Each time the $\Delta a$ 's are shifted from $2^{-5}$ or $2^{-8}$ to $2^1$ for adding to the $a$ 's the remainder is saved and summed into the $\Delta a$ 's on the next cycle)	HRS* $\pm 2^{-5}$ LRS* $\pm 2^{-8}$	$2^{-22}$ $2^{-25}$
$\Delta V_X, \Delta V_Y,$ $\Delta V_Z$	DVX, DVY, DVZ	Compensated incremental velocity components accumulated per 20 msec along X, Y, and Z body axes, respectively (fps)	$\pm 2^1$	$2^{-16}$
$\Delta V_{xs}, \Delta V_{ys},$ $\Delta V_{zs}$ (Also designated $\Delta V_{-s}$ )	DVSX, DVSY, DVSZ	Accumulated total of sensed velocity increments along the x, y and z inertial axes, respectively, from the start of the present 2 sec computer cycle (fps)	$\pm 2^7$	$2^{-10}$
HRF	HRF	Flag indicating high (1) or low (0) rate gyro inputs		
FLAG 1	FLAG1	2 sec/40 msec logic control flag		

\*HRS denotes high angular rate scaling (See Figure 7.4)  
LRS denotes low angular rate scaling (See Figure 7.4)

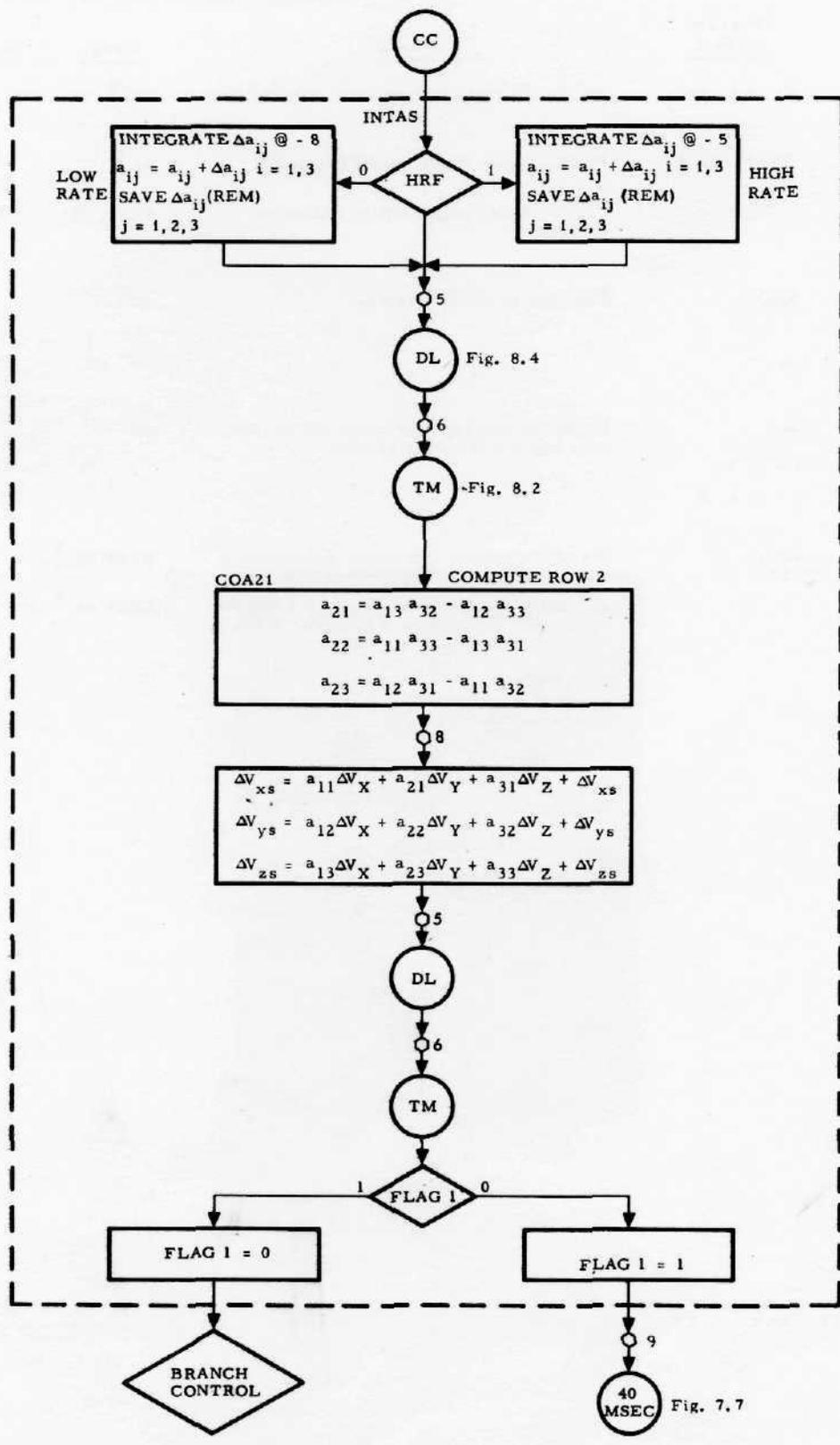


Figure 7.5. Direction Cosine Update and Velocity Resolution

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$E_1, E_3, E_{13}$	E1, E3, E13	Normality and orthogonality corrections	$\pm 2^{-9}$	$2^{-16}$
$\theta_{pi}, \psi_{pi}, \phi_{pi}$	THEP, PHIP, PSIP	PGNCS Euler angle inputs (pulses)	$\pm 2^{14}$	1
$\theta_p, \psi_p, \phi_p$	*	PGNCS Euler angle inputs scaled in radians	0 to $+ 2\pi$	$2\pi \times 2^{-15}$
$a_{ijD}$ $i = 1, 3$ $j = 1, 2, 3$	AijD	Desired direction cosines	$\pm 2^1$	$2^{-16}$
$a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	Aij $i = 1, 3$ $j = 1, 2, 3$	Cosine of the angle between the ith body axis and the jth inertial axis	$\pm 2^1$	$2^{-16}$
$\Delta a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	DAij	Direction cosine increment remainders (Each time the $\Delta a$ 's are shifted from $2^{-5}$ or $2^{-8}$ to $2^1$ for adding to the a's the remainder is saved and summed into the $\Delta a$ 's on the next cycle)	HRS* $\pm 2^{-5}$ LRS* $\pm 2^{-8}$	$2^{-22}$ $2^{-25}$

Constants:  $K_{25}^1$

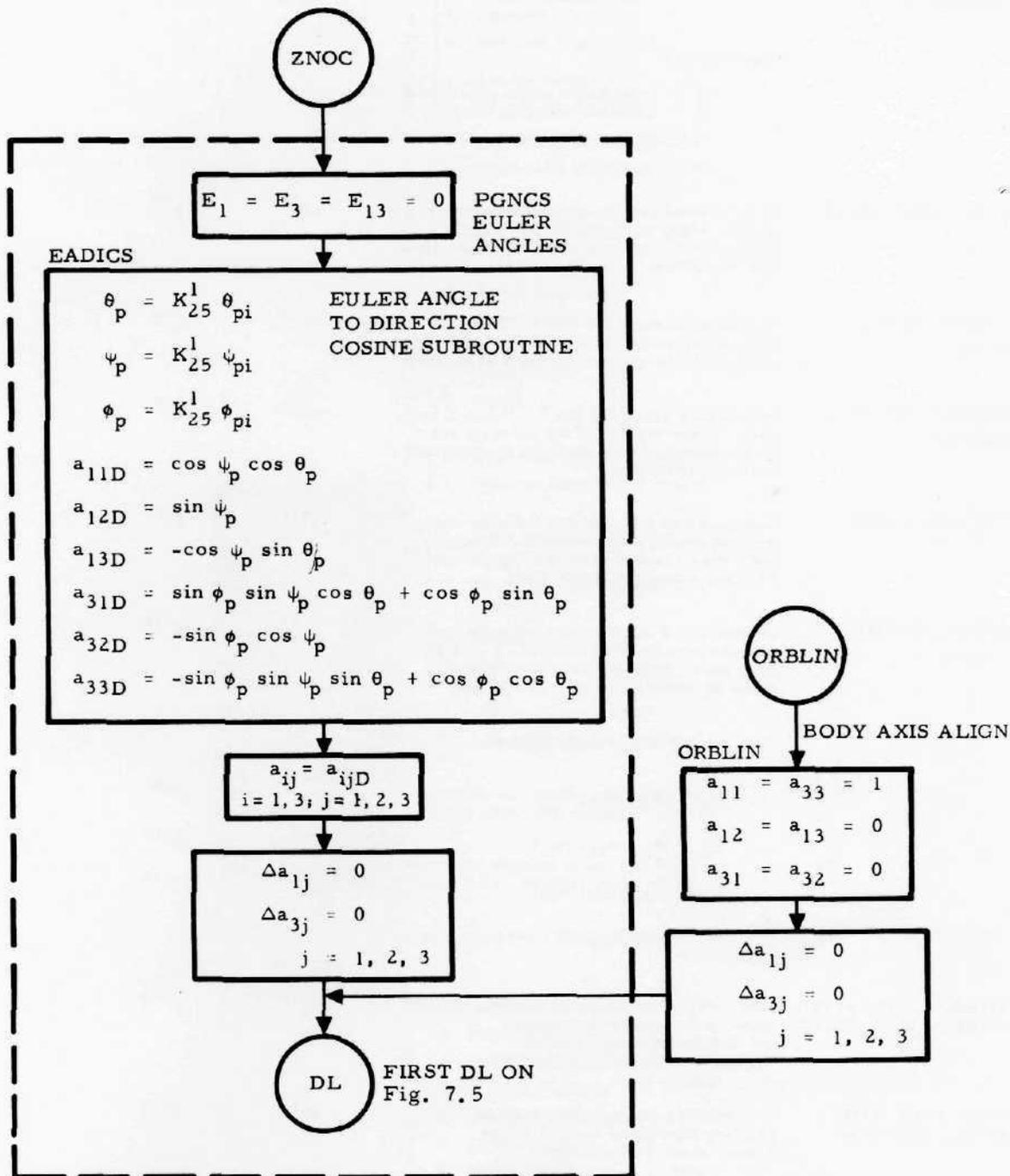


Figure 7.6. Processing of PGNCs Angles, Body Axis Align Computations

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\beta_1$	*	= { 1 Descent Engine ON 0 Descent Engine OFF					Engine Select		
$\beta_2$	*	= { 1 Ascent Engine ON 0 Ascent Engine OFF			$S_{11}$	S11	= { 1 Select cant angle corrections to guidance steering attitude errors. 0 Do not select cant angle corrections.		
$\beta_3$	*	= { 1 Follow-up signal present 0 Follow-up signal not present			$V_{dX}, V_{dY}, V_{dZ}$	VD1X, VD1Y, VD1Z	Accumulated total of sensed velocity increments, along X, Y and Z body axes, respectively, updated every 40 msec (fps) The remainder is saved to $2^{-15}$ .	$\pm 2^{13}$	$2^{-4}$
$\beta_4$	*	= { 1 Auto Discrete YES 0 Auto Discrete NO			$V_{gX}, V_{gY}, V_{gZ}$	VSMGX, VSMGY, VSMGZ	Components along the LM X, Y and Z body axes, respectively, of the total LM velocity-to-be-gained during a burn (fps)	$\pm 2^{13}$	$2^{-4}$
$\beta_5$	*	= { 1 Abort Stage YES 0 Abort Stage NO			$\Delta V_{gX}, \Delta V_{gY}, \Delta V_{gZ}$	DELVGX, DELVGY, DELVGZ	Increments along the LM X, Y and Z body axes, respectively, of the velocity yet to-be-gained during the burn (determined each 40 msec) (fps)	$\pm 2^{13}$	$2^{-4}$
$\beta_6$	*	= { 1 Abort YES 0 Abort NO			$\Delta V_X, \Delta V_Y, \Delta V_Z$	DVX, DVY, DVZ	Compensated incremental velocity components accumulated along X, Y and Z body axes respectively, during current 20 msec computing cycle (fps)	$\pm 2^1$	$2^{-16}$
$\delta_2$	DEL 2	= { 1 Descent section staged 0 Descent section not staged			$\Delta V_{X,m-1}, \Delta V_{Y,m-1}, \Delta V_{Z,m-1}$	DVXM1, DVYM1, DVZM1	Compensated incremental velocity components accumulated along X, Y and Z body axes, respectively, during previous 20 msec computing cycle (fps)	$\pm 2^1$	$2^{-16}$
$\delta_5$	DEL 5	= { 1 Lock Attitude Hold reference direction cosines 0 Set reference cosines to body cosines			$\underline{x}_b, \underline{y}_b, \underline{z}_b$		Unit vectors of LM body attitude		
$\delta_{20}$	DEL 20	= { 1 Engine follow-up 0 AEA engine decision					$\underline{x}_b = (a_{11}, a_{12}, a_{13})$	$\pm 2^1$	$2^{-16}$
$\delta_{21}$	DEL 21	= { 1 LM on lunar surface 0 LM not on lunar surface					$\underline{y}_b = (a_{21}, a_{22}, a_{23})$	$\pm 2^1$	$\pm 2^{-15}$
$\mu_6$	MU6	Delay counter; maintains attitude hold momentarily after LM stages					$\underline{z}_b = (a_{31}, a_{32}, a_{33})$	$\pm 2^1$	$\pm 2^{-16}$
$E_X, E_Y, E_Z$	EX, EY, EZ	Attitude error signals about LM X, Y and Z body axes, respectively (rad)	$\pm 2^2$	$2^{-15}$			For definition of $a_{11}$ , etc., see Figure 7.4		
Output $E_X, E_Y, E_Z$		Output attitude error signals (deg)	$\pm 15^\circ$	$15 \times 2^{-9}$	$\underline{x}_{bD}, \underline{z}_{bD}$	A11BD, A12BD, A13BD A31BD, A32BD, A33BD	Unit vectors commanding desired pointing directions in Guidance and Z Axis steering, respectively (determined in 2 sec cycle)	$\pm 2^1$	$2^{-16}$
$S_{00}$	S00	AGS Function Selector = { 0 Attitude Hold } Inertial Reference Mode { 1 Guidance Steering } { 2 Z Axis Steering } { 3 PGNCS/AGS Align } { 4 Lunar Align } Align Mode { 5 Body Axis Align } { 6 Calibrate } { 7 Accelerometer Calibrate }			$\underline{x}_D, \underline{z}_D$	A11D, A12D, A13D A31D, A32D, A33D	Unit vectors commanding desired pointing directions for LM X and Z body axes, respectively	$\pm 2^1$	$2^{-16}$
					$\underline{w}_b$	WBX, WBY, WBZ	Unit vector defining plane used for yaw steering ( $S_{623} = 1$ ) in the guidance steering mode ( $S_{00} = 1$ )	$\pm 2^1$	$\pm 2^{-16}$
					$S_{623}$	S623	= { 1 Steer Z body axis parallel to plane defined by its unit normal $\underline{w}_b$ 0 Steer Z body axis parallel to CSM orbit plane		
					$S_{507}$	S507	= { 1 Steer Z body axis in computed thrust direction 0 Steer Z body axis in computed CSM direction		
Constants: $K_7^4, K_8^4, K_{23}^4, K_{21}^4, \underline{w}_b$									

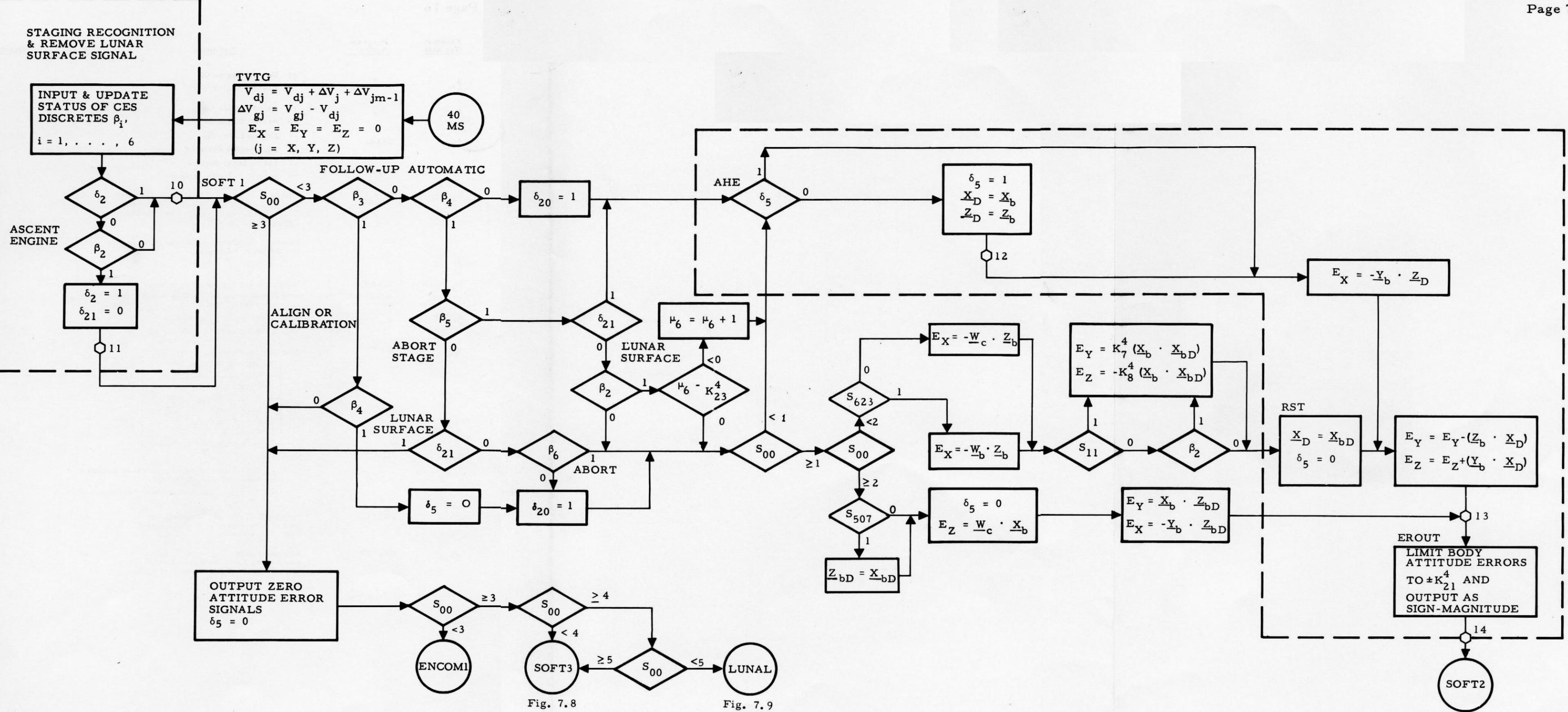


Figure 7.7. 40-msec Attitude Error Computations

Equation Variable	Program Symbol	Definition	Range	Quantization
$\beta_1$	*	= $\begin{cases} 1 & \text{Descent Engine ON} \\ 0 & \text{Descent Engine OFF} \end{cases}$		
$\beta_2$	*	= $\begin{cases} 1 & \text{Ascent Engine ON} \\ 0 & \text{Ascent Engine OFF} \end{cases}$		
$\delta_{20}$	DEL20	= $\begin{cases} 1 & \text{Engine follow-up} \\ 0 & \text{AEA engine decision} \end{cases}$		
$\delta_{21}$	DEL21	= $\begin{cases} 1 & \text{LM on Lunar Surface} \\ 0 & \text{LM not on Lunar Surface} \end{cases}$		
$\mu_8$	MU8	Ullage counter ( $K_9^1$ consecutive 2-second compute cycles with thrust acceleration $a_T \geq K_{35}^2$ are required before Engine ON signal is issued.)	$2^{17}$	$2^0$
$a_{ij}$ $i = 1, 3$ $j = 1, 2, 3$	Aij	Cosine of the angle between the $i$ th body axis and the $j$ th inertial axis	$\pm 2^1$	$2^{-16}$
$E_1, E_3, E_{13}$ $E_{ON+S_{10}}$	E1, E3, E13 EONS10	Normality and orthogonality errors Combination engine ON/OFF and $S_{10}$ for telemetry (see Figure 8.2)	$2^{-9}$	$2^{-16}$
E/OFF	*	Engine OFF signal from AGS		
E/ON	*	Engine ON signal from AGS		
$S_{00}$	S00	AGS Function Selector $= \begin{cases} 0 & \text{Attitude Hold} \\ 1 & \text{Guidance Steering} \\ 2 & \text{Z Axis Steering} \\ 3 & \text{PGNCS/AGS Align} \\ 4 & \text{Lunar Align} \\ 5 & \text{Body Axis Align} \\ 6 & \text{Calibrate} \\ 7 & \text{Accelerometer Calibrate} \end{cases}$		
			Inertial Reference Mode	
			Align Mode	
$S_{11}$	S11	Engine Select $= \begin{cases} 1 & \text{Select cant angle correction to guidance steering attitude errors.} \\ 0 & \text{Do not select cant angle corrections.} \end{cases}$		
$\Delta V_G$	DELVG	Magnitude of the velocity yet-to-be-gained during the burn (fps). Updated every two seconds.	$\pm 2^{13}$	$2^{-4}$
$\Delta V_{gX}$	DELVGX	Increment of the total velocity-to-be-added along LM X-body axis during the burn (fps). Updated every 40 msec.	$\pm 2^{13}$	$2^{-4}$

Constants:  $K_9^1, K_{25}^4, K_{26}^4$

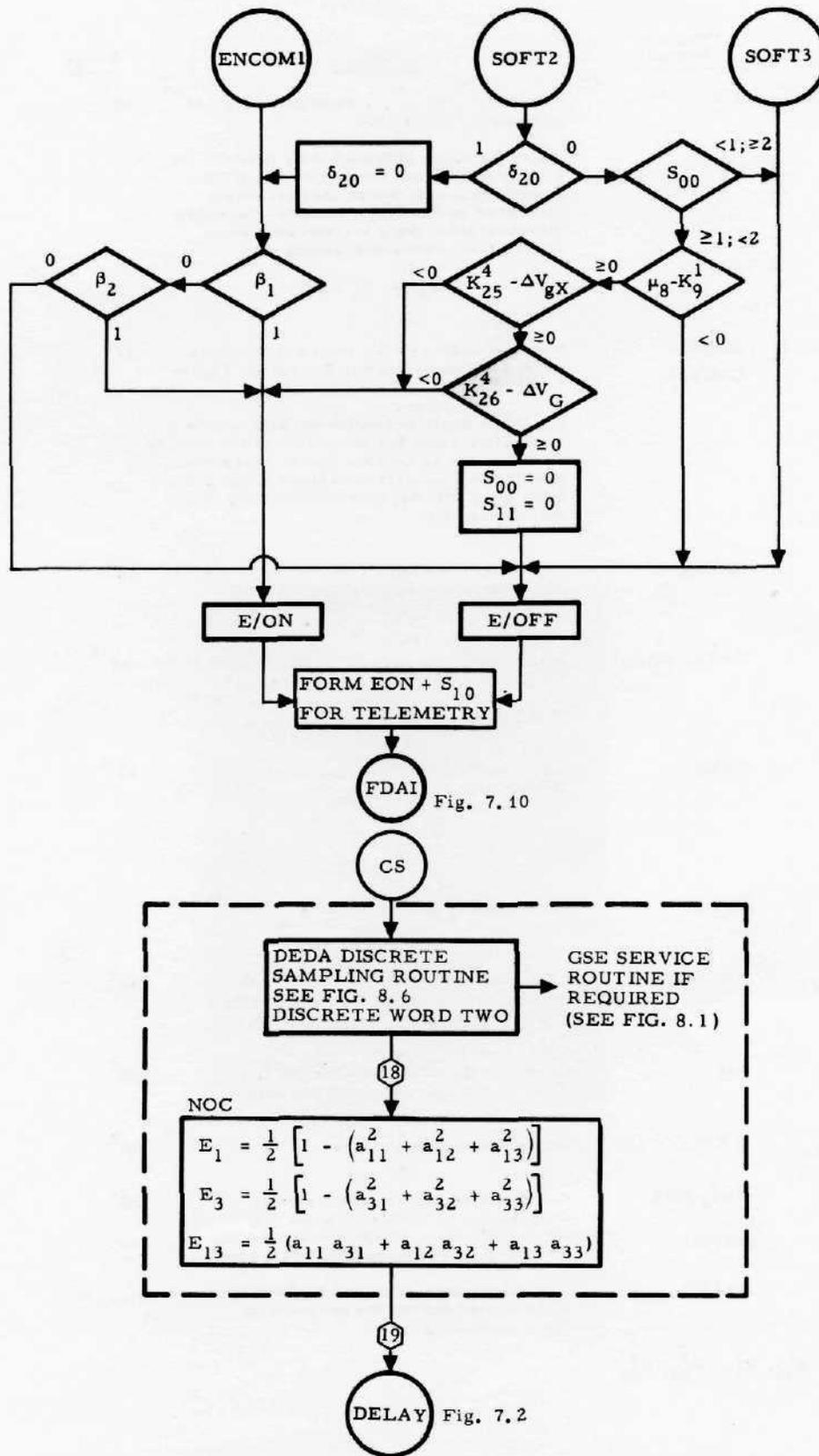


Figure 7.8. Engine Discrete Logic; Direction Cosine Normality and Orthogonality Computations

Equation Variable	Program Symbol	Definition	Range	Quantization
$\sin \delta_A, \cos \delta_A$	*,*	Sine and cosine of the azimuth of the LM at nominal launch time  $\delta_A$ is the angle at launch time between the y inertial axis and the projection of the Z-body axis onto the yz inertial plane, measured positive in a counter-clockwise direction from the y inertial axis when viewed from above the landing site  $\delta_A = \delta_L - \Delta\delta$	$\pm 2^1$	$2^{-16}$
$\sin \delta_L, \cos \delta_L$	SIDELL, CODELL	Sine and cosine of the stored landing azimuth $\delta_L$ (Determined in Branch 50, Figure 7.36)  $\delta_L$ is the angle at touchdown between the y inertial axis and the projection of the Z-body axis onto the yz inertial plane, measured positive in a counter-clockwise direction from the y inertial axis when viewed from the landing site	$\pm 2^1$	$2^{-16}$
$\Delta\delta$	DDEL	Change in LM azimuth during lunar stay (measured in yz inertial plane) (rad) (Input in octal via the DEDA)	$\pm 2^0$	$2^{-15}$
$\Delta\alpha_{iA}$ $i = Y, Z$	DAYA, DAZA	Alignment error signals about Y and Z body axes (rad) (Division by 4 necessary to reduce loop gain because of accelerometer pulse moding)	$\pm 2^{-8}$	$2^{-23}$
$\Delta\alpha_{XA}$	DAXA	Alignment error signals about the X-body axis (rad)	$\pm 2^{-6}$	$2^{-23}$
$\delta_{42}$	DEL42	This quantity has no meaning		
$a_{21}, a_{22}, a_{23}$	A21, A22, A23	Cosine of the angle between the LM Y-body axis and the x, y and z inertial axes, respectively	$\pm 2^1$	$2^{-15}$
$a_{31}$	A31	Cosine of the angle between the LM Z-body axis and the x inertial axis	$\pm 2^1$	$2^{-16}$
$a_{32D}, a_{33D}$	*,*	Desired direction cosines	$\pm 2^1$	$2^{-16}$
$\Delta V_Y, \Delta V_Z$	DVY, DVZ	Compensated incremental velocity components accumulated along LM Y and Z body axes, respectively, during current 20 msec computing cycle (fps)	$\pm 2^1$	$2^{-16}$
$\Delta V_{Y, m-1}$	DVYM1			
$\Delta V_{Z, m-1}$	DVZM1	The subscript m-1 refers to the values accumulated during the previous 20 msec computing cycle		
Constants:	$K_{26}^1, K_{27}^1, K_{28}^1, K_{29}^1$			

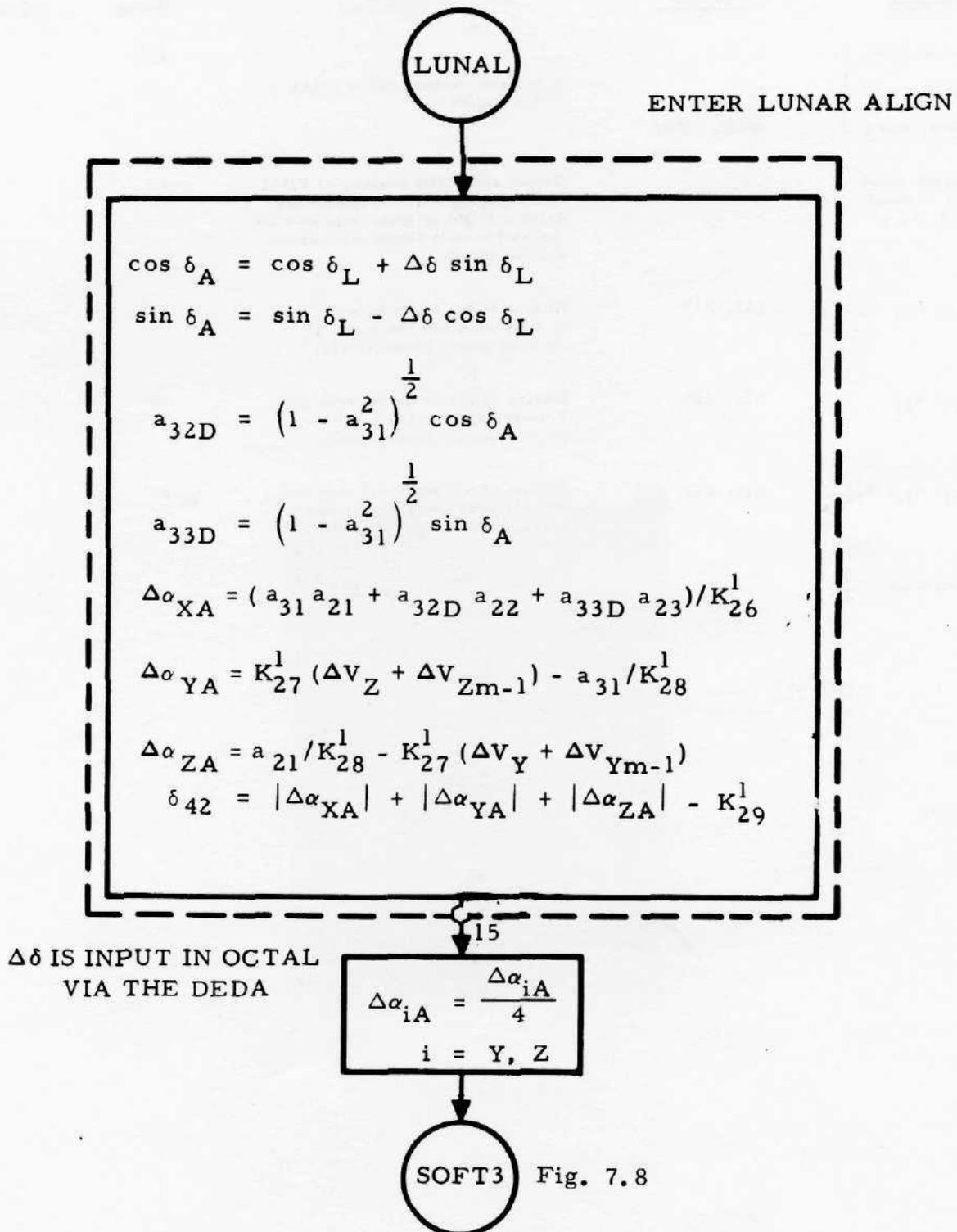


Figure 7.9. Lunar Align Computations

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
sin $\alpha$ , cos $\alpha$ } sin $\beta$ , cos $\beta$ } sin $\gamma$ , cos $\gamma$ }	*, * *, * SIGA, COGA	Sines and cosines of the FDAI Euler angles	$\pm 2^1$	$2^{-16}$
Output sines and cosines of $\alpha$ , $\beta$ , $\gamma$	*	Output sines and cosines of FDAI Euler angles. These quantities drive a flight attitude indicator on the astronaut's flight instrument display panel.	$\pm 1$	$2^{-9}$
$a_{11}$ , $a_{13}$	A11, A13	Cosine of the angle between the X-body axis and the x and z inertial axes, respectively.	$\pm 2^1$	$2^{-16}$
$a_{21}$ , $a_{23}$	A21, A23	Cosine of the angle between the Y-body axis and the x and z inertial axes, respectively	$\pm 2^1$	$2^{-15}$
$a_{31}$ , $a_{32}$ , $a_{33}$	A31, A32, A33	Cosine of the angle between the Z-body axis and the x, y and z inertial axes, respectively.	$\pm 2^1$	$2^{-16}$
Constants: $K_{24}^1$				

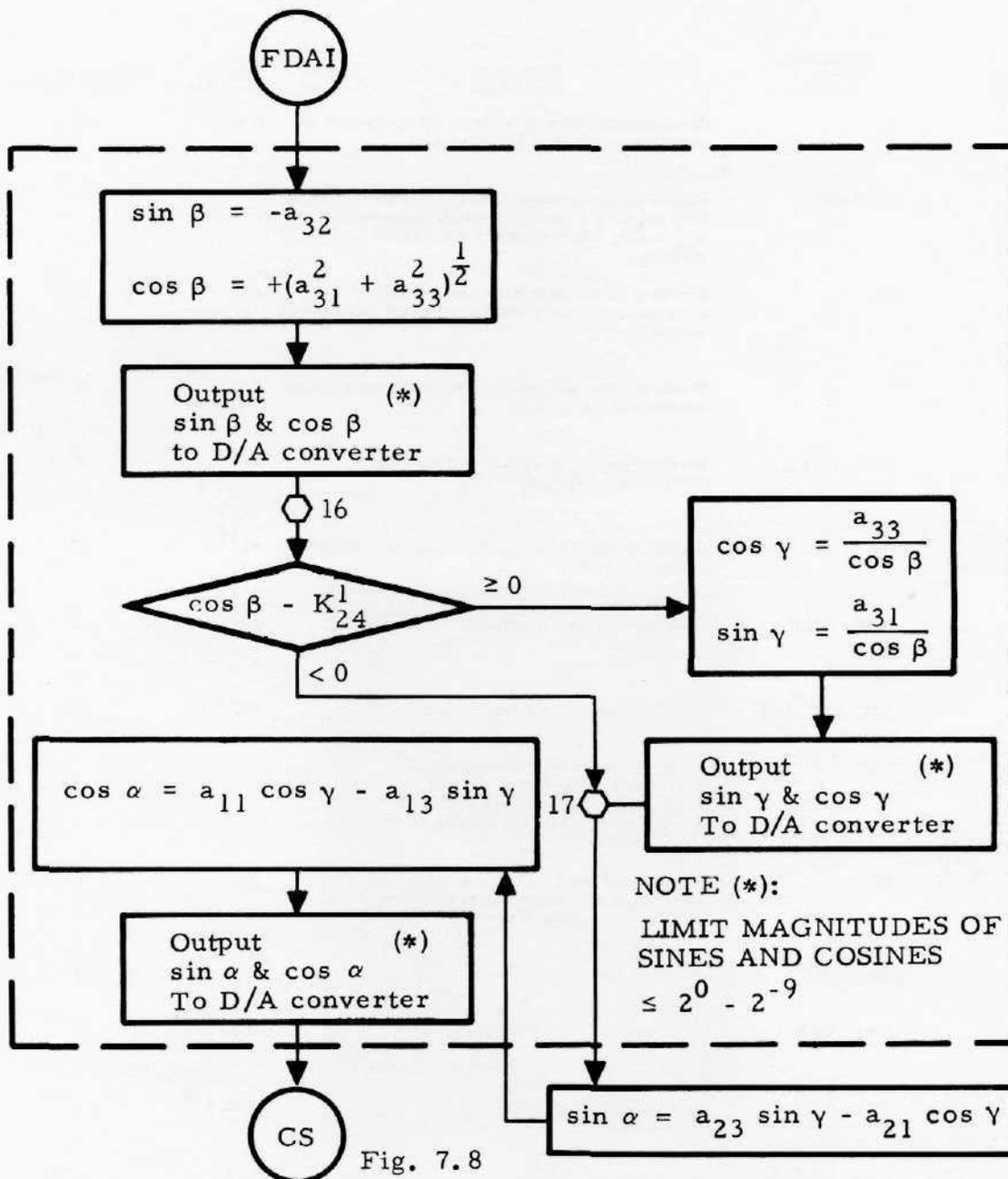


Fig. 7.8

Figure 7.10. FDAI Computations

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$a_i$	AI	Semi-major axis of ellipse as computed in orbit parameter subroutine (ft).	0 to $2^{23}$	$2^6$
$\mu_{20}$	DEDASC	DEDA cycle counter (program enters DEDA loop when it reaches branch counter with $\mu_{20} = 11$ ). Implemented as a shift counter.		
$C_i$	CI	Product of eccentricity and cosine of eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$n_i$	NI	Mean orbital rate output by orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$
$\underline{r}_0$	R0X, R0Y, R0Z	Inertial position vector input to orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$r_i$	R0	Radial distance output by orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$\underline{r}_E$	REX, REY, REZ	CSM ephemeris position vector (ft)	$\pm 2^{23}$	$2^6$
$\underline{r}_L$	1J1, 1J2, 1J3	LM ephemeris position vector (ft)	$\pm 2^{23}$	$2^6$
$S_{14}$	S14	= $\begin{cases} 0 & \text{Initialization complete} \\ 1 & \text{Initialize LM and CSM via Downlink} \\ 2 & \text{Initialize LM via DEDA} \\ 3 & \text{Initialize CSM via DEDA} \end{cases}$		
$S_i$	SI	Product of eccentricity and sine of the eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$t$	TA1	Present absolute time (sec)	0 to $2^{18}$	$2^1$
$t_E$	TE1, TE2	CSM epoch time (sec)	0 to $2^{18}$	$2^{-10}$
$t_L$	TL1, TL2	LM epoch time (sec)	0 to $2^{18}$	$2^{-10}$
$\underline{v}_0$	V0X, V0Y, V0Z	Inertial velocity vector input to orbit parameter subroutine (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_E$	VEX, VEY, VEZ	CSM ephemeris velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_L$	1J4, 1J5, 1J6	LM ephemeris velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$

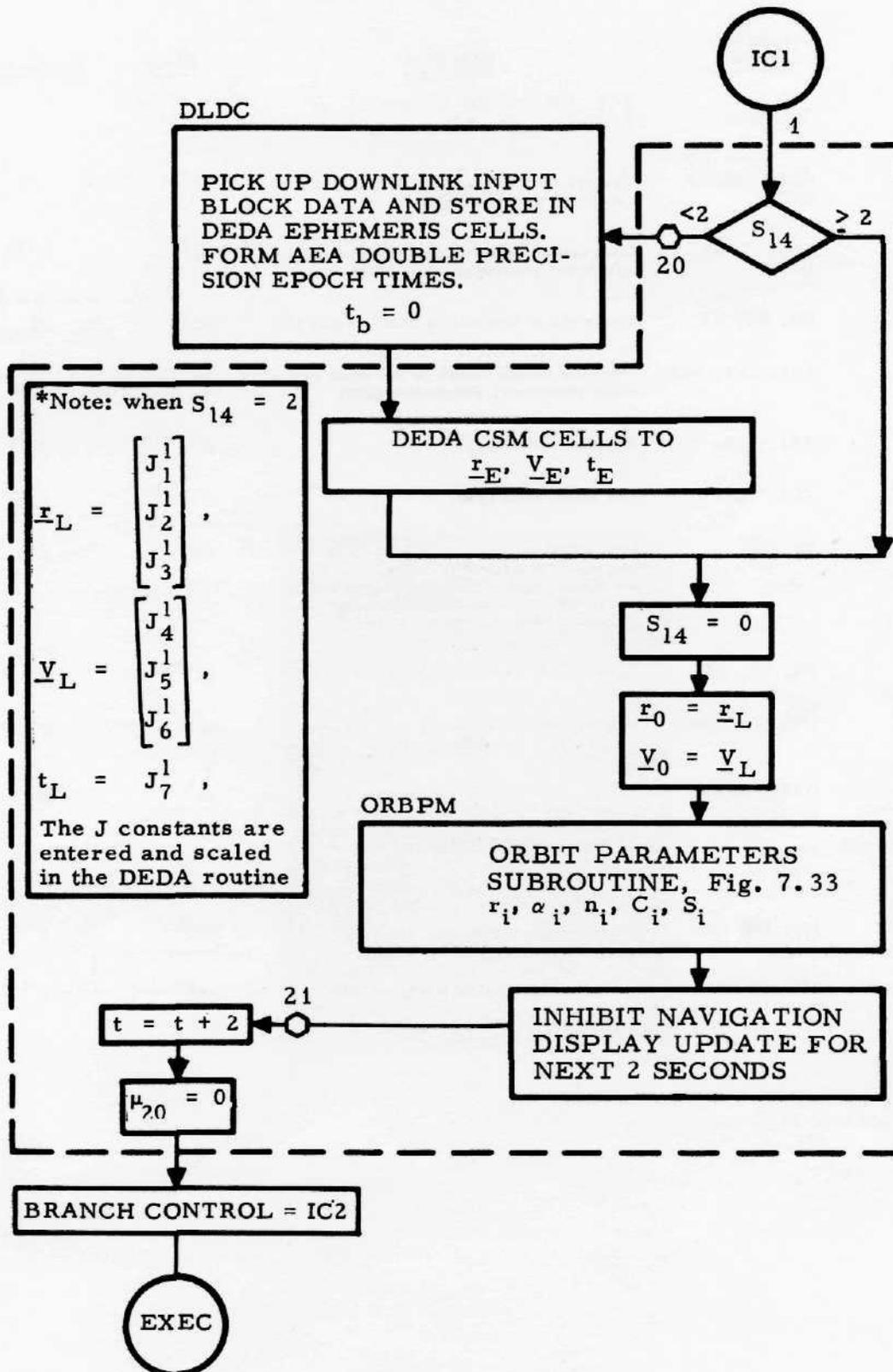


Figure 7.11. LM Initialization, Branch IC1

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\delta_{21}$	DEL21	= $\begin{cases} 1 & \text{LM on Lunar Surface} \\ 0 & \text{LM not on Lunar Surface} \end{cases}$		
$\underline{G}\Delta t$	GXDT, GYDT, GZDT	Gravity vector computed in present 2 sec computer cycle (fps <sup>2</sup> )	$\pm 2^7$	$2^{-10}$
$\underline{\Delta IG}$	DIGX, DIGY, DIGZ	Integrated gravity acceleration vector per 2-sec computer cycle (fps)	$\pm 2^7$	$2^{-10}$
$\underline{r}$	RX, RY, RZ	Present LM inertial position vector (ft)	$\pm 2^{23}$	$2^6$
$\underline{r}_0$	R0X, R0Y, R0Z	Position vector which is the input to the orbit parameter subroutine (fps)	$\pm 2^{23}$	$2^6$
$t$	TA1, TA2	Present time (sec)	0 to $2^{18}$	$2^{-16}$
$t_L$	TL1, TL2	LM epoch time (sec)	0 to $2^{18}$	$2^{-10}$
$T_i$	TI	Prediction time input to ellipse prediction subroutine. It is measured from the epoch time corresponding to the ephemeris data from which the orbit parameters were computed (sec)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}$	VX, VY, VZ	Present LM inertial velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_0$	V0X, V0Y, V0Z	Velocity vector which is input to the orbit parameter subroutine (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{\Delta v}_s$	DVSX, DVSY, DVSZ	Vector of the sensed velocity increments, along the x, y and z inertial axes, accumulated during the 2 sec computer cycle ( $\underline{v}_s$ updated every 20 msec, but only used every 2 sec)	$\pm 2^7$	$2^{-10}$
$\underline{r}_L$	1J1, 1J2, 1J3	LM ephemeris position vector (ft)	$\pm 2^{23}$	$2^6$
$\underline{v}_L$	1J4, 1J5, 1J6	LM ephemeris velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$

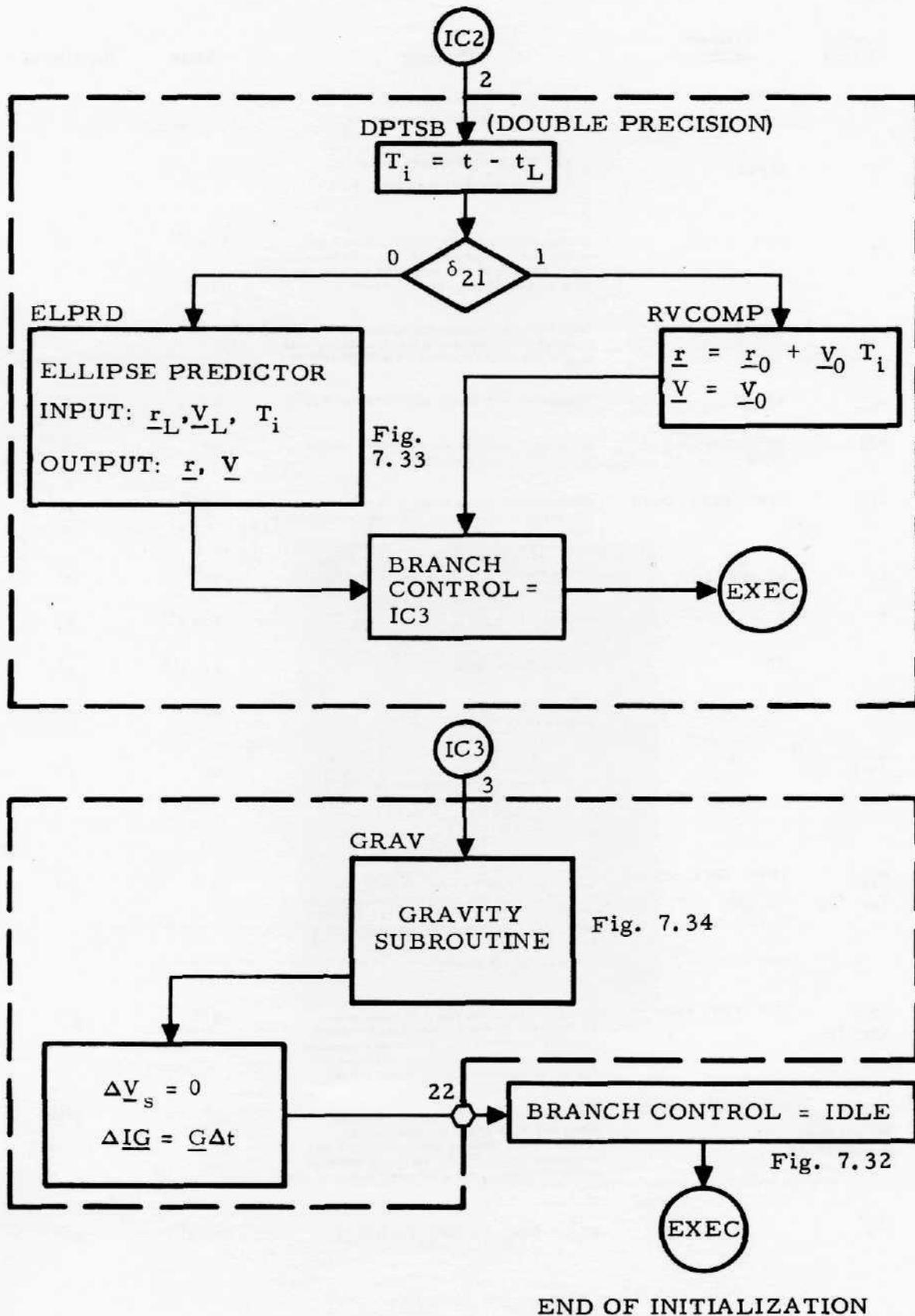


Figure 7.12. LM State Vectors, Branch IC2, Gravity and Display Computations, IC3

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\delta_{21}$	DEL21	= $\begin{cases} 1 & \text{LM on Lunar Surface} \\ 0 & \text{LM not on Lunar Surface} \end{cases}$		
$\delta_{33}$	DSPF1	= $\begin{cases} 1 & \text{Navigation update cycle} \\ 0 & \text{Not navigation update cycle} \end{cases}$		
$\mu_8$	MU8	Ullage counter (1K9 consecutive 2 sec computer cycles with thrust acceleration $a_T \geq 4K35 \text{ fps}^2$ are required before the Engine ON signal can be issued.)	0 to $2^{17}$	$2^0$
$\mu_{20}$	DEDASC	DEDA cycle counter (program enters DEDA loop when it reaches branch counter with $\mu_{20} = 11$ ) Implemented as a shift counter		
$a_T$	AT	Computed LM thrust acceleration ( $\text{fps}^2$ )	0 to $2^7$	$2^{-5}$
$\Delta IG$	DIGX, DIGY, DIGZ	Integrated gravity acceleration vector per 2 sec computer cycle ( $\text{fps}$ )	$\pm 2^7$	$2^{-10}$
$\Delta g_s$	DQSX, DQSY, DQSZ	Accumulated total of $\Delta V_s$ , the vector of the sensed velocity in inertial coordinates ( $\text{fps}$ )	$\pm 2^{13}$	$2^{-4}$
$\underline{r}$	RX, RY, RZ	Present LM inertial position vector (ft)	$\pm 2^{23}$	$2^6$
$r$	R	Magnitude of $\underline{r}$ (ft)	0 to $2^{23}$	$2^6$
$t$	TA1	Present time (sec)	0 to $2^{18}$	$2^1$
$\underline{V}$	VX, VY, VZ	Present LM inertial velocity vector ( $\text{fps}$ )	$\pm 2^{13}$	$2^{-4}$
$\underline{V}_{n-1}$	*	LM inertial velocity vector computed in previous 2 sec computer cycle ( $\text{fps}$ )	$\pm 2^{13}$	$2^{-4}$
$V_{dX}, V_{dY}, V_{dZ}$	VD1X, VD1Y, VD1Z	Components along the LM body axes, X, Y and Z, of the accumulated sensed velocity increments which are updated every 40 msec ( $\text{fps}$ ). Remainder is saved to $2^{-16}$ .	$\pm 2^{13}$	$2^{-4}$
$V_{DX}, V_{DY}, V_{DZ}$	VDX, VDY, VDZ	Components along the LM body axes, X, Y and Z of the accumulated sensed velocity increments, which are updated only every 2 sec. ( $V_{Di}$ is set equal to $V_{di}$ at navigation update time.) ( $\text{fps}$ )	$\pm 2^{13}$	$2^{-4}$
$\Delta V_{xs}, \Delta V_{ys}, \Delta V_{zs}$ (Also designated $\Delta V_s$ )	DVSX, DVSY, DVSZ	Total of sensed velocity increments along the x, y and z inertial axes, respectively, during most recent 2 sec computer cycle ( $\text{fps}$ ). ( $\Delta V_{is}$ is updated every 20 msec, but only used every 2 sec)	$\pm 2^7$	$2^{-10}$
$\Delta V_s^*$	*	$\Delta V_s^* =  \Delta V_{xs}  +  \Delta V_{ys}  +  \Delta V_{zs} $	0 to $2^7$	$2^{-10}$

Constants:  $K_9^1, K_{35}^1, K_{34}^4, K_{35}^4$

BEGIN "2 SEC COMPUTATIONS"

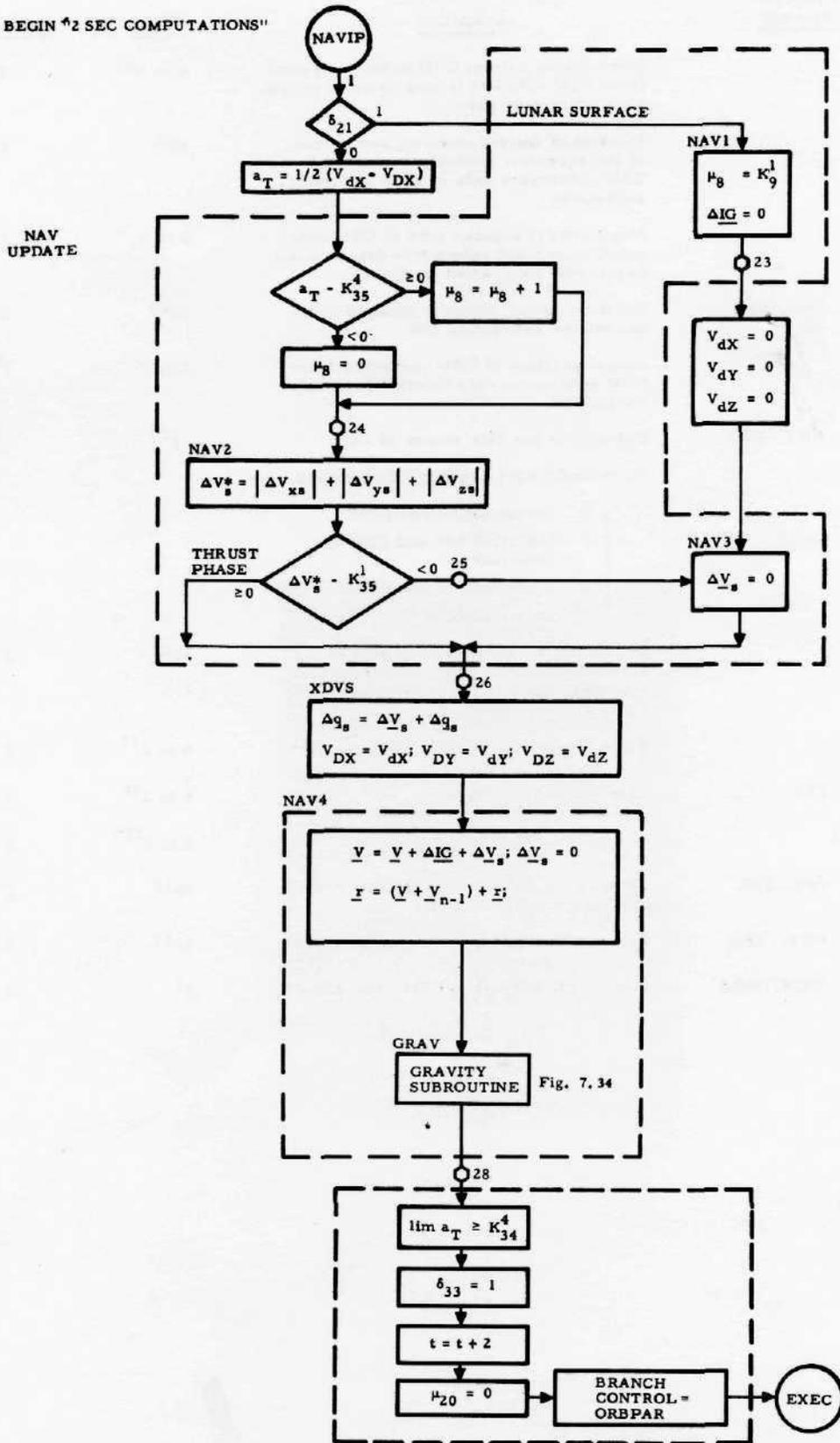


Figure 7.13. Navigation Update Equations, Branch 1

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$a_E$	AE	Semi-major axis of CSM orbit computed from CSM ephemeris data in orbit parameter subroutine (ft)	0 to $2^{23}$	$2^6$
$C_E$	CI	Product of the eccentricity and cosine of the eccentric anomaly computed from CSM ephemeris data in orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$n_E$	NE	Mean orbital angular rate of CSM computed from CSM ephemeris data in orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$
$\underline{r}_0$	R0X, R0Y, R0Z	Position vector which is input to orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$r_E$	R0	Radial position of CSM computed from CSM ephemeris data in orbit parameter subroutine	0 to $2^{23}$	$2^6$
$\underline{r}_E$	REX, REY, REZ	Ephemeris position vector of CSM	$\pm 2^{23}$	$2^6$
$S_{14}$	S14	Astronaut DEDA request - Ephemeris $= \begin{cases} 0 & \text{Initialization complete} \\ 1 & \text{Initialize LM and CSM via Downlink} \\ 2 & \text{Initialize LM via DEDA} \\ 3 & \text{Initialize CSM via DEDA} \end{cases}$		
$S_E$	SI	Product of eccentricity and sine of the eccentric anomaly computed from CSM ephemeris data in orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$t_b$	TB	Time elapsed since CSM epoch, $t_E$ (sec)	0 to $2^{13}$	$2^{-4}$
$t_E$	TE1, TE2	CSM epoch time (sec)	0 to $2^{18}$	$2^{-10}$
$T_{CSM}$	TCSM	Period of CSM orbit (sec)	0 to $2^{13}$	$2^{-4}$
$\underline{V}_0$	V0X, V0Y, V0Z	Velocity vector which is input to orbit parameter subroutine (ft)	$\pm 2^{13}$	$2^{-4}$
$\underline{V}_E$	VEX, VEY, VEZ	Ephemeris velocity vector of CSM (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{W}_c$	WCX, WCY, WCZ	Unit vector normal to CSM orbit plane	$2^1$	$2^{-16}$

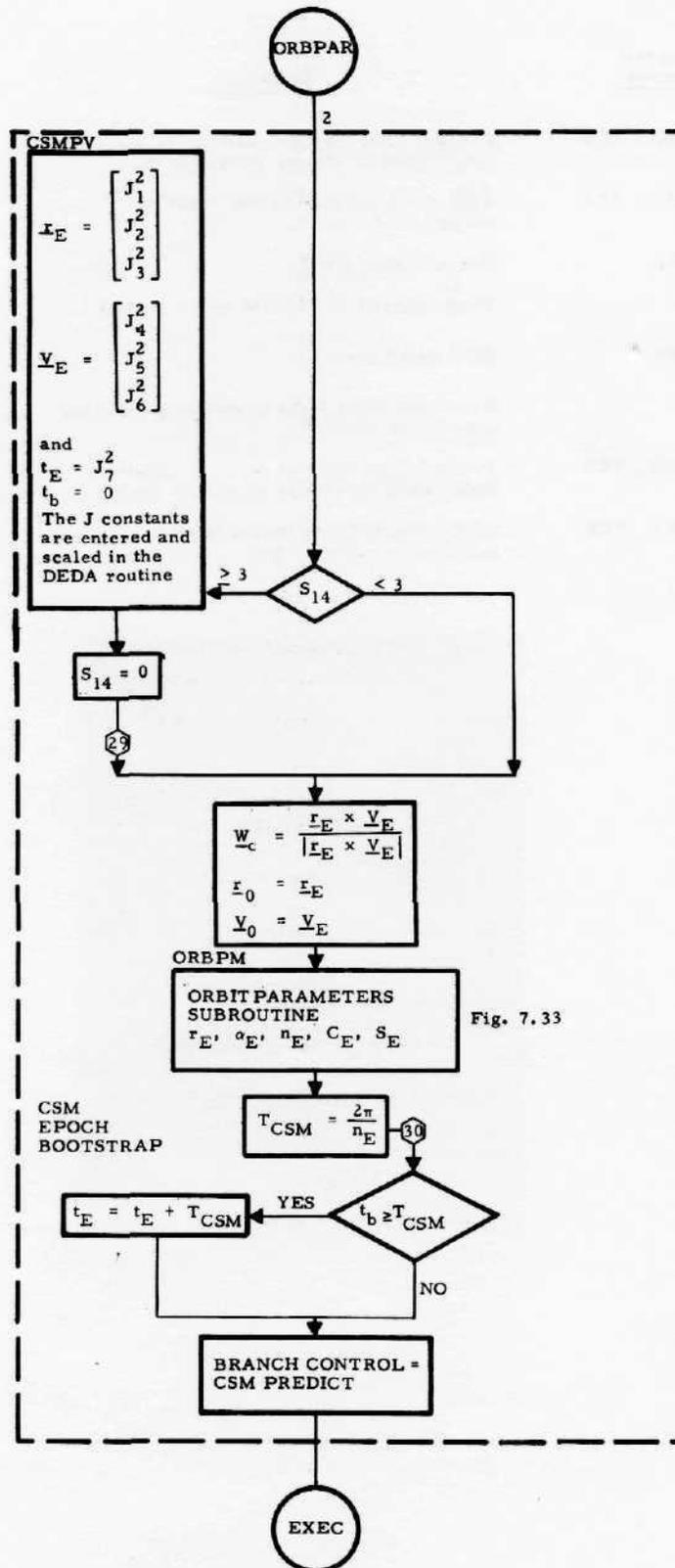


Figure 7.14. Computation of CSM Orbit Parameters from Ephemeris Data, Branch 2

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\underline{r}_c$	RCX, RCY, RCZ	Present CSM inertial position vector as determined by ellipse predictor (ft)	$\pm 2^{23}$	$2^6$
$\underline{r}_E$	REX, REY, REZ	CSM epoch position vector input to ellipse predictor (ft)	$\pm 2^{23}$	$2^6$
t	TA1, TA2	Present time (sec)	0 to $2^{18}$	$2^{-16}$
$t_b$	TB	Time elapsed since CSM epoch, $t_E$ (sec)	0 to $2^{13}$	$2^{-4}$
$t_E$	TE1, TE2	CSM epoch (sec)	0 to $2^{18}$	$2^{-10}$
$T_i$	TI	Prediction time input to ellipse prediction subroutine (sec)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_c$	VCX, VCY, VCZ	Present CSM inertial velocity vector as determined by ellipse predictor (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_E$	VEX, VEY, VEZ	CSM epoch velocity vector input to ellipse predictor (fps)	$\pm 2^{13}$	$2^{-4}$

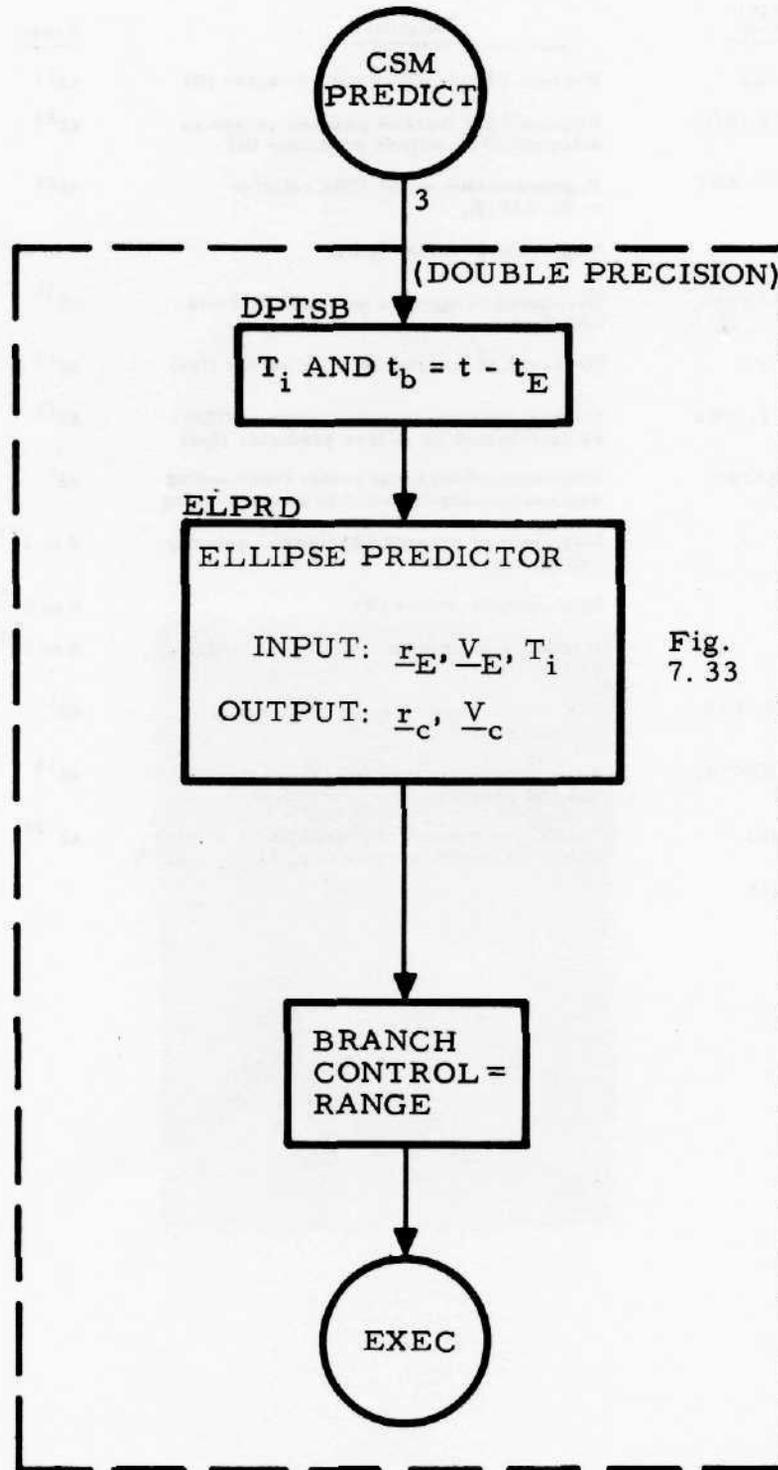


Fig. 7.33

Figure 7.15. Prediction of Present CSM Position and Velocity, Branch 3

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\underline{r}$	RX, RY, RZ	Present LM inertial position vector (ft)	$\pm 2^{23}$	$2^6$
$\underline{r}_c$	RCX, RCY, RCZ	Present CSM inertial position vector as determined by ellipse predictor (ft)	$\pm 2^{23}$	$2^6$
$\underline{R}$	RRX, RRY, RRZ	Position vector of the CSM relative to the LM (ft)	$\pm 2^{23}$	$2^6$
R	RR	Magnitude of vector $\underline{R}$ (ft)	0 to $2^{23}$	$2^6$
$\dot{R}$	RRDOT	Estimated range rate between CSM and LM (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}$	VX, VY, VZ	Present LM inertial velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_c$	VCX, VCY, VCZ	Current inertial velocity vector of CSM as determined by ellipse predictor (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{z}_{bD}$	A31BD, A32BD, A33BD	Unit vector along range vector commanding desired pointing direction in Z axis steering	$\pm 2^1$	$2^{-16}$
V	V	Magnitude of present LM inertial velocity vector (fps)	0 to $2^{13}$	$2^{-4}$
$J^{25}$	25J	Input altitude update (ft)	0 to $2^{23}$	$2^6$
r	R	Magnitude of present LM inertial position vector (ft)	0 to $2^{23}$	$2^6$
$\underline{u}_1$	U1X, U1Y, U1Z	Unit vector along present LM inertial position vector	$\pm 2^1$	$2^{-16}$
$\dot{R}$	RDOTX, RDOTY, RDOTZ	Relative velocity vector between the CSM and LM (fps)	$\pm 2^{13}$	$2^{-4}$
a, j, c	LILA, LILJ, LILC	Partial derivatives of gravitational acceleration vector with respect to $r_x$ and $r_z$ ( $\text{sec}^{-2}$ )	$\pm 2^{-15}$	$2^{-32}$
Constants: $K_1^2, J^5, J^{25}$				

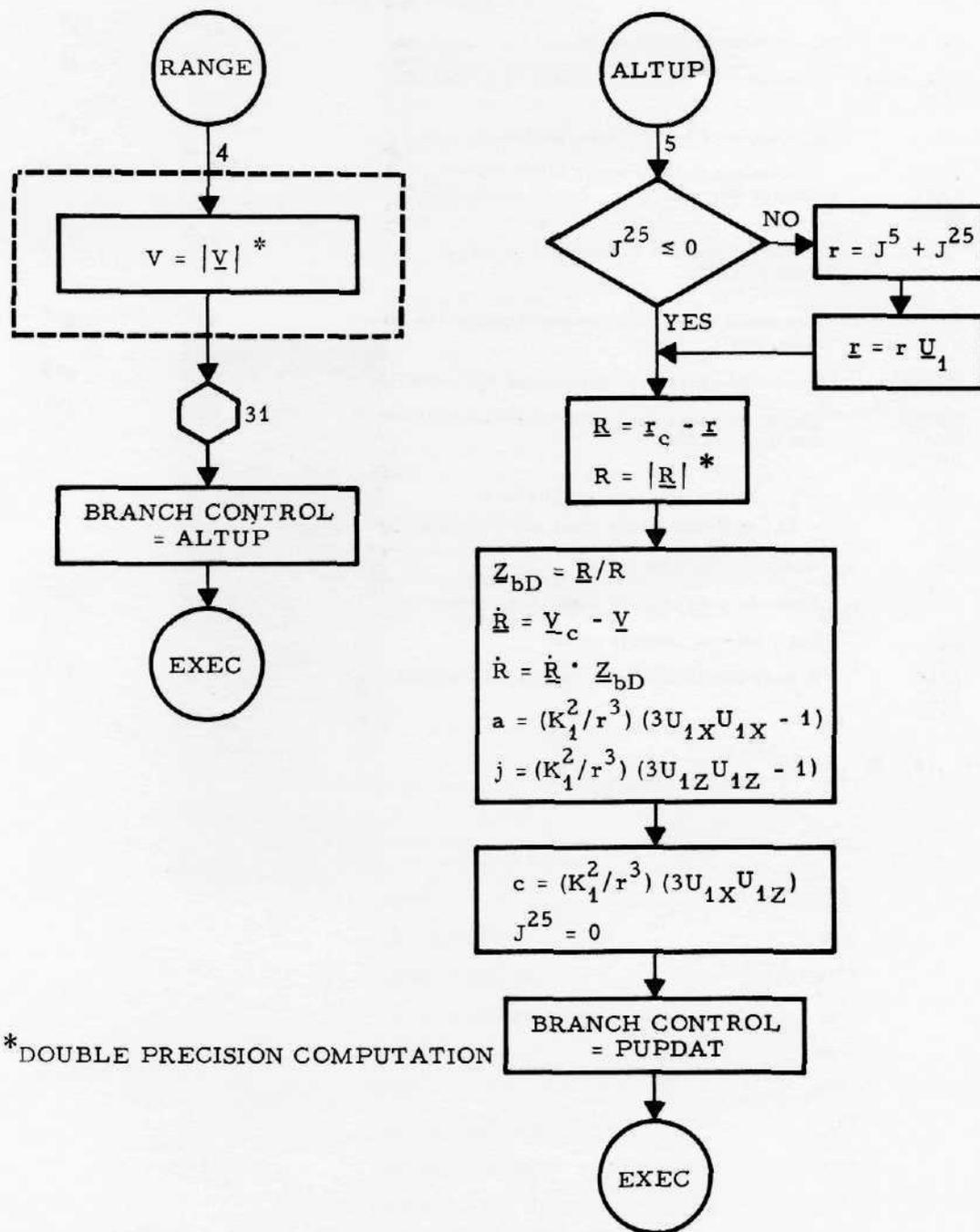


Figure 7.16. Total Velocity, Altitude Update, Range and Range Rate

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\mu_{17}$	MU17	Radar filter P matrix update counter	0 to $2^{17}$	$2^0$	$q_{11}$	Q11	Elements of the intermediate computation symmetric matrix Q	$\pm 2^{30}$ $\pm 2^{10}$	$2^{13}$ $2^{-7}$
$\Delta$	DELTA	Determinant of Q $\begin{cases} \text{RANGE (ft}^4\text{)} \\ \text{RATE (fps)}^2 \end{cases}$	$\pm 2^{60}$ $\pm 2^{10}$	$2^{43}$ $2^{-7}$	$q_{12}$	Q12		$\pm 2^{30}$	$2^{13}$
$\Delta r_1$	DELR1	RANGE: difference between x components of input range and computed range (ft) RATE: difference between input range rate and computed range rate (fps)	$\pm 2^{23}$ $\pm 2^{13}$	$2^6$ $2^{-4}$	$q_{22}$	Q22		$\pm 2^{30}$ $\pm 2^0$	$2^{13}$ $2^{-17}$
$\Delta r_2, \Delta r_3$	DELR2, DELR3	RANGE: differences between z and y components, respectively, of input range and computed range (ft) RATE: unused	$\pm 2^{23}$	$2^6$	$r_y$	RY	LM y inertial position (ft)	$\pm 2^{23}$	$2^6$
$\Delta t$	DELTAT	Time between last radar range input and latest $S_{15}$ entry (sec)	$2^{18}$	$2^1$	$R_x, R_y, R_z$	RRX, RRY, RRZ	Computed range vector from LM to CSM (ft)	$\pm 2^{23}$	$2^6$
$a_{31bD}, a_{32bD}, a_{33bD}$	A31BD, A32BD, A33BD	Unit vector along LM-CSM range	$\pm 2^1$	$2^{-16}$	R	RR	Magnitude of LM-CSM range (ft)	$2^{23}$ $\pm 2^{23}$	$2^6$ $2^6$
a, c, j	LILA, LILC, LILJ	Partial derivatives of gravitational acceleration vector with respect to $r_x$ and $r_z$ ( $\text{sec}^{-2}$ )	$\pm 2^{-15}$	$2^{-32}$	$R_x^*, R_y^*, R_z^*$	RRSX, RRSY, RRSZ	Computed LM-CSM range saved when $S_{15}$ entered (ft)	$\pm 2^{23}$	$2^6$
$J^{17}$	17J	Range rate DEDA input (fps)	$\pm 2^{13}$	$2^{-4}$	$R_x^{**}, R_y^{**}, R_z^{**}$	*, *, *	Computed inertial components of range input $J^{18}$ (ft)	n $\pm 2^{23}$	$2^6$
$J^{18}$	18J	Range DEDA input (input in nmi, stored in ft)	0 to $2^{23}$	$2^6$	R	RD0T	Estimated range rate between LM and CSM (fps)	$\pm 2^{13}$	$2^{-4}$
$L_{i2}$ i = 1, 2, 3, 4	Li2	RANGE: Set to second row of P matrix RATE: Set to zero	$\pm 2^{13}$	$2^{-4}$	$\dot{R}^*$	RD0TS	Computed range rate saved when $S_{15}$ input (fps)	$\pm 2^{13}$	$2^{-4}$
$L_{i1}$ i = 1, 2, 3, 4	Li1	RANGE: Set to negative of first row of P matrix RATE: Elements of vector $\underline{L} = P \cdot M$ , $M = (m_{11}, m_{12}, m_{13}, m_{14})$ $\begin{cases} L_{11}, L_{21} (\text{ft}^2/\text{sec}) \\ L_{31}, L_{41} (\text{fps})^2 \end{cases}$	$\pm 2^{20}$ $\pm 2^{10}$	$2^3$ $2^{-7}$	$\dot{R}_x, \dot{R}_y, \dot{R}_z$	RD0TX, RD0TY, RD0TZ	Computed range rate vector between the LM and the CSM (fps)	$\pm 2^{13}$	$2^{-4}$
$m_{11}$	*	Partial derivative of computed range rate with respect to $r_x$ ( $\text{sec}^{-1}$ )	$\pm 2^{-9}$	$2^{-26}$	$S_{17}$	S17	= $\begin{cases} 0 & \text{Do not perform initialization} \\ 1 & \text{Initialize radar filter covariance matrix and } t_1 \end{cases}$	$2^{18}$	$2^1$
$m_{12}$	*	Partial derivative of computed range rate with respect to $r_z$ ( $\text{sec}^{-1}$ )	$\pm 2^{-9}$	$2^{-26}$	t	TA1	AGS absolute time (sec)	$\pm 2^{18}$	$2^1$
$m_{13}$	*	Partial derivative of computed range rate with respect to $V_x$ (no units)	$\pm 2^1$	$2^{-16}$	$t_1$	T1	Absolute time of last radar range update (sec)	$\pm 2^{13}$	$2^{-4}$
$m_{14}$	*	Partial derivative of computed range rate with respect to $V_z$ (no units)	$\pm 2^1$	$2^{-16}$	$V_y$	VY	LM y inertial velocity (fps)	$\pm 2^1$	$2^{-16}$
$P_{ij}$		LM state vector error covariance (symmetric) matrix. See description of the elements below			$\underline{Z}_b^*$	A31S, A32S, A33S	Z-body direction cosines saved when S15 entered		
$P_{11}$	P11	$r_x$ error variance ( $\text{ft}^2$ )	$2^{30}$	$2^{13}$	Constants: $J^{17}, J^{18}, J^{29}, K_2^6, K_4^6, K_5^6, K_6^6, K_8^6, K_9^6, K_{10}^6$				
$P_{12}$	P12	$r_x - r_z$ error covariance ( $\text{ft}^2$ )	$\pm 2^{30}$	$2^{13}$					
$P_{13}$	P13	$r_x - V_x$ error covariance ( $\text{ft}^2/\text{sec}$ )	$\pm 2^{20}$	$2^3$					
$P_{14}$	P14	$r_x - V_z$ error covariance ( $\text{ft}^2/\text{sec}$ )	$\pm 2^{20}$	$2^3$					
$P_{21}$	P21	$r_x - r_z$ error covariance ( $\text{ft}^2$ )	$\pm 2^{30}$	$2^{13}$					
$P_{22}$	P22	$r_z$ error variance ( $\text{ft}^2$ )	$2^{30}$	$2^{13}$					
$P_{23}$	P23	$r_z - V_x$ error covariance ( $\text{ft}^2/\text{sec}$ )	$\pm 2^{20}$	$2^3$					
$P_{24}$	P24	$r_z - V_z$ error covariance ( $\text{ft}^2/\text{sec}$ )	$\pm 2^{20}$	$2^3$					
$P_{33}$	P33	$V_x$ error variance ( $\text{fps}^2$ )	$2^{10}$	$2^{-7}$					
$P_{34}$	P34	$V_x - V_z$ error covariance ( $\text{fps}^2$ )	$\pm 2^{10}$	$2^{-7}$					
$P_{44}$	P44	$V_z$ error variance ( $\text{fps}^2$ )	$2^{10}$	$2^{-7}$					
$P_{43}$	P43	$V_x - V_z$ error covariance ( $\text{fps}^2$ )	$\pm 2^{10}$	$2^{-7}$					

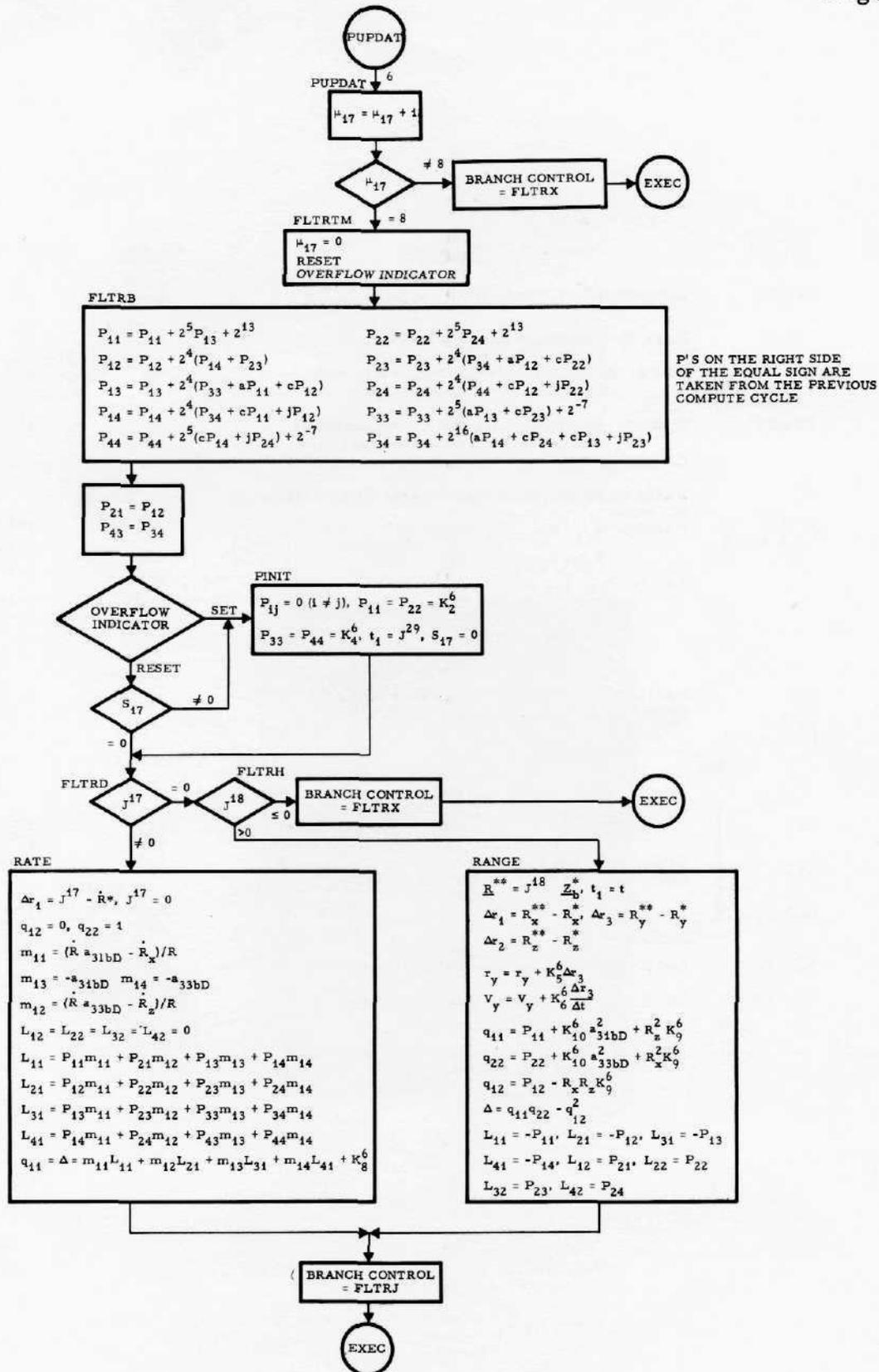


Figure 7.17. Covariance Matrix Update and Processing of Range and Range Rate Inputs

Equation Variable	Program Symbol	Definition	Range	Quantization
$\Delta$	DELTA	Determinant of matrix Q	$\pm 2^{60}$ $\pm 2^{10}$	$2^{43}$ $2^{-7}$
$\Delta r_1$	DELR1	RANGE: difference between X components of input range and computed range (ft) RATE: difference between input range rate and computed range rate (fps)	$\pm 2^{23}$ $\pm 2^{13}$	$2^6$ $2^{-4}$
$\Delta r_2$	DELR2	RANGE: difference between Z components of input range and computed range (ft) RATE: unused	$\pm 2^{23}$	$2^6$
$b_{ij}$ $i = 1, 2$ $i = 1, 2, 3, 4$	Bij $j = 1, 2$ $i = 1, 2, 3, 4$	Range or range rate radar in plane filter weights (no units) RANGE: $b_{11}, b_{12}, b_{21}, b_{22}, b_{31}, b_{32}, b_{41}, b_{42}$ RATE: $b_{11}, b_{21}, b_{31}, b_{41}$ $b_{i2}$ equals zero $i = 1, 2, 3, 4$	$\pm 2^3$ $\pm 2^{-7}$ $\pm 2^{13}$ $\pm 2^3$	$2^{-14}$ $2^{-24}$ $2^{-4}$ $2^{-14}$
$L_{ij}$ $j = 1, 2$ $i = 1, 2, 3, 4$	Lij $j = 1, 2$ $i = 1, 2, 3, 4$	Intermediate results in range or range rate radar updates (see Figure 7.17 for descriptions)		
$P_{ij}$	Pij	LM state vector error covariance (symmetric) matrix (see Figure 7.17 for description)		
$q_{11}$	Q11	Elements of the intermediate computation symmetric matrix Q	$\pm 2^{30}$ $\pm 2^{10}$	$2^{13}$ $2^{-7}$
$q_{12}$	Q12		$\pm 2^{30}$	$2^{13}$
$q_{22}$	Q22		$\pm 2^{30}$ $\pm 2^0$	$2^{13}$ $2^{-17}$
$r_x, r_z$	RX, RZ	LM X and Z inertial position components (ft)	$\pm 2^{23}$	$2^6$
$V_x, V_z$	VX, VZ	LM X and Z inertial velocity components (fps)	$\pm 2^{13}$	$2^{-4}$

Constants: J<sup>18</sup>

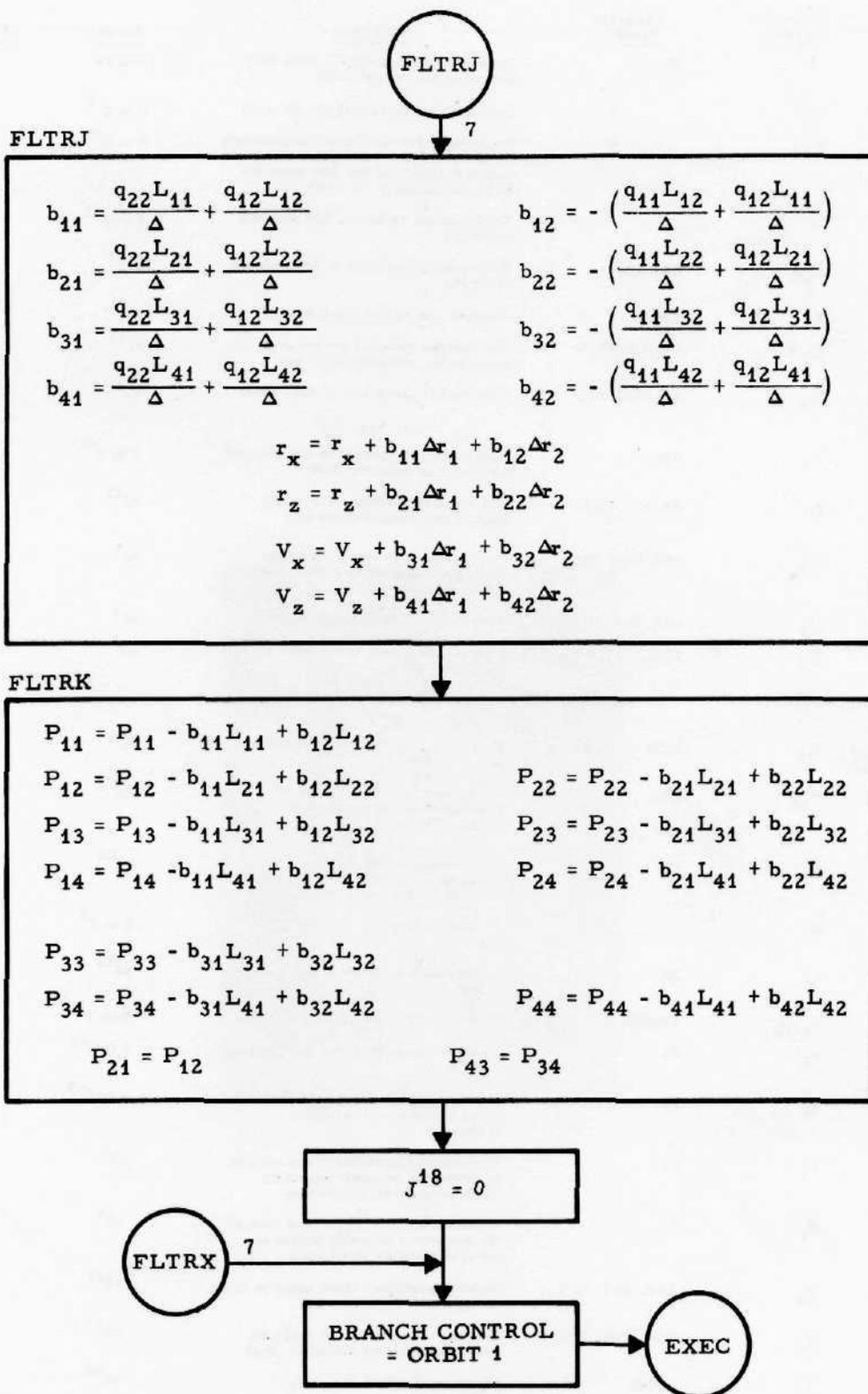


Figure 7.18. LM State Vector and Covariance Matrix Update Following a Radar Input

Equation Variable	Program Symbol	Definition	Range	Quantization
$\xi$	XI	Inplane angle between Z body axis and local horizontal (rad)	0 to $2\pi$	$2^{-14}$
$e_1^2$	*	Eccentricity (squared) of LM orbit	0 to $2^{-6}$	$2^{-23}$
P	*	Semi-latus rectum (semi-parameter) of LM elliptical orbit (ft). It is the radial distance of the LM when the LM true anomaly is $\pm 90^\circ$ .	0 to $2^{23}$	$2^6$
q	*	Pericynthion radius of LM present orbit (ft)	0 to $2^{23}$	$2^6$
$q_{LT}$	QLTELE	Pericynthion altitude of LM present orbit (ft)	$\pm 2^{23}$	$2^6$
$\dot{z}$	HDOT	Present LM radial velocity (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{V}, V$	VX, VY, VZ, V	LM inertial velocity vector and magnitude, respectively (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{Z}_b$	A31, A32, A33	Unit vector along LM Z body axis: $\underline{Z}_b = (a_{31}, a_{32}, a_{33})$	$\pm 2^1$	$2^{-16}$
$V_h$	VH	Horizontal component of the present LM inertial velocity (fps)	0 to $2^{13}$	$2^{-4}$
$\underline{r}, r$	RX, RY, RZ, R	LM inertial position vector and magnitude, respectively (ft)	$\pm 2^{23}$	$2^6$
$\underline{V}_1$	V1X, V1Y, V1Z	Unit vector directed downrange from the LM and parallel to the CSM orbit plane	$\pm 2^1$	$2^{-16}$
$\underline{U}_1$	U1X, U1Y, U1Z	Normalized LM position vector	$\pm 2^1$	$2^{-16}$
$\underline{W}_1$	W1X, W1Y, W1Z	Unit vector given by the equation: $\underline{W}_1 = \underline{U}_1 \times \underline{V}_1$	$\pm 2^1$	$2^{-16}$
$\underline{W}_c$	WCX, WCY, WCZ	Unit vector normal to CSM orbit plane	$\pm 2^1$	$2^{-16}$
$V_{y0}$	VY0	Component of LM velocity in the direction perpendicular to the CSM orbit (fps)	$\pm 2^{13}$	$2^{-4}$
y	Y	Component of LM inertial position perpendicular to the CSM orbit plane (ft)	$\pm 2^{23}$	$2^6$
E	*	Eccentric anomaly of LM (rad)	0 to $2^3$	$2^{-13}$
$q_a$	QA	Apocynthion altitude of LM present orbit	$\pm 2^{23}$	$2^6$
$T_{perg}$	TPERG	Time of LM to perilune	0 to $2^{13}$	$2^{-4}$
$a_L$	AI	Semi-major axis of the LM present orbit	0 to $2^{23}$	$2^6$
$n_i$	NI	Mean orbital angular rate output by orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$
$C_i$	CI	Product of eccentricity and cosine of eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$S_i$	SI	Product of eccentricity and sine of the eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$\underline{E}_0$	R0X, R0Y, R0Z	Inertial position vector input to orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$\underline{V}_0$	V0X, V0Y, V0Z	Inertial velocity vector input to orbit parameter subroutine (fps)	$\pm 2^{13}$	$2^{-4}$
$V_h^2$	VHSQ	$V_h$ squared saved for orbit insertion	$\pm 2^{26}$	$2^9$
$\underline{E}_s$	R5X, R5Y, R5Z	Current LM position and velocity saved for the orbit insertion equations (ft, fps)	$\pm 2^{23}$	$2^6$
$\underline{V}_s$	V5X, V5Y, V5Z			

Constants:  $K_2^2, K_3^2, J^5$

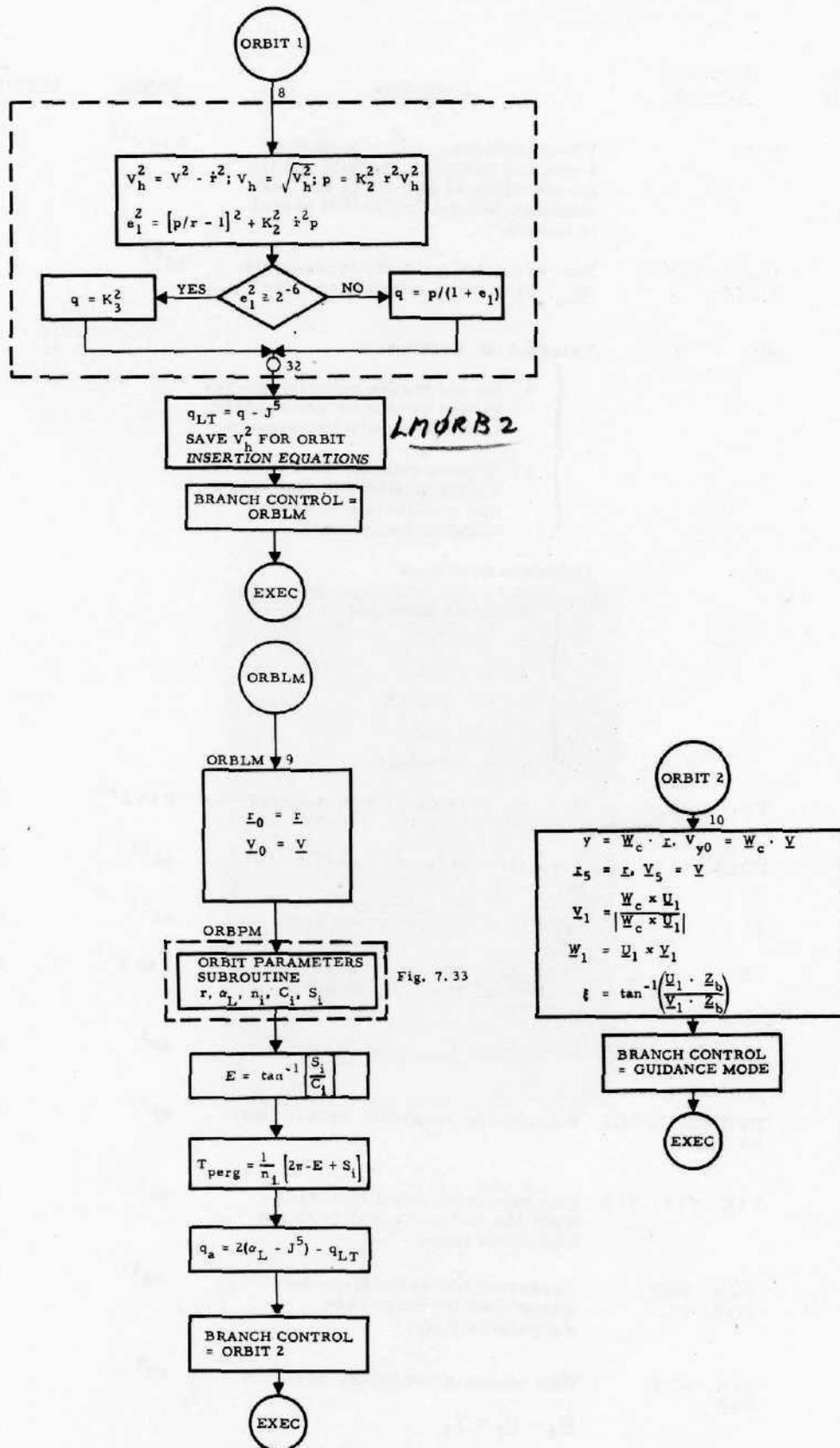


Figure 7.19. LM Orbit Parameters, Branches 8, 9, and 10

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\mu_8$	MU8	Ullage counter. ( $K_9^1$ consecutive 2 sec computer cycles with thrust acceleration at $a_T \geq K_{35}^4 \text{ fps}^2$ are required before Engine/ON signal is issued.)	0 to $2^{17}$	$2^0$
$\Delta \underline{v}_s$	DQSX, DQSY, DQSZ	Sum of sensed velocity increments, $\Delta \underline{v}_s$ , in inertial coordinates (fps)	$\pm 2^{13}$	$2^{-4}$
$S_{07}$	S07	External $\Delta V$ Reference = $\left\{ \begin{array}{l} 0 \text{ Do not freeze velocity-to-be-gained vector or accumulate sensed velocity increments} \\ 1 \text{ Freeze velocity-to-be-gained vector in AGS inertial frame and accumulate sensed velocity increments} \end{array} \right.$		
$S_{10}$	S10	Guidance Selection = $\left\{ \begin{array}{l} 0 \text{ Orbit Insertion} \\ 1 \text{ CSI} \\ 2 \text{ CDH} \\ 3 \text{ TPI Search} \\ 4 \text{ TPI Execute} \\ 5 \text{ External } \Delta V \end{array} \right.$		
$t_{ig}$	TIG	Desired absolute time of next maneuver (CSI, CDH, TPI) (sec)	0 to $2^{18}$	$2^1$
$T_\Delta$	TDEL	Time from present to next maneuver (sec)	$\pm 2^{13}$	$2^1$
$T_i$	TI	Input to ellipse predictor (sec)	$\pm 2^{13}$	$2^{-4}$
$T_r$	TR	Predicted time from present to rendezvous (sec) (meaningful only in TPI)	0 to $2^{13}$	$2^{-4}$
$\underline{U}_1$	U1X, U1Y, U1Z	Normalized LM position vector	$\pm 2^1$	$2^{-16}$
$\Delta \underline{V}$	DVGXX, DVGXY, DVGXZ	Velocity-to-be-gained vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{V}_1$	V1X, V1Y, V1Z	Unit vector directed downrange from the LM and parallel to the CSM orbit plane	$\pm 2^1$	$2^{-16}$
$\underline{V}_G, V_G$	VGX, VGY, VGZ, VG	Vector of LM velocity-to-be-gained and its magnitude, respectively (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{W}_1$	W1X, W1Y, W1Z	Unit vector given by the equation $\underline{W}_1 = \underline{U}_1 \times \underline{V}_1$	$\pm 2^1$	$2^{-16}$
$t$	TA1	Present Time (sec)	$2^{18}$	$2^1$

Constants:  $J^6, J_1^{28}, J_2^{28}, J_3^{28}, K_{11}^2$

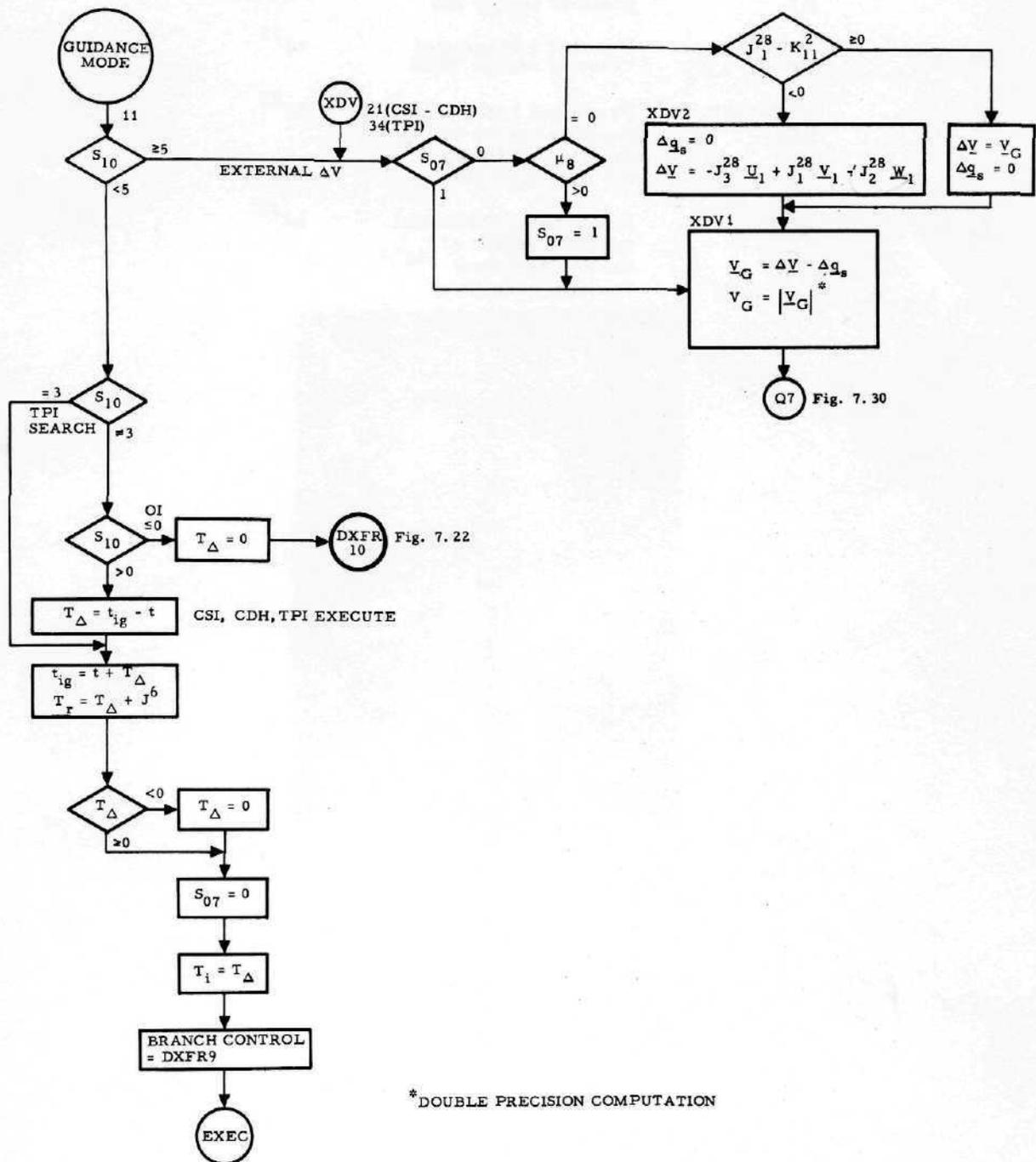


Figure 7.20. Guidance Selection Logic, Branch 11

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$T_i$	TI	Input to ellipse predictor (sec)	$\pm 2^{13}$	$2^1$
$T_{\Delta}$	TDEL	Time from present to next maneuver (sec)	$\pm 2^{13}$	$2^1$
$\underline{r}$	RX, RY, RZ	Present LM inertial position vector (ft)	$\pm 2^{23}$	$2^6$
$\underline{V}$	VX, VY, VZ	Present LM inertial velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{r}_5$	R5X, R5Y, R5Z	Predicted LM inertial position vector at $t_{ig}$ , the desired time for next burn (ft)	$\pm 2^{23}$	$2^6$
$\underline{V}_5$	*	Predicted LM inertial velocity vector at $t_{ig}$ , the desired time for next burn (fps)	$\pm 2^{13}$	$2^{-4}$

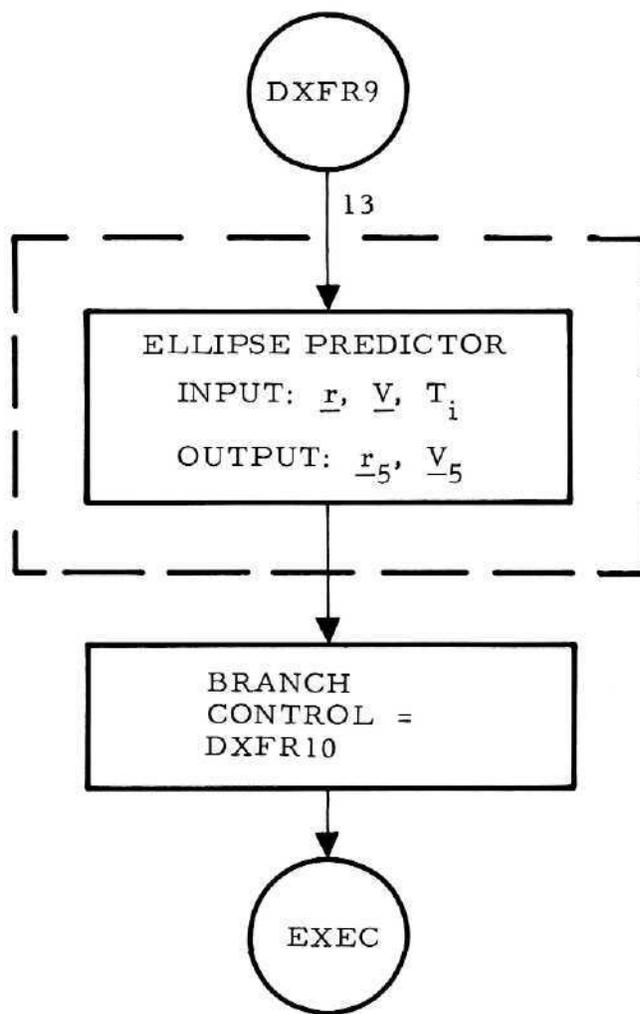


Figure 7.21. LM Prediction

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\underline{r}_5$	R5X, R5Y, R5Z	Predicted LM inertial position vector at $t_{ig}$ , the desired time for the next burn; current LM position for OI (ft)	$\pm 2^{23}$	$2^6$	$V_{ha}$	VHA	Magnitude of $\underline{V}_5$ (fps)	$\pm 2^{13}$	$2^{-4}$
$r_f$	RF	Predicted LM radial distance at $t_{ig}$ , current for OI (ft)	0 to $2^{23}$	$2^6$	$\underline{W}_C$	WCX, WCY, WCZ	Unit vector normal to CSM orbit plane	$2^1$	$2^{-16}$
p	P	Semi-latus rectum of LM orbit (TPI only) (ft)	0 to $2^{23}$	$2^6$	$\alpha_E$	AE	Semimajor axis of CSM orbit computed from CSM ephemeris data in orbit parameter subroutine (ft)	0 to $2^{23}$	$2^6$
$\underline{U}_1$	U1X, U1Y, U1Z	Unit vector in direction of $\underline{r}_5$	$\pm 2^1$	$2^{-16}$	$C_E$	CI	Product of the eccentricity and cosine of the eccentric anomaly computed from CSM ephemeris data in orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$T_i$	TI	Input to ellipse predictor (sec)	$\pm 2^{13}$	$2^{-4}$	$n_E$	NE	Mean orbital angular rate of CSM computed from CSM ephemeris data in orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$
$t_b$	TB	Time elapsed since CSM epoch, $t_E$ (sec)	0 to $2^{13}$	$2^{-4}$	$\underline{r}_0$	R0X, R0Y, R0Z	Position vector which is input to orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$T_{CSM}$	TCSM	Period of CSM orbit (sec)	0 to $2^{13}$	$2^{-4}$	$r_E$	R0	Radial position of CSM computed from CSM ephemeris data in orbit parameter subroutine (ft)	0 to $2^{23}$	$2^6$
$T_\Delta$	TDEL	Time from present to next maneuver (sec)	$\pm 2^{13}$	$2^1$	$S_E$	SI	Product of eccentricity and sine of the eccentric anomaly computed from CSM ephemeris data in orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$\underline{r}_c$	*	Predicted CSM inertial position vector at $t_{ig}$ ; current for OI (ft)	$\pm 2^{23}$	$2^6$	$\underline{V}_0$	VOX, VOY, VOZ	Velocity vector which is input to orbit parameter subroutine (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{V}_c$	*	Predicted CSM inertial velocity vector at $t_{ig}$ ; current for OI (fps)	$\pm 2^{13}$	$2^{-4}$	Constants: $K_{14}^2$				
$\underline{r}_E$	REX, REY, REZ	Ephemeris position vector of CSM (fps)	$\pm 2^{23}$	$2^6$					
$\underline{V}_E$	VEX, VEY, VEZ	Ephemeris velocity vector of CSM (ft)	$\pm 2^{13}$	$2^{-4}$					
$\dot{r}_A$	RADOT	Predicted LM altitude rate at $t_{ig}$ for CSI, CDH, and TPI; current altitude rate for OI (fps)	$\pm 2^{13}$	$2^{-4}$					
$\underline{V}_5$	V5X, V5Y, V5Z	Predicted LM inertial velocity vector at $t_{ig}$ for CSI, CDH, and TPI; current velocity vector for OI (fps)	$\pm 2^{13}$	$2^{-4}$					
$V_{py}$	VPY	Predicted LM out-of-CSM-plane velocity at $t_{ig}$ for CSI, CDH, and TPI; current LM out-of-CSM-plane velocity for OI (fps)	$\pm 2^{13}$	$2^{-4}$					

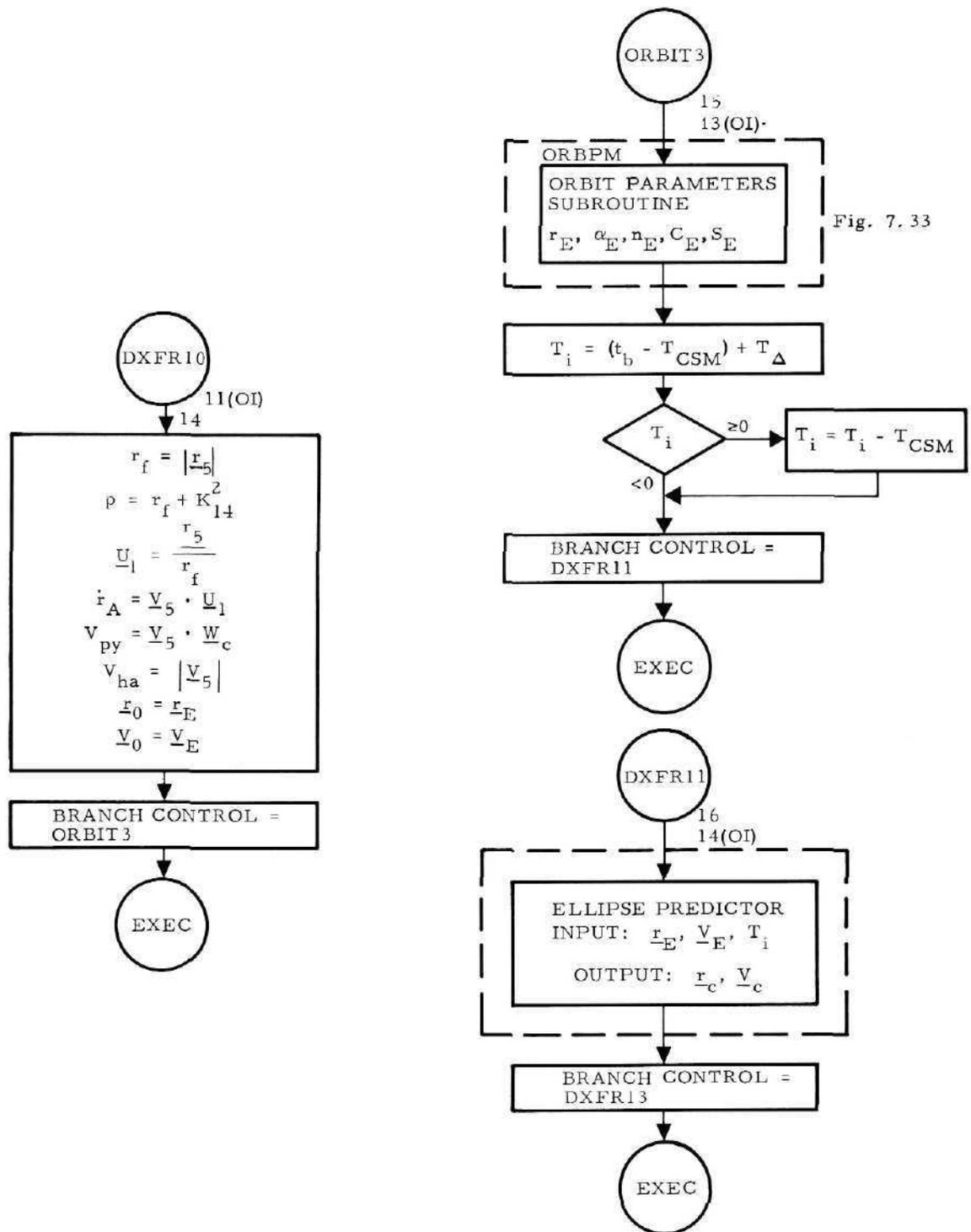


Figure 7.22. CSM Prediction

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\theta_{LOS}$	TLOS	Angle that the projection of the LM-CSM line of sight (LOS) on the plane determined by the LM local vertical and a local horizontal line parallel to the CSM orbit makes with the LM local horizontal at the desired TPI time (rad)	0 to $2^3$	$2^{-14}$	$\underline{V}'_1$	*	Unit vector normal to $\underline{U}_1$ and parallel to the CSM orbit plane at $t_{ig}$ in CSI, CDH, and TPI; at present in OI	$\pm 2^1$	$2^{-16}$
$\underline{r}_5$	R5X, R5Y, R5Z	Predicted LM inertial position vector at $t_{ig}$ , in CSI, CDH, or TPI; at present in OI(ft)	$\pm 2^{23}$	$2^6$	$\underline{W}_c$	WCX, WCY, WCZ	Unit vector normal to the CSM orbit plane	$\pm 2^1$	$2^{-16}$
$\underline{r}_c$	*	Predicted CSM inertial position vector at $t_{ig}$ , in CSI, CDH, or TPI; at present in OI(ft)	$\pm 2^{23}$	$2^6$	$S_{10}$	S10	Guidance Selection $\left\{ \begin{array}{l} 0 \text{ Orbit Insertion} \\ 1 \text{ CSI} \\ 2 \text{ CDH} \\ 3 \text{ TPI Search} \\ 4 \text{ TPI Execute} \\ 5 \text{ External } \Delta V \end{array} \right.$		
$\underline{r}_E$	REX, REY, REZ	CSM inertial position vector (ft)	$\pm 2^{23}$	$2^6$	$t_b$	TB	Elapsed time since CSM epoch, $t_E$ (sec)	$+2^{13}$	$2^{-4}$
$\underline{r}_T$	*	First computed as predicted CSM inertial position vector at rendezvous time, then computed as predicted CSM inertial position vector at rendezvous time retarded by $J^4$ in TPI. Predicted CSM inertial position vector as it crosses vector in $\underline{r}_5$ direction in CSI and CDH(ft)	$\pm 2^{23}$	$2^6$	$\theta_f$	THETA F	Central angle between the CSM and LM at $t_{ig}$ in CSI or CDH; central angle between the LM and CSM at present in OI(rad)	$-\pi$ to $\pi$	$2^{-14}$
$\underline{R}$	*	Predicted relative position vector of CSM with respect to the LM at $t_{ig}$ in CSI, CDH, and TPI; at present in OI(ft)	$\pm 2^{23}$	$2^6$	$r_f$	RF	Predicted LM altitude at $t_{ig}$ for CSI, CDH, and TPI, at present for OI(fps)	$\pm 2^{13}$	$2^{-4}$
$T_{CSM}$	TCSM	Period of CSM orbit (sec)	0 to $2^{13}$	$2^{-4}$	$T_\delta$	*	Time required for CSM to move through angle $\theta_f$ (sec)	$-2^{13}$	$2^{-4}$
$R_x$	*	Component of $\underline{R}$ along LM local vertical ( $\underline{U}_1$ ) axis (ft)	$\pm 2^{23}$	$2^6$	$n_E$	NI	Mean orbital angular rate of CSM computed from CSM ephemeris data in the orbit parameters subroutine (rad/sec)	$2^{-9}$	$2^{-26}$
$R_z$	*	Component of $\underline{R}$ along LM local horizontal ( $\underline{V}_1$ ) axis (ft)	$\pm 2^{23}$	$2^6$	Constants: $J^3$				
T	T	Predicted time from TPI to rendezvous (sec)	0 to $2^{13}$	$2^{-4}$					
$T_\Delta$	TDEL	Time from present to next maneuver in CSI, CDH, and TPI (sec)	$\pm 2^{13}$	$2^1$					
$T_i$	TI	Input to ellipse predictor (sec):  First computed as time from CSM epoch to rendezvous, then computed as time from CSM epoch to rendezvous retarded by $J^4$ in TPI. Time from CSM epoch to CSM crossing vector in $\underline{r}_5$ direction in CSI and CDH (sec)	$\pm 2^{13}$	$2^{-4}$					
$T_r$	TR	Predicted time from present to rendezvous in TPI (sec)	0 to $2^{13}$	$2^{-4}$					
$\underline{U}_1$	U1X, U1Y, U1Z	Unit vector in direction of $\underline{r}_5$	$\pm 2^1$	$2^{-16}$					
$\underline{V}_E$	VEY, VEY, VEZ	CSM inertial velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$					
$\underline{V}_T$	*	First computed as predicted CSM inertial velocity vector at rendezvous, then computed as predicted CSM inertial velocity vector at rendezvous time retarded by $J^4$ in TPI. Predicted CSM inertial velocity vector as it crosses vector in $\underline{r}_5$ direction in CSI and CDH (fps)	$\pm 2^{13}$	$2^{-4}$					

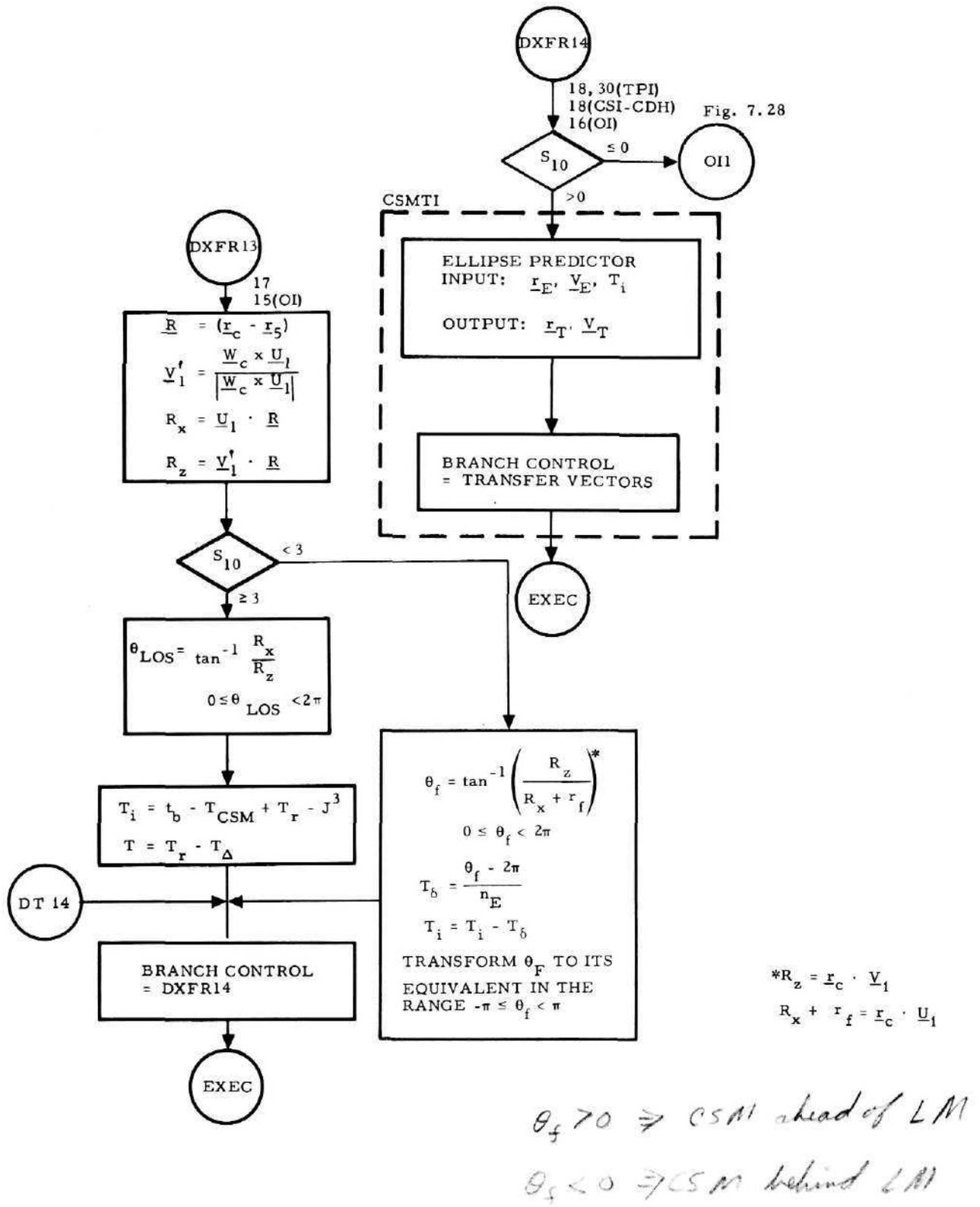


Figure 7.23. Central Angle/Line of Sight Angle at  $t_{ig}$  and CSM at CDH/Rendezvous

Equation Variable	Program Symbol	Definition	Range	Quantization
$\mu_3$	MU3	p-iteration control counter	0 to 8	$2^0$
$c_1, c_2$	C1, C2	<p>a) First computed as cosine and sine, respectively, of the central angle between LM now and CSM at rendezvous time in TPI.</p> <p>b) Then computed as cosine and sine, respectively, of the central angle between LM now and CSM at rendezvous time retarded by <math>J^4</math> in TPI</p> <p>The sign of <math>c_2</math> is positive if <math>y_1</math> is parallel to <math>y_c</math>, negative otherwise. These parameters are not used in CSI or CDH.</p>	$\pm 2^1$	$2^{-16}$
$r_T, r_T$	* , RT	First computed as predicted CSM inertial position vector and magnitude, respectively, at rendezvous time; then computed as predicted CSM inertial position vector and magnitude, respectively, at rendezvous time retarded by $J^4$ in TPI; in CSI and CDH it is the predicted CSM position vector and magnitude as it crosses vector in the $r_5$ direction (ft)	$\pm 2^{23}$	$2^6$
$u_1$	U1X, U1Y, U1Z	Unit vector in direction of LM inertial position vector at $t_{ig}$	$\pm 2^1$	$2^{-16}$
$u_2$	U2X, U2Y, U2Z	First computed as unit vector pointing toward rendezvous point, then computed as unit vector pointing toward rendezvous point retarded by $J^4$ in TPI; in CSI and CDH the unit vector pointing toward the CSM at $t_{ig}$	$\pm 2^1$	$2^{-16}$
$v_1$	V1X, V1Y, V1Z	Horizontal unit vector in transfer orbit plane	$\pm 2^1$	$2^{-16}$
$v_2$	V2X, V2Y, V2Z	<p>a) First computed as unit vector, horizontal at rendezvous point and in transfer orbit plane</p> <p>b) Then computed as unit vector, horizontal at rendezvous point retarded by <math>J^4</math> and in transfer orbit plane</p>	$\pm 2^1$	$2^{-16}$
$w_1$	W1X, W1Y, W1Z	Unit vector perpendicular to transfer orbit plane	$\pm 2^1$	$2^{-16}$
$w_c$	WCX, WCY, W CZ	Unit vector normal to CSM orbit plane	$\pm 2^1$	$2^{-16}$
$y_1$	W1Y	Component of $w_1$ along y inertial axis	$\pm 2^1$	$2^{-16}$
$y_c$	WCY	Component of $w_c$ along y inertial axis	$\pm 2^1$	$2^{-16}$
$\delta_{10}$	DEL10	$= \begin{cases} 0 & \text{Transfer orbit computations made for CSM at rendezvous} \\ 1 & \text{Transfer orbit computations made for CSM at rendezvous retarded by } J^4 \end{cases}$		
$s_{10}$	$s_{10}$	<p>Guidance Selector</p> $= \begin{cases} 0 & \text{Orbit Insertion} \\ 1 & \text{CSI} \\ 2 & \text{CDH} \\ 3 & \text{TPI Search} \\ 4 & \text{TPI Execute} \\ 5 & \text{External } \Delta V \end{cases}$		

Constants:  $K_{11}^2, K_4^3, J^4, J_1^{28}$

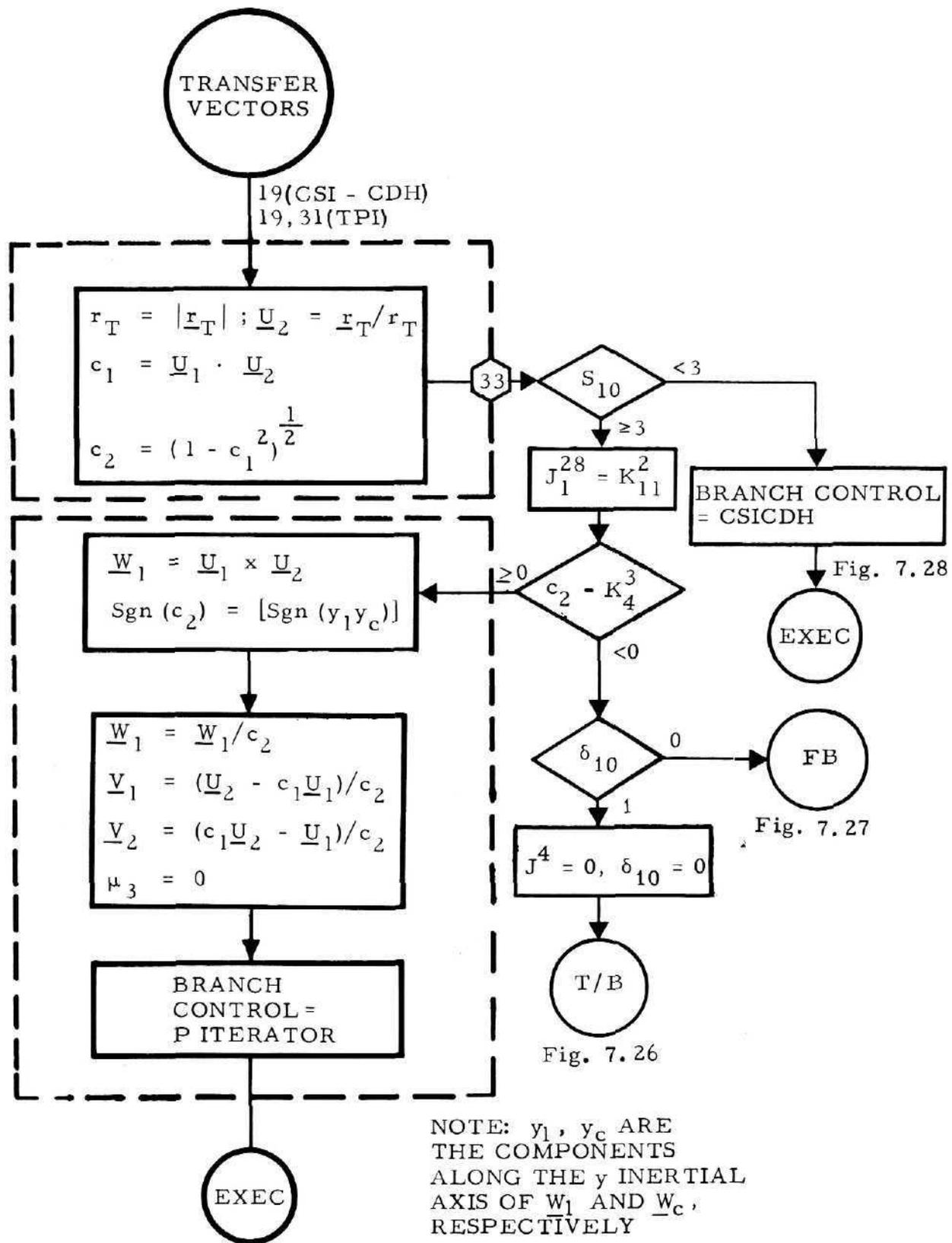


Figure 7.24. Prediction of Transfer Orbit Vectors and Central Angle Sine and Cosine

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\alpha$	ALPHA	Semi-major axis of transfer orbit (ft)	0 to $2^{23}$	$2^6$
$\theta_T$	*	Numerically computed partial derivative of p with respect to $T_p$ (fps)	$\pm 2^{14}$	$2^{-3}$
$\mu_3$	MU3	p iteration control counter	0 to 8	$2^0$
$C_{\Delta E}$	*	Term proportional to cosine of $\Delta E$ ( $C_{\Delta E}$ and $S_{\Delta E}$ have same positive constant of proportionality)	$\pm 2^1$	$2^{-16}$
$c_1, c_2$	C1, C2	Cosine and sine of the central angle between the LM now and CSM at rendezvous time	$\pm 2^1$	$2^{-16}$
$e^2$	ESQ	Square of eccentricity of transfer orbit	0 to $2^{-2}$	$2^{-19}$
$\Delta E$	*	Difference between initial and final eccentric anomalies of transfer orbit (rad)	0 to $2^3$	$2^{-14}$
n	*	Mean orbital angular rate of LM in transfer orbit (rad/sec)	0 to $2^{-9}$	$2^{-26}$
p	P	Semi-latus rectum (semi-parameter) of LM transfer orbit (ft)	0 to $2^{23}$	$2^6$
$p'$	*	Previous value of p in the iteration (ft)	0 to $2^{23}$	$2^6$
$\Delta p$	*	Change in p for next iteration (ft)	$\pm 2^{23}$	$2^6$
$r_f$	RF	Predicted radial distance of LM at TPI (ft)	0 to $2^{23}$	$2^6$
$r_T$	RT	Predicted CSM radial distance at rendezvous time (ft)	0 to $2^{23}$	$2^6$
$S_{\Delta E}$	*	Term proportional to the sine of $\Delta E$	$\pm 2^1$	$2^{-16}$
T	T	Desired transfer time (sec)	0 to $2^{13}$	$2^{-4}$
$T_p$	TP	Time from TPI to rendezvous, corresponding to the current p in the iteration (sec)	0 to $2^{13}$	$2^{-4}$
$T_p'$	*	$T_p$ corresponding to p of the previous iteration (sec)	0 to $2^{13}$	$2^{-4}$
$x_1$	*	$x_1 = e \cos \theta_1$ where $\theta_1$ is the initial true anomaly of the transfer orbit at TPI time	$\pm 2^0$	$2^{-17}$
$x_2$	*	$x_2 = e \cos \theta_2$ , where $\theta_2$ is the final true anomaly of the transfer orbit at rendezvous	$\pm 2^0$	$2^{-17}$
$x_3$	X3	$x_3 = e \sin \theta_1$	$\pm 2^0$	$2^{-17}$
$x_4$	X4	$x_4 = e \sin \theta_2$	$\pm 2^0$	$2^{-17}$
$x_8$	*	$x_8 = e \sin E_1$ where $E_1$ is the initial value of the eccentric anomaly of the transfer orbit	$\pm 2^3$	$2^{-14}$
$x_9$	*	$x_9 = e \sin E_2$ where $E_2$ is the final eccentric anomaly of the transfer orbit at rendezvous	$\pm 2^3$	$2^{-14}$
$\delta_{10}$	DEL10	$= \begin{cases} 0 & \text{Transfer orbit computations for CSM at rendezvous} \\ 1 & \text{Transfer orbit computations for CSM at rendezvous retarded by } J^4 \end{cases}$		

Constants:  $K_{12}^2, K_{14}^2, K_{17}^2, K_{18}^2, K_{19}^2, K_{20}^2$

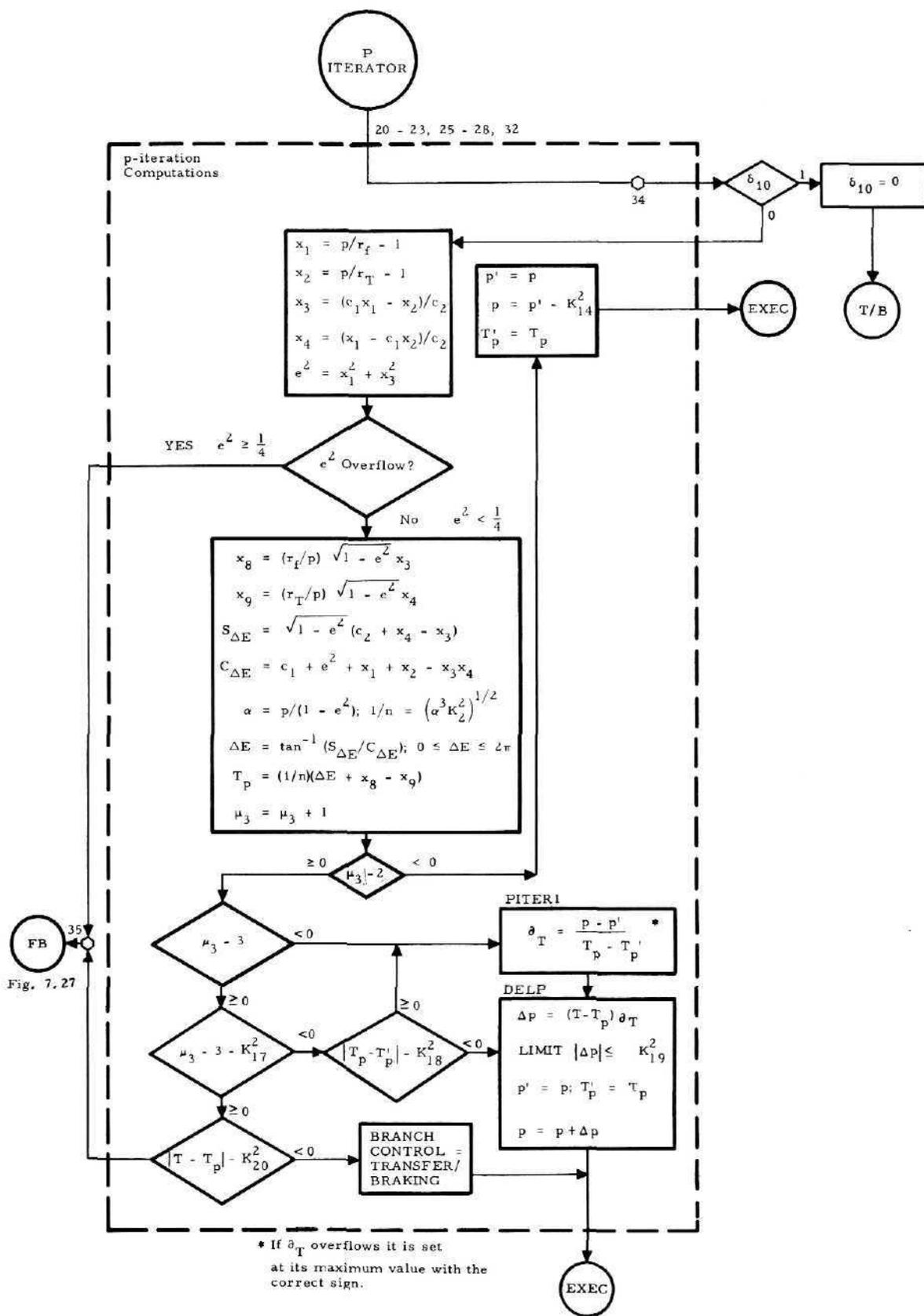


Figure 7.25. p-Iteration Calculations

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\alpha$	ALPHA	Semi-major axis of transfer orbit (ft)	0 to $2^{23}$	$2^6$	$\underline{V}_F$	VFX, VFY, VFZ	Predicted braking velocity vector	$\pm 2^{13}$	$2^{-4}$
$e$	*	Eccentricity of transfer orbit	0 to $2^{-1}$	$2^{-17}$	$\underline{V}_G, V_G$	VGX, VGY, VGZ, VG	Increment in velocity vector and magnitude, respectively, required by the LM for transfer orbit injection (fps)	$\pm 2^{13}$	$2^{-4}$
$e^2$	ESQ	Square of eccentricity of transfer orbit	0 to $2^{-2}$	$2^{-19}$					
$p$	P	Semi-latus rectum (semi-parameter) of LM transfer orbit (ft)	0 to $2^{23}$	$2^6$	$\underline{V}_T$	*	Predicted CSM inertial velocity vector at rendezvous (fps)	0 to $2^{13}$	$2^{-4}$
$q_1$	Q1	Pericynthion radius of LM transfer orbit (ft) measured from center of moon	0 to $2^{23}$	$2^6$	$X_3$	X3	$x_3 = e \sin \theta_1$ where $\theta_1$ is the initial true anomaly of the transfer orbit at TPI time	$\pm 2^0$	$2^{-17}$
$r_f$	RF	Predicted radial distance of LM at TPI (ft)	0 to $2^{23}$	$2^6$	$X_4$	X4	$x_4 = e \sin \theta_2$ where $\theta_2$ is the final true anomaly of the transfer orbit at rendezvous	$\pm 2^0$	$2^{-17}$
$r_T$	RT	Predicted CSM radial distance at rendezvous time (ft)	0 to $2^{23}$	$2^6$	$\delta_{10}$	DEL10	$= \begin{cases} 1 & \text{Transfer orbit computations made for CSM at rendezvous retarded by } J^4 \\ 0 & \text{Transfer orbit computations made for CSM at rendezvous} \end{cases}$		
$\dot{r}_f$	RFDOT	LM radial velocity after first impulse (of the two impulses) for transfer orbit (fps)	$\pm 2^{13}$	$2^{-4}$					
$\dot{\underline{r}}_F$	*	LM velocity vector just prior to the second impulse in the transfer orbit (fps)	$\pm 2^{13}$	$2^{-4}$	$T_i$	TI	Time from CSM epoch to rendezvous retarded by $J^4$ (sec)	$\pm 2^{13}$	$2^{-4}$
$\dot{\underline{r}}_I$	*	LM velocity vector after first impulse (of the two impulses) for the transfer orbit (fps)	$\pm 2^{13}$	$2^{-4}$					
$\underline{U}_1$	U1X, U1Y, U1Z	Unit vector in the direction of the LM inertial position vector at TPI time	$\pm 2^1$	$2^{-16}$					
$\underline{U}_2$	U2X, U2Y, U2Z	Unit vector in direction of rendezvous point retarded by $J^4$	$\pm 2^1$	$2^{-16}$					
$\underline{V}_1$	V1X, V1Y, V1Z	Horizontal unit vector in transfer plane	$\pm 2^1$	$2^{-16}$					
$\underline{V}_2$	V2X, V2Y, V2Z	Unit vector, horizontal at rendezvous point retarded by $J^4$ and in transfer orbit plane	$\pm 2^1$	$2^{-16}$					
$\underline{V}_5$	V5X, V5Y, V5Z	Predicted LM inertial velocity vector at $t_{ig}$ , the desired time for the TPI burn (fps)	$\pm 2^{13}$	$2^{-4}$					

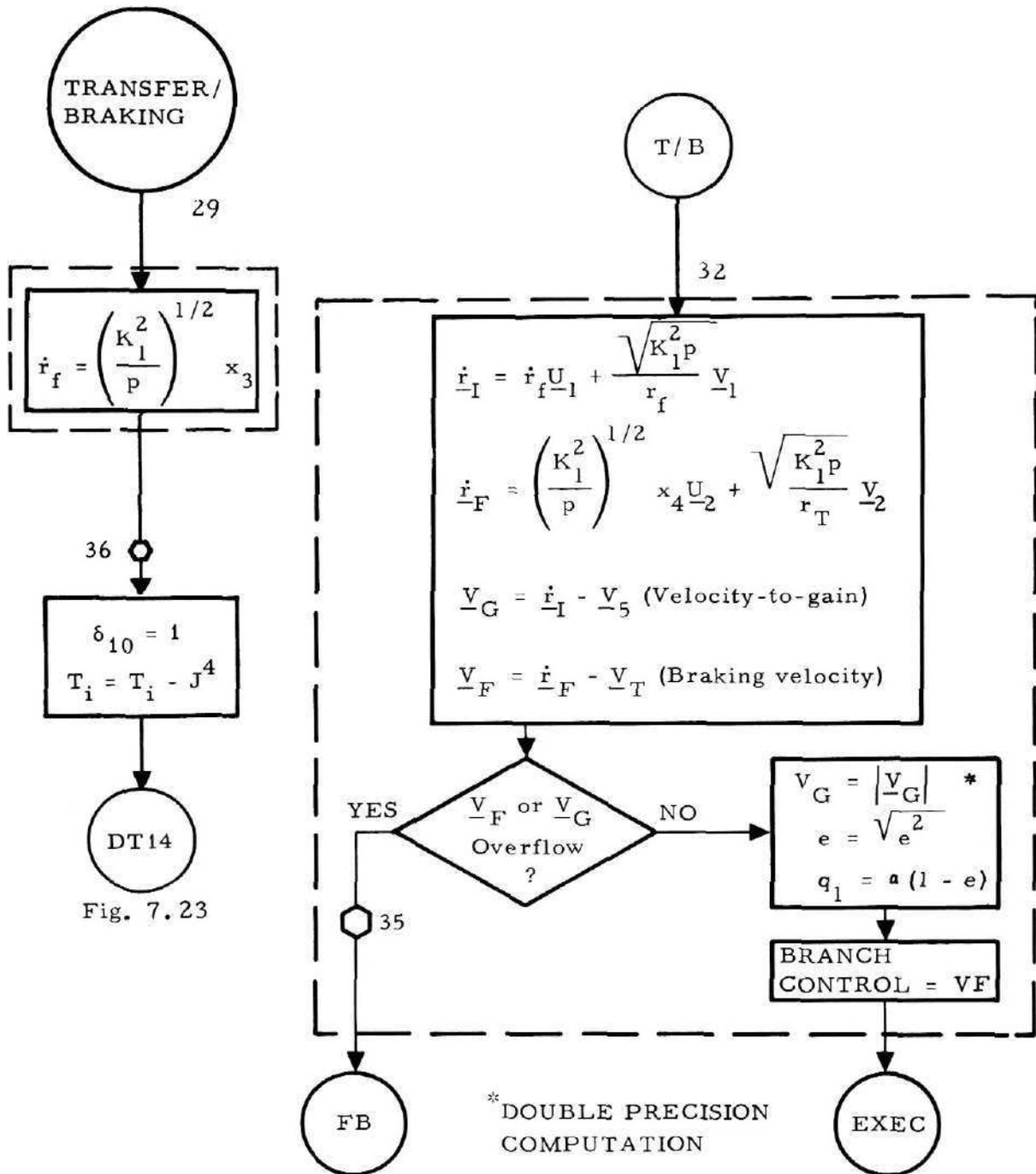


Figure 7.26. Transfer and Braking Impulse Computations

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Description</u>	<u>Range</u>	<u>Quantization</u>
$\underline{V}_F, V_F$	VFX, VFY, VFZ *	Predicted braking velocity vector and magnitude	$\pm 2^{13}$	$2^{-4}$
$V_T$	VT	Total velocity required to rendezvous	0 to $2^{13}$	$2^{-4}$
$V_G$	VG	Magnitude of velocity vector required by the LM for transfer orbit injection (fps)	0 to $2^{13}$	$2^{-4}$
$q_1$	Q1	LM transfer orbit pericyynthion radius, measured from center of the moon	0 to $2^{23}$	$2^6$
$q_{1d}$	Q1DEDA	LM transfer orbit pericynthion altitude	0 to $2^{23}$	$2^6$
Constants: $K_{11}^2, J^5$				

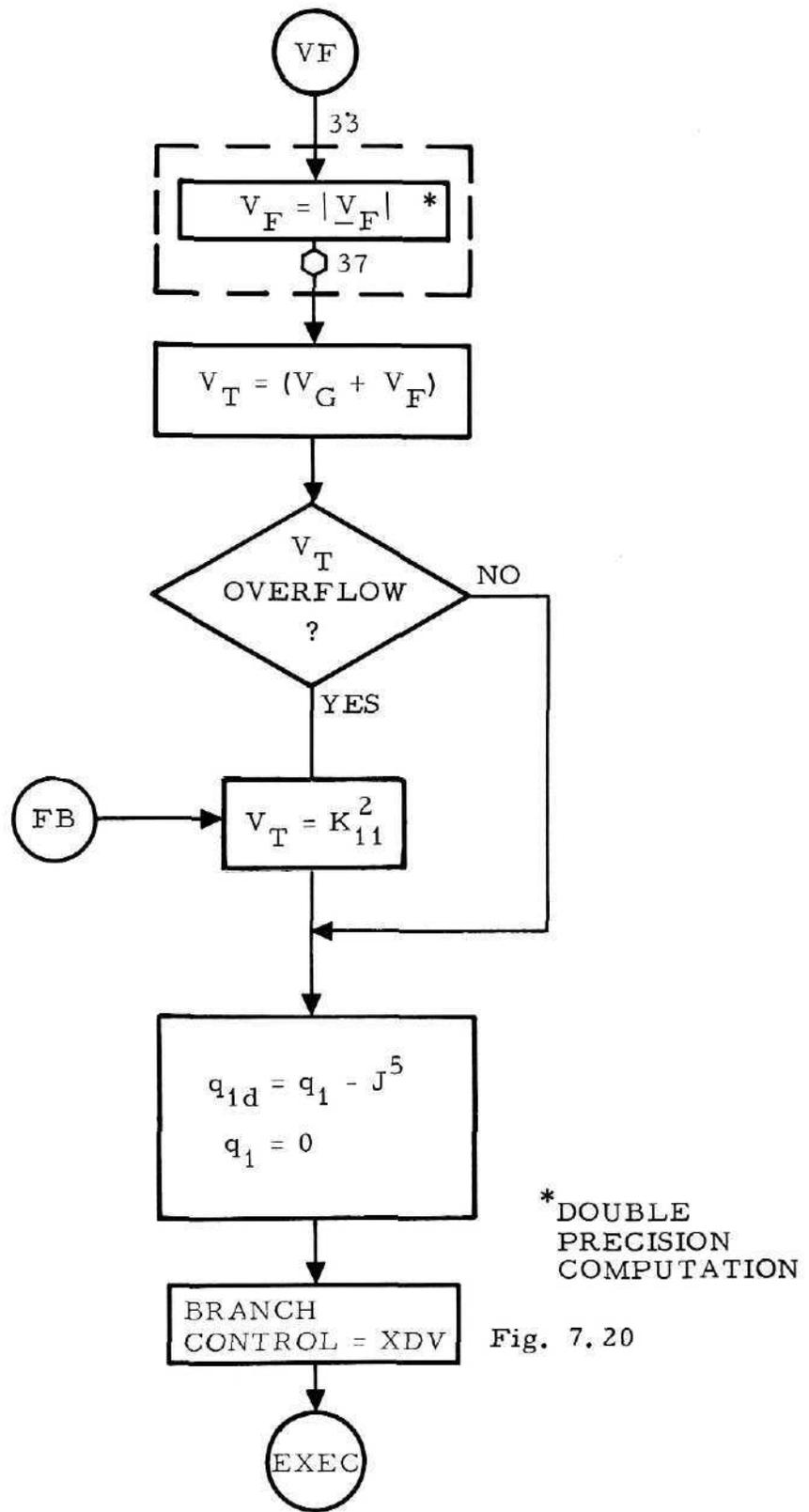


Figure 7.27. Total Velocity for TPI

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\alpha_E$	AE	Semimajor axis of the CSM orbit as computed in orbit parameter subroutine (ft)	0 to $2^{23}$	$2^6$	$S_{16}$	S16	CDH selection switch for CSI equations = $\begin{cases} 1 & \text{Compute CDH at 1/2 LM orbital period after CSI} \\ 3 & \text{Compute CDH at 3/2 LM orbital periods after CSI} \end{cases}$		
$\alpha_L$	AL	Desired semimajor axis of LM orbit after burn	0 to $2^{23}$	$2^6$	$T_{A0}$	TA0	Time from nominal CSI burn to CDH burn (sec)	0 to $2^{13}$	$2^{-4}$
$\theta_f$	THETAF	Limited semimajor axis of LM orbit at completion of OI (ft)			$t_{ig}$	TIG	Absolute time of next (CSI or CDH) maneuver (sec)	$2^{18}$	$2^1$
		Central angle between the CSM and LM at $t_{ig}$ in CSI; at present in OI (rad)	$-\pi$ to $\pi$	$2^{-14}$	$\underline{U}_2$	U2X, U2Y, U2Z	Unit radial vector in direction of maneuver point	$\pm 2^1$	$2^{-16}$
$\Delta r$	DELH	If a line is drawn from the center of the moon to the LM position at the predicted time of the CDH burn, $\Delta r$ is the difference of the radial distance of the CSM at the time it crosses that line and the LM radial distance measured along that line (ft)	$\pm 2^{23}$	$2^6$	$V_h^2$	VHSQ	LM present horizontal velocity squared, saved in the LM orbit parameters branch (fps) <sup>2</sup> (see Figure 7.19)	$2^{26}$	$2^9$
$\delta_r$	DELRP	If a line is drawn from the center of the moon to the LM position at time $t_{ig}$ , $\delta_r$ is the difference of the radial distance of the CSM at the time it crosses that line and the LM radial distance measured along that line (ft)	$\pm 2^{23}$	$2^6$	$V_{ha}$	VHA	Horizontal component of the LM velocity vector parallel to the CSM orbit in OI; predicted horizontal component of the LM velocity vector parallel to the CSM orbit at $t_{ig}$ in CSI and CDH (fps)	$2^{13}$	$2^{-4}$
A	*	CSI temporary storage quantity (rad)	$2^6$	$2^{-11}$	$V_{hf}$	*	Required horizontal component of LM velocity parallel to CSM plane for the current guidance mode (fps)	$2^{13}$	$2^{-4}$
B	*	Coefficients in the quadratic equation in $\Delta r/\alpha_E$	$\pm 2^7$	$2^{-10}$	$V_{p0}$	VP0	Predicted velocity-to-be-gained at CDH; computed in the CSI mode only (fps)	$\pm 2^{13}$	$2^{-4}$
C	*				$\underline{V}_T$	*	Predicted CSM inertial velocity as the CSM crosses the vector ( $\underline{r}_5$ ) in the direction of the LM at $t_{ig}$ for CSI and CDH (fps)	$\pm 2^{13}$	$2^{-4}$
$n_E$	NI	Mean CSM orbital rate output by orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$	$V_{y0}$	VY0	Component of LM velocity in the direction perpendicular to the CSM orbit (fps)	$\pm 2^{13}$	$2^{-4}$
$\dot{r}_A$	RADOT	Predicted LM altitude rate at $t_{ig}$ for CSI and CDH; current altitude rate for OI (fps)	$\pm 2^{13}$	$2^{-4}$	Constants: $K_1^2, K_4^4, K_5^4, K_6^4, K_{10}^4, J^1, J^2, J^7, J^8, J^9, J^{23}, J_1^{28}, J_3^{28}$				
$r_b$	RB	Predicted LM burnout radius saved from the previous 2-sec GDLAW branch (ft) (see Figure 7.29)	$2^{23}$	$2^6$					
$r_f$	RF	Predicted radial distance of LM at $t_{ig}$ in CSI and CDH; at burnout in OI (ft)	$2^{23}$	$2^6$					
$\dot{r}_f$	RFDOT	Predicted radial rate of CSM as it crosses the vector ( $\underline{r}_5$ ) in the direction of the LM at $t_{ig}$ for CDH; predicted final radial rate of LM in OI; predicted radial rate of LM at $t_{ig}$ for CSI (fps)	$\pm 2^{13}$	$2^{-4}$					
$r_T$	RT	Predicted radial distance of CSM as it crosses the vector ( $\underline{r}_5$ ) in the direction of the LM at $t_{ig}$ (ft)	$2^{23}$	$2^6$					
$S_{10}$	S10	Guidance Selection = $\begin{cases} 0 & \text{Orbit Insertion} \\ 1 & \text{CSI} \\ 2 & \text{CDH} \\ 3 & \text{TPI Search} \\ 4 & \text{TPI Execute} \\ 5 & \text{External } \Delta V \end{cases}$							

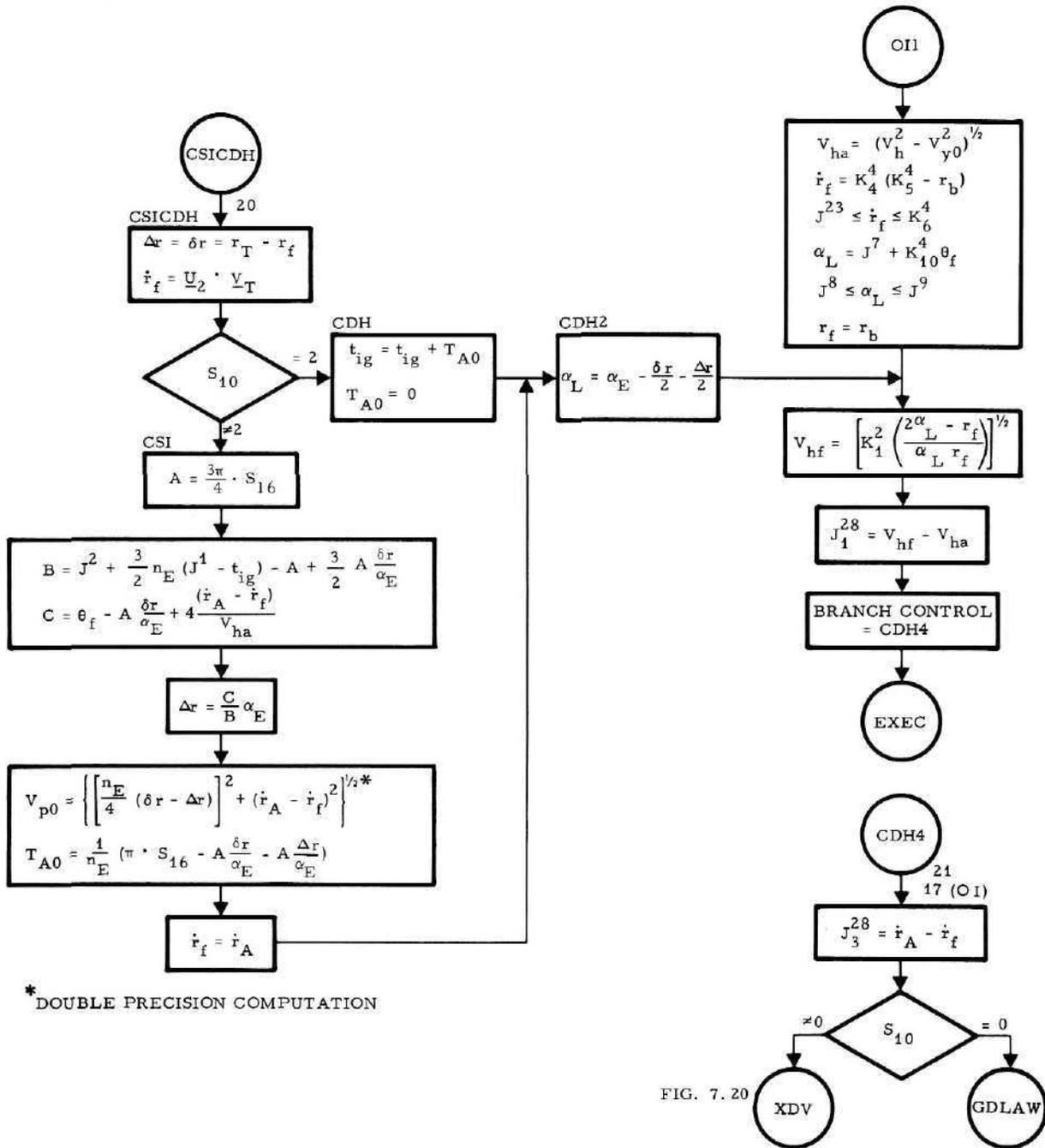


FIG. 7.20

Figure 7.28. Computation of  $V_G$  Components for CSI, CDH, and OI

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$a_T$	AT	Computed LM thrust acceleration (fps <sup>2</sup> )	0 to 2 <sup>7</sup>	2 <sup>-5</sup>
$r$	R	Radial distance of LM from lunar center (ft)	0 to 2 <sup>23</sup>	2 <sup>6</sup>
$r_f$	RF	Predicted burnout radius (ft)	0 to 2 <sup>23</sup>	2 <sup>6</sup>
$r_b$	RB	= $r_f$ for use in next 2-sec cycle OI equations (fps)	0 to 2 <sup>23</sup>	2 <sup>6</sup>
$\delta_2$	DEL2	= $\begin{cases} 1 & \text{Descent Section Staged} \\ 0 & \text{Descent Section Not Staged} \end{cases}$		
$\dot{r}_f$	RFDOT	Predicted LM radial velocity after maneuver (fps)	$\pm 2^{13}$	2 <sup>-4</sup>
$\ddot{r}_d$	RD2DOT	Desired radial acceleration (fps <sup>2</sup> )	$\pm 2^7$	2 <sup>-10</sup>
$\ddot{\dot{r}}_d$	RD3DOT	Desired derivative of radial acceleration (fps <sup>3</sup> )	2 <sup>-2</sup>	2 <sup>-19</sup>
$T_B$	TBO	Time to LM engine burnout (sec)	0 to 2 <sup>9</sup>	2 <sup>-8</sup>
$V_h$	VH	Horizontal component of the present LM inertial velocity (fps)	0 to 2 <sup>13</sup>	2 <sup>-4</sup>
$\ddot{\ddot{R}}_{DTL}$	RD3DTL	Lower limit of $\ddot{\dot{r}}_d$ , computed as a function of vehicle configuration (fps <sup>3</sup> )	$\pm 2^{-2}$	2 <sup>-19</sup>
$V_G$	VG	Magnitude of the velocity-to-be-gained by the LM (fps)	2 <sup>13</sup>	2 <sup>-4</sup>
$V_{y0}$	VY0	Component of the LM inertial velocity perpendicular to the CSM orbit plane (fps)	$\pm 2^{13}$	2 <sup>-4</sup>
$y$	Y	Component of the LM inertial position vector perpendicular to the CSM orbit plane (ft)	$\pm 2^{23}$	2 <sup>6</sup>
$y_f$	*	Component of the final LM inertial position vector at burnout perpendicular to the transfer orbit plane (ft)	$\pm 2^{23}$	2 <sup>6</sup>
$\ddot{y}_d$	YD2DOT	Desired out-of-plane acceleration (fps <sup>2</sup> )	$\pm 2^7$	2 <sup>-10</sup>
$\ddot{\dot{y}}_d$	YD3DOT	Desired derivative of out-of-plane acceleration (fps <sup>3</sup> )	$\pm 2^{-2}$	2 <sup>-19</sup>
$\dot{r}_A$	RADOT	LM altitude rate	$\pm 2^{13}$	2 <sup>-4</sup>

Constants:  $K_1^2, K_2^4, K_3^4, K_{12}^4, K_{14}^5, K_{16}^5, K_{17}^5, K_{18}^5, K_{20}^5, J^5, J^{16}, J_1^{28}, J_3^{28}$

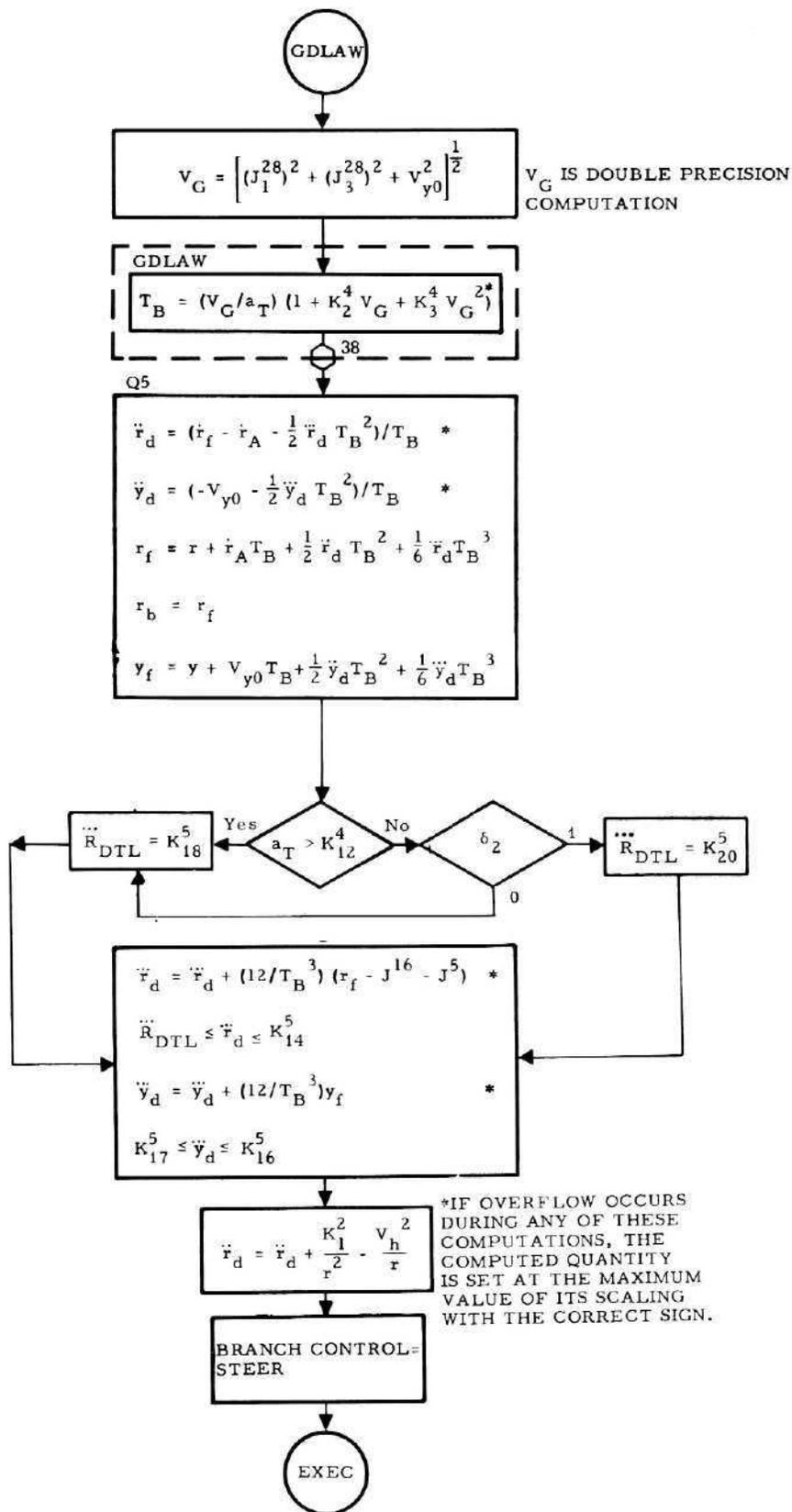


Figure 7.29. Guidance Law

Equation Variable	Program Symbol	Definition	Range	Quantization
$\mu_8$	MU8	Ullage counter. ( $K_9^1$ consecutive 2 second compute cycles with thrust acceleration $a_T \geq K_{35}^4$ fps <sup>2</sup> are required before Engine ON signal is issued.)	0 to $2^{17}$	$2^0$
$J_1^{28}$	28J1	+ means LM thrust in forward downrange direction - means LM thrust in negative downrange direction		
$\psi_p$	*	Sine of desired LM pitch attitude	$\pm 2^0$	$2^{-17}$
$\psi_y$	*	Sine of desired LM out-of-plane attitude	$\pm 2^0$	$2^{-17}$
$a_T$	AT	Computed LM thrust acceleration (fps <sup>2</sup> )	0 to $2^7$	$2^{-5}$
$\dot{r}$	HDOT	Present LM radial velocity (fps)	$\pm 2^{13}$	$2^{-4}$
$\ddot{r}_d$	RD2DOT	Desired radial acceleration (fps <sup>2</sup> )	$\pm 2^7$	$2^{-10}$
$S_{10}$	S10	Guidance Selector  = $\begin{cases} 0 & \text{Orbit Insertion} \\ 1 & \text{CSI} \\ 2 & \text{CDH} \\ 3 & \text{TPI Search} \\ 4 & \text{TPI Execute} \\ 5 & \text{External } \Delta V \end{cases}$		
$\underline{u}_1$	U1X, U1Y, U1Z	Unit vector in direction of LM inertial position vector	$\pm 2^1$	$2^{-16}$
$\Delta V_G$	DELVG	Magnitude of the velocity-to-be gained used in the 40 msec engine logic.	$2^{13}$	$2^{-4}$
$h$	H	LM Altitude (ft)	0 to $2^{23}$	$2^6$
$\ddot{y}_d$	YD2DOT	Desired out-of-plane acceleration (fps <sup>2</sup> )	$\pm 2^7$	$2^{-10}$
$\underline{v}_1$	V1X, V1Y, V1Z	a) Horizontal unit vector in transfer orbit plane in TPI mode b) Horizontal unit vector parallel to CSM orbit plane in all other modes	$\pm 2^1$	$2^{-16}$
$V_{DX}, V_{DY}, V_{DZ}$	VDX, VDY, VDZ	Components along the LM X, Y and Z body axes, respectively, of the accumulated sensed velocity increments which are updated every 2 seconds (fps)	$\pm 2^{13}$	$2^{-4}$
$V_{GX}, V_{GY}, V_{GZ}$	VSMGX, VSMGY, VSMGZ	Components along the LM X, Y and Z body axes, respectively, of the total LM velocity-to-be-gained during the burn (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{V}_G, V_G$	VGX, VGY, VGZ, VG	Vector of the velocity-yet-to-be-gained and its magnitude, respectively (fps). The components are along inertial axes	$\pm 2^{13}$	$2^{-4}$
$V_{GX}, V_{GY}, V_{GZ}$	*	Components along the LM X, Y and Z body axes, respectively, of the velocity-yet-to-be-gained (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{w}_1$	W1X, W1Y, W1Z	a) Unit vector normal to the transfer orbit plane in TPI mode b) Unit vector normal to CSM orbit plane in all other modes	$\pm 2^1$	$2^{-16}$
$\underline{x}_b$	A11, A12, A13	Unit vector along LM X -body axis: $\underline{x}_b = (a_{11}, a_{12}, a_{13})$	$\pm 2^1$	$2^{-16}$
$\underline{x}_{bD}$	A11BD, A12BD, A13BD	Unit vector along desired pointing direction in steering	$\pm 2^1$	$2^{-16}$
$\underline{y}_b$	A21, A22, A23	Unit vector of LM Y-body axis: $\underline{y}_b = (a_{21}, a_{22}, a_{23})$	$\pm 2^1$	$2^{-15}$
$\underline{z}_b$	A31, A32, A33	Unit vector along LM Z-body axis: $\underline{z}_b = (a_{31}, a_{32}, a_{33})$	$\pm 2^1$	$2^{-16}$
Constants:	$K_{26}^5, J_1^{21}, J_1^{22}, J_1^{28}$			

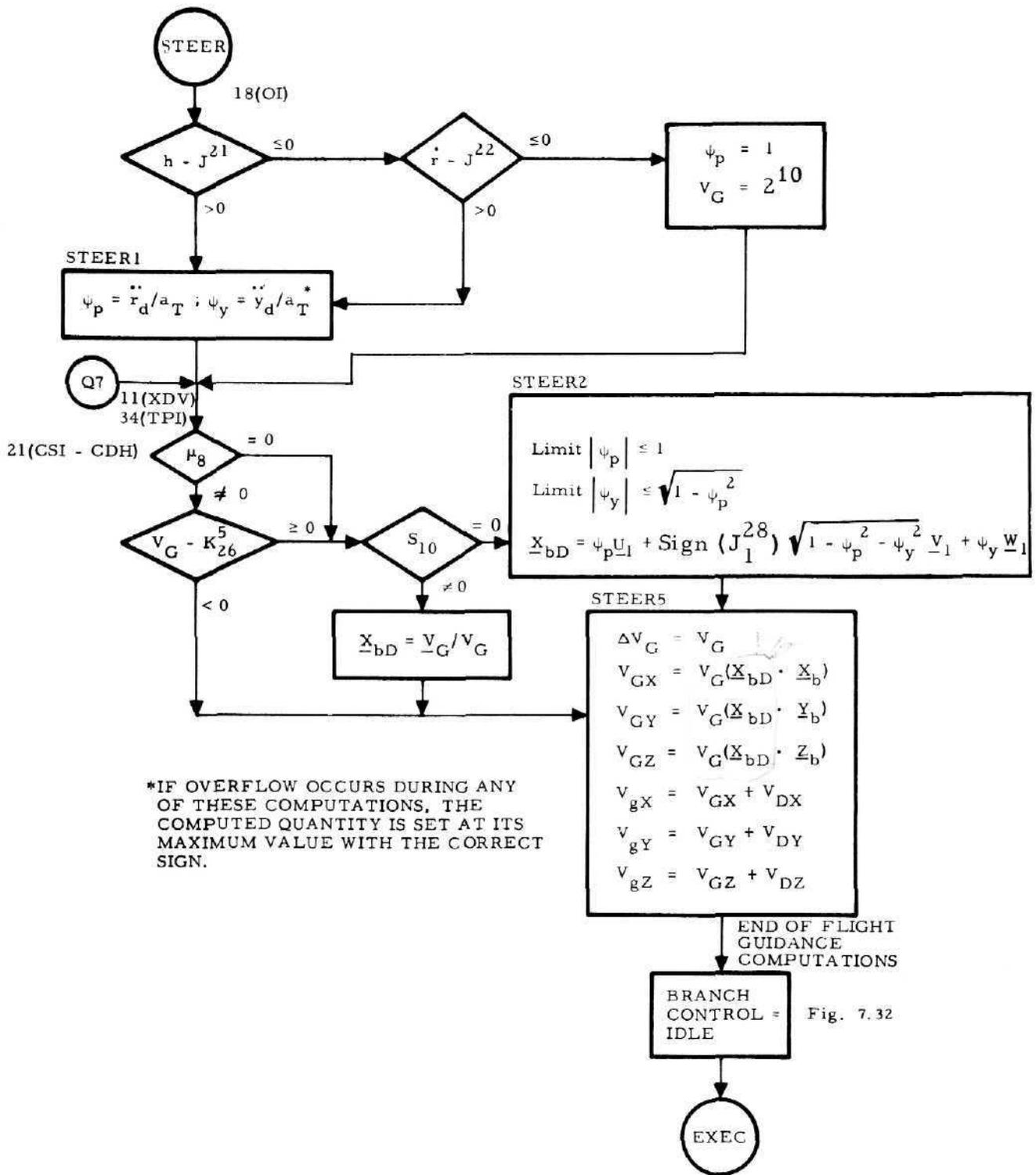
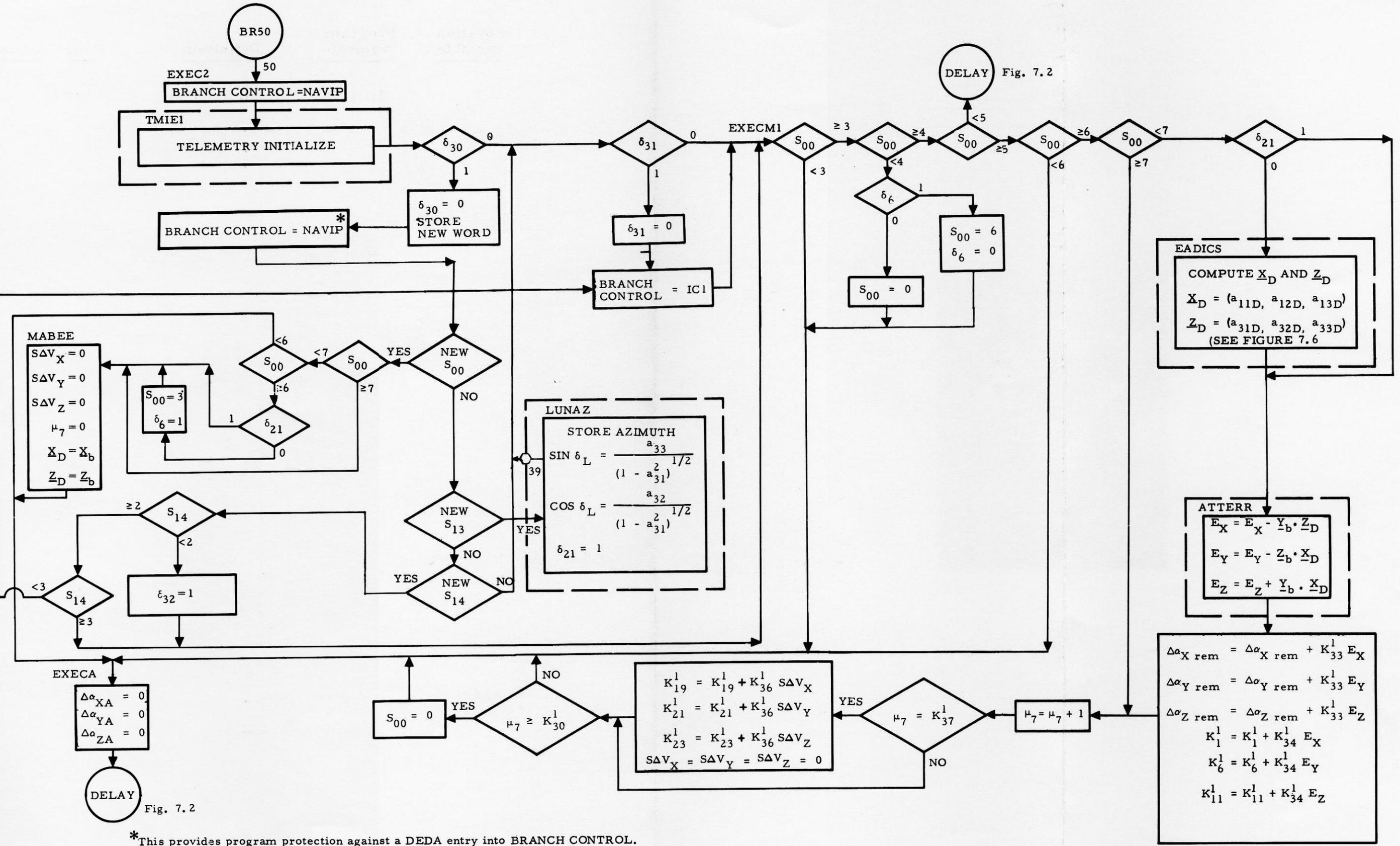


Figure 7.30. Pitch and Yaw Steering Equations Velocity-to-be-Gained Equations

Equation Variable	Program Symbol	Definition	Range	Quantization	Equation Variable	Program Symbol	Definition	Range	Quantization
$\Delta\alpha_{XA}, \Delta\alpha_{YA}, \Delta\alpha_{ZA}$	DAXA, DAYA, DAZA	Alignment error signals about LM X, Y and Z body axes, respectively (rad)	$\pm 2^{-6}$	$2^{-23}$	$a_{31D}, a_{32D}, a_{33D}$ ( $\underline{Z}_D$ )	A31D, A32D, A33D	Desired direction cosines of LM Z-body axis	$\pm 2^1$	$2^{-16}$
$\Delta\alpha_{xrem}, \Delta\alpha_{yrem}, \Delta\alpha_{zrem}$	DAXREM, DAYREM, DAZREM	$\Delta\alpha_x, \Delta\alpha_y, \Delta\alpha_z$ remainders used to obtain extra precision (rad)	$\pm 2^{-13}$	$2^{-30}$	$E_X, E_Y, E_Z$	EX, EY, EZ	Attitude error signals about LM X, Y and Z body axes, respectively (rad)	$\pm 2^2$	$2^{-15}$
$\delta_6$	DEL6	= $\begin{cases} 1 & \text{Set when in calibration mode on lunar surface to put computer in PGNCs/AGS align mode for 2 sec prior to going into calibrate mode} \\ 0 & \text{Rest after completion of PGNCs/AGS align} \end{cases}$			$S_{00}$	S00	AGS Function Selector		
$\delta_{21}$	DEL21	= $\begin{cases} 1 & \text{LM on lunar surface} \\ 0 & \text{LM not on lunar surface} \end{cases}$					$\begin{cases} 0 & \text{Attitude Hold} \\ 1 & \text{Guidance Steering} \\ 2 & \text{Z Axis Steering} \end{cases}$ } Inertial Reference Mode $\begin{cases} 3 & \text{PGNCs/AGS Align} \\ 4 & \text{Lunar Align} \\ 5 & \text{Body Axis Align} \\ 6 & \text{Calibrate} \\ 7 & \text{Accelerometer Calibrate} \end{cases}$ } Align Mode Any entry into $S_{13}$ will Store Landing Azimuth and set Lunar Flag ( $\delta_{21}$ )		
$\delta_{30}$	DDF	= $\begin{cases} 1 & \text{New word from DEDA} \\ 0 & \text{No new word from DEDA} \\ & (\delta_{30} \text{ is set in DEDA routine}) \end{cases}$			$S_{13}$	S13			
$\delta_{31}$	IDRF	= $\begin{cases} 1 & \text{Downlink complete} \\ 0 & \text{Downlink not complete} \end{cases}$			$S_{14}$	S14	Astronaut DEDA request, Ephemeris Input		
$\delta_{32}$	DL1F	= $\begin{cases} 1 & \text{Downlink enabled} \\ 0 & \text{Downlink not enabled} \end{cases}$					= $\begin{cases} 0 & \text{Initialization Complete} \\ 1 & \text{Initialize LM and CSM via Downlink} \\ 2 & \text{Initialize LM via DEDA} \\ 3 & \text{Initialize CSM via DEDA} \end{cases}$		
$\sin \delta_L, \cos \delta_L$	SIDELL, CODELL	Sine and cosine respectively, of the stored landing azimuth, $\delta_L$ , which is the angle at touchdown between the y inertial axis and the projection of the Z-body axis onto the yz inertial plane, measured positive in a counterclockwise direction from the y inertial axis when viewed from above the landing site	$\pm 2^1$	$2^{-16}$	$\Delta V_X, \Delta V_Y, \Delta V_Z$	SDVX, SDVY, SDVZ	Sum of body axis velocity increments accumulated during accelerometer bias calibration (fps)	$\pm 2^1$	$2^{-16}$
$\mu_7$	MU7	Counts 2 sec cycles of gyro calibrate	0 to $2^{17}$	$2^0$	$\underline{X}_b$	A11, A12, A13	Unit vectors of LM body attitude	$\pm 2^1$	$2^{-16}$
$a_{11D}, a_{12D}, a_{13D}$ ( $\underline{X}_D$ )	A11D, A12D, A13D	Desired direction cosines of LM X-body axis	$\pm 2^1$	$2^{-16}$	$\underline{Y}_b$	A21, A22, A23			
					$\underline{Z}_b (a_{31}, a_{32}, a_{33})$	A31, A32, A33			

Constants:  $K_{33}^1, K_{34}^1, K_{36}^1, K_{30}^1, K_1^1, K_6^1, K_{11}^1, K_{19}^1, K_{21}^1, K_{23}^1, K_{37}^1$



\*This provides program protection against a DEDA entry into BRANCH CONTROL.

Figure 7.31. The Executive Branch, Branch 50

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\mu_{10}$	MU10	Cycle Counter	0 to 49	$2^0$

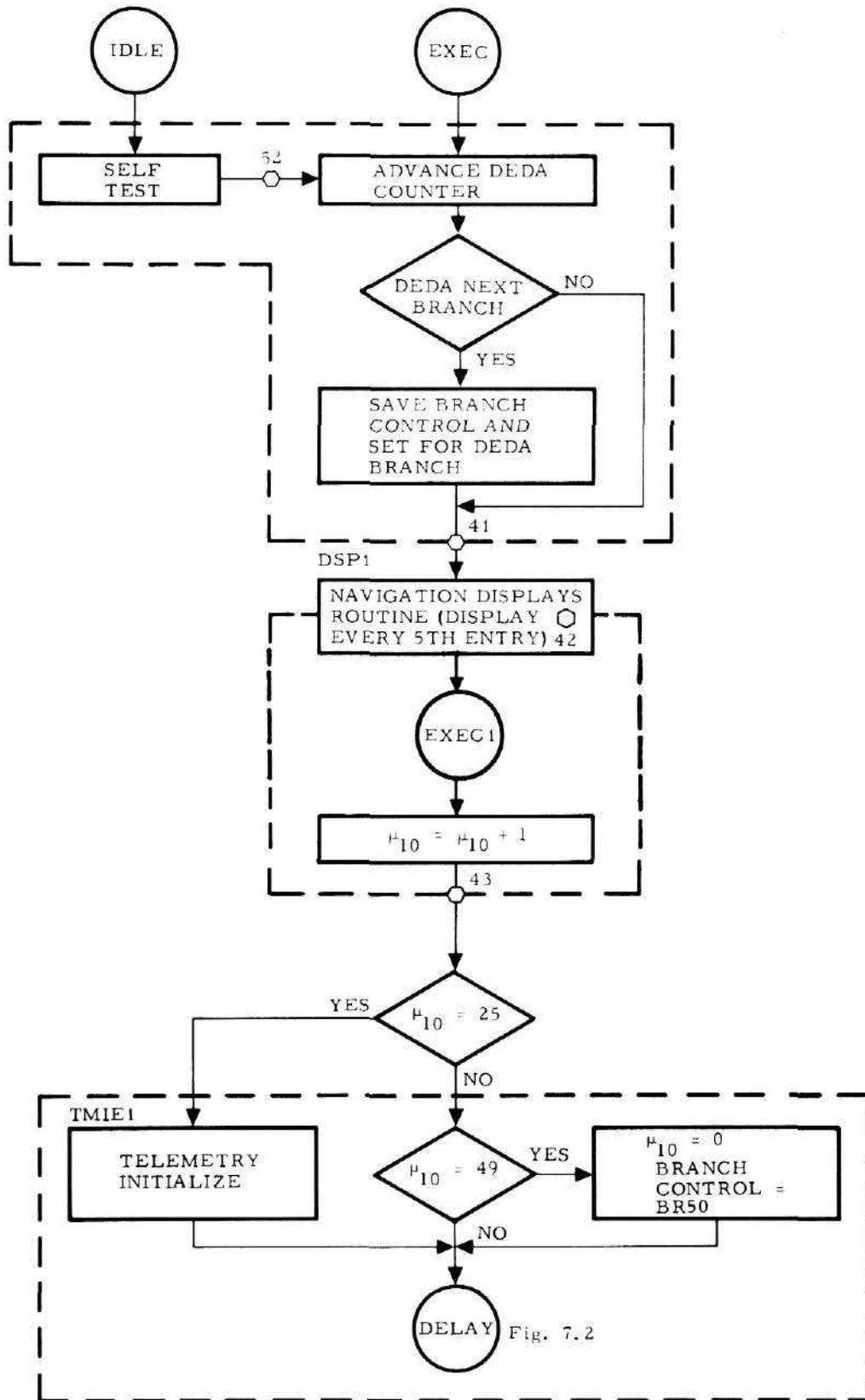


Figure 7.32. Branch Control

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\alpha_i$	AI	Semi-major axis of ellipse as computed in orbit parameter subroutine (ft)	0 to $2^{23}$	$2^6$
$\mu_2$		Counter which controls iterations of Kepler's equations in ellipse prediction	0 to $2^{17}$	1
$C_i$	CI	Product of eccentricity and cosine of eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$\Delta E_i$	*	Difference between the present and final eccentric anomalies (rad)	0 to $2^3$	$2^{-14}$
$\sin(\Delta E_i)$ $\cos(\Delta E_i)$	*	Sine and cosine, respectively, of $\Delta E_i$	$\pm 2^1$	$2^{-16}$
$f$ $i$ $g$ $\dot{g}$	*	Coefficients used in Keplerian orbit prediction equations	$\pm 2^1$ $\pm 2^{-9}$ $\pm 2^{11}$ $\pm 2^1$	$2^{-16}$ $2^{-26}$ $2^{-6}$ $2^{-16}$
$\Delta M_i$	*	Difference between the present and final mean anomalies, or $\Delta M_i = n_i T_i$ (rad)	0 to $2^3$	$2^{-14}$
$n_i$	NI	Mean orbital angular rate output by orbit parameter subroutine (rad/sec)	0 to $2^{-9}$	$2^{-26}$
$\underline{r}_0$	ROX, ROY, ROZ	Inertial position vector input to orbit parameter subroutine (ft)	$\pm 2^{23}$	$2^6$
$r_i$	RO	Radial distance output by orbit parameter subroutine (ft)	$2^{23}$	$2^6$
$\underline{r}_i$	RIX, RIY, RIZ	Inertial position vector output by ellipse predictor (ft)	$\pm 2^{23}$	$2^6$
$S_i$	SI	Product of eccentricity and sine of the eccentric anomaly output by orbit parameter subroutine	$\pm 2^0$	$2^{-17}$
$T_i$	TI	Prediction time input to ellipse prediction subroutine (sec)	$\pm 2^{13}$	$2^{-4}$
$v_0$	*	Magnitude of velocity input to orbit parameter subroutine (fps)	0 to $2^{13}$	$2^{-4}$
$\underline{v}_0$	VOX, VOY, VOZ	Inertial velocity vector input to orbit parameter subroutine (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{v}_i$	VIX, VIY, VIZ	Inertial velocity vector output by ellipse prediction (fps)	$\pm 2^{13}$	$2^{-4}$
$x_{11}$	*	Error in trial solution of Kepler's equation (rad)	$\pm 2^1$	$2^{-16}$
$x_{12}$	*	Derivative of $x_{11}$	$\pm 2^1$	$2^{-16}$
Constants:	$K_2^2, K_1^2$			

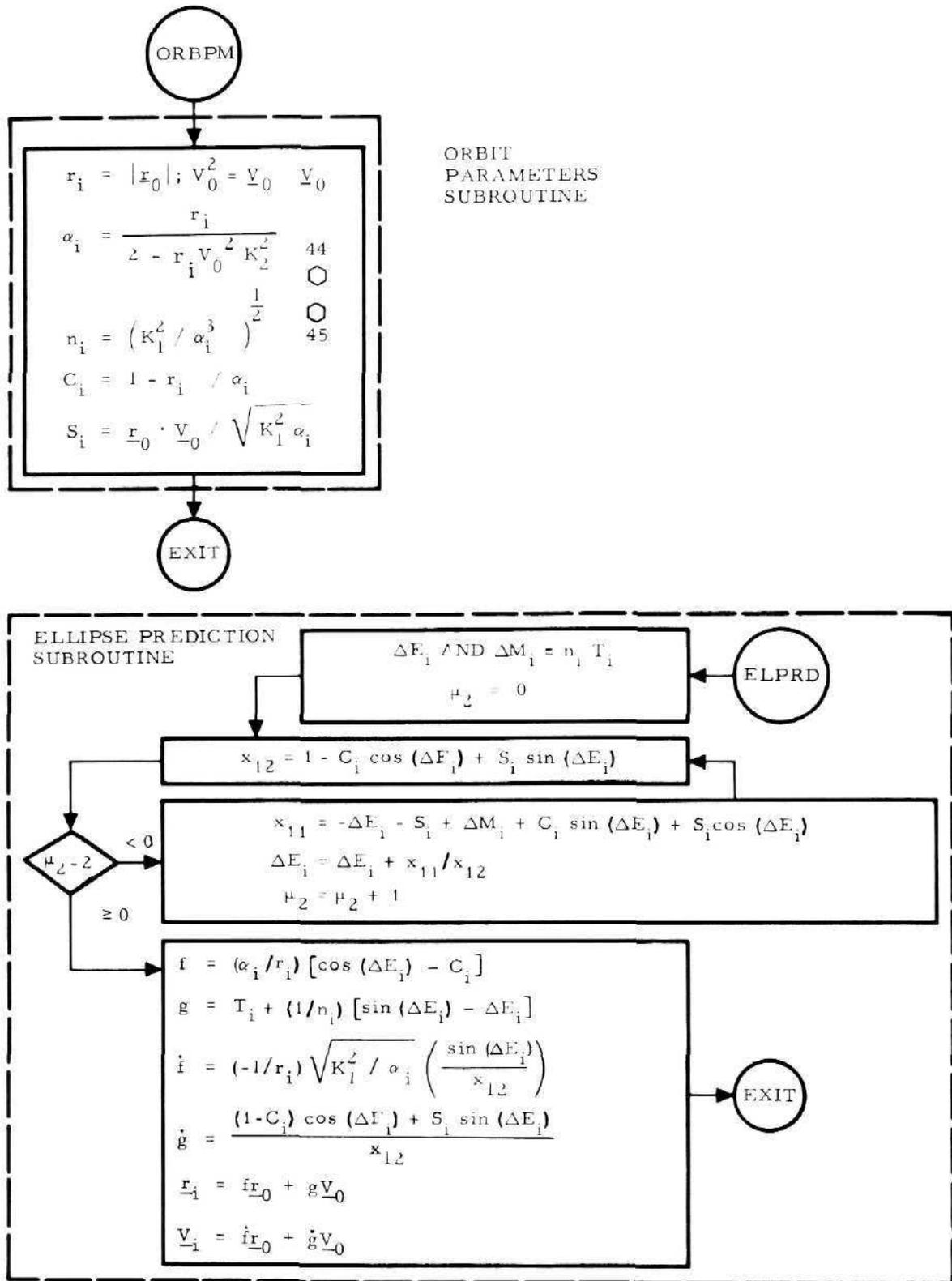


Figure 7.33. Orbit Parameters and Ellipse Prediction Subroutines

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\underline{r}, r$	RX,RY,RZ,R	Present LM inertial position vector and magnitude, respectively (ft)	$\pm 2^{23}$	$2^6$
$\underline{v}$	VX,VY,VZ	LM inertial position vector (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{G}\Delta t$	GXDT, GYDT, GZDT	Gravity vector computed in present 2 sec computer cycle (fps)	$\pm 2^7$	$2^{-10}$
$\underline{G}_{n-1}\Delta t$	*	Gravity vector computed in previous 2 sec computer cycle (fps)	$\pm 2^7$	$2^{-10}$
$h$	H	LM altitude above lunar landing site (ft)	$\pm 2^{23}$	$2^6$
$\dot{h}$	HDOT	LM altitude rate (fps)	$\pm 2^{13}$	$2^{-4}$
$\dot{r}$	HDOT	LM altitude rate (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{U}_1$	U1X,U1Y,U1Z	Normalized LM position vector	$\pm 2^1$	$2^{-16}$
$\underline{\Delta IG}$	DIGX, DIGY, DIGZ	Integrated gravity acceleration vector per 2-sec computer cycle (fps)	$\pm 2^7$	$2^{-10}$

Constants:  $K_4^2, J^5$

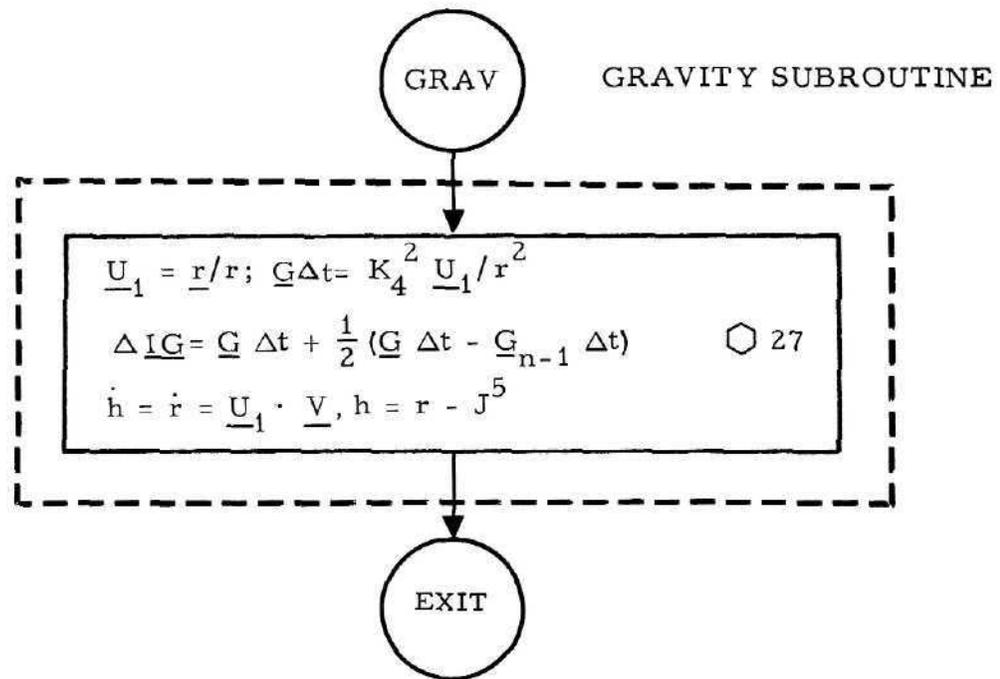


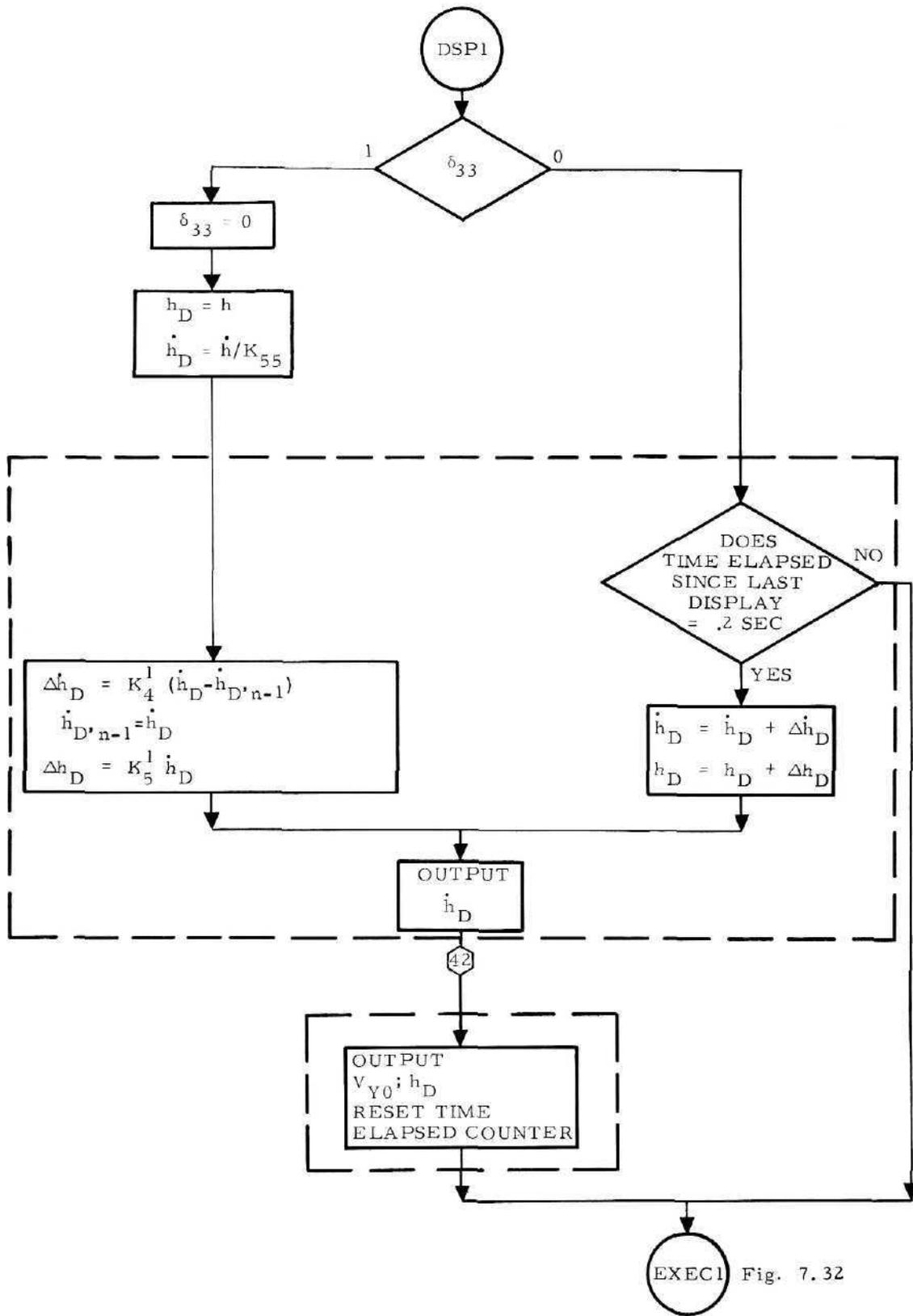
Figure 7. 34. Gravity Subroutine

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\delta_{33}$	DSPF1	Flag indicating that in this basic 20 msec cycle the navigation equations have been updated. Flag set = 1 at end of Branch 1 calculation.		
$h, h_D$	H, POUT	LM altitude above the landing site (ft) (The subscript D stands for displayed value which is interpolated when $\delta_{33} \neq 1$ )	$76,840^\dagger$	$2.345^{**}$
$\dot{h}, \dot{h}_D$	HDOT, PDOUT	LM altitude rate (fps) (The subscript D has the same meaning as for $h, h_D$ ).	$\pm 8191^\dagger$	$0.5^{**}$
$\dot{h}_{D,n}, \dot{h}_{D,n-1}$	*, *	Value of $\dot{h}_D$ in the present 2 sec computing cycle and the previous 2 sec computing cycle, respectively (fps)	$\pm 8191^\dagger$	$0.5^{**}$
$\Delta h_D, \Delta \dot{h}_D$	DPOUT, DPDOUT	Increments in displayed altitude and altitude rate, respectively (ft, fps)	$\pm 3277$ ft, $18$ fps $^\dagger$	$0.2925$ ft $1/16$ fps $^{**}$
$\underline{V}$	VX, VY, VZ	LM inertial velocity vector (fps)	$\pm 2^{13}$	$2^{-4}$
$V_{Y0}$	VY0	Out-of-CSM plane component of LM velocity (fps)	$200^\dagger$	$200 \times 2^{-8}$

Constants:  $K_{55}^1, K_4^1, K_5^1$

$^\dagger$ Output range

$^{**}$ The output registers have less bits than the arithmetic registers internal to the AGS computer. The values given are for the output registers.



EXEC1 Fig. 7.32

Figure 7.35. Display Routine: Altitude, Altitude Rate and Out of CSM Plane Velocity

## 7.1 DEFINITIONS OF DISCRETES, FLAGS, AND COUNTERS

7.1.1 Discretes7.1.1.1 "Discrete Word One"

"Discrete Word One" has the following discretes:

## a) From the Control Electronics System (CES)

$$\beta_1 = \begin{cases} 1 \text{ Descent Engine ON} \\ 0 \text{ Descent Engine OFF} \end{cases}$$

$$\beta_2 = \begin{cases} 1 \text{ Ascent Engine ON} \\ 0 \text{ Ascent Engine OFF} \end{cases}$$

$$\beta_3 = \begin{cases} 1 \text{ Follow-up Signal Present} \\ 0 \text{ Follow-up Signal Not Present} \end{cases}$$

$$\beta_4 = \begin{cases} 1 \text{ Auto Discrete YES} \\ 0 \text{ Auto Discrete NO} \end{cases}$$

$$\beta_5 = \begin{cases} 1 \text{ Abort Stage YES} \\ 0 \text{ Abort Stage NO} \end{cases}$$

$$\beta_6 = \begin{cases} 1 \text{ Abort YES} \\ 0 \text{ Abort NO} \end{cases}$$

## b) From PGNCS Downlink

$$\xi_1 = \begin{cases} 1 \text{ Downlink Stop Pulse Received} \\ 0 \text{ Downlink Stop Pulse Not Received} \end{cases}$$

## c) From AGS Telemetry

$$\xi_2 = \begin{cases} 1 \text{ Telemetry Stop Pulse Received} \\ 0 \text{ Telemetry Stop Pulse Not Received} \end{cases}$$

The status of the CES discretes is checked each 40 msec and updated at the start of the 40 msec computations (see Figure 7.7). The PGNCS Downlink and AGS Telemetry discretes are checked as shown in Figures 8.2 and 8.4.

7.1.1.2 "Discrete Word Two"

This word contains the following discrettes:

## a) From the Ground Service Equipment (GSE)

$\xi_3$  GSE Discrete No. 1

$\xi_4$  GSE Discrete No. 2

$\xi_5$  GSE Discrete No. 3

## b) From the DEDA

$\xi_6$  DEDA Clear

$\xi_7$  DEDA Hold

$\xi_8$  DEDA Enter

$\xi_9$  DEDA Readout

These discrettes are checked every 40 msec and updated in a portion of the 40 msec computations (see Figure 7.8).

7.1.2 Internal Decision Logic Flags

A complete list of the internal decision logic flags is given below. The 1 state corresponds to a negative value and the 0 state to a positive value in the computer.

<u>Flag</u>	<u>Function</u>
$\delta_2 =$	$\left\{ \begin{array}{l} 1 \text{ Descent Section Staged} \\ 0 \text{ Descent Section Not Staged} \end{array} \right.$
$\delta_5 =$	$\left\{ \begin{array}{l} 1 \text{ Lock Attitude Hold reference direction cosines} \\ 0 \text{ Set commanded direction cosines } (\underline{X}_D, \underline{Z}_D) \text{ equal} \\ \text{to present body direction cosines } (\underline{X}_b, \underline{Z}_b) \end{array} \right.$
$\delta_6 =$	$\left\{ \begin{array}{l} 1 \text{ Set in calibration mode to put computer in PGNCS/} \\ \text{AGS align mode for 2 sec prior to going into cali-} \\ \text{brate mode when not on lunar surface} \\ 0 \text{ Reset after completion of PGNCS/AGS align} \end{array} \right.$

<u>Flag</u>	<u>Function</u>
$\delta_{10} =$	$\left\{ \begin{array}{l} 1 \text{ Transfer orbit computations made} \\ \text{for CSM at rendezvous retarded by } J^4 \\ 0 \text{ Transfer orbit computations made} \\ \text{for CSM at rendezvous} \end{array} \right.$
$\delta_{20} =$	$\left\{ \begin{array}{l} 1 \text{ Engine followup indicator} \\ 0 \text{ AGS engine control} \end{array} \right.$
$\delta_{21} =$	$\left\{ \begin{array}{l} 1 \text{ LM on lunar surface} \\ 0 \text{ Not on lunar surface} \end{array} \right.$
$\delta_{30} =$	$\left\{ \begin{array}{l} 1 \text{ New word from DEDA} \\ 0 \text{ No new word from DEDA} \end{array} \right.$
$\delta_{31} =$	$\left\{ \begin{array}{l} 1 \text{ Downlink complete} \\ 0 \text{ Downlink not complete} \end{array} \right.$
$\delta_{32} =$	$\left\{ \begin{array}{l} 1 \text{ Downlink enabled} \\ 0 \text{ Downlink not enabled} \end{array} \right.$
$\delta_{33} =$	$\left\{ \begin{array}{l} 1 \text{ Navigation display update cycle} \\ 0 \text{ Not navigation display update cycle} \end{array} \right.$

<u>Flag</u>	<u>Function</u>
$\delta_{34} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA clear discrete input received} \\ 0 \text{ Not received} \end{array} \right.$
$\delta_{35} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA hold discrete input received} \\ 0 \text{ Not received} \end{array} \right.$
$\delta_{36} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA enter discrete input received} \\ 0 \text{ Not received} \end{array} \right.$
$\delta_{37} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA readout discrete input received} \\ 0 \text{ Not received} \end{array} \right.$
$\delta_{38} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA clear mode} \\ 0 \text{ Not DEDA clear mode} \end{array} \right.$
$\delta_{39} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA hold mode} \\ 0 \text{ Not DEDA hold mode} \end{array} \right.$
$\delta_{41} =$	$\left\{ \begin{array}{l} 1 \text{ DEDA readout mode} \\ 0 \text{ Not DEDA readout mode} \end{array} \right.$
$\delta_{60} =$	$\left\{ \begin{array}{l} 1 \text{ Downlink identification word found} \\ 0 \text{ Downlink identification word not found} \end{array} \right.$

7.1.3 Counters

A complete list of the counters used together with their function is given below.

<u>Counter</u>	<u>Function</u>
$\mu_2^*$	Controls iterations of Kepler's equations in ellipse prediction
$\mu_3$	Controls p-iterator sequence
$\mu_6$	Controls delay in Attitude Hold after Abort Stage before allowing guidance steering
$\mu_7$	Counts 2 sec cycles of calibrate
$\mu_8$	Ullage counter. $K_9^1$ consecutive 2-sec compute cycles with thrust acceleration $a_T \geq K_{35}^4 \text{ fps}^2$ are required before the Engine ON signal can be issued.
$\mu_{10}$	Telemetry and 2-sec computation synchronization counter. This insures that navigation equations are updated every 2 sec and telemetry is reset every second.
$\mu_{20}^{**}$	DEDA counter, insures that in each 12 <sup>th</sup> basic 2-sec branch the program goes through the DEDA subroutine
$\mu_{30}$	PGNCS Downlink input counter which counts the 16 words entered into the downlink register when commanded.
$\mu_{17}$	Controls entry into the radar filter equations each 16 sec.

---

\* This counter does not appear explicitly in the program.

\*\* This counter is implemented as a shift counter (DEDASC).

## 7.2 PROGRAM SUBROUTINES

Various subroutines are included in the program in order to maximize computer memory usage. These subroutines accept data in specified temporary storage cells, perform the required operations, leave the answers in temporary storage cells, and return control to the point in the program from which they were accessed. The following paragraphs describe the subroutines in terms of their method and temporary storage usage.

### 7.2.1 Normalize

**Purpose:** To rescale a double-precision number such that its fixed-point magnitude is greater than or equal to  $1/4$ .

**Input:** Most significant part of input in the accumulator, least significant part in cell TS2 (temporary storage 2).

**Output:** TS1 contains the normalized number; that is the original number  $X$  times an even power of two ( $2^{2i}$ ) such that  $2^{2(i+1)}(X)$  would cause computer overflow. The accumulator contains an instruction to multiply by  $2^{-i}$ . The maximum value of  $i$  is 8.

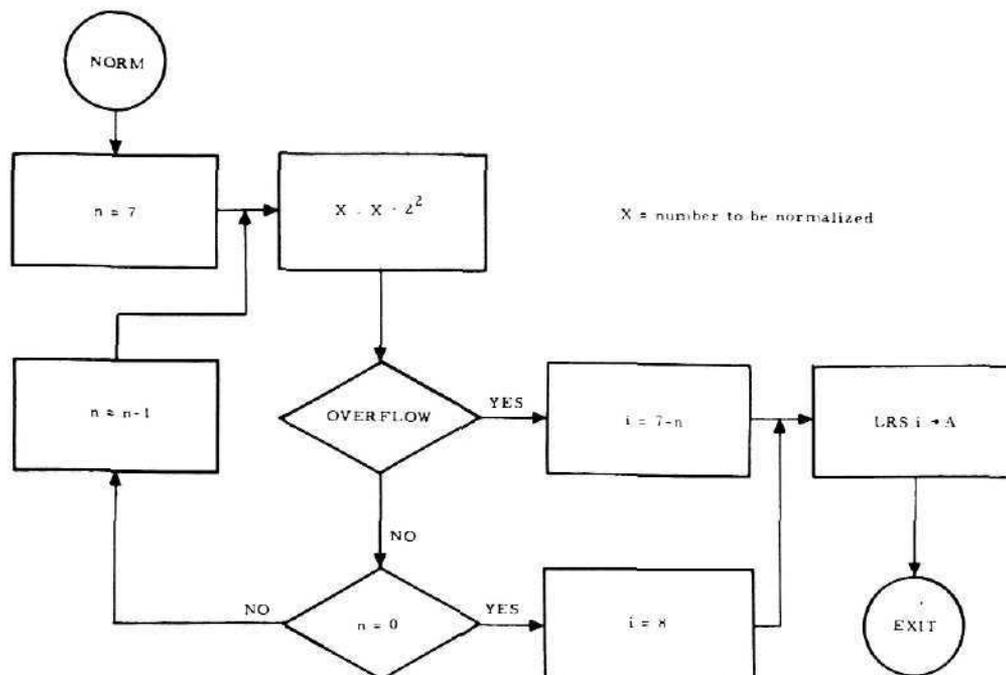


Figure 7.36. Normalize Subroutine

7.2.2 Square Root

**Purpose:** To obtain the square root of a given number,  $X^2$ .

**Input:**  $X^2$  in accumulator at even scaling  $2N$ .

**Output:**  $(X^2)^{1/2}$  in accumulator at scaling  $N$ ; that is, one half the scaling of the input.

**Method:** Heron's method which can be derived from the Newton-Raphson technique is used. Successive approximations to the root are found by:

$$X_n = 1/2 \left( X_{n-1} + \frac{X^2}{X_{n-1}} \right)$$

This method is rapidly convergent for  $1/4 \leq X^2 < 1$  with an initial guess  $X_0 = .375 + .625 X^2$ . The normalize subroutine is utilized to put  $X^2$  into the above range.

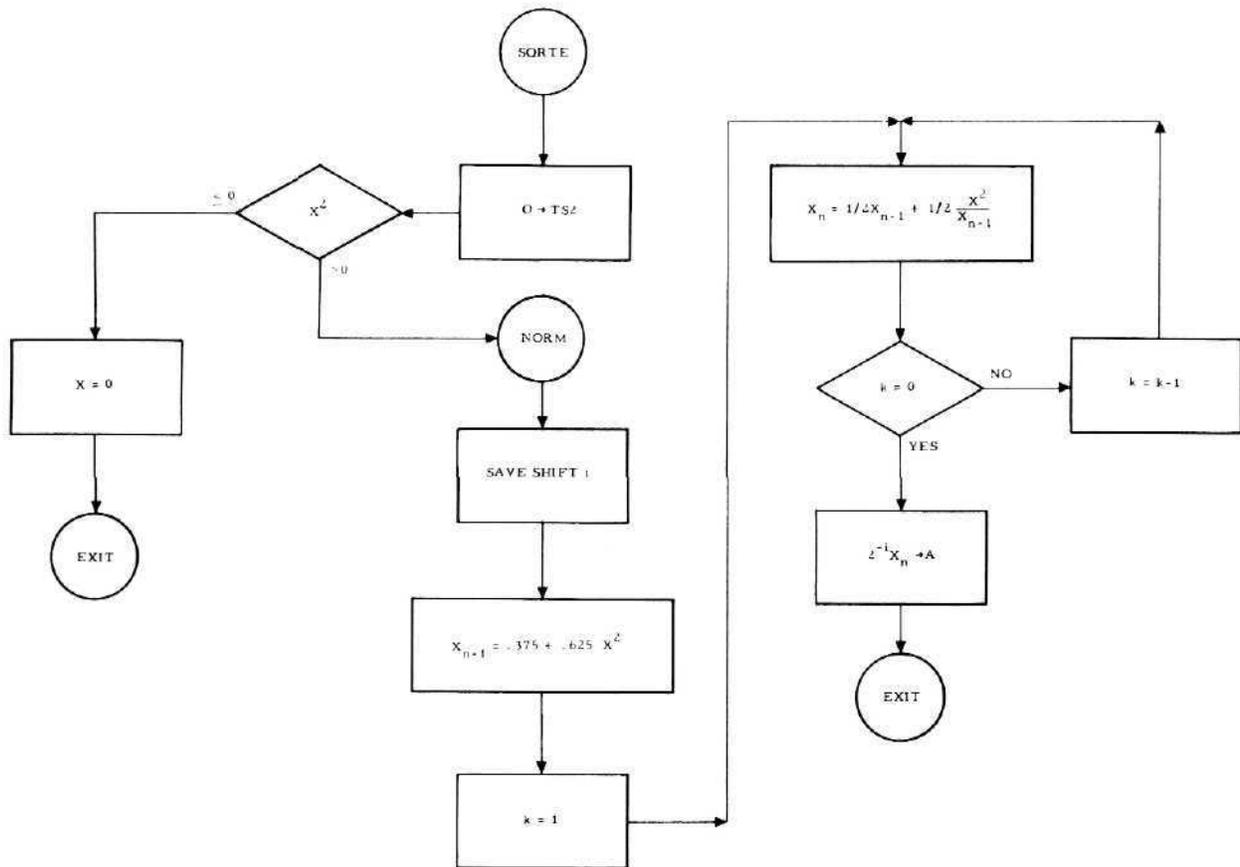


Figure 7.37. Square Root Subroutine

### 7.2.3 Sine-Cosine

**Purpose:** To obtain the sine and cosine of a given angle.

**Input:** Angle,  $a$ , in radians in the accumulator at a binary scaling of 3.

**Output:** Sin  $a$  in TS1; cos  $a$  in TS0 at scaling of 1.

**Method:** Truncated Taylor's series with coefficients adjusted slightly to provide a best fit in range  $-\pi/2$  to  $+\pi/2$ .

$$\sin X = X + K_3 X^3 + K_5 X^5 + K_7 X^7$$

where

$$K_3 = - .16665554 \approx - \frac{1}{3!}$$

$$K_5 = + .0083119 \approx \frac{1}{5!}$$

$$K_7 = - .00018488 \approx - \frac{1}{7!}$$

The subroutine first converts the input angle to an angle ( $\theta_1$ ) in the range  $-\pi/2$  to  $\pi/2$  at a binary scaling of +1 such that the sine of the reduced angle equals the sine of the input angle. A second angle ( $\theta_2$ ) in the range of  $-\pi/2$  to  $\pi/2$  is also generated such that the sine of this second angle equals the cosine of the input angle. The above formula is then applied to the two reduced angles to obtain both the sine and cosine of the original angle.

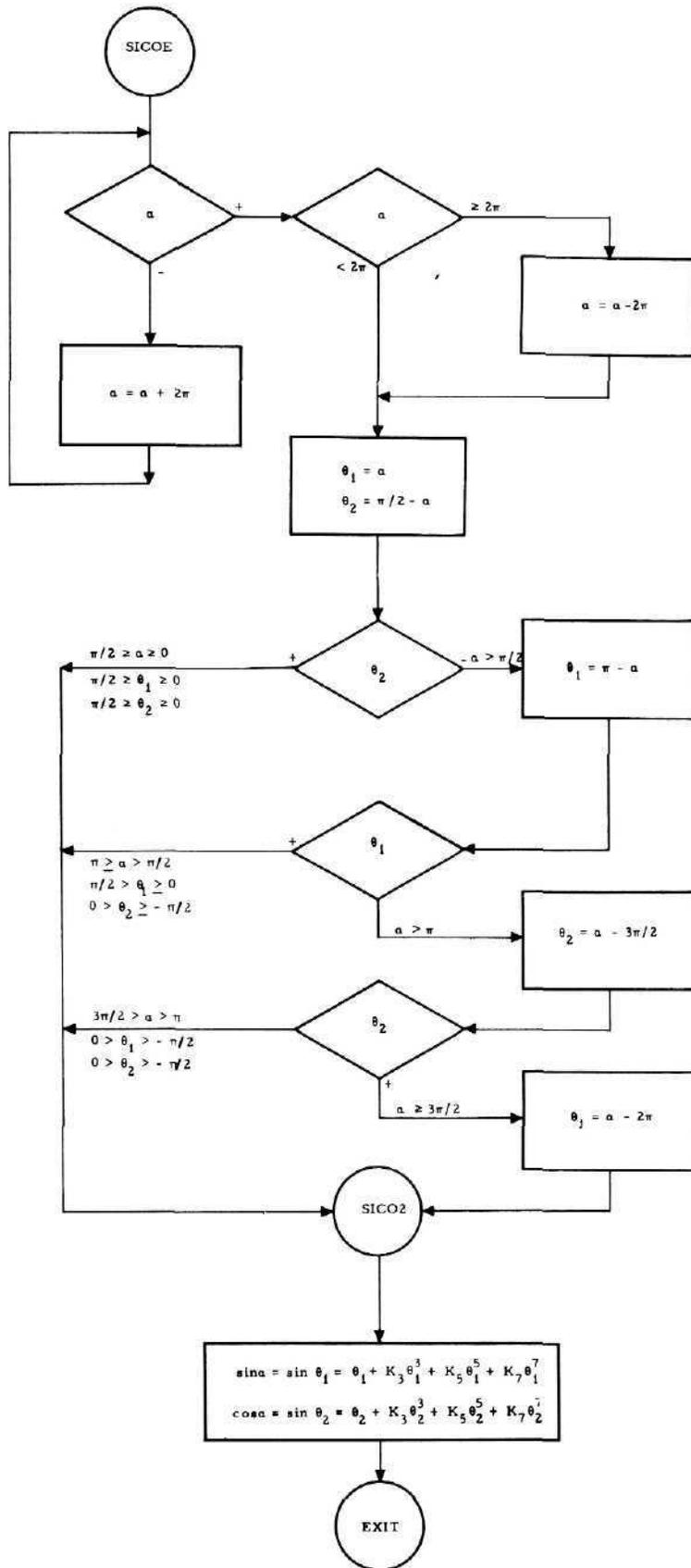


Figure 7. 38. Sine - Cosine Subroutine

#### 7.2.4 Arctangent

**Purpose:** To compute an angle, given numbers proportional to its sine and cosine.

**Inputs:** Sin  $\alpha$  in TS3 and cos  $\alpha$  in TS4. These numbers may be proportional to the sine and cosine. If TS3 + TS4 is greater than 2 the inputs are multiplied by 1/2 to avoid overflow in the computations.

**Output:**  $\alpha$  in accumulator at binary scale + 3 where  $0 \leq \alpha < 2\pi$  radians.

**Method:** A best fit polynomial approximation of the  $\tan^{-1}$  function valid between the range of -1 to +1 is used where:

$$\tan^{-1}X = C_1X + C_3X^3 + C_5X^5 + C_7X^7 \quad (-1 \leq X \leq 1)$$

$$C_1 = .999215$$

$$C_3 = -.3211819$$

$$C_5 = .1462766$$

$$C_7 = -.0389929$$

To obtain the tangent in the range -1 to +1, the following is performed:

a) For 1st and 4th quadrant angles define:

$\sin \alpha' = |\sin \alpha|$ ,  $\cos \alpha' = \cos \alpha$ . Then

$\pi/2 \geq \alpha' \geq 0$  and  $-1 \leq \tan(\alpha' - \pi/4) \leq 1$ .

$$\tan(\alpha' - \pi/4) = \frac{|\sin \alpha| - \cos \alpha}{\cos \alpha + |\sin \alpha|}$$

b) For 2nd and 3rd quadrant angles define:

$\sin \alpha' = \sin \alpha$ ,  $\cos \alpha' = \cos \alpha$ . Then

$\pi \geq \alpha' > \pi/2$  and  $-1 < \tan(\alpha' - 3\pi/4) \leq 1$ .

$$\tan(\alpha' - 3\pi/4) = \frac{|\sin \alpha| + \cos \alpha}{\cos \alpha - |\sin \alpha|}$$

The angle  $\alpha$  is recovered from  $\alpha'$  by a final quadrant check using  $\sin \alpha$ .

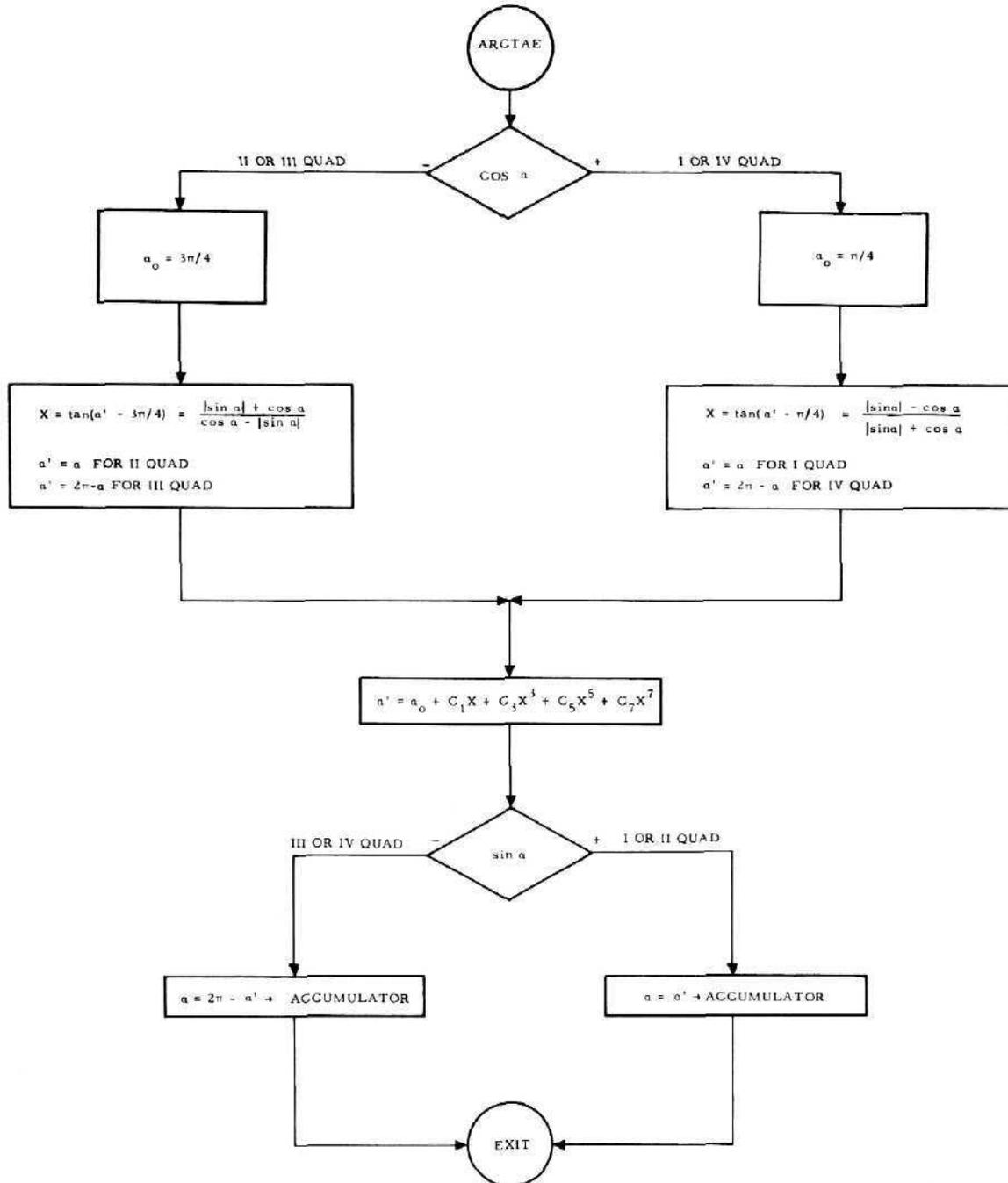


Figure 7.39. Arctangent Subroutine

### 7.2.5 Vector Cross Product

**Purpose:** To compute the cross product of two vectors specified in terms of their Cartesian components.

**Inputs:** Vector A in TS4, TS5, and TS6, vector B in TS10, TS11 and TS12.

**Outputs:** Vector C = A x B in TS14, TS15 and TS16.

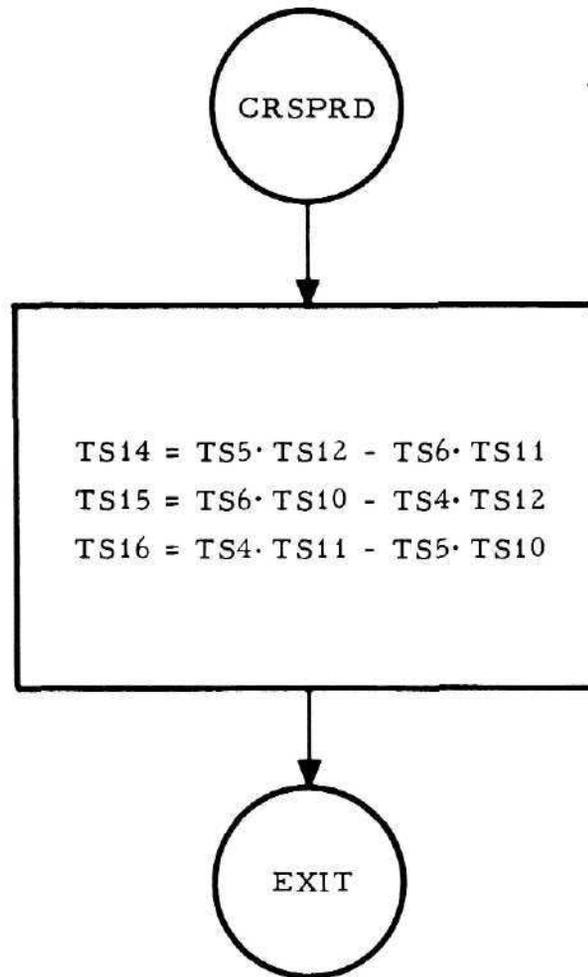


Figure 7.40. Cross Product Subroutine

7.2.6 Normalize Vector

Purpose: Convert a given vector to unity magnitude.

Input: Vector  $\underline{A}$  in TS14, TS15, and TS16.

Outputs:  $|\underline{A}|^2$  in TS17 at twice input scaling

$|\underline{A}|$  in TS13 at input scaling

$\frac{A_1}{|\underline{A}|}$  in TS10,  $\frac{A_2}{|\underline{A}|}$  in TS11,  $\frac{A_3}{|\underline{A}|}$  in TS12, all at binary scaling of 1.

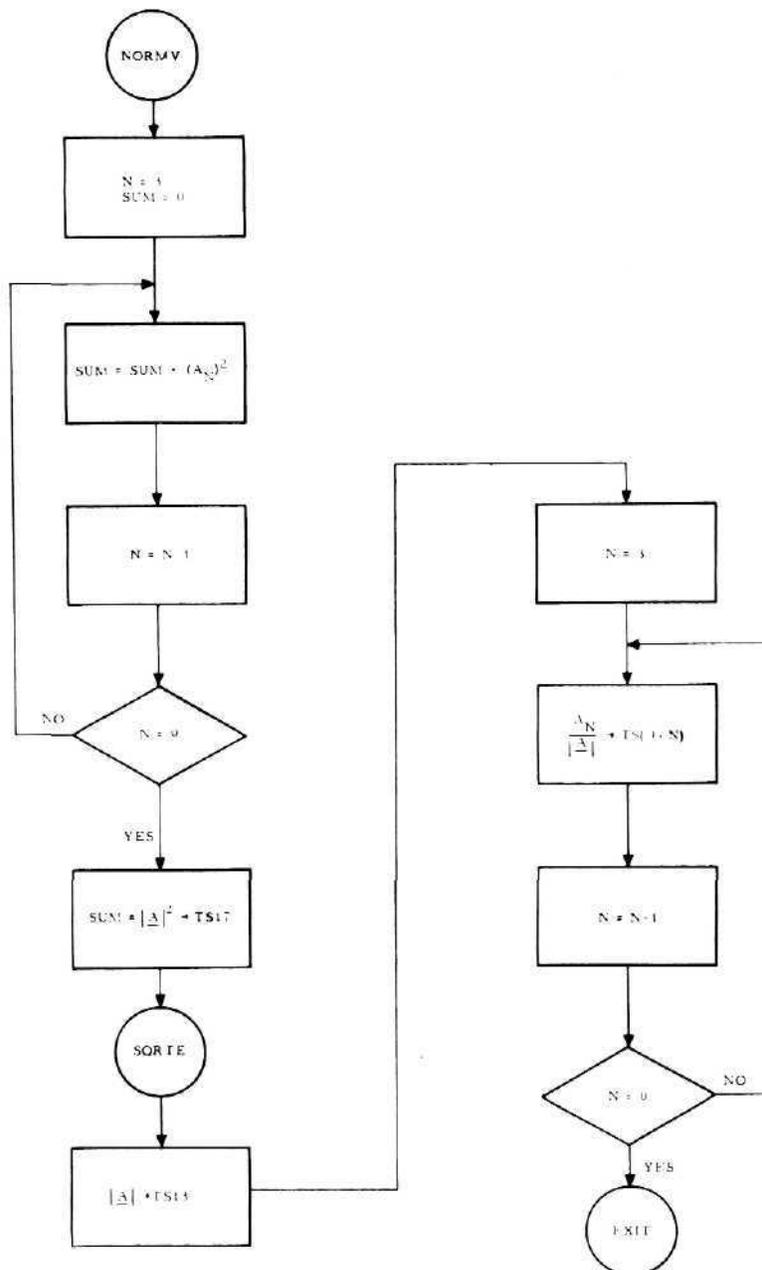


Figure 7.41. Normalize Vector Subroutine

### 7.2.7 Double Precision Vector Magnitude

**Purpose:** Obtain the accurate magnitude of a vector.

**Input:** Vector A in TS4, TS5 and TS6.

**Output:**  $|\underline{A}|$  in Accumulator at same scaling as input quantity.

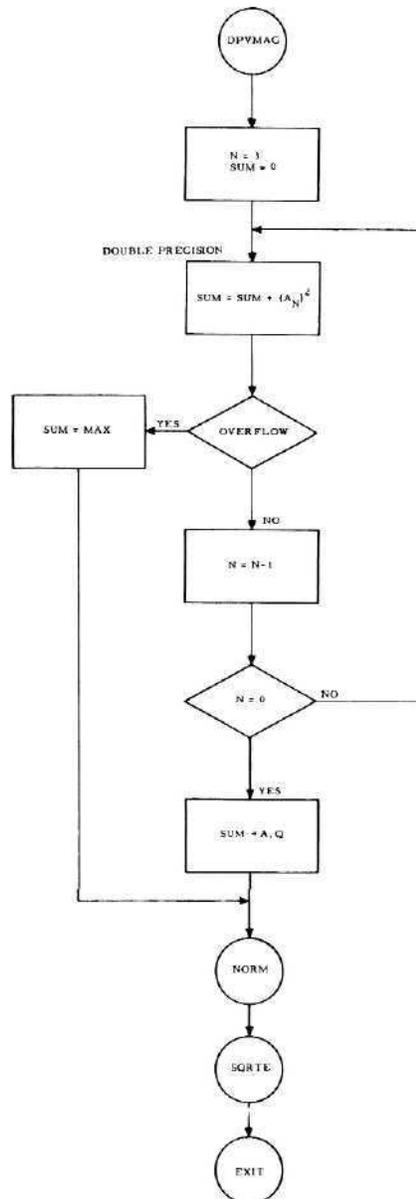


Figure 7.42. Vector Magnitude Subroutine

## 8. DESCRIPTION OF AUXILIARY ROUTINES

### 8.1 GROUND SUPPORT EQUIPMENT (GSE) SERVICE ROUTINE FOR THE AEA FLIGHT PROGRAM

The softwired portion of the AEA flight program is initially transferred to the computer through the GSE service routine shown schematically in Figure 8.1. This GSE service routine is a hardwired program which facilitates the transfer of a block of computer words from the ground support equipment (GSE) to the erasable portion of the AEA's memory. The number of 18-bit words in the block is determined by the GSE. Initiation of the information transfer is also controlled by the GSE. Figure 8.6 shows how receipt of the GSE 1 discrete  $\xi_3$  causes the program to transfer to the service routine.

The starting address, from which this block of information is to be stored, is determined by the first word transmitted to the computer. This word must contain the starting address and a special code which identifies the word. The format of this word, expressed in octal, is as follows:

$$77X_1 X_2 X_3 X_4$$

Bits 0 through 5 (bits are numbered from left to right) constitute the start code, and bits 6 through 17 contain the starting address.

#### 8.1.1 Use of GSE Discrettes

The routine utilizes two input discrettes (GSE 1 and 2) and one output discrete (GSE 5) to control the transfer information. GSE 1 is set by the test equipment to notify the computer that a word has been transmitted. The computer acknowledges that the word has been accepted and stored by setting GSE 5. GSE 2 is used in conjunction with GSE 1 to denote that the present word being transmitted constitutes the end of the block. The last word is not processed by this routine. When GSE 2 is sensed, the program transfers control to the first word loaded.

The GSE first loads the word into the downlink telemetry register. After the word has been loaded, GSE 1 is set so that the computer may process the word. The computer's response (set GSE 5) is used by the test equipment to initiate the transfer of the next word.

#### 8.1.2 Flow Diagram of GSE Routine

Upon entering the routine, GSE 1 is sampled to determine if a loading operation is required at this time. If the discrete is not present, the routine will exit to the Initial Start-up routine. If the discrete is present, the computer will input the word and extract the starting address. This address is inserted into a store command to facilitate sequential storage of the information from this starting address. The computer then responds by setting GSE 5. The computer waits until GSE 1 is reset. The program will then wait for additional settings of GSE 1 which denote that new words have been transmitted. GSE 2 is also interrogated at this time to determine if the present word constitutes the last one in the information transfer. If not, the word is stored in memory and the memory address is incremented.

#### 8.1.3 Tape Load Startup Checksum

To provide assurance that the soft portion of the program has been loaded correctly, a tape load checksum routine is included for execution when the BTME is used to load the program.

Memory cells  $0026_8$  to  $0066_8$  contain the startup checksum routine which sums the contents of memory between cells  $0206_8$  and  $3777_8$ . If the sum is equal to zero, the program will transfer to EXIT 40 (Figure 8.7). If the sum is nonzero, the program will set  $S_{12}$  to 5 and turn on the Test Mode Failure indicator. The program then inhibits the resetting of the S switches and exits to block SINITC (Figure 8.7). The startup checksum routine is destroyed after use.

When the program is loaded by the ACE, the AGS load and verify routine provides verification of the load.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>
$\xi_3$	*	= { 1 Word has been transmitted from GSE to AEA 0 Otherwise (also called GSE1)
$\xi_4$	*	= { 1 Word being transmitted is the last word 0 Otherwise (also called GSE2)
GSE5	*	= { 1 AEA is ready for the next word from GSE 0 Otherwise

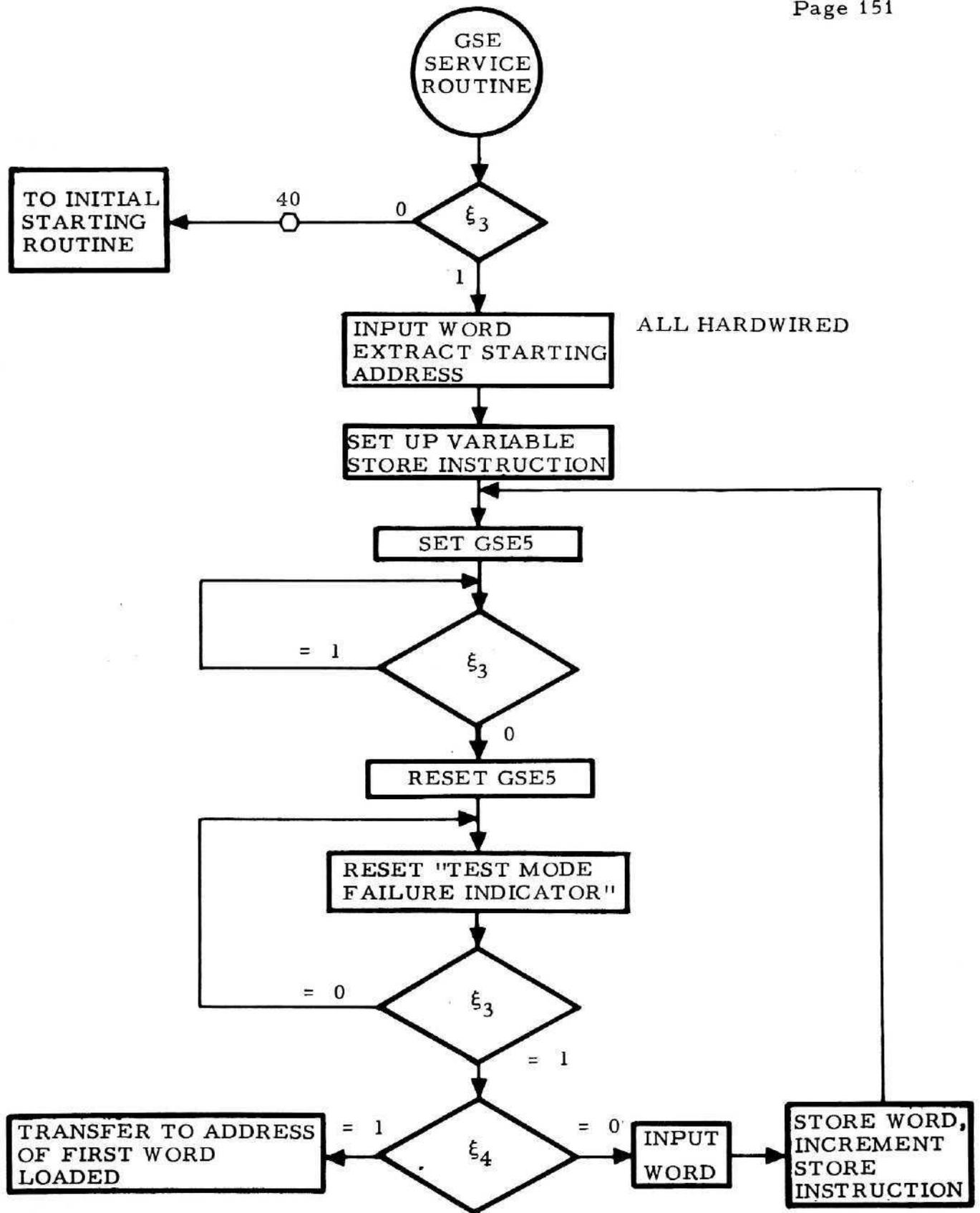


Figure 8.1. GSE (Ground Support Equipment) Service Routine

## 8.2 LM AGS TELEMETRY

The AGS downlist is comprised of a 50-word block of data which is repeated every second (one word is normally output every 20 msec provided a telemetry stop pulse  $\xi_2$  has been sensed). Since the major cycle time of the AGS computations is 2 sec, the full telemetry list will be transmitted twice per computer cycle. Thus, many of the words are redundant during the second half of a cycle and can be used for data checking. At the end of each second, the program initializes the output routine to cause the next word output to be the first word of the block. At the same time, the current values of six direction cosines, the ullage counter ( $\mu_g$ ), and the self test indicator ( $S_{12}$ ) are stored for later output.

Each output word is comprised of 24 bits, a 6-bit identification code followed by an 18-bit AEA computer word. The identification code is a binary count which represents the sequential position of the word in the list ( $01_8$  to  $62_8$ ). The computer word contains either codes or numeric data. The codes are described in Tables 8.1 to 8.4. The numeric data are in two's complement binary format. The most significant bit (Bit 0) is a sign bit, with 0 for positive and 1 for negative data. The remaining bits are numbered 1-17, from most significant to least significant, respectively.

For identification purposes, the 20-msec segments of the 2-sec cycle will be referred to as cycles 1 through 100. The first word on the list ( $\delta_{41}$ ) will be output at cycles 1 and 51.

The most significant half of AGS absolute time (Word 43) is updated by 1 count each 2 sec, during cycle 2. The change in this word can, therefore, be used to identify the first block of a compute cycle. This first block will be designated Block A and the second as Block B in the following discussion.

### 8.2.1 Description of Telemetered Quantities

The following list gives the word number, identification code, equation symbol, and time applicability of all words on the current Flight Program 6 Downlink.

Unless otherwise indicated, the word is current at the time of output.

Word 1                      Octal Code 01                       $\delta_{41}$

$\delta_{41}$  is the DEDA routine readout mode flag. A one will exist in the sign bit whenever the AEA is outputting a word to the DEDA. The changes of  $\delta_{41}$  and  $\delta_{38}$  (Word 3) can be used to deduce all operator DEDA actions.  $\delta_{41}$  is set to one when depression of the DEDA Readout pushbutton is sensed.  $\delta_{41}$  is reset to zero when depression of either the Clear or Hold pushbutton is sensed. If  $\delta_{38}$  is set to one, the Clear pushbutton has been depressed. This word as well as Words 2, 3 and 7 can change at any time.

Word 2                      Octal Code 02                      DD

DD is the most recent computer word entered or read out via DEDA. The parameter being processed can be determined from its computer memory address (Word 7). When DEDA is first cycled from the clear mode to the readout mode, DD can be zero, but ADST (Word 7) can contain the address of the desired readout. DD does not contain the data because the initial DEDA readout takes 1 sec (the address input takes the first half second and the data stored in DD and the output take the other half second).

Word 3                      Octal Code 03                       $\delta_{38}$

$\delta_{38}$  is the DEDA routine clear mode flag. A one will exist in the sign bit after the AEA has detected depression of the Clear pushbutton.  $\delta_{38}$  will remain in the one state until either the Enter or Readout pushbutton is depressed. If Readout is depressed,  $\delta_{41}$  will be set to one at the time  $\delta_{38}$  is reset to zero. If Enter is depressed,  $\delta_{41}$  will be unchanged and the word entered and its address will appear in Words 2 and 7. If an operator error is detected when either Readout or Enter is depressed, Words 2 and 7 will be reset to zero; Word 1 will remain unchanged.

Word 4                      Octal Code 04                       $a_{11}$

Word 5                      Octal Code 05                       $a_{12}$

Word 6                      Octal Code 06                       $a_{13}$

These three words are the first row of the orthonormal direction cosine matrix relating the LM body axes with the AGS inertial coordinate axes. They are scaled at binary 1. The Block A outputs are from the previous cycle 100 and the Block B outputs are from cycle 50.

Word 7                      Octal Code 07                      ADST

The least significant nine bits of this word are the address associated with Word 2 on the list. They are normally grouped as three octal digits. If an illegal entry (ADST < 0026 or ADST > 704) is attempted, Words 1, 2, and 3 will be zeroed.

Word 8                      Octal Code 10                       $a_{31}$

Word 9                      Octal Code 11                       $a_{32}$

Word 10                      Octal Code 12                       $a_{33}$

These three words are the third row of the direction cosine matrix. Their scaling and time significance are the same as for Words 4 to 6.

Word 11                      Octal Code 13                      h

This word is the LM altitude above the nominal lunar radius. It is scaled at binary 23 in units of feet. This word and all other LM and CSM navigation quantities on the list are computed for the time at the start of the 2-sec cycle, which is given by Words 23 and 43.

Word 12                      Octal Code 14                       $r_x$

Word 13                      Octal Code 15                       $r_y$

Word 14                      Octal Code 16                       $r_z$

These three words are the current components of the LM position vector in AGS coordinates, scaled at binary 23 in units of feet.

Word 15

Octal Code 17

EONS10

This word is a combination of the engine command indicator and the guidance mode indicator. If the AEA commands engine ON, the sign bit will be a one. Bits 1 to 3 give the state of  $S_{10}$ , the guidance mode selector. The remaining bits are always zeroes. Table 8.1 gives the  $S_{10}$  codes corresponding to the different modes.

Table 8.1 Bit Pattern, Guidance Selector  $S_{10}$ 

Mode	AEA Bit Pattern				
	0	1	2	3	4-17
Orbit Insertion (OI)	0	0	0	0	0
Coelliptic Sequence Initiate (CSI)	0	0	0	1	0
Constant Differential Altitude (CDH)	0	0	1	0	0
TPI Search	0	0	1	1	0
TPI Execute	0	1	0	0	0
External $\Delta V$ (XDV)	0	1	0	1	0

Word 16

Octal Code 20

 $r_{cx}$ 

Word 17

Octal Code 21

 $r_{cy}$ 

Word 18

Octal Code 22

 $r_{cz}$ 

These three words are the current components of the CSM position vector in AGS coordinates, scaled at binary 23 in units of feet.

Word 19

Octal Code 23

 $r_f$ 

$r_f$  is the predicted value of the LM radial distance at time  $t_{ig}$  in CSI, CDH, and TPI. In orbit insertion,  $r_f$  is the predicted LM radial distance at the completion of OI. The Block A output is redundant with the previous Block B output. The output has no meaning in the external  $\Delta V$  mode.  $r_f$  is scaled at binary 23 in units of feet.

Word 20	Octal Code 24	$\Delta V_X$
Word 21	Octal Code 25	$\Delta V_Y$
Word 22	Octal Code 26	$\Delta V_Z$

These three words are the compensated incremental velocity changes along the respective body axes, sensed by the accelerometers in the 20-msec cycle prior to their output. They are scaled at binary 1 in units of feet per second.

Word 23	Octal Code 27	$t_2$
---------	---------------	-------

The least significant word of the double precision AGS absolute time is  $t_2$ . It is scaled at binary 1 in units of seconds. It is changed only by an absolute time initialize and is always positive. Refer to Word 43 for a further discussion of AGS time.

Words 24	Octal Code 30	$\Delta a_X$
Words 25	Octal Code 31	$\Delta a_Y$
Words 26	Octal Code 32	$\Delta a_Z$

These three words are the compensated incremental components of vehicle rotation about the body axes, sensed by the gyros in the 20-msec cycle prior to their output. They are scaled at binary -6 in units of radians and are quantized at  $2^{-16}$  by the computer. The least significant seven bits are always zeroes except during the lunar align submode when the alignment increments which are quantized at  $2^{-23}$  are included in the words. In the PGNC/AGS align or body axis align submode, the inputs are not processed and the raw gyro counts, at binary 11, are output.

Word 27	Octal Code 33	$T_B$
---------	---------------	-------

$T_B$  is the predicted time to engine cutoff in the orbit insertion guidance mode. It is scaled at binary 9 in units of seconds. When in a nonthrusting phase, the computation of  $T_B$  is based on a limited value of thrust acceleration. The Block A output is redundant with the previous Block B output.  $T_B$  has no meaning in the other guidance modes.

Word 28	Octal Code 34	$V_x$
Word 29	Octal Code 35	$V_y$
Word 30	Octal Code 36	$V_z$

These three words are the current components of the LM inertial velocity in AGS coordinates, scaled at binary 13 in units of feet per second.

Word 31	Octal Code 37	$\mu_{8T}/S_{12T}$
---------	---------------	--------------------

This output is the sum of the computer self-test status indicator  $S_{12}$  and the ullage counter  $\mu_8$ .  $\mu_8$  is the number of consecutive 2-sec cycles in which the sensed X-body axis thrust acceleration has exceeded a threshold value. It is scaled at binary 17 in counts. Refer to Table 8.2 for the  $S_{12}$  codes. The Block A value is stored at the previous cycle 100 and the Block B value is stored at the previous cycle 50.

Table 8.2. Bit Pattern, Self-Test Indicator  $S_{12}$

Mode	AEA Bit Pattern				
	0	1	2	3	4-17
Entry-Initialize Test, Readout-Test Not Completed	0	0	0	0	0
Test Successfully Completed	0	0	0	1	0
Logic Test Failure	0	0	1	1	0
Memory Test Failure	0	1	0	0	0
Logic and Memory Test Failure	0	1	1	1	0

Word 32	Octal Code 40	$V_{cx}$
Word 33	Octal Code 41	$V_{cy}$
Word 34	Octal Code 42	$V_{cz}$

These three words are the current components of CSM inertial velocity in AGS coordinates, scaled at binary 13 in units of feet per second.

Word 35                      Octal Code 43                       $\dot{h}$

The current LM altitude rate is  $\dot{h}$ , scaled at binary 13 in units of feet per second.

Word 36                      Octal Code 44                       $V_G$

$V_G$  is the magnitude of the velocity-to-be-gained vector for the existing guidance mode. It is scaled at binary 13 in units of feet per second. During the CSI or TPI modes, the Block A output is redundant with the previous Block B output. In the orbit insertion mode,  $V_G$  will be set to a large value prior to the time of Block B output if the combination of altitude and altitude rate does not indicate a safe condition. Otherwise, during the orbit insertion mode, the Block B value is redundant with the Block A value. During CDH, this quantity is not valid.

Word 37                      Octal Code 45                       $V_{P0}/V_T$

This output is dependent upon the guidance mode. The output is scaled at binary 13 in units of feet per second.  $V_{P0}$ , the predicted value of the CDH burn magnitude, is output during the CSI mode.  $V_T$ , the total predicted velocity required to rendezvous, is output during the TPI mode. In both of the modes, the Block A value is redundant with the previous Block B value. In all other guidance modes, the word is unused.

Word 38                      Octal Code 46                       $T_{A0}$

$T_{A0}$  is the predicted time between the CSI maneuver and the CDH maneuver, output during the CSI mode. The word has no meaning in any other guidance mode.  $T_{A0}$  is scaled at binary 18 in units of seconds. The Block A output is redundant with the previous Block B output.

Word 39                      Octal Code 47                       $t_{ig}$

$t_{ig}$  is the absolute time of the next maneuver. It is scaled at binary 18 in units of seconds. It has no meaning in external  $\Delta V$  or orbit insertion.

Word 40	Octal Code 50	$a_{11bD}$
Word 41	Octal Code 51	$a_{12bD}$
Word 42	Octal Code 52	$a_{13bD}$

These three words are the components of the unit vector defining the guidance desired pointing direction for the LM thrust axis.

They are scaled at binary 1. For the external  $\Delta V$  and orbit insertion modes, the Block B output is redundant with the Block A output. For the CSI and TPI modes, the Block A output is redundant with the previous Block B output. During CDH, these quantities are not valid.

Word 43	Octal Code 53	t
---------	---------------	---

The most significant part of the double precision AGS absolute time is t. It is scaled at binary 18 in units of seconds. The sum of t and  $t_2$  is the AGS absolute time at which all navigation values for the 2-sec cycle are valid. t is set by an absolute time initialize and is incremented by one count (2-sec) at the start of each guidance cycle.

Word 44	Octal Code 54	$S_{00}$
---------	---------------	----------

$S_{00}$  is the function selector which controls submode logic. Refer to Table 8.3 for a description of its codes.

Table 8.3 Bit Pattern, Mode Word  $S_{00}$

Mode	AEA Bit Pattern				
	0	1	2	3	4-17
Attitude Hold	0	0	0	0	0
Guidance Steering	0	0	0	1	0
Z Axis Steering	0	0	1	0	0
PGNCS/AGS Align	0	0	1	1	0
Lunar Align	0	1	0	0	0
Body Axis Align	0	1	0	1	0
Calibrate	0	1	1	0	0
Accelerometer Calibrate	0	1	1	1	0

Word 45

Octal Code 55

DISC1C

DISC1C is the Discrete Word No. 1 input as seen by the computer. A zero indicates the ON condition and a one the OFF condition. Refer to Table 8.4 for the bit structure of the word.

Table 8.4 Bit Pattern, Discrete Input Word Number One

Bit No.	Discrete
1	Downlink Telemetry Stop ( $\xi_1$ )
2	Output Telemetry Stop ( $\xi_2$ )
3	Follow-Up ( $\beta_3$ )
4	Automatic ( $\beta_4$ )
5	Descent Engine ON ( $\beta_1$ )
6	Ascent Engine ON ( $\beta_2$ )
7	Abort ( $\beta_6$ )
8	Abort Stage ( $\beta_5$ )

Word 46

Octal Code 56

 $\Delta r/q_{1d}$ 

This output is dependent upon the guidance mode. It is scaled at binary 23 in units of feet.  $\Delta r$ , the predicted differential altitude between the LM and CSM orbits after the coelliptic CDH burn, is output during both the CSI and CDH modes.  $q_{1d}$ , the perifocus altitude of the computed transfer orbit, is output during the TPI mode. The Block A value is redundant with the previous Block B output. In the other guidance modes the word is unused.

Word 47

Octal Code 57

 $q_{LT}$ 

The perifocus altitude of the present LM orbit is  $q_{LT}$ . It is scaled at binary 23 in units of feet.

Word 48	Octal Code 60	$V_{dX}$
Word 49	Octal Code 61	$V_{dY}$
Word 50	Octal Code 62	$V_{dZ}$

These three words are the sums of the sensed velocity increments along the body axes. They are scaled at binary 13 in units of feet per second. On the lunar surface, all three are set to zero. Whenever computer power is turned on,  $V_{dX}$  is set to a large negative value ( $K_{27}^4$ ).

In the TPI modes, overflow may be detected during the p-iterator computations in which case no solution is possible. In that situation  $V_T$  will be set to  $K_{11}^2$  and Words 19, 36, and 46 will be invalid.

#### 8.2.2 Program Logic for Telemetry

A schematic representation of the telemetry logic is shown in Figure 8.2.

#### 8.2.3 Telemetry List for Flight Program 6

Table 8.5 gives the telemetry word list, its sequence, identification code, equation symbol, scaling, units, and the definition of each word.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\mu_8$	MU8	Ullage counter	0 to $2^{17}$	1
$S_{12}$	S12	Self-test flag	0 to 7	
$\xi_2$	*	= $\begin{cases} 1 & \text{Telemetry output word requested} \\ 0 & \text{Otherwise} \end{cases}$		
$\mu_{8T}/S_{12T}$	MU8S12	Combination of the ullage counter and the self-test flag stored for telemetry		

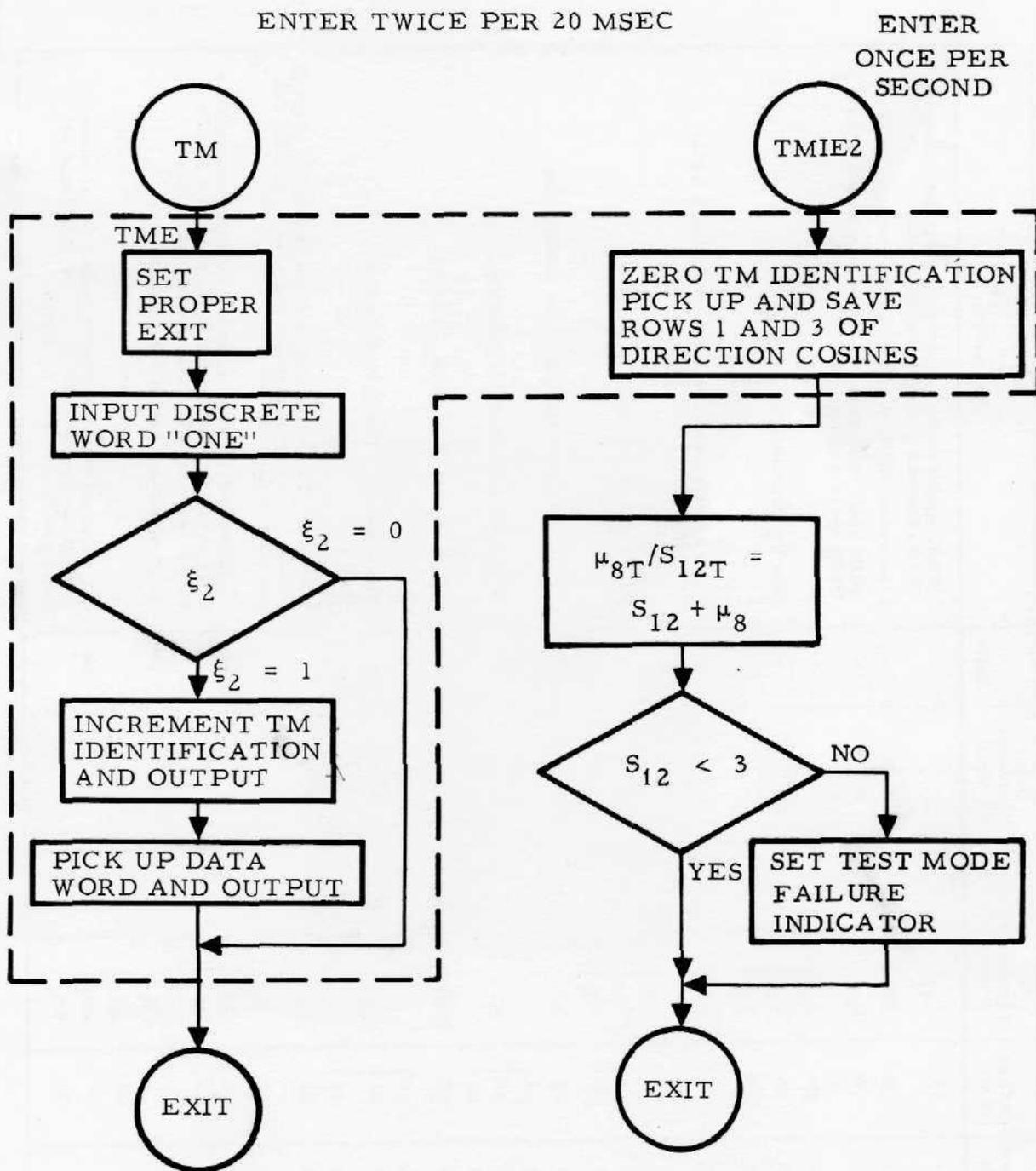


Figure 8.2. AGS Telemetry Subroutines

Table 8.5. AEA Telemetry Word List

Word No.	ID Code (Octal)	Equation Symbol	Binary Scaling	Least Significant Bit Weight	Units	Description
1	01	$\delta_{41}$	-	-		DEDA readout mode flag; a 1 in the sign bit indicates DEDA processing is in the readout mode
2	02	DD	-	-	-	Most recent DEDA data word in the computer units
3	03	$\delta_{38}$	-	-	-	DEDA clear mode flag; a 1 in the sign bit indicates DEDA processing is in the clear mode
4	04	$\left. \begin{array}{l} a_{11} \\ a_{12} \\ a_{13} \end{array} \right\}$	1	$2^{-16}$	-	Row 1 of direction cosine matrix
5	05					
6	06					
7	07	ADST	-	-	-	Octal address associated with most recent DEDA communication
8	10	$\left. \begin{array}{l} a_{31} \\ a_{32} \\ a_{33} \end{array} \right\}$	1	$2^{-16}$	-	Row 3 of direction cosine matrix
9	11					
10	12					
11	13	h	23	$2^6$	ft	LM altitude above nominal lunar landing site
12	14	$\underline{r}$	23	$2^6$	ft	Components of LM inertial position
13	15					
14	16					
15	17	EONS10	-	-	-	Engine On indicator and $S_{10}$ switch
16	20	$\underline{r}_c$	23	$2^6$	ft	Components of CSM inertial position
17	21					
18	22					
19	23	$r_f$	23	$2^6$	ft	Predicted value of LM radius at the completion of the OI guidance mode; predicted LM radius at $t_{ig}$ in TPI, CSI, and CDH; unused in XDV
20	24	$\left. \begin{array}{l} \Delta V_x \\ \Delta V_y \\ \Delta V_z \end{array} \right\}$	1	$2^{-16}$	ft/sec	Compensated incremental velocity components accumulated per 20 msec along the X, Y, or Z body axis by the corresponding X, Y, or Z accelerometer
21	25					
22	26					
23	27	$t_2$	1	$2^{-16}$	sec	Least significant half of AGS absolute time
24	30	$\left. \begin{array}{l} \Delta \alpha_x \\ \Delta \alpha_y \\ \Delta \alpha_z \end{array} \right\}$	-6	$2^{-16}$	rad	Compensated incremental components per 20 msec of vehicle rotation about the X, Y, or Z body axis as measured by the X, Y, or Z gyro
25	31					
26	32					

Table 8.5. AEA Telemetry Word List (Continued)

Word No.	ID Code (Octal)	Equation Symbol	Binary Scaling	Least Significant Bit Weight	Units	Description
27	33	$T_B$	9	$2^{-8}$	sec	Time to LM engine burnout in OI guidance mode, unused in all other modes
28	34	$\underline{V}$	13	$2^{-4}$	ft/sec	Components of present LM inertial velocity vector
29	35					
30	36					
31	37	$\mu_{8T}/S_{12T}$	-	-	-	Ullage counter and self-test status for telemetry
32	40	$\underline{V}_c$	13	$2^{-4}$	ft/sec	Components of CSM inertial velocity vector
33	41					
34	42					
35	43	$\dot{h}$	13	$2^{-4}$	ft/sec	LM altitude rate
36	44	$V_G$	13	$2^{-4}$	ft/sec	Magnitude of the velocity-to-be-gained, not valid in CDH
37	45	$\{V_{P0}$	$\{13$	$\{2^{-4}$	$\{ft/sec$	$\{Predicted velocity-to-be-gained in CDH maneuver (CSI)$
		$\{V_T$	$\{13$	$\{2^{-4}$	$\{ft/sec$	$\{Total velocity to rendezvous in TPI$
38	46	$T_{A0}$	18	2	sec	Time from nominal CSI maneuver to CDH maneuver (CSI)
39	47	$t_{ig}$	18	2	sec	Absolute time of next maneuver in TPI, CSI, or CDH only
40	50	$\underline{X}_{bD}$	1	$2^{-16}$	-	Components along the X, Y, and Z inertial axes of unit vector commanding desired pointing direction for LM thrust axis, not valid in CDH
41	51					
42	52					
43	53	$t$	18	$2^1$	sec	Most significant half of AGS absolute time
44	54	$S_{00}$	-	-	-	Function selector by which submode logic is selected via DEDA
45	55	$\beta's$	-	-	-	Discretes which make up "Discrete Word One"
46	56	$\{\Delta r$	$\{23$	$\{2^6$	$\{ft$	$\{Differential LM-CSM altitude after CDH maneuver in CSI or CDH$
		$\{q_{1d}$	$\{23$	$\{2^6$	$\{ft$	$\{Transfer orbit perifocus altitude in TPI$
47	57	$q_{LT}$	23	$2^6$	ft	Perifocus altitude of present LM orbit
48	60	$\underline{V}_d$	13	$2^{-4}$	ft/sec	Sensed velocity increments along LM body axes
49	61					
50	62					

### 8.3 INFLIGHT SELF-TEST ROUTINE

The inflight self-test routine checks on a continuous basis that the computer's logic and memory are functioning properly. A special DEDA selector,  $S_{12}$ , is used in conjunction with the test to indicate test results to the astronaut and to allow resetting of error indications. The test is mechanized in such a manner as to utilize all unused 2-sec guidance program branches, but it runs asynchronous with the basic 2-sec cycle. The duration of a single pass through the test, which depends on the number of guidance program branches available per 2 sec, is given in Table 8.6.

Table 8.6 Time Required to Complete Self-Test Following the Entry of  $S_{12} = 0$  via DEDA

Guidance Mode	Time (sec)
Orbit insertion	11.2
CSI	12.4
CDH	12.4
TPI	22.0
External Delta V	9.6

The  $S_{12}$  selector has the following states:

- 0 Entry-initialize test, Readout-test not complete
- 1 Test successfully completed
- 3 Logic test failure
- 4 Memory test failure
- 7 Logic and memory test failure

State 0 is used to allow the astronaut to reinitiate testing and reset the Test Mode Failure indicator after any failure is detected. Whenever a failure is encountered during a pass through the test,  $S_{12}$  and the test mode failure indicator are set accordingly and future passes through the test are inhibited until astronaut reinitiation.

### 8.3.1 Test Operation

Before making a pass through the test routines, the test status word is examined. If it contains any failure indication, the tests are bypassed and the computer returns to the guidance program. Otherwise, the logic test is executed. If a failure is indicated by this routine, the successful completion indication is reset and the logic test failure state is set into the test status word. Successful completion of the logic test will result in no change to the test status word. Upon completion of the logic test, the memory test is executed. If the memory test is successfully executed, the test status word is examined to determine whether or not it is equal to zero. If it is equal to zero, the successful completion indication bit is inserted into the test status word. If the test status word is not equal to zero, no change is made in it. An incorrect result in the memory test will result in the resetting of the successful completion indication bit and setting of the memory test failure indicator in the test status word.

### 8.3.2 Logic Test

This routine checks the computer's ability to execute its instructions and verifies its computational capability. All instructions except those associated with input/output operations are tested. Instructions are executed and the results are used as initial conditions for the checkout of additional instructions. This interweaving of the test provides a test result that propagates through the test until it is verified at some convenient location. All the internal logic of the computer is verified in this manner.

### 8.3.3 Memory Test

The memory test performs a checksum operation on the AEA's permanent memory. The permanent memory is defined as all of the hardwired cells plus a certain portion of the erasable memory. The erasable portion is comprised of the memory cells between octal addresses 1005 and 3777.

The program sequentially sums all the previously defined locations including a predetermined checksum constant. It is then verified that the summation is equal to zero. This checks that each location can be individually accessed and that each one contains the correct configuration of ones and zeros.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>
$S_{12}$	S12	= { <ul style="list-style-type: none"> <li>0 Entry - initialize test, Readout - test not completed</li> <li>1 Test successfully completed</li> <li>3 Logic test failure</li> <li>4 Memory test failure</li> <li>7 Logic and memory test failures</li> </ul>

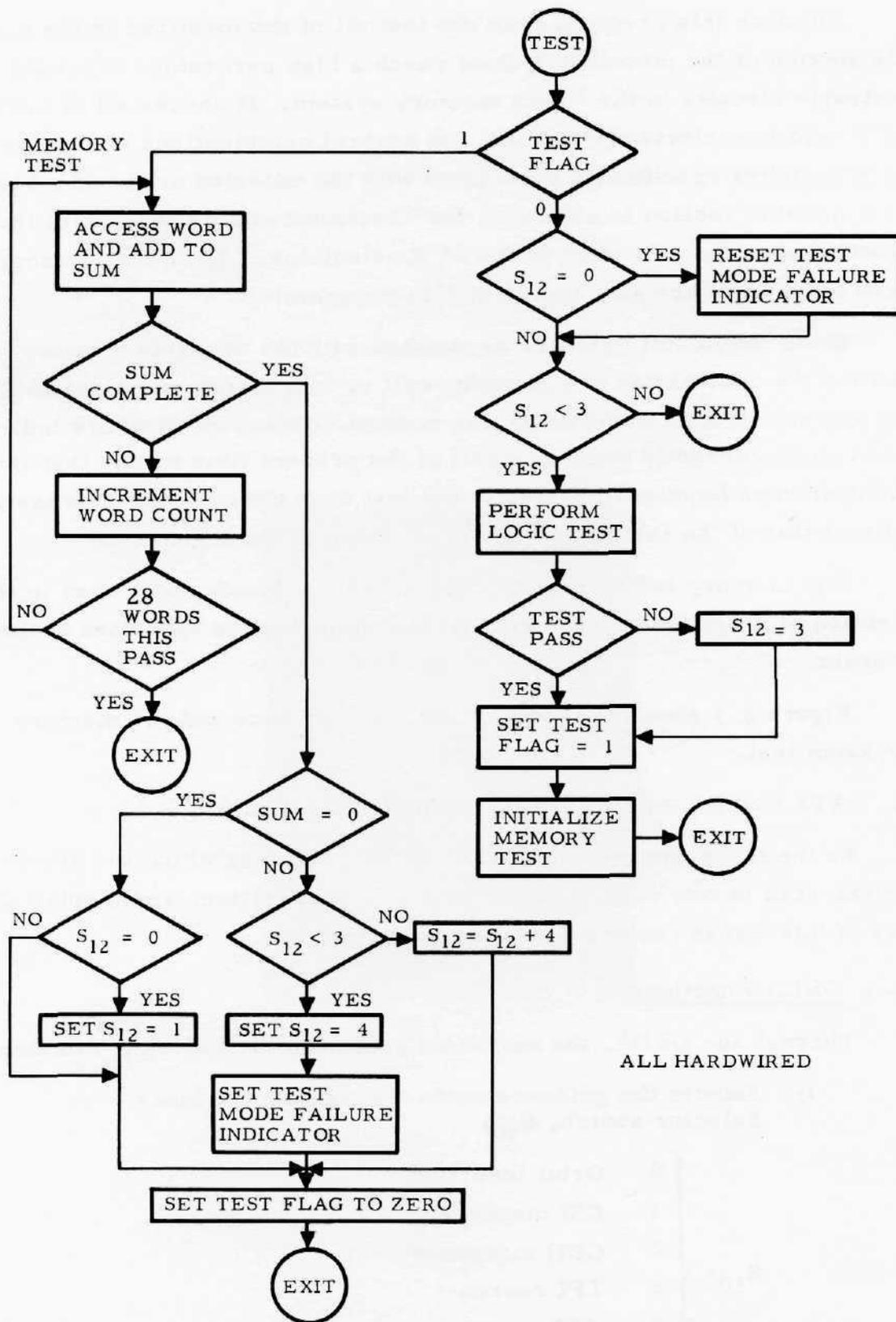


Figure 8. 3. Inflight Self-Test Routine

Although this program does not test all of the locations in the erasable section of the memory, it does check a high percentage of all the electronic circuits in the entire memory system. It checks all of the X and Y selection electronics except the several combinations of X drivers and X switches specifically associated with the untested area. If a 512-word untested section is assumed, the checksum would check all of the Y selection electronics and 56 of the 64 X selections. All of the memory sense amplifiers are also tested by this program.

Since the inhibit drivers, associated with the erasable memory, are used for the restoration of a memory cell or in a direct store operation, they are not directly checked by this routine. However, they are indirectly tested since correctly reading a cell at the present time means that the inhibit drivers functioned perfectly the last time the cell was addressed. A direct test of the inhibit drivers is provided in the logic test.

The memory buffer register and all of the decode logic used in the selection of X and Y drivers are also tested during the execution of the program.

Figure 8.3 shows the logic of the test sequence and the memory checksum test.

#### 8.4 DATA ENTRY AND DISPLAY ASSEMBLY (DEDA)

As the name implies, the DEDA is the means by which the astronaut can read in new data, change the mode of operation, and display quantities of interest in certain computer locations.

##### 8.4.1 DEDA Functions

Through the DEDA, the astronaut performs the following functions:

- 1) Selects the guidance mode through the Guidance Selector switch,  $S_{10}$ ,

$S_{10} =$	{	0	Orbit insertion
		1	CSI maneuver
		2	CDH maneuver
		3	TPI search
		4	TPI execute
		5	External $\Delta V$

## 2) Controls certain guidance options

a) CDH Selection

$$S_{16} = \begin{cases} 1 & \text{Compute CDH solution at CSI time plus 1/2 LM} \\ & \text{orbit period} \\ 3 & \text{Compute CDH solution at CSI time plus 3/2 LM orbit} \\ & \text{period} \end{cases}$$

b) Engine Select

$$S_{11} = \begin{cases} 1 & \text{Select cant angle corrections to guidance steering} \\ & \text{attitude errors} \\ 0 & \text{Do not select cant angle corrections} \end{cases}$$

c) Store Landing Azimuth and Set Lunar Surface Flag,  $\delta_{21}$ 

$$S_{13} = \begin{cases} \text{Any entry into } S_{13} \text{ will perform function} \end{cases}$$

d) Select Radar Filter Initialize

$$S_{17} = \begin{cases} 0 & \text{Normal Readout} \\ 1 & \text{Entered to initialize radar filter} \end{cases}$$

e) Sets External  $\Delta V$  Reference

$$S_{07} = \begin{cases} 0 & \text{Do not freeze velocity-to-be-gained} \\ & \text{vector or accumulate sensed velocity} \\ & \text{increments} \\ 1 & \text{Freeze velocity-to-be-gained vector in} \\ & \text{AGS inertial frame and accumulate sensed} \\ & \text{velocity increments} \end{cases}$$

f) Radar

$$S_{15} = \begin{cases} \text{Any entry into } S_{15} \text{ will store LM Z-axis} \\ & \text{direction cosines, computed range vector,} \\ & \text{range rate, and the time since the last radar} \\ & \text{range update} \end{cases}$$

## 3) Selects submode logic via the Function Selector Switch

$S_{00} =$	{	0	Attitude hold	}	<b>Inertial Reference Mode</b>	
		1	Guidance steering			
		2	Z Axis steering			
		}	}	3	PGNCS/AGS align	<b>Align Mode</b>
				4	Lunar align	
				5	Body axis align	
				6	Calibrate	
7	Accelerometer calibrate only					

## 4) Initializes Navigation

$S_{14} =$	{	0	Initialization complete
		1	Initialize LM and CSM using PGNCS downlink data
		2	Initialize LM using DEDA data
		3	Initialize CSM using DEDA data

## 5) Inputs certain data (see Section 9 for DEDA Input Parameters)

## 6) Reads out certain data for display (see Section 9 for DEDA output parameters)

## 7) Reads out the results of the inflight self-test and resets the error indications as described in Section 8.3

$S_{12} =$	{	0	Entry- reinitialize test, Readout test not completed
		1	Test successfully completed
		3	Logic test failure
		4	Memory test failure
		7	Logic and memory test failure

8) Selects yaw steering when in guidance steering ( $S_{00} = 1$ )

$S_{623} =$	{	1	steer Z body axis parallel to plane defined by $\underline{W}_b$
		0	steer Z body axis parallel to CSM orbit plane

- 9) Select acquisition or Z-axis steering when  $S_{00} = 2$

$$S_{507} = \begin{cases} 1 & \text{steer Z body axis in desired} \\ & \text{thrust direction} \\ 0 & \text{steer Z body axis in computed} \\ & \text{direction of CSM} \end{cases}$$

#### 8.4.2 PGNCS Downlink Input Routine

The downlink of the Primary Guidance Navigation and Control System (PGNCS) initializes or updates the LM and CSM positions and velocities during nonthrusting flight phases of the mission. The AGS has a command,  $S_{14} = 1$ , which is inserted via the DEDA and which gives a command to the AGS to extract the desired data from the PGNCS downlink. (A similar command must be issued to the primary system in order to have the correct data transmitted via the downlink.) This subroutine is shown schematically in Figure 8.4.

When  $S_{14}$  is set to 1, the AGS starts the following sequence of operations:

- 1) Flag  $\delta_{32}$  is set to 1 in Branch 50, allowing the AGS program logic to receive the PGNCS downlink input data.
- 2) The PGNCS downlink input subroutine searches the PGNCS downlink for the identification word which signals the start of a block of ephemeris data.
- 3) Upon finding the correct identification word, the subroutine reads in and stores in the downlink registers the next 16 words.
- 4) Upon completion of the entry of all 16 inputs,  $\delta_{32}$  is set to zero and the flag  $\delta_{31}$  is set to 1 and sensed in Branch 50.
- 5) The program is then transferred to the navigation initialize routines in branches IC1, IC2 and IC3 as shown in Figures 7.11 and 7.12.

One word is read into the input register each time a stop pulse ( $\xi_1$ ) is sensed. Note that the status of  $\xi_1$  is updated just prior to checking it by input of Discrete Word One.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\delta_{31}$	IDRF	= { 1 Downlink telemetry has been completed 0 Otherwise		
$\delta_{32}$	DLIF	= { 1 Downlink telemetry has been commanded 0 Otherwise		
$\delta_{60}$	IDIF	= { 1 Telemetry ID word has been transmitted 0 Otherwise		
$\mu_{30}$	DLWN	Downlink telemetry input word counter	0 to $2^{17}$	1
$\xi_1$	*	= { 1 Downlink word has been transmitted to AEA 0 Otherwise		

ALL HARDWIRED

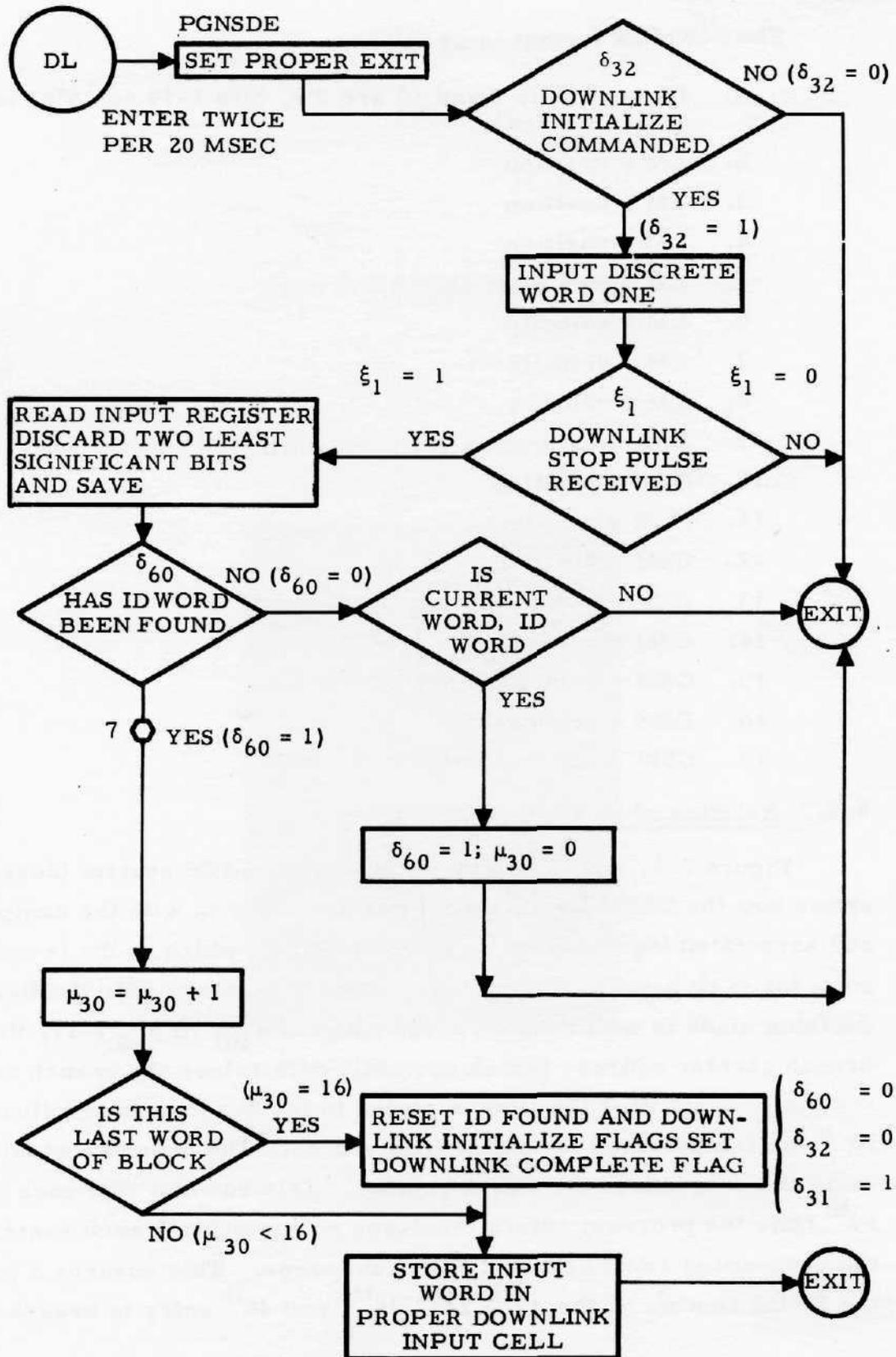


Figure 8. 4. PGNSCS Downlink Input Subroutine

The Downlink format is as follows:

1. ID word (bits 0 and 15 are 0's, bits 1-14 are 1's, bits 16 and 17 are unused)
2. LM x position
3. LM y position
4. LM z position
5. LM epoch most significant word
6. LM x velocity
7. LM y velocity
8. LM z velocity
9. LM epoch least significant word
10. CSM x position
11. CSM y position
12. CSM z position
13. CSM epoch most significant word
14. CSM x velocity
15. CSM y velocity
16. CSM z velocity
17. CSM epoch least significant word

#### 8.4.3 Relation of DEDA to AGS System

Figure 7.1, containing the total software AGS system block diagram, shows how the DEDA inputs or outputs are fitted in with the computations and associated logic. Starting at Point EXEC, which is the terminating point for each branch computation (except for Branch 50), the first logical decision made is with respect to the count of  $\mu_{20}$ . If  $\mu_{20} = 11$ , the present branch counter address (which normally determines the branch control routing the next time the flow is routed to the 2-sec branch following the 20 msec computation) is temporarily stored. The branch control is set to route the program to the DEDA routine. This ensures that each 12<sup>th</sup> or 14<sup>th</sup> time the program enters the 2-sec computation branch control point, the program is routed to the DEDA subroutine. This ensures a pass through the DEDA routine on the 12<sup>th</sup>, 24<sup>th</sup>, 36<sup>th</sup>, and 48<sup>th</sup> entry to branch control. Since

there are 50 branches in the 2-sec computation, the DEDA readout might be expected to be on cycle 10 of the next 2-sec computation. This situation is prevented by resetting the DEDA counter  $\mu_{20}$  at the end of the branch 1 computations. This means that the time between passes through the DEDA routine is  $(12 \times 40 \text{ msec}) = 0.48 \text{ sec}$  75 percent of the time and  $(14 \times 40 \text{ msec}) = 0.56 \text{ sec}$  25 percent of the time. Figure 8.5 details the DEDA routine and the logic and operations contained therein.

Data that is being displayed on the DEDA is seen to be updated approximately every half second. However, data generated by the 2-sec computations will change every 2 sec only. DEDA inputs are processed every 2 sec except for absolute time initialization (t) and  $S_{15}$ .

Contained in the 40-msec computation is the sampling of Discrete Word 2. This word contains, in addition to the GSE discrettes, the DEDA CLEAR, HOLD, READOUT and ENTER discrettes. Figure 8.6 shows the sampling and flag setting connected with these discrettes. The flags set are then utilized in the DEDA routine the next time it is entered.

All DEDA data transfers will start with  $\delta_{38}$  being set to one. For a data entry,  $\delta_{38}$  will be set to zero at the time the data is entered. In the case of a DEDA readout,  $\delta_{41}$  will go to 1 at the time  $\delta_{38}$  goes to zero. If DEDA hold mode is entered from readout,  $\delta_{41}$  will go to zero and  $\delta_{39}$  will be set to one. If readout is terminated,  $\delta_{41}$  will go to zero and  $\delta_{38}$  will go to one.

At the bottom of Figure 8.5, note the setting of flag  $\delta_{30}$ . This signifies that a new word has been entered from DEDA and, if necessary, scaled. Flag  $\delta_{30}$  is acted upon at entry into Branch 50 (see Figure 7.31). If  $\delta_{30} = 1$ ,  $\delta_{30}$  is set to zero, and the new word is stored in the addressed location.

Two inputs are processed immediately by the DEDA routine. When either absolute time or  $S_{15}$  is input (as determined by its respective address) special processing is performed.

If a new absolute time is input via DEDA, the program proceeds to the absolute time initialize routine, where 2 sec is subtracted from the time and this number is stored in the most significant part of absolute

time. The least significant part of absolute time equals 2 sec plus a time bias,  $\left(0.25 - \frac{\mu_{10}}{25}\right)\text{sec}$  ( $\mu_{10}$  may be equal 11, 23, 35, or 47), which synchronizes the time with the time update in Branch 1.

When  $S_{15}$  is entered, the Z body axis unit vector, the computed range vector and range rate are stored for use in the radar filter. Also, the time since the last radar range update is computed and stored.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\delta_{34}$	CLEARP	= $\begin{cases} 1 & \text{DEDA clear requested} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{35}$	HOLDP	= $\begin{cases} 1 & \text{DEDA hold requested} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{36}$	ENTERP	= $\begin{cases} 1 & \text{DEDA enter requested} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{37}$	READP	= $\begin{cases} 1 & \text{DEDA readout requested} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{38}$	CMF	= $\begin{cases} 1 & \text{DEDA clear mode set} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{39}$	HMF	= $\begin{cases} 1 & \text{DEDA hold mode set} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{41}$	RMF	= $\begin{cases} 1 & \text{DEDA readout mode set} \\ 0 & \text{Otherwise} \end{cases}$		
$\delta_{30}$	DDF	= $\begin{cases} 1 & \text{Word has been entered via DEDA} \\ 0 & \text{Otherwise} \end{cases}$		
$\mu_{20}$	DEDASC	DEDA shift counter used to schedule the DEDA routine on minor cycles 24, 48, 72, and 96 of each major cycle		
t	TA1, TA2	Time output from time initialize routine 0 to $2^{18}$	0 to $2^{18}$	$2^{-2}$
$S_{15}$	S15	Any entry commands the storing of LM Z axis direction cosines, computed range vector, range rate, and the time since the last radar range update		
$\underline{Z}_b^*$	A31S, A32S, A33S	Radar range direction cosines stored when an $S_{15}$ entry is made	$\pm 2^1$	$2^{-16}$
$\underline{Z}_b$	A31, A32, A33	Unit vector of LM Z body axis (see Figure 7.4 for definition of each component)	$\pm 2^1$	$2^{-16}$
$\underline{R}^*$	RRSX, RRSY, RRSZ	Calculated range at the time of a radar measurement	$\pm 2^{23}$	$2^6$
$\dot{\underline{R}}^*$	RDOTS	Calculated range rate at the time of a radar measurement	$\pm 2^{13}$	$2^{-4}$
$t_1$	T1	Absolute time of the last radar range input	$2^{18}$	$2^1$
$\Delta t$	DELTAT	Elapsed time since last radar range input	$2^{18}$	$2^1$
$\underline{R}$	RRDOT	Estimated range rate between CSM and LM (fps)	$\pm 2^{13}$	$2^{-4}$
$\underline{R}$	RRX, RRY, RRZ	Relative position vector of the CSM relative to the LM (ft)	$\pm 2^{23}$	$2^6$



<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>
$\xi_3$	*	= $\begin{cases} 1 & \text{Word transmitted from GSE to ASA} \\ 0 & \text{Otherwise} \end{cases}$ (also called GSE1)
$\xi_6$	*	= $\begin{cases} 1 & \text{DEDA clear requested} \\ 0 & \text{Otherwise} \end{cases}$
$\xi_7$	*	= $\begin{cases} 1 & \text{DEDA hold requested} \\ 0 & \text{Otherwise} \end{cases}$
$\xi_8$	*	= $\begin{cases} 1 & \text{DEDA enter requested} \\ 0 & \text{Otherwise} \end{cases}$
$\xi_9$	*	= $\begin{cases} 1 & \text{DEDA readout requested} \\ 0 & \text{Otherwise} \end{cases}$
$\delta_{34}$	CLEARP	= $\begin{cases} 1 & \text{DEDA clear requested} \\ 0 & \text{Otherwise} \end{cases}$
$\delta_{35}$	HOLDP	= $\begin{cases} 1 & \text{DEDA hold requested} \\ 0 & \text{Otherwise} \end{cases}$
$\delta_{36}$	ENTERP	= $\begin{cases} 1 & \text{DEDA enter requested} \\ 0 & \text{Otherwise} \end{cases}$
$\delta_{37}$	READP	= $\begin{cases} 1 & \text{DEDA readout requested} \\ 0 & \text{Otherwise} \end{cases}$

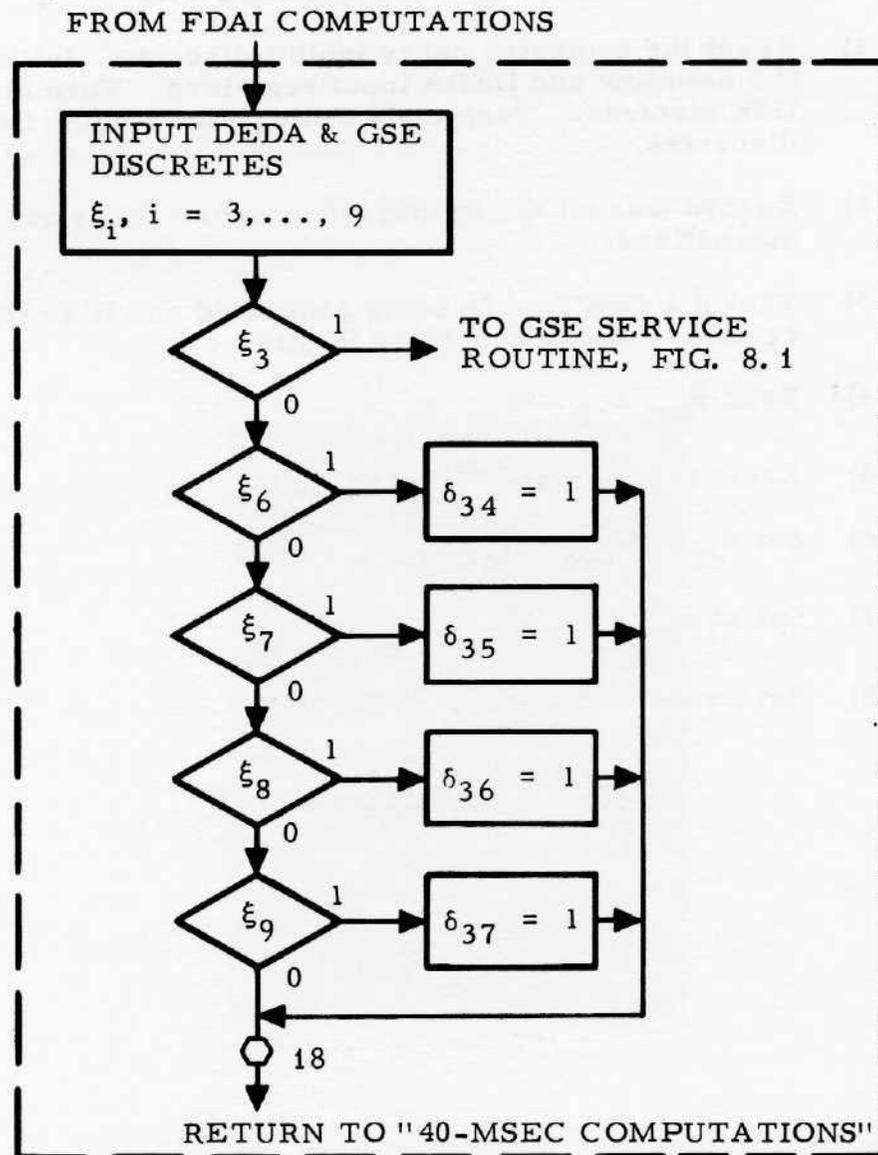


Figure 8.6. 40-msec Sampling of Discrete Word Two

## 8.5 POWER TURN ON SEQUENCE

Whenever the AGS Status Switch is set to OPERATE, turning computer power on, a startup sequence is initiated. This sequence is illustrated in Figure 8.7 and consists of the following steps:

- 1) Reset the computer carry inhibit discrete. Initialize the downlink and DEDA input registers. Turn off all GSE discrettes, Test Mode Failure and Engine On discrettes.
- 2) Ensure that all the hardwired memory cores are magnetized.
- 3) Test if a GSE load is being requested and if so transfer control to the GSE Service Routine.
- 4) Zero  $S_{00}$ ,  $S_{10} - S_{17}$ , and  $\delta_5$ .
- 5) Reset test flag used by self test.
- 6) Set  $V_{dX} = V_{DX} = K_{27}^4$ .
- 7) Set  $\mu_{10} = 0$ .
- 8) Set Branch Control for Branch 50.

<u>Equation Variable</u>	<u>Program Symbol</u>	<u>Definition</u>	<u>Range</u>	<u>Quantization</u>
$\xi_3$	*	= { 1 Word has been transmitted from GSE to AEA 0 Otherwise (also called GSE1)		
$\delta_5$	DEL5	= { 1 Lock attitude hold reference direction cosines 0 Set desired direction cosines equal to body direction cosines		
$S_{00}$	S00	Mode switch ( $S_{00} = 0$ is attitude hold)		
$S_{10}$	S10	Guidance mode switch ( $S_{10} = 0$ is orbit insertion)		
$S_{11}$	S11	= { 1 Select cant angle corrections to guidance steering attitude errors 0 Do not select cant angle corrections		
$S_{12}$	S12	Self test switch (0 reinitializes the test and resets the failure indicator)		
$S_{13}$	S13	Performs no function unless entered via DEDA		
$S_{14}$	S14	Navigation initialization switch ( $S_{14} = 0$ is initialization completed)		
$S_{15}$	S15	Performs no function unless entered via DEDA		
$S_{16}$	S16	CDH burn time selector switch ( $S_{16} = 0$ is not valid selection for the CSI logic)		
$S_{17}$	S17	= { 0 Performs no function 1 Initialize radar filter		
$V_{DX}$	VDX	Accumulated sensed velocity along X body axis, updated every 2 sec (fps)	$\pm 2^{13}$	$2^{-4}$
$V_{dX}$	VDIX	Accumulated sensed velocity along X body axis, updated every 40 ms (fps)	$\pm 2^{13}$	$2^{-4}$
$\mu_{10}$	MU10	2-sec cycle counter used to schedule output telemetry and executive branch (BR 50)	0 to 49	1

Constants:  $K_{27}^4$

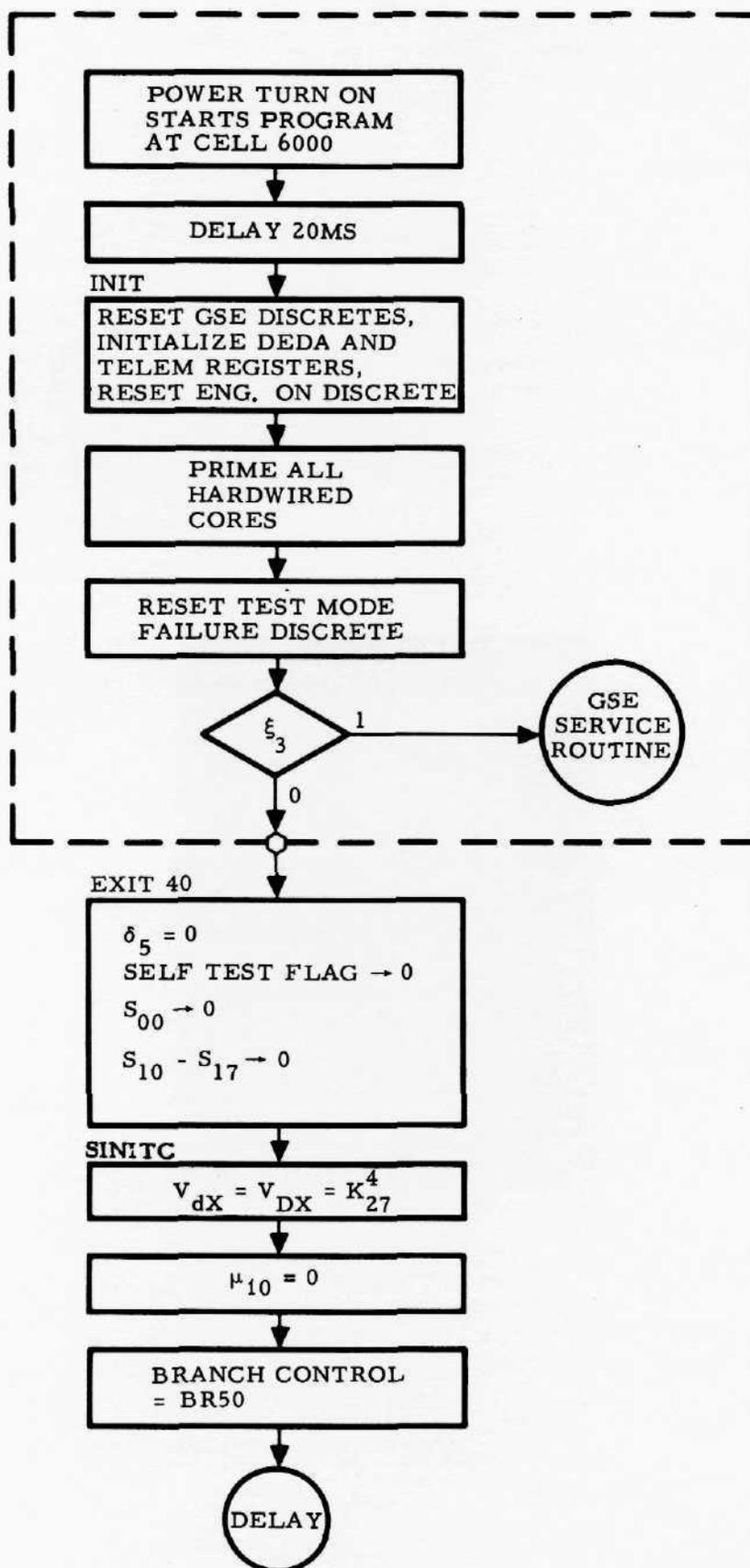


Figure 8.7. Power Turn-On Startup Sequence

## 9. DEDA ACCESSIBLE PARAMETERS

Constants and variables are entered via the DEDA in either octal or decimal form. Whether it is octal or decimal is determined by the program from the input address of the quantity. This address is tested to select a scale factor for decimal entries, which is used to convert the input quantity to its binary equivalent and proper dimensional units. For octal entries no scale factor is used.

Parameters output via the DEDA go through a reverse process before being displayed. Although the tables herewith presented separate inputs from outputs, this does not mean that only listed outputs can be displayed. The AEA storage area from 026-704 (octal) is DEDA accessible for both inputs and outputs, and Table 9.1 divides this region into subregions and shows the associated quantizations and scalings assumed by the program for making scale factor selections.

When data is entered or displayed through the DEDA, the numbers and their placement in the five digit data word determine the decimal or binary point, since the point is not displayed. For the data which is in decimal form, the least significant digit (LSD) is given in Tables 9.2 and 9.3. If a position variable has an LSD of 100 ft, then +42000 ft is entered or displayed as +00420.

The data which must be entered or displayed in octal form requires considerably more manual conversion to interpret in decimal form. These variables are identified in Table 9.2 by listing their binary scaling as opposed to the LSD for decimal variables. Octal data which is displayed may be converted to the decimal system by first converting to the DEDA binary word form, next taking the 2's complement if the data has a negative sign, then entering the binary point as specified by the scaling, and finally converting the data from binary to decimal. This procedure is reversed to convert data from decimal to DEDA octal form for entry via the DEDA.

The binary scaling which determines the placement of the binary point specifies the range of a variable. For example, a variable whose maximum value is 0.01 which is less than  $0.015625 = 2^{-6}$  may be scaled at binary -6, which is designated B-6. The most significant bit of the

variable is then equal to  $2^{-7} = 0.0078125$  and the range is actually  $\pm(2^{-6} - 1 \text{ LSB})$ . This scaling definition is applied to the DEDA conversions in the following examples.

**Example 1. - Decimal-to-DEDA octal conversion**

Given  $\sin \delta_L = -.5625$

Conversion to binary gives

$$-.5625|_{10} \rightarrow -.100100|_2 \text{ exactly}$$

Shift the binary point one place to the left, since  $\sin \delta_L$  is scaled at B1, to get the scaled binary number

$$-.0100100|_2$$

If the number to be input is negative, the scaled binary number must be converted to the 2's complement number. This may be done by inverting all bits to the left of the first least significant bit which is a one, resulting in

$$-.1011100|_2$$

The 2's complement number is now converted to octal form.

$$-.1011100|_2 \rightarrow -.56000|_8 \text{ exactly}$$

The number  $-56000$  is then the DEDA octal equivalent of  $-.5625|_{10}$  for  $\sin \delta_L$ .

**Example 2. - Decimal-to-DEDA Octal Conversion**

Given that  $1K22 = .00354|_{10}$

Conversion to binary gives

$$.00354|_{10} \rightarrow .000000001110011111111111010|_2$$

Next shift the binary point 8 places to the right since  $1K22$  is scaled at B-8.

$$.1110011111111111010|_2$$

The binary number is truncated at 15 bits since the DEDA only displays 5 octal digits.

$$.111001111111111$$

## Converting to octal for DEDA

$$+.111001111111111|_2 \rightarrow +71777|_8$$

The number +71777 is then the DEDA octal of  
+.00354|<sub>10</sub> for 1K22

## Example 3. - DEDA Octal-to-Decimal Conversion

When converting an octal DEDA readout, the procedure is the reverse of what it is for converting to a DEDA input.

Given the DEDA readout of 1K13 = +00200|<sub>8</sub>

Converting to binary

$$+00200|_8 \rightarrow +.00000010000000|_2$$

Since 1K13 is scaled at binary -7, shift the point 7 bits to the left to get

$$+.00000000000001000000|_2 \rightarrow 2^{-15}|_{10}$$

Hence a DEDA readout of +00200 for 1K13 is equivalent to

$$+.00003052|_{10}$$

**CAUTION:** No new value should be entered for any of the listed output parameters! The only safeguard check made by the program on the address entered via DEDA is to assure that it is a legal address, i. e., in the DEDA accessible region.

The values of the various "S" switches are shown in Table 9.4. These are listed separately from the other DEDA inputs to show their coded form.

Table 9.1. DEDA Accessible Region

Octal Region	Decimal Display Quantization*	DEDA Assumed Internal Computer Units and Binary Scaling*
026 - 172	Octal	---
173 - 253	100 ft	ft at B23
254 - 257	0.1 min	sec at B18
260 - 271	0.1 fps	fps at B13
272 - 276	0.1 min	sec at B18
277 - 305	0.01 deg	rad at B3
306 - 313	0.01 min	sec at B13
314 - 337	0.1 nmi	ft at B23
340 - 351	100 ft	ft at B23
352 - 371	0.1 fps	fps at B13
372 - 377	0.1 min	sec at B18
400 - 401	Octal	---
402 - 405	0.1 nmi	ft at B23
406 - 417	Octal	---
420 - 503	0.1 fps	fps at B13
504 - 537	Octal	---
540 - 543	0.001 ft/sec <sup>2</sup>	fps/20 msec at B1
544 - 546	0.01 deg/hr	rad/20 msec at B-13
547 - 613	Octal	---
614 - 622	1. Count	Counts at B17
623 - 704	Octal	---

\*Scale factors stored in FP6 are used to convert to the Display Quantizations on the basis of the specified scaling and dimensional units. The list holds for all parameters listed in the DEDA Input and DEDA Output Tables. Within a given computer memory region, certain parameters may be stored at binary scalings other than shown above and also may have different dimensional units; hence, these quantities are not intended to be DEDA accessible.

Table 9.2. DEDA Inputs

Symbol	Quantization	Range <sup>(a)</sup>	Address	Definition
1J1	100 ft	±8,388,500 ft	240	LM Initial Position, x component
1J2	100 ft	±8,388,500 ft	241	LM Initial Position, y component
1J3	100 ft	±8,388,500 ft	242	LM Initial Position, z component
1J4	0.1 ft/sec	±8191.9 ft/sec	260	LM Initial Velocity, x component
1J5	0.1 ft/sec	±8191.9 ft/sec	261	LM Initial Velocity, y component
1J6	0.1 ft/sec	±8191.9 ft/sec	262	LM Initial Velocity, z component
1J7	0.1 min	4369 min	254	LM Epoch Time (AGS referenced)
2J1	100 ft	±8,388,500 ft	244	CSM Initial Position, x component
2J2	100 ft	±8,388,500 ft	245	CSM Initial Position, y component
2J3	100 ft	±8,388,500 ft	246	CSM Initial Position, z component
2J4	0.1 ft/sec	±8191.9 ft/sec	264	CSM Initial Velocity, x component
2J5	0.1 ft/sec	±8191.9 ft/sec	265	CSM Initial Velocity, y component
2J6	0.1 ft/sec	±8191.9 ft/sec	266	CSM Initial Velocity, z component
2J7	0.1 min	4369 min	272	CSM Epoch Time (AGS referenced)
TA1	0.1 min	4369 min	377	Absolute Time
28J1	0.1 ft/sec	±8191.9 ft/sec	450	Component of External $\Delta V$ input in Downrange Direction
28J2	0.1 ft/sec	±8191.9 ft/sec	451	Component of External $\Delta V$ input in Minus Crossrange Direction
28J3	0.1 ft/sec	±8191.9 ft/sec	452	Component of External $\Delta V$ input in Minus Radial Direction

Table 9.2. DEDA Inputs (Continued)

Symbol	Quantization	Range <sup>(a)</sup>	Address	Definition
1J	0.1 min	4369 min	275	Desired TPI Time for CSI Computation
2J	B7 <sup>(b)</sup>	N.A.	605	Cotangent of Desired LOS angle at TPI Time for CSI Computation
3J	0.01 min	136.5 min	312	Rendezvous Offset Time
4J	0.01 min	136.5 min	306	Time of Node Prior to Rendezvous
5J	100 ft	8,388,500 ft	231	Nominal Landing Site Radius
6J	0.01 min	136.5 min	307	Desired LM Transfer Time for Direct Intercept Transfer Routine
7J	100 ft	8,388,500 ft	224	Term in LM Semi-major Axis $\alpha_L$ Calculation (OI)
8J	100 ft	8,388,500 ft	225	Lower Limit of $\alpha_L$ (OI)
9J	100 ft	8,388,500 ft	226	Upper Limit of $\alpha_L$ (OI)
16J	100 ft	8,388,500 ft	232	Orbit Insertion Targeted Injection Altitude
17J	0.1 ft/sec	$\pm 8191.9$ ft/sec	503	Radar Range Rate
18J	0.1 nm	1379.7 nm	316	Radar Range
21J	100 ft	8,388,500 ft	233	Vertical Pitch Steering Altitude Threshold
22J	0.1 ft/sec	$\pm 8191.9$ ft/sec	464	Vertical Pitch Steering Altitude Rate Threshold
23J	0.1 ft/sec	$\pm 8191.9$ ft/sec	465	Orbit Insertion Targeted Injection Radial Rate
25J	100 ft	8,388,500 ft	223	DEDA entry for Altitude Update
29J	0.1 min	0 to -4396 min	274	Radar filter update time initialization value

Table 9.2. DEDA Inputs (Continued)

Symbol	Quantization	Range <sup>(a)</sup>	Address	Definition
$T_{\Delta}$	0.01 min	136.5 min	310	$\Delta t$ Time Increment from Present Time (now) to next Maneuver
$t_{ig}$	0.1 min	4369 min	373	Desired Absolute Time of Next Maneuver
$\sin \delta_L$	B1 <sup>(b)</sup>	N.A.	047	Sine of the Landing Azimuth Angle
$\cos \delta_L$	B1 <sup>(b)</sup>	N.A.	053	Cosine of the Landing Azimuth Angle
$\Delta_{\delta}$	B0 <sup>(b)</sup>	N.A.	547	Lunar Align Azimuth Correction
$W_{bx}$	B1 <sup>(b)</sup>	N.A.	514	Unit Vector for Yaw Steering when in Guidance Steering Mode ( $S_{00} = 1$ )
$W_{by}$	B1 <sup>(b)</sup>	N.A.	515	
$W_{bz}$	B1 <sup>(b)</sup>	N.A.	516	

- (a) Inputting a quantity greater than the range shown results in computer overflow, with the stored number being arbitrary. Subsequent computations will be meaningless. Note that overflow can be detected by reading out the number loaded.
- (b) Octal input. The numeral after B indicates the internal binary scaling. The decimal quantity must be entered in the proper octal representation which reflects the scaling. No programmed conversion is made.

Table 9.3. DEDA Outputs

Symbol	Quantization	Range	Address	Definition
$r_x$	100 ft	$\pm 8,388,500$ ft	340	X component of LM Position
$r_y$	100 ft	$\pm 8,388,500$ ft	341	Y component of LM Position
$r_z$	100 ft	$\pm 8,388,500$ ft	342	Z component of LM Position
$V_x$	0.1 ft/sec	$\pm 8191.9$ ft/sec	360	X component of LM Velocity
$V_y$	0.1 ft/sec	$\pm 8191.9$ ft/sec	361	Y component of LM Velocity
$V_z$	0.1 ft/sec	$\pm 8191.9$ ft/sec	362	Z component of LM Velocity
$V$	0.1 ft/sec	8191.9 ft/sec	433	LM Velocity Magnitude
$r_{cx}$	100 ft	$\pm 8,388,500$ ft	344	X component of CSM Position
$r_{cy}$	100 ft	$\pm 8,388,500$ ft	345	Y component of CSM Position
$r_{cz}$	100 ft	$\pm 8,388,500$ ft	346	Z component of CSM Position
$V_{cx}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	364	X component of CSM Velocity
$V_{cy}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	365	Y component of CSM Velocity
$V_{cz}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	366	Z component of CSM Velocity
$t$	0.1 min	4369 min	377	Absolute Time
$q_{LT}$	0.1 nm	$\pm 1379.7$ nm	403	Pericynthion Altitude of Present LM Trajectory
$q_{1d}$	0.1 nm	$\pm 1379.7$ nm	402	Pericynthion Altitude of Predicted LM Transfer Trajectory (TPI Mode Only)

Table 9.3. DEDA Outputs (Continued)

Symbol	Quantization	Range	Address	Definition
$\Delta V_{gX}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	500	Body Axis Components of Velocity-to-be-Gained to Engine Shutdown
$\Delta V_{gY}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	501	
$\Delta V_{gZ}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	502	
$V_{DX}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	470	Accumulated Total Sensed Velocity Increment in X Body Direction
$V_{DY}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	471	Accumulated Total Sensed Velocity Increment in Y body Direction
$V_{DZ}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	472	Accumulated Total Sensed Velocity Increment in Z body Direction
$V_{p0}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	371	$\Delta V$ for CDH Maneuver (CSI Only)
$\Delta r$	0.1 nm	$\pm 1379.7$ nm	402	Predicted Differential Altitude after CDH
$\delta r$	0.1 nm	$\pm 1379.7$ nm	314	Differential Altitude at $t_{ig}$ (CSI or CDH only)
$\xi$	0.01 deg	360 deg	277	Inplane angle between Z-body Axis and the Local Horizontal
$T_{A0}$	0.1 min	4369 min	372	Coast Time from CSI to CDH (CSI only)
R	0.1 nm	1379.7 nm	317	Range from LM to CSM
$\dot{R}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	440	Range Rate
$\dot{r}$	0.1 ft/sec	$\pm 8191.9$ ft/sec	367	LM Altitude Rate
h	0.1 nm	1379.7 nm	337	LM Altitude
$V_T$	0.1 ft/sec	8191.9 ft/sec	371	TPI Plus Breaking Impulse $\Delta V$ (TPI Mode Only)
$r_f$	100 ft	8,388,500 ft	347	Predicted LM Orbit Radial Distance at Maneuver; at Completion of Maneuver for OI

Table 9. 3. DEDA Outputs (Continued)

Symbol	Quantization	Range	Address	Definition
y	100 ft	8, 388, 500 ft	211	Out-of-CSM Orbit Plane Displacement
$\delta_{21}$	N. A.	N. A.	604	Lunar Surface Flag
1K1	0. 01 deg/hr	$\pm 10$ deg/hr	544	X-Axis Gyro Drift Compensation Coefficient
1K6	0. 01 deg/hr	$\pm 10$ deg/hr	545	Y-Axis Gyro Drift Compensation Coefficient
1K11	0. 01 deg/hr	$\pm 10$ deg/hr	546	Z-Axis Gyro Drift Compensation Coefficient
1K19	0. 001 ft/sec <sup>2</sup>	$\pm 0. 064$ ft/sec <sup>2</sup>	540	X-Axis Accelerometer compensation
1K21	0. 001 ft/sec <sup>2</sup>	$\pm 0. 064$ ft/sec <sup>2</sup>	541	Y-Axis Accelerometer compensation
1K23	0. 001 ft/sec <sup>2</sup>	$\pm 0. 064$ ft/sec <sup>2</sup>	542	Z-Axis Accelerometer compensation
1K9	2-sec count	131072 counts	616	Ullage counter limit
$V_{y0}$	0. 1 ft/sec	$\pm 8191. 9$ ft/sec	270	LM Out-of-CSM-Plane Velocity
$\delta_2$	N. A.	N. A.	574	Staging Flag
$J^4$	0. 01 min	136. 5 min	306	Time of Node Prior to Rendezvous
$T_{perg}$	0. 01 min	136. 5 min	313	Time to Perigee
$T_{\Delta}$	0. 01 min	136. 5 min	310	$\Delta t$ Time Increment from Present Time to Next Maneuver
$V_{py}$	0. 1 ft/sec	$\pm 8191. 9$ ft/sec	263	Out-of-CSM Plane Velocity at $t_{ig}$ in CSI, CDH, or TPI and at present in OI
$\Delta V_G$	0. 1 ft/sec	8191. 9 ft/sec	267	Magnitude of LM Velocity-to-be-Gained
$\theta_{LOS}$	0. 01 deg	360 deg	303	Predicted LOS Angle at $t_{ig}$ (TPI Mode Only)
$\dot{r}_A$	0. 1 ft/sec	$\pm 8191. 9$ ft/sec	477	Radial Velocity at $t_{ig}$ in CSI, CDH, TPI, and at present in OI

Table 9.4. "S" Selectors

Symbol	Code	Address	Definition	
$S_{00}$	= 0	+00000	400	Attitude hold
	= 1	+10000		Guidance steering
	= 2	+20000		Z axis steering
	= 3	+30000		PGNCS/AGS Align
	= 4	+40000		Lunar align
	= 5	+50000		Body axis align
	= 6	+60000		Inflight calibrate
	= 7	+70000		Accelerometer calibrate
$S_{07}$	= 0	+00000	407	Do not freeze velocity-to-be-gained vector or accumulate sensed velocity increments
	= 1	+10000		Freeze velocity-to-be-gained vector in AGS inertial frame and accumulate sensed velocity increments
$S_{10}$	= 0	+00000	410	Orbit insertion
	= 1	+10000		Coelliptic sequence initiate
	= 2	+20000		Constant delta "h"
	= 3	+30000		TPI search
	= 4	+40000		TPI execute
	= 5	+50000		External $\Delta V$
$S_{11}$	= 0	+10000	411	Select cant angle corrections to guidance steering attitude errors
	= 1	+00000		Do not select cant angle corrections
$S_{12}$	= 0	+00000	412	Entry-initialize test, Readout-test not completed
	= 1	+10000		Test successfully completed
	= 3	+30000		Logic test failure
	= 4	+40000		Memory test failure
	= 7	+70000		Logic and memory test failures
$S_{13}$			413	Any entry into $S_{13}$ will store lunar aximuth and set lunar surface flag $\delta_{21}$

Table 9.4. "S" Selectors (Continued)

Symbol	Code	Address	Definition
$S_{14} = 0$	+00000	414	Initialization Complete Indication
$= 1$	+10000		Initialize LM and CSM via Downlink
$= 2$	+20000		Initialize LM via DEDA
$= 3$	+30000		Initialize CSM via DEDA
$S_{15}$		415	Any Entry into $S_{15}$ will Store Z-Axis Direction Cosines, Range Vector, Range Rate, and time since the Last Radar Update
$S_{16} = 1$	+10000	416	Compute CDH Solution at CSI time plus 1/2 LM Orbit Period
$= 3$	+30000		Compute CDH Solution at CSI time plus 3/2 LM Orbit Period
$S_{17} = 1$	+10000	417	Initialize Radar Filter
$S_{507} = 0$	+00000	507	In Z-Axis Steering Mode ( $S_{00} = 2$ ), Steer Z-Body Axis in Direction of CSM
$= 1$	+10000		In Z-Axis Steering Mode, Steer Z-Body Axis in Desired Thrust Direction
$S_{623} = 0$	+00000	623	In Guidance Steering ( $S_{00} = 1$ ), Steer Z-Body Axis Parallel to CSM Orbit Plane
$= 1$	+10000		In Guidance Steering, Steer Z-Body Axis Parallel to Plane Defined by its Unit Normal Vector $\underline{W}_b$

### 10. DEFINITIONS OF CONSTANTS

The constants contained in the program either explicitly or implicitly are listed below in Table 10.1. Included are their memory addresses (if they appear explicitly in the program), definition, computer quantized value in LM AGS FP6 S03, range and quantization.

Table 10.1. Equation Constants

Constant	Address	Definition and Internal Computer Units	Value	Range <sup>†</sup>	Quantization
1K1	544	X-axis gyro drift compensation constant (rad/20 ms)	0	$\pm 2^{-13}$	$2^{-30}$
*1K2		Gyro scale factor (rad/pulse)	$2^{-16}$		
1K3	550	Compensation constant for the attitude rate scale factor error of X gyro (no units)	0	$\pm 2^{-7}$	$2^{-24}$
1K4	624	Constant in altitude and altitude rate 200-msec readout (no units)	0.1	$\pm 2^0$	$2^{-17}$
*1K5		Constant in altitude and altitude rate 200-msec readout (no units)	0.2		
1K6	545	Y-axis gyro drift compensation constant (rad/20 ms)	0	$\pm 2^{-13}$	$2^{-30}$
*1K7		Gyro and accelerometer null bias (pulses)	640		
1K8	551	Compensation constant for the attitude rate scale factor error of Y gyro (no units)	0	$\pm 2^{-7}$	$2^{-24}$
1K9	616	Number of 2-sec cycles for ullage	3	$\pm 2^{17}$	$2^0$
1K11	546	Z-axis gyro drift compensation constant (rad/20 ms)	0	$\pm 2^{-13}$	$2^{-30}$
1K13	552	Compensation constant for the attitude rate scale factor error of Z gyro (no units)	0	$\pm 2^{-7}$	$2^{-24}$
1K14	537	Compensation constant for X gyro spin axis mass unbalance drift (rad/ft/sec)	0	$\pm 2^{-14}$	$2^{-31}$

Table 10.1. Equation Constants (Continued)

Constant	Address	Definition	Value	Range <sup>†</sup>	Quantization
1K18	534	Compensation and scale factor constant for X-axis accelerometer (fps/pulse)	0.003125	$\pm 2^{-8}$	$2^{-25}$
1K19	540	Compensation constant for X-axis accelerometer bias (fps)	0	$\pm 2^{-6}$	$2^{-16}$
1K20	535	Compensation and scale factor constant for Y-axis accelerometer (fps/pulse)	0.003125	$\pm 2^{-8}$	$2^{-25}$
1K21	541	Compensation constant for Y accelerometer bias (fps)	0	$\pm 2^{-6}$	$2^{-16}$
1K22	536	Compensation and scale factor constant for Z-axis accelerometer (fps/pulse)	0.003125	$\pm 2^{-8}$	$2^{-25}$
1K23	542	Compensation constant for Z-axis accelerometer bias (fps)	0	$\pm 2^{-6}$	$2^{-16}$
1K24	625	FDAI computation singularity region (no units)	0.00086975	$\pm 2^1$	$2^{-16}$
*1K25		PGNCS Euler angle scale factor (rad/pulse)	$\frac{6.2831853}{2^{15}}$		
1K26	626	X-axis alignment gain constant (no units)	-142.857	$\pm 2^8$	$2^{-9}$
1K27	627	Lunar align constant (rad/ft/sec)	0.0434999	$\pm 2^{-4}$	$2^{-21}$
1K28	630	Lunar align constant (no units)	107.9375	$\pm 2^7$	$2^{-10}$
1K29	631	Lunar align constant (rad)	0.0009999	$\pm 2^{-4}$	$2^{-21}$

Table 10.1. Equation Constants (Continued)

Constant	Address	Definition	Value	Range†	Quantization
1K30	617	Gyro calibrate time (2 sec)	150	$2^{17}$	$2^0$
1K33	632	Gyro calibrate gain constant (no units)	0.079999	$\pm 2^{-3}$	$2^{-20}$
1K34	633	Gyro calibrate gain constant (1/20 ms)	0.000019999	$\pm 2^{-15}$	$2^{-32}$
1K35	634	Navigation sensed velocity threshold (fps)	0.25	$2^3$	$2^{-10}$
1K36	635	Accelerometer calibrate gain constant (no units)	$-0.663757 \times 10^{-3}$	$\pm 2^0$	$2^{-17}$
1K37	621	Accelerometer calibrate time (2 sec)	15	$2^{17}$	$2^0$
1K56	673	Negative of the product of lunar rotation rate and 20-msec compute cycle period, $\Omega_M \Delta t$ (rad)	$-0.53085 \times 10^{-7}$	$\pm 2^{-14}$	$2^{-31}$
2K1	636	Lunar gravitational constant ( $\text{ft}^3/\text{sec}^2$ )	$0.173188 \times 10^{15}$	$\pm 2^{48}$	$2^{31}$
2K2	637	Reciprocal of $K_1^2$ ( $\text{sec}^2/\text{ft}^3$ )	$0.57740 \times 10^{-14}$	$\pm 2^{-47}$	$2^{-64}$
2K3	216	q value if overflow occurs in LM $e^2$ (ft)	$1.048576 \times 10^6$	$\pm 2^{23}$	$2^6$
2K4	674	$-2K1 \Delta t$ ( $\Delta t = 2$ sec) ( $\text{ft}^3/\text{sec}$ )	$-0.34637 \times 10^{15}$	$\pm 2^{49}$	$2^{32}$
2K11	526	Set value of $V_T$ (fps)	6000	$\pm 2^{13}$	$2^{-4}$

Table 10.1. Equation Constants (Continued)

Constant	Address	Definition	Value	Range <sup>†</sup>	Quantization
2K14	217	Initial p perturbation (ft)	4.9984 $\times 10^4$	$2^{23}$	$2^6$
2K17	620	Controls iterations of p-iterator (number of iterations -3)	5	$2^{17}$	$2^0$
2K18	447	Partial derivative, $\partial_T$ , protector (sec)	15	$\pm 2^{13}$	$2^{-4}$
2K19	230	( $\Delta_p$ ) limiter (ft)	5.00032 $\times 10^5$	$\pm 2^{23}$	$2^6$
2K20	453	p-iterator convergence check (sec)	2	$\pm 2^{13}$	$2^{-4}$
3K4	613	Sine of Central angle limit in TPI (no units)	0.173645	$2^0$	$2^{-16}$
4K2	654	Factor in $T_B$ computation (sec/ft)	-0.49928 $\times 10^{-4}$	$\pm 2^{-12}$	$2^{-29}$
4K3	655	Factor in $T_B$ computation ( $\text{sec}^2/\text{ft}^2$ )	0.12464625 $\times 10^{-8}$	$\pm 2^{-25}$	$2^{-42}$
4K4	565	Coefficient in linear expression for $\dot{r}_f$ as a function of $r_f$ ( $\text{sec}^{-1}$ )	0.004	$\pm 2^{-7}$	$2^{-24}$
4K5	662	Constant in linear expression for $\dot{r}_f$ as a function of $r_f$ (ft)	5742400	$\pm 2^{23}$	$2^6$
4K6	527	Upper limit on $\dot{r}_f$ (fps)	80	$\pm 2^{13}$	$2^{-4}$
4K7	566	Cant angle of engine in pitch plane (rad)	0.08730316	$\pm 2^0$	$2^{-17}$
4K8	602	Cant angle of engine in roll (X-Y) plane (rad)	0.0339966	$\pm 2^0$	$2^{-17}$

Table 10.1. Equation Constants (Continued)

Constant	Address	Definition	Value	Range†	Quantization
4K10	227	Factor in LM desired semimajor axis $a_L$ (OI) (fps/rad)	-651360	$\pm 2^{20}$	$2^3$
4K12	506	Acceleration check for RD3DTL in OI (fps <sup>2</sup> )	5.0	$2^7$	$2^{-10}$
4K21	666	Limit on body attitude errors (rad)	0.261810	$2^2$	$2^{-15}$
4K23	622	Staging time (1 count/40 msec) to maintain attitude hold momentarily after staging (counts)	25	0 to $2^{17}$	$2^0$
4K25	657	Ascent engine cutoff impulse compensation (fps)	2.125	$2^{13}$	$2^{-4}$
4K26	454	$V_G$ threshold on engine cutoff computations (fps)	100	$2^{13}$	$2^{-4}$
4K27	473	Descent stage $\Delta V$ capability (fps)	-7332	$\pm 2^{13}$	$2^{-4}$
4K34	660	Lower limit on $a_T$ (fps <sup>2</sup> )	1	$\pm 2^7$	$2^{-10}$
4K35	661	Ullage threshold (fps <sup>2</sup> )	0.099609	$2^7$	$2^{-10}$
5K14	560	Upper limit on $\ddot{r}_d$ (fps <sup>3</sup> )	0	$\pm 2^{-2}$	$2^{-19}$
5K16	561	Upper limit on $\ddot{y}_d$ (fps <sup>3</sup> )	0.0079994	$\pm 2^{-2}$	$2^{-19}$
5K17	601	Lower limit on $\dot{y}_d$ (fps <sup>3</sup> )	-0.0079994	$\pm 2^{-2}$	$2^{-19}$
5K18	564	$\ddot{r}_d$ lower limit (fps <sup>3</sup> )	-0.1	$\pm 2^{-2}$	$2^{-19}$
5K20	523	$\dot{r}_d$ lower limit (fps <sup>3</sup> )	0	$\pm 2^{-2}$	$2^{-19}$

Table 10.1. Equation Constants (Continued)

Constant	Address	Definition	Value	Range <sup>†</sup>	Quantization
6K2	457	Radar filter initialization value of $P_{11}$ and $P_{22}$ (ft <sup>2</sup> )	0.0999997 $\times 10^9$	$2^{30}$	$2^{13}$
6K4	456	Radar filter initialization value of $P_{33}$ and $P_{44}$ (fps) <sup>2</sup>	100	$2^{10}$	$2^{-7}$
6K5	656	Radar filter factor in $r_y$ update (no units)	-0.730003	$\pm 2^0$	$2^{-17}$
6K6	522	Radar filter factor in $V_y$ update (no units)	-0.240234	$\pm 2^8$	$2^{-9}$
6K8	304	Radar filter term in $q_{11}$ (variance of range rate uncertainty) (fps) <sup>2</sup>	0.21875	$2^{10}$	$2^{-7}$
6K9	611	Radar filter factor in $q_{11}$ , $q_{12}$ , and $q_{22}$ (rad <sup>2</sup> ) (variance of gimbal angle uncertainty)	$0.3029 \times 10^{-4}$	$2^{-15}$	$2^{-32}$
6K10	517	Radar filter factor in $q_{11}$ and $q_{22}$ (variance of range uncertainty) (ft <sup>2</sup> )	0.62505 $\times 10^7$	$2^{28}$	$2^{11}$
5K26	466	Velocity-to-be-gained threshold (fps)	15	$\pm 2^{13}$	$2^{-4}$
K55	607	Scale factor for $\dot{h}$ display (no units)	0.99999237	$2^0$	$2^{-17}$
$W_{bx}$	514	Unit vector in AGS inertial coordinates, normal to the plane in which the Z-body axis is commanded during guidance steering ( $S_{00} = 1$ ) when $S_{623} = 1$ (no units)	0	$2^1$	$2^{-16}$
$W_{by}$	515		0	$2^1$	$2^{-16}$
$W_{bz}$	516		0	$2^1$	$2^{-16}$
<sup>†</sup> Some range values reflect equation limitations which are less than that permitted by their scaling. *These constants do not explicitly appear in the program.					

REFERENCES

1. "LM AGS Computer Program Specification, Flight Program Number 6" 11176-6042-T000, TRW Systems, Redondo Beach, California, March 1969.
2. "LM AGS Operating Manual, Flight Program Number 6," 11176-6033-T000, TRW Systems, Redondo Beach, California, April 1969.
3. T. S. Bettwy, "LM AGS Flight Equations Narrative Description," 05952-6076-T000, TRW Systems, Redondo Beach, California, 25 January 1967.
4. 05952-6214-T000, "LM/AGS Radar Filter Improvement Feasibility Study," TRW Systems, Redondo Beach, California, June 1968.
5. TRW Document No. 11176-6024-T000, "FPX CSI/CDH Equations Including Test Results," by R. L. Eshbaugh, dated 20 December 1968 (U).
6. MIT/IL R-567, "Guidance System Operations Plan for Manned LM Earth Orbital and Lunar Missions using Program Luminary, Section 5, Guidance Equations (Revision 1)," November 1968.